

Recursos de Acessibilidade em Componentes do Android Jetpack Compose

Saulo Matos Pereira Gomes¹, João Soares de Oliveira Neto¹

¹Instituto de Ciência, Tecnologia e Inovação – Universidade Federal da Bahia (UFBA)
Rua do Telégrafo, S/N, Centro – Camaçari – BA – Brasil

{saulo.gomes, j-neto}@ufba.br

Abstract. *Accessible mobile applications play a key role in the social, economic and political inclusion of people with disabilities, in line with the Sustainable Development Goals in achieving the reduction of inequalities. In this article, we explore interface design, accessibility guidelines for digital content and introduce the implementation of accessibility features in Jetpack Compose components for the development of accessible, effective and visually pleasing native Android applications.*

Resumo. *As aplicações móveis acessíveis desempenham um papel fundamental na inclusão social, econômica e política de pessoas com deficiência, alinhado aos Objetivos de Desenvolvimento Sustentável na conquista da redução das desigualdades. Neste artigo, exploramos o design de interface, as diretrizes de acessibilidade para conteúdo digital e introduzimos a implementação dos recursos de acessibilidade em componentes do Jetpack Compose para o desenvolvimento de aplicações acessíveis, eficazes e visualmente agradáveis no Android nativo.*

1. Introdução

É evidente que os dispositivos móveis – principalmente smartphones – tornaram-se parte essencial no cotidiano da sociedade moderna. Segundo a [ANATEL 2023], o Brasil teve, até Junho de 2023, mais de 250 milhões de acessos móveis. De acordo com o Banco Mundial, 15% da população global tem algum tipo de deficiência. Pessoas com deficiência dependem de aplicações e serviços de acessibilidade para se comunicar, aprender e trabalhar. No entanto, uma parte considerável das aplicações móveis ainda apresentam limitações quando tratamos do uso por pessoas com deficiência (PcD).

O tema da acessibilidade e da inclusão social da PcD constitui uma das pautas primordiais que devem reger os debates em nossa sociedade. Sendo assim, os Objetivos de Desenvolvimento Sustentável (ODS), principalmente o ODS 10 - Redução das desigualdades, abordam as pessoas com deficiência, promovendo a igualdade de oportunidade para todos. Especificamente, a meta 10.2 do ODS 10 busca empoderar e promover a inclusão social, econômica e política de todas as pessoas, incluindo pessoas com deficiência [Nações Unidas Brasil 2023]. Em vista disso, exploramos as diretrizes de acessibilidade para conteúdo digital e os conceitos por trás de um bom design de interface. Neste artigo, buscamos introduzir como os componentes do Jetpack Compose habilitam a implementação de acessibilidade no desenvolvimento de aplicações móveis para Android nativo, contribuindo diretamente para a redução das desigualdades e para a promoção de uma sociedade mais inclusiva e sustentável em linha com os Objetivos de Desenvolvimento Sustentável das Nações Unidas.

2. Metodologia

Inicialmente, o foco do estudo foi direcionado à identificação dos critérios essenciais pelos quais uma aplicação é classificada como acessível. Além disso, o design de interface na interação humano-computador foi investigado, com ênfase nos princípios de design que visam melhorar a experiência do usuário. Uma abordagem centrada no usuário foi adotada para a compreensão do Design de Interação. Os conceitos relacionados à usabilidade em interfaces e as heurísticas para sua implementação foram estudados, assim como a compreensão das diretrizes de acessibilidade aplicadas a conteúdos digitais. A compreensão desses conceitos desempenha um papel crucial na orientação do desenvolvimento de aplicações móveis bem estruturadas, com o objetivo primordial de atender de maneira eficaz a todos os usuários.

Na segunda fase da pesquisa, a literatura técnica relacionada ao desenvolvimento de aplicações para Android nativo foi investigada. Isso abrangeu o estudo da linguagem de programação Kotlin, do software Android Studio e da biblioteca Jetpack Compose. A base do estudo foi constituída pela documentação oficial dessas ferramentas, complementada por cursos de capacitação oferecidos pelo Google na plataforma Developer Android. Ao longo do estudo foi possível compreender conceitos fundamentais de desenvolvimento mobile e, mais especificamente, dos recursos de acessibilidade disponíveis no Android Jetpack Compose.

