Acessibilidade no desenvolvimento de aplicativos móveis: uma revisão sistemática sobre problemas recorrentes, métodos de verificação e ferramentas

Séphora C. C. Marques[1], **Matheus C. Viana**[1], **Dárlinton B. F. Carvalho**[1]

[1]Departamento de Ciência da Computação
Universidade Federal de São João del-Rei (UFSJ)
São João del-Rei – MG – Brazil

marques.scar@gmail.com, {matheuscviana,darlinton}@ufsj.edu.br

Abstract. Introduction: *With the popularization of smartphones, there has been an increase in the supply of applications for mobile devices. However, although they can assist in digital inclusion and facilitate daily activities, such applications can represent a barrier for people with disabilities if they have not been developed considering accessibility criteria.* ***Objective:*** *This research aimed to systematize knowledge about mobile application accessibility.* ***Methodology:*** *A systematic review of the literature published between 2016 and 2025 was carried out.* ***Results:*** *The most recurrent problems were identified, as well as the most mentioned tools and methods. Finally, limitations and advantages identified in the adopted approaches were highlighted.*
Keywords *Accessibility evaluation, Mobile development, Mobile accessibility, Accessibility assessment.*

Resumo. Introdução: *Com a popularização dos smartphones, houve um aumento na oferta de aplicativos para dispositivos móveis. Entretanto, ainda que possam auxiliar na inclusão digital e facilitar atividades do cotidiano, eles podem representar uma barreira para pessoas com deficiência, caso não tenham sido desenvolvidos considerando critérios de acessibilidade.* ***Objetivo:*** *Esta pesquisa buscou sistematizar os conhecimentos sobre acessibilidade de aplicativos.* ***Metodologia:*** *Foi realizada uma revisão sistemática da literatura publicada entre 2016 e 2025.* ***Resultados:*** *Foram identificados os problemas mais recorrentes, bem como ferramentas e métodos mais mencionados. Por fim, foram destacadas limitações e vantagens nas abordagens adotadas.*
Palavras-Chave *Avaliação de acessibilidade, Desenvolvimento móvel, Acessibilidade móvel, Análise de acessibilidade.*

1. Introdução

O uso de smartphones se popularizou a ponto de, no final do ano de 2023, cerca de 70% da população mundial utilizar um aparelho desse tipo [Laricchia 2024]. Entretanto, considerando que esses dispositivos geralmente têm tela sensível ao toque, um aplicativo que possibilita acessos e facilidades para alguns usuários pode, ao mesmo tempo, representar uma barreira para pessoas com deficiência (PCD). Tendo em vista as estimativas da Organização Mundial de Saúde (OMS) de que aproximadamente 16% da população mundial [World Health Organization 2023] têm algum tipo de deficiência, é necessário pensar em maneiras de tornar a utilização do smartphone mais acessível.

Para projetar aplicações acessíveis, é necessária a adoção de técnicas específicas. Nesse sentido, existem diretrizes que podem orientar a implementação de critérios de acessibilidade. Uma delas é a *Web Content Accessibility Guidelines* (WCAG), criada pelo *World Wide Web Consortium* (W3C)[1], em 1999, com o objetivo de tornar o conteúdo da Web acessível [W3C Web Accessibility Initiative 1999]. Ainda que sua criação tenha sido em um período anterior ao lançamento dos smartphones, a WCAG vem evoluindo ao longo dos anos e traz diretrizes direcionadas para conteúdos em dispositivos móveis [W3C Web Accessibility Initiative (WAI) 2025].

Deve-se observar que é possível adotar diferentes métodos para avaliar a acessibilidade, como ferramentas automatizadas, inspeções manuais e avaliações de usuários. Particularmente, de acordo com [Mateus et al. 2021], a utilização de ferramentas automatizadas traz como vantagem a agilidade na identificação de problemas em fases iniciais do processo de desenvolvimento. Cabe ressaltar que, apesar das diretrizes e métodos existentes, a inacessibilidade nas aplicações ainda é um problema. Estudos anteriores mostram que parte considerável dos aplicativos na Google Play Store apresenta falhas de acessibilidade [Chen et al. 2022, Alshayban et al. 2020, Acosta-Vargas et al. 2019, Acosta-Vargas et al. 2021a].

Pesquisas e estudos sobre a acessibilidade mostram-se relevantes ao se considerar que um dos Grandes Desafios de Pesquisa de IHC no Brasil, identificado pelo II GranDIHC-BR, está relacionado a GC3: Pluralidade e Decolonialidade. Esse desafio enfatiza a necessidade de revisitar as abordagens adotadas no desenvolvimento de sistemas interativos para garantir que os diferentes grupos sociais - incluindo pessoas com deficiência - sejam consideradas ao longo do processo [Pereira et al. 2024]. Assim, visando contribuir para uma compreensão mais abrangente do tema, este artigo apresenta uma revisão sistemática baseada em estudos realizados entre 2016 e 2025. O recorte temporal foi definido levando em consideração que, na primeira metade da década de 2010, ainda existiam sistemas operacionais que foram descontinuados posteriormente e, por isso, não seriam de interesse para este trabalho. A pesquisa foi conduzida no intuito de identificar os problemas mais recorrentes, bem como as ferramentas e métodos adotados para verificar a acessibilidade nos aplicativos para dispositivos móveis.

2. Revisão Sistemática sobre acessibilidade no desenvolvimento de aplicativos móveis

Para realizar a revisão sistemática de literatura (RSL) foram seguidas as diretrizes de relato PRISMA [Prisma Statements 2020]. As seções seguintes apresentam o protocolo, a execução, os resultados, as discussões e as ameaças à validade.

2.1. Protocolo

O objetivo da revisão sistemática foi **pesquisar estudos sobre a verificação da acessibilidade no desenvolvimento de aplicativos para dispositivos móveis, em especial, que tratassem de problemas, ferramentas e abordagens utilizadas**. Dessa forma, a revisão foi guiada pelas seguintes questões de pesquisa:

1. Quais são os problemas de acessibilidade mais recorrentes na literatura?

[1]O W3C foi fundado em 1994 por Tim Berners-Lee - criador da World Wide Web - para desenvolver padrões web abertos, visando à interoperabilidade global da web [W3C Brasil 2025].

2. Quais ferramentas auxiliam a verificar critérios de acessibilidade no desenvolvimento de aplicativos móveis?
3. Quais abordagens são adotadas pelas ferramentas analisadas?
4. De quais formas as ferramentas podem ser utilizadas?
5. Para quais sistemas operacionais essas ferramentas estão disponíveis?

O objetivo da Questão 1 foi mapear as falhas que apareceram com maior recorrência nos estudos. Para isso, adotou-se como critério de classificação o número de ocorrências de cada falha nos aplicativos testados. Assim, a falha com maior número de ocorrências foi classificada em primeiro lugar, e as demais foram ordenadas de acordo com a sua frequência. A Questão 2 teve como propósito levantar e analisar as ferramentas que existem para verificação de acessibilidade. O critério adotado para essa análise foi contabilizar o número de vezes que cada ferramenta foi mencionada nos artigos revisados, com o intuito de verificar as mais recorrentes na literatura. Com as Questões 3 e 4, buscou-se categorizar e analisar as abordagens adotadas pelas ferramentas, com o objetivo de identificar possíveis áreas de melhoria para trabalhos futuros. Por fim, a Questão 5 objetivou identificar os sistemas operacionais para os quais as ferramentas são destinadas.

Após a definição das perguntas de pesquisa, procedeu-se com a seleção dos estudos primários e a formulação da estratégia de busca, as quais foram estabelecidas com base nos seguintes critérios:

- **Critérios para seleção das fontes:** Foram selecionadas as principais conferências e periódicos em que são publicados estudos da área de Engenharia de Software.
- **Métodos de Pesquisa:** Foi realizada uma busca automatizada, na qual uma *string* de busca foi criada e executada nas fontes selecionadas.
- **Palavras-chave:** Accessibility evaluation, mobile development, static analysis.
- **String de busca:** (‘accessibility evaluation’ OR ‘accessibility assessment’) AND (ios OR android) AND (tool OR plugin OR lint OR ‘static analysis’ OR ‘continuous integration’)
- **Fontes Selecionadas:** ACM, IEEE, Science Direct e Springer.

A partir disso, foram definidos critérios de inclusão e exclusão para a seleção dos estudos obtidos a partir da busca inicial, sendo eles:

Critérios de inclusão

- Critério 1: Estudos publicados nos últimos 10 anos;
- Critério 2: Estudos completos em periódicos, eventos ou capítulos de livros.

Critérios de exclusão

- Critério 1: Estudos que não estejam em português ou inglês;
- Critério 2: Estudos que não tratem sobre etapas ou ferramentas do processo de desenvolvimento mobile;
- Critério 3: Estudos que não estejam relacionados à acessibilidade para aplicações móveis;
- Critério 4: Estudos relacionados à acessibilidade no desenvolvimento de jogos para dispositivos móveis.

