

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

CAMILA FERREIRA PEREIRA SILVA

**Um estudo sobre a automação de testes de acessibilidade em dispositivos
móveis: uma análise de recomendações para deficiência visual da BBC**

São Paulo

2020

CAMILA FERREIRA PEREIRA SILVA

**Um estudo sobre a automação de testes de acessibilidade em dispositivos
móveis: uma análise de recomendações para deficiência visual da BBC**

Versão original

Dissertação de mestrado apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo como parte dos requisitos para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de concentração: Engenharia de Software

Orientador: Prof. Dr. Marcelo Medeiros Eler

São Paulo

2020

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

CATALOGAÇÃO-NA-PUBLICAÇÃO

(Universidade de São Paulo. Escola de Artes, Ciências e Humanidades. Biblioteca)

CRB 8 - 4936

Silva, Camila Ferreira Pereira

Um estudo sobre a automação de testes de acessibilidade em dispositivos móveis: uma análise de recomendações para deficiência visual da BBC / Camila Ferreira Pereira Silva ; orientador, Marcelo Medeiros Eler. – 2020.

83 f. : il

Dissertação (Mestrado em Ciências) - Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo.

Versão original

1. Teste e avaliação de software. 2. Deficiência visual - Programação e programas; Testes. I. Eler, Marcelo Medeiros, orient. II. Título

CDD 22. ed. – 005.14

Dissertação de autoria de Camila Ferreira Pereira Silva, sob o título “**Um estudo sobre a automação de testes de acessibilidade em dispositivos móveis: uma análise de recomendações para deficiência visual da BBC**”, apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Metodologia e Técnicas da Computação, aprovada em _____ de _____ de _____ pela comissão julgadora constituída pelos doutores:

Prof. Dr. -----
Instituição: -----
Presidente

Prof. Dr. -----
Instituição: -----

Prof. Dr. -----
Instituição: -----

Prof. Dr. -----
Instituição: -----

Agradecimentos

Primeiramente gostaria de agradecer a Deus e a minha família que apoiou a minha mudança para São Paulo a fim de realizar esse mestrado, especialmente a minha mãe, Sandra Pereira, que me apoiou de todas as formas, além de ter me ajudado nos diversos momentos que pensei em desistir.

Gostaria de agradecer também ao meu orientador Marcelo Eler que compartilhou comigo um pouco dos seus ensinamentos me ajudando a fazer com que essa dissertação e os artigos que foram apresentados se tornasse realidade.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

SILVA, Camila Ferreira Pereira. **Um estudo sobre a automação de testes de acessibilidade em dispositivos móveis:** uma análise de recomendações para deficiência visual da BBC. 2020. 82 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2020.

Acessibilidade digital é a capacidade de um software poder ser usado pelo maior número possível de pessoas, incluindo usuários com algum tipo de deficiência, seja física ou mental. O grande esforço necessário para avaliar se um software atende às diretrizes mínimas de acessibilidade motivou o desenvolvimento de diversas ferramentas de automação. Entretanto, não existe ainda nenhum estudo abrangente sobre quais recomendações de acessibilidade podem ou não ser avaliadas automaticamente para dispositivos móveis, apenas estudos que mostram quais diretrizes são automatizadas no desenvolvimento web. Portanto, o objetivo deste projeto de pesquisa é realizar um estudo abrangente para identificar, entre as recomendações, a possibilidade de se fazer avaliação automática.

Palavras-chaves: Acessibilidade; Testes automatizados; Aplicações móveis.

Abstract

SILVA, Camila Ferreira Pereira. **A study of the automation of mobile accessibility tests:** an analysis of BBC's recommendations for visual impairment. 2020. 82 p.
Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2020.

Digital accessibility is the software capability to be used by the most possible number of people, including users with some kind of disability, either physical or mental. The great effort required to evaluate whether a software meets basic accessibility guidelines motivated the development of several automation tools. However, there has not been yet any comprehensive study about which accessibility guidelines can be automatically evaluated. Therefore, this research project aims at conducting a comprehensive study to identify, among the guidelines, the possibility of performing automatic evaluation.

Keywords: Accessibility, Automated tests, Mobile.

Lista de figuras

Figura 1 – Quantidade de artigos que responde cada questão de pesquisa.	26
Figura 2 – Distribuição dos artigos conforme os anos de publicação.	26
Figura 3 – Graduação de contraste de cores.	37
Figura 4 – Exemplo de violação de contraste de cores.	38
Figura 5 – Exemplo de violação de Cor e Significado.	40
Figura 6 – Exemplo de violação de tamanho de item clicável.	42
Figura 7 – Tamanho de item clicável analisado pelo Accessibility Scanner.	43
Figura 8 – Exemplo de violação de espaçamento.	44
Figura 9 – Exemplo de violação de conteúdo dimensionável.	45
Figura 10 – Exemplo de violação de elementos acionáveis.	47
Figura 11 – Exemplo de violação de foco visível.	48
Figura 12 – Violação de consistência.	50
Figura 13 – Violação da recomendação Flicker.	52
Figura 14 – Violação de foco manuseável.	57
Figura 15 – Botões com o mesmo nome e sem descrição adicional para diferenciar a ação de cada um.	59
Figura 16 – Botões com o mesmo nome e sem descrição adicional para diferenciar a ação de cada um.	60
Figura 17 – Exemplo de violação de links repetidos.	62
Figura 18 – Exemplo de violação da apresentação de mensagem de erro em formulários.	64
Figura 19 – Item não-textual sem alternativa de descrição por texto	67
Figura 20 – Exemplo de violação de conteúdo decorativo mencionado por leitor de tela	68
Figura 21 – Exemplo de violação de dicas e informações complementares	70
Figura 22 – Exemplo de violação de tarefas, marcas e propriedades	72

Lista de tabelas

Tabela 1 – Termos Utilizados na pesquisa	25
Tabela 2 – Artigos utilizados na revisão sistemática	27
Tabela 3 – Diretrizes de acessibilidade cobertas pelas ferramentas.	33
Tabela 4 – Lista de tópicos, recomendações e rótulos atribuídos: automatizável (SIM), não-automatizável (NÃO) ou parcialmente automatizável (PARC)	36

Sumário

1	Introdução	11
1.1	<i>Motivação</i>	12
1.2	<i>Objetivos</i>	13
1.3	<i>Justificativa</i>	13
1.4	<i>Metodologia</i>	14
1.5	<i>Organização do documento</i>	14
2	Revisão de literatura	15
2.1	<i>Padrões de acessibilidade para web</i>	15
2.2	<i>Padrões de acessibilidade para dispositivos móveis</i>	16
2.3	<i>Avaliação de acessibilidade</i>	21
2.4	<i>Trabalhos relacionados</i>	23
3	Ferramentas que analisam diretrizes de acessibilidade	24
3.1	<i>Protocolo</i>	24
3.2	<i>Artigos encontrados</i>	25
3.3	<i>Ampliação das bases de pesquisa</i>	26
3.4	<i>Ferramentas de avaliação automática de acessibilidade digital</i>	27
3.4.1	Android	28
3.4.2	iOS	29
3.4.3	Windows Phone	29
3.5	<i>Categorização das ferramentas de análise de acessibilidade</i>	30
3.6	<i>Diretrizes de acessibilidade cobertas pelas ferramentas</i>	31
4	Guia de automação de testes de acessibilidade para deficiências visuais	34
4.1	<i>Introdução</i>	34
4.2	<i>Design</i>	37
4.2.1	Contraste de cores	37
4.2.2	Cor e significado	39
4.2.3	Estilo e leitura	41

4.2.4	Tamanho de itens clicáveis	41
4.2.5	Espaçamento	43
4.2.6	Conteúdo dimensionável	45
4.2.7	Elementos acionáveis	46
4.2.8	Foco visível	48
4.2.9	Consistência	49
4.2.10	Escolha	50
4.2.11	Ajustabilidade	51
4.2.12	Flicker	52
4.3	<i>Formulários</i>	53
4.3.1	Rótulos dos controles dos formulários	53
4.3.2	Entrada de formulários	54
4.3.3	Layout do formulário	55
4.3.4	Agrupamento de elementos	56
4.3.5	Foco manuseável	57
4.4	<i>Links</i>	58
4.4.1	Links descritivos	58
4.4.2	Links para formatos alternativos	60
4.4.3	Combinando links repetidos	61
4.5	<i>Notificações</i>	62
4.5.1	Notificações inclusivas	62
4.5.2	Notificações do sistema operacional	63
4.5.3	Mensagens de erro e correção	64
4.5.4	Feedback e assistência	65
4.6	<i>Textos equivalentes</i>	66
4.6.1	Alternativas para elementos não-textuais	66
4.6.2	Conteúdo decorativo	67
4.6.3	Dicas e informações complementares	69
4.6.4	Papéis, marcas e propriedades	71
4.6.5	Formatação visual	73
4.7	<i>Considerações finais</i>	74
5	Conclusão	75

5.1	<i>Contribuição</i>	75
5.2	<i>Limitações</i>	76
5.3	<i>Trabalhos futuros</i>	76
5.4	<i>Publicações relacionadas a esse trabalho</i>	76
	Referências ¹	78

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 Introdução

Cerca de 15% da população mundial possui algum tipo de deficiência ou enfrenta barreiras para realizar atividades cotidianas (World Bank, 2011), incluindo acesso a informações e serviços eletrônicos. No Brasil, 19% da população possui deficiência visual, 7% possui deficiência motora, 5% possui deficiência auditiva, e 1% possui deficiência mental ou intelectual (IBGE, Acesso em: 22/02/2018). Este cenário torna evidente a necessidade de que a acessibilidade digital seja tratada como atributo de qualidade fundamental no desenvolvimento de qualquer software. Em linhas gerais, a acessibilidade digital refere-se à capacidade de um software de prover recursos e eliminar barreiras para que ele seja acessível por qualquer usuário, independentemente de suas condições físicas, mentais ou intelectuais.

A acessibilidade digital tem sido explorada de forma mais popular no contexto de desenvolvimento de aplicações web, mas a intensificação do uso de dispositivos móveis em todo o mundo fez com que atenção especial fosse dada a esta plataforma no que se refere ao desenvolvimento de aplicações móveis acessíveis (NAYEBI; DESHARNAIS; ABRAN, 2012). Desta forma, várias iniciativas ligadas ao W3C (World Wide Web Consortium), a principal organização internacional que define e mantém padrões para a Internet, começaram a desenvolver ou criar seções com padrões de acessibilidade específicos para aplicações móveis: WCAG¹ (Web Content Accessibility Guidelines), UAAG² (User Agent Accessibility Guidelines), ATAG³ (Authoring Tool Accessibility Guidelines), WAI-ARIA⁴ (Affordable Rich Internet Applications).

Os padrões de acessibilidade digital estabelecem uma série de recomendações para garantir que as funções e o conteúdo digital fornecido pelas aplicações móveis sejam acessíveis a todos os usuários, mas tem foco especial em questões técnicas e conceituais relacionadas à usabilidade da aplicação por parte de usuários com deficiência física, mental ou intelectual. Em geral, os padrões dividem as recomendações em seções específicas, como, por exemplo, formulários, imagens, notificações, áudio e vídeo; e podem estabelecer níveis de severidade ou impacto de cada recomendação, ou até mesmo recomendações obrigatórias ou opcionais.

¹ <https://www.w3.org/WAI/intro/wcag>

² <https://www.w3.org/WAI/intro/uaag.php>

³ <https://www.w3.org/TR/ATAG20/>

⁴ <https://www.w3.org/WAI/intro/aria>

Embora existam diversos padrões de acessibilidade digital e em muitos países a acessibilidade não seja apenas um requisito desejável, mas também obrigatório por lei, muitas aplicações web e móvel ainda apresentam muitos problemas básicos de acessibilidade, inclusive no contexto do governo brasileiro (SERRA et al., 2015; CARVALHO et al., 2016; OLIVEIRA; SOUZA; ELER, 2017; OLIVEIRA; ELER, 2017; ELER et al., 2018; RUBIN; CHISNELL, 2008). A falta de acessibilidade das aplicações pode ter diversas causas: falta de conhecimento dos desenvolvedores (FREIRE; RUSSO; FORTES, 2008), alta demanda e prazos curtos (OLIVEIRA; ELER, 2017), limitação das ferramentas de apoio (VIGO; BROWN; CONWAY, 2013; SILVA; ELER; FRASER, 2018), avaliação onerosa, entre outras.

Em particular, a avaliação da acessibilidade digital em aplicações móveis são custosas porque tendem a ser realizadas de forma manual, seja por meio de técnicas de inspeção conduzidas por especialistas, seja por meio de testes realizados com usuários. Desta forma, além de não incluir acessibilidade digital como prioridade na implementação das aplicações, poucos desenvolvedores e organizações possuem processos apropriados para avaliar se seus produtos são acessíveis.

1.1 Motivação

Neste contexto, abordagens estáticas e dinâmicas que automatizam a avaliação de acessibilidade em aplicações móveis tem sido propostas ao longo do tempo para apoiar a atividade, pois elas permitem que propriedades sejam analisadas em pouco tempo e com muita precisão (SILVA; ELER; FRASER, 2018; ELER et al., 2018). Em alguns casos, a ferramenta pode simplesmente indicar pontos que exigem maior atenção do avaliador.

As abordagens e ferramentas que automatizam a avaliação da acessibilidade em aplicações móveis, entretanto, ainda possuem diversas limitações. Uma das principais limitações é a de que apenas uma pequena porcentagem das recomendações de acessibilidade são avaliadas automaticamente. Uma pesquisa recente (SILVA; ELER; FRASER, 2018) mostrou que a conjunção das propriedades avaliadas por todas as ferramentas estudadas cobrem cerca de 12% das recomendações do padrão da BBC (BBC, 2018), modelo de referência usado pela pesquisa. Isoladamente, a maioria das ferramentas cobrem menos de

8% das propriedades, fato que limita a eficácia da aplicação de abordagens automatizadas em um processo de desenvolvimento.

De fato, muitas recomendações de acessibilidade definidas por diversos padrões são difíceis ou não podem ser avaliadas automaticamente porque não existem critérios objetivos de validação. Em muitos casos, apenas o julgamento de um usuário em contextos específicos pode definir se uma recomendação está sendo seguida ou não. Entretanto, nenhum padrão de acessibilidade define claramente quais recomendações podem ser avaliadas objetivamente, e potencialmente de forma automática, e quais não podem ser avaliadas objetivamente, pelo menos de forma completa. A inexistência de tais critérios impede que o potencial de contribuição das ferramentas de avaliação automática seja atingido e beneficie o processo de desenvolvimento de aplicações acessíveis.

1.2 Objetivos

O objetivo geral deste projeto de pesquisa é definir quais recomendações de acessibilidade podem ser avaliadas automaticamente e definir critérios objetivos de aceitação que possam ser validados por testes gerados automaticamente. Em especial, serão analisadas as recomendações das seções Design, Formulários, Links, Notificações e Textos equivalentes de acordo com o padrão definido pela BBC. Às recomendações serão atribuídos rótulos de acordo com o nível de avaliação automática determinado pelos critérios: automatizável, quando é possível definir automaticamente e de forma autônoma se a recomendação foi ou não seguida; parcialmente automatizável, quando é possível indicar potenciais problemas de acessibilidade, mas uma decisão humana é necessária; e não automatizável, quando a avaliação é totalmente subjetiva e ainda não aplicável quando a diretriz não se aplica a aplicações para Android, que é o foco de avaliação desse trabalho. automática.

1.3 Justificativa

Embora a automação total da avaliação de acessibilidade digital não seja recomendada em razão de suas diversas limitações (VIGO; BROWN; CONWAY, 2013; SILVA; ELER; FRASER, 2018), a automação da avaliação das recomendações de acessibilidade com critérios objetivos de validação pode contribuir com a redução do tempo e esforço

empreendidos na atividade. Ainda que de forma limitada, desenvolvedores e organizações que não dispõem de recursos para avaliação manual poderão ao menos avaliar de forma automática se seus produtos atendem a uma parcela definida de recomendações definidas pelos padrões internacionais. Ainda que de forma limitada, defeitos de acessibilidade simples de serem detectados poderão ser removidos das aplicações.

1.4 Metodologia

Os objetivos deste projeto de pesquisa serão alcançados por meio das seguintes atividades: revisão bibliográfica sobre padrões de acessibilidade digital para web e para dispositivos móveis além de métodos de avaliação de acessibilidade; estudo de ferramentas de automação de avaliação de acessibilidade digital móvel por meio de uma revisão sistemática complementada por um estudo exploratório; análise das recomendações de acessibilidade de maneira mais aprofundada àquelas que foram delimitadas para a definição dos critérios de aceitação e escrita do guia; e por último validação dos critérios de aceitação utilizando a implementação na ferramenta MATE das diretrizes julgadas como automatizáveis e parcialmente automatizáveis.

1.5 Organização do documento

Esse trabalho está organizado de forma que no Capítulo 2 são apresentados alguns conceitos que serão amplamente usados no decorrer desse trabalho como uma visão geral das diretrizes de acessibilidade propostas pelos órgãos que definem padrões e as diretrizes utilizadas de exemplo para esse trabalho, no Capítulo 3 estão as ferramentas que avaliam as diretrizes de acessibilidade verificadas através de revisão sistemática e estudo exploratório, no Capítulo 4 será apresentado o guia de automatização para ferramentas de testes de acessibilidade com foco em deficiências visuais, e no Capítulo 5 mostraremos a Conclusão desse trabalho.

2 Revisão de literatura

O conceito de acessibilidade segundo o dicionário Aurélio é “A qualidade de acessível; facilidade na aproximação, no trato ou na obtenção”. A acessibilidade pode ser considerada como a possibilidade e condição de alcançar os elementos funcionais do ambiente construído, para assim permitir sua utilização (MAZZONI et al., 2001).