3. Fundamentação teórica

3.1. O Design de Interface na Interação Humano-Computador

A interface de um sistema interativo compreende toda a porção do sistema com a qual o usuário mantém contato físico (motor ou perceptivo) ou conceitual durante a interação [Moran 1981, citado por Barbosa 2010]. Ela é o único meio de contato entre o usuário e o sistema. Por isso, a grande maioria dos usuários acredita que o sistema é a interface com a qual entram em contato [Hix e Hartson 1993, citado por Barbosa 2010]. Assim, as interfaces desempenham um papel fundamental no campo da Interação Humano-Computador (IHC), uma vez que têm como objetivo facilitar a comunicação e a execução de tarefas pelos usuários.

Segundo [Barbosa 2010], a interface determina os processos de interação possíveis em um sistema, ela delimita o que o usuário pode fazer, de que maneira e em que ordem. Portanto, ao definirmos como a interação deve ocorrer, estamos restringindo ou determinando algumas características da interface. Nesse aspecto, é muito importante planejar um bom design de interface. [Norman 2013] destaca alguns princípios de design que devem ser implementados para criar produtos e sistemas que atendam às necessidades e expectativas dos usuários, a saber:

- **Visibilidade:** os principais elementos da interface devem ser facilmente visíveis e compreensíveis para o usuário;
- **Feedback:** as interfaces devem oferecer um feedback claro sobre as ações do usuário e o estado do sistema;
- **Affordance:** é a qualidade de um sistema intuitivo, que permite ao usuário imaginar seu uso apenas ao olhar para ele;

- Mapeamento: refere-se ao relacionamento claro entre os controles e o resultado de suas ações;
- Restrições: limitar e controlar as ações do usuário na interface torna o sistema mais fácil de usar e reduz a quantidade de erros;
- Consistência: é necessário manter um padrão em relação ao comportamento e estética dos componentes da interface.

Essas seis diretrizes fornecem o esboço básico para uma ótima experiência e design de aplicativos, possibilitando aos desenvolvedores criar sistemas que são mais fáceis de usar, mais eficientes e mais satisfatórios para os usuários.

3.2. Usabilidade em Interfaces de Usuário

A Interface de Usuário (ou UI, abreviatura do termo em inglês *User Interface*) de um aplicativo móvel engloba tudo que é mostrado em tela: textos, imagens, botões, dentre outros componentes; além de como eles são organizados na tela. É a forma como a aplicação mostra as informações ao usuário e como o usuário interage com a aplicação.

Segundo [Preece 2005], ao planejar uma UI que permita a todos os usuários realizarem suas tarefas de maneira intuitiva e eficiente, é crucial a implementação de uma abordagem de design centrada no usuário, de modo a envolvê-lo no processo de criação. Isso significa que a interface deve ser projetada considerando as expectativas, habilidades, limitações e preferências dos usuários, ao invés de se concentrar apenas nas funcionalidades da aplicação.

Desse modo, [Preece 2005] destaca que o design de interação consiste em redirecionar essa preocupação, trazendo os princípios da usabilidade para dentro do processo de design. Uma preocupação central no processo do design de interação é o desenvolvimento de produtos interativos que sejam fáceis de aprender e utilizar, eficazes no uso e que proporcionem uma experiência agradável – sempre na perspectiva do usuário. Para a NBR 9241-11, usabilidade é uma “medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso” [NBR 2002]. Na prática, a usabilidade de uma interface é uma característica mais notável quando sua implementação é ruim do que quando é boa. Os problemas de usabilidade – como inconsistência na interface, falhas na interação e comunicação com o aplicativo, ausência de feedback e informações confusas ou conflitantes – afetam diretamente o usuário, levando ao sentimento de irritação e frustração, podendo motivá-lo a desistir da aplicação.

No contexto da IHC, podemos entender o conceito de [Nielsen 1994] para a usabilidade como uma junção de cinco atributos, que podem ser usados como diretrizes para verificar e garantir – através de testes com os usuários – um bom design de interface. São eles:

- Facilidade de aprendizado: ele se refere à facilidade com que os novos usuários podem aprender a completar tarefas básicas através de uma interface;
- Eficiência de uso: posto que os usuários aprenderam a utilizar a interface, ela deve possibilitar a execução rápida de tarefas;

- Facilidade de memorização: a interface deve possibilitar que os usuários memorizem suas funções com facilidade;
- Erros: deve haver uma baixa taxa de erros na utilização da interface e uma boa gestão caso venham a ocorrer;
- Satisfação: os usuários devem se sentir satisfeitos ao utilizar a aplicação. A interface deve ser agradável.