Considerando que uma parte significativa das interfaces gráficas de jogos são desenvolvidas utilizando motores de jogo, como Unreal [Epic Games 2025] e Unity [Unity Technologies 2025], e não utilizam componentes nativos, foi definido que os estudos relacionados ao tema seriam excluídos da revisão (Critério 4).

2.2. Execução

Nesta seção, são apresentados os detalhes da execução da RSL. A busca foi realizada em janeiro de 2025 nos repositórios mencionados e os resultados foram compilados em uma planilha para a realização do processo de triagem e seleção de estudos.

Na Figura 1, é apresentado o diagrama que detalha a abordagem adotada para a seleção dos estudos provenientes das bases de dados. O processo teve início com a fase de identificação, na qual foi realizada a pesquisa utilizando a *string* de busca, resultando em 245 estudos, distribuídos da seguinte forma: 132 na ACM, 4 na IEEE, 37 na Science Direct e 72 na Springer. Em seguida, aplicou-se o critério de seleção, considerando apenas os estudos publicados nos últimos 10 anos. Como resultado, 36 trabalhos foram excluídos, totalizando 209 registros para a fase de triagem.

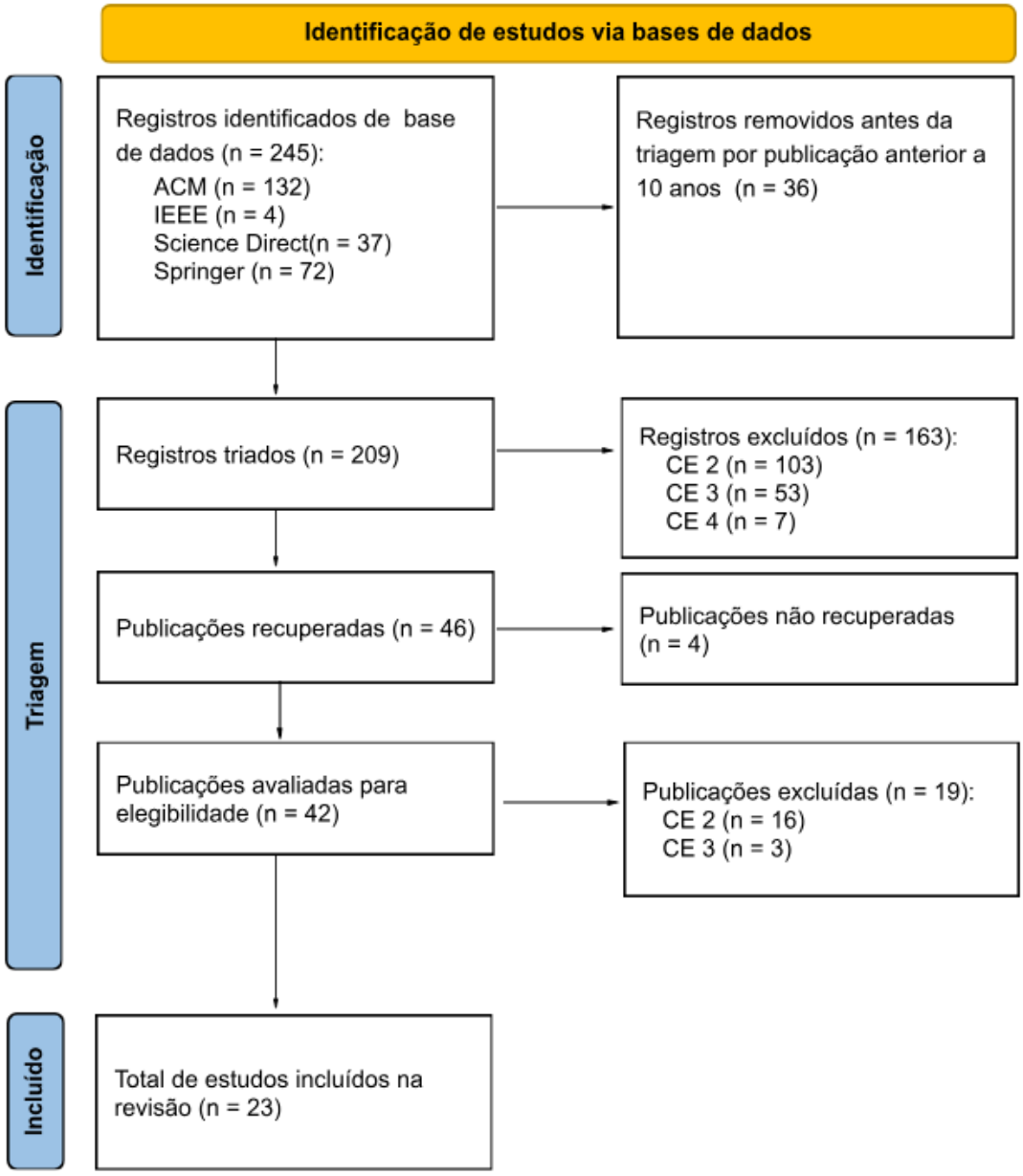

Figura 1. Diagrama de fluxo da RSL, de acordo com o modelo PRISMA 2020.

Na triagem, foi realizada a leitura do título e resumo dos 209 estudos. Com base nos critérios de exclusão, foram removidos 161 estudos, sendo 103 pelo Critério 2, 53 pelo Critério 3 e 7 pelo Critério 4, resultando em 46 publicações. Contudo, 4 delas não foram obtidas na íntegra e foram definidas como não recuperadas. Assim, 42 publicações foram selecionadas para análise de elegibilidade, que ocorreu com a leitura completa do texto. Nessa etapa, foram removidos 19 estudos, sendo 16 pelo Critério 2 e 3 pelo Critério 3. Ao final, foram incluídos na revisão 23 estudos, dispostos no Apêndice A deste artigo.

2.3. Resultado

Nesta seção, são apresentados os resultados obtidos a partir dos textos selecionados na RSL. Os estudos foram organizados de acordo com o ano de publicação, conforme a Figura 2. A maior parte das publicações ocorreu entre 2018 e 2024. Observa-se que os números de publicações em 2023 e 2024 foram inferiores aos de 2021 e 2022, indicando oscilações nos estudos sobre o tema e ausência de tendência de crescimento evidente.

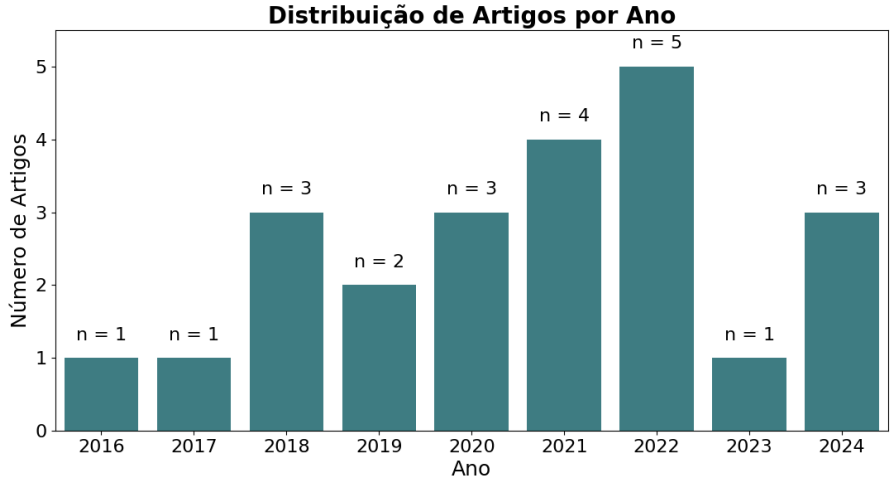

Figura 2. Número de estudos publicados por ano.

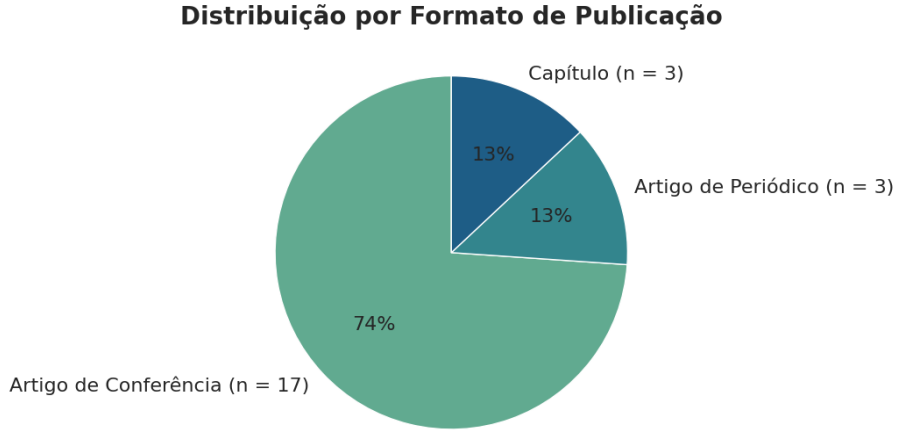

Figura 3. Número de estudos publicados por formato de publicação.