Acessibilidade (digital) significa também proporcionar flexibilidade para adaptação às necessidades de cada usuário e a suas preferências e limitações. Acessibilidade também pode ser considerada quando relacionada à web como sinônimo de aproximação, um meio de disponibilizar a cada indivíduo interfaces que respeitem suas necessidades e preferências. Muitas vezes as discussões sobre acessibilidade ficam reduzidas às limitações físicas ou sensoriais dos sujeitos com necessidades especiais, mas esses aspectos podem trazer benefícios a um número bem maior de usuários, por exemplo usuários idosos, permitindo que os conhecimentos disponibilizados na Web possam estar acessíveis a uma audiência muito maior, sem com isso, prejudicar suas características gráficas ou funcionais. Um software ou página Web que não é acessível a uma determinada pessoa tão pouco pode ser considerado eficaz, eficiente ou mesmo agradável a essa pessoa (SANTAROSA, 2002; DIAS, 2007).

2.1 Padrões de acessibilidade para web

Na década de 1990, com a popularização da Internet e de comunidades virtuais nos EUA, percebe-se a necessidade de prover acesso universal na web com a construção de ambientes virtuais acessíveis. Neste sentido, dois consórcios mundiais, o W3C (Consórcio para a Web) e a WAI (Iniciativa para a Acessibilidade na Rede) são os principais responsáveis pela popularização do termo a partir do fim dos anos 1990, quando a Internet, em especial a web, tornou-se mais expressiva, ao estabelecerem os padrões e protocolos que sistemas computacionais deveriam seguir para serem considerados acessíveis (PASSERINO; MONTARDO, 2007).

Os padrões de acessibilidade para web são definidos especificamente pela iniciativa WCAG 2.0¹ (Web Content Accessibility Guidelines), que define como construir aplicações web acessíveis a pessoas com deficiência. O uso das diretrizes para a construção de software

¹ <https://www.w3.org/WAI/intro/wcag>

ajuda a garantir que será mais usável não em casos específicos mas garante, de modo geral, uma melhor experiência no uso do software.

Os padrões de acessibilidade são divididos em 4 princípios básicos e todas as diretrizes de acessibilidade são definidas para satisfazê-los (CONSORTIUM, 2015):

- **Percebível**, que significa que a informação deve ser apresentada na interface de forma que todos os usuários possam perceber.
- **Operável**, que significa que todos os componentes e navegação do aplicativo possam ser operáveis, por exemplo, os usuários devem ser capazes de interagir com o aplicativo.
- **Entendível**, que significa que os usuários devem entender a informação passada em cada interface com o usuário.
- **Robusto**, que significa que o conteúdo em cada interface deve ser robusto o suficiente para que possa ser interpretado pelos usuários e por tecnologias assistivas, por exemplo, leitores de tela.

É possível dizer que se um dos princípios não está satisfeito, usuários com deficiência não são capazes de utilizar a aplicação.

2.2 Padrões de acessibilidade para dispositivos móveis

Os padrões inicialmente definidos pela W3C para web (CONSORTIUM, 2015) não atendiam as especificidades do uso de dispositivos móveis, dessa forma foi necessário evoluir os padrões para então atendê-los. Mas uma vez que é uma evolução das diretrizes já definidas, podem ser aplicados para web também. A WCAG 2.1 mapeou os padrões de acessibilidade móvel nos mesmos quatro princípios citados na Seção 2.1 mas incluiu 17 novas diretrizes mais específicas para dispositivos móveis (SHANLEY, 2016):

Além dos padrões produzidos pela W3C, também existem recomendações de acessibilidade sendo produzidas de maneira independente. Em particular the British Broadcasting Corporation (BBC), um serviço público que criou a BBC Mobile Accessibility Guidelines (BBC, 2018; SHANLEY, 2016). As recomendações de acessibilidade criadas pela BBC são muito compreensíveis e contemplam a maioria das recomendações da W3C para acessibilidade móvel (W3C, 2015). A lista das recomendações de acessibilidade estão separadas por 9 categorias e serão utilizadas como base para análise nesse trabalho por terem sido

criadas já com o propósito de atender as especificidades de aplicações para dispositivos móveis e estarem colocadas de forma mais organizada e clara para entendimento:

- **Áudio e vídeo:**
 - Alternativas para conteúdo de áudio e conteúdo visual: alternativas para títulos e subtítulos por linguagem de sinais, áudio descrição e transcrição devem estar disponíveis.
 - Autoplay: Elementos de áudio não podem começar automaticamente a não ser que o usuário permita, além de estar disponível botões de pause, stop e mute.
 - Metadados: Os metadados relevantes devem estar disponíveis para todas as mídias.
 - Controle de volume: Controle de volume separados devem estar disponíveis caso tenha música de fundo, sons ambientes e narrativas.
 - Conflito de áudio: Narrativas de áudio ou mídias de interação não podem estar em conflito com a narrativa de tecnologias assistivas.

- **Design:**
 - Contraste de cores: O contraste de cores do texto deve ser suficiente.
 - Cor e significado: A informação não pode ser passada apenas pela diferença de cores.
 - Estilo e leitura: O conteúdo principal deve ser acessado quando o estilo principal for removido.
 - Tamanho de itens clicáveis: Os itens clicáveis devem ter o tamanho adequado.
 - Espaçamento: Deve haver um espaço inativo entre os itens clicáveis.
 - Conteúdo dimensionável: Usuários devem poder controlar o tamanho da fonte dos conteúdos apresentados na interface.
 - Elementos acionáveis: Links e outros elementos acionáveis devem estar claramente distinguidos.
 - Foco: Quando focado, todos os elementos devem continuar visíveis.
 - Consistência: A experiência do usuário deve ser consistente.
 - Escolha: As interfaces devem prover múltiplas maneiras de interagir com o conteúdo.
 - Ajustabilidade: Mídias interativas deve ser ajustáveis pelo usuário conforme sua habilidade e preferência.

- Flicker: Conteúdos flash não devem ser intencionalmente acionados mais que 3 vezes por período de 1 segundo.
- Editorial:
 - Títulos consistentes: Títulos consistentes devem ser usados em web sites e aplicações nativas.
 - Indicação de língua: A língua utilizada pela aplicação deve estar especificada, e as mudanças de língua devem ser indicadas.
 - Instruções: Quando necessário, instruções adicionais devem ser dadas para ajuda visual e de áudio.
- Foco:
 - Elementos focáveis: Todos os elementos interativos devem ser focáveis.
 - Armadilhas de teclado: não podem haver armadilhas de teclado.
 - Ordem do conteúdo: A ordem do conteúdo deve ser lógica.
 - Ordem do foco: Conteúdo clicável deve ser navegável em uma sequência entendível.
 - Interações do usuário: Ações devem desencadear outra interação apropriada.
 - Métodos de entrada alternativos: Devem ser suportados métodos de entrada alternativos.
- Formulários:
 - Títulos dos controles dos formulários: Todos os controles do formulários devem conter títulos.
 - Entrada de formulários: O tipo de entrada padrão do formulário deve ser indicada.
 - Layout do formulário: Títulos devem ser colocados perto do controle do formulário.
 - Agrupamento de elementos: Controles, títulos e outros elementos do formulário devem estar agrupados.
 - Foco manuseável: Foco ou contexto não podem mudar automaticamente durante as entradas do formulário.
- Imagens:
 - Imagens de texto: Imagens de texto devem ser evitadas.

- Imagens de fundo: Imagens de fundo que contenham informações ou significado devem conter uma alternativa acessível adicional.
- Links:
 - Links descritivos: Link e navegação de texto devem ser unicamente descritos como alvo ou função.
 - Links para formatos alternativos: Links para formatos alternativos devem indicar que uma página alternativa será aberta.
 - Combinando links repetidos: Links repetidos devem ser combinados em um único link.
- Notificações:
 - Notificações inclusivas: Notificações devem ser visíveis e audíveis.
 - Notificações do sistema operacional: Notificações padrão do sistema operacional devem ser usadas quando disponíveis e apropriadas.
 - Mensagens de erro e correção: Mensagens de erro devem estar claras.
 - Feedback e assistência: Feedback ou assistência devem ser providas quando apropriado.
- Scripts e conteúdo dinâmico:
 - Funcionamento progressivo: Apps e web sites devem trabalhar de maneira progressiva para assegurar uma experiência funcional ao usuário.
 - Controle de mídia: Atualização das mídias ou conteúdo animado devem possuir botão de pausa, stop e esconder controle.
 - Atualização de página: Atualização das páginas não devem ser feitas sem aviso.
 - Tempos de espera: O tempo de resposta deve ser ajustável.
 - Controle de entrada: Interações de entrada devem ser adaptáveis.
- Estrutura:
 - Título único para páginas: Todas as páginas ou telas devem conter um único e identificável título.
 - Cabeçalhos: O conteúdo deve estar colocado de maneira lógica e hierárquica no cabeçalho suportado pela plataforma.
 - Containers e marcadores: Containers devem ser usados para descrever a estrutura de páginas ou telas.

- Grupos de elementos: Controles, objetos e grupo de interface devem representar um único componente acessível.
- Textos equivalentes:
 - Alternativas para elementos não textuais: Devem ser brevemente descritas a intenção ou propósito da imagem, objeto ou elemento.
 - Conteúdo decorativo: Imagens decorativas devem ser escondidas de tecnologias assistivas.
 - Dicas e informações complementares: Dicas não devem ser repetidas em textos de links ou outras alternativas.
 - Tarefas, marcas e propriedades: Elementos devem conter propriedades de acessibilidade apropriados.
 - Formatação visual: Formatação visual não deve ser usada para indicar significado sozinho.

Ainda que as diretrizes de acessibilidade propostas pela BBC sejam compreensíveis, elas não cobrem todas as diretrizes propostas pela W3C. As diretrizes que não são cobertas pela BBC são as seguintes:

- Montante de informações: diminuir o montante de informações colocadas na tela comparando com as versões para desktop.
- Posição dos títulos: a posição dos títulos dos formulários deve ser colocado em cima ao invés de ao lado.
- Teclado: todas as funcionalidades devem ser operáveis sem a interface do teclado.
- Gestos: os gestos devem ser tão fáceis quanto possível.
- Posição dos elementos interativos: os elementos interativos devem estar posicionados onde possam ser facilmente achados.
- Orientação da tela: aplicações móveis devem suportar as duas orientações de tela. Mudanças na orientação da tela devem ser facilmente percebidas pelas tecnologias assistivas.
- Posicionamento dos elementos: Informações importantes devem estar visíveis sem precisar descer a tela.
- Instruções: títulos ou instruções são necessárias quando o usuário necessita inserir informações.

- Ajuda: Ajudas devem estar disponíveis.
- Fácil entrada de dados: troque a quantidade de texto necessária por selecionar menus, botões radio, caixas de seleção ou por informação de entrada automática.

Tanto a BBC quanto a W3C propõem um grande número de padrões de acessibilidade mas os padrões apresentados não são completos o que significa que podem ser usados padrões de acessibilidade definidos por outros grupos ou companhias.

A avaliação da usabilidade de um sistema computacional passa pela verificação de sua acessibilidade relacionada ao contexto de uso, às tarefas que apoia, bem como às necessidades e às preferências dos diferentes usuários prospectivos (MELO; BARANAUSKAS; BONILHA, 2004).

2.3 Avaliação de acessibilidade

No contexto da Web, métodos para avaliação de acessibilidade têm sido propostos e a complementaridade desses métodos possibilita a verificação da acessibilidade Web sob diferentes perspectivas. Alguns desses métodos são mencionados a seguir (MELO; BARANAUSKAS; BONILHA, 2004):

- Uso de navegadores gráficos e textuais: permite investigar questões relacionadas à interpretação de páginas Web por meio de diversas configurações de acesso.
- Validação automática da linguagem de marcação: identifica problemas relacionados à sintaxe HTML e CSS, por exemplo; atividade difícil de ser realizada depois que as páginas já estão codificadas.
- Verificação de acessibilidade por ferramentas semi automáticas: evidencia erros no design de páginas HTML que prejudicam sua acessibilidade, além de lembrarem verificações importantes que devem ser realizadas manualmente
- Avaliação com usuários com diferentes habilidades e/ou deficiências: possibilita a observação das estratégias de interação construídas pelos diferentes usuários na realização de tarefas típicas, em contextos diversificados e com o uso de tecnologias assistivas, por exemplo leitores de tela, ampliadores de tela, bem como a identificação das dificuldades que enfrentam

Com base nas recomendações internacionais do W3C/WAI, foram desenvolvidos softwares, descritos no Capítulo 3 que avaliam o nível de acessibilidade em páginas na Internet. Tais programas produzem relatórios precisos com os problemas encontrados e que deveriam ser corrigidos para que a página se torne acessível (TANGARIFE; aO, 2005).

A acessibilidade de um software só pode ser afirmada para necessidades específicas de cada usuário em seu contexto de uso. Se certo usuário usar a aplicação nesse determinado contexto, ele será acessível para ele. Não é razoável testar cada contexto de uso possível, então uma série de critérios práticos devem ser aceitos, baseados na experiência da comunidade usuária, como referência para determinar se o software é acessível ou não. Pensando nisso, não é viável testar de forma manual cada cenário possível, dessa forma ferramentas de teste automático de acessibilidade analisam determinados critérios que normalmente são os padrões de acessibilidade definidos (MARTÍN; YELMO, 2014).

Dispositivos móveis apresentam novas oportunidades e desafios para tecnologia em sociedade, como estar sempre conectado, portabilidade, ser ainda mais pessoal que um computador pessoal, a democratização do acesso a informação, a oportunidade de interação e a redução de complexidade. Por outro lado, incluiu algumas limitações ao uso dos mesmos como tamanho da tela, limitação de entrada de dados, conectividade limitada, recursos computacionais limitados além de limitação da duração das baterias. Com as oportunidades e limitações surgem também novos jeitos de evoluir apropriadamente as aplicações para dispositivos móveis para os usuários levando em consideração a usabilidade e acessibilidade (BILLI et al., 2010a).

Testes automatizados ou semi automatizados de acessibilidade envolve verificar a conformidade da aplicação móvel com as diretrizes de acessibilidade conhecidas cobertas por determinada ferramenta. Existem algumas ferramentas disponíveis que mostram diferentes resultados e cobrem também diferentes diretrizes. Ainda assim as ferramentas não conseguem detectar todos os problemas de acessibilidade das ferramentas devido a muitas dessas diretrizes necessitarem de intervenção humana para detecção (BRAJNIK; YESILADA; HARPER, 2011).

2.4 Trabalhos relacionados

Uma iniciativa de criar testes unificados para padrões de acessibilidade para web traz as diretrizes de acessibilidade listadas pela WCAG² (Web Content Accessibility Guidelines) de forma a listar as diretrizes e para cada diretriz indicar o que deve ser testado para que aquela diretriz seja satisfeita com uma descrição sucinta do que diz respeito àquela diretriz e seu teste separado por critérios de aceitação podendo esse documento ser usado tanto para testes de um software específico em seu desenvolvimento quanto para auxiliar desenvolvedores de ferramentas de testes automatizados para web (VELLEMAN et al., 2010).

O guia traz os testes HTML, CSS e testes para objetos externos específicos para cada diretriz, onde cada teste tem título, critério de aplicabilidade que são elementos, atributos ou combinações usadas para determinar a aplicabilidade do teste, o procedimento do teste que é uma descrição dos múltiplos passos para a execução do teste, os resultados esperados definindo condições de falha e de sucesso do teste e por último, se o teste pode ou não ser totalmente automatizado (VELLEMAN et al., 2010).

Com a intenção de mensurar a cobertura, completude e corretude de ferramentas de análise de acessibilidade web, Markel (VIGO; BROWN; CONWAY, 2013) verifica se as ferramentas conseguem de fato avaliar aplicações web baseado no estado atual das aplicações. A avaliação das ferramentas de teste de aplicações web é feita de 3 formas: pelo número de diferentes critérios de sucesso com falhas apontados, completude em relação as violações reportadas sobre o real número de violações da aplicação, e quão bem as ferramentas apontam as violações encontradas.

A fim de verificar como os usuários reportam os problemas de acessibilidade dos aplicativos móveis para Android, Eler (ELER; ORLANDIN; OLIVEIRA, 2019) investigou mais de 700 aplicativos na Google Play Store e verificou que destes apenas 40% dos aplicativos continham revisões que continham algum conteúdo de acessibilidade, notou ainda que desses aplicativos, àqueles que eram notados problemas para operar, perceber e entender os elementos no aplicativo possuíam score mais baixo. O estudo entendeu que acessibilidade é um dos passos para melhorar a qualidade de aplicações móveis.

² <https://www.w3.org/WAI/intro/wcag>

3 Ferramentas que analisam diretrizes de acessibilidade

Uma revisão sistemática foi realizada para obter um levantamento dos trabalhos e ferramentas que avaliam automaticamente a acessibilidade digital de aplicações móveis. Os detalhes desta revisão são apresentados a seguir.

3.1 Protocolo

O primeiro passo da revisão foi definir os objetivos da pesquisa que foram:

- Identificar as diretrizes de acessibilidade para dispositivos móveis por meio de um levantamento bibliográfico sobre as diretrizes e recomendações de acessibilidade disponíveis para dispositivos móveis.
- Identificar as ferramentas de análise e teste de software automatizados que verificam diretrizes de acessibilidade.
- Mapear quais diretrizes de acessibilidade para dispositivos móveis estão sendo analisadas pelas ferramentas encontradas

Com os objetivos estabelecidos foi possível levantar as questões de pesquisa a serem respondidas pela revisão:

- Q1 - Quais as diretrizes de acessibilidade para dispositivos móveis foram definidas pelos órgãos que definem padrões para a internet?
- Q2 - Quais as ferramentas de teste/análise de aplicativos para dispositivos móveis verificam diretrizes de acessibilidade?
- Q3 - Quais as diretrizes de acessibilidade são verificadas pelas ferramentas de teste/análise de aplicativos para dispositivos móveis?

