

Flutter Access: Avaliando Acessibilidade em Dispositivos Móveis com Flutter

Caroline Matilde Lima Vitória
Instituto Federal de Educação, Ciência e
Tecnologia da Bahia
Rua Emídio dos Santos, s/n – Barbalho
carol.lima.vitoria@gmail.com

Simone da Silva Amorim
Instituto Federal de Educação, Ciência e
Tecnologia da Bahia
Rua Emídio dos Santos, s/n – Barbalho
simone.amorim@ifba.edu.br

RESUMO

Aplicativos móveis têm papel fundamental na vida cotidiana, oferecendo praticidade e acesso a serviços em diferentes contextos. No entanto, usuários com algum tipo de deficiência frequentemente encontram barreiras que comprometem sua interação com essas tecnologias. Embora tenha havido avanços para melhoria desta interação nos últimos anos, ainda são escassas as soluções que apoiam equipes de desenvolvimento de software na detecção e correção de barreiras de acessibilidade durante a etapa de codificação. Para tentar responder a essa lacuna, este trabalho apresenta o *Flutter Access*, uma ferramenta criada a partir das necessidades relatadas por desenvolvedores e gestores, com o objetivo de apoiar a avaliação e a melhoria contínua da acessibilidade em aplicativos móveis diretamente no fluxo de desenvolvimento. A solução foi validada junto a profissionais da área, demonstrando eficácia na identificação precoce de barreiras de acessibilidade e boa aceitação no fluxo de trabalho. Nesse sentido, ao integrar boas práticas à etapa de escrita do código, a proposta busca facilitar a construção de aplicativos em *Flutter* mais inclusivos e capazes de oferecer experiências satisfatórias a um público diverso.

Palavras-chave

Acessibilidade digital; Flutter; Aplicativos móveis; Desenvolvimento de software

1. INTRODUÇÃO

A popularização dos dispositivos móveis transformou a forma como as pessoas se comunicam, trabalham e acessam informações, tornando os aplicativos indispensáveis na vida cotidiana. Segundo dados do portal FGV, o Brasil possui mais de um smartphone por habitante, consolidando esses aparelhos como o principal meio de acesso à internet no país [1]. Nesse cenário, a experiência do usuário tornou-se um fator determinante para o sucesso de aplicativos, exigindo que as aplicações sejam funcionais, intuitivas e acessíveis a todos os públicos. Um exemplo prático disso é observado

em aplicativos de e-commerce: plataformas que apresentam fluxos de finalização de compra complexos ou botões de difícil identificação geram frustração e abandono do carrinho, impactando diretamente o faturamento da empresa. Nesse sentido, um design bem executado deve permitir que o usuário entenda como utilizar o produto sem a necessidade de manuais, focando na visibilidade das funções e no feedback imediato das ações [2]. Com o crescimento do uso de dispositivos móveis, essas ferramentas passaram a atender demandas essenciais da sociedade. De acordo com a Pesquisa TIC Domicílios, o uso de celulares é predominante para a realização de serviços públicos, atividades de ensino e busca por informações de saúde, tornando-se o pilar da inclusão digital contemporânea [3].

Apesar dessa importância, ainda existem muitos obstáculos de acessibilidade que dificultam ou até impossibilitam o uso de aplicativos por pessoas com deficiência. Usuários com deficiências visuais, motoras ou cognitivas frequentemente enfrentam barreiras relacionadas à ausência de descrições semânticas, má utilização de componentes de interface e falta de compatibilidade com tecnologias assistivas, como leitores de tela [4]. Essas limitações impactam diretamente a inclusão digital e a autonomia desses usuários no uso de aplicações móveis.

No ambiente de desenvolvimento de software, a acessibilidade ainda é tratada, em muitos casos, como um requisito secundário ou como uma etapa posterior ao desenvolvimento da aplicação. A ausência de ferramentas automatizadas de apoio, aliada à falta de conhecimento técnico específico sobre acessibilidade por parte de desenvolvedores, contribui para que erros recorrentes sejam introduzidos no código-fonte e só sejam identificados em fases avançadas do projeto ou, em alguns casos, nunca sejam corrigidos. Esse cenário evidencia a necessidade de soluções que auxiliem os desenvolvedores a identificar barreiras de acessibilidade ainda durante a implementação do código.

Diante desse contexto, este trabalho propõe o desenvolvimento do *Flutter Access*, uma ferramenta de análise estática voltada à identificação de problemas de acessibilidade em aplicações Flutter. A ferramenta atua diretamente sobre o código-fonte, utilizando o Dart Analyzer para percorrer a Árvore Sintática Abstrata (AST) e aplicar regras de acessibilidade previamente definidas, fornecendo alertas claros e precisos sobre possíveis falhas que afetam a acessibilidade da aplicação.

2. MOTIVAÇÃO E JUSTIFICATIVA

De acordo com a Organização das Nações Unidas (ONU), existem atualmente 1,3 bilhão de pessoas com algum tipo de deficiência no mundo, o que representa aproximadamente 16% da população global [5]. Esse dado revela a dimensão da necessidade de promover inclusão e acessibilidade em todos os aspectos da vida social, incluindo o ambiente digital. No artigo “Inovação para mais inclusão marca Dia Internacional das Pessoas com Deficiência”, publicado em 2022 pela ONU News, a organização destaca que a acessibilidade digital é um direito humano fundamental, sendo a tecnologia uma das principais ferramentas para garantir que pessoas com deficiência possam estudar, trabalhar, se comunicar e participar ativamente da sociedade [5].

No contexto brasileiro, segundo o portal Governo Digital, a acessibilidade digital tem como propósito democratizar o acesso à informação e garantir que todos os cidadãos possam compreender e interagir com os serviços públicos e privados oferecidos na internet, independentemente de suas limitações físicas, sensoriais ou cognitivas [6]. Ainda segundo dados do Censo IBGE de 2010, divulgados pelo mesmo órgão, aproximadamente 45 milhões de brasileiros, o que equivale a 23,9% da população, apresentam algum tipo de deficiência [6]. Esses números demonstram a relevância social do tema e evidenciam que a acessibilidade deve ser tratada como requisito essencial no desenvolvimento de soluções digitais.

Apesar dos avanços obtidos em políticas públicas e normas técnicas, como a ABNT NBR 17060 [7], que trata da acessibilidade em dispositivos móveis, e das recomendações internacionais da W3C (*World Wide Web Consortium*) [8], ainda é comum observar que os aplicativos móveis apresentam barreiras de navegação, contraste inadequado, ausência de rótulos descritivos e falta de compatibilidade com tecnologias assistivas. Grande parte dessas falhas decorre não da ausência de ferramentas, mas da insuficiência ou das limitações das soluções atualmente disponíveis, que nem sempre facilitam de forma integrada a aplicação e a verificação das boas práticas de acessibilidade durante o desenvolvimento de software.

O cenário se torna ainda mais crítico com a popularização de frameworks multiplataforma, como o Flutter, amplamente utilizado por desenvolvedores para criar aplicativos Android e iOS a partir de um único código-fonte. Embora o Flutter ofereça suporte básico à acessibilidade [9], ele ainda carece de soluções específicas que auxiliem o programador a avaliar e corrigir problemas de acessibilidade diretamente no código, de forma automatizada e prática.

Essa carência de ferramentas integradas faz com que muitos desenvolvedores de aplicações móveis deixem de implementar recursos acessíveis, seja por desconhecimento técnico ou pela percepção de alta complexidade envolvida na aplicação das diretrizes em interfaces compactas. Sob tal aspecto, a acessibilidade em dispositivos móveis é frequentemente negligenciada devido à falta de suporte adequado nas ferramentas de desenvolvimento e ao desconhecimento dos desenvolvedores sobre como as tecnologias assistivas, como leitores de tela, interagem com os componentes nativos dos aplicativos [4].

Dessa forma, a motivação para o desenvolvimento de uma solução voltada à avaliação de acessibilidade em dispositivos móveis desenvolvidos com Flutter surge da necessidade de reduzir as barreiras enfrentadas por pessoas com deficiência e de fornecer suporte técnico aos desenvolvedores, promovendo um ecossistema digital mais inclusivo. Nesse sentido, além de contribuir para o cumprimento de leis e normas de acessibilidade, uma ferramenta com esse propósito pode melhorar a qualidade, a experiência do usuário e a responsabilidade social das aplicações produzidas, tornando a tecnologia um instrumento efetivo de inclusão e cidadania.

3. TRABALHOS RELACIONADOS

Com o avanço das tecnologias móveis e o aumento da preocupação com a inclusão digital, a acessibilidade em aplicações para dispositivos móveis tem se tornado um tema de grande relevância tanto na indústria quanto na academia. Diversos estudos e projetos têm buscado compreender as barreiras enfrentadas por pessoas com deficiência no uso de aplicativos e propor ferramentas e metodologias que auxiliem desenvolvedores a criar interfaces mais acessíveis [10].

Nos últimos anos, observa-se um crescimento no número de pesquisas voltadas à avaliação automática de acessibilidade, à implementação de diretrizes do WCAG (*Web Content Accessibility Guidelines*) em contextos mobile, e à integração de práticas de acessibilidade em frameworks de desenvolvimento, como Android, iOS e Flutter. Esses trabalhos buscam reduzir as dificuldades enfrentadas por desenvolvedores ao aplicar boas práticas de acessibilidade, seja por meio de ferramentas de análise automática, assistentes inteligentes ou plugins integrados a ambientes de desenvolvimento.

Nesta seção são apresentados três trabalhos que possuem como tema central ou secundário a avaliação de acessibilidade digital, destacando suas principais contribuições, limitações e de que forma cada um deles se relaciona com o presente estudo, servindo de base e inspiração para o desenvolvimento da ferramenta proposta.

3.1 Um estudo sobre a avaliação automática de diretrizes de acessibilidade para dispositivos móveis.

O trabalho apresentado por Silva e Eler [10] aborda a avaliação automática de diretrizes de acessibilidade em aplicações móveis, destacando os desafios existentes na aplicação de recomendações de acessibilidade nesse contexto. Os autores partem da constatação de que, embora a acessibilidade digital seja amplamente discutida no contexto web, sua adoção em aplicações móveis ainda enfrenta limitações técnicas e metodológicas, especialmente no que diz respeito à automação de testes.

O estudo realiza uma análise detalhada das principais diretrizes de acessibilidade aplicáveis a dispositivos móveis, investigando quais critérios podem ser avaliados de forma automática, parcial ou não automatizada. Para isso, os autores conduzem um levantamento bibliográfico, analisam ferramentas existentes no ecossistema Android, como Android Lint, Espresso, *Robolectric*, PUMA e *Accessibility Scanner*, complementando a análise com entrevistas realizadas com especialistas na área de acessibilidade.

Os resultados do trabalho indicam que apenas uma parcela das diretrizes de acessibilidade pode ser totalmente verificada de maneira automática, enquanto muitas dependem de avaliação humana, principalmente aquelas relacionadas à clareza semântica, à compreensão do conteúdo e à adequação de descrições textuais. O artigo evidencia que ferramentas automatizadas são eficazes para identificar problemas objetivos, como ausência de rótulos ou baixo contraste, mas apresentam limitações ao avaliar aspectos subjetivos da experiência do usuário.

A principal contribuição do trabalho está na classificação das propriedades de acessibilidade em três categorias, as automatizáveis, as parcialmente automatizáveis e as não automatizáveis, fornecendo uma visão clara das capacidades e limitações das abordagens automáticas existentes. Essa classificação auxilia desenvolvedores e pesquisadores a compreenderem até que ponto a automação pode apoiar o processo de validação da acessibilidade em aplicações móveis.

A abordagem apresentada por Silva e Eler [10] contribui diretamente para este trabalho ao evidenciar a importância de ferramentas de apoio à avaliação de acessibilidade móvel, bem como a necessidade de integração entre análises automáticas e avaliações manuais. Os resultados reforçam a relevância de soluções que auxiliem desenvolvedores durante o processo de desenvolvimento, alinhando-se ao objetivo deste estudo de promover a criação de aplicações móveis mais acessíveis.

3.2 Avaliação de Interfaces de Usuário voltada à Acessibilidade em Dispositivos Móveis: Boas práticas para experiência de usuário.

O trabalho desenvolvido por Guimarães e Tavares [11] investiga a avaliação de interfaces de usuário sob a perspectiva da acessibilidade em dispositivos móveis, com ênfase na experiência do usuário. As autoras partem da constatação de que grande parte das aplicações móveis apresenta barreiras de acessibilidade que dificultam ou inviabilizam seu uso por pessoas com deficiência, evidenciando a necessidade de métodos de avaliação que considerem simultaneamente o contexto móvel e os princípios de acessibilidade.