A Figura 3 apresenta a distribuição dos estudos por formato de publicação. Foram identificados os formatos de Artigo de Conferência, Artigo de Periódico e Capítulo de

Livro. Os Artigos de Conferência representaram 74% do total de estudos selecionados. Tal percentual sugere que a temática está em fase de exploração pela comunidade científica, com proposições de novas discussões e soluções. Com a evolução dinâmica dos smartphones e seus sistemas operacionais nas últimas duas décadas, é possível que muitos temas ainda não estejam consolidados na literatura. Após a análise dos estudos, obtiveram-se as seguintes respostas para as perguntas que guiaram a RSL:

Pergunta 1: Quais problemas de acessibilidade são mais recorrentes na literatura?

Para responder a essa questão, foram selecionados os artigos que apresentavam dados sobre a avaliação de falhas de acessibilidade em aplicativos utilizando as ferramentas disponíveis. Alguns desses textos, apesar de apresentarem dados, não foram incluídos nesta resposta por tratarem de análises específicas, como o AdMole [He et al. 2024], que verifica a acessibilidade em propagandas exibidas no aplicativo.

No estudo de [Chen et al. 2022], que avaliou 2.270 aplicativos, os problemas mais recorrentes foram rótulo do item, descrições do item, alvo de toque, contraste do texto e o contraste da imagem. Em [Acosta-Vargas et al. 2019], os autores verificaram que os problemas mais frequentes foram respectivamente contraste do texto, alvo de toque, rótulo do item, descrições dos itens, contraste da imagem e itens clicáveis. Já em estudo posterior [Acosta-Vargas et al. 2021a], os pesquisadores encontraram, na análise da amostra, que as violações mais frequentes foram alvo de toque, rótulo de item, contraste de texto e contraste da imagem. Já o artigo [Alshayban et al. 2020] apresenta a avaliação de 1135 aplicativos e as principais falhas foram contraste de texto, alvo de toque, contraste da imagem e texto falado[2].

Alguns estudos citaram problemas não abordados no parágrafo anterior. No entanto, como o objetivo era identificar as principais falhas de acessibilidade, adotou-se a estratégia de compilar as menções ao tipo de violação e, em seguida, somar as ocorrências de cada uma nos artigos. Problemas mencionados em apenas um artigo foram descartados. A Tabela 1 apresenta os principais problemas com a soma das menções.

Tabela 1. Principais problemas de acessibilidade encontrados.

Problema	**Menções**
Alvo de toque	4
Contraste de imagem	4
Contraste de texto	4
Rótulo do item	4
Descrição do item	2
Itens clicáveis	2
Total	**20**

A Tabela 2 apresenta as definições correspondentes aos problemas encontrados, tendo sido utilizados como referência os estudos [Chen et al. 2022] e [Ross et al. 2017], baseados nas diretrizes disponibilizadas pelo Google.

[2]Refere-se ao conteúdo lido por leitores de tela, tendo sido considerado equivalente ao rótulo do item nesta pesquisa.

Tabela 2. Descrição dos problemas encontrados.

Problema	Descrição
Alvo de toque	Área de toque inferior a 48dp X 48dp em componentes que podem ser clicáveis
Contraste de imagem	Imagens que tenham relação de contraste inferior a 3.0 entre as cores do primeiro plano e do fundo
Contraste de texto	Contraste inferior a 3:1 entre as cores do texto e do fundo
Rótulo do item	Item focável pelo leitor que não possui descrição a ser lida
Descrição do item	Mais de um item possui o mesmo texto lido pelo leitor de tela
Itens clicáveis	Sobreposição de itens, fazendo com que mais de um item compartilhe o mesmo local na tela

Considerando que os artigos analisaram amostras de tamanhos diferentes, optou-se por ponderá-los com base no percentual entre a amostra avaliada e o total de aplicativos avaliados (Tabela 3), para, então, calcular a média ponderada (Tabela 4):

Tabela 3. Peso dos artigos para análise dos problemas.

Artigo	Aplicativos avaliados	Peso
Chen *et al.*, 2022	2270	0,6635
Alshayban *et al.*, 2020	1135	0,3318
Acosta-Vargas *et al.*, 2019	10	0,0029
Acosta-Vargas *et al.*, 2021	6	0,0018
Total	**3421**	1,0000

Tabela 4. Nível dos problemas nos artigos.

Artigo	Alvo de toque	Contraste de texto	Rótulo do item	Contraste de imagem	Descrição do item	Itens clicáveis
Chen *et al.*, 2022	6	5	4	3	2	1
Alshayban *et al.*, 2020	5	6	3	4	0	0
Acosta-Vargas *et al.*, 2019	5	6	4	2	3	1
Acosta-Vargas *et al.*, 2021	6	4	5	3	0	0
Média ponderada	**5,67**	**5,33**	**3,67**	**3,33**	**1,34**	**0,67**

Para o cálculo dos valores da Tabela 4, foram atribuídos valores numéricos de 0 a 6 aos seis problemas mais mencionados, sendo 6 correspondente ao maior número de

ocorrências nos aplicativos analisados no artigo e 0 ausência da menção do problema no artigo. Em seguida, a média ponderada de cada problema foi calculada, utilizando os pesos da Tabela 3.

Pergunta 2: Quais ferramentas auxiliam a verificar critérios de acessibilidade no desenvolvimento de aplicativos móveis?

As ferramentas foram organizadas em tabelas, sendo desconsideradas aquelas relacionadas a mobile web, assim como as ferramentas para o sistema operacional Windows Phone, descontinuado pela Microsoft [3]. Foram contabilizadas 41 ferramentas, divididas em dois grupos: as não específicas para validar acessibilidade [4] e as específicas para tal fim.

As 13 ferramentas não específicas identificadas estão listadas na Tabela 5, que apresenta o nome da ferramenta, o sistema operacional para o qual é destinada, o desenvolvedor e as menções recebidas por cada uma. Ressalta-se que, na coluna 'Desenvolvedor', o valor 'Autoral' foi atribuído às descritas em estudos próprios e o valor 'Colaborativo' às que estão abertas para desenvolvimento pela comunidade. Na coluna 'Menções', os números correspondem a quantos artigos selecionados na RSL mencionaram a ferramenta.

Tabela 5. Ferramentas não específicas para acessibilidade.

Ferramenta	Sistema Operacional	Desenvolvedor	Menções
Android Lint	Android	Google	11
Appium	Android, iOS	Colaborativo	2
EarlGrey	iOS	Google	2
Espresso	Android	Google	10
Google Monkey	Android	Google	2
Layout inspector	Android	Google	1
OwlEye	Android, iOS	Autoral	1
Puma	Android	Autoral	4
Roboelectric	Android	Colaborativo	8
Sapienz	Android	Autoral	1
Stoat	Android	Autoral	1
UI Automator	Android	Google	4
XCTest	iOS	Apple	1
Total de ferramentas			**13**

Foram identificadas 28 ferramentas específicas para verificar a acessibilidade em aplicativos móveis, listadas na Tabela 6. Para a construção desta tabela, foram adotados os mesmos critérios de divisão das colunas utilizadas na tabela anterior:

[3] O suporte oficial à última versão do sistema operacional Windows 10 Mobile foi encerrado no final do ano de 2019, representando o fim do ciclo de atualizações e manutenção por parte da Microsoft.

[4] Ferramentas voltadas à verificação de layout, análise de estilo de código, dentre outras funcionalidades.

Tabela 6. Ferramentas relacionadas à acessibilidade.

Ferramentas	Sistema	Desenvolvedor	Menções
A11yPuppetry	Android	Autoral	2
AATK	Android	Autoral	1
AccessibiLint	Android	Autoral	1
Accessibility Testing Framework	Android	Google	7
Acessibility Scanner	Android	Google	19
AdMole	Android	Autoral	1
Alotaibi Detection of TalkBack Failures	Android	Autoral	1
Alotaibi Repair of Size-Based Issues	Android	Autoral	1
Alshayban et al. Crawler	Android	Autoral	6
Apple's Accessibility Inspector/Scanner	iOS	Apple	8
axe	Android, iOS	Deque	1
COALA	Android	Autoral	2
Enhanced UI Automator Viewer	Android	Autoral	3
Evinced	Android, iOS	Evinced	1
Fok et al. Crawler	Android	Autoral	1
forApp	Android, iOS	SCE Korea	1
Groundhog	Android	Autoral	3
IBM's MAC	Android	IBM	8
Kashif et al. Evaluation Plugin	Não se aplica	Autoral	2
KIF	iOS	Colaborativo	1
LabelDroid	Android	Autoral	5
Latte	Android	Autoral	5
MATE	Android	Autoral	9
Moura et al. API for Testing	Android	Autoral	2
Park et al. Tool	Android	Autoral	3
Swearngin et al. system	iOS	Autoral	1
WorldSpace Attest Mobile	Android, iOS	Deque	1
Xbot	Android	Autoral	3
Total			**28**

Durante o levantamento das ferramentas, observou-se a menção a quatro métodos específicos relacionados à verificação ou à implementação da acessibilidade para aplicativos móveis. Krainz *et al.* propuseram uma abordagem de desenvolvimento orientado por modelos (*model-driven development*) para aumentar a acessibilidade dos aplicativos [Krainz et al. 2018]. O estudo de Ross *et al.* apresentou um modelo baseado na epidemiologia para avaliar a acessibilidade em um nível sistêmico, observando não apenas um único aplicativo, mas o ecossistema em que ele está inserido [Ross et al. 2017]. O estudo [Zhang et al. 2021] propôs inferir metadados de acessibilidade a partir dos pixels das interfaces visuais. Em [Zhang et al. 2018], os autores desenvolveram uma abordagem para anotação de elementos de interface de aplicativos móveis com técnicas para reparo de acessibilidade em tempo de execução. A Tabela 7 apresenta os métodos, acompanhados da soma das menções.