Portanto, ao se tratar de aplicações móveis, onde as interfaces são projetadas para serem utilizadas em dispositivos com telas menores e diferentes modos de interação, como toque e gestos, é necessário garantir um bom designer da experiência do usuário (ou UX, abreviatura do termo em inglês *User Experience*), de modo a prover uma melhor interação entre o usuário e a aplicação. Isso pode significar a utilização de elementos visuais claros e intuitivos, a organização das informações de forma lógica e coerente, e a implementação de recursos de acessibilidade como legendas, descrições de áudio, contraste de cores e outras técnicas que tornem as interfaces mais inclusivas. Por conseguinte, podemos definir que um bom design de interface permite que a aplicação seja utilizada por qualquer pessoa, a qualquer momento e em qualquer lugar.

3.3. Acessibilidade em Aplicações Móveis

Um conjunto de critérios que visa tornar o conteúdo digital mais acessível a todas as pessoas, incluindo aquelas com deficiências, são as Diretrizes de Acessibilidade de Conteúdo da Web (WCAG 2.1). Diretrizes essas que também podem ser aplicadas no design de aplicações móveis. Segundo a [WCAG 2.1 2018], um produto ou serviço digital inclusivo deve atender a quatro princípios fundamentais, sendo eles:

- Princípio Perceptível: para que o conteúdo seja perceptível, o usuário deve conseguir identificar – de forma clara e fácil – todos os elementos e informações presentes na interface, independente de sua capacidade sensorial. Isso inclui critérios como contraste de cor, tamanho de fonte, textos alternativos para conteúdo não textual e legenda.
- Princípio Operável: para que o conteúdo da sua aplicação móvel seja operável, é necessário que ele seja fácil de usar e navegar, independentemente da capacidade física do usuário. Isso inclui critérios como navegação por teclado, tempo suficiente para ler e usar o conteúdo, realizar o preenchimento de dados de modo simples e evitar conteúdo que possa causar convulsões.
- Princípio Compreensível: para que o conteúdo seja compreensível, é necessário que ele seja fácil de entender, independentemente da capacidade cognitiva do usuário. Isso inclui critérios como linguagem clara e simples, consistência na apresentação do conteúdo, apresentar navegação previsível e oferecer suporte a ajuda contextual.
- Princípio da Robustez: para que o conteúdo seja robusto, é necessário que ele seja compatível com diferentes tecnologias assistivas.

Na plataforma Android, o Google disponibiliza um serviço de acessibilidade projetado para usuários com deficiência que introduz ferramentas que alteram a maneira de interagir com o Android, mudando a forma como a interface funciona no dispositivo do usuário. Dentre essas ferramentas temos:

- Talkback: trata-se de um software leitor de tela para celulares. Um recurso de acessibilidade que ajuda pessoas com deficiência visual a selecionarem as opções presentes em menus do smartphone. O suporte de voz, para quem tem baixa ou perda total de visão, fala em voz alta (como um assistente pessoal) quais são as alternativas na tela [UFSM 2018]. É possível utilizar gestos para navegar e realizar ações frequentes no dispositivo Android [Google Support 2023].
- Switch Access: realiza a verificação dos itens na tela, destacando um por vez, até que o usuário faça uma seleção. Ele permite a interação com o dispositivo Android usando um ou mais interruptores em vez da tela touchscreen. [Google Support 2023].
- Voice Access: permite ao usuário controlar o seu dispositivo com comandos de voz. Use a voz para abrir apps, navegar e editar texto em modo mãos-livres [Google Support 2023].

Para que um app seja compatível com serviços de acessibilidade, é necessário usar APIs no *framework* do Android para expor informações semânticas sobre os elementos da UI. Em seguida, o *framework* do Android transmitirá essas informações semânticas aos serviços de acessibilidade; Cada serviço de acessibilidade pode escolher a melhor forma de descrever o app para o usuário [Developer Android 2022]. Apesar dos serviços de acessibilidade estarem em constante evolução, muitos desenvolvedores não sabem como construir sua aplicação para que ela funcione de forma adequada.