Para encontrar o maior número possível de artigos abrangendo os tópicos da pesquisa, foi criada a Tabela 1 com os termos relacionados à pesquisa e que deveriam ser utilizados na *string* de busca que foi posteriormente utilizada na base da Scopus. A partir desses termos foi definida a *string* de busca genérica:

- TITLE-ABS-KEY (accessibility AND (mobile* OR smartphone* OR "smart phone*" OR "cell phone*" OR "cellular phone*")) AND (test* OR analys* OR evaluat* OR assess* OR validat*) AND (tool* OR framework*)

Tabela 1 – Termos Utilizados na pesquisa

Palavras chave	Termos relacionados em inglês
Acessibilidade	"accessibility"
Dispositivos móveis	"mobile", "smartphone", "smart phone", "cellphone", "cellular phone"
Teste	"test", "analys", "evaluat", "assess", "validat"
Ferramentas	"tool", "framework"

A *string* foi definida para encontrar os termos no título, *abstract* e palavras chave dos artigos. Após a definição da *string* foram definidos critérios de inclusão(CI) e critérios de exclusão(CE) dos artigos na base de dados:

- CI- Estar disponível de forma integral online;
- CI- Artigos na área de computação, engenharia, matemática ou *business*;
- CI- Artigos publicados a partir de 2008;
- CE- Artigos relacionados à acessibilidade para web em geral;
- CE- Estudos secundários ou terciários;
- CE- Estudos não publicados em eventos ou periódicos;
- CE- Artigos não escritos em língua inglesa;

Com a aplicação dos critérios de inclusão e exclusão foi encontrada a string de busca específica para este trabalho aplicada a base de dados *Scopus*:

- TITLE-ABS-KEY (accessibility AND (mobile* OR smartphone* OR "smart phone*" OR "cell phone*" OR "cellular phone*") AND (test* OR analys* OR evaluat* OR assess* OR validat*) AND (tool* OR framework*)) AND (LIMIT-TO (LANGUAGE,"English")) AND (EXCLUDE (DOCTYPE,"cr") OR EXCLUDE (DOCTYPE,"bk") OR EXCLUDE (DOCTYPE,"ed") OR EXCLUDE (DOCTYPE,"le")) AND (LIMIT-TO (SUBJAREA,"COMP") OR LIMIT-TO (SUBJAREA,"ENGI") OR LIMIT-TO (SUBJAREA,"MATH") OR LIMIT-TO (SUBJAREA,"DECI") OR LIMIT-TO (SUBJAREA,"BUSI"))

3.2 Artigos encontrados

Como resultado dessa busca, foram encontrados 285 artigos. Após a primeira análise dos títulos, *abstract* e palavras-chave dos artigos de maneira criteriosa para definir daquele grupo quais seriam lidos de forma integral, foram escolhidos 31 artigos para leitura. Para

a definição sumária se os artigos entrariam no estudo eles deveriam responder ao menos uma questão de pesquisa. Dos artigos analisados, apenas 17 deles respondiam a alguma questão de pesquisa, conforme mostra o gráfico da Figura 1. O Gráfico da Figura 2 mostra como a publicação desses artigos se distribui ao longo dos últimos 10 anos. Os artigos definidos para a revisão sistemática estão listados na Tabela 2.

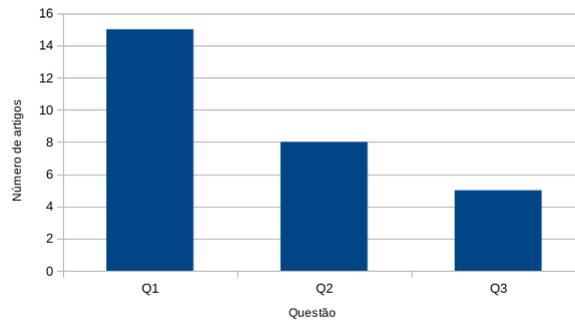


Figura 1 – Quantidade de artigos que responde cada questão de pesquisa.

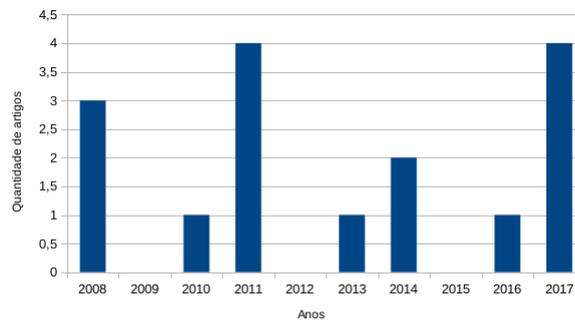


Figura 2 – Distribuição dos artigos conforme os anos de publicação.

3.3 Ampliação das bases de pesquisa

Como muitas ferramentas são desenvolvidas e não são objetos de publicação científica, especialmente quando são produzidas pelas próprias plataformas de desenvolvimento, para a complementação do estudo, as pesquisas por ferramentas de avaliação automática de acessibilidade foram também realizadas em mecanismos de busca, como o Google, por exemplo; em lojas de aplicações móveis; e nos sites oficiais das próprias plataformas iOS, Android e Microsoft.

Tabela 2 – Artigos utilizados na revisão sistemática

Ano	Título com referência
2008	Automatic evaluation of mobile web accessibility (ARRUE; VIGO; ABASCAL, 2007)
2008	A web compliance engineering framework to support the development of accessible rich internet applications (VELASCO et al., 2008)
2008	Evaluating web accessibility for specific mobile devices (VIGO et al., 2008)
2010	A unified methodology for the evaluation of accessibility and usability of mobile applications (BILLI et al., 2010b)
2011	Crosschecking the mobile Web for people with visual impairments (CARRIÇO; LOPES; BANDEIRA, 2011)
2011	A unified environment for accessing a suite of accessibility evaluation facilities (PARTARAKIS et al., 2011)
2011	Results from multi-dimensional accessibility assessment (BANDEIRA; LOPES; CARRIÇO, 2011)
2011	Barriers common to mobile and disabled web users (YESILADA; BRAJNIK; HARPER, 2011)
2013	Accessibility of Mobile Web Apps by Screen Readers of Touch-Based Mobile Phones (AL-SUBAIHIN; AL-KHALIFA; AL-KHALIFA, 2013)
2014	Appstrument - A unified app instrumentation and automated playback framework for testing mobile applications (NANDAKUMAR; EKAMBARAM; SHARMA, 2014)
2014	A heuristic checklist for an accessible smartphone interface design (MI et al., 2014)
2015	Getting smartphones to talkback: Understanding the smartphone adoption process of blind users (RODRIGUES et al., 2015)
2016	Validation of Toolkits for developing third-generation Android accessible mobile applications (GEMOU et al., 2016)
2017	Mobile web accessibility readiness of government websites using diagnostic tools: An exploratory study (KAUR; DANI, 2017)
2017	Enhanced UI Automator Viewer with improved Android accessibility evaluation features (PATIL; BHOLE; SHETE, 2017)
2017	Accessibility analysis of e-governance oriented mobile applications (BALAJI; KUPPUSAMY, 2017)
2017	A checklist for assessing blind users' usability of educational smartphone applications (LEE; LEE, 2017)

3.4 Ferramentas de avaliação automática de acessibilidade digital

A análise dos artigos e das ferramentas encontradas nas lojas oficiais e nos sites oficiais das plataformas permitiu identificar diversas ferramentas que avaliam automaticamente a acessibilidade em dispositivos móveis. Uma breve descrição de cada uma delas agrupadas por plataforma é apresentada a seguir.

3.4.1 Android

As ferramentas para avaliação automática de acessibilidade de aplicações Android são apresentadas a seguir.

- *Accessibility Scanner*¹: é um aplicativo móvel Android que permite que, para cada tela de um aplicativo em uso, seja possível obter recomendações de melhoria de acessibilidade. As propriedades verificadas são rótulos de componentes não textuais, contraste entre elementos da interface com o plano de fundo, tamanho dos itens clicáveis, duplicação de descrição de elementos não textuais, sobreposição de elementos clicáveis, e ações em palavras específicas de um texto (*clickable span*).
- *Android Lint* (GOOGLE, 2018a) verifica todos os arquivos de um projeto Android, incluindo o código fonte e reporta problemas estruturais. No contexto de acessibilidade ela analisa se componentes não textuais possuem descrição além de verificar se os métodos dão suporte a tecnologias assistivas.
- *Espresso* (GOOGLE, 2018c) é um framework de teste que simula as interações do usuário na aplicação móvel detectando alguns tipos de diretrizes de acessibilidade como descrição de elementos não textuais, contraste de cores, tamanho de elementos clicáveis, textos alternativos duplicados e os elementos clicáveis.
- *Robolectric* (GOOGLE, 2018d) é um framework de teste que permite ao usuário executar os testes com a máquina virtual do Java como emulador de um dispositivo real e assim como o Espresso faz a análise de acessibilidade durante a execução checando as mesmas diretrizes de acessibilidade.
- *PUMA* (HAO et al., 2014) é um framework de análise dinâmica onde se pode customizar o que vai ser checado. Mas pode ser avaliado tamanho de itens clicáveis, espaçamento entre os itens e contraste de cores.
- *forApp* (FORAPP, 2018) é um serviço online para checar aplicações móveis onde se faz o upload do arquivo a ser analisado e é gerado documento PDF com as violações às diretrizes de descrição de elementos não textuais além de identificação de elementos. É um serviço pago.
- *MATE* (ELER et al., 2018) é uma ferramenta de teste que explora automaticamente o comportamento da aplicação móvel com entradas aleatórias checando as diretrizes de

¹ <https://play.google.com/store/>

acessibilidade a cada nova tela encontrada. A ferramenta testa as mesmas diretrizes que o Espresso e Robolectric.

- *IBM AbilityLab Mobile Accessibility Checker* (IBM, 2018) provê uma otimização para o usuário de aplicações móveis resolvendo problemas de usabilidade e acessibilidade para aplicações Android e IOS nativos checando algumas diretrizes de acessibilidade.

3.4.2 iOS

As ferramentas para avaliação automática de acessibilidade de aplicações iOS são apresentadas a seguir.

- *Accessibility Inspector* é uma ferramenta nativa que provê informações sobre a acessibilidade dos elementos da interface em um aplicativo. Os desenvolvedores podem identificar as falhas manualmente com elementos não textuais sem descrição alternativa.
- *Mobile Web Accessibility Checker* (MWAC, 2018) é uma aplicação móvel para evoluir a acessibilidade de outra aplicação móvel, mostrando problema com o contraste de cores e se os elementos são executáveis por um leitor de tela.
- *EarlGrey* (GOOGLE, 2018b) é um framework de teste automatizado nativo do iOS que analisa diretrizes de acessibilidade. Ele funciona como o *Espresso* para dispositivos iOS identificando as mesmas diretrizes.
- *KIF* (KIF, 2018) é um framework de teste integrado onde os desenvolvedores podem escrever os testes para checar a acessibilidade e simular as entradas do usuário como cliques e pode ser comparado ao Robolectric além de analisar as mesmas diretrizes.

3.4.3 Windows Phone

As ferramentas para avaliação automática de acessibilidade de aplicações para Windows Phone são apresentadas a seguir.

O *Windows Software Development Kit (SDK)* inclui 3 ferramentas de teste para identificar problemas de acessibilidade devendo ser utilizadas diretamente no *Visual Studio*.

- *AccScope*(MICROSOFT, 2018b) permite aos desenvolvedores testar algumas diretrizes de acessibilidade durante o desenvolvimento da aplicação. Sua principal função é verificar em cada tela se o leitor de tela está funcionando corretamente na aplicação.
- *Inspect*(MICROSOFT, 2018c) é uma ferramenta que permite aos desenvolvedores selecionar qualquer elemento da tela e a ferramenta provê as informações de acessibilidade. A ferramenta também checa a estrutura de navegação da aplicação testando assim seu *layout* além de textos alternativos para elementos não textuais e botões.
- *AccChecker*(MICROSOFT, 2018a) é geralmente utilizado para testar aplicações que já estão finalizadas verificando se o leitor de tela consegue identificar os elementos, além de ser possível selecionar um elemento da tela para ser checado se existe descrição e *layout* consistente.

3.5 Categorização das ferramentas de análise de acessibilidade

Como qualquer ferramenta de análise, avaliação automática de acessibilidade de aplicações móveis pode ser feita de diferentes maneiras, as ferramentas de análise encontradas nesse estudo são de duas categorias:

Análise estática é uma eficiente maneira de analisar código fonte e arquivos de design para encontrar problemas de acessibilidade. Contudo, este tipo de análise falha em detectar problemas que apenas seriam encontrados durante a execução da aplicação, por exemplo tamanho de itens clicáveis e contraste que depende de configurações específicas de ambiente.

Análise dinâmica é utilizada para encontrar problemas de acessibilidade enquanto a aplicação está sendo executada. este tipo de análise não tem a mesma limitação das ferramentas de análise estática e pode ser classificada ainda de três formas:

- *Exploração manual*. Os desenvolvedores devem explorar a aplicação manualmente para encontrar problemas de acessibilidade nas telas da aplicação. Algumas ferramentas necessitam ainda que o desenvolvedor indique qual tela deve ser explorada pela ferramenta de análise enquanto outras ferramentas fazem a análise de todas as telas. A limitação desse tipo de ferramenta é que o teste pode ser tedioso uma vez que precisa ser indicada tela a tela para análise, em grande escala isso pode se tornar complexo de ser gerenciado.

- *Scripts de teste.* É uma alternativa que mais se aproxima a automação de testes unitários. Alguns frameworks de teste permitem que os desenvolvedores escrevam os scripts de teste para testar as ferramentas como a interação de usuários o que possibilita a análise de acessibilidade das telas. Dessa forma, o conjunto de testes analisa em tempo de execução cada tela. A limitação desse tipo de ferramenta é a eficiência uma vez que é necessária a escrita dos casos de teste criando as interações do usuário. (ELER et al., 2018).
- *Teste GUI automatizado.* Consiste em interações com a aplicação de maneira automática, por exemplo com entradas de dados de maneira randômica. Durante a exploração automática, a ferramenta consegue avaliar os problemas de acessibilidade em tempo de execução. A limitação desse tipo de ferramenta é que existe a dependência da capacidade de exploração do algoritmo em cobrir diferentes operações e telas da aplicação. Essa particularidade é ainda mais difícil com dados sensíveis como por exemplo, login, dados validados, que retém o acesso a algumas telas da aplicação.

Baseado nessas categorias foi possível agrupar as ferramentas da seguinte maneira:

- Análise estática: Lint;
- Análise dinâmica - exploração manual: Accessibility Scanner, Mobile Web Accessibility Checker, AccScope, IBM AbilityLab Mobile Accessibility Checker, Accessibility Inspector, AccChecker;
- Análise dinâmica - scripts de teste: Espresso, Robolectric, EarlGrey, KIF;
- Análise dinâmica - teste GUI automatizado: MATE, PUMA, forApp.

3.6 Diretrizes de acessibilidade cobertas pelas ferramentas

Durante a investigação das ferramentas foi possível notar que cada ferramenta usa uma terminologia diferente para as diretrizes de acessibilidade checadas. Por exemplo, alternativas para elementos não textuais foi chamado de título, descrição do conteúdo ou texto explicativo. Também existem casos em que a diretriz de acessibilidade é interpretada de diferentes maneiras. E ainda mais de um tipo de teste da acessibilidade são endereçados a uma mesma diretriz. A tabela 3 mostra de maneira geral a investigação na qual buscamos levantar quais diretrizes de acessibilidade são testadas pelas ferramentas.

No total, apenas 8 das 64 diretrizes de acessibilidade apresentadas no Capítulo 2 estão cobertas pelas ferramentas pesquisadas descritas neste Capítulo, isso corresponde a 8% das diretrizes recomendadas. levando em consideração a quantidade de diretrizes de acessibilidade proposta pelos órgãos especializados, o número de diretrizes cobertas é frustrante. Esse resultado mostra o quão distante as ferramentas de testes estão de automatizar os testes de diretrizes de acessibilidade. Por exemplo, não existe suporte automático para testar áudio e vídeo, imagens, links, notificações e estrutura.

Analisando as diretrizes de acessibilidade cobertas pelas ferramentas é possível concluir que verificar elementos não textuais está presente em quase todas as ferramentas. Talvez isso seja porque a avaliação dessa diretriz de acessibilidade pode ser concluída de duas maneiras: a presença de alternativas para elementos não textuais é suficiente para definir se a aplicação é acessível; e a ausência de alternativas para elementos não textuais não é suficiente para definir que não é acessível. Contraste de cores, tamanho dos itens clicáveis e espaçamento dos elementos também são diretrizes cobertas comuns entre as ferramentas. Novamente, isso levanta um critério específico sobre como cobrir as diretrizes de acessibilidade. Por outro lado tem as propriedades que raramente são testadas pelas ferramentas, por exemplo armadilhas de teclado e foco visível.

Tabela 3 – Diretrizes de acessibilidade cobertas pelas ferramentas.

Ferramentas	Diretrizes										
	Contraste de cores	Tamanho de itens clicáveis	Espaçamento	Elementos acionáveis	Foco visível	Armadilhas de Teclado	Consistência	Alternativas para elementos não textuais			
Android Lint				X							X
Espresso	X	X	X	X							X
Robolectric	X	X	X	X							X
PUMA		X	X								X
forApp					X						X
Accessibility Scanner	X	X	X								X
MATE	X	X	X	X							X
IBM AbilityLab	X										X
Accessibility Inspector									X		X
Mobile Web Accessibility Checker	X				X	X					X
EarlGrey	X	X	X	X							X
KIF	X	X	X	X							X
AccScope											X
Inspect										X	X
AccChecker				X						X	X

4 Guia de automação de testes de acessibilidade para deficiências visuais

4.1 Introdução

O teste de acessibilidade em dispositivos móveis é geralmente realizado de forma manual principalmente porque as ferramentas de automação disponíveis são capazes de detectar um pequeno número de violações com base em recomendações como o WCAG, o guia BBC ou o e-MAG, por exemplo. Uma das limitações das ferramentas de automação é que avaliar se uma recomendação foi seguida ou não depende muitas vezes de julgamento humano e não de critérios objetivos e que podem ser verificáveis automaticamente. Em muitos casos, é possível definir critérios objetivos para identificar situações em que provavelmente houve uma violação de acessibilidade, mas não é possível afirmar com certeza se houve tal violação.