Tabela 7. Métodos relacionados à acessibilidade de aplicativos.

Métodos	**Menções**
Krainz *et al.* Model-driven Development	2
Ross *et al.* epidemiology-inspired framework	3
Zhang *et al.* accessibility metadata from pixels approach	3
Zhang *et al.* Annotation for Accessibility Repair	3
Total de métodos	4

Pergunta 3: Quais abordagens são adotadas pelas ferramentas analisadas?

Foram identificadas, nas ferramentas, as abordagens de análise estática e de análise dinâmica. A Tabela 8 apresenta a distribuição por abordagem utilizada. Cabe ressaltar que algumas ferramentas, como as pagas, não puderam ser classificadas, sendo, portanto, excluídas dessa categorização. Também foram excluídas as que utilizam *Machine Learning* para reparar problemas de acessibilidade. Assim, no total, foram categorizadas 23 ferramentas, sendo 2 delas com adoção de análise estática e 21 com análise dinâmica.

Tabela 8. Distribuição de ferramentas por tipo de abordagem e estratégia adotadas.

Abordagem	**Estratégia**	**Ferramenta**	**Total**
Estática	Análise do código fonte	Android Lint AccessibiLint	2
Dinâmica	*Crawler*	AdMole Alshayban et al. Crawler Fok et al. Crawler Groundhog Xbot	5
	Record & Play	A11yPuppetry	1
	Scanner	Accessibility Scanner Apple's Accessibility Inspector IBM's MAC UI Automator Viewer Enhanced UI Automator Viewer	5
	Testes (Scripts ou automatizados de GUI)	AATK Alotaibi et al. Detection of Failures EarlGrey Espresso KIF Latte MATE Puma Roboelectric XCTest	10
Total			**23**

A análise estática utiliza recursos para examinar o código sem executá-lo, analisando sua estrutura e o processamento das variáveis [Li et al. 2017]. Este tipo de análise pode ser empregada para identificar erros funcionais e não funcionais no software. Entretanto, o número de verificações de problemas acaba sendo restrito por não ser possível contemplar os cenários que só podem ocorrer com o aplicativo em execução [Silva et al. 2020]. A análise dinâmica, por sua vez, pode encontrar mais violações do que a estática e utilizar diversas estratégias (como execução manual ou scripts de teste) para detectar problemas durante a execução do aplicativo. No entanto, esse processo tende a ser mais demorado e pode abranger apenas uma parte dos cenários, quando há, por exemplo, casos de testes insuficientes ou geração limitada de entradas [Silva et al. 2020].

Pergunta 4: De quais formas as ferramentas podem ser utilizadas?

As ferramentas dinâmicas, como frameworks de testes, scanners e *Record & Play* e *Crawlers*, podem ser aplicadas ao longo do processo de desenvolvimento ou até mesmo após o desenvolvimento ter sido finalizado. A estratégia adotada por *Crawlers* [Alshayban et al. 2020, Fok et al. 2022, Salehnamadi et al. 2023, He et al. 2024] é explorar um aplicativo e capturar dados de acessibilidade para cada estado visitado. Dessa forma, eles precisam que o aplicativo esteja em execução para que possa ocorrer a exploração.

Ferramentas do tipo *Record & Play* permitem que as interações com a tela sejam gravadas para análise posterior. Já os scanners podem operar de forma manual ou automatizada, ao varrer a interface em busca de falhas de acessibilidade. Como tanto os scanners quanto as ferramentas *Record & Play* podem ser utilizados com o aplicativo em emulador, o desenvolvedor pode realizar o escaneamento e implementar as correções necessárias. Os frameworks de testes podem ser adotados durante o desenvolvimento e até incorporados como critério de qualidade na integração de código. Contudo, para utilizá-los, o desenvolvedor deve configurar as bibliotecas adequadas em seu projeto.

Quanto às ferramentas que utilizam abordagem estática, seu funcionamento se dá por meio da análise do código-fonte. Para isso, é possível integrá-las no ambiente de desenvolvimento, permitindo verificações automáticas e emissão de avisos sobre violações. Observou-se que as ferramentas fornecidas pelos desenvolvedores dos sistemas operacionais são, frequentemente, disponibilizadas por bibliotecas ou integradas em *Integrated Development Environment*[5] (IDE), como o Accessibility Scanner [Google 2025d], o Accessibility Inspector [Apple Inc. 2025] e o Lint [Google 2025c].

Pergunta 5: Para quais sistemas operacionais essas ferramentas estão disponíveis?

Foram identificadas 13 ferramentas de uso geral, das quais 11 estão disponíveis para Android e 4 para iOS Figura(4). Além disso, foram encontradas 27 ferramentas específicas para a verificação de critérios de acessibilidade, sendo 24 para Android e 7 para iOS (Figura 5). Uma das ferramentas não foi considerada, pois é utilizada na etapa de design e não está diretamente associada a nenhum sistema operacional. Assim, considerando o total de 40 ferramentas, 35 estão disponíveis para Android e 11 para iOS. Ressalta-se que o total não corresponde ao somatório das ferramentas para cada SO porque algumas ferramentas podem ser utilizadas em ambas as plataformas.

[5]Em tradução liberal “Ambiente de Desenvolvimento Integrado”.

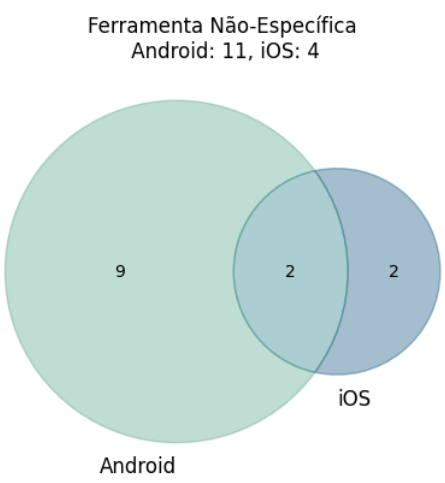

Figura 4. Distribuição de ferramentas não específicas para acessibilidade por SO.

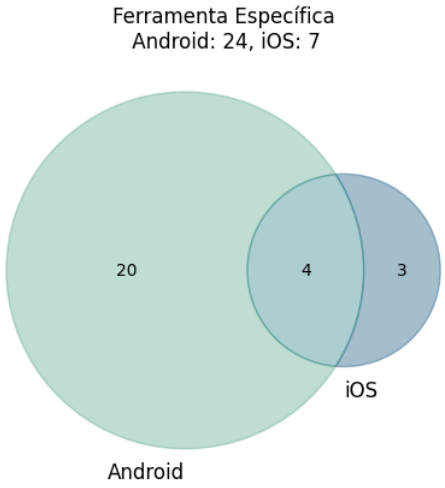

Figura 5. Distribuição de ferramentas específicas para acessibilidade por SO.

2.4. Discussão

Nesta seção, apresenta-se uma análise dos resultados obtidos após a condução da RSL. Inicialmente, buscou-se identificar os problemas de acessibilidade mais recorrentes. Ao analisar estudos que avaliaram aplicativos, os principais problemas encontrados foram: 1) alvo de toque, 2) contraste de texto, 3) rótulo do item, 4) contraste de imagem, 5) descrição dos itens e 6) itens clicáveis. No estudo de Chen *et al.* [Chen et al. 2022], que analisou o maior número de aplicativos, os problemas 1 a 5 representaram 93,1% do total. Isso sugere que tais falhas são reiteradas no desenvolvimento de aplicativos. Ainda no intuito de verificar a principalidade dos problemas, observa-se, na Figura 6, que as quatro maiores violações possuem médias significativamente superiores às demais.

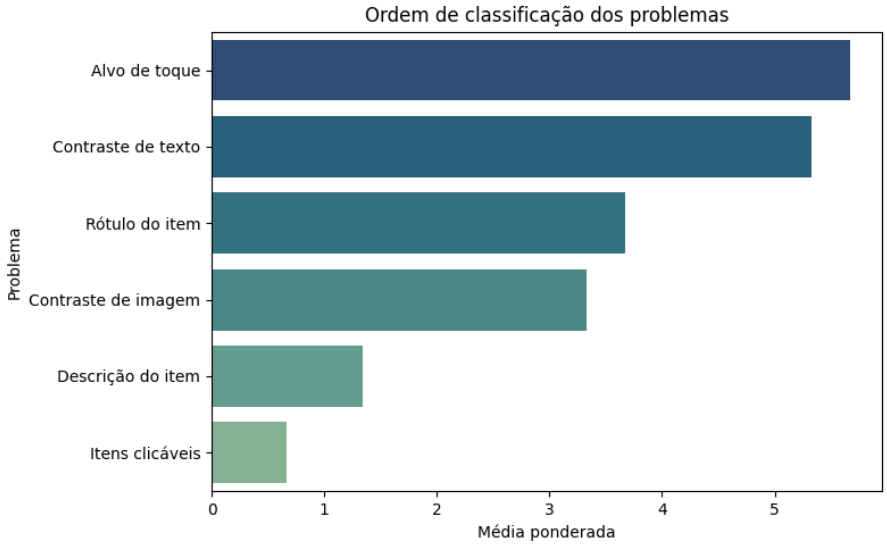

Figura 6. Classificação dos problemas por número de ocorrências.