O estudo propõe um modelo de avaliação de acessibilidade para interfaces móveis que integra aspectos de usabilidade, acessibilidade e experiência do usuário, levando em consideração as particularidades dos dispositivos móveis, como portabilidade, contexto de uso e diversidade de perfis de usuários. Diferentemente de abordagens tradicionais focadas apenas na conformidade com diretrizes, o modelo enfatiza a observação da experiência real dos usuários durante a interação com as aplicações.

Como parte da validação do modelo proposto, as autoras conduzem experimentos com usuários com deficiência auditiva, utilizando aplicações móveis voltadas à tradução para Libras. Os resultados obtidos evidenciam desafios relevantes no processo de avaliação, como a adequação dos instrumentos de coleta de dados, a necessidade de avaliadores com conhecimento em acessibilidade e a limitação de métodos tradicionais quando aplicados ao contexto móvel. Além disso, o estudo aponta que muitas aplicações acessíveis tornam o

usuário um agente passivo, com pouca interação efetiva com a interface.

A principal contribuição do trabalho está na identificação de lacunas nos métodos existentes de avaliação de acessibilidade, especialmente no que diz respeito à consideração do contexto móvel e da experiência do usuário. O estudo reforça que avaliações baseadas exclusivamente em diretrizes ou ferramentas automáticas são insuficientes para capturar problemas reais enfrentados por usuários com deficiência em dispositivos móveis.

Este trabalho contribui diretamente para o presente estudo ao evidenciar a importância de modelos de avaliação que integrem acessibilidade e experiência do usuário no contexto móvel. As conclusões apresentadas reforçam a necessidade de ferramentas e abordagens que auxiliem desenvolvedores a identificar barreiras de acessibilidade durante o desenvolvimento de aplicações móveis, alinhando-se ao objetivo desta pesquisa de apoiar a criação de interfaces mais acessíveis desde as fases iniciais do projeto.

3.3 Inspeção da Acessibilidade de Aplicativos Móveis Utilizando Software Leitor de Telas.

O trabalho apresentado por Silva e Freire [12] aborda a inspeção da acessibilidade de aplicativos móveis com foco na utilização por pessoas com deficiência visual que fazem uso de softwares leitores de tela. Os autores destacam que, apesar da ampla adoção de diretrizes e técnicas consolidadas para acessibilidade na Web, o contexto de aplicações móveis ainda carece de métodos disseminados e sistemáticos para apoiar o desenvolvimento e a avaliação da acessibilidade.

O estudo enfatiza que a popularização de dispositivos móveis impõe novos desafios ao design de interação, especialmente devido às particularidades desse contexto, como o uso predominante de gestos, a navegação linear mediada por síntese de voz e a diversidade de componentes de interface presentes em aplicações nativas e híbridas. Nesse cenário, os autores ressaltam a importância de considerar as formas reais de interação utilizadas por pessoas com deficiência visual, indo além da simples verificação de conformidade com diretrizes.

Como contribuição principal, o trabalho apresenta um conjunto de técnicas para inspeção de acessibilidade de aplicativos móveis, direcionadas a avaliadores, desenvolvedores e designers. As técnicas são fundamentadas em recomendações de acessibilidade amplamente reconhecidas, como as WCAG 2.1 [8], além de diretrizes específicas para o contexto móvel, e contemplam a análise de diferentes tipos de conteúdo e componentes de interface, incluindo textos, imagens, elementos estruturais, formulários, controles e mídias audiovisuais.

Um aspecto relevante do trabalho é a abordagem prática adotada pelos autores, que inclui demonstrações e simulações do uso de leitores de tela, como o *TalkBack no Android* e o *VoiceOver no iOS*. Essa perspectiva permite que avaliadores sem deficiência visual compreendam melhor as estratégias de navegação e interação utilizadas por esses usuários, auxiliando na identificação de problemas de acessibilidade que não são facilmente detectados por inspeções tradicionais ou ferramentas automáticas.

O trabalho contribui significativamente para o presente estudo ao evidenciar a necessidade de métodos de avaliação que considerem o uso efetivo de tecnologias assistivas no contexto móvel. As técnicas apresentadas reforçam a importância de apoiar desenvolvedores na identificação de barreiras de acessibilidade desde as fases iniciais do desenvolvimento, alinhando-se ao objetivo desta pesquisa de propor uma ferramenta que auxilie na avaliação e na incorporação de práticas de acessibilidade em aplicações móveis.

4. FERRAMENTAS RELACIONADAS

Para apoiar o desenvolvimento deste trabalho, foram analisadas ferramentas especializadas em avaliação de acessibilidade para dispositivos móveis. Nesse contexto, foram selecionadas três instrumentos específicos: *Google Accessibility Scanner*¹, *axe DevTools for Mobile*² e *Evinced Mobile*³. Estas ferramentas foram escolhidas por sua ampla adoção no mercado e capacidade de realizar análises automatizadas que auxiliam os desenvolvedores na identificação de problemas relacionados à acessibilidade.

4.1 Google Accessibility Scanner

O *Google Accessibility Scanner* é uma ferramenta destinada a desenvolvedores de aplicativos Android, que permite identificar de forma automática problemas de acessibilidade [13]. Ele analisa a interface do usuário e sugere melhorias em elementos que não possuem *labels* adequados para leitores de tela, botões com área de toque reduzida, baixo contraste de cores, textos cortados ou elementos fora da hierarquia de navegação esperada.

Uma das vantagens do *Scanner* é a simplicidade de uso, pois não exige configuração complexa e pode ser utilizado diretamente no dispositivo ou em emuladores. Além disso, gera sugestões práticas de ajustes, auxiliando na redução do tempo necessário para a identificação de problemas comuns de acessibilidade. Entretanto, suas recomendações são limitadas a verificações automáticas e não consideram nuances de interação do usuário, como fluxos complexos ou gestos específicos, o que reforça a necessidade de testes complementares com usuários reais.

4.2 axe DevTools for Mobile

O *axe DevTools for Mobile* é uma ferramenta de auditoria de acessibilidade que oferece análises detalhadas tanto para aplicativos móveis quanto para páginas web responsivas [14]. A ferramenta inspeciona os elementos da interface, identificando barreiras como ausência de descrição textual em imagens, problemas de contraste, estruturas inadequadas de *heading* e componentes interativos de difícil acesso.

Entre suas principais vantagens está a geração de relatórios estruturados e detalhados, que podem ser integrados a pipelines de CI/CD, possibilitando o monitoramento contínuo da acessibilidade durante o ciclo de desenvolvimento. Por outro lado, o *axe DevTools* exige maior conhecimento técnico para

¹<https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor>

²<https://docs.deque.com/devtools-mobile/2025.7.2/en/intro>

³<https://www.evinced.com/products/flow-analyzer-for-mobile>

a correta interpretação dos resultados e implementação das correções sugeridas. Além disso, sua análise apresenta limitações em aplicativos nativos mais complexos que utilizam gestos avançados ou interações altamente personalizadas.

4.3 Evinced Mobile

O *Evinced Mobile* é uma ferramenta avançada de análise de acessibilidade para aplicações nativas e híbridas (*iOS* e *Android*). Ela utiliza algoritmos de inteligência artificial para detectar barreiras de acessibilidade que ferramentas de análise estática tradicionais podem ignorar, como a ordem de foco em componentes personalizados e a hierarquia de elementos complexos [15].

A ferramenta permite capturar instantâneos (*snapshots*) da interface durante a execução do aplicativo, gerando relatórios que destacam violações das diretrizes WCAG. Sua principal vantagem é a capacidade de identificar problemas em componentes altamente customizados, comuns em *frameworks* como o *Flutter*. Entretanto, o *Evinced* exige a integração de um SDK ao projeto ou a conexão com um ambiente de execução emulado, o que demanda tempo de configuração inicial. Essa característica reforça a importância de ferramentas de análise de código-fonte, como o *Flutter Access*, que permitem obter diagnósticos rápidos sobre a estrutura do projeto sem a necessidade de instrumentação complexa ou execução dinâmica da aplicação.

A Tabela 1 apresenta uma síntese comparativa entre as ferramentas analisadas e a proposta deste trabalho. O objetivo desta comparação é destacar a complementaridade entre as abordagens, evidenciando como o *Flutter Access* preenche lacunas de agilidade no ciclo de desenvolvimento.

Table 1: Comparativo de Ferramentas

Ferramenta	Análise	Momento	Emulador
Google Scanner	Dinâmica	Execução	Sim
axe DevTools	Dinâmica	CI/CD	Sim
Evinced	Dinâmica	Execução	Sim
Flutter Access	Estática	Escrita	Não

Como observado, enquanto as soluções existentes focam na robustez da análise dinâmica pós-implementação, o *Flutter Access* atua de forma preventiva. A principal diferença reside na detecção de erros durante a escrita do código, o que reduz o custo de retrabalho e dispensa a necessidade de ambientes de execução complexos para validações estruturais.

5. FUNDAMENTAÇÃO TEÓRICA

5.1 Acessibilidade Digital

A acessibilidade digital consolidou-se como um dos pilares fundamentais para a construção de ambientes virtuais inclusivos, assegurando que todas as pessoas, independentemente de suas limitações físicas, sensoriais, cognitivas ou tecnológicas, possam interagir, compreender e utilizar conteúdos e serviços online [16]. A crescente digitalização das atividades sociais, profissionais e institucionais intensificou a necessidade de inclusão equitativa no cenário virtual, situando a acessibilidade como um tema que ultrapassa barreiras téc-

nicas e alcança dimensões sociais, éticas e de justiça informacional [16].

O avanço de recursos como interfaces responsivas, tecnologias assistivas, leitores de tela e legendas automáticas contribui notavelmente para um ecossistema digital mais universal. No entanto, apesar dos progressos observados, a literatura aponta que uma parcela significativa de sites, aplicativos e sistemas ainda não está em conformidade com diretrizes de acessibilidade, o que mantém a exclusão digital como realidade cotidiana para muitos usuários [16]. Tal cenário revela uma coexistência entre desenvolvimento tecnológico e desafios estruturais de implementação, reforçando a necessidade de políticas contínuas e fiscalização sobre padrões inclusivos.

Dentro do debate teórico, destacam-se conceitos como usabilidade, design universal e tecnologias assistivas. A usabilidade refere-se à experiência intuitiva e funcional promovida ao usuário; o design universal propõe soluções já concebidas para atender a múltiplos perfis de interação; e as tecnologias assistivas funcionam como facilitadoras para superar barreiras de comunicação e navegação. Esses elementos, quando integrados, fortalecem a construção de espaços digitais que acolham diferentes tipos de usuários e necessidades [16].

O desenvolvimento histórico da acessibilidade digital também demonstra transformação significativa. Desde os primeiros anos da internet, quando não havia padronização mínima, até a consolidação das Diretrizes de Acessibilidade para Conteúdo Web (WCAG), evidencia-se uma evolução contínua em torno do compromisso social e jurídico da inclusão digital. As discussões atuais avançam para além do cumprimento normativo, alcançando perspectivas de impacto social, responsabilidade corporativa e aprimoramento constante das interações humanas mediadas por tecnologia [16].

Dessa forma, a acessibilidade digital não deve ser entendida como um acréscimo opcional ao processo de desenvolvimento, mas como requisito fundamental para o exercício pleno da cidadania digital. Nesse sentido, compreender suas implicações e desafios é essencial para aprimorar práticas, fortalecer regulamentações e ampliar o alcance da inclusão. Nesse sentido, o referencial teórico evidencia a urgência de integrar tais diretrizes aos modelos de desenvolvimento digital, reforçando a responsabilidade social de profissionais, instituições e governos no compromisso de construir ambientes mais justos e acessíveis para todos [16].

Diante desse cenário geral, torna-se necessário compreender como tais princípios se materializam no contexto de maior uso atual: os dispositivos móveis.

5.2 Acessibilidade em Dispositivos Móveis

A acessibilidade móvel refere-se à garantia de que conteúdos e serviços digitais sejam acessíveis e utilizáveis por todas as pessoas por meio de dispositivos móveis, considerando limitações diversas (visuais, auditivas, motoras, cognitivas e de usabilidade), assegurando inclusão, autonomia e equidade no acesso à informação [17].

Diante de tal cenário, mesmo com a consolidação dos dispo-

itivos móveis como principal meio de acesso à informação digital, persistem barreiras que dificultam a navegação e comprometem a experiência do usuário. Entre essas barreiras estão falhas de adaptação de layout, páginas não responsivas, elementos de interação pouco claros, baixa legibilidade e inadequações estruturais para telas pequenas, o que impede ou limita o uso pleno de conteúdos e aplicativos [17].