Os padrões e guias de acessibilidade em geral apresentam as recomendações e os critérios de aceitação em uma perspectiva conceitual e também do ponto de vista da avaliação manual das aplicações. Utilizar os guias atuais como requisitos para a implementação de ferramentas automatizadas de avaliação não é uma tarefa trivial.

Neste capítulo são apresentados os resultados de um estudo realizado sobre a possibilidade de automação (total ou parcial) da avaliação de recomendações para deficiência visual do guia da BBC. A fonte primária deste estudo é o próprio guia da BBC mas também foram consultadas outras fontes como o e-MAG e o WCAG.

Os resultados deste estudo são apresentados como um guia para apoiar a implementação ou a evolução de ferramentas de teste automático de acessibilidade. Como a automação depende de uma plataforma específica, este guia foi escrito de ponto de vista do teste de acessibilidade de aplicações Android e de ferramentas que utilizam as APIs fornecidas pela plataforma¹ para coletar informações de aplicações que estão sob teste. Portanto, ainda que conceitualmente seja possível verificar automaticamente certos critérios de aceitação, por vezes o limitante são as informações possíveis de serem obtidas da plataforma em si.

¹ <https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo>. Em especial a API: <https://developer.android.com/reference>

Como o guia da BBC contém muitas recomendações, o escopo deste estudo foi limitado para analisar as recomendações das seguintes seções: Design, Formulários, Links, Notificações e Textos equivalentes. Cada tópico do guia aqui apresentado está dividido em:

- Descrição da recomendação: uma visão geral sobre a recomendação em questão e sua importância para assegurar a acessibilidade das aplicações móveis.
- Violação: definição das condições em que ocorre uma violação da recomendação.
- Exemplo: um exemplo de violação com uma ilustração de um aplicativo quando a violação puder ser facilmente percebida de forma visual.
- Avaliação manual: descrição detalhada sobre como um aplicativo móvel pode ser avaliado manualmente para se verificar se a recomendação em questão está sendo atendida ou não.
- Possibilidade de avaliação automática: definida como totalmente automatizável, parcialmente automatizável, não automatizável ou ainda não aplicável quando a diretriz não é aplicável para a plataforma Android pela BBC.
- Justificativa: uma justificativa para os casos em que a recomendação não pode ser totalmente automatizada.
- Avaliação automática: descrição detalhada sobre como implementar mecanismos para avaliar automaticamente se a recomendação em questão está sendo implementada ou não.
- Avaliação da possibilidade de automatização: validação feita por meio da implementação da avaliação automática ou parcialmente automática na ferramenta MATE e a avaliação manual em contraponto com a avaliação automática.

Um sumário das recomendações de acessibilidade avaliadas e o rótulo associado a cada uma delas sobre a possibilidade de avaliação automática está apresentado na Tabela 4.1.

A validação do guia aqui apresentado no que se refere à possibilidade de avaliação automática de forma total ou parcial das recomendações de acessibilidade foi feita por meio da implementação da avaliação automática na ferramenta MATE (ELER et al., 2018). Os códigos gerados pela implementação podem ser consultados em <https://github.com/mate-android-testing/mate>.

A possibilidade de implementar a avaliação automática na ferramenta dá indícios de que as informações obtidas da plataforma apoiam a avaliação automática ou parcialmente

Tabela 4 – Lista de tópicos, recomendações e rótulos atribuídos: automatizável (SIM), não-automatizável (NÃO) ou parcialmente automatizável (PARC)

Tópico	Diretriz	Automatizável?
Design	Contraste de cores	SIM
	Cor e Significado	PARC
	Estilo e Leitura	NÃO
	Tamanho de itens clicáveis	SIM
	Espaçamento	SIM
	Conteúdo dimensionável	NÃO
	Elementos acionáveis	PARC
	Foco visível	SIM
	Consistência	NÃO
	Escolha	NÃO
	Ajustabilidade	NÃO
Flicker	SIM	
Formulários	Rótulos dos controles dos formulários	SIM
	Entrada de formulários	SIM
	Layout do formulário	SIM
	Agrupamento de elementos	NÃO
	Foco manuseável	SIM
Links	Links descritivos	PARC
	Links para formatos alternativos	PARC
	Combinando links repetidos	NÃO
	Notificações inclusivas	NÃO
Notificações	Notificações do SO	NÃO
	Mensagens de erro e correção	PARC
	Feedback e assistência	NÃO
Textos equivalentes	Alternativas para elementos não-textuais	PARC
	Conteúdo decorativo	PARC
	Dicas e informações complementares	PARC
	Papéis, marcas e propriedades	PARC
	Formatação visual	NÃO

automática das recomendações. Não foi possível no período deste estudo, entretanto, contrastar os resultados da avaliação automática com a avaliação manual feita por especialistas. Em alguns casos, os próprios autores deste trabalho encontraram manualmente violações de acessibilidade relacionados às recomendações e utilizaram a ferramenta para encontrar o mesmo problema e assim validar a implementação. Para as avaliações manuais, foram utilizados os passos encontrados em cada item do guia.

Em todo caso, avaliadores diferem em critérios utilizados para definir se certa recomendação foi seguida ou não, e por isso uma validação robusta exigiria o envolvimento de diversos profissionais. Entende-se, portanto, que a implementação na ferramenta indica que, com base nas informações coletadas do aplicativo, é possível aplicar diferentes critérios de aceitação de acordo com o contexto e o avaliador. No guia aqui apresentado, os critérios

de aceitação foram definidos livremente pelos autores deste trabalho com base em suas interpretações do guia da BBC, apoiados por consultas feitas a outros guias como e-MAG e WCAG.

4.2 Design

4.2.1 Contraste de cores

O contraste de cores do texto deve ser suficiente para diferenciar os elementos de uma página ou tela da aplicação. Uma relação de contraste adequada entre o texto e o plano de fundo é fundamental para que todos possam visualizar as informações de forma clara e sem grandes esforços. Além disso, um bom contraste é essencial para pessoas com baixa visão, com daltonismo e usuários que utilizam telas monocromáticas (EMAG, Acesso em: 23/04/2019).

O contraste de cores ocorre quando uma superfície de uma cor induz a cor imediatamente oposta de uma superfície. Normalmente, a superfície de indução é grande, que no caso da computação pode ser considerada a cor de fundo que envolve toda a tela ou pelo menos a área circundante do componente visual em questão (HURLBERT; WOLF, 2004). A Figura 3 mostra um exemplo de razões de contraste baixo, suficiente e alto.

Contraste Baixo	Contraste Suficiente	Alto Contraste
Verde no cinza	Verde no cinza	Verde no cinza
Amarelo no azul	Amarelo no azul	Amarelo no azul

Figura 3 – Graduação de contraste de cores.

Além de escolher cores que tenham uma boa relação de contraste, é preciso tomar cuidado com imagens utilizadas de plano de fundo. Um fundo decorado ou em forma de figura, como uma paisagem, por exemplo, torna o conteúdo da página de difícil visualização, além de desviar a atenção do usuário.

Violação: O mínimo contraste de cores para textos e imagens com o fundo deve ser de 4.5:1. Qualquer variação que não obedeça a esse critério está fora dos padrões aceitáveis.

Exemplo: A Figura 4 apresenta uma tela de uma aplicação móvel comercial que não apresenta contraste suficiente nos quadros de inserção de dados.

The image shows a mobile application interface for MaxMilhas. At the top, there is a status bar with various icons and a battery level of 19%. Below the status bar is the MaxMilhas logo. The main heading is "Compre passagens econômicas" with the subtext "Sem precisar ter milhas". The form consists of several input fields: "Insira a origem" (with a swap icon), "Insira o destino", "Data de ida", and "Volta (Opcional)". Below these is a summary box showing "1 pessoa, Classe econômica". A large green button labeled "Pesquisar passagem" is at the bottom. At the very bottom, there is a navigation bar with three icons: "Busca", "Ofertante", and "Conta". The text in the form is light gray, making it difficult to read against the white background.

Figura 4 – Exemplo de violação de contraste de cores.

Avaliação manual: A relação de contraste pode ser encontrada dividindo-se o valor da luminosidade relativa da cor mais clara de um dos planos pelo valor da luminosidade relativa da cor mais escura do outro plano. Levando-se em consideração a perda de percepção do contraste resultante da baixa acuidade visual, cromodeficiência ou perda de sensibilidade ao contraste devido ao envelhecimento, é recomendada aqui uma relação de contraste, de, no mínimo, 4,5:1 (EMAG, Acesso em: 23/04/2019).

O cálculo exato da razão de contraste não pode ser feito de forma manual, por isso a avaliação é geralmente apoiada por ferramentas que permitem selecionar o elemento visual a ser avaliado e em seguida o plano de fundo. Assim, a ferramenta é capaz de fazer um cálculo dos valores indicados.

Possibilidade de avaliação automática: Automatizável.

Avaliação automática: Para verificar o contraste de cores de forma automatizada pode-se seguir os passos:

1. Gere uma imagem da tela toda ou da região de interesse.
2. Utilize um algoritmo de segmentação da região de interesse (área do componente avaliado) para separar os elementos do plano principal do plano de fundo. Uma abordagem para é a segmentação baseada em pixels, em que uma característica

escalar de cor abaixo do limiar pertence ao plano de fundo, e os demais pertencem ao objeto da imagem.

3. Identifique os valores de luminância das cores de fundo e do objeto do primeiro plano.
4. Divida o valor encontrado da cor mais clara com a cor mais escura.
5. Verifique se a razão encontrada é maior ou igual a 4.5:1. Caso positivo, o contraste de cores está correto; em caso de falha, apresente a violação.

Avaliação da possibilidade de automatização: A diretriz contraste de cores já estava implementada na ferramenta MATE e diversas outras ferramentas antes do início desse trabalho (SILVA; ELER; FRASER, 2018). Portanto, confirma-se que a recomendação pode ser avaliada automaticamente.

4.2.2 Cor e significado

As informações não podem ser transmitidas unicamente por meio de características sensoriais, tais como cor, forma, tamanho, localização visual, orientação ou som (EMAG, Acesso em: 23/04/2019).

Cores podem ser difíceis de distinguir em um ambiente claro e não é percebida por pessoas com problemas de visão. Leitores de tela não distinguem cores e alguns usuários precisam de modificar as cores da tela.

Cores são comumente usadas para mostrar a seleção de um campos, um link que está disponível, uma mensagem de erro, gráficos e informações mais importantes.

Violação: A cor é usada como único significado sem nenhuma informação adicional não passada através do uso da cor.

Exemplo: A Figura 5 mostra a tela de uma aplicação que usa apenas cores para identificar áreas que estão mais congestionadas de trânsito.

Avaliação manual: Para verificar manualmente, abra a aplicação utilizando um leitor de tela, observe as informações que estão sendo passadas com o uso de cores e verifique se existe alguma informação adicional passada através do leitor de tela.

Possibilidade de avaliação automática: Parcialmente automatizável.

Avaliação automática: Não é possível verificar se uma informação está sendo passada apenas pela diferenciação de cores de maneira totalmente automatizada, mas é possível identificar se elementos semelhantes são apresentados com cores diferentes e se

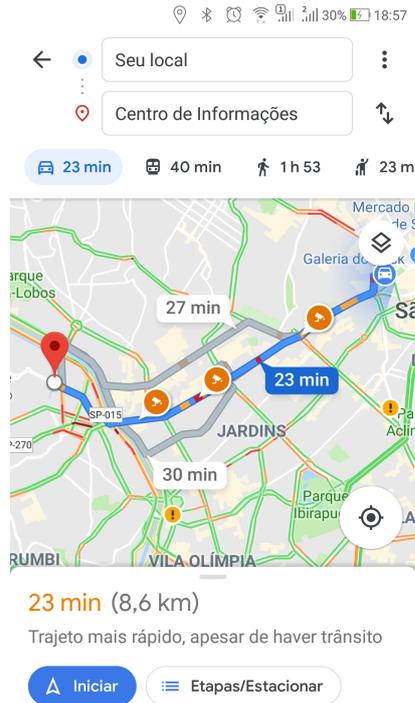


Figura 5 – Exemplo de violação de Cor e Significado.

há informação adicional informada para o elemento da tela que possui cores diferentes. Neste caso, portanto, é possível apenas sinalizar que a avaliação manual é necessária para garantir que informações estejam sendo transmitidas por mais de uma maneira e não só por meio do uso de diferentes cores.

Para verificar automaticamente o uso de informação adicional por meio de cores, os seguintes passos podem ser seguidos:

1. Identifique as cores dos componentes visuais como rótulos, botões, campos de texto, por exemplo.
2. Verifique se diferentes cores são usadas para os mesmos componentes visuais.
3. Verifique se informações adicionais estão presentes nos elementos com cores diferentes, como um *hint* ou uma descrição de conteúdo.
4. Se componentes visuais do mesmo tipo possuírem cores diferentes na tela e nenhuma informação adicional for encontrada, emita um alerta para que uma avaliação manual seja realizada.

Avaliação da possibilidade de automatização: Um algoritmo foi implementado na ferramenta MATE para detectar situações em que componentes do mesmo tipo apresentam cores diferentes. A validação é configurável e os usuários podem definir a

partir de quantas cores diferentes cada componente precisa apresentar para que um alerta seja emitido. Em comparação com avaliação manual, a ferramenta foi capaz de identificar situações em que rótulos, campos de entrada e botões possuíam cores diferentes. Nesses casos, é preciso intervenção manual para avaliar se realmente as cores tem alguma significado diferente ou foi somente uma questão de projeto de interface.

4.2.3 Estilo e leitura

O conteúdo principal deve ser acessado quando o estilo principal for removido ou substituído. Dispositivos móveis mais antigos não tem suporte a fontes, cores e estilos e dessa forma leitores de tela não conseguem acessá-los e algumas pessoas alteram suas configurações para melhor os atender. O uso de estilos, entretanto, não é possível em sistemas operacionais como Android e iOS.

Violação: Alterações no estilo fazem com que elementos da tela não sejam acessíveis.

Avaliação manual: Não é possível fazer a troca de estilos no Android.

Possibilidade de avaliação automática: Não aplicável para Android.

Justificativa: Não aplicável para Android.

4.2.4 Tamanho de itens clicáveis

Todos os elementos clicáveis devem ter o tamanho mínimo de 48dp x 48dp. A unidade *dp* é relativa à resolução da tela e equivale a 1 pixel em uma resolução de 160dpi. Em uma tela de resolução de 320dpi, são necessários 2 pixels para representar 1dp. Para calcular o tamanho em dp, é preciso dividir a quantidade de pixels pela resolução da tela. Esse tamanho é o para que o usuário consiga utilizar o elemento de maneira confortável, o que é equivalente a 7 - 10mm.

Esse tamanho é equivalente ao menor dedo médio. Um item interativo precisa ter no mínimo 7 x 7mm, se não for possível, e o menor tamanho aceitável é de 5 x 5 mm dentro da área clicável, embora o recomendável é que se tenha a maior área clicável possível.

Violação: Área clicável em um componente interativo inferior a 5 x 5mm.

Exemplo: A figura 6 apresenta a tela de uma aplicação amplamente usada onde um ícone que acessa um menu e diversos outros elementos interativos tem a área clicável menor do que a recomendada.



Figura 6 – Exemplo de violação de tamanho de item clicável.

Avaliação manual: Localize os itens clicáveis da tela, meça os itens e verifique se tem o tamanho mínimo recomendado. A medição manual do tamanho dos elementos interativos exige um grande esforço, e por isso muitos avaliadores não o fazem. Em geral, para fazer a medição, utiliza-se ferramentas de apoio como o Accessibility Scanner que apresenta a violação quando o item é menor que o recomendado, como mostrado na Figura 7.

Possibilidade de avaliação automática: Automatizável.

Avaliação automática: Para verificar o tamanho dos itens clicáveis na tela:

1. Identifique o item clicável.
2. Calcule a largura e a altura do item por meio de suas coordenadas na tela.
3. Calcule o fator de resolução da tela dividindo-se a resolução da tela por 160.
4. Calcule o tamanho relativo em dp da altura e da largura dividindo-os pelo fator de resolução da tela.
5. Verifique se a largura e a altura do item possuem tamanho mínimo de 48dp.

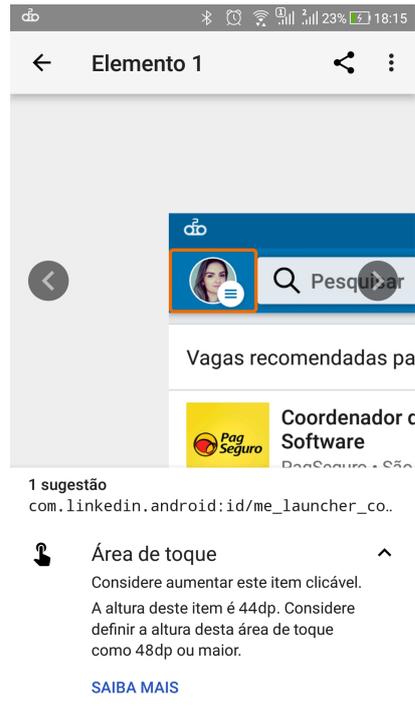


Figura 7 – Tamanho de item clicável analisado pelo Accessibility Scanner.

6. Se a largura ou a altura do item não possui o tamanho mínimo recomendado, emita um alerta de violação se o tamanho mínimo não for atingido.

Avaliação da possibilidade de automatização: Bem como contraste de cores, o tamanho de itens clicáveis já havia sido implementada na ferramenta MATE e outras ferramentas antes do início desse trabalho (SILVA; ELER; FRASER, 2018). Portanto, confirma-se que a recomendação pode ser avaliada automaticamente.

4.2.5 Espaçamento

Os elementos acionáveis de uma tela podem ser difíceis de serem acessados quando muito próximos, portanto a utilização de um espaçamento mínimo ajuda o usuário a não clicar onde não gostaria em uma tela. O espaçamento mínimo entre dois elementos acionáveis de uma tela deve ser de 1 pixel.

Violação: Espaçamento entre itens acionáveis menor que 1 pixel.