A recorrência dos quatro primeiros problemas indica que esses ofensores devem ser analisados com atenção, o que justifica o desenvolvimento de ferramentas e estudos para sua identificação e correção. Como exemplo, tem-se o estudo de Fok *et al.*, que avaliou falhas de rótulo de item ausente em aplicativos Android [Fok et al. 2022]. Em outro estudo, Ross *et al.* analisaram 5.753 aplicativos para verificar o rótulo de botão baseado em imagem e constataram a existência de barreiras de acessibilidade [Ross et al. 2018].

Acerca dos impactos causados por tais problemas, observa-se, ao analisar suas descrições (Tabela 2), que todos podem prejudicar, em especial, a experiência de pessoas com deficiência visual. Isso porque as falhas interferem diretamente na interação com a UI, seja ela por toque ou por utilização de leitor de tela. Assim, a adoção de medidas e ferramentas que possam ajudar a desenvolver aplicações acessíveis torna-se necessária.

Com o levantamento das ferramentas disponíveis, foi possível observar que a maioria das mais mencionadas está associada aos desenvolvedores dos sistemas operacionais (Figura 7). Em particular, as ferramentas Accessibility Scanner e Android Lint, desenvolvidas pelo Google, se destacaram pelo número de menções (19 e 11, respectivamente). Entre os possíveis motivos para esse destaque, tem-se o fato de parte dessas ferramentas possuírem integração com a IDE Android Studio e oferecerem documentação detalhada [Google 2025c, Google 2025d] em páginas oficiais do Android.

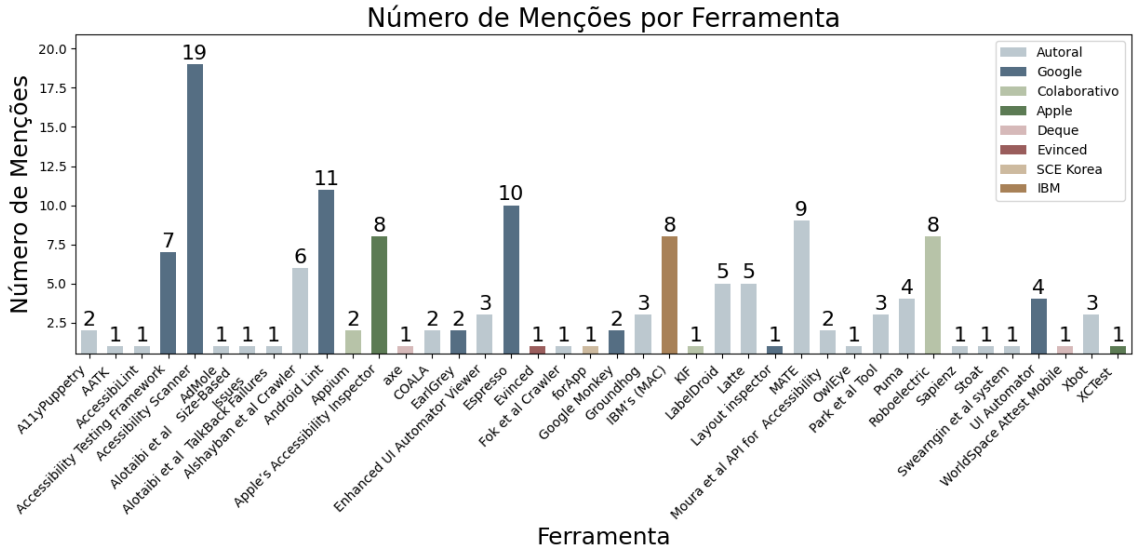

Figura 7. Número de menções por ferramenta.

Observa-se que várias ferramentas não destinadas exclusivamente à verificação de acessibilidade foram mencionadas nos estudos, destacando-se Lint, Espresso e Roboelectric. Ao considerar que tais ferramentas não são específicas para tal fim, pressupõe-se que o desenvolvedor deve ter o conhecimento necessário para utilizá-las de forma efetiva. No entanto, o estudo sobre práticas relacionadas à acessibilidade no desenvolvimento de software realizado por Bi *et al.* [Bi et al. 2022] evidenciou que uma parcela considerável de participantes não possuía nenhuma experiência profissional em acessibilidade.

Ao analisar os dados da distribuição de ferramentas por sistema operacional

(Figura 8), não surpreende o fato de que o número para Android seja três vezes maior que o para iOS, considerando que a base de usuários e o número de aplicativos para o sistema operacional do Google são significativamente superiores. Entretanto, mesmo com um maior número de ferramentas disponíveis, os estudos que avaliaram a acessibilidade dos aplicativos Android mostraram que muitos ainda apresentam violações. Isso indica a existência de espaço para pesquisa e desenvolvimento de novas soluções.

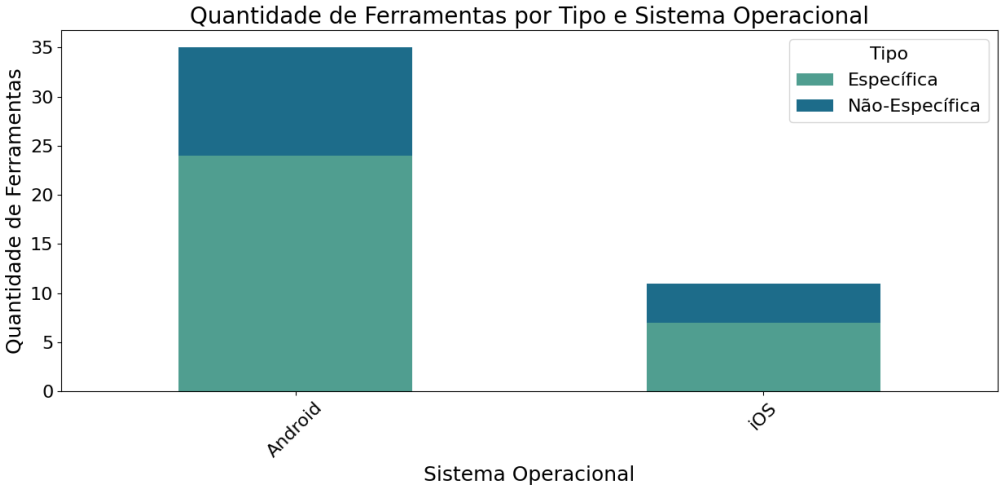

Figura 8. Distribuição de ferramentas por Sistema Operacional.

A partir dos dados da Tabela 8, nota-se que a maioria das ferramentas adota uma abordagem dinâmica, com diferentes estratégias e finalidades: algumas são voltadas para testes de scripts, outras para testes automatizados de interfaces gráficas (GUI), enquanto outras funcionam como *crawlers* para identificar falhas de acessibilidade. Além disso, existem aquelas que têm como objetivo tentar reparar os problemas encontrados. No entanto, apesar da predominância de ferramentas de análise dinâmica, é importante destacar que a segunda ferramenta mais mencionada nos artigos (Lint) adota uma abordagem estática.

A abordagem adotada impacta diretamente nas capacidades e limitações das ferramentas. A análise estática, por exemplo, pode ser utilizada para avaliar o código durante o desenvolvimento. Ferramentas que adotam esta abordagem, como o Lint, podem analisar o código sem a necessidade de execução do aplicativo ou da criação de casos de teste [Google 2025c]. Entretanto, a análise estática se mostra ineficiente ao não conseguir captar falhas que ocorrem em tempo de execução [Silva et al. 2020, Chen et al. 2022].

A análise dinâmica, por sua vez, é aplicada para identificar falhas enquanto o aplicativo está sendo executado. Esta abordagem supera as limitações da análise estática ao permitir configurar e avaliar todas as propriedades dinâmicas enquanto o aplicativo está em funcionamento [Silva et al. 2018]. Ferramentas com esta abordagem utilizam diferentes estratégias, que também resultam em diferentes ganhos e limitações.

Como exemplo, as ferramentas do tipo *scanner* que não possuem automação precisam de interação humana para visitar as telas dos aplicativo. Já frameworks de script de teste, como Espresso [Google 2025b], Robolectric [Robolectric 2025], Earl

Grey [Google 2025a] e KIF [KIF 2025], permitem verificar determinados critérios de acessibilidade, mas dependem da programação de cenários e têm como limitação o nível de abrangência desses cenários para simular as interações do usuário.

