

ESCOLA POLITÉCNICA
ENGENHARIA DE COMPUTAÇÃO

RODRIGO PASTORELLO PEROZZO

**DESIGN SYSTEM COMO FERRAMENTA DE ACESSIBILIDADE PARA DEFICIENTES
VISUAIS EM APLICATIVOS MÓVEIS**

Porto Alegre
2023

GRADUAÇÃO



Pontifícia Universidade Católica
do Rio Grande do Sul

Rodrigo Pastorello Perozzo

Design System como Ferramenta de Acessibilidade para Deficientes Visuais em Aplicativos Móveis

Porto Alegre - RS, Brazil

2023

Rodrigo Pastorello Perozzo

Design System como Ferramenta de Acessibilidade para Deficientes Visuais em Aplicativos Móveis

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Computação da Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul.

Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS

Escola Politécnica

Engenharia de Computação

Orientador: Prof. Dr. César Marcon

Porto Alegre - RS, Brazil

2023

Agradecimentos

Primeiramente, e acima de todos, agradeço imensamente aos meus pais, cujo apoio incondicional tem sido o alicerce de todas as minhas decisões. Vocês tornaram tudo possível, formaram a base de quem eu sou, e por isso sou extremamente grato.

Ao meu irmão, meu primeiro companheiro, com quem tanto divido. Nosso crescimento compartilhado é fonte de força e inspiração.

À minha namorada, parceira constante nessa e em tantas outras jornadas. Sua incansável presença é o apoio perfeito e essencial que me faz seguir em frente.

Aos meus amigos de vida, estudos e carreira, que fizeram cada momento valer a pena. Vocês tornaram a jornada tão gratificante quanto o destino.

Estendo meus agradecimentos a todos os professores que cruzaram meu caminho acadêmico. Cada um de vocês desempenhou um papel fundamental na minha formação, seja desafiando minha percepção, alimentando minha curiosidade ou moldando minha ética profissional. Em especial, ao meu orientador, pelo constante incentivo, prontidão e dedicação ao aceitar o desafio.

A todos vocês, minha sincera gratidão.

Resumo

O crescimento exponencial da tecnologia móvel resultou em um mundo cada vez mais digital, com aplicativos servindo como um aspecto essencial da vida diária. No entanto, nessa revolução digital, a acessibilidade para usuários com deficiência visual muitas vezes não é endereçada de forma adequada. Apesar de diretrizes de acessibilidade e tecnologias assistivas estarem disponíveis, existe uma lacuna significativa entre esses guias e a implementação real da acessibilidade nas aplicações atuais. Este trabalho tem como objetivo levantar uma hipótese para preencher essa lacuna, desenvolvendo um sistema que incorpore a acessibilidade em seu núcleo. Reconhecendo a função dos *design systems* na construção de interfaces de usuário consistentes, eficazes e dimensionáveis, este trabalho propõe uma abordagem em que os princípios de acessibilidade são integrados à própria base dos elementos e componentes do sistema. Para avaliar a eficácia do *design system* proposto, foi criado um aplicativo de validação, com foco na experiência do usuário, reunindo respostas valiosas sobre o desempenho real do sistema. Essa experiência prática enfatizou ainda mais a importância de entender a acessibilidade do ponto de vista do usuário e reiterou a necessidade da acessibilidade ser presente nos processos de desenvolvimento. Ao implementar padrões de design acessíveis em um *design system*, os desenvolvedores podem promover a inclusão, aprimorar a usabilidade geral e contribuir para um mundo digital mais inclusivo.

Palavras-chave: Acessibilidade, *design system*, aplicações móveis.

Abstract

The exponential growth of mobile technology has resulted in an increasingly digital world, with mobile applications as an essential aspect of daily life. However, in this digital revolution, accessibility for visually impaired users is often not adequately addressed. Although accessibility guidelines and assistive technologies are available, there is a significant gap between these guides and the actual implementation of accessibility in current applications. This paper aims to raise a hypothesis to bridge this gap by developing a system that incorporates accessibility at its core. Recognizing the role of design systems in building consistent, efficient, and scalable user interfaces, this work proposes an approach in which accessibility principles are integrated into the foundation of system elements and components. To evaluate the efficiency of the proposed design system, a validation application was created, focusing on user experience gathering valuable feedback on the actual performance of the system. This practical experience further emphasized the importance of understanding accessibility from the user's point of view and reiterated the need for accessibility in development processes. By implementing accessible design patterns into a design system, developers can promote inclusion, enhance overall usability, and contribute to a more inclusive digital world.