Exemplo: Na figura 8, existe sobreposição dos campos de entrada de origem e destino com o botão de inversão de origem e destino.



Figura 8 – Exemplo de violação de espaçamento.

Avaliação manual: Localize os itens clicáveis da tela e meça se entre cada item há o espaçamento de no mínimo 1 pixel. É difícil avaliar manualmente se existe uma distância de 1 pixel entre os elementos.

Possibilidade de avaliação automática: Automatizável.

Avaliação automática: Para verificar automaticamente o espaçamento entre elementos acionáveis:

1. Identifique os itens clicáveis da tela.
2. Identifique as coordenadas na tela do item de interesse.
3. Para cada limite (inferior, superior, esquerdo e direito) do item de interesse, verifique se existe algum outro item cujas coordenadas terminem a menos de 1 pixel de distância.
4. Se não houver a distância mínima de 1 pixel, emita um alerta de violação de recomendação.

Avaliação da possibilidade de automatização: O espaçamento entre itens também está entre as diretrizes que já haviam sido implementadas em outras ferramentas antes da escrita do guia (SILVA; ELER; FRASER, 2018). Além disso, a validação

automática desta recomendação foi implementada na ferramenta MATE e foi possível identificar violações desta recomendação.

4.2.6 Conteúdo dimensionável

Usuários devem poder controlar o tamanho da fonte dos conteúdos apresentados na tela para poder melhor visualizar seu conteúdo. Essa adaptação pode se dar de forma constante ou temporária.

O uso de tecnologias que já proveêm o uso de zoom de forma nativa já facilitam o trabalho entregando escalabilidade que implica que cada cliente pode redimensionar e usar a barra e rolagem, enquanto preserva as posições corretas dos ponteiros e anotações para as várias resoluções (LEE; KO; FOX, 2003).

Violação: A aplicação não permite aplicar zoom nas telas da aplicação, ou quando o zoom é aplicado, a barra de rolagem não aparece impedindo ainda de navegar pela aplicação com o zoom definido.

Exemplo: A figura 9 apresenta a tela de uma aplicação móvel que não pode ser aplicado zoom utilizando o recurso de ampliação de tela do Android, e esta tela tem vários textos e itens clicáveis.



Figura 9 – Exemplo de violação de conteúdo dimensionável.

Avaliação manual: Para verificar manualmente, abra a aplicação e verifique se é possível ampliar a tela. Verifique se o conteúdo da aplicação pode ser acessado quando o zoom está definido. Além disso, mude o tamanho da fonte do texto e verifique se a barra de rolagem está habilitada e se ainda é possível acessar o conteúdo da tela.

Possibilidade de avaliação automática: Não Automatizável.

Justificativa: Não foi possível implementar mecanismos para redimensionar a tela ou a fonte da aplicação de forma automática utilizando a API fornecida pelo Android.

4.2.7 Elementos acionáveis

Links e outros elementos acionáveis devem estar claramente distinguidos. Qualquer usuário deve conseguir visualizar um elemento acionável mesmo na forma de link, botão, navegação e deve ser óbvio. Usuário também devem ser capazes de acionar esses elementos naturalmente. Quando os controles padrões da plataforma são utilizados, isso facilita o uso pois os usuários já estão familiarizados.

O propósito de cada elemento acionável deve estar descrito no elemento para que seja entendido fora de contexto. Leitores de tela possuem a funcionalidade de listar os links de uma tela. No entanto, se o texto dos links não for suficientemente descritivo, essa funcionalidade terá pouco valor (EMAG, Acesso em: 23/04/2019).

Violação: Elementos acionáveis que não são distinguidos visualmente e não são listados pelos leitores de tela.

Exemplo: A Figura 10 apresenta links que não estão claramente distinguidos em sua apresentação. O logotipo e o texto “UNINOVE” parecem um título, mas são links para a página da instituição.

Avaliação manual: Ative um leitor de tela, identifique os elementos acionáveis e verifique se são distinguíveis visualmente. Por exemplo, deve-se verificar se um botão tem aparência de botão, por exemplo. Em muitas aplicações os botões não possuem borda ou plano de fundo e podem ser confundidos como simples rótulos. Utilize o leitor para determinar se todos os elementos acionáveis estão sendo listados pelo leitor de tela.

Possibilidade de avaliação automática: Parcialmente automatizável.

Avaliação automática: É possível avaliar se alguns componentes nativos da plataforma e que são acionáveis são claramente distinguíveis. Por exemplo, é possível



Figura 10 – Exemplo de violação de elementos acionáveis.

definir se um botão tem as características de um botão, como borda ou cor de fundo distinta. Ou se um label tem um comportamento adicionado a ele quando outros componentes deveriam ser utilizado para isso. Entretanto, não é possível garantir que seja possível identificar se todos os elementos acionáveis são ou não distinguíveis na tela.

Para verificar automaticamente os elementos acionáveis da tela deve-se seguir os seguintes passos:

1. Localize os elementos acionáveis da tela.
2. Utilizando processamento de imagem, identifique se o elemento possui borda ou cor de fundo distinta da cor de fundo de seu contexto. Neste caso, emita um alerta de violação.
3. Identifique os elementos de texto que sejam acionáveis. Neste caso, emita um alerta de violação.

Avaliação da possibilidade de automatização: A avaliação automática desta recomendação foi implementada na ferramenta MATE. A execução da ferramenta permite identificar botões que não possuem borda ou plano de fundo diferente do plano de fundo da tela, caso em que o botão fica parecido com um texto avulso na tela. Também foi possível identificar rótulos ou textos que possuem comportamento adicionado a ele. Como

não é possível determinar se todos os elementos acionáveis são claramente distinguíveis, principalmente os criados pelos usuários e que não são da plataforma, considera-se que a diretriz é parcialmente automatizável.

4.2.8 Foco visível

Quando focados, todos os elementos devem claramente indicar que estão focados por meio de uma alteração visual. Por exemplo, ao clicar em um campo de entrada, o usuário deve claramente distinguir que os dados inseridos serão inseridos naquele campo específico. Em geral, isso é indicado por um cursor piscante e possivelmente pela mudança de cor da borda deste tipo de elemento.

Violação: Não é possível distinguir qual elemento da tela está com o foco da aplicação.

Exemplo: Na Figura 11 não é possível distinguir claramente se o campo de busca está focado ou não. O cursor piscante tem a mesma cor do texto e tem pouco contraste com o plano de fundo.

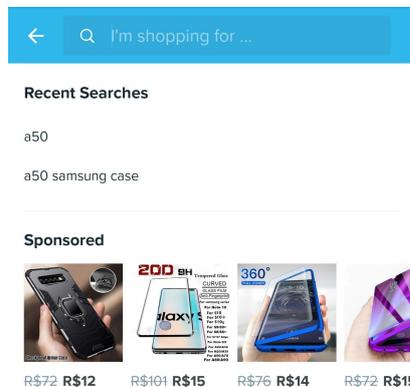


Figura 11 – Exemplo de violação de foco visível.

Avaliação manual: Para verificar manualmente se o foco é visível, verifique se é possível perceber visualmente quando um elemento da tela recebeu o foco. Verifique se os campos de entrada indicam que os dados serão ali inseridos e se telas que possuem abas mostram claramente qual aba está ativa.

Possibilidade de avaliação automática: Automatizável.

Avaliação automática: Para verificar automaticamente se foco é visível é possível seguir os seguintes passos:

- Identifique o item de interesse na tela.
- Registre o estado do item antes de receber o foco. Pode-se gerar uma imagem da área de interesse (coordenadas do item) ou armazenar informações sobre cores utilizadas, quantidade de pixels de cada cor, etc.
- Faça com o que o item receba o foco por meio de um click gerado.
- Compare o estado do item que recebeu o foco com o estado antes de receber o foco.
- Se o estado for o mesmo, emita uma alerta de violação de recomendação.

Em muitos casos, uma mudança sutil no elemento que recebe o foco pode não ser claramente visível pelo usuário, mas será percebido pela avaliação automática como uma mudança de estado. Portanto, recomenda-se que seja definido um limiar de mudança para indicar que de fato houve uma mudança de estado.

Avaliação da possibilidade de automatização: Foi possível implementar a avaliação desta recomendação na ferramenta MATE e detectar situações em que campos de texto que receberam o foco não tiveram uma mudança de estado significativa e por isso uma violação de acessibilidade foi identificada.

4.2.9 Consistência

A experiência do usuário deve ser consistente. Uma experiência consistente significa que a estrutura lógica e a linguagem usada da mesma forma em toda a aplicação ajuda os usuários a entender onde e como navegar.

Devido à falta de consistência, os usuários podem acabar confusos em encontrar e aprender as operações básicas da interface tornando mais difícil do que se usado padrões consistentes. Consistência também se aplica para o layout geral e posição na tela de elementos como menus, botões de saída e outros controles, uso de terminologia padrão (por exemplo, para arquivos, operações como "Novo", "aberto" e "fechado" usados em todo o sistema também ajuda a manter a consistência (KUSHNIRUK; PATEL, 2004).

Violação: Uso de diferentes menus ou botões em diferentes telas da aplicação com o mesmo propósito, uso de diferentes layouts em telas e diferenciação na linguagem podem ser definidos como violações dessa diretriz.

Exemplo: Os itens da aplicação apresentada nas Figuras 9 e 12 são apresentados de forma diferente. Na primeira imagem é apresentado um menu em grade e na segunda um menu vertical.

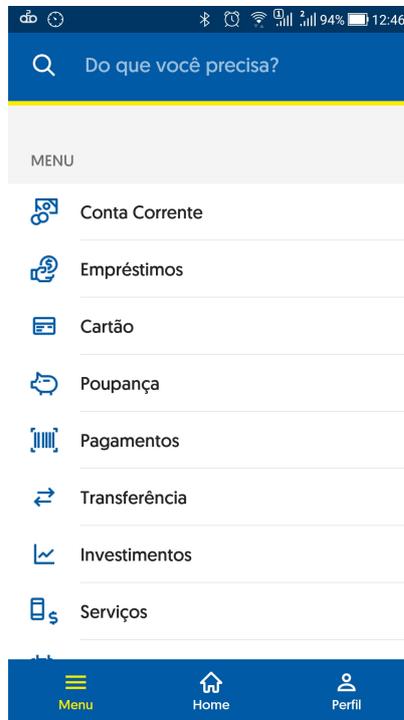


Figura 12 – Violação de consistência.

Avaliação manual: Para verificar manualmente, abra a aplicação, verifique o uso da linguagem, os botões e menus em diversas telas da aplicação, observe se são utilizados os mesmos menus e textos com as mesmas funções em toda a aplicação.

Possibilidade de avaliação automática: Não automatizável.

Justificativa: Não é possível verificar de forma automática se a estrutura do sistema é lógica ou mesmo se a linguagem utilizada é de fácil compreensão sem a avaliação do usuário. Não é possível também verificar se componentes do mesmo tipo são acionados de forma diferentes. Computacionalmente não é possível verificar se está sendo utilizada uma estrutura lógica em uma aplicação e se essa mesma estrutura está sendo utilizada em todas as telas da aplicação.

4.2.10 Escolha

As interfaces devem prover múltiplas maneiras de interagir com o conteúdo, pois dessa forma os usuários podem escolher como navegar pelo sistema. Prover múltiplas

maneiras de interagir com o conteúdo oferece aos usuários uma opção e aumenta a acessibilidade. Essa multiplicidade de interações compreende a diferentes expectativas do usuário sobre como interagir com os elementos, e os usuários que têm dificuldades com um tipo de interação podem usar outro.

Essa diretriz visa verificar se um usuário consegue acessar a aplicação de por outro meio seja, mouse ou teclado além do touch.

Violação: Ao alterar a forma de interação com a aplicação, a aplicação deixa de se comportar da maneira esperada.

Exemplo: Não foi possível incluir um exemplo no guia de escolha de interação com uma aplicação móvel.

Avaliação manual: Para verificar manualmente a escolha, inclua uma maneira de interagir com a aplicação móvel diferente da convencional, como mouse ou teclado, navegue pela aplicação verificando se o comportamento da aplicação se mantém o mesmo.

Possibilidade de avaliação automática: Não automatizável.

Justificativa: Não é possível verificar de maneira automática se ao trocar a forma de interação com a aplicação toda a aplicação continua a se comportar da maneira esperada.

4.2.11 Ajustabilidade

Mídias interativas devem ser ajustáveis pelo usuário conforme sua habilidade e preferência. A aplicação deve respeitar as configuração de mídia previamente ajustadas na configuração do aparelho, pois dessa forma oferece uma experiência semelhante a que o usuário já está acostumado. Os usuários devem ser capaz de ajustar cores de textos, luminosidade, legendas, entre outras coisas.

Violação: A aplicação não oferece opções de configuração.

Exemplo: Não foi possível encontrar exemplos de violação desta recomendação.

Avaliação manual: Verifique se existe a opção de configurações, se as configurações funcionam adequadamente, e se os conteúdos de mídia acionados obedecem as configurações previamente ajustados no dispositivo móvel.

Possibilidade de avaliação automática: Não automatizável.

Justificativa: É possível verificar se a aplicação fornece a opção de configurações por meio de nomes padrões como “configurações” ou “settings” em inglês, mas não é

possível verificar automaticamente se os ajustes feitos pelos usuários são de fato respeitados pela aplicação.

4.2.12 Flicker

Cintilação da tela ou *flashes* não devem ser acionados mais do que três vezes em um período de 1 segundo. Todos os usuários são suscetíveis a mal estar quando utilizado o conteúdo de flashes contínuos, mas para pessoas com algumas doenças o uso do desse artifício pode causar náuseas, dores de cabeça, ataques epiléticos entre outros sintomas.

Violação: Conteúdos de flash acionados pela aplicação mais de 3 vezes em um intervalo de 1 segundo.

Exemplo: A violação não pode ser percebida por uma imagem estática, mas a Figura 13 mostra um exemplo de uma sequência de imagens de uma aplicação que mostra claramente a violação.

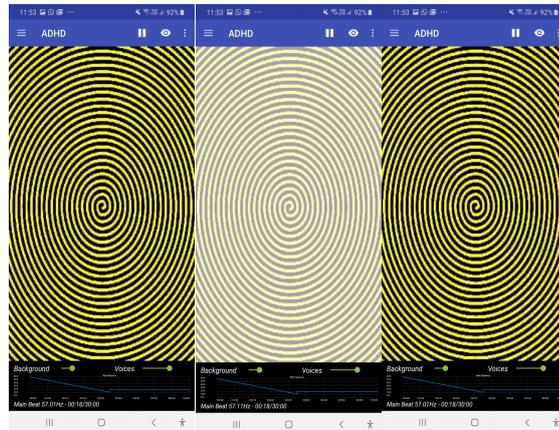


Figura 13 – Violação da recomendação Flicker.

Avaliação manual: Para verificar o uso de flashes na aplicação, navegue pela aplicação e verifique se durante o uso é notada que a aplicação pisca. A avaliação manual desta recomendação pode ser limitada pois é difícil saber se três ou mais flashes foram emitidos no período de um segundo. Entretanto, visualmente é possível perceber se a tela está piscando mais do que o que seria considerado comum para qualquer aplicação.

Possibilidade de avaliação automática: Automatizável

Avaliação automática: Para verificar automaticamente o uso de flicker:

- Gere várias imagens (frames) da tela durante 1 segundo.

- Utilize um algoritmo de segmentação de cores da imagem e identifique as cores mais claras e as cores mais escuras utilizando um limiar.
- Verifique se há uma alternância entre as cores mais claras e mais escuras ao analisar as várias imagens geradas durante 1 segundo.
- Se houver mais do que três alternâncias de cores claras-escuras-claras, emita um alerta de violação de recomendação.

Avaliação da possibilidade de automatização: Até a finalização deste guia a automatização desta recomendação ainda não havia sido implementada na ferramenta MATE. Entretanto, acredita-se que esta recomendação seja totalmente automatizável dada a possibilidade de utilizar algoritmos de processamento de imagem para identificar as alternâncias entre as cores claras e escuras de sucessivas imagens.

4.3 Formulários

4.3.1 Rótulos dos controles dos formulários

Todos os controles do formulários devem conter rótulos. Os rótulos ajudam os usuários a entender o que cada item do formulário é e o que é esperado do usuário. A entradas dos formulários podem ser de diferentes formas, como entradas de texto, radio buttons, check boxes, seleção em uma lista, entre outros. Propriedades que fornecem dicas (*hint*) ou descrevem o item, embora úteis, são temporários em muitas vezes e não substituem os rótulos.

Violação: Os elementos de controle do formulário não possuem um rótulo associado.

Exemplo: Na figura 4, além do problema de contraste já apresentado, as entradas dos formulários não contém rótulos, apenas a propriedade *hint* está definida.

Avaliação manual: Para verificar manualmente é preciso ligar o leitor de tela, navegar em cada item do formulário, e verificar se para todos os itens o leitor de tela é capaz de anunciar o rótulo e também se o rótulo está visível.

Possibilidade de avaliação automática: Automatizável.

Avaliação automática: Para verificar automaticamente os títulos do formulário pode-se executar os seguintes passos:

1. Acessar as propriedades do item de interesse e identificar se a propriedade *labelBy* está definida. Se a propriedade não estiver definida execute o passo 2.
2. Acessar as propriedades dos elementos de texto do formulário e verificar se a propriedade *labelFor* de algum deles refere-se ao item de interesse. Se a propriedade não estiver definida execute o passo 3.
3. Acessar as propriedades do item de interesse e verificar se a propriedade *hint* está definida. Se a propriedade não estiver definida emitir um alerta de violação de recomendação. Se a propriedade estiver definida execute o passo 4.
4. Verifique se a propriedade *hint* está associada a um campo que possui um rótulo flutuante (*floating label*), em que ao dar o foco no campo de entrada o texto da propriedade *hint* assume a posição de um rótulo. Se o campo não tiver um rótulo flutuante, emita um alerta de violação de acessibilidade.
5. Em todos os casos, verifique se campos de entrada de texto ou números possuem a propriedade *contentDescription* definida. Se a propriedade estiver definida emita um alerta de que violação de recomendação, pois esta propriedade é sempre lida pelo leitor de tela em conjunto com o conteúdo do campo, o que pode causar confusão para o usuário.