4. Desenvolvimento Android: a biblioteca Jetpack Compose

O Jetpack Compose é uma biblioteca moderna desenvolvida pelo Google, que utiliza o paradigma de programação declarativa para a criação de interfaces de usuário no desenvolvimento de aplicativos Android. Com o Jetpack Compose, ao invés de definir a UI por meio de arquivos XML (do termo em inglês *Extensible Markup Language*) e manipulação de *Views*, os desenvolvedores criam a UI diretamente em código Kotlin, de modo que não é mais necessário alternar entre as linguagens. Desse modo, o programador descreve como os componentes da UI devem ser renderizados, com base no estado e nos dados fornecidos, e a biblioteca lida com os detalhes da implementação. Isso não apenas torna o desenvolvimento mais eficiente, mas também oferece maior flexibilidade, reutilização de componentes e facilita a criação de interfaces acessíveis. Através dele, os desenvolvedores podem criar interfaces de usuário dinâmicas, interativas e visualmente atraentes de maneira mais simples e eficaz.

4.1. Recurso de inclusão e acessibilidade no Jetpack Compose

Quando se trata de acessibilidade, o Jetpack Compose oferece recursos e práticas que podem ajudar os desenvolvedores a criar aplicativos acessíveis. Desse modo, não é necessário que o desenvolvedor crie uma aplicação diferente para atender a cada tipo de deficiência. Ao invés disso, o Compose usa propriedades semânticas para transmitir as informações exibidas nos elementos da UI para os serviços de acessibilidade.

A árvore semântica descreve a UI de uma forma alternativa, que é compreensível para os serviços de acessibilidade e para o framework de testes. Os serviços de acessibilidade usam a árvore para descrever o app para usuários com necessidades

específicas. Já o framework de testes usa a árvore para interagir com o app e fazer declarações sobre ele. A árvore semântica não contém informações sobre como mostrar os elementos que podem ser compostos, mas sim sobre o significado semântico deles [Developer Android 2022]. Para visualizar a árvore semântica, é possível usar a ferramenta *Layout Inspector* no Android Studio.

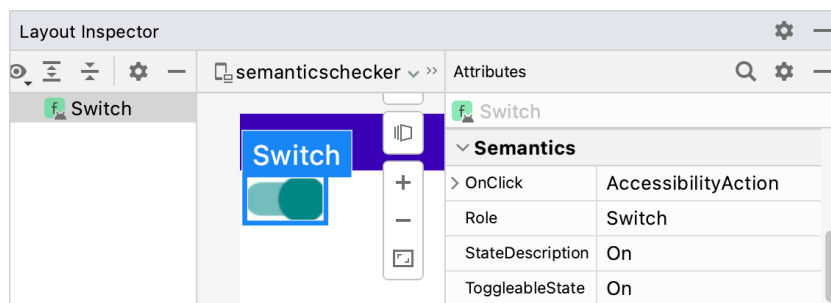


Figura 1. Propriedades semânticas do componente Switch no Layout Inspector [Developer Android 2022]

A árvore semântica será preenchida e gerada automaticamente caso o aplicativo seja formado por funções que podem ser compostas. Entretanto, ao criar suas próprias funções que podem ser compostas, será necessário informar a semântica correspondente manualmente. Além disso, podem haver situações em que a árvore não representa o significado dos elementos na tela de forma correta ou completa. Dentro do modificador *semantics* é possível editar as propriedades *contentDescription* para descrever o rótulo do componente e o *stateDescription* para definir os valores do seu estado mutável (caso haja). O atributo *Role* indica o tipo de elemento que estamos vendo.

A rotulação apropriada dos elementos interativos da interface é o primeiro passo para viabilizar a acessibilidade no Android. Os rótulos de acessibilidade auxiliam os usuários que não podem confiar na interface visual de um produto. Rótulos bem pensados ajudam a tornar a experiência baseada em texto tão utilizável quanto a experiência visual. Como observado na Figura 2, os rótulos devem descrever concisamente o conteúdo, a finalidade e o comportamento de um elemento – e não a aparência do ícone – por exemplo, é correto descrever “Pesquisa” ao invés de “Lupa”.