Key-words: Accessibility, design system, mobile applications.

Lista de ilustrações

Figura 1 – Visibilidade Interação <i>Design System</i> (NZONGO, 2022)	20
Figura 2 – Anatomia de uma aplicação (FLUTTER, 2023)	27
Figura 3 – Arquitetura da metodologia (FROST, 2016)	29
Figura 4 – Cores do sistema proposto (reprodução do autor)	31
Figura 5 – Tipografia do sistema proposto (reprodução do autor)	31
Figura 6 – Raios de borda do sistema proposto (reprodução do autor)	32
Figura 7 – Espaçamentos do sistema proposto (reprodução do autor)	32
Figura 8 – Cálculo do contraste baseado em luminâncias (reprodução do autor) . . .	33
Figura 9 – Botão de Ícone — visibilidade normal e semântica (reprodução do autor)	36
Figura 10 – Botão de Texto — visibilidade normal e semântica (reprodução do autor)	37
Figura 11 – Campo de Entrada de Texto — visibilidade normal e semântica (reprodução do autor)	38
Figura 12 – Imagem — visibilidade normal e semântica (reprodução do autor)	38
Figura 13 – Link de Texto — visibilidade normal e semântica (reprodução do autor)	39
Figura 14 – Snackbar — visibilidade normal e semântica (reprodução do autor) . . .	40
Figura 15 – PDP — visibilidade normal e semântica (reprodução do autor)	43
Figura 16 – Carrinho de Compras — visibilidade normal e semântica (reprodução do autor)	45
Figura 17 – Endereço e Entrega — visibilidade normal e semântica (reprodução do autor)	46

Sumário

1	INTRODUÇÃO	13
1.1	Motivação	14
1.2	Objetivos	16
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	<i>Design System</i>	19
2.1.1	Componentes	19
2.1.2	Princípios	20
2.1.3	Estudos de Caso	20
2.1.4	Acessibilidade	22
3	PROTÓTIPO	25
3.1	Flutter	26
3.2	Pacote	27
3.3	Arquitetura	28
3.3.1	Átomos	30
3.3.1.1	Cores	31
3.3.1.2	Tipografia	31
3.3.1.3	Raio de borda	32
3.3.1.4	Espaçamento	32
3.3.2	Calculadora de Contraste de Cor	32
3.3.3	Semântica	34
3.3.4	Moléculas	35
3.3.4.1	Botão de Ícone	35
3.3.4.2	Botão de Texto	36
3.3.4.3	Campo de Entrada de Texto	37
3.3.4.4	Imagem	37
3.3.4.5	Link de Texto	39
3.3.4.6	Snackbar	40
4	VALIDAÇÃO	41
4.1	Aplicação Desenvolvida	42
4.1.1	Páginas	42
4.1.1.1	Página de Detalhes de Produto (PDP)	43

4.1.1.2	Carrinho de Compras	44
4.1.1.3	Endereço e Entrega	45
4.1.2	Validação Manual	46
4.1.3	Validação de Ferramenta	47
4.1.4	Validação de Usuário	47
5	CONCLUSÃO	51
5.1	Trabalhos Futuros	52
	REFERÊNCIAS	53

1 Introdução

A revolução digital tornou aplicações para dispositivos móveis parte integral de nossas rotinas diárias, servindo como ferramentas de comunicação, entretenimento, trabalho e aprendizado. Nesse cenário, métodos de desenvolvimento dessas aplicações cresceram e popularizaram-se, com a implementação de *design systems* como um recurso valioso para conquistar interfaces e experiências de usuário consistentes, eficazes e escaláveis; esses sistemas são encapsulamento de um conjunto de padrões e diretrizes que permitem que os desenvolvedores criem interfaces uniformes entre diferentes aplicações (FROST, 2016).