Avaliação da possibilidade de automatização: a avaliação automática desta recomendação já estava implementada na ferramenta MATE e em outras ferramentas anteriormente (SILVA; ELER; FRASER, 2018). Portanto, considera-se que de fato a avaliação desta recomendação pode ser feita automaticamente.

4.3.2 Entrada de formulários

O tipo de entrada padrão do formulário deve ser indicada, seja texto, número, data ou outro tipo de entrada específico, pois isso reduz os erros de preenchimento do formulário. A indicação de tipo de entrada pode ser especificada como parte do rótulo do item do formulário e o teclado no modo correto deve ser aberto.

Violação: Não haver indicação de tipo de entrada no título do formulário.

Exemplo: Na figura 4, além do problema de contraste e falta de títulos nos formulários já apresentados, não há indicação de tipo de entrada uma vez que também não há título.

Avaliação manual: Para verificar manualmente é preciso ligar o leitor de tela, navegar até os itens do formulários e verificar se há indicação visual do tipo de entrada de formulário e se o leitor de tela anuncia o tipo de dado de entrada.

Possibilidade de avaliação automática: Automatizável.

Avaliação automática: Para verificar automaticamente a indicação de entrada de formulário:

1. Acesse as propriedades do item de formulário de interesse.
2. Verifique se a propriedade que define o tipo de entrada do campo de entrada está definido. Emita uma violação de acessibilidade se a propriedade não estiver definida.

Avaliação da possibilidade de automatização: A avaliação automática desta recomendação foi implementada na ferramenta MATE e foi possível identificar violações que também foram encontradas manualmente. Ressalta-se que é possível saber se existe um tipo de entrada indicado, mas não é possível avaliar se o tipo de entrada está correto de acordo com os objetivos do formulário e da aplicação em si.

4.3.3 Layout do formulário

Os rótulos dos itens de formulário ajudam o usuário a entender como preenchê-lo. Os rótulos precisam ser posicionados próximos aos itens a que se referem para que o usuário não fique desorientado ao que responder em cada questão. Os títulos devem estar em cima do controle de entrada do formulário ou do lado esquerdo para ser mais intuitivo a quem está o respondendo.

Violação: Título longe da entrada do formulário, título do lado direito ou abaixo da entrada do formulário.

Exemplo: Na figura 4, o formulário não contém títulos, havendo apenas as caixinhas para inclusão do texto.

Avaliação manual: Para verificar manualmente é preciso primeiro usar um leitor de tela para saber se um rótulo está associado ao campo de entrada do formulário. Em seguida, é preciso verificar se os rótulos estão próximos aos campos de entrada conforme orientação da recomendação.

Possibilidade de avaliação automática: Automatizável.

Avaliação automática: Para verificar automaticamente o layout do formulário:

1. Identificar as coordenadas na tela do item de interesse.
2. Identificar o rótulo do item de interesse. Se não houver rótulo emitir um alerta de violação de acessibilidade.
3. Verificar se o rótulo está à esquerda ou acima do item de interesse utilizando as coordenadas de ambos os componentes. Se o rótulo não estiver à esquerda ou acima e a uma distância mínima definida pelo usuário (e.g. máximo 10 pixels), emitir um alerta de violação de acessibilidade.

Avaliação da possibilidade de automatização: A avaliação automática desta recomendação foi implementada na ferramenta MATE e ela foi capaz de detectar violações de acessibilidade que também foram detectadas manualmente.

4.3.4 Agrupamento de elementos

Controles, rótulos e outros elementos do formulário devem estar agrupados quando forem relacionados. O agrupamento dos elementos relacionados ajuda os usuário a usar melhor o formulário e reduz a complexidade para aqueles que usam tecnologias assistivas. Para o agrupamento dos elementos devem ser considerados: a correta associação dos rótulos aos itens do formulário, o relacionamento correto associado a radio buttons e check boxes, e o correto agrupamento de itens do formulário em containers.

Violação: As seguintes situações configuram uma violação: itens do formulário estão agrupados mas não há uma identificação visual sobre o que se refere aquele grupo; para cada item do formulário dentro de um grupo o leitor de tela não anuncia primeiro o rótulo do grupo e depois o rótulo do item; não é possível navegar entre os grupos de itens do formulário independentemente.

Exemplo: A plataforma Android não possui métodos para agrupar elementos com exceção de radio group para agrupar radio buttons. Portanto nenhum aplicativo possui agrupamentos, com exceção do agrupamento de radio buttons.

Avaliação manual: A avaliação manual consiste em ligar o leitor de tela e navegar entre os grupos e acessar os itens de cada grupo. Deve-se verificar se existe um identificador visual de cada grupo, se é possível navegar pelos grupos, e se para cada elemento de um grupo o leitor de tela primeiro anuncia o rótulo do grupo e depois o rótulo do item. Para

que isso seja possível é preciso que os rótulos dos elementos do formulário tenham sido definidos.

Possibilidade de avaliação automática: Não automatizável.

Justificativa Não é possível identificar se a existência de mais de um radio button exige a existência de um radio group, pois não é possível definir se há uma associação entre os itens do tipo radio button ou se eles podem ser usados independentemente no formulário. A verificação se o radio group possui um rótulo já é feita por outra recomendação.

4.3.5 Foco manuseável

Foco ou contexto não podem mudar automaticamente durante as entradas do formulário. O foco da tela pode mudar apenas através da interação do usuário porque isso pode atrapalhar usuários a verificar a informação digitada ou corrigir erros se o foco muda automaticamente quando isso não é esperado.

Violação: Mudança inesperada no foco ou contexto da aplicação durante uma interação do usuário, sem que esta tenha sido solicitada.

Exemplo: Ao digitar o número da agência na tela apresentada na Figura 14, o foco muda para o campo conta corrente automaticamente sem a solicitação do usuário.

← Nova transferência

+ ☆ ↺

DADOS DE DESTINO

Banco
 Banco do Brasil S.A. 001 >

Tipo de conta
 Conta Corrente ▾

Agência
 8480-8

Conta

Troque o X por 0, se necessário

^ ▾ AVANÇAR

Figura 14 – Violação de foco manuseável.

Avaliação manual: Para verificar manualmente a mudança inesperada de foco, navegue pelo formulário e clique em cada entrada de formulário e verifique se o foco é alterado para alguma entrada durante ou ao finalizar o seu preenchimento.

Possibilidade de avaliação automática: Automatizável.

Avaliação automática: Para verificar automaticamente o foco manuseável dos formulários pode-se executar os seguintes passos:

1. Clique no item de interesse (campo de entrada).
2. Insira o tipo de dado indicado até o limite indicado na propriedade do item. Se não houver, insira o limite do campo.
3. Verifique se o foco permanece no item. Emita um alerta de violação de acessibilidade se o item perdeu o foco.

Avaliação da possibilidade de automatização: A avaliação automática desta recomendação foi implementada na ferramenta MATE e foi possível identificar violações de recomendação que também foram encontradas manualmente.

4.4 Links

4.4.1 Links descritivos

Links e itens de navegação devem descrever unicamente o alvo ou a função do item. Links e itens de navegação precisam ser unicamente identificáveis porque nem sempre os usuários de leitores de tela conseguem entender o contexto de todas as ações. Se há dois itens de navegação com o mesmo nome, como, por exemplo, “clique aqui para ver mais informações”, o usuário pode não saber do que se trata. Itens com texto duplicados devem ter uma descrição de conteúdo (propriedade *contentDescription*) para que o usuário de leitor de tela diferencie um do outro.

Violação: Mais de um link ou item de navegação possuem o mesmo texto e não há diferenciação por meio da descrição de conteúdo.

Exemplo: Na Figura 16, os botões “SELECT” dizem respeito à reserva de diferentes quartos em um hotel, mas possuem o mesmo nome. Usando um leitor de tela não foi possível distinguir para qual quarto cada botão se referia.

Avaliação manual: É preciso ativar o leitor de tela e selecionar os links ou itens de navegação para verificar se o leitor anuncia uma descrição diferente para itens com textos repetidos, o que significa que cada item é unicamente identificado. Ainda, é preciso verificar se a descrição em si é suficiente para que o usuário entenda do que se trata.

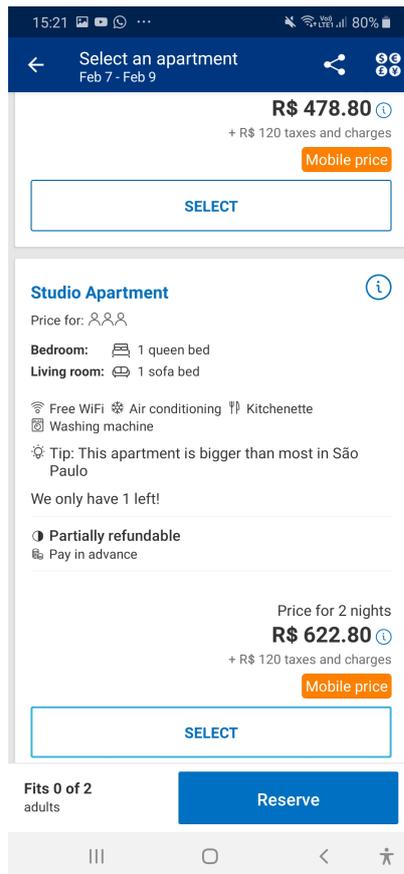


Figura 15 – Botões com o mesmo nome e sem descrição adicional para diferenciar a ação de cada um.

Possibilidade de avaliação automática: Parcialmente automatizável.

Avaliação automática: É possível identificar se links ou itens de navegação com textos iguais possuem descrições de conteúdo diferentes, mas não é possível decidir se as descrições são claras o suficiente para apresentar seu propósito.

Para verificar automaticamente os links descritivos de uma aplicação:

1. Encontre links e itens de navegação que possuem o mesmo texto.
2. Para cada par de itens com o mesmo texto, verifique se possuem descrição de conteúdo diferentes.
3. Emita um alerta de violação de recomendação se dois itens com o mesmo texto não tiverem diferentes descrições de conteúdo.

Avaliação da possibilidade de automatização: A implementação da avaliação automática desta diretriz foi implementada na ferramenta MATE e foi possível detectar violações de acessibilidade também encontradas manualmente. Entretanto não é possível

decidir se as descrições dos links e itens navegáveis são suficientes para anunciar o propósito daquele item de forma clara e objetiva.

4.4.2 Links para formatos alternativos

Links para formatos alternativos devem indicar que uma página alternativa será aberta ou que o usuário irá sair da aplicação atual. A abertura de links para sites fora da aplicação ou em outra aplicação de maneira inesperada pode deixar o usuário que não consegue perceber essa mudança de forma visual confuso. Por isso se deve informar ao usuário que ao clicar no link ele será redirecionado para outra aplicação.

Violação: Um link levar o usuário a sair da aplicação atual sem nenhum aviso.

Exemplo: Na figura 16 o menu Pontos Lívolo leva o usuário a abrir outra aplicação sem nenhum aviso.

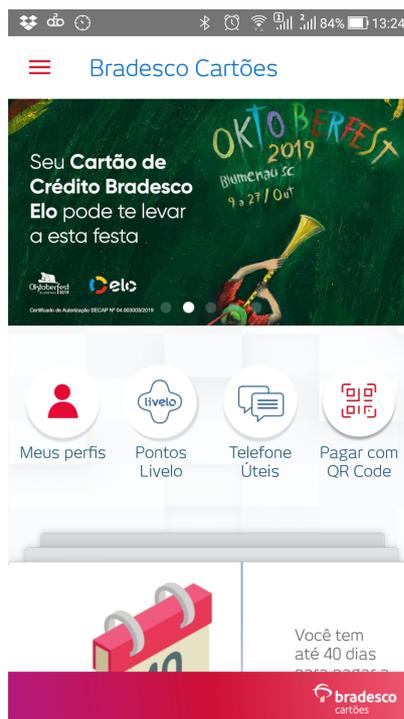


Figura 16 – Botões com o mesmo nome e sem descrição adicional para diferenciar a ação de cada um.

Avaliação manual: É preciso abrir o leitor de tela e acionar os links e itens de navegação da aplicação da aplicação, e então verificar se para os links que saem da aplicação ou mudam o formato há um aviso de que isso vai acontecer.

Possibilidade de avaliação automática: Parcialmente automatizável.

Avaliação automática: É possível verificar se ao acionar os links ou itens de navegação uma nova aplicação é aberta, mas não é possível definir se um aviso foi emitido de forma precisa. É possível seguir os seguintes passos para isso:

1. Acione cada link ou item de navegação da aplicação.
2. Identifique se o efeito da ação provocou a abertura de outra aplicação ou a opção de seleção da aplicação que deve ser usada para completar a ação, como, por exemplo, compartilhar uma imagem da galeria de fotos. A verificação se a aplicação sob teste continua sendo executada pode ser feita consultando qual é o pacote que está sendo executado naquele exato momento pelo Android.
3. Se uma nova aplicação foi aberta, ou a seleção de outras aplicações, emitir um alerta para que a validação manual seja realizada.

Avaliação da possibilidade de automatização: A avaliação automática desta recomendação foi implementada de tal forma que ações que levam a outras aplicações são identificadas para que posteriormente sejam validadas manualmente.

4.4.3 Combinando links repetidos

Links repetidos devem ser combinados em um único link para ajudar os usuários que utilizam leitores de tela a navegar de maneira mais rápida. Links repetidos podem ser imagens, título e tópico quando na mesma página.

Violação: Mais de um link levando a mesma página.

Exemplo: O mesmo link que aparece na Figura 16 com o texto “Pontos Nivelô” se repete ao final da tela apenas com o nome “Nivelô”, como mostrado na Figura 17, mas que leva à mesma aplicação externa.

Avaliação manual: Para verificar links repetidos manualmente é preciso navegar pela aplicação e verificar se existem mais de um link para o mesmo endereço ou itens de navegação que levam a aplicação para o mesmo lugar.

Possibilidade de avaliação automática: Não Automatizável.

Justificativa: Ainda que seja possível verificar por meio de análise estática se mais de um item de navegação tem o mesmo comportamento, não é possível realizar tal automatização utilizando a API do Android e as propriedades fornecidas para coletar informações dos componentes da tela e da aplicação em si.



Figura 17 – Exemplo de violação de links repetidos.

4.5 Notificações

4.5.1 Notificações inclusivas

Notificações devem ser visíveis e audíveis. Todos os usuários se beneficiam quando as notificações são claras e podem ser percebidas de diferentes formas. As notificações aqui colocadas englobam as notificações não somente quando o dispositivo móvel está inativo quanto por exemplo, a indicação de erro de preenchimento de um formulário.

As notificações visíveis podem ser feitas usando as notificações padrão do sistema operacional e as notificações audíveis podem ser feitas utilizando as notificações padrão utilizadas por tecnologias assistivas.

Violação: Notificações não visíveis ou audíveis além de notificações que não podem ser anunciadas por leitores de tela.

Exemplo: As notificações não inclusivas não são facilmente percebíveis e não foi possível coletar um exemplo.

Avaliação manual: Para verificar manualmente é preciso ligar o leitor de tela e navegar pela aplicação completando ações onde são ativadas mensagens de erro, notificações

de sucesso de tarefas, e verificar se as mesmas são visíveis e audíveis, inclusive os textos das mensagens.

Possibilidade de avaliação automática: Não Automatizável.

Justificativa: Não foi possível implementar a diretriz conforme condições apresentadas no guia, não identificamos computacionalmente se um alerta é visível ou audível ou até mesmo se a notificação não é um elemento esperado e dessa forma definimos a recomendação como não automatizável.

4.5.2 Notificações do sistema operacional

Os meios padronizados do sistema operacional de emitir notificações devem ser usadas quando disponíveis e apropriados. O usuário que utiliza as tecnologias assistivas já configurou as notificações de seu dispositivo móvel da maneira mais adequada ao seu uso, ao utilizar as notificações padrão, por default, a aplicação passa a estar configurada da forma que o usuário pré determinou facilitando o seu uso. Outra vantagem do uso de notificações padrão é o fato das mesmas já estarem adequadas ao uso de tecnologias assistivas.

Violação: Não utilizar as notificações padrão do sistema operacional para a aplicação.

Exemplo: As aplicações que não utilizam os meios padrão do sistema operacional para emitir notificações não são facilmente localizáveis, portanto não é apresentado um exemplo aqui.

Avaliação manual: Para verificar manualmente a utilização de notificações padrão ligue o leitor de tela, navegue pela aplicação verificando as notificação de novidades, tempo, erros entre outros alertas e verifique se estão de acordo com o padrão do sistema operacional.

Possibilidade de avaliação automática: Não Automatizável.

Justificativa: É possível identificar quando notificações utilizam o padrão do sistema operacional, mas não é possível identificar quando uma notificação não utiliza o padrão do sistema operacional, pois não é possível distinguir se uma notificação foi emitida quando os meios padronizados da plataforma não foram utilizados.

4.5.3 Mensagens de erro e correção

Mensagens de erro claras ajudam todos os usuários a perceber onde está o erro indicado para fazer a correção adequadamente. As mensagens de erro inclusivas ajudam ainda pessoas que utilizam tecnologias assistivas. Como alguns usuários não estão de fato vendo a tela, incluir notificações em forma de figuras ou cores não é uma boa alternativa, pois isso vai dificultar para esses usuários perceber onde está o erro. As mensagens de erro devem ser informativas e estarem claras e sucintas.

Violação: Não haver mensagens de erro ou alerta além de não ter informações suficientes de onde se encontra o erro.

Exemplo: Na figura 18 não é apresentada nenhuma mensagem de alerta sobre onde está o erro no preenchimento do formulário. Isso é feito apenas de forma visual pela mudança de cor do campo.