Para mitigar esses obstáculos, a documentação oficial de diretrizes de acessibilidade para interfaces móveis disponibilizada pelo Android propõe um conjunto de boas práticas fundamentais [18]. Entre as recomendações estão:

- Garantir contraste adequado entre texto e fundo, bem como tamanho de fonte legível, favorecendo a percepção visual e a leitura;
- Assegurar que elementos interativos possuam área de toque mínima de 48 dp × 48 dp, facilitando a interação por pessoas com mobilidade reduzida;
- Fornecer descrições claras em elementos visuais e botões (labels ou content descriptions), permitindo correta interpretação por leitores de tela e tecnologias assistivas;
- Oferecer alternativas a gestos complexos, como permitir que ações feitas por *swipe* possam também ser executadas por botões acessíveis;
- Evitar depender exclusivamente de cor como elemento transmissor de informação, integrando ícones, texto e outros recursos perceptivos.

Ao relacionar as barreiras apontadas com às recomendações técnicas da documentação do Android, evidencia-se que a acessibilidade móvel demanda tanto compreensão social e inclusiva quanto rigor no design e desenvolvimento [19]. A integração de tais perspectivas é essencial para que aplicativos e interfaces sejam efetivamente utilizáveis por todos, considerando diferenças de capacidades, contextos e dispositivos. Assim, para além das boas práticas de interface, é fundamental compreender quais entidades definem padrões universais que orientam tais recomendações: o W3C.

5.3 Padrões de Acessibilidade

O World Wide Web Consortium (W3C) é uma organização internacional sem fins lucrativos voltada ao desenvolvimento de padrões abertos para a Web, reunindo organizações com membros, corpo técnico e comunidade global para definir tecnologias interoperáveis e acessíveis [20]. A missão do W3C é “fazer a Web funcionar para todos” (“*make the web work for everyone*”), o que reflete seu compromisso com inclusão, diversidade, acessibilidade, internacionalização, segurança e privacidade [21].

Dentro do W3C, a *Web Accessibility Initiative* (WAI) tem papel central na promoção da acessibilidade digital. A WAI desenvolve diretrizes, recomendações, ferramentas e recursos educacionais com o objetivo de garantir que pessoas com diferentes tipos de deficiência tenham acesso equitativo a conteúdos, serviços e aplicações na Web, sem depender de

hardware, software, idioma, localização ou capacidade específica [22].

Assim, o W3C e a WAI fornecem a base normativa e técnica para que a inclusão digital seja efetivada, promovendo padrões, como guidelines, recomendações e boas práticas, com o objetivo de que desenvolvedores e autores de conteúdo construam interfaces que atendam os princípios de acessibilidade, usabilidade e universalidade. A adesão a esses padrões é essencial para assegurar que em ambientes tecnológicos se permaneça um espaço aberto e acessível para todos os usuários, de forma democrática e sustentável. Entre os padrões propostos, destacam-se as WCAG, consolidando-se como referência normativa e técnica de acessibilidade digital.

As *Web Content Accessibility Guidelines* (WCAG) são diretrizes desenvolvidas pelo W3C por meio da Iniciativa de Acessibilidade na Web (WAI), com o propósito de estabelecer requisitos normativos e boas práticas para tornar conteúdos digitais acessíveis a pessoas com diferentes limitações sensoriais, motoras, cognitivas e tecnológicas [20]. Essas diretrizes não se limitam à web tradicional, abrangendo também interfaces móveis, documentos digitais, aplicativos e qualquer meio que represente informação por tela ou áudio.

As WCAG organizam seus fundamentos em quatro princípios centrais, conhecidos pelo acrônimo “POUR” (*Perceivable, Operable, Understandable, Robust*). Esses princípios orientam não apenas o produto final, mas todo o ciclo de desenvolvimento, desde arquitetura de informação, design de interação, prototipagem e implementação, até testes com tecnologias assistivas. A Tabela 2 apresenta a descrição detalhada de cada um desses conceitos:

Table 2: Princípios da WCAG

Princípio	Descrição
Perceptível (Perceivable)	Informações e elementos da interface devem ser apresentados de forma perceptível, incluindo alternativas textuais, legendas, contraste adequado e estrutura compreensível. [8]
Operável (Operable)	Os componentes e a navegação devem ser plenamente operáveis, sem exigir gestos complexos ou ações dependentes de tempo e garantindo interação por múltiplos dispositivos de entrada. [8]
Compreensível (Understandable)	O conteúdo e a operação devem ser claros e previsíveis, com linguagem acessível, rótulos adequados e consistência na navegação. [8]
Robusto (Robust)	O conteúdo deve ser interpretável por tecnologias assistivas e navegadores variados, mantendo compatibilidade futura por meio de marcação correta e padrões acessíveis. [8]

Além da estrutura conceitual, as WCAG apresentam critérios de sucesso objetivos e verificáveis, que são agrupados em três níveis de conformidade, conforme descrito a seguir:

- **A:** estabelece os requisitos básicos e essenciais de acessibilidade. Sem o cumprimento deste nível, o conteúdo torna-se praticamente impossível de ser acessado por grande parte dos usuários com deficiência.

- **AA:** aborda as barreiras mais comuns e é o padrão internacionalmente adotado como referência para ambientes públicos, governamentais e educacionais, por equilibrar exigências técnicas e viabilidade de implementação.
- **AAA:** corresponde ao nível máximo de conformidade, exigindo melhorias mais complexas e específicas. É frequentemente aplicado em contextos institucionais especializados, embora nem todo conteúdo consiga atingir este nível em sua totalidade.

As diretrizes também se consolidaram como instrumento normativo de grande impacto global, sendo utilizadas como referência em políticas nacionais e internacionais de inclusão digital, certificações de acessibilidade e diretrizes institucionais de design universal. Sua adoção contribui para garantir interoperabilidade com tecnologias assistivas, ampliar o alcance do conteúdo e promover equidade de acesso, sobretudo em ecossistemas móveis, onde há maior diversidade de dispositivos de entrada, tamanhos de tela e modos de navegação [23].

Assim, a WCAG não atua apenas como um documento de recomendações técnicas, mas como uma base estruturante de acessibilidade digital, orientando a construção de experiências inclusivas, consistentes e sustentáveis no tempo, alinhadas aos padrões oficiais de acessibilidade propostos pelo W3C.

5.4 Acessibilidade no Brasil

No Brasil, a acessibilidade digital é reconhecida como uma prioridade institucional para garantir que toda a população, independentemente de suas capacidades físicas, sensoriais ou cognitivas, tenha acesso a conteúdos e serviços públicos online de forma equitativa. O portal oficial de “Acessibilidade Digital” do governo federal afirma que a acessibilidade digital busca eliminar barreiras na Web, promovendo democratização do acesso, inclusão social e garantia de participação cidadã. [6]

Para regular essa demanda, foi criado o *Modelo de Acessibilidade em Governo Eletrônico* (e-MAG), que estabelece um conjunto de padrões e recomendações obrigatórias para portais e serviços públicos eletrônicos no âmbito federal [24]. O e-MAG procura padronizar a implementação da acessibilidade digital no Brasil, garantindo conformidade com padrões internacionais, como as diretrizes da World Wide Web Consortium (W3C), ao mesmo tempo em que adapta tais recomendações às especificidades nacionais [24].

O e-MAG, em sua versão 3.1 (publicada originalmente em 2014), define que os sites do governo devem adotar mecanismos de acessibilidade como: alternativas de navegação via teclado, barra de acessibilidade com controle de contraste e tamanho de fonte, estrutura semântica adequada, compatibilidade com leitores de tela e outras tecnologias assistivas, de modo a permitir o acesso universal ao conteúdo, independentemente do dispositivo utilizado ou da condição do usuário [24].

Essa padronização institucional é significativa, tendo em vista que no Brasil, uma parcela importante da população

apresenta alguma deficiência ou limitação. A adoção do e-MAG representa um esforço para tornar os serviços públicos acessíveis de forma estruturada e permanente, indo além de iniciativas pontuais ou dependentes de cada órgão.

6. METODOLOGIA

Para alcançar os objetivos propostos neste trabalho, foi adotada uma abordagem metodológica de caráter exploratório e qualitativo, composta por duas etapas principais: revisão bibliográfica e aplicação de questionário com potenciais usuários e desenvolvedores. Essa combinação permitiu compreender os fundamentos teóricos da acessibilidade digital, por meio de uma revisão bibliográfica detalhada das normas e diretrizes vigentes, bem como analisar a percepção prática de profissionais sobre a acessibilidade em aplicações móveis.

Inicialmente, foi realizada uma revisão *ad hoc*, sem a aplicação de critérios metodológicos. Esta etapa consistiu em uma busca exploratória por artigos científicos e documentos técnicos utilizando descritores como acessibilidade digital, acessibilidade em dispositivos móveis e avaliação automática de acessibilidade. A partir dessas buscas, foram identificadas e analisadas as diretrizes do W3C (especialmente as WCAG) e o Modelo de Acessibilidade em Governo Eletrônico (e-MAG) como pilares normativos. O objetivo desta fase foi aprofundar o entendimento sobre conceitos e desafios existentes, além de identificar lacunas para o desenvolvimento de ferramentas de apoio à acessibilidade móvel.

A partir da revisão teórica, foi elaborado um questionário com o intuito de coletar percepções, opiniões e sugestões relacionadas à acessibilidade em dispositivos móveis. O questionário buscou identificar o nível de conscientização sobre o tema, bem como levantar contribuições práticas que pudessem subsidiar decisões de projeto e desenvolvimento da solução proposta neste trabalho. Com base nos resultados desse questionário, foi implementada a solução *Flutter Access* com o objetivo de ajudar a melhorar o processo de desenvolvimento de software mobile sob a perspectiva da acessibilidade digital.

7. ESTUDO EXPLORATÓRIO

O questionário foi aplicado de forma online e obteve um total de 15 participantes. O perfil da amostra foi composto por profissionais que já atuam no mercado de desenvolvimento de software e por estudantes iniciantes na área de tecnologia, interessados no tema de acessibilidade. As perguntas foram formuladas de maneira aberta e fechada, permitindo que os respondentes expressassem livremente suas percepções, experiências e sugestões relacionadas à acessibilidade em dispositivos móveis. O roteiro completo das perguntas aplicadas encontra-se disponível no Apêndice B deste trabalho. Os dados e percepções coletados a partir deste instrumento foram consolidados e utilizados como insumos para o levantamento de requisitos da ferramenta.

A principal questão analisada solicitava que os participantes deixassem observações, contribuições ou sugestões sobre acessibilidade em dispositivos móveis. A análise das respostas evidenciou um consenso quanto à relevância do tema no contexto atual do desenvolvimento de software. Os participantes destacaram que a acessibilidade deve ser considerada um requisito essencial, capaz de facilitar a interação

entre pessoas e sistemas, ampliando o alcance e a usabilidade das aplicações móveis.

Ao serem questionados se acreditavam que um plugin ou ferramenta que verificasse a acessibilidade durante o desenvolvimento seria útil, 100% dos participantes responderam positivamente, indicando uma forte demanda por soluções integradas ao fluxo de trabalho, como ilustra o gráfico da Figura 1.

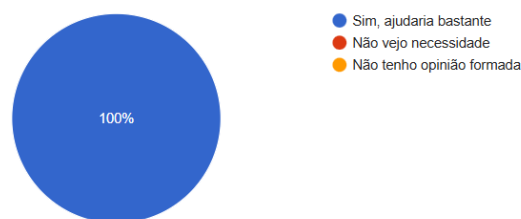


Figure 1: O apoio de ferramentas de verificação de acessibilidade no desenvolvimento de software mobile

Em contrapartida, quando indagados sobre a utilização atual de algum recurso voltado à acessibilidade no desenvolvimento mobile, os dados revelam um cenário de baixa adoção. Conforme mostra o gráfico da Figura 2, a grande maioria (93,3%) afirmou não utilizar nenhum recurso específico, enquanto apenas 6,7% responderam que já fazem uso de alguma ferramenta.

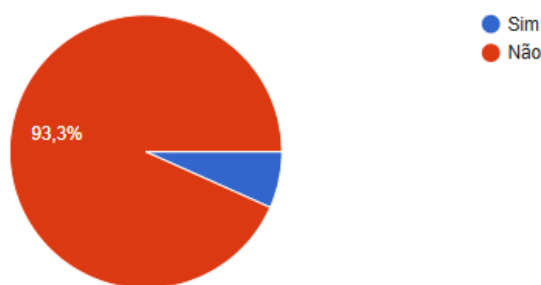


Figure 2: Percepção sobre quem faz a utilização de ferramentas de verificação de acessibilidade durante o desenvolvimento

A análise das respostas abertas evidenciou um consenso quanto à relevância do tema. Quando indagados sobre observações ou sugestões adicionais sobre acessibilidade em dispositivos móveis, alguns participantes ressaltaram o papel dos dispositivos móveis como facilitadores de acesso à informação. Conforme destacado por um dos respondentes: "o dispositivo móvel é um facilitador e, portanto, quanto mais fácil for o uso, melhor".