A Organização Mundial da Saúde (OMS), em 2019, estimou que cerca de 2,2 bilhões de pessoas têm dificuldades visuais, 237 milhões com dificuldade moderada a severa, trazendo à tona a necessidade de uma experiência digital acessível. Em 2010, o Instituto Brasileiro de Geografia e Estatística (IBGE) aferiu que 6,5 milhões de brasileiros têm deficiência visual severa. Entretanto, a acessibilidade para usuários com deficiência visual continua sendo um problema significativo no desenvolvimento de sistemas digitais, móveis ou não. Mesmo com progresso considerável das diretrizes e guias de acessibilidade, bem como das tecnologias assistivas, como leitores de tela e sistemas de reconhecimento de voz, muitas aplicações seguem inacessíveis para uma parcela considerável da população mundial.

Design systems, na sua capacidade e potencial de padronizar o processo de visualização e desenvolvimento, podem ser instrumentos essenciais ao implementar princípios de acessibilidades ao longo de todo o ciclo de vida de uma aplicação. Ao integrar padrões de acessibilidade com o *design system*, desenvolvedores fomentam inclusão e melhoram de maneira geral a usabilidade de suas aplicações (STORY; MUELLER; MACE, 1998).

A união de *design systems* com padrões bem definidos de acessibilidade em aplicações móveis é uma área que asseguradamente pode ser melhor explorada. A intenção desse trabalho é endereçar os benefícios do desenvolvimento voltado à acessibilidade, especificamente para deficientes visuais, propondo uma maneira prática e eficaz de desenvolver experiências digitais mais inclusivas.

1.1 Motivação

Enquanto as tecnologias digitais revolucionaram a maneira com que a sociedade opera, a importância da acessibilidade nesse contexto ainda não foi completamente reconhecida e integrada. Hoje, o uso de aplicativos é parte integral das vidas de uma grande parcela da população; entretanto, mesmo as aplicações mais utilizadas no Brasil e no mundo falham em oferecer uma experiência satisfatória a usuários com algum tipo de deficiência visual.

De acordo com o Movimento Web Para Todos, organização nacional dedicada a promover acessibilidade digital, com apoio técnico do World Wide Web Consortium (W3C) Brasil e do Núcleo de Informação e Coordenação do Ponto BR (NIC, órgão do Comitê Gestor da Internet no Brasil), em 2021, os 2.369 aplicativos *Android* mais populares do Brasil, que alcançaram a marca de 10 milhões de downloads no país na *Play Store*, têm, em média, texto alternativo em menos de 15% das imagens, menos de 12% dos elementos rotulados corretamente e menos de 1% dos campos editáveis claramente descritos, conforme demonstra a Tabela 1.

Um ponto interessante, no entanto, é levantado quando observada mais profundamente a metodologia da pesquisa. A coleta detectou que boa parte dos aplicativos tem, pelo menos, alguma descrição de um dos elementos aferidos; aproximadamente 94,72% dos aplicativos têm pelo menos um texto alternativo localizado, seja em imagens, botões ou campos editáveis. Porém, quando verifica-se a totalidade das descrições, menos de 1% tem a maior parte (três quartos ou mais) dos elementos descritos, indicando uma clara lacuna de consistência na construção de componentes.

Tabela 1 – Dados coletados dos apps

% dos Apps	Descrição de Imagens	Botões com Rótulos	Campos Editáveis com Descrição
0 a 10%	57%	79%	98%
11 a 25%	28%	15%	1%
25 a 50%	10%	5%	1%
50 a 75%	2%	1%	<1%
75 a 100%	3%	<1%	<1%

Fonte: Movimento Web Para Todos (2021)

De maneira similar, a Web Accessibility in Mind (WebAIM), organização sem fins lucrativos parte do Instituto de Pesquisa para Deficiências da Universidade de Utah, dedicada também à promoção de acessibilidade digital, conduz uma auditoria anual de um milhão de páginas iniciais das aplicações mais acessadas do mundo. A Tabela 2 ilustra sua edição

mais recente, de 2023, na qual a organização trouxe o dado alarmante de que 97,4% das páginas iniciais têm falhas significativas e detectáveis das diretrizes Web Content Accessibility Guidelines (WCAG) 2.1, sendo as mais comuns o baixo contraste de texto, falta de texto alternativo para imagens, links sem destino ou quebrados e campos editáveis mal indicados. Problemas como esse afetam gravemente a habilidade de deficientes visuais, cegos ou não, interagirem com a internet ou plataformas digitais.