The image shows a mobile application interface for Booking.com's car rental service. The title is "Encontre seu carro ideal". The form includes the following fields and options:

- Local de retirada:** A text input field with a pink background and the placeholder text "p. ex. cidade ou aeroporto".
- Devolver o carro no mesmo local
- Data de retirada:** A date picker showing "19 Ago Seg" at "10:00".
- Data de devolução:** A date picker showing "22 Ago Qui" at "10:00".
- Conductor entre 30 e 65 anos de idade?
- Cancelamento grátis na maioria das reservas**
- Pesquisar** (blue button)
- Footer: "Dirija. Descubra."

Figura 18 – Exemplo de violação da apresentação de mensagem de erro em formulários.

Avaliação manual Para verificar manualmente é preciso ligar o leitor de tela, navegar pela aplicação verificando se existem mensagens de erro, e se estão claras e se são audíveis para o leitor de tela.

Possibilidade de avaliação automática: Parcialmete Automatizável.

Avaliação automática: É possível identificar se houve alguma mudança de estado resultante de uma ação, mas não é possível saber se a mudança é relativa a algum erro. Quando há uma mudança de estado e a propriedade *error* dos campos do formulário são preenchidos, há indícios de que um erro aconteceu. Quando há mudança de estado e não há preenchimento das propriedades de erro, não é possível saber se de fato alguma indicação de erro foi feita. Portanto, é possível apenas alertar o usuário para que uma validação manual seja feita.

Os seguintes passos podem ser executados para a avaliação automática:

1. Identifique mudanças de estado (cores diferentes de itens do mesmo tipo). Para isso pode ser utilizado o algoritmo utilizado no Cor e Significado.
2. Se houver mudança de estado, verifique se há mensagens de erros associadas aos campos (propriedade *error*).
3. Se não houver mensagens de erros associadas, emita um alerta para que uma possível violação de recomendação seja validada manualmente.

Apesar de ser possível detectar possíveis situações em que mensagens de erros são apresentadas no próprio formulário, não é possível determinar se as mensagens de erros são precisas e instrutivas.

Avaliação da possibilidade de automatização: A avaliação automática desta diretriz foi implementada na ferramenta MATE e foi possível localizar potenciais situações em que mensagens de erro deveriam ser exibidas mas não foram, mostrando assim que é possível emitir alertas para posterior verificação manual.

4.5.4 Feedback e assistência

Feedback ou assistência devem ser providas quando apropriado. Esses recursos ajudam os usuários a entender e utilizar certa funcionalidade da aplicação.

A assistência deve ajudar também o usuário a entender o real objetivo da aplicação e se está utilizando de maneira correta.

Violação: A aplicação não prover feedback e assistência.

Exemplo: A aplicação mostrada na figura 18 também não possui nenhum link de feedback e assistência para ajudar o usuário a preencher os itens do formulário corretamente.

Avaliação manual: Para verificar se a aplicação provê feedback e assistência navegue pela aplicação e verifique se a mesma possui esse recurso, ligue o leitor de tela e verifique ainda se a assistência é audível pelo leitor de tela.

Possibilidade de avaliação automática: Não Automatizável.

Justificativa: Não é possível verificar se algumas mensagens da aplicação podem ser consideradas feedback e assistência ou apenas alertas ou um texto descritivo, por exemplo. Por esse motivo definimos esta recomendação como não automatizável.

4.6 Textos equivalentes

4.6.1 Alternativas para elementos não-textuais

Textos Alternativas para elementos não-textuais servem para que tecnologias assistivas consigam transmitir o significado de imagens ou ícones no contexto em que são apresentadas. Os leitores de tela não conseguem analisar a imagem, portanto é necessário que seja indicado o conteúdo e a função de cada imagem da tela. Essa indicação ainda deve fazer sentido com o contexto e deve ser única, a fim de não confundir o usuário. Deve-se evitar textos redundantes como “imagem de” ou “selecione para”, pois o leitor de tela anuncia o tipo de objeto ao anunciar o texto alternativo.

Violação: Elementos não-textuais sem alternativa textual.

Exemplo: Na Figura 19, o item de visualizar a senha, que é a imagem de um olho riscado, não tem descrição textual. Desta forma, o leitor de tela identificará que há uma imagem ali, ou um ícone, mas não saberá o efeito resultante de acionar aquele elemento. Ainda que o usuário cego não precise ou consiga ver sua senha, há o risco de acionar o botão sem querer e ela ser revelada para alguém que possa estar observando sua tela.

Avaliação manual: É preciso acionar o leitor de tela e navegar pelos elementos não-textuais. É preciso verificar se em cada elemento não-textual o leitor de tela anuncia um texto alternativo.

Possibilidade de avaliação automática: Parcialmente automatizável.

Avaliação automática: É possível identificar se existe um texto alternativo para elementos não-textuais, mas não é possível decidir se o texto é adequado ou suficiente ao contexto, e por isso só é possível automatizar parcialmente a avaliação desta recomendação. Os seguintes passos podem ser seguidos para fazer a avaliação automática desta diretriz:

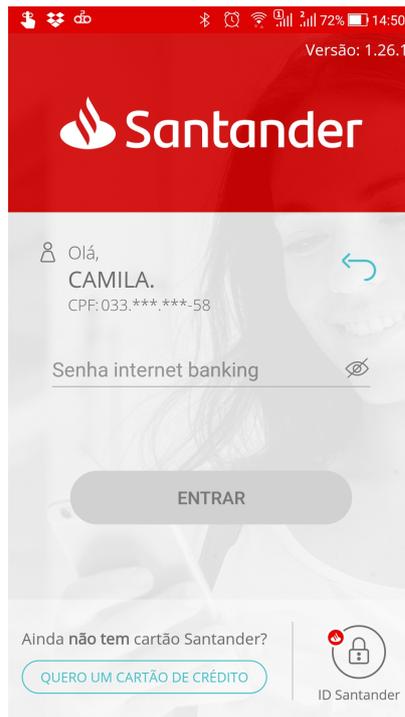


Figura 19 – Item não-textual sem alternativa de descrição por texto

- Acesse as propriedades do item de interesse (elemento não-textual).
- Verifique se a propriedade *contentDescription* ou *hint* foram definidas.
- Caso não haja texto alternativo (passo 2), emita uma alerta de violação de recomendação de acessibilidade.

Avaliação da possibilidade de automatização: A diretriz já havia sido implementada na ferramenta MATE e outras ferramentas antes da escrita do guia (SILVA; ELER; FRASER, 2018), portanto essa recomendação é considerada automatizável.

4.6.2 Conteúdo decorativo

Imagens decorativas devem ser escondidas de tecnologias assistivas. Se uma imagem ou ícone não tem nenhuma função na aplicação, não devem ser mencionadas por tecnologias assistivas.

Violação: Imagens decorativas mencionadas por tecnologias assistivas.

Exemplo: Na figura 20, a imagem do logotipo da distribuidora de combustível pode ser acessado pelo leitor de tela e é anunciado como um elemento sem descrição

textual. Neste caso, o logo foi apresentado com um botão que não possui comportamento, e por isso ele pode ser identificado pelo leitor de tela.



Figura 20 – Exemplo de violação de conteúdo decorativo mencionado por leitor de tela

Avaliação manual: Para verificar manualmente o conteúdo decorativo, ligue o leitor de tela, navegue pela aplicação e verifique se as imagens decorativas são mencionadas pelo leitor.

Possibilidade de avaliação automática: Parcialmente automatizável.

Avaliação automática:

Para verificar automaticamente o conteúdo decorativo:

- Encontre os elementos não-textuais decorativos da aplicação, que são os elementos que não tem ação associada a ele. Isso pode ser obtido por meio das propriedades do item fornecidas pela API do Android.
- Verifique se os elementos decorativos com os quais o usuário não consegue interagir estão marcados na API como importantes para acessibilidade, o que significa que serão visíveis ao leitor de telas.

- Se conteúdos decorativos forem destacados como importantes para acessibilidade, emita um alerta de violação de acessibilidade.

O teste desta recomendação é parcialmente automatizável porque não é possível, pelo menos utilizando esta API, definir se elementos interativos como botões e *labels* acionáveis que podem ser imagens, por exemplo, foram utilizados para exibir um conteúdo decorativo. Neste caso o algoritmo identificará que é um item com o qual o usuário pode interagir por ser clicável e por isso não o classificará como decorativo. No entanto, para a aplicação aquele conteúdo é decorativo porque nenhum comportamento foi adicionado a ele. Ainda que seja possível realizar análise de código para entender quais elementos não possuem comportamento adicionado apesar de serem acionáveis, não se pode definir automaticamente se o elemento é decorativo ou se algum comportamento será adicionado em outras versões da aplicação, o que também configuraria um erro na construção de interface, mas de outra natureza.

Avaliação da possibilidade de automatização: A avaliação automática desta recomendação foi implementada na ferramenta MATE mas não foram encontrados exemplos para validar a implementação utilizando resultados da avaliação manual para os casos em que a ferramenta é capaz de identificar a violação da recomendação. O exemplo apresentado na Figura 20 é o caso em que a validação automática não consegue detectar pois o logo é exibido por meio de um botão e não um elemento não-textual. Portanto, a validação automática não consegue identificar que aquele botão não tem comportamento associado para o definir como decorativo.

4.6.3 Dicas e informações complementares

Dicas e informações complementares associadas aos elementos interativos da interface não podem incluir informações essenciais para o uso da aplicação uma vez que nem todos os usuários conseguem visualizar ou acessam o conteúdo adicional. As dicas e informações complementares devem adicionar uma explicação extra sobre os elementos da interface, portanto não devem simplesmente repetir o texto e o tipo de elemento. Por exemplo, no botão “Próxima tela” a informação adicional não pode ser “botão para ir para a próxima tela”.

Violação: Repetir o texto ou o tipo de elemento nas informações complementares ou fornecer informação essencial para interagir com a aplicação.

Exemplo: Para cada item do menu da Figura 21, o texto é repetido como informação complementar.

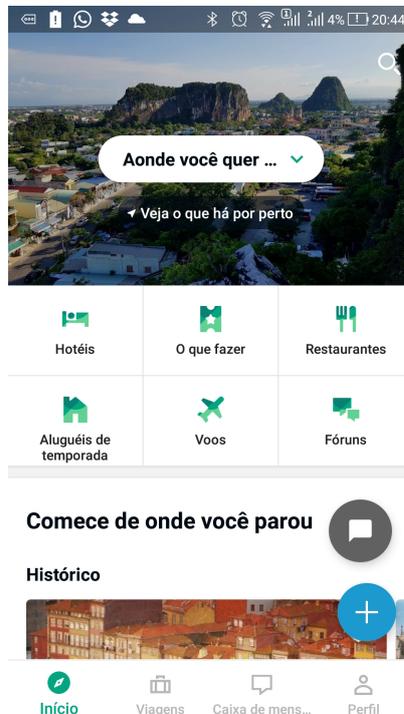


Figura 21 – Exemplo de violação de dicas e informações complementares

Avaliação manual: Ligue o leitor de tela e acesse os elementos acionáveis da tela. Verifique se a informação complementar apresentada pelo leitor de tela repete o texto ou o tipo de componente acionado.

Possibilidade de avaliação automática: Parcialmente automatizável.

Avaliação automática: No Android, a forma de adicionar dicas ou informações complementares é por meio da propriedade *contentDescription*. É possível identificar casos em que a informação complementar repete o texto ou o tipo de elemento de interação, mas não é possível decidir se informações essenciais estão sendo transmitidas utilizando informação complementar. Para avaliar automaticamente esta recomendação pode-se executar os seguintes passos:

- Acesse as propriedades do item de interesse.
- Verifique se o texto e o tipo do elemento é repetido no conteúdo da propriedade *contentDescription*.

- Se a propriedade repetir o texto ou o tipo de elemento, emita um alerta de violação de recomendação.

Avaliação da possibilidade de automatização: A avaliação automática desta recomendação foi implementada na ferramenta MATE e foi possível identificar situações em que o texto das dicas complementares repetem os textos ou o nome dos tipos dos elementos acionados.

4.6.4 Papéis, marcas e propriedades

Usuários que utilizam tecnologias assistivas necessitam que os elementos tenham as propriedades de acessibilidade corretamente associadas para que a tecnologia assistiva informe corretamente quais ações são esperadas para o elemento para que dessa forma o usuário consiga interagir adequadamente. Nome, papel e valor são exemplos de propriedades que devem ser informadas nos elementos.

Violação: Os elementos da tela, seus rótulos, papéis, valores, estados e mudanças de estado não são definidos e anunciados corretamente pelo leitor de tela.

Exemplo: Para cada item do menu da figura 22 é ditado o código fonte para aparecimento da imagem além do label abaixo de cada imagem do menu.

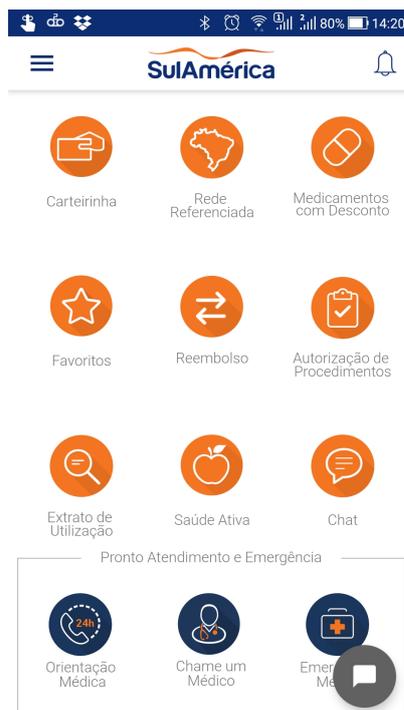


Figura 22 – Exemplo de violação de tarefas, marcas e propriedades

Avaliação manual: Para verificar manualmente as tarefas, marcas e propriedades é preciso ligar o leitor de tela, navegar pela aplicação e para cada elemento encontrado verificar se todas as propriedades de acessibilidade estão configuradas e anunciadas corretamente. É preciso ver se o tipo de elemento anunciado pelo leitor de tela é o mesmo acionado, e se o estado e o valor do item anunciado estão corretos.

Possibilidade de avaliação automática: Parcialmente automatizável.

Avaliação automática: No Android, os componentes padrões da plataforma possuem papéis associados mas não é possível definir marcações (*traits*) como no iOS. Os componentes criados pelos desenvolvedores devem descrever o papel do elemento utilizando a propriedade *contentDescription*.

É possível identificar automaticamente se componentes criados pelos desenvolvedores possuem a propriedade *contentDescription* definida, mas não é possível e a informação é suficiente para apresentar o papel e outras informações como o estado, por exemplo. Por isso esta recomendação pode ser avaliada automaticamente apenas de forma parcial. Para verificar automaticamente esta recomendação pode-se executar os seguintes passos.

- Encontre os elementos de interface que foram criados pelo usuário (não são padrão da plataforma, como um *Button*, por exemplo).
- Verifique se a propriedade *contentDescription* foi definida.
- Emita uma alerta de violação de recomendação se a propriedade *contentDescription* não foi definida.

Avaliação da possibilidade de automatização: A avaliação automática da recomendação foi implementada na ferramenta MATE mas não foram encontrados exemplos para validar a implementação utilizando resultados de avaliação manual.

4.6.5 Formatação visual

Formatação visual não deve ser usada para indicar significado, pois usuários com deficiência visual podem não conseguir perceber a diferença do formato visual, como tamanho, forma, localização, cor, e até mesmo formatação de texto como negrito ou itálico. É preciso utilizar meios alternativos e não visuais para comunicar significado além da formatação visual, como rótulos e descrições de conteúdo que possam ser acessados por leitores de tela. Em particular, alternativas para elementos predominantemente visuais,

como mapas e gráficos, por exemplo, precisam de uma representação não-visual bem detalhada.

Violação: Uso exclusivo da formatação visual da tela para transmitir informação ao usuário.

Exemplo: Na Figura 18 é utilizada apenas a cor vermelha para informar o usuário que o campo está com o preenchimento incorreto.

Verificação manual: para verificar manualmente a formatação visual navegue pela aplicação notando se para alguma tela a informação é transmitida para o usuário apenas com o uso da diferenciação na formatação visual. Em caso positivo, ligue o leitor de tela e navegue pela tela para verificar se a informação transmitida pela formatação visual é informada pelo leitor de tela.

Possibilidade de avaliação automática: Não Automatizável.

Justificativa: A avaliação automática da diretriz não é automatizável porque não é possível definir todos os elementos de design possíveis e suas variações para transmitir significado por meio de sua formatação visual. É possível identificar quando cores diferentes são utilizadas conforme recomendação descrita na Seção 4.2.2, mas isso já é coberto por outras diretrizes. Ainda que seja possível definir se diferentes tamanhos são utilizados por itens do mesmo tipo, não é possível definir se a variação de tamanho tem o objetivo de transmitir significado ou é só uma questão de decisão de desenho da interface.

4.7 Considerações finais

Ao todo, 29 recomendações foram analisadas. Dentre elas, 10 não podem ser avaliadas automaticamente, 9 podem ser parcialmente avaliadas, 9 podem ser totalmente avaliadas, e 1 não é aplicável à plataforma Android. Ainda que grande parte das recomendações possam ser avaliadas automaticamente apenas de forma parcial, isso indica que ou existe uma possível violação, ou uma violação clara, porém existem casos que não são cobertos.

O estudo aprofundado sobre a avaliação automática de acessibilidade é importante, ainda que grande parte das recomendações precisem ser verificadas manualmente. Acredita-se que a disponibilização de ferramentas que implementem a avaliação automática das diretrizes possam ser fatores fundamentais para aumentar a produtividade da ta-

refa de avaliação automática de acessibilidade. Em um cenário em que as equipes de desenvolvimento não possuem muito tempo disponível para avaliar suas aplicações, uma abordagem automática pode ao menos revelar uma porcentagem significativa de violações de acessibilidade presentes na aplicação.