No mesmo tópico, enfatizou-se a integração entre acessibilidade e Experiência do Usuário (UX), sugerindo-se extrapolar as verificações estritamente técnicas: "seria interessante também considerar aspectos de UX sobre heurísticas de usabilidade, como as Heurísticas de Nielsen". Essa visão

reforça que a acessibilidade deve ser tratada de forma holística.

Além disso, no campo de contribuições livres, surgiram sugestões sobre aspectos específicos, como o daltonismo. Um dos participantes apontou que, "se o daltonismo não estiver sendo considerado, considero que seja uma adição válida para versões futuras do plugin", reforçando a necessidade de ampliar o escopo das ferramentas para além das barreiras visuais tradicionais.

A partir dessa análise, identificou-se que as contribuições corroboram a necessidade de ferramentas que auxiliem desenvolvedores na mitigação de problemas de acessibilidade. Essas percepções foram fundamentais para a fundamentação deste trabalho, especialmente no que se refere à integração entre acessibilidade e usabilidade.

8. O PROJETO

Esta seção tem como finalidade apresentar a ferramenta desenvolvida, elaborado com o propósito de abordar aspectos relacionados à acessibilidade no desenvolvimento de software mobile.

8.1 A Ferramenta

O Flutter Access é uma ferramenta voltada à análise estática de código-fonte, desenvolvida com o objetivo de auxiliar na identificação automática de potenciais problemas de acessibilidade em projetos Flutter. A escolha por este framework justifica-se por sua crescente adoção no desenvolvimento multiplataforma, permitindo a criação de interfaces consistentes para Android e iOS a partir de um único código-base.

Além disso, o Flutter possui um motor de renderização próprio e uma árvore de semântica dedicada (*Semantics Tree*), o que segundo Napier [25], oferece aos desenvolvedores um controle granular sobre como os elementos da interface são expostos às tecnologias assistivas, tornando-o um ecossistema estratégico para a implementação de automações de verificação de acessibilidade.

A Figura 3 apresenta a ferramenta integrada ao Visual Studio Code como uma extensão, possibilitando sua utilização diretamente no ambiente de desenvolvimento. A extensão está disponível gratuitamente para instalação, permitindo a realização de testes práticos e a validação de acessibilidade em projetos Flutter. A visualização completa da interface integrada pode ser consultada no apêndice A.1.

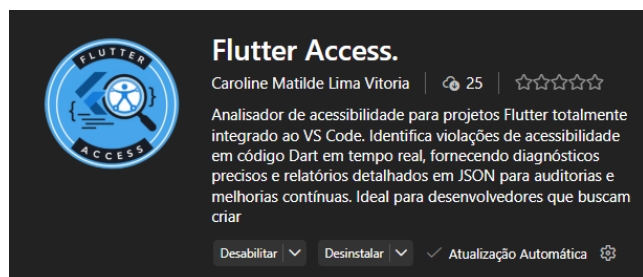


Figure 3: Extensão da ferramenta integrada ao VsCode

O diferencial estratégico do Flutter Access reside em sua capacidade de antecipar a correção de falhas. Diferentemente de abordagens tradicionais baseadas em testes manuais ou avaliações pós-implementação, a ferramenta integra-se diretamente ao fluxo de trabalho do desenvolvedor. Essa abordagem permite a identificação e mitigação de problemas de acessibilidade ainda durante a fase de escrita do código-fonte. Ao atuar de forma preventiva, o Flutter Access reduz drasticamente o custo de retrabalho e garante que a acessibilidade seja tratada como um requisito técnico contínuo, e não como uma etapa final e isolada do desenvolvimento.

A partir da análise realizada, o Flutter Access gera relatórios detalhados contendo os problemas de acessibilidade identificados no código-fonte. O relatório é estruturado em formato JSON (*JavaScript Object Notation*), escolha motivada pela sua leveza e pela familiaridade que desenvolvedores já possuem com esse padrão, o que reduz a curva de aprendizado para interpretar os dados.

Além disso, o arquivo é gerado diretamente na pasta raiz do projeto analisado, facilitando o acesso imediato sem a necessidade de ferramentas externas ou portais específicos. Essa estrutura de intercâmbio de dados permite que os resultados sejam consumidos tanto por humanos quanto por máquinas, simplificando a integração em pipelines de Integração e Entrega Contínua (CI/CD) para a automação de verificações de qualidade.

A Figura 4 exemplifica um relatório gerado após a análise de um projeto Flutter.

```
{
  "gerado_em": "2026-06-09T21:32:50.940768",
  "problemas": [
    {
      "regra": "ContrastRule",
      "mensagem": "Este texto apresenta um baixo contraste (proporção 1.00:1). O padrão WCAG exige no mínimo 4.5:1 para garantir a legibilidade. Sugestão: Aumente o contraste entre a cor do texto e o fundo (assumindo fundo branco neste cálculo). (assento_page.dart:91:26)",
      "arquivo": "c:/Users/Carol/meufilmeapp/lib/pages/assento_page.dart"
    },
    {
      "regra": "FocusRule",
      "mensagem": "O widget interativo \"InkWell\" está sem um \"FocusNode\" definido. Isso impossibilita ou dificulta a navegação por teclado ou outros dispositivos de entrada. Sugestão: Implemente e associe um \"FocusNode\" para garantir que o elemento possa ser focado. (assento_page.dart:73:22)",
      "arquivo": "c:/Users/Carol/meufilmeapp/lib/pages/assento_page.dart"
    }
  ],
  "resumo": {
    "quantidade_de_violacoes": 2,
    "quantidade_de_regras_violadas": 2
  }
}
```

Figure 4: Exemplo de relatório gerado após a análise de um projeto Flutter.

No exemplo apresentado, observa-se a organização das informações em campos específicos para facilitar a depuração:

- **gerado_em:** Registra a data e o horário exatos da conclusão da análise, permitindo o rastreamento histórico e auditorias de acessibilidade.

- **problemas:** Uma lista de objetos onde cada item detalha uma falha. O campo regra identifica a norma técnica violada, enquanto a mensagem fornece uma descrição didática do erro com sugestões de correção. O campo arquivo indica o caminho do código-fonte onde o problema foi localizado.
- **resumo:** Representa o objeto que consolida as métricas da varredura. Na Figura 4, o campo `quantidade_de_violacoes` mostra a quantidade de problemas encontrados, distribuídos entre a quantidade de regras distintas representado pela quantidade de regras violadas.

Um exemplo de relatório completo pode ser visualizado no apêndice A.2.

Essa abordagem garante que o desenvolvedor tenha uma visão sistêmica e prática do projeto, permitindo priorizar as correções de forma eficiente e garantindo que as interfaces sejam compatíveis com tecnologias assistivas.

Além do relatório final, a ferramenta também apresenta os apontamentos diretamente no terminal, permitindo feedback imediato ao desenvolvedor durante a execução da análise. As mensagens exibidas são geradas por um motor de análise estática que utiliza o pacote *analyzer* do Dart. Este motor realiza o *parsing* do código-fonte para construir uma Árvore de Sintaxe Abstrata (*Abstract Syntax Tree* — AST), permitindo que o *Flutter Access* identifique padrões que violam as regras de acessibilidade implementadas. Cada alerta no terminal é estruturado para fornecer o diagnóstico do problema, seu impacto na experiência do usuário e uma sugestão de correção.

Entre as principais mensagens de erro que podem ser disparadas pela ferramenta, destacam-se:

- **Botões sem descrição:** Disparada quando *widgets* como *ElevatedButton* ou *IconButton* carecem de um tooltip ou rótulo semântico, impossibilitando que leitores de tela identifiquem a ação do botão.
- **Imagens sem rótulo:** Ocorre quando o widget *Image* não possui a propriedade *semanticLabel*, deixando usuários com deficiência visual sem o contexto da imagem.
- **Falta de semântica em textos:** Alertada quando o widget *Text* não possui um *semanticsLabel* ou um *style* definido, o que pode dificultar a interpretação por tecnologias assistivas em interfaces complexas.
- **Campos de formulário sem identificação:** Mensagens que apontam a ausência de *labelText* ou *hintText* em componentes de entrada, como *TextField*, impedindo a compreensão do propósito do campo.

Dessa forma, a ferramenta contribui para a incorporação da acessibilidade desde as fases iniciais do ciclo de vida do software, favorecendo a criação de aplicações mais inclusivas e alinhadas com padrões internacionais.

8.2 Tecnologias

Conforme ilustrado na Figura 5, a arquitetura do Flutter é composta por camadas independentes. O Flutter Access atua especificamente na camada de Framework, onde se concentram as definições de *Widgets* e a lógica de interface em Dart. Por ser uma estrutura modular, é possível interceptar e analisar essas definições de forma estática, garantindo a validação de critérios de acessibilidade antes mesmo que o código seja processado pela Engine de renderização, o que otimiza o fluxo de desenvolvimento ao evitar execuções desnecessárias.

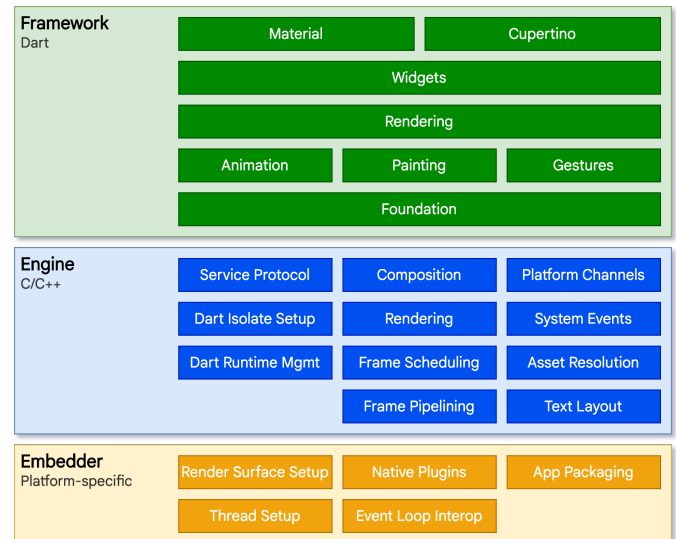


Figure 5: Camadas da arquitetura do framework Flutter [26]).

A base técnica da ferramenta utiliza o pacote oficial *analyzer* da linguagem Dart [27], responsável por interpretar a estrutura sintática e semântica do código-fonte. Esse pacote possibilita a inspeção detalhada de elementos como *widgets*, atributos e declarações, fornecendo os recursos necessários para a aplicação das regras de acessibilidade definidas no projeto.

Para exemplificar, considere um trecho de código Flutter onde um botão é declarado sem o atributo *tooltip*:

```
IconButton(
  icon: Icon(Icons.share),
  onPressed: () => shareContent(),
)
```

Ao processar esse código, a regra *ButtonRule* utiliza o padrão *Visitor* para percorrer a AST. O trecho abaixo demonstra como a ferramenta identifica a ausência do atributo obrigatório acessando os metadados do código:

```
@override
void visitInstanceCreationExpression(
  InstanceCreationExpression node) {
  final buttonNames = ['ElevatedButton', '
    IconButton', 'TextButton'];
  final className = node.constructorName.type.
    name2.lexeme;
```

```

if (buttonNames.contains(className)) {
  final hasTooltip = node.argumentList.
    arguments
    .any((arg) => arg.toSource().contains('
      tooltip'));

  if (!hasTooltip) {
    errors.add('Botão sem tooltip detectado
      em: ${node.toSource()}');
  }
}
super.visitInstanceCreationExpression(node);
}

```

Dessa forma, a ferramenta consegue apontar o erro de acessibilidade e sua localização exata sem que o desenvolvedor precise executar a aplicação, integrando-se de forma eficiente ao fluxo de desenvolvimento.

8.2.1 Dart Analyzer

O Dart Analyzer é uma ferramenta oficial do ecossistema Dart responsável pela análise estática de código-fonte. Ele permite a construção e navegação da Árvore de Sintaxe Abstrata (AST), possibilitando a identificação precisa de componentes, propriedades e estruturas utilizadas nos projetos Flutter.

No contexto do Flutter Access, o analyzer é utilizado como base para a implementação das regras de acessibilidade, permitindo a verificação de aspectos como uso adequado de semântica, presença de rótulos descritivos, contraste de cores, focado nas propriedades declaradas de forma estática no código e limitando-se a analisar os valores definidos explicitamente, além de tamanhos de áreas interativas e acessibilidade em elementos multimídia. Essa análise detalhada viabiliza a detecção de violações às boas práticas antes da execução do aplicativo.