Tabela 2 – Falhas do WCAG nas páginas iniciais ao longo dos anos

Tipo de falha WCAG	2023	2022	2021	2020	2019
Baixo Contraste de Texto	83.6%	83.9%	86.4%	86.3%	85.3%
Imagens sem texto alternativo	58.2%	55.4%	60.6%	66.0%	68.0%
Links vazios ou quebrados	50.1%	49.7%	51.3%	59.9%	58.1%
Rótulos de campos de entrada ausentes	45.9%	46.1%	54.4%	53.8%	52.8%
Botões vazios	27.5%	27.2%	26.9%	28.7%	25.0%

Fonte: WebAIM Million (2023)

Mesmo que as legislações avancem no sentido de um mundo digital mais acessível, com oficializações de leis e atos como o Americans with Disabilities Act (ADA) (EUA, 1990) nos Estados Unidos, o European Accessibility Act (EAA) (UNIÃO EUROPEIA, 2019) na União Europeia, ou, no contexto brasileiro, a Normativa ABNT NBR 17060:2022 (BRASIL, 2022), uma significativa barreira de acessibilidade claramente permanece, sugerindo que somente a ação legal não é suficiente para alcançar-se um patamar satisfatório.

A motivação em desenvolver um *design system* com foco intenso em acessibilidade tem duas principais vertentes. Por um lado, é uma maneira direta de demonstrar como melhorar, com tecnologias atuais, o estado da acessibilidade digital; por outro, busca fomentar uma cultura que herda, mesmo a nível de metodologia, valores que incorporem acessibilidade, promovendo inclusão e diversidade em espaços digitais. Um mundo digital mais acessível não

é benéfico somente àqueles que mais necessitam, mas enriquece a experiência de usuário de todos.

1.2 Objetivos

O objetivo geral deste trabalho é explorar o potencial do uso de *design systems* como ferramenta para melhorar a experiência de usuários com deficiência visual em aplicativos móveis. Esse objetivo é motivado pela ideia de diminuir a distância entre o crescimento rápido da tecnologia e dos desafios enfrentados por pessoas com dificuldade de visão ao acessarem conteúdo digital.

Mais especificamente, os objetivos deste estudo são:

1. Entender e conceituar *design systems* — o primeiro objetivo é crucial porque estabelece a fundação para o entendimento de todo o trabalho. É importante entender o que são *design systems*, como podem ser estruturados e os motivos pelos quais podem ser uma maneira sustentável e eficaz de produzir aplicações. O objetivo é demonstrar o potencial que a ferramenta pode ter ao incorporar padrões de acessibilidade, assegurando que produtos construídos com esses sistemas sejam mais acessíveis por definição.
2. Investigar o atual estado da acessibilidade digital para deficientes visuais — é necessário explorar o estado da arte das recomendações de acessibilidade. Essa exploração envolve examinar como grandes empresas tratam acessibilidade em seus produtos e serviços, conduzindo uma análise crítica das soluções existentes.
3. Desenvolver um *design system* acessível — esse objetivo será baseado nas informações observadas nos pontos anteriores. A ideia aqui é projetar e prototipar um *design system* que seja inerentemente acessível, com as cinco principais falhas das diretrizes WCAG apontadas pelo estudo Million do instituto WebAIM como fatores guia. O sistema será implementado de maneira a assegurar que o conteúdo criado é operável, entendível, e, sobretudo, robusto no que diz respeito à acessibilidade visual, alinhando os principais conceitos da WCAG, referência mundial nesse quesito.
4. Validar o *design system* proposto — uma jornada comum de fluxo de aplicações móveis será implementada utilizando o sistema desenvolvido, sendo testada por um usuário deficiente visual, coletando dados, opiniões e sentimentos acerca da experiência de uso, de modo a comparar com o existente no mercado. As informações coletadas serão utilizadas de maneira a refinar o sistema, buscando maior efetividade e impacto possíveis.