5 Conclusão

O objetivo principal deste projeto de pesquisa foi definir quais recomendações de acessibilidade para dispositivos móveis com foco em deficiências visuais poderiam ser avaliadas automaticamente de tal forma que critérios de aceitação pudessem ser validados por testes gerados automaticamente. As recomendações, de acordo com as análises realizadas e os critérios definidos, foram rotuladas em: automatizável, quando a implementação da recomendação pode ser avaliada completamente de forma automática usando critérios de aceitação objetivos; parcialmente automatizável, quando os critérios de aceitação definidos possam indicar apenas indícios de que a recomendação não foi implementada, mas existindo a necessidade de julgamento humano; não automatizável, quando não é possível definir critérios objetivos que possam ser validados automaticamente; e ainda não aplicável quando a recomendação não se aplica a aplicações Android.

Acredita-se que os resultados dessa pesquisa serão muito importantes para guiar desenvolvedores na criação e evolução de ferramentas de apoio à avaliação de acessibilidade móvel, tema que tem sido alvo de grande atenção no mundo inteiro uma vez que o número de usuários de aplicações móveis tem crescido significativamente e legislações específicas de inclusão digital tem sido criadas, inclusive no Brasil. Acredita-se, também, que uma vez que as ferramentas de avaliação automática sejam robustas o suficientes, desenvolvedores as utilizarão no processo de desenvolvimento de aplicações e conseguirão detectar, antes do lançamento de seus produtos, problemas de acessibilidade que são possíveis de serem automatizados, restando aos especialistas e usuários a avaliação manual das demais propriedades.

Ainda que de forma limitada, o uso de ferramentas de automação é muito importante para apoiar a avaliação de acessibilidade móvel pois elas permitem que propriedades sejam analisadas em pouco tempo e com muita precisão. Em alguns casos, ela pode simplesmente indicar pontos que exigem maior atenção do avaliador.

5.1 Contribuição

A principal contribuição desse trabalho é um guia para desenvolvimento e evolução de ferramentas de avaliação automática de dispositivos móveis com foco em deficiências

visuais, contendo uma descrição sobre como avaliar automaticamente as propriedades de acessibilidade digital caso seja possível a avaliação automática.

5.2 *Limitações*

O objetivo principal desse trabalho é a construção do guia para evolução de ferramentas de análise acessibilidade com foco em deficiências visuais onde foram analisadas 29 das 54 diretrizes de acessibilidade definidas pela BBC. Portanto ainda restaram 25 diretrizes de acessibilidade que não são mencionadas no produto final desse trabalho. Além disso, o julgamento sobre a possibilidade de avaliação automática foi feita do ponto de vista de quem usa apenas informações coletadas por meio da API fornecida pelo Android, sem considerar a possibilidade de realizar análise estática para extrair informações do código da aplicação.

5.3 *Trabalhos futuros*

Apenas um subconjunto das recomendações de acessibilidade do guia da BBC foi analisado. Portanto, é preciso que as recomendações das demais seções do guia sejam analisadas. Ainda, é interessante que a mesma análise seja feita para as recomendações do WCAG. Como há uma forte sobreposição de diretrizes entre diversas normas, a análise aqui apresentada poderá ser utilizada para isso. Adicionalmente, é preciso validar o guia por meio de uma avaliação rigorosa feita pela comparação dos resultados obtidos pela avaliação automática implementada seguindo o guia com os resultados obtidos pela análise de especialistas ou usuários com deficiência em suas avaliações manuais ou mesmo com apoio de outras ferramentas.

5.4 *Publicações relacionadas a esse trabalho*

As seguintes publicações resultaram deste trabalho:

- Camila P. F. Silva e Marcelo M. Eler. Um estudo sobre a avaliação automática de diretrizes de acessibilidade para dispositivos móveis. Em: XI Workshop de Teses e Dissertações em Sistemas de Informação (WTDSI 2018).

- Camila P. F. Silva, Marcelo M. Eler, and Gordon Fraser. 2018. A survey on the tool support for the automatic evaluation of mobile accessibility. In Proceedings of the 8th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2018). Association for Computing Machinery, New York, NY, USA, 286–293. DOI:<https://doi.org/10.1145/3218585.3218673>

Referências¹

- AL-SUBAIHIN, A. A.; AL-KHALIFA, A. S.; AL-KHALIFA, H. S. Accessibility of mobile web apps by screen readers of touch-based mobile phones. In: SPRINGER. *International Conference on Mobile Web and Information Systems*. [S.l.], 2013. p. 35–43. Citado na página 27.
- ARRUE, M.; VIGO, M.; ABASCAL, J. *Automatic evaluation of mobile web accessibility*. [S.l.: s.n.], 2007. v. 4397 LNCS. 244-260 p. (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v. 4397 LNCS). Citado na página 27.
- BALAJI, V.; KUPPUSAMY, K. S. Accessibility analysis of e-governance oriented mobile applications. In: *2016 International Conference on Accessibility to Digital World, ICADW 2016 - Proceedings*. [s.n.], 2017. p. 141–144. Disponível em: <www.scopus.com>. Citado na página 27.
- BANDEIRA, R.; LOPES, R.; CARRIÇO, L. *Results from multi-dimensional accessibility assessment*. [S.l.: s.n.], 2011. v. 6765 LNCS. 187-196 p. (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), PART 1). Citado na página 27.
- BBC. *BBC Mobile Accessibility Guidelines*. 2018. Disponível em: <<http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile>>. Citado 2 vezes nas páginas 12 e 16.
- BILLI, M. et al. A unified methodology for the evaluation of accessibility and usability of mobile applications. *Universal Access in the Information Society*, Springer, v. 9, n. 4, p. 337–356, 2010. Citado na página 22.
- 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. Citado na página 27.
- BRAJNIK, G.; YESILADA, Y.; HARPER, S. The expertise effect on web accessibility evaluation methods. *Human-Computer Interaction*, Taylor & Francis, v. 26, n. 3, p. 246–283, 2011. Citado na página 22.
- CARRIÇO, L.; LOPES, R.; BANDEIRA, R. Crosschecking the mobile web for people with visual impairments. In: *W4A 2011 - International Cross-Disciplinary Conference on Web Accessibility*. [S.l.: s.n.], 2011. Citado na página 27.
- CARVALHO, L. P. et al. Accessible smart cities?: Inspecting the accessibility of brazilian municipalities' mobile applications. In: *Proc. of the 15th Brazilian Symp. on Human Factors in Comp. Syst.* New York, NY, USA: ACM, 2016. (IHC '16), p. 17:1–17:10. ISBN 978-1-4503-5235-2. Disponível em: <<http://doi.acm.org/10.1145/3033701.3033718>>. Citado na página 12.
- CONSORTIUM, W. W. W. *WCAG 2.0 Techniques that Apply to Mobile*. 2015. <<https://www.w3.org/WAI/GL/mobile-a11y-tf/MobileTechniques/>>. Citado na página 16.

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

- DIAS, C. *Usabilidade na web: criando portais mais acessíveis*. [S.l.]: Alta Books, 2007. Citado na página 15.
- ELER, M. M.; ORLANDIN, L.; OLIVEIRA, A. D. A. Do android app users care about accessibility?: An analysis of user reviews on the google play store. In: *Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019. (IHC '19), p. 23:1–23:11. ISBN 978-1-4503-6971-8. Disponível em: <http://doi.acm.org/10.1145/3357155.3358477>. Citado na página 23.
- ELER, M. M. et al. Automated accessibility testing of mobile apps. In: IEEE. *Proceedings 11th IEEE Conference on Software Testing, Validation and Verification (to appear)*. [S.l.], 2018. p. 1–11. Citado 4 vezes nas páginas 12, 28, 31 e 35.
- EMAG. *Modelo De Acessibilidade em Governo Eletrônico - e-MAG*. Acesso em: 23/04/2019. <http://emag.governoeletronico.gov.br>. Citado 4 vezes nas páginas 37, 38, 39 e 46.
- FORAPP. *forApp Mobile Accessibility Inspection Solution*. 2018. <http://www.forapp.org>. Citado na página 28.
- FREIRE, A. P.; RUSSO, C. M.; FORTES, R. P. M. A survey on the accessibility awareness of people involved in web development projects in brazil. In: *Proceedings of the 2008 International Cross-disciplinary Conference on Web Accessibility (W4A)*. New York, NY, USA: ACM, 2008. (W4A '08), p. 87–96. ISBN 978-1-60558-153-8. Disponível em: <http://doi.acm.org/10.1145/1368044.1368064>. Citado na página 12.
- GEMOU, M. et al. Validation of toolkits for developing third-generation android accessible mobile applications. *Universal Access in the Information Society*, v. 15, n. 1, p. 101–127, 2016. Citado na página 27.
- GOOGLE. *Android Lint*. 2018. <https://developer.android.com/studio/write/lint.html>. Citado na página 28.
- GOOGLE. *EarlGrey: iOS UI Automation Test Framework*. 2018. <https://github.com/google/EarlGrey>. Citado na página 29.
- GOOGLE. *Espresso*. 2018. <https://google.github.io/android-testing-support-library/docs/espresso/index.html>. Citado na página 28.
- GOOGLE. *Robolectric*. 2018. <http://robolectric.org/>. Citado na página 28.
- HAO, S. et al. Puma: programmable ui-automation for large-scale dynamic analysis of mobile apps. In: ACM. *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. [S.l.], 2014. p. 204–217. Citado na página 28.
- HURLBERT, A.; WOLF, K. Color contrast: a contributory mechanism to color constancy. *Progress in brain research*, Elsevier, v. 144, p. 145–160, 2004. Citado na página 37.
- IBGE. *Vamos conhecer o Brasil: características da população*. Acesso em: 22/02/2018. <https://7a12.ibge.gov.br/vamos-conhecer-o-brasil/nosso-povo/pessoas-com-deficiencia.html>. Citado na página 11.
- IBM. *IBM AbilityLab Mobile Accessibility Checker*. 2018. <https://www-03.ibm.com/able/mobile-accessibility-checker.html>. Citado na página 29.

KAUR, A.; DANI, D. Mobile web accessibility readiness of government websites using diagnostic tools: an exploratory study. *Electronic Government, an International Journal*, Inderscience Publishers (IEL), v. 13, n. 1, p. 1–30, 2017. Citado na página 27.

KIF. *KIF: Keep It Functional - An iOS Functional Testing Framework*. 2018. [⟨https://github.com/kif-framework/KIF⟩](https://github.com/kif-framework/KIF). Citado na página 29.

KUSHNIRUK, A. W.; PATEL, V. L. Cognitive and usability engineering methods for the evaluation of clinical information systems. *Journal of biomedical informatics*, Elsevier, v. 37, n. 1, p. 56–76, 2004. Citado na página 49.

LEE, S.; KO, S. H.; FOX, G. C. Adapting content for mobile devices in heterogeneous collaboration environments. In: CITESEER. *International Conference on Wireless Networks*. [S.l.], 2003. p. 211–217. Citado na página 45.

LEE, Y.; LEE, J. A checklist for assessing blind users' usability of educational smartphone applications. *Universal Access in the Information Society*, p. 1–18, 2017. Article in Press. Citado na página 27.

MARTÍN, Y.-S.; YELMO, J. C. Guidance for the development of accessibility evaluation tools following the unified software development process. *Procedia Computer Science*, Elsevier, v. 27, p. 302–311, 2014. Citado na página 22.

MAZZONI, A. A. et al. Aspectos que interferem na construção da acessibilidade em bibliotecas universitárias. *Ciência da Informação*, SciELO Brasil, 2001. Citado na página 15.

MELO, A. M.; BARANAUSKAS, M. C. C.; BONILHA, F. F. G. Avaliação de acessibilidade na web com a participação do usuário: um estudo de caso. *Simpósio Sobre Fatores Humanos Em Sistemas Computacionais*, v. 6, p. 165–168, 2004. Citado na página 21.

MI, N. et al. A heuristic checklist for an accessible smartphone interface design. *Universal Access in the Information Society*, v. 13, n. 4, p. 351–365, 2014. Citado na página 27.

MICROSOFT. *AccChecker*. 2018. [⟨https://msdn.microsoft.com/pt-br/library/windows/desktop/hh920985\(v=vs.85\).aspx⟩](https://msdn.microsoft.com/pt-br/library/windows/desktop/hh920985(v=vs.85).aspx). Citado na página 30.

MICROSOFT. *AccScope*. 2018. [⟨https://msdn.microsoft.com/en-us/library/windows/desktop/dn433239\(v=vs.85\).aspx⟩](https://msdn.microsoft.com/en-us/library/windows/desktop/dn433239(v=vs.85).aspx). Citado na página 30.

MICROSOFT. *Inspect*. 2018. [⟨https://msdn.microsoft.com/library/windows/desktop/Dd318521⟩](https://msdn.microsoft.com/library/windows/desktop/Dd318521). Citado na página 30.

MWAC. *Mobile Web Accessibility Checker*. 2018. [⟨http://www.userlight.com/apps/mwac/⟩](http://www.userlight.com/apps/mwac/). Citado na página 29.

NANDAKUMAR, V.; EKAMBARAM, V.; SHARMA, V. *Appstrument - A unified app instrumentation and automated playback framework for testing mobile applications*. [S.l.: s.n.], 2014. v. 131. 474-486 p. (Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, v. 131). Citado na página 27.

NAYEBI, F.; DESHARNAIS, J.-M.; ABRAN, A. The state of the art of mobile application usability evaluation. In: IEEE. *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*. [S.l.], 2012. p. 1–4. Citado na página 11.

OLIVEIRA, A. D. A.; ELER, M. M. Strategies and challenges on the accessibility and interoperability of e-government web portals: A case study on brazilian federal universities. In: IEEE. *Computer Software and Applications Conference (COMPSAC), 2017 IEEE 41st Annual*. [S.l.], 2017. v. 1, p. 737–742. Citado na página 12.

OLIVEIRA, A. D. d.; SOUZA, E. M. d.; ELER, M. M. Accessibility model in electronic government: Evaluation of brazilian web portals. In: *Proceedings of the Brazilian Symposium on Information Systems (SBSI)*. [S.l.]: SBSI, 2017. p. 332–339. Citado na página 12.

PARTARAKIS, N. et al. *A unified environment for accessing a suite of accessibility evaluation facilities*. [s.n.], 2011. v. 6765 LNCS. 267-275 p. (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), PART 1). Disponível em: <www.scopus.com>. Citado na página 27.

PASSERINO, L. M.; MONTARDO, S. P. Inclusão social via acessibilidade digital: proposta de inclusão digital para pessoas com necessidades especiais. In: *E-Compós*. [S.l.: s.n.], 2007. v. 8. Citado na página 15.

PATIL, N.; BHOLE, D.; SHETE, P. Enhanced ui automator viewer with improved android accessibility evaluation features. In: *International Conference on Automatic Control and Dynamic Optimization Techniques, ICACDOT 2016*. [S.l.: s.n.], 2017. p. 977–983. Citado na página 27.

RODRIGUES, A. et al. Getting smartphones to talkback: Understanding the smartphone adoption process of blind users. In: *ASSETS 2015 - Proceedings of the 17th International ACM SIGACCESS Conference on Computers and Accessibility*. [S.l.: s.n.], 2015. p. 23–32. Citado na página 27.

RUBIN, J.; CHISNELL, D. *Handbook of usability testing: howto plan, design, and conduct effective tests*. [S.l.]: John Wiley & Sons, 2008. Citado na página 12.

SANTAROSA, L. M. C. Inclusão digital: espaço possível para pessoas com necessidades educativas especiais. *Revista Educação Especial*, p. 13–30, 2002. Citado na página 15.

SERRA, L. C. et al. Accessibility evaluation of e-government mobile applications in brazil. *Procedia Computer Science*, v. 67, p. 348 – 357, 2015. ISSN 1877-0509. Proceedings of the 6th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050915031257>>. Citado na página 12.

SHANLEY, C. Cracking accessibility on mobile devices: The definitive field guide to accessibility and digital inclusion for business managers and project teams. In: _____. [S.l.]: RS Books, 2016. p. 258. Citado na página 16.

SILVA, C. P. F.; ELER, M. M.; FRASER, G. A survey on the tool support for the automatic evaluation of mobile accessibility. In: *Software Development and Technologies*

for Enhancing Accessibility and Fighting Info-exclusion (DSAI). [S.l.: s.n.], 2018. p. 1–8. Citado 7 vezes nas páginas 12, 13, 39, 43, 44, 54 e 67.

TANGARIFE, T.; aO, C. M. Estudo comparativo utilizando uma ferramenta de avaliação de acessibilidade para web. In: *Proceedings of the 2005 Latin American Conference on Human-computer Interaction*. New York, NY, USA: ACM, 2005. (CLIHIC '05), p. 313–318. ISBN 1-59593-224-0. Disponível em: <http://doi.acm.org/10.1145/1111360.1111394>. Citado na página 22.

VELASCO, C. A. et al. A web compliance engineering framework to support the development of accessible rich internet applications. In: *W4A'08: Proceedings of the 2008 International Cross-Disciplinary Conference on Web Accessibility, W4A*. [S.l.: s.n.], 2008. p. 45–49. Citado na página 27.

VELLEMAN, E. et al. *D-WAB4 Unified Web Evaluation Methodology*. [S.l.]: Web Accessibility Benchmarking Cluster. Retrieved 14th February, 2010. Citado na página 23.

VIGO, M. et al. Evaluating web accessibility for specific mobile devices. In: *ACM. Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)*. [S.l.], 2008. p. 65–72. Citado na página 27.

VIGO, M.; BROWN, J.; CONWAY, V. Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests. In: *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*. New York, NY, USA: ACM, 2013. (W4A '13), p. 1:1–1:10. ISBN 978-1-4503-1844-0. Disponível em: <http://doi.acm.org/10.1145/2461121.2461124>. Citado 3 vezes nas páginas 12, 13 e 23.

W3C. *Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile*. 2015. Disponível em: <https://www.w3.org/TR/mobile-accessibility-mapping/>. Citado na página 16.

World Bank. *Main Report - World Report on Disability*. 2011. Disponível em: <http://documents.worldbank.org/curated/en/665131468331271288/Main-report>. Citado na página 11.

YESILADA, Y.; BRAJNIK, G.; HARPER, S. Barriers common to mobile and disabled web users. *Interacting with Computers*, v. 23, n. 5, p. 525–542, 2011. Citado na página 27.