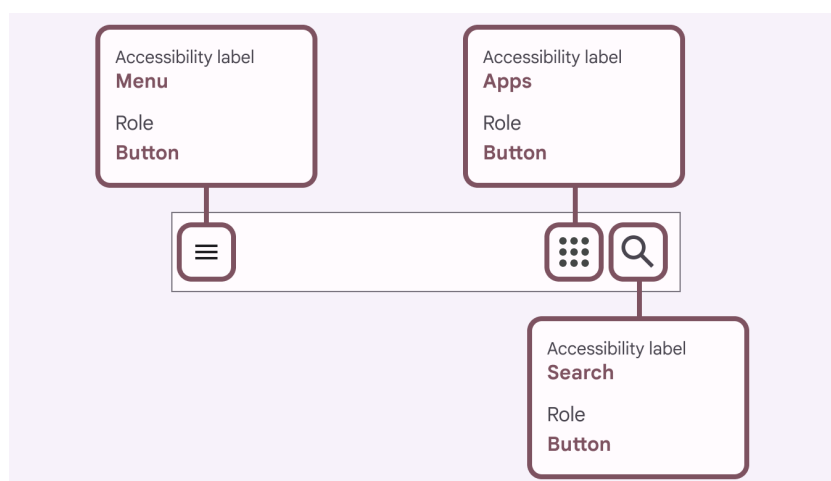


Figura 2. Exemplos de rótulos de acessibilidade [Material Design 3 2023]

Os serviços de acessibilidade, como o TalkBack e o Switch Access, permitem que os usuários coloquem o foco em diferentes elementos na tela. Em alguns casos, é preferível que os serviços de acessibilidade enxerguem um conjunto de elementos como uma peça única. Por exemplo, imagine um elemento de composição que mostra o avatar de um usuário, o nome e algumas informações complementares:

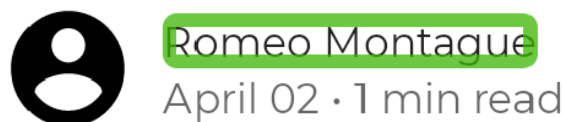


Figura 3. Elementos não agrupados [Developer Android 2022]

Assim, de modo a facilitar a navegação pelo aplicativo e evitar o excesso de componentes, é recomendável a formação de um grupo lógico. Para isso, é necessário instruir o Compose a mesclar esses elementos através do parâmetro *mergeDescendants* no modificador *semantics* do elemento pai. Dessa forma, os serviços de acessibilidade irão enxergar o grupo como um único componente, que terá todas as propriedades semânticas dos descendentes mescladas. A definição do modificador *clickable* em um componente sempre agrupa os elementos filhos.

```
@Composable
private fun PostMetadata(metadata: Metadata) {
    // Merge elements below for accessibility purposes
    Row(modifier = Modifier.semantics(mergeDescendants = true) {}) {
        Image(
            imageVector = Icons.Filled.AccountCircle,
            contentDescription = null // decorative
        )
        Column {
            Text(metadata.author.name)
            Text("${metadata.date} · ${metadata.readTimeMinutes} min read")
        }
    }
}
```

Figura 4. Exemplo de código para agrupar os elementos através do modificador *semantics* [Developer Android 2022]

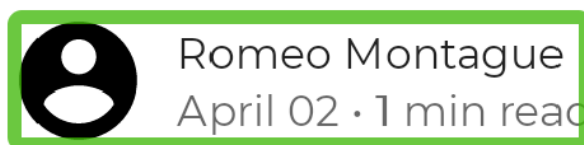


Figura 5. Elementos agrupados [Developer Android 2022]

Alguns conceitos comuns de design de acessibilidade precisam ser aplicados no desenvolvimento da UI. Esses conceitos são abordados no Material Design 3, que é um conjunto de diretrizes de design e acessibilidade sugerido pelo Google para criação de interfaces. Um bom exemplo é a utilização de textos alternativos, que ajudam a traduzir uma UI visual em uma UI baseada em texto. Quando você define um elemento de composição *Image* ou *Icon*, não há como o framework do Android entender de forma automática o que está sendo mostrado. É necessário fornecer uma descrição textual do

elemento visual. O parâmetro *contentDescription* é usado para descrever um elemento visual. O texto alternativo é um rótulo curto no código que descreve uma imagem para usuários que não conseguem vê-la. Um leitor de tela lerá o texto alternativo em voz alta no lugar da imagem. O texto alternativo também é valioso para usuários com visão, porque ele aparece se uma imagem não for carregada. Caso o elemento visual seja apenas decorativo, é indicado definir o parâmetro *contentDescription* como *null*, indicando ao framework do Android que esse elemento não contém ações ou estados associados.