Ao dispor desses objetivos, este trabalho tenta contribuir à discussão e ao desenvolvimento da acessibilidade digital, provendo uma solução prática que melhore a experiência do usuário. É intenção do trabalho que essa tese não só traga uma ideia sobre inclusão, mas que inspire futura exploração e inovação no campo.

2 Fundamentação Teórica

2.1 *Design System*

Design Systems são coleções de componentes, constantes, guias e princípios que permitem designers e desenvolvedores a criar interfaces e experiências de usuário consistentes e eficientes entre diversas aplicações e plataformas; provendo um framework de trabalho para criar componentes modulares e reutilizáveis, que podem ser facilmente combinados para criar novas interfaces e fluxos. Ao usar essa metodologia, desenvolvedores podem acelerar o processo de codificação, reduzir inconsistências e melhorar, de forma geral, a usabilidade de suas aplicações (BROWN, 2020).

No contexto de aplicativos móveis, como demonstra a Figura 1, *design systems* podem ser particularmente úteis devido às características únicas de celulares ou dispositivos de mão, como espaço de tela limitado, resoluções e brilhos de tela variantes e a necessidade de rápida navegação. Os sistemas desenvolvidos podem incluir componentes e guias de tipografia, esquema de cor, iconografia, disposição e traçado de telas e interações. Ao usar um *design system* bem definido, as aplicações tendem a ser mais consistentes com as regras e princípios que o definem.

Design systems melhoram a eficácia do desenvolvedor ao permitir reutilização de componentes predefinidos, reduzindo a necessidade da construção de código customizado e permitindo foco maior na funcionalidade da aplicação (BROWN, 2020). Ao padronizar a linguagem de programação com a de design, via um sistema, é possível projetar soluções de interface mais rapidamente (SUAREZ et al., 2019). Adicionalmente, ao utilizar um *design system* robusto, reduz-se significativamente o risco da introdução de novas falhas, uma vez que parte significativa da aplicação será importada da biblioteca de componentes, e assegura-se que a aplicação está de acordo com as especificações desejadas.

2.1.1 Componentes

Os blocos fundamentais de um *design system* são seus componentes. Consistem de elementos customizáveis e reutilizáveis, como botões, campos de texto, de seleção, e a mais diversa gama de componentes, de acordo com a aplicação a ser desenvolvida. Idealmente, esses são construídos utilizando código conciso e claro, frequentemente disponibilizados como uma biblioteca a ser importada pelos desenvolvedores nos seus projetos.

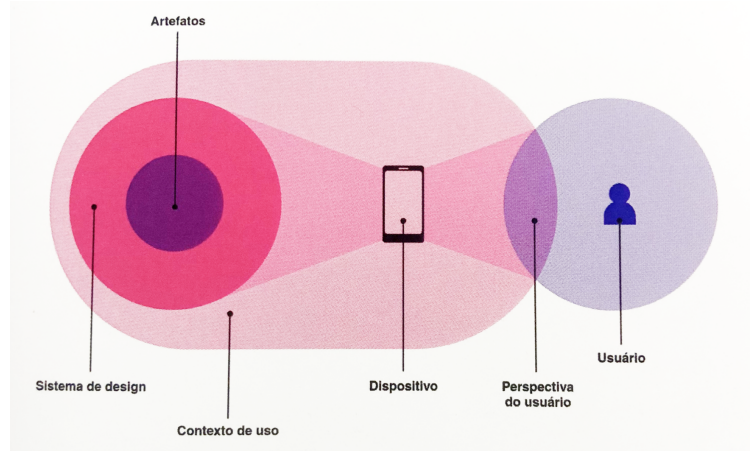


Figura 1 – Visibilidade Interação *Design System* (NZONGO, 2022)

Cada componente deve ser construído para funcionar de maneira eficaz individualmente ou em combinação com outros elementos. Uma biblioteca propriamente implementada e aproveitada deve sempre levar a um melhor desempenho do time de desenvolvimento e uma experiência de produto mais coerente (FROST, 2016).