Existem também ferramentas de teste automatizado de GUI utilizadas para testar o comportamento da tela do aplicativo. Dentre elas, estão MATE [Eler et al. 2018] que verifica uma série de critérios a cada tela analisada e PUMA [Hao et al. 2014] que, apesar de ser utilizada para outros fins, permite a definição de checagem customizada de propriedades de acessibilidade. Entretanto, essas ferramentas possuem a limitação de depender da capacidade de cobertura dos diferentes cenários e telas do aplicativo [Silva et al. 2018].

Assim, mesmo com a abordagem dinâmica, existem pontos que não podem ser avaliados. Por isso, torna-se necessário combinar diferentes ferramentas ou processos para garantir a acessibilidade do aplicativo. Como exemplo, tem-se o estudo [Acosta-Vargas et al. 2021a], no qual o Acessibility Scanner foi utilizado combinado com uma revisão manual baseada nos critérios da WCAG 2.1. Em 2020, Acosta-Vargas *et al.* adotaram a mesma ferramenta em estudo no qual propuseram o desenvolvimento de um aplicativo seguindo critérios de acessibilidade ao longo do processo de desenvolvimento [Acosta-Vargas et al. 2020].

Nesse sentido, ao considerar como as ferramentas podem ser utilizadas, observa-se que aquelas que adotam abordagem estática tendem a ser empregadas durante etapas iniciais do desenvolvimento. Quanto às ferramentas dinâmicas, percebe-se que podem ser adotadas em diferentes estágios. Ferramentas de teste, por exemplo, podem ser utilizadas durante o desenvolvimento e até mesmo incorporadas à esteira de desenvolvimento para garantir a integridade da aplicação. No entanto, o desenvolvedor precisa detalhar os casos de teste para que os cenários sejam efetivamente validados, caso contrário, podem ocorrer testes insuficientes ou cobertura apenas de cenários parciais da aplicação. Ferramentas como scanner e *crawlers* de UI podem ser utilizadas após ou durante o desenvolvimento, desde que o aplicativo já esteja em execução (em um emulador, por exemplo).

Swearngin *et al.* [Swearngin et al. 2024] enfatizam a importância de verificar a acessibilidade desde os estágios iniciais, propondo, inclusive, que a ferramenta detalhada em seu estudo possa ser incorporada ao processo de integração contínua. De fato, incluir a acessibilidade durante o processo de desenvolvimento pode se mostrar vantajoso por diversos aspectos. Por exemplo, Ross *et al.* [Ross et al. 2018] mostraram que a ausência de rótulo de item e rótulos de item com descrições pouco informativas são problemas recorrentes. Porém, os autores também identificaram que, uma vez que o rótulo é adicionado, ele tende a ter um conteúdo informativo. Nesse sentido, a ideia de incluir um rótulo durante o desenvolvimento poderia representar uma diminuição nas falhas de acessibilidade tanto para falha de rótulo ausente quanto para a falha de rótulo pouco informativo.

Na mesma perspectiva, Alshabayan *et al.* apontam que, ao identificar problemas de acessibilidade no início da fase de *design*, em vez de no final do processo de desenvolvimento, é possível que o desenvolvedor faça ajustes na interface com menor esforço [Alshayban et al. 2020]. Isso pode resultar em menos horas de retrabalho, o que, por sua vez, contribui para a redução dos custos com correções. Siebra *et al.* também

afirmam que testes de acessibilidade devem ser iniciados o quanto antes e citam como exemplo a adoção de ferramentas de análise estática [Siebra et al. 2018].

Em adição às ferramentas, foram identificados métodos voltados ao desenvolvimento e avaliação para aplicativos mais acessíveis. Entretanto, mesmo ao considerar que a adoção de tais métodos seja vantajosa, ferramentas voltadas à acessibilidade ainda são fundamentais durante o processo de desenvolvimento. Considerando o conhecimento insuficiente de parte dos desenvolvedores sobre o tema, bem como a necessidade de combinar diferentes ferramentas para se alcançar resultados satisfatórios, observa-se que garantir a acessibilidade nos aplicativos representa um desafio significativo. Percebe-se que o tema tem sido explorado de diferentes formas pela comunidade científica. Porém, a inacessibilidade nos aplicativos apontada por estudos anteriores [Chen et al. 2022, Alshayban et al. 2020, Acosta-Vargas et al. 2019, Acosta-Vargas et al. 2021a] indica que ainda há trabalho a ser feito. Ressalta-se a importância dos estudos voltados à garantia de que pessoas com deficiência sejam consideradas no processo de desenvolvimento dos aplicativos, o que reforça a relevância de a temática estar inserida em um dos grandes desafios de pesquisa em IHC no Brasil [Pereira et al. 2024].

2.5. Ameaças à validade

Esta revisão sistemática foi conduzida de forma criteriosa, seguindo o modelo PRISMA. Contudo, deve-se considerar que a existência de ameaças à validade da pesquisa pode ter impacto sobre resultados e conclusões. A fim de evidenciar tais pontos para que trabalhos futuros de revisão possam ser aprimorados, são listadas, a seguir, as ameaças relacionadas ao presente estudo, tendo sido adotada a classificação de Wohlin *et al* [Wohlin et al. 2012]

Validade de construção: Uma ameaça à validade desta revisão está relacionada à construção da string de busca, que exigiu explicitamente a presença dos termos Android ou iOS no título, resumo ou palavras-chave. Essa decisão foi motivada por um estudo preliminar que resultou em um número elevado de publicações irrelevantes ao tema investigado, o que dificultava a triagem e comprometia a precisão dos resultados. No entanto, reconhece-se que essa abordagem pode ter levado à exclusão de trabalhos pertinentes que, embora discutam plataformas móveis ou experiências de uso em sistemas Android ou iOS, não mencionam diretamente esses termos nos campos considerados na busca. Assim, existe a possibilidade de que contribuições relevantes à área tenham sido inadvertidamente omitidas, o que constitui uma limitação na abrangência da revisão.

Validade externa: A seleção das bases de dados foi motivada pela relevância e representatividade na área de Computação e pela disponibilidade de acesso integral aos artigos. Optou-se por não incluir bases indexadoras como Scopus, por serem agregadores e não fontes originais, o que poderia causar redundância e dificultar o acesso completo aos textos. Nesse sentido, considera-se que tal seleção também se configura como uma possível ameaça, uma vez que os resultados encontrados podem não corresponder a uma visão mais ampla sobre o tema.

Validade interna: É possível que algum estudo relevante não seja retornado, por motivos operacionais, durante a realização da pesquisa. Para verificar a reprodutibilidade dos resultados encontrados, as buscas foram realizadas duas vezes: no primeiro dia, os resultados foram tabelados. No segundo dia, a busca foi novamente realizada para garantir

que todos os resultados fossem obtidos.

Validade da conclusão: A definição de critérios de inclusão e exclusão pode impactar o resultado de estudos selecionados, ocasionando um grande número de trabalhos retornados para avaliação ou a exclusão de algum trabalho relevante. Para mitigar essa ameaça, os critérios foram inicialmente definidos e, ao longo da execução da RSL, reavaliados. Como os estudos encontrados estavam atendendo aos objetivos definidos para a pesquisa, os critérios foram considerados como adequados. Outro ponto de atenção está na definição de problemas mais recorrentes, visto que é possível que problemas relevantes para a pesquisa não tenham sido reportados em algum dos estudos selecionados. Para mitigar essa ameaça, a estratégia adotada consistiu em verificar diferentes estudos que avaliavam a acessibilidade em aplicativos, a fim de aumentar a cobertura de menções.

3. Trabalhos relacionados

Dentre os estudos relacionados ao teor desta pesquisa, está o de Masruroh *et al.* [Masruroh et al. 2022] com uma revisão sistemática realizada para identificar quais tópicos e métodos são utilizados na avaliação da usabilidade e acessibilidade. Os autores pesquisaram estudos no período compreendido entre 2017 e 2021 e, como resultado, indicaram quatorze métodos de avaliação e sete tópicos nos quais se concentram os testes de usabilidade, sendo um deles a acessibilidade.

No artigo [Oliveira e Eler 2024], os pesquisadores apresentaram uma revisão sistemática da literatura sobre a acessibilidade de aplicativos móveis a partir da coleta e análise de avaliações de usuários. Como resultado, foram identificadas quatro estratégias para coleta e análise de avaliações e barreiras que impactam usuários com deficiência.

Acosta-Vargas *et al.* [Acosta-Vargas et al. 2021b] realizaram uma revisão de escopo acerca da acessibilidade em aplicativos móveis para pessoas com deficiência cognitiva, motora e sensorial. Após avaliarem estudos entre 2000 e 2020, os autores evidenciaram as deficiências que são mais consideradas para avaliação, as diretrizes aplicadas, os métodos e técnicas utilizados, os domínios mais avaliados e as ferramentas para testar acessibilidade. Ressalta-se que, quanto à identificação de ferramentas, o foco foi dado àquelas destinadas a teste, tendo sido citado apenas o Accessibility Scanner.