8.3 Arquitetura da Ferramenta

A arquitetura do *Flutter Access* é fundamentada em um diagrama de componentes que ilustra a interação entre o núcleo de análise e o ecossistema Dart. A Figura A.4, localizada no Apêndice A, evidencia essa estrutura, destacando a integração com o *Dart Analyzer* para a obtenção da Árvore de Sintaxe Abstrata (AST).

O componente *Flutter Access* atua como o orquestrador do sistema. Ele recebe a AST e submete o código a um motor de análise composto por regras que derivam da interface *AccessibilityRule*. Cada regra concreta utiliza o padrão *Visitor* para percorrer os nós da árvore e identificar padrões que violem as diretrizes de acessibilidade. O contrato comum a todas as implementações é definido pela classe abstrata abaixo:

```

abstract class AccessibilityRule {
  String get name;
  List<RuleViolation> check(CompilationUnit
    compilationUnit, String filePath);
}

```

A execução desse processo fica a cargo da classe *FlutterAccessibilityAnalyzer*, a qual itera sobre o conjunto de regras registradas. Quando uma violação é detectada pelo método

check, as informações de localização (arquivo, linha e coluna) são encapsuladas em instâncias de *AccessibilityIssue*, que servem de base para a geração dos relatórios finais, conforme demonstrado no código a seguir:

```

class AccessibilityIssue {
  final String ruleName;
  final String message;
  final String filePath;
  final int? line;
  final int? column;

  AccessibilityIssue({
    required this.ruleName,
    required this.message,
    required this.filePath,
    this.line,
    this.column,
  });

  @override
  String toString() {
    final loc = line != null ? ':$line$' : ''
      column != null ? ':$column' : '';
    return '$ruleName - $message ($filePath$loc)';
  }
}

class FlutterAccessibilityAnalyzer {
  final List<AccessibilityRule> rules;

  FlutterAccessibilityAnalyzer({required this.rules});

  List<AccessibilityIssue> analyzeResolvedUnit(
    ResolvedUnitResult result,
    [String? filePath]) {
    final unit = result.unit;
    final issues = <AccessibilityIssue>[];

    for (var rule in rules) {
      final errors = rule.check(unit, filePath
        ?? 'arquivo desconhecido');
      for (var err in errors) {
        issues.add(AccessibilityIssue(
          ruleName: rule.name,
          message: err.message,
          filePath: filePath ?? 'arquivo
            desconhecido',
          line: err.line,
          column: err.column,
        ));
      }
    }
    return issues;
  }
}

```

A aplicação prática dessa lógica em uma verificação concreta é exemplificada na regra *SemanticsRule*, cujo comportamento são detalhados no código abaixo:

```

class SemanticsRule extends AccessibilityRule {
  @override
  String get name => 'SemanticsRule';

  @override
  List<RuleViolation> check(dynamic
    compilationUnit, String filePath) {
    final errors = <RuleViolation>[];
    final visitor = _SemanticsVisitor(errors,
      compilationUnit, filePath);
  }
}

```

```

        compilationUnit.accept(visitor);
        return errors;
    }
}

class _SemanticsVisitor extends
    RecursiveAstVisitor {
    final List<RuleViolation> errors;
    final CompilationUnit compilationUnit;
    final String filePath;

    _SemanticsVisitor(this.errors, this.
        compilationUnit, this.filePath);

    @override
    void visitInstanceCreationExpression(
        InstanceCreationExpression node) {
        final typeName = node.constructorName.type;
        if (typeName is NamedType && typeName.name2
            .lexeme == 'Semantics') {
            bool hasLabelOrValueOrHint = false;
            for (var arg in node.argumentList.
                arguments) {
                if (arg is NamedExpression) {
                    final name = arg.name.label.name;
                    if (name == 'label' || name == 'value'
                        || name == 'hint') {
                        if (arg.expression.toSource().trim
                            ().isEmpty) {
                            hasLabelOrValueOrHint = true;
                        }
                    }
                }
            }
        }
    }
}

}

    }
    if (!hasLabelOrValueOrHint) {
        final loc = compilationUnit.lineInfo.
            getLocation(node.offset);
        errors.add(RuleViolation(
            message: 'O widget "Semantics" foi
                utilizado, mas não possui "
                label", "value" ou "hint". Sem
                um desses campos, o widget não
                fornece nenhuma informação útil
                para tecnologias assistivas.
                Sugestão: Defina a propriedade
                "label" com uma descrição curta
                e clara do elemento.',
            line: loc.lineNumber,
            column: loc.columnNumber));
    }
    super.visitInstanceCreationExpression(node);
}
}

```

As 15 regras implementadas no *Flutter Access* foram extraídas das recomendações da *WCAG 2.1*, dos guias de acessibilidade do *Android* e da documentação oficial do *Flutter*. A Tabela 3 detalha cada verificação, agrupando-as conforme a natureza do impacto na experiência do usuário, servindo como uma síntese descritiva das funcionalidades da ferramenta.

Table 3: Detalhamento das Regras de Acessibilidade por Eixo

Regra (Classe)	Descrição Detalhada da Verificação Técnica	Fonte
Eixo 1: Percepção de Conteúdo e Multimídia		
<i>TextAccessibilityRule</i>	Analisa se rótulos textuais estão acessíveis e se não estão ocultos para motores de síntese de voz.	W3C 1.3.1
<i>ImageAccessibilityRule</i>	Valida a presença obrigatória do <i>semanticLabel</i> em imagens para compreensão de usuários cegos.	Flutter
<i>IconAccessibilityRule</i>	Assegura que ícones interativos possuam descrições, evitando botões sem contexto para o leitor de tela.	Flutter
<i>MediaAccessibilityRule</i>	Verifica a existência de controles acessíveis e alternativas para conteúdos de áudio e vídeo dinâmicos.	W3C 1.2
<i>InternationalizationRule</i>	Valida o uso de I18n, garantindo que a tecnologia assistiva utilize o idioma correto do usuário.	Android
Eixo 2: Interatividade e Operação		
<i>TapTargetSizeRule</i>	Verifica se alvos de toque possuem área mínima de 48x48 DP para suporte a usuários com dificuldades motoras.	Android
<i>ButtonTooltipRule</i>	Garante que botões possuam <i>tooltips</i> ativos para fornecer feedback visual e suporte cognitivo adicional.	Android
<i>FormFieldLabelRule</i>	Valida se campos de entrada possuem rótulos persistentes que orientam o preenchimento de formulários.	W3C 3.3.2
<i>FocusNodeAccessibilityRule</i>	Gerencia a visibilidade e a lógica de navegação do foco via teclado ou dispositivos de entrada externa.	Android
<i>NavigationFlowRule</i>	Assegura que o fluxo entre telas seja previsível, movendo o foco para o topo em novas rotas.	W3C 3.2.3
Eixo 3: Estrutura Semântica e Hierarquia		
<i>SemanticsWidgetRule</i>	Valida o agrupamento lógico de elementos através do widget básico de semântica do Flutter.	Flutter
<i>AdvancedSemanticsRule</i>	Verifica papéis específicos como <i>headers</i> ou <i>links</i> para definir o comportamento esperado do componente.	Flutter
<i>HeadingHierarchyRule</i>	Organiza e valida a hierarquia de títulos para permitir a escaneabilidade estruturada da página.	W3C 1.3.1
<i>RedundantLabelRule</i>	Identifica descrições repetitivas que causam poluição sonora e prejudicam a fluidez da leitura assistiva.	W3C Mob.
Eixo 4: Design Visual e Legibilidade		
<i>ColorContrastRule</i>	Analisa a relação de contraste entre texto e fundo, exigindo a taxa de 4.5:1 para garantir a leitura.	W3C 1.4.3

8.4 Segurança e Privacidade

Um aspecto fundamental considerado na arquitetura do *Flutter Access* refere-se à segurança e à privacidade do código-fonte analisado. A ferramenta foi projetada para executar todo o processo de análise de forma local, diretamente no ambiente do desenvolvedor, sem a necessidade de comunicação com servidores externos ou serviços em nuvem.

Durante a execução da análise, o *Flutter Access* utiliza exclusivamente recursos locais, como o *Dart Analyzer*, para processar a Árvore de Sintaxe Abstrata (AST) do projeto *Flutter*. Em nenhum momento o código-fonte analisado é armazenado de forma persistente, transmitido pela rede ou compartilhado com terceiros. Todas as informações utilizadas pela ferramenta permanecem restritas ao ambiente

local do usuário, enquanto o código-fonte é descartado da memória ao final da execução, as inconsistências de acessibilidade identificadas são compiladas em um relatório final, permitindo que o desenvolvedor visualize e corrija as falhas apontadas.

Essa abordagem garante total privacidade do código analisado, eliminando riscos relacionados à exposição de informações sensíveis, reutilização indevida do código-fonte ou violação de propriedade intelectual. Tal característica é especialmente relevante em contextos profissionais e acadêmicos, nos quais projetos podem conter dados confidenciais ou soluções proprietárias.

Além disso, o fato de a análise ser realizada de maneira offline contribui para a confiabilidade e a segurança da ferramenta, uma vez que seu funcionamento independe de conexões externas, autenticação em serviços de terceiros ou permissões adicionais além do acesso local ao projeto. Dessa forma, o Flutter Access alinha-se a boas práticas de segurança no desenvolvimento de ferramentas de apoio, oferecendo aos desenvolvedores uma solução segura, confiável e compatível com diferentes ambientes de trabalho.

8.5 Benefícios da Ferramenta

O uso do Flutter Access proporciona diversos benefícios a desenvolvedores, pesquisadores e estudantes, destacando-se a possibilidade de identificar problemas de acessibilidade de forma antecipada. A análise é realizada de forma granular, processando cada arquivo individualmente como uma unidade de compilação independente, o que garante precisão na localização dos erros. Essa flexibilidade permite que o usuário opte por analisar um arquivo específico em desenvolvimento ou execute uma varredura completa por todo o projeto, adaptando-se a diferentes fluxos de trabalho.

Outro diferencial do Flutter Access é o alinhamento das regras de análise com as diretrizes WCAG 2.1 e WAI-ARIA, assegurando conformidade com padrões internacionais. Além disso, a escolha das tecnologias utilizadas possibilita a evolução futura da ferramenta, incluindo sua integração como extensão em ambientes de desenvolvimento, como o Visual Studio Code.

9. VALIDAÇÃO

Esta seção descreve o método utilizado para a realização dos testes e validações da ferramenta *Flutter Access*. O objetivo desta validação foi verificar a capacidade da ferramenta em identificar potenciais problemas de acessibilidade em projetos *Flutter*, bem como avaliar sua aplicação no apoio ao desenvolvimento de aplicações mais acessíveis.

9.1 Prática

A realização da atividade de validação ocorreu durante o período de 1 mês, contando com a participação de dois desenvolvedores de software. Ambos os participantes utilizaram a ferramenta proposta em seus fluxos de trabalho em projetos pessoais de estudo e portfólio, permitindo avaliar seu uso em um contexto real de desenvolvimento.

Previamente ao início da coleta de dados, todos os participantes assinaram o Termo de Consentimento Livre e Es-

clarecido exemplificado no apêndice B.12, garantindo o anonimato e o uso dos dados exclusivamente para fins acadêmicos. A validação foi conduzida em três etapas principais: (i) assinatura do TCLE e alinhamento inicial; (ii) utilização prática e contínua da ferramenta nos projetos por um período de experimentação; e (iii) aplicação do Questionário de Feedback de Uso B.13, focado na experiência real com a ferramenta.

Os relatos coletados confirmaram as dificuldades mapeadas no questionário inicial B.6, no qual 92,3% dos participantes haviam afirmado não utilizar recursos de acessibilidade por escassez de ferramentas integradas. Durante a fase prática, os dois desenvolvedores ratificaram esse cenário, destacando a complexidade em organizar a semântica da interface sem o apoio de uma análise automatizada.

Após a utilização, foi aplicado o questionário de feedback B.13 para coletar as percepções sobre usabilidade e clareza das informações. Quanto à instalação e configuração, os participantes classificaram o processo como claro e conciso. Sobre a percepção de acessibilidade durante a escrita do código, um dos desenvolvedores relatou: *“A utilização da ferramenta alterou minha percepção ao apontar falhas semânticas em componentes que eu considerava prontos. Percebi que muitos erros de acessibilidade são invisíveis sem uma ferramenta de apoio em tempo real.”*

Em relação aos pontos positivos, as respostas destacaram a agilidade da análise estática integrada ao editor, ressaltando que receber alertas de rótulos ausentes durante a codificação evita que erros básicos cheguem às fases avançadas de teste, gerando economia de tempo.

Nesse sentido, tais apontamentos foram fundamentais para validar a efetividade da solução e reforçar a importância de ferramentas que auxiliem o desenvolvedor na incorporação da acessibilidade desde o início do ciclo de vida do software.

9.2 Resultados

Nesta seção são apresentados os resultados obtidos a partir das respostas do questionário de feedback aplicado aos desenvolvedores participantes da validação prática da ferramenta.

Antes de detalhar a percepção dos usuários sobre o ecossistema, foi mapeado o perfil demográfico e técnico da amostra através das duas questões iniciais do formulário. A Figura 6 apresenta a distribuição das respostas referentes ao nível de experiência com o framework *Flutter* (Questão 1), enquanto a Figura 7 ilustra o contexto de aplicação da ferramenta (Questão 2).

A análise descritiva dos gráficos revela que a amostra de validação foi composta por um desenvolvedor de nível Iniciante/Estudante e um de nível Júnior. Quanto ao cenário de experimentação, ambos os participantes concentraram o uso do *Flutter Access* no escopo de projetos acadêmicos e de portfólio pessoal. As demais respostas coletadas foram compiladas textualmente e categorizadas a seguir.

9.2.1 Facilidade de uso

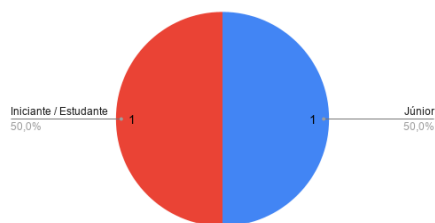


Figure 6: Perfil sociotécnico dos participantes: Experiência com o framework Flutter.

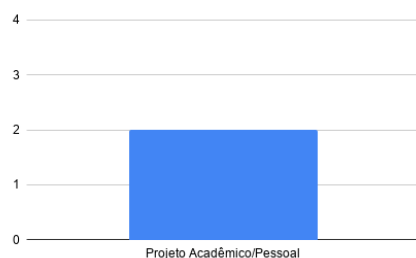


Figure 7: Perfil sociotécnico dos participantes: Contexto de aplicação da ferramenta.

Um dos principais requisitos considerados no desenvolvimento da ferramenta foi a facilidade de uso, buscando atender desenvolvedores com diferentes níveis de experiência sem demandar curva de aprendizado elevada. A interface foi projetada para oferecer clareza na apresentação das informações e integração simples ao fluxo de desenvolvimento. Para validar este requisito, os participantes foram questionados: “A instalação e configuração da ferramenta foi clara e concisa? Se não, descreva as dificuldades enfrentadas” (Questão 3 do Apêndice B.13).

A partir da análise das respostas, observou-se que ambos os desenvolvedores destacaram positivamente a facilidade de uso. Segundo um dos participantes, “a configuração foi extremamente simples, bastando adicionar a dependência no projeto”, o que contribuiu para uma experiência fluida e permitiu que a ferramenta fosse incorporada aos projetos pessoais sem causar interrupções significativas no fluxo de trabalho. A organização das funcionalidades e a clareza das recomendações apresentadas reforçaram a viabilidade do *Flutter Access* como suporte técnico imediato durante a codificação.

9.2.2 Apoio à detecção de falhas de acessibilidade

Outro ponto central avaliado foi a eficácia da ferramenta no suporte ao diagnóstico de barreiras de acessibilidade em aplicações móveis. O foco do trabalho esteve em auxiliar desenvolvedores na detecção de falhas comuns relacionadas a diretrizes de acessibilidade, especialmente aquelas alinhadas às recomendações do WCAG e às boas práticas para dispositivos móveis.

De acordo com os desenvolvedores participantes, a ferramenta mostrou-se útil ao evidenciar aspectos técnicos que, em muitos casos, não eram considerados durante o fluxo de desenvolvimento convencional. Como exemplo prático,

os participantes destacaram a identificação de falhas relacionadas à ausência de rótulos em botões e inadequações na organização semântica da interface. Um dos desenvolvedores mencionou que:

“O plugin alertou para a falta do *semanticsLabel* em diversos elementos interativos, um detalhe que costuma passar despercebido sem uma validação automática.”

Ademais, os relatos indicaram que o uso da ferramenta contribuiu para elevar a conscientização sobre acessibilidade digital, auxiliando no refinamento da estrutura da interface, no uso correto de descrições textuais e na disposição lógica dos elementos interativos.

9.2.3 Contribuição da ferramenta no processo de desenvolvimento

Em relação ao impacto da ferramenta no processo de desenvolvimento, os desenvolvedores apontaram que seu uso favoreceu a incorporação de práticas de acessibilidade desde etapas iniciais dos projetos pessoais. A possibilidade de avaliar aspectos de acessibilidade durante a codificação foi considerada um ponto positivo, trazendo o benefício direto da redução de retrabalho e da mitigação de erros que demandariam correções posteriores onerosas. Essa antecipação no ciclo de desenvolvimento promove um maior alinhamento com as diretrizes recomendadas e garante a entrega de um software com maior qualidade técnica inicial.

Os participantes também destacaram que a ferramenta atua como um apoio complementar às práticas já adotadas no desenvolvimento, não substituindo avaliações manuais ou testes com usuários, mas contribuindo para a identificação prévia de problemas passíveis de verificação automatizada ou assistida.

10. LIMITAÇÕES E SUGESTÕES DE MELHORIA

Apesar da avaliação positiva, os desenvolvedores apontaram algumas limitações da ferramenta, principalmente relacionadas à abrangência das verificações realizadas. Foi mencionado que determinados aspectos de acessibilidade ainda dependem fortemente de avaliação humana, especialmente aqueles de natureza mais subjetiva.

Diante de tal cenário, um exemplo prático é a validação do conteúdo de um *semanticsLabel*: embora a ferramenta identifique com precisão a ausência do atributo em um botão de ícone, cabe ao desenvolvedor julgar se a descrição textual fornecida é clara e condizente com a ação executada pelo componente. Essa distinção reforça que o *Flutter Access* atua como um suporte complementar, não substituindo a necessidade de inspeções manuais para garantir a clareza semântica e a adequação do conteúdo ao contexto de uso.

Como sugestões de melhoria, os participantes indicaram a possibilidade de ampliar o conjunto de verificações suportadas, sugerindo a implementação de regras mais abrangentes voltadas à acessibilidade visual, como a análise

de contraste e legibilidade para diferentes níveis de daltonismo. Isso permitiria que a ferramenta identificasse combinações de cores que, embora legíveis para usuários normovisuais, tornam-se indistinguíveis para pessoas com deficiências na percepção de cores.

Quanto ao feedback da ferramenta, atualmente apresentado de forma textual e descritiva no relatório ou no terminal(indicando a linha do erro), os desenvolvedores sugeriram que as mensagens passassem a incluir o grau de severidade da violação. Essa melhoria permitiria categorizar os problemas encontrados em níveis, auxiliando o desenvolvedor a priorizar correções que impedem o uso básico da aplicação por tecnologias assistivas antes de focar em ajustes puramente estéticos.

De forma geral, os resultados obtidos indicam que a ferramenta atende aos objetivos propostos neste trabalho, apresentando-se como um apoio relevante para desenvolvedores interessados em incorporar práticas de acessibilidade no desenvolvimento de aplicações móveis, ao mesmo tempo em que evidencia oportunidades para sua evolução futura.

11. CONCLUSÃO

Este trabalho apresentou o *Flutter Access*, uma extensão para o Visual Studio Code voltada ao apoio à análise de acessibilidade em aplicações móveis desenvolvidas com Flutter/Dart. A ferramenta foi concebida com o objetivo de auxiliar desenvolvedores na identificação de potenciais problemas de acessibilidade, contribuindo para a incorporação de boas práticas desde as etapas iniciais do desenvolvimento de software.

A proposta do Flutter Access fundamenta-se em três pilares principais: (i) o estudo teórico sobre acessibilidade digital e móvel, com base em diretrizes reconhecidas como as WCAG; (ii) a análise de trabalhos acadêmicos e ferramentas existentes voltadas à avaliação automática e inspeção de acessibilidade em aplicações móveis; e (iii) a validação prática da ferramenta por desenvolvedores, que a utilizaram em projetos pessoais, permitindo avaliar sua aplicabilidade, usabilidade e utilidade no contexto real de desenvolvimento.

Durante o processo de validação, observou-se que a ferramenta atende aos objetivos propostos, sendo considerada de fácil utilização e integrada de forma satisfatória ao fluxo de trabalho dos desenvolvedores. O Flutter Access mostrou-se eficaz como ferramenta de apoio, especialmente ao evidenciar aspectos de acessibilidade que frequentemente passam despercebidos durante o desenvolvimento convencional de aplicações móveis.

Os resultados indicam que o *Flutter Access* atua como um suporte relevante para a detecção prévia de problemas passíveis de análise automatizada durante o desenvolvimento da aplicação. Entretanto, é importante ressaltar que a ferramenta não elimina a necessidade de realizar testes com usuários finais no período pós-desenvolvimento, etapa essencial para verificar a eficácia das decisões de acessibilidade tomadas.

Dessa forma, a ferramenta contribui para a conscientização dos desenvolvedores quanto à importância da acessibilidade

digital e para a melhoria da qualidade inclusiva das aplicações móveis. Assim, conclui-se que o Flutter Access representa uma contribuição válida no contexto de ferramentas de apoio ao desenvolvimento acessível, reforçando a necessidade de integrar práticas de acessibilidade de forma contínua e sistemática nos processos de desenvolvimento de software móvel.

12. TRABALHOS FUTUROS

A partir do desenvolvimento e da validação do *Flutter Access*, foram identificadas oportunidades de evolução técnica e funcional para a ferramenta. Como perspectivas para a continuidade desta pesquisa, propõem-se os seguintes pontos:

- **Suporte multiplataforma (Linux e macOS):** Atualmente o *Flutter Access* apresenta limitações de execução fora do ambiente Windows, uma vez que parte de sua implementação depende da utilização de um executável no formato `.exe`, exclusivo desse sistema operacional. Como trabalho futuro, propõe-se a adaptação da ferramenta para garantir compatibilidade nativa com sistemas Linux e macOS, por meio da substituição do executável por soluções multiplataforma ou da utilização direta de mecanismos do ecossistema Dart/Flutter.
- **Adição de novas regras de acessibilidade:** A ampliação do conjunto de regras de acessibilidade analisadas pela ferramenta é um ponto relevante para evolução do *Flutter Access*. Como trabalho futuro, pretende-se incorporar novas verificações baseadas nas diretrizes WCAG, bem como em recomendações específicas para aplicações móveis desenvolvidas com Flutter, aumentando a cobertura e a precisão das análises realizadas. Ademais, embora a versão atual da ferramenta realize a análise estática de cores definidas explicitamente no código-fonte, propõe-se para trabalhos futuros expandir essa verificação para englobar contrastes e temas injetados de forma dinâmica em tempo de execução (*runtime*).
- **Configuração personalizada das regras de análise:** Uma possível melhoria consiste em permitir que os desenvolvedores configurem quais regras de acessibilidade desejam aplicar em seus projetos, possibilitando análises personalizadas de acordo com o contexto da aplicação e o público-alvo.
- **Integração com pipelines de integração contínua (CI/CD):** Propõe-se como trabalho futuro a integração do *Flutter Access* com pipelines de CI/CD, permitindo que as análises de acessibilidade sejam executadas automaticamente durante o processo de build, contribuindo para a detecção precoce de problemas.
- **Validação em cenários de desenvolvimento diversificados:** Como perspectiva de evolução, propõe-se a aplicação do *Flutter Access* em uma base mais ampla e diversificada de projetos *Flutter*. Isso envolve a análise de repositórios com diferentes padrões arquiteturais, níveis de complexidade e utilização de bibliotecas de terceiros. Tal ampliação permitirá validar a eficácia da ferramenta em variados contextos de codificação,

garantindo a precisão das auditorias independentemente da estrutura organizacional do código-fonte.

- **Apoio à análise de acessibilidade em tempo real:** Outra possibilidade de evolução consiste em fornecer feedback em tempo real durante a escrita do código, alertando o desenvolvedor sobre possíveis problemas de acessibilidade à medida que o código é produzido.

13. REFERÊNCIAS

- [1] Fundação Getúlio Vargas. *Pesquisa revela que Brasil tem 480 milhões de dispositivos digitais em uso, sendo 2,2 por habitante*. 2024. <https://portal.fgv.br/noticias/pesquisa-revela-brasil-tem-480-milhoes-dispositivos-digitais-uso-sendo-22-habitante>.
- [2] NORMAN, D. A. *O design do dia a dia*. New York, NY, USA: Basic Books, 2013. ISBN 9780465050659.
- [3] Cetic.br. *Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos domicílios brasileiros — TIC Domicílios 2024*. 2025. https://cetic.br/media/docs/publicacoes/2/pt-br/20260511173200/tic_domicilios_2025_livro%20pt.pdf. Acesso em: maio 2026.
- [4] MARQUES, S.; VIANA, M.; CARVALHO, D. Acessibilidade no desenvolvimento de aplicativos móveis: uma revisão sistemática sobre problemas recorrentes, métodos de verificação e ferramentas. In: *Anais do XXIV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais*. Porto Alegre, RS, Brasil: SBC, 2025. p. 1277–1300. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/ihc/article/view/37710>>.
- [5] ONU News. *Inovação para mais inclusão marca Dia Internacional das Pessoas com Deficiência*. 2022. <https://news.un.org/pt/story/2022/12/1806232>. Publicado em 3 dez. 2022. Acesso em: nov. 2025.
- [6] Governo Digital. *Acessibilidade Digital*. 2025. <https://www.gov.br/governodigital/pt-br/acessibilidade-e-usuario/acessibilidade-digital>.
- [7] Associação Brasileira de Normas Técnicas. *ABNT NBR 17060:2022 — Acessibilidade em aplicativos de dispositivos móveis — Requisitos*. 2022. <https://cta.ifrs.edu.br/abnt-nbr-17060-2022-acessibilidade-em-aplicativos-de-dispositivos-moveis-requisitos/#:~:text=A%20Norma%20Brasileira%20ABNT%20NBR,dispositivos%20m%C3%B3veis%2C%20sejam%20eles%20nativos%2C>. Norma técnica que estabelece requisitos para facilitar e otimizar o acesso de pessoas com deficiência a ambientes virtuais por meio de aplicativos de dispositivos móveis, visando eliminar ou reduzir barreiras de uso.
- [8] World Wide Web Consortium (W3C). *WCAG — Web Content Accessibility Guidelines*. 2023. <https://www.w3.org/WAI/standards-guidelines/wcag/>.
- [9] Flutter Documentation. *Accessibility*. 2025. <https://docs.flutter.dev/ui/accessibility>. Documentação oficial do Flutter sobre acessibilidade em aplicações, incluindo suporte a tecnologias assistivas, boas práticas de design e checklist para acessibilidade de interfaces de usuário.
- [10] SILVA, C.; ELER, M. Um estudo sobre a avaliação automática de diretrizes de acessibilidade para dispositivos móveis. In: SBC. *Simpósio Brasileiro de Sistemas de Informação (SBSI)*. [S.l.], 2018. p. 111–113.
- [11] GUIMARÃES, A. P. N.; TAVARES, T. A. Avaliação de interfaces de usuário voltada à acessibilidade em dispositivos móveis: Boas práticas para experiência de usuário. In: SBC. *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*. [S.l.], 2014. p. 22–29.
- [12] SILVA, C. A.; FREIRE, A. P. Inspeção da acessibilidade de aplicativos móveis utilizando software leitor de telas. In: SBC. *Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC)*. [S.l.], 2018.
- [13] Google. *Usar o Accessibility Scanner*. 2025. <https://support.google.com/accessibility/android/answer/6376570?hl=pt-BR>. Acesso em: 2025.
- [14] Deque Systems. *axe DevTools for Mobile Accessibility Testing*. 2025. Acesso em: 2025. Disponível em: <<https://www.deque.com/axe/devtools/mobile-accessibility/>>.
- [15] Evinced Inc. *Mobile Accessibility Digital Helper*. 2025. <https://www.evinced.com/products/flow-analyzer-for-mobile>. Acessado em: 03 abr. 2026.
- [16] SILVA, G. d. M. F. et al. Acessibilidade digital: Quando a tecnologia quebra barreiras. *ARACÊ*, v. 7, n. 3, p. 11604–11618, 2025.
- [17] COSTA, L. C. A. da; SOUSA, M. R. F. de. Acessibilidade web móvel: identificação de barreiras de acesso informação em páginas de redes sociais. *Informação & Informação*, v. 26, n. 3, p. 207–228, 2021.
- [18] DEVELOPERS, A. *Android Accessibility Guidelines*. <https://developer.android.com/guide/topics/ui/accessibility?hl=pt-br>.
- [19] Comitê Gestor da Internet no Brasil. *Acessibilidade e tecnologias: um panorama sobre acesso e uso de Tecnologias de Informação e Comunicação por pessoas com deficiência no Brasil*. São Paulo: Comitê Gestor da Internet no Brasil, 2020. <https://cetic.br/media/docs/publicacoes/7/20200528104403/estudos-setoriais-acessibilidade-e-tecnologias.pdf>. Acesso em: 18 abr. 2026.
- [20] World Wide Web Consortium (W3C). *About us*. 2025. <https://www.w3.org/about/>.
- [21] World Wide Web Consortium (W3C). *Our mission*. 2025. <https://www.w3.org/mission/>.
- [22] World Wide Web Consortium (W3C) Web Accessibility Initiative (WAI). *Accessibility — WAI*. 2025. <https://www.w3.org/mission/accessibility/>.
- [23] World Wide Web Consortium (W3C). *Web Content Accessibility Guidelines (WCAG) 2.1*. 2018. Diretrizes de Acessibilidade para Conteúdo Web 2.1 — recomendação da W3C para tornar o conteúdo web mais acessível a pessoas com deficiência. Disponível em: <<https://www.w3.org/TR/WCAG21/>>.
- [24] Governo Federal do Brasil. *e-MAG — Modelo de Acessibilidade em Governo Eletrônico, versão 3.1*. 2014. Manual técnico. Disponível em <https://emag.governoeletronico.gov.br/>.
- [25] AGARWAL, H. *Mastering Accessibility (A11y) in Flutter: The Only Guide You'll Ever Need*. 2025. <https://himanshu-agarwal.medium.com/mastering-accessibility-a11y-in-flutter-the-only-guide-youll-ever-need-05cfd4dbf664>. Acesso em: 2026.

- [26] Flutter Dev. *Flutter Architectural Overview*. 2026.
<https://docs.flutter.dev/resources/architectural-overview#architectural-layers>. Acessado em: 15 jun. 2026.
- [27] Dart Team. *analyzer – Dart package*. tools.dart.dev, 2025. Acesso em 2025. Disponível em:
<<https://pub.dev/packages/analyzer>>.

APÊNDICE

A. FIGURAS

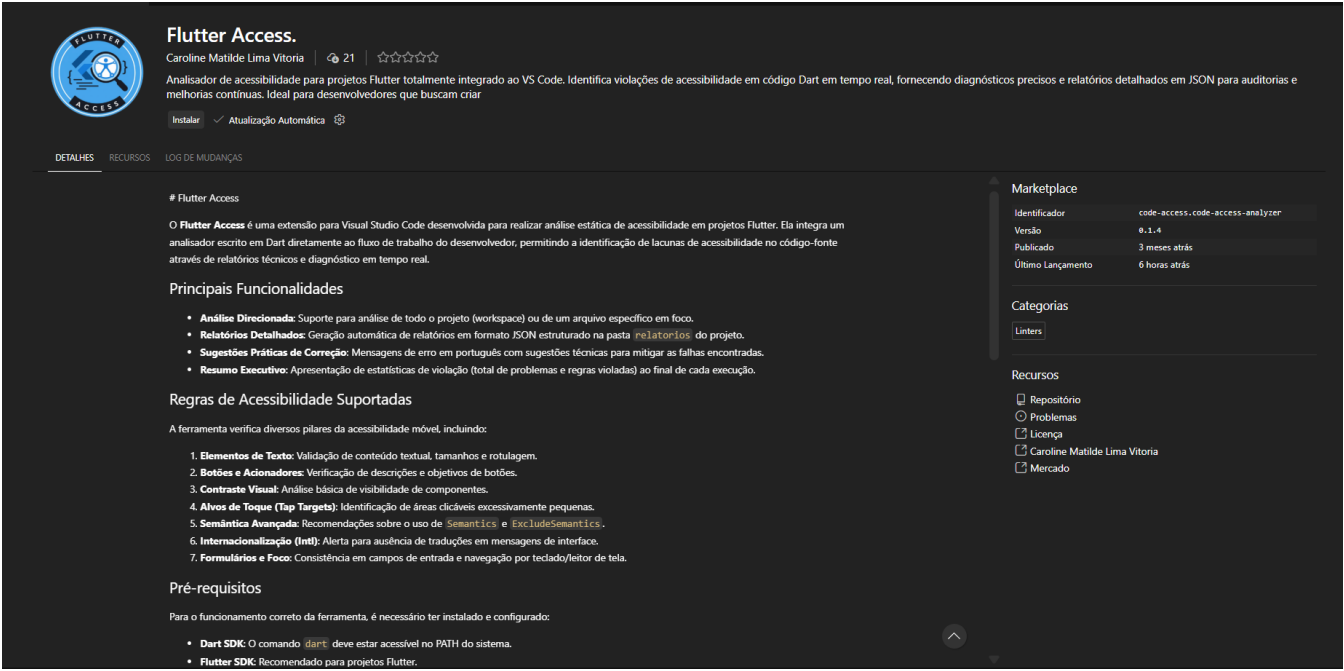


Figure A.1: Extensão Flutter Access integrada ao Visual Studio Code

```

relatorios > {} accessibility-report.json > ...
1 {
2   "geradoEm": "11/01/2026, 18:48:17",
3   "totalDeProblemas": 102,
4   "problemas": [
5     {
6       "regra": "TextRule",
7       "mensagem": "Text sem Semantics ou style legível: Text('Escolha seu assento') (assento_page.dart:45:36)",
8       "arquivo": "c:\\Users\\Carol\\meufilmeapp\\lib\\pages\\assento_page.dart",
9       "sugestao": "Verifique o contraste das cores do texto para garantir legibilidade (mínimo 4.5:1 para texto normal, 3:1 para texto grande). Use ferramentas como o Contrast Checker.",
10    },
11    {
12      "regra": "TextRule",
13      "mensagem": "Text sem Semantics ou style legível: Text('\\Selecione um Assento!\\') (assento_page.dart:109:36)",
14      "arquivo": "c:\\Users\\Carol\\meufilmeapp\\lib\\pages\\assento_page.dart",
15      "sugestao": "Verifique o contraste das cores do texto para garantir legibilidade (mínimo 4.5:1 para texto normal, 3:1 para texto grande). Use ferramentas como o Contrast Checker.",
16    },
17    {
18      "regra": "TextRule",
19      "mensagem": "Text sem Semantics ou style legível: const Text('\\Próximo\\') (assento_page.dart:119:26)",
20      "arquivo": "c:\\Users\\Carol\\meufilmeapp\\lib\\pages\\assento_page.dart",
21      "sugestao": "Verifique o contraste das cores do texto para garantir legibilidade (mínimo 4.5:1 para texto normal, 3:1 para texto grande). Use ferramentas como o Contrast Checker.",
22    },
23    {
24      "regra": "ButtonRule",
25      "mensagem": "Botão sem tooltip: ElevatedButton(onPressed: () {if (indice < 0) {ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text('\\Selecione um Assento!\\'))});} else {widget.in",
26      "arquivo": "c:\\Users\\Carol\\meufilmeapp\\lib\\pages\\assento_page.dart",
27      "sugestao": "Envolva o botão com 'Tooltip' para fornecer uma dica de texto (ex.: 'Tooltip(message: '\\Comprar Ingresso\\', child: ElevatedButton(...))' ou adicione 'Semantics' com a propriedade 'labe",
28    },
29    {
30      "regra": "ContrastRule",
31      "mensagem": "Texto com baixo contraste (ratio 1.00) – verificar contraste WCAG 4.5:1 (assento_page.dart:56:24)",
32      "arquivo": "c:\\Users\\Carol\\meufilmeapp\\lib\\pages\\assento_page.dart",
33      "sugestao": "Verifique o contraste das cores do texto para garantir legibilidade (mínimo 4.5:1 para texto normal, 3:1 para texto grande). Use ferramentas como o Contrast Checker.",
34    },
35    {
36      "regra": "ContrastRule",
37      "mensagem": "Texto com baixo contraste (ratio 1.00) – verificar contraste WCAG 4.5:1 (assento_page.dart:91:26)",
38      "arquivo": "c:\\Users\\Carol\\meufilmeapp\\lib\\pages\\assento_page.dart",
39      "sugestao": "Verifique o contraste das cores do texto para garantir legibilidade (mínimo 4.5:1 para texto normal, 3:1 para texto grande). Use ferramentas como o Contrast Checker.",
40    },
41  ],
42 }

```

Figure A.2: Relatório de acessibilidade gerado pelo Flutter
Access

```
SAÍDA Filtro Code Access

=====
Arquivo: assento_page.dart
=====

Problemas encontrados neste arquivo (14): assento_page.dart
- assento_page.dart - Text sem Semantics ou style legível: Text('Escolha seu assento') (assento_page.dart:45:36)
- assento_page.dart - Text sem Semantics ou style legível: Text("Selecione um Assento!") (assento_page.dart:109:36)
- assento_page.dart - Text sem Semantics ou style legível: const Text("Próximo") (assento_page.dart:119:26)
- assento_page.dart - Botão sem tooltip: ElevatedButton(onPressed: () {if (indice < 0) {ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text("Selecione um Assento!"))}); else {widget.ingresso.assento = "A${indice + 1}"; Navigator.push(context, MaterialPageRoute(builder: (context) => CombosPage(ingresso: widget.ingresso))));}, child: const Text("Próximo")) (assento_page.dart:105:15)
- assento_page.dart - Texto com baixo contraste (ratio 1.00) - verificar contraste WCAG 4.5:1: (assento_page.dart:56:24)
- assento_page.dart - Texto com baixo contraste (ratio 1.00) - verificar contraste WCAG 4.5:1: (assento_page.dart:91:26)
- assento_page.dart - InkWell interativo sem FocusNode - considere adicionar para navegação via teclado: (assento_page.dart:73:22)
- assento_page.dart - ElevatedButton interativo sem FocusNode - considere adicionar para navegação via teclado: (assento_page.dart:105:15)
- assento_page.dart - Text usado como título sem Semantics(header: true): (assento_page.dart:91:26)
- assento_page.dart - String literal sem internacionalização: "Escolha seu assento" (assento_page.dart:45:41)
- assento_page.dart - String literal sem internacionalização: "TELA" (assento_page.dart:57:19)
- assento_page.dart - String literal sem internacionalização: "A${indice + 1}" (assento_page.dart:92:21)
- assento_page.dart - String literal sem internacionalização: "Selecione um Assento!" (assento_page.dart:109:41)
- assento_page.dart - String literal sem internacionalização: "Próximo" (assento_page.dart:119:37)

=====
Arquivo: cart_page.dart
=====

Problemas encontrados neste arquivo (12): cart_page.dart
- cart_page.dart - Text sem Semantics ou style legível: const Text('Minhas Compras') (cart_page.dart:15:16)
- cart_page.dart - Text sem Semantics ou style legível: Text('Filme: ${listaIngresso[index].nomeFilme}') (cart_page.dart:45:28)
- cart_page.dart - Imagem sem semanticLabel: Image.network(listaIngresso[index].ImageUrlFilme!, fit: BoxFit.cover) (cart_page.dart:40:30)
- cart_page.dart - Navegação detectada dentro de callback - evite navegação automática que confunda usuário: (cart_page.dart:27:24)
- cart_page.dart - GestureDetector interativo sem FocusNode - considere adicionar para navegação via teclado: (cart_page.dart:27:24)
- cart_page.dart - Text usado como título sem Semantics(header: true): (cart_page.dart:19:22)
- cart_page.dart - Text usado como título sem Semantics(header: true): (cart_page.dart:50:34)
- cart_page.dart - Text usado como título sem Semantics(header: true): (cart_page.dart:57:25)
- cart_page.dart - String literal sem internacionalização: "Minhas Compras" (cart_page.dart:15:27)
- cart_page.dart - String literal sem internacionalização: "Você ainda não possui compras." (cart_page.dart:20:17)
- cart_page.dart - String literal sem internacionalização: "Filme: ${listaIngresso[index].nomeFilme}" (cart_page.dart:45:33)
```

Figure A.3: Apontamento de erros de acessibilidade no terminal

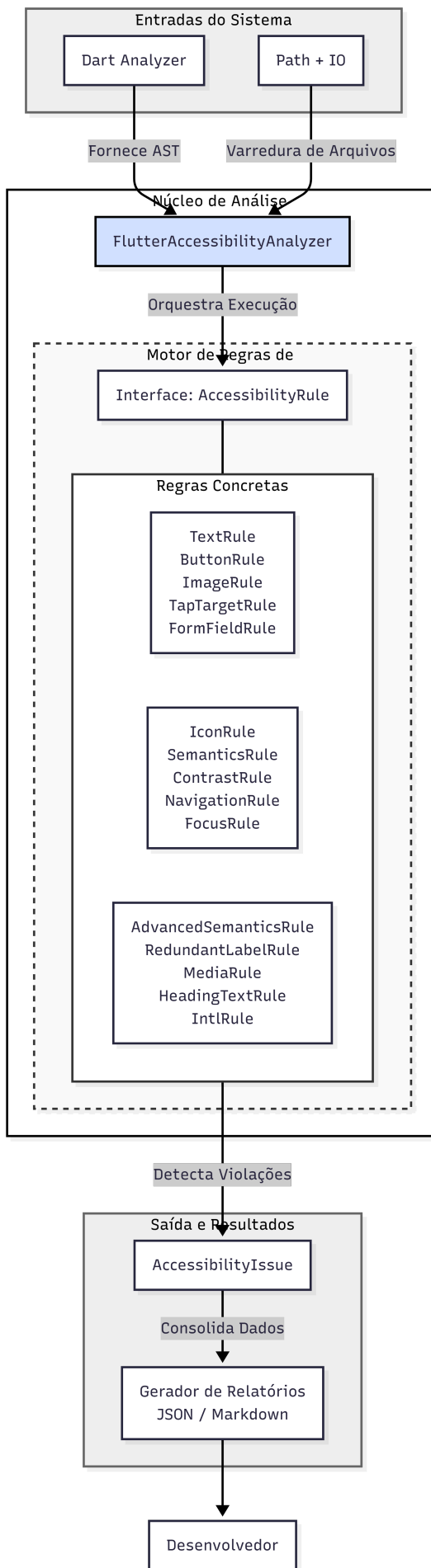


Figure A.4: Arquitetura da Ferramenta

B. QUESTIONÁRIO - FLUTTER ACCESS

Questionário sobre Acessibilidade no Desenvolvimento de Software Mobile

Você está sendo convidado(a) a participar do estudo de validação do Trabalho de Conclusão de Curso intitulado **"Acessibilidade no Desenvolvimento de Software: Análise e Proposta de Ferramenta de Verificação"**, de responsabilidade da aluna **Caroline Matilde Lima Vitória**, do Instituto Federal da Bahia (IFBA).

Leia atentamente as informações abaixo e sinta-se à vontade para esclarecer quaisquer dúvidas antes de prosseguir.

1. Objetivo:

O estudo tem como objetivo analisar o uso de práticas e ferramentas de acessibilidade no desenvolvimento de software, além de validar a proposta de criação de um plugin para apoio aos desenvolvedores.

2. Participação:

A participação consiste em responder um questionário com perguntas sobre seu perfil profissional, experiência em desenvolvimento e percepções sobre acessibilidade em código.

3. Riscos e benefícios:

A pesquisa não oferece riscos aos participantes. Os dados coletados contribuirão para avanços no tema de acessibilidade digital e poderão embasar futuras ferramentas para a área.

4. Sigilo e privacidade:

As respostas serão utilizadas apenas para fins acadêmicos e apresentadas de forma agregada. Nenhum dado pessoal (como nome ou e-mail) será divulgado.

5. Direitos do participante:

A participação é voluntária. Você pode desistir a qualquer momento, sem qualquer prejuízo.

6. Contato:

Para dúvidas ou mais informações, entre em contato com a autora, e-mail: carol.lima.vitoria@gmail.com.

Figure B.5: Questionário - Introdução

...

Você declara que leu o termo acima e concorda em participar voluntariamente desta pesquisa?

☐ Sim, concordo em participar do estudo.

☐ Não concordo e não desejo participar.

Qual seu gênero?

☐ Feminino

☐ Masculino

☐ Prefiro não responder

☐ Outro:

Qual sua faixa etária?

☐ Entre 15 a 24 anos

☐ Entre 25 a 34 anos

☐ Entre 35 a 45 anos

☐ Acima de 45 anos

Figure B.6: Questionário - Questões 1, 2 e 3

...

Há quanto tempo você atua com desenvolvimento de software?

- ☐ Menos de 1 ano
- ☐ 1 a 2 anos
- ☐ 3 a 5 anos
- ☐ 6 a 10 anos
- ☐ Mais de 10 anos
- ☐ Não atuo diretamente com desenvolvimento de software

Há quanto tempo você atua com desenvolvimento de aplicativos mobile?

- ☐ Não atuo com desenvolvimento mobile
- ☐ Menos de 1 ano
- ☐ 1 a 2 anos
- ☐ 3 a 5 anos
- ☐ Mais de 5 anos

Figure B.7: Questionário - Questões 4 e 5

Quais linguagens ou frameworks você utiliza no desenvolvimento mobile?

☐ Flutter

☐ Kotlin

☐ React Native

☐ Java

☐ Swift

☐ Outro:

Você utiliza alguma ferramenta ou recurso voltado à acessibilidade durante o desenvolvimento de software mobile?

☐ Sim

☐ Não

Caso tenha respondido “**sim**” à pergunta anterior, quais ferramentas você utilizou?

Texto de resposta longa

Figure B.8: Questionário - Questões 6, 7 e 8

...

Caso tenha respondido “**sim**”, descreva brevemente como a acessibilidade é considerada e aplicada no seu processo de desenvolvimento de software.

Texto de resposta longa

Você acredita que um plugin ou ferramenta que verifique acessibilidade durante o desenvolvimento seria útil?

☐ Sim, ajudaria bastante

☐ Não vejo necessidade

☐ Não tenho opinião formada

☐ Outro:

Gostaria de deixar alguma observação, contribuição ou sugestão sobre acessibilidade em dispositivos móveis?

Texto de resposta longa

Figure B.9: Questionário - Perguntas Finais

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Você está sendo convidado(a) para participar do estudo de validação do Trabalho de Conclusão de Curso intitulado "Flutter Access: Avaliando acessibilidade em dispositivos móveis com flutter", de responsabilidade da aluna Caroline Matilde Lima Vitória. Leia cuidadosamente o que se segue e caso tenha alguma dúvida, sinta-se à vontade para perguntar. Caso esteja de acordo com as informações apresentadas neste termo e aceite participar da pesquisa, peço que assine ao final deste documento. Saiba que você tem total direito de não querer participar.

1.O presente trabalho tem como objetivo analisar a aplicação de práticas de acessibilidade no desenvolvimento de software, com foco em aplicações para dispositivos móveis. A pesquisa pretende contribuir para a melhoria da qualidade e inclusão durante o desenvolvimento, incentivando a adoção de práticas que tornem aplicações digitais mais acessíveis para todos os usuários.

2. A coleta de dados ocorrerá via questionário, abordando o perfil profissional do participante e sua experiência com desenvolvimento mobile. O objetivo é avaliar a percepção dos desenvolvedores sobre o uso de práticas inclusivas e a utilidade de ferramentas que automatizam a verificação de acessibilidade no fluxo de trabalho

3. A participação nesta pesquisa não envolve qualquer custo para os participantes. Além disso, o participante poderá retirar sua concordância e interromper sua participação a qualquer momento, sem qualquer prejuízo.

4. Não há qualquer tipo de pagamento, recompensa financeira ou cobrança pela participação nesta pesquisa.

5. As informações coletadas serão tratadas de forma confidencial e anônima. Nenhum dado pessoal que permita identificar os participantes será divulgado. Os resultados serão apresentados apenas de forma agregada e exclusivamente para fins acadêmicos. Os participantes também terão acesso, caso desejem, a informações adicionais sobre a pesquisa antes, durante ou após sua participação.

6. Os dados coletados serão utilizados exclusivamente para fins acadêmicos, no contexto do Trabalho de Conclusão de Curso, podendo também ser apresentados em relatórios, artigos científicos ou apresentações acadêmicas relacionadas ao estudo.

Em caso de dúvidas ou para obter mais informações sobre a pesquisa, entre em contato com Caroline Matilde Lima Vitória, aluna responsável pela pesquisa. Telefone: 71 985286351, email: carol.lima.vitoria@gmail.com

Assinatura do participante



Data ____ / ____ / ____

Nome _____

Questionário de Feedback

1 - Qual o seu nível de experiência com o framework Flutter?

- ☐ Iniciante / Estudante
- ☐ Júnior
- ☐ Pleno/Sênior

2 - Em qual contexto você aplicou a ferramenta Flutter Access?

- ☐ Projeto Acadêmico/Pessoal
- ☐ Projeto Freelancer
- ☐ Ambiente Corporativo (Empresa)

3 - A instalação e configuração da ferramenta foi clara e concisa? Se não, descreva as dificuldades enfrentadas.

- ☐ Sim
- ☐ Não

4 - A utilização do Flutter Access durante a escrita do código (análise estática) alterou a sua percepção sobre a acessibilidade do projeto? De que forma?

5 – Quais pontos positivos você pode ressaltar sobre a ferramenta?

Figure B.11: Questionário de Feedback

6 - Quais dificuldades ou pontos negativos você encontrou durante o uso da ferramenta?

7 – Quais melhorias ou novas funcionalidades você poderia sugerir para a ferramenta?