Segundo o [Material Design 3 2023], todos os elementos mostrados na tela que permitem interação precisam ser suficientemente amplos para proporcionar uma interação confiável. No Android, a mínima área recomendada deve ser de 48 dp. Como pode ser observado na Figura 4, os valores de padding permitem que o tamanho visível de um objeto seja menor que 48 dp x 48 dp e ainda tenha o tamanho recomendado para a área de toque. Além disso, é recomendado manter o espaçamento mínimo de 8 dp entre os botões, para garantir uma melhor usabilidade.

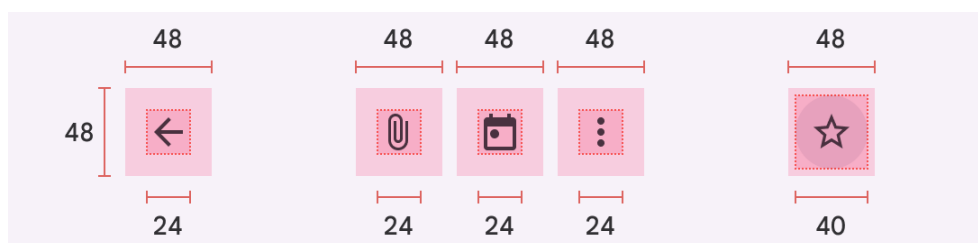


Figura 6. Área clicável em um ícone [Material Design 3 2023]

A cor e o contraste podem ser usados para ajudar os usuários a ver e interpretar o conteúdo do seu aplicativo, interagir com os elementos certos e compreender as ações. Cores primárias, secundárias e de destaque podem ser selecionadas para oferecer suporte à usabilidade [Material Design 3 2023]. Garantir o contraste apropriado é importante para os usuários distinguirem os elementos textuais e gráficos. Um contraste mais alto torna as imagens mais fáceis de ver, facilitando o uso da aplicação em um dia muito ensolarado ou à noite. Textos grandes (14 pt negrito ou 18 pt regular) e componentes devem atender a uma taxa de contraste de 3:1 contra a cor do plano de fundo. Já para textos pequenos, é recomendado o contraste de 4.5:1. O Material Design 3 disponibiliza uma ferramenta onde é possível definir as cores para o tema da sua interface e avaliar o contraste entre elas.

4.2. Testes de Acessibilidade

Os testes de acessibilidade desempenham um papel fundamental no desenvolvimento de aplicativos móveis, garantindo que eles sejam inclusivos e utilizáveis por todas as pessoas, independente de suas habilidades ou necessidades. Existem duas abordagens principais para realizar testes de acessibilidade em aplicações móveis: formas automáticas e formas manuais.

Uma forma eficaz de avaliar a acessibilidade da sua aplicação é utilizar ferramentas de análise automática, como o Scanner de Acessibilidade desenvolvido pelo Google para aplicativos Android. Essas ferramentas digitalizam a interface do usuário em busca de problemas comuns de acessibilidade como áreas de toque pequenas, baixo

contraste e descrições de conteúdo. Ele analisa a tela da aplicação, destaca os componentes que podem ter problemas de acessibilidade e oferece sugestões de como corrigir esses problemas. O Scanner de acessibilidade é uma introdução à acessibilidade em sua aplicação [Google Support 2023].

Embora as ferramentas automáticas sejam valiosas, os testes manuais são essenciais para uma avaliação completa da acessibilidade de um aplicativo móvel. Através de uma experiência de usuário simulada, os desenvolvedores devem interagir com a aplicação utilizando os serviços de acessibilidade do Android e testar o suporte da aplicação a várias formas de interação (toque, gestos e voz). Isso pode ajudar a detectar problemas que podem não ser detectados automaticamente. Além disso, é fundamental conduzir testes de acessibilidade com usuários reais, visto que a acessibilidade precisa ser observada holisticamente. Esses testes proporcionam valiosos insights e feedback direto sobre a acessibilidade da aplicação, ajudando a abordar problemas específicos de usabilidade que podem passar despercebidos de outra forma.