Este estudo se diferenciou dos demais ao realizar uma revisão sistemática da acessibilidade com foco no processo de desenvolvimento. Para isso, além da identificação dos problemas mais recorrentes nos aplicativos, esta pesquisa direcionou a atenção para as ferramentas, buscando identificá-las e classificá-las quanto à abordagem, a fim de entender como e em que etapa são utilizadas. Assim, buscou-se contribuir com a temática para que novas pesquisas e estudos possam abordar a criação de ferramentas ou recomendações que possibilitem o desenvolvimento de aplicações mais acessíveis.

4. Cuidados éticos

Este artigo apresentou uma revisão sistemática da literatura, cujos dados foram obtidos a partir de estudos disponíveis em repositórios públicos de artigos acadêmicos. Durante a execução da pesquisa, não foram coletados dados primários e não houve envolvimento direto de seres humanos ou animais. Por essa razão, não foi necessária a submissão ao Comitê de Ética em Pesquisa.

5. Considerações Finais

Baseado no levantamento realizado, conclui-se que garantir a acessibilidade em aplicativos é uma tarefa complexa, que demanda a combinação de diferentes esforços [Seixas Pereira et al. 2024, Silva et al. 2020, Silva et al. 2018]. Reforça-se que a adoção de determinadas abordagens ou ferramentas para validação dos fluxos não exclui a necessidade de testes automatizados ou manuais, incluindo testes realizados por profissionais especializados em acessibilidade ou pessoas com deficiência. [Swearngin et al. 2024].

Os resultados indicaram a prevalência da adoção da abordagem dinâmica em relação à estática. Tais dados podem ser explicados pelo fato de a última apresentar limitações para avaliar cenários que só ocorrem em tempo de execução, uma vez que sua estratégia consiste em analisar o código fonte. Entretanto, um ponto importante a ser considerado é que, na integração de código, é possível que sejam inseridos novos problemas de acessibilidade ou que alguma funcionalidade seja impactada negativamente.

Evidencia-se, assim, que diretrizes e técnicas direcionadas à acessibilidade devem ser adotadas no desenvolvimento de software [Santiago e Marques 2023]. A exemplo disso, o estudo de Dos Santos *et al.* [Dos Santos et al. 2024] alerta para que sejam implementadas medidas que impeçam a adição de novas barreiras de acessibilidade quando novas funcionalidades são adicionadas às aplicações. Tendo em vista a integração de código no processo de desenvolvimento de software, percebe-se que a abordagem estática pode ser vantajosa. Isso indica que há espaço para se pensar em novas ferramentas que utilizem tal abordagem para minimizar os riscos de adição de códigos com falhas de acessibilidade.

Diante disso, como trabalhos futuros, pretende-se realizar uma atualização da RSL que inclua bases indexadoras, como SCOPUS, e bases relevantes para IHC no contexto brasileiro, como SBC OpenLib (SOL). Almeja-se, nessa atualização, aprofundar a análise qualitativa da usabilidade das ferramentas. Em adição, ao considerar os problemas identificados, como continuação desta pesquisa, espera-se elaborar um conjunto de regras baseadas na WCAG 2.2. Essas regras poderão ser utilizadas por uma ferramenta de análise estática durante o desenvolvimento de aplicativos Android nativos a fim de verificar critérios de acessibilidade que impactem o uso por pessoas com deficiência visual.

Referências

Acosta-Vargas, P., Guaña-Moya, J., Jadán-Guerrero, J., Alvites-Huamaní, C., e Salvador-Ullauri, L. (2021a). Towards accessibility assessment with a combined approach for native mobile applications. In Nunes, I. L., editor, *Advances in Human Factors and System Interactions*, pages 234–241, Cham. Springer International Publishing.

Acosta-Vargas, P., Salvador-Acosta, B., Salvador-Ullauri, L., Villegas-Ch., W., e Gonzalez, M. (2021b). Accessibility in native mobile applications for users with disabilities: A scoping review. *Applied Sciences*, 11(12).

Acosta-Vargas, P., Serrano-Costales, L., Salvador-Ullauri, L., Nunes, I. I., e Gonzalez, M. (2020). Toward accessible mobile application development for users with low vision. In *Advances in Human Factors and Systems Interaction: Proceedings of the AHFE 2020 Virtual Conference on Human Factors and Systems Interaction, July 16-20, 2020, USA*, pages 236–241. Springer.

Acosta-Vargas, P., Zalakeviciute, R., Luján-Mora, S., e Perdomo, W. (2019). *Accessibility Evaluation of Mobile Applications for Monitoring Air Quality: Helping Teachers Develop Research Informed Practice*, pages 638–648.

Alshayban, A., Ahmed, I., e Malek, S. (2020). Accessibility issues in android apps: state of affairs, sentiments, and ways forward. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ICSE '20, page 1323–1334, New York, NY, USA. Association for Computing Machinery.

Apple Inc. (2025). Accessibility Inspector. Disponível em: `https://developer.apple.com/documentation/accessibility/accessibility-inspector`. Acesso em: 26 jan. 2025.

Bi, T., Xia, X., Lo, D., Grundy, J., Zimmermann, T., e Ford, D. (2022). Accessibility in software practice: A practitioner's perspective. *ACM Trans. Softw. Eng. Methodol.*, 31(4).

Chen, S., Chen, C., Fan, L., Fan, M., Zhan, X., e Liu, Y. (2022). Accessible or not? an empirical investigation of android app accessibility. *IEEE Transactions on Software Engineering*, 48(10):3954–3968.

Dos Santos, P. S. H., Oliveira, A. D. A., De Jesus, T. B. N., Aljedaani, W., e Eler, M. M. (2024). Evolution may come with a price: analyzing user reviews to understand the impact of updates on mobile apps accessibility. In *Proceedings of the XXII Brazilian Symposium on Human Factors in Computing Systems*, IHC '23, New York, NY, USA. Association for Computing Machinery.

Eler, M. M., Rojas, J. M., Ge, Y., e Fraser, G. (2018). Automated accessibility testing of mobile apps. In *2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)*, pages 116–126.

Epic Games (2025). Unreal Engine. Disponível em: `https://www.unrealengine.com/en-US`. Acesso em: 15 jan. 2025.

Fok, R., Zhong, M., Ross, A. S., Fogarty, J., e Wobbrock, J. O. (2022). A large-scale longitudinal analysis of missing label accessibility failures in android apps. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Google (2025a). Earlgrey. Disponível em: `https://github.com/google/EarlGrey`. Acesso em: 23 fev. 2025.

Google (2025b). Espresso | teste seu aplicativo no android. Disponível em: `https://developer.android.com/training/testing/espresso?hl=pt-br`. Acesso em: 23 fev. 2025.

Google (2025c). Melhorar seu código com verificações de lint. Disponível em: `https://developer.android.com/studio/write/lint?hl=pt-br#overview`. Acesso em: 25 jan. 2025.

Google (2025d). Testar a acessibilidade do seu app. Disponível em: `https://developer.android.com/guide/topics/ui/accessibility/testing?hl=pt-br`. Acesso em: 26 jan. 2025.

Hao, S., Liu, B., Nath, S., Halfond, W., e Govindan, R. (2014). Puma: programmable ui-automation for large-scale dynamic analysis of mobile apps.

He, Z., Huq, S. F., e Malek, S. (2024). "I tend to view ads almost like a pestilence": On the Accessibility Implications of Mobile Ads for Blind Users. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, New York, NY, USA. Association for Computing Machinery.

KIF (2025). Kif - keep it functional. Disponível em: `https://github.com/kif-framework/KIF`. Acesso em: 23 fev. 2025.

Krainz, E., Miesenberger, K., e Feiner, J. (2018). Can we improve app accessibility with advanced development methods? In *Computers Helping People with Special Needs: 16th International Conference, ICCHP 2018, Linz, Austria, July 11-13, 2018, Proceedings, Part I*, page 64–70, Berlin, Heidelberg. Springer-Verlag.

Laricchia, F. (2024). Global smartphone penetration 2016-2023. Disponível em: `https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/`. Acesso em: 22 fev. 2025.

Li, L., Bissyandé, T., Papadakis, M., Rasthofer, S., Bartel, A., Octeau, D., Klein, J., e Le Traon, Y. (2017). Static analysis of android apps: A systematic literature review. *Information and Software Technology*, 88.

Masruroh, S. U., Rizqy Vitalaya, N. A., Sukmana, H. T., Subchi, I., Khairani, D., e Durachman, Y. (2022). Evaluation of usability and accessibility of mobile application for people with disability: Systematic literature review. In *2022 International Conference on Science and Technology (ICOSTECH)*, pages 1–7.

Mateus, D. A., Silva, C. A., de Oliveira, A. F. B. A., Costa, H., e Freire, A. P. (2021). A systematic mapping of accessibility problems encountered on websites and mobile apps: A comparison between automated tests, manual inspections and user evaluations. *Journal on Interactive Systems*, 12(1):145–171.

Oliveira, A. e Eler, M. (2024). Exploring accessibility of mobile applications through user feedback: Insights from app reviews in a systematic literature review. In *Anais do XXIII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais*, pages 633–647, Porto Alegre, RS, Brasil. SBC.