2.1.2 Princípios

Cada *design system* é guiado por um grupo de princípios fundamentais. Consistência é chave, devendo ser foco principal do desenvolvimento de um novo sistema: elementos devem parecer e comportar-se de maneira similar ao longo de toda e qualquer aplicação desenvolvida com determinado *design system*, não só melhorando a experiência do usuário, mas promovendo uma curva de aprendizado e familiaridade com o sistema (NZONGO, 2022). Não apenas em design, mas também em código, é importante desenvolver e manter padrões.

Eficácia é vital, tanto em desenho como em implementação, principalmente quando aliada à flexibilidade, adaptando-se a novos contextos, necessidades do usuário e atualizações constantes em tecnologia, permitindo que os componentes reutilizáveis não só acelerem e facilitem o trabalho, mas diminuam drasticamente o risco de retrabalho (SUAREZ et al., 2019).

Outros princípios podem variar de acordo com o objetivo de cada *design system*, mas podem incluir simplicidade, escalabilidade e inclusividade, foco principal deste trabalho.

2.1.3 Estudos de Caso

O desenvolvimento e manutenção de um *design system* requer planejamento cuidadoso, organização e comprometimento entre duas principais partes: designers e desenvolve-

dores. É comum que, em grandes empreitadas, exista um time dedicado e responsável para a implementação e atualização constante do sistema, assegurando que seus princípios e componentes estejam consistentemente alinhados e aplicados dentre as diversas etapas dos projetos que o utilizam (KHOLMATOVA, 2017).

Um *design system* não é uma entidade estática, mas um produto em evolução que precisa se adaptar às necessidades do usuário, avanços tecnológicos e objetivos da aplicação. Requer-se atenção e atualizações constantes, com componentes e padrões sendo adicionados, alterados, ou removidos de acordo com as aplicações e respostas dos usuários (SAARINEN, 2016). Idealmente, a documentação é também mantida atualizada, servindo como histórico e repositório da biblioteca.

Alguns *design systems* são renomados e reconhecidos por terem significativamente impactado a indústria digital, e possuem suas particularidades, de acordo com a maneira que são utilizados:

1. Material Design (Google) — sistema abrangente, combina teoria, recursos, ferramentas e componentes, ajudando desenvolvedores e times de produto a criarem experiências digitais. Oferece guias em cores, layout, tipografia, animação, interação e som entre diversas plataformas. O Material Design é o *design system* de maior uso no mercado, adotado por uma gama de aplicativos no ecossistema do Google, provendo um caminho consistente para mais de 2,5 bilhões de usuários Android ativos (GOOGLE, 2023).
2. Human Interface Guidelines (Apple) — provê um detalhado e completo guia de princípios de design para a criação de interfaces em plataformas da Apple, incluindo iOS, iPadOS, macOS e watchOS. Com mais de 1,5 bilhão de usuários ativos mundialmente, a adoção dessas guias é intuitiva e familiar aos usuários da Apple (APPLE, 2023).
3. Fluent Design (Microsoft) — introduz diretrizes para a criação de aplicações intuitivas e performáticas em plataformas Windows. Com mais de 1 bilhão de usuários Windows 10 ativos e recentemente atualizado para sua segunda versão, o Fluent 2, o sistema permite experiência contínua entre dispositivos distintos, criando um ambiente familiar mas inovador (MICROSOFT, 2023).
4. Polaris (Shopify) — com o objetivo de criar uma experiência similar e consistente entre todas as experiências dos usuários, seja vendedor ou comprador, até 2021 mais de 1 milhão de negócios em 175 países utilizaram o sistema (SHOPIFY, 2023).
5. Lightning Design System (Salesforce) — *framework* agnóstico a plataforma utilizado para construir telas consistentes. Introduziu o conceito de design tokens, permitindo

uma maior customização de pontos individuais da aplicação enquanto ainda mantendo padrões visuais. Salesforce, com seu design system, dá suporte a mais de 150.000 negócios (SALESFORCE, 2023).

6. Carbon Design System (IBM) — *design system* de código aberto para produtos e experiências da IBM. Apresenta uma biblioteca abrangente e completa de componentes, padrões e guias que trabalham em conjunto para facilitar o desenvolvimento eficiente. O Carbon é utilizado por uma vasta gama de produtos, incluindo milhares de funcionalidades utilizadas por mais de 95% das empresas da lista Fortune 500 (IBM, 2023).

2.1.4 Acessibilidade

Incorporar padrões e boas práticas de acessibilidade em um *design system* promove acessibilidade em um nível fundamental durante todo o processo de desenvolvimento de uma aplicação, evitando que venha somente como uma preocupação futura.

Ter acessibilidade como princípio guia designers e desenvolvedores no caminho da criação de produtos digitais que podem ser usados por todos; essa preocupação embutida no *design system* trabalhado diminui drasticamente a chance do desenvolvimento de funcionalidades inacessíveis, uma vez que os componentes mais básicos, os blocos de construção de cada fluxo do aplicativo, já entregam soluções acessíveis.

De toda maneira, manter a acessibilidade em um sistema requer atualizações regulares e melhorias constantes, assim como um entendimento dos princípios de inclusão por parte dos times de desenvolvimento.

Os *design systems* previamente expostos incluem as seguintes diretrizes sobre acessibilidade:

1. Material Design (Google) — oferece sugestões em diversos aspectos, como contraste de cor, tamanhos de tipografia, tamanhos de áreas para telas de toque e mais. As diretrizes também ressaltam a importância de fornecer informações adicionais para tecnologias assistivas em todas as mídias, incluindo texto alternativo para imagens e transcrição para vídeos. Traz apontamentos específicos acerca de como tratar indivíduos com deficiências de visão de cores, como daltonismo, utilizando padrões e textos alternativos em adição às cores para prover informações (GOOGLE, 2023).
2. Human Interface Guidelines (Apple) — apresenta recursos extensivos em acessibilidade, incluindo guias para o desenvolvimento com foco em diversos tipos de deficiências, sempre alavancando as ferramentas já existentes nos sistemas operacionais do ambiente

Apple, como a criação de aplicativos que suportem tamanhos de fonte diferentes, garantindo que o usuário possa ajustar o tamanho do texto nas configurações do próprio dispositivo. A Apple também reforça a ideia de usar suficiente contraste de cor e evitar o uso de coloração como a única maneira de passar uma informação (APPLE, 2023).

3. Fluent Design (Microsoft) — oferece sugestões específicas em como projetar sites e aplicações acessíveis, com foco em cor, tipografia e componentes interativos. Traz diretrizes para criar conteúdo web acessível, com guias para a estruturação de elementos semânticos HTML5 que facilitem o entendimento de tecnologias assistivas (MICROSOFT, 2023).
4. Polaris (Shopify) — inclui um conjunto abrangente de diretrizes para a construção de layouts de seus usuários com foco em acessibilidade. Foca na importância de acessibilidade por teclado, priorizando que todas as funcionalidades possam ser acessadas somente com o uso do teclado. As diretrizes também cobrem a escrita de conteúdo acessível, incluindo a criação de textos alternativos efetivos e a utilização de tamanhos diferentes de texto para a estruturação do conteúdo (SHOPIFY, 2023).
5. Lightning Design System (Salesforce) — o sistema apresenta instruções detalhadas para a utilização dos seus componentes de maneira acessível, englobando cargos, estados e propriedades que devem ser incluídos quando da utilização de um componente. Também provê uma lista de pontos a serem checados como requerimento para o desenvolvimento correto de um componente (SALESFORCE, 2023).
6. Carbon Design System (IBM) — reforça a importância de suprir os requisitos dos padrões WCAG 2.1 e oferece um guia para como atingir esse objetivo. As diretrizes cobrem diversos aspectos de acessibilidade, desde cor e tipografia, até interação e planejamento de animações. A proposta da IBM se destaca na quantidade de detalhes na criação de campos de texto e tabelas, dois campos frequentemente problemáticos na área de acessibilidade digital (IBM, 2023).

É importante perceber que, apesar dos *design systems* mais conhecidos e utilizados do mundo oferecerem guias, como demonstrado anteriormente, poucos implementam de maneira prática recursos de acessibilidade nos seus componentes base. Essa lacuna passa a responsabilidade para os desenvolvedores, que frequentemente não possuem o conhecimento ou recursos necessários para aderir de maneira coesa e eficaz às diretrizes.

3 Protótipo

A fim de investigar o potencial, avaliando viabilidade e eficácia, de um *design system* como solução de acessibilidade para deficientes visuais em aplicativos móveis, propõe-se a construção de um protótipo: um *design system* implementado como um pacote Flutter, buscando empregar princípios de acessibilidade a seus componentes de maneira obrigatória e preemptiva, melhorando a usabilidade geral para usuários com acuidade visual variada, utilizando de um sistema bem definido.

No cerne do protótipo está a integração de ferramentas de acessibilidade simples e poderosas. Essas funcionalidades, incluídas nos componentes básicos, planejam manter os guias estabelecidos pela WCAG, focando em solucionar os cinco problemas mais comuns de acessibilidade apontados pelo projeto Million da WebAIM.

Ao incorporar recursos de acessibilidade no nível de componente, o protótipo tem como objetivo fornecer, no mínimo, recursos básicos de acessibilidade, estabelecendo um esboço base acessível como padrão, à medida que essas características se propagam aos níveis mais altos das aplicações.

Esse desenvolvimento, no entanto, não sugere que a implementação do *design system* por si só resolverá todos os problemas de acessibilidade. Cada aplicação, apesar de utilizar o sistema proposto, ainda deverá passar por testes e potenciais adaptações rigorosas para atender às variadas necessidades da sua base de usuários focal. O sistema proposto serve como um passo fundamental e crucial em direção à inclusão, provendo uma base robusta para desenvolvedores trabalharem e expandirem.

Mesmo com o sistema como um ponto de início, é fundamental reconhecer a necessidade de melhorias e adaptações contínuas. O objetivo, por esse contexto, se alinha com os princípios do Design Universal, focando na criação de um produto inclusivo utilizável por todos, sem a necessidade de separações ou abordagens especiais para grupos específicos de usuários (STORY; MUELLER; MACE, 1998). Esta abordagem inclusiva garante que as aplicações desenvolvidas sejam acessíveis, adaptáveis e escaláveis.

3.1 Flutter

Flutter é um *toolkit* de *User Interface* (UI) de código aberto desenvolvido pelo Google que permite que desenvolvedores compilem nativamente suas aplicações para dispositivos móveis, *web*, e plataformas *desktop* a partir de uma base de código única, como demonstra a Figura 2. Ele introduz uma nova visão de construção de interfaces de usuário, enfatizando customização, performance e produtividade (FLUTTER, 2023). O *framework* utiliza a linguagem de programação Dart, que oferece benefícios como segurança a nulos, tipagem forte, e concorrência objetiva. Isso leva a um código mais previsível, reduzindo a chance de encontrar causas comuns de erros e travamentos em aplicações móveis (DART, 2023).

No centro da arquitetura do Flutter está a filosofia "tudo é um componente". Os blocos básicos da construção de uma aplicação Flutter são componentes que descrevem como sua aparência e funcionalidade devem se comportar, frente às atuais configurações e estados. Esse paradigma de design permite interfaces altamente dinâmicas e responsivas, com uma extensa seleção de componentes pré-construídos fornecidos que permitem desenvolvimento veloz. O protótipo proposto neste trabalho faz uso constante desses componentes, alterando e adicionando a eles funcionalidades de acessibilidade (FLUTTER, 2023).

O Flutter provou ser uma escolha adequada para o desenvolvimento do *design system* proposto por uma série de razões. Primeiramente, o *framework* é ativamente atualizado, mantendo-se no estado da arte da tecnologia, demonstrando o compromisso da sua comunidade