5. Considerações Finais

Diante dos Objetivos de Desenvolvimento Sustentável das Nações Unidas, em especial o de promover autonomia e igualdade de oportunidades a todos, incluindo pessoas com deficiência, é fundamental considerar o papel do desenvolvedor mobile na criação de aplicações móveis acessíveis que impactam positivamente a vida dessas pessoas.

Por meio da revisão bibliográfica, foi possível aprofundar a compreensão dos princípios subjacentes a um bom design de interface. As interfaces de usuário oferecem diversos modos de interação com o sistema, seja por toques, gestos ou voz. Nesse contexto, o Google introduziu no Android ferramentas que permitem a adaptação da interface por meio do seu serviço de acessibilidade. O design de interação centrado no usuário visa o desenvolvimento de aplicações com foco na usabilidade, tornando-as fáceis de aprender e utilizar, eficazes e proporcionando uma experiência de uso agradável. Além disso, abordamos as diretrizes de acessibilidade para conteúdo digital propostas pela WCAG 2.1 e as diretrizes de design do Material Design 3 propostas pelo Google.

Por último, exploramos como esses conceitos podem ser aplicados em conjunto com os componentes do Jetpack Compose durante o desenvolvimento da interface de usuário, abrangendo a árvore semântica, descrições de conteúdo, áreas de toque adequadas, cores e contrastes. Também introduzimos dois métodos preliminares de teste de acessibilidade. A condução de testes com usuários reais, especialmente aqueles com deficiências, é fundamental para garantir uma experiência positiva para todos os usuários. Ao adotar as metodologias e diretrizes apresentadas ao longo do artigo, os desenvolvedores têm capacidade de criar aplicações móveis que atendam a um público mais amplo, proporcionando uma experiência inclusiva e satisfatória para todos os usuários.

Reconhecemos o papel fundamental das aplicações móveis na melhoria da qualidade de vida e igualdade de oportunidades para todos os cidadãos quando projetadas com foco na acessibilidade. Para dar continuidade a esta pesquisa, propomos o desenvolvimento de um aplicativo que incorpore os conceitos de acessibilidade e usabilidade discutidos ao longo do artigo.

Referências

- ANATEL. (2023). *Agência Nacional de Telecomunicações*, <https://informacoes.anatel.gov.br/paineis/aceessos>, Agosto.
- Barbosa, S., & Silva, B. (2010). *Interação humano-computador*. Elsevier Brasil.
- Developer Android. (2022). *Acessibilidade no Compose*, <https://developer.android.com/jetpack/compose/accessibility>, Agosto.
- Developer Android. (2022). *Semântica no Compose*, <https://developer.android.com/jetpack/compose/semantics>, Agosto.
- Google Support. (2023). *Usar Gestos no TalkBack – Acessibilidade no Android*, <https://support.google.com/accessibility/android/answer/6151827?sjid=14822687814288420523-SA>, Agosto.
- Google Support. (2023). *Controlar o dispositivo Android usando o acesso com interruptor*, <https://support.google.com/accessibility/android/answer/6122836>, Agosto.
- Google Support. (2023). *Noções básicas sobre os comandos de fala do Voice Access*, <https://support.google.com/accessibility/android/answer/6151848>, Agosto.
- Google Support. (2023). *Primeiros passos com o Scanner de acessibilidade*. <https://support.google.com/accessibility/android/answer/6376570>, Agosto.
- Material Design 3. (2023). *Accessible design*, <https://m3.material.io/foundations/accessible-design/overview>, Agosto.
- NBR, A. (2002). 9241-11. Requisitos ergonômicos para trabalho de escritório com computadores: Parte 11 – Orientação sobre usabilidade. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Rio de Janeiro: sn, 21.
- Nações Unidas Brasil. (2023). *Os Objetivos de Desenvolvimento Sustentável no Brasil*, <https://brasil.un.org/pt-br/sdgs/10>, Agosto.
- Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann.
- Preece, J., Rogers, Y., Sharp, H. (2005). *Design de interação: Além da interação homem-computador*. Porto Alegre: bookman.
- UFSM. (2018). *O que é TalkBack?*, <https://ufsm.br/r-391-1722>, Agosto.
- WCAG 2.1. (2018). *Web Content Accessibility Guidelines (WCAG) 2.1*, <https://www.w3.org/TR/WCAG21/>, Agosto.