Pereira, R., Darin, T., e Silveira, M. (2024). Grandihc-br: Grand research challenges in human-computer interaction in brazil for 2025-2035. In *Anais do XXIII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais*, pages 915–938, Porto Alegre, RS, Brasil. SBC.

Prisma Statements (2020). Prisma 2020. Disponível em: `https://www.prisma-statement.org/`. Acesso em: 20 nov. 2024.

Robolectric (2025). Robolectric. Disponível em: `https://robolectric.org/`. Acesso em: 23 fev. 2025.

Ross, A. S., Zhang, X., Fogarty, J., e Wobbrock, J. O. (2017). Epidemiology as a framework for large-scale mobile application accessibility assessment. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers*

and Accessibility, ASSETS '17, page 2–11, New York, NY, USA. Association for Computing Machinery.

Ross, A. S., Zhang, X., Fogarty, J., e Wobbrock, J. O. (2018). Examining image-based button labeling for accessibility in android apps through large-scale analysis. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '18, page 119–130, New York, NY, USA. Association for Computing Machinery.

Salehnamadi, N., Mehralian, F., e Malek, S. (2023). Groundhog: An automated accessibility crawler for mobile apps. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, ASE '22, New York, NY, USA. Association for Computing Machinery.

Santiago, M. T. e Marques, A. B. (2023). Exploring user reviews to identify accessibility problems in applications for autistic users. *Journal on Interactive Systems*, 14(1):317–330.

Seixas Pereira, L., Matos, M., e Duarte, C. (2024). Exploring mobile device accessibility: Challenges, insights, and recommendations for evaluation methodologies. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

Siebra, C. A., Correia, W., Penha, M., Macêdo, J., Quintino, J., Anjos, M., Florentin, F., Silva, F. Q. B., e Santos, A. L. M. (2018). An analysis on tools for accessibility evaluation in mobile applications. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*, SBES '18, page 172–177, New York, NY, USA. Association for Computing Machinery.

Silva, C., Eler, M. M., e Fraser, G. (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 '18, page 286–293, New York, NY, USA. Association for Computing Machinery.

Silva, H., Endo, A., Eler, M., Vergilio, S., e Durelli, V. (2020). On the relation between code elements and accessibility issues in android apps.

Swearngin, A., Wu, J., Zhang, X., Gomez, E., Coughenour, J., Stukenborg, R., Garg, B., Hughes, G., Hilliard, A., Bigham, J. P., e Nichols, J. (2024). Towards automated accessibility report generation for mobile apps. *ACM Trans. Comput.-Hum. Interact.*, 31(4).

Unity Technologies (2025). Unity. Disponível em: `https://unity.com/`. Acesso em: 15 jan. 2025.

W3C Brasil (2025). Conhecendo o W3C. Disponível em: `https://www.w3c.br/conhecendo-o-w3c`. Acesso em: 24 fev. 2025.

W3C Web Accessibility Initiative (1999). Web Content Accessibility Guidelines 1.0. Disponível em: `https://www.w3.org/TR/WAI-WEBCONTENT/`. Acesso em: 24 fev. 2025.

W3C Web Accessibility Initiative (WAI) (2025). What's New in WCAG 2.2. Disponível em: https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/. Acesso em: 16 fev. 2025.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A., et al. (2012). *Experimentation in software engineering*, volume 236. Springer.

World Health Organization (2023). Disability and health. Disponível em: https://www.who.int/news-room/fact-sheets/detail/disability-and-health. Acesso em: 22 out. 2024.

Zhang, X., de Greef, L., Swearngin, A., White, S., Murray, K., Yu, L., Shan, Q., Nichols, J., Wu, J., Fleizach, C., Everitt, A., e Bigham, J. P. (2021). Screen recognition: Creating accessibility metadata for mobile applications from pixels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA. Association for Computing Machinery.

Zhang, X., Ross, A. S., e Fogarty, J. (2018). Robust annotation of mobile application interfaces in methods for accessibility repair and enhancement. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, page 609–621, New York, NY, USA. Association for Computing Machinery.

A. Apêndice

Relação dos estudos selecionados incluídos na revisão sistemática	
1	ACOSTA-VARGAS, Patricia et al. Accessibility evaluation of mobile applications for monitoring air quality. In: Information Technology and Systems: Proceedings of ICITS 2019. Springer International Publishing, 2019. p. 638-648.
2	ACOSTA-VARGAS, Patricia et al. Toward accessible mobile application development for users with low vision. In: Advances in Human Factors and Systems Interaction: Proceedings of the AHFE 2020 Virtual Conference on Human Factors and Systems Interaction, July 16-20, 2020, USA. Springer International Publishing, 2020. p. 236-241.
3	ACOSTA-VARGAS, Patricia et al. Towards accessibility assessment with a combined approach for native mobile applications. In: Advances in Human Factors and System Interactions: Proceedings of the AHFE 2021 Virtual Conference on Human Factors and Systems Interaction, July 25-29, 2021, USA. Springer International Publishing, 2021. p. 234-241.
4	ALOTAIBI, Ali S.; CHIOU, Paul T.; HALFOND, William GJ. Automated repair of size-based inaccessibility issues in mobile applications. In: 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2021. p. 730-742.
5	ALSHAYBAN, Abdulaziz; AHMED, Iftekhar; MALEK, Sam. Accessibility issues in android apps: state of affairs, sentiments, and ways forward. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering. 2020. p. 1323-1334.
6	BI, Tingting et al. Accessibility in software practice: A practitioner's perspective. ACM Transactions on Software Engineering and Methodology (TOSEM), v. 31, n. 4, p. 1-26, 2022.
7	CHEN, Sen et al. Accessible or not? an empirical investigation of android app accessibility. IEEE Transactions on Software Engineering, v. 48, n. 10, p. 3954-3968, 2021.
8	DA SILVA, Henrique Neves et al. On the relation between code elements and accessibility issues in android apps. In: Proceedings of the 5th Brazilian Symposium on Systematic and Automated Software Testing. 2020. p. 40-49.
9	FOK, Raymond et al. A large-scale longitudinal analysis of missing label accessibility failures in android apps. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. 2022. p. 1-16.
10	HE, Ziyao; HUQ, Syed Fatiul; MALEK, Sam. "I tend to view ads almost like a pestilence": On the Accessibility Implications of Mobile Ads for Blind Users. In: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering. 2024. p. 1-13.
11	KASHIF, Sidrah et al. Development of An Automated Accessibility Evaluation Plugin Tool for Mobile Applications. In: 2022 3rd International Conference on Innovations in Computer Science & Software Engineering (ICONICS). IEEE, 2022. p. 1-10.

12	MATOS, Maria; SEIXAS PEREIRA, Letícia; DUARTE, Carlos. Evaluation of the Accessibility of Mobile Applications: Current Approaches and Challenges. In: International Conference on Human-Computer Interaction. Cham: Springer Nature Switzerland, 2023. p. 352-371.
13	MEHRALIAN, Forough; SALEHNAMADI, Navid; MALEK, Sam. Data-driven accessibility repair revisited: on the effectiveness of generating labels for icons in Android apps. In: Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2021. p. 107-118.
14	PARK, Eunju et al. Development of automatic evaluation tool for mobile accessibility for android application. In: 2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS). IEEE, 2019. p. 1-6.
15	PATIL, Neha; BHOLE, Dhananjay; SHETE, Prasanna. Enhanced UI Automator Viewer with improved Android accessibility evaluation features. In: 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT). IEEE, 2016. p. 977-983.
16	ROSS, Anne Spencer et al. Epidemiology as a framework for large-scale mobile application accessibility assessment. In: Proceedings of the 19th international ACM SIGACCESS conference on computers and accessibility. 2017. p. 2-11.
17	ROSS, Anne Spencer et al. Examining image-based button labeling for accessibility in Android apps through large-scale analysis. In: Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility. 2018. p. 119-130.
18	SALEHNAMADI, Navid et al. Latte: Use-case and assistive-service driven automated accessibility testing framework for android. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. 2021. p. 1-11.
19	SEIXAS PEREIRA, Letícia; MATOS, Maria; DUARTE, Carlos. Exploring Mobile Device Accessibility: Challenges, Insights, and Recommendations for Evaluation Methodologies. In: Proceedings of the CHI Conference on Human Factors in Computing Systems. 2024. p. 1-17.
20	SIEBRA, Clauirton A. et al. An analysis on tools for accessibility evaluation in mobile applications. In: Proceedings of the XXXII Brazilian Symposium on Software Engineering. 2018. p. 172-177.
21	SILVA, Camila; ELER, Marcelo Medeiros; FRASER, Gordon. 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. 2018. p. 286-293.
22	SWEARNGIN, Amanda et al. Towards automated accessibility report generation for mobile apps. ACM Transactions on Computer-Human Interaction, v. 31, n. 4, p. 1-44, 2024.
23	ZHANG, Xiaoyi et al. Screen recognition: Creating accessibility metadata for mobile applications from pixels. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. 2021. p. 1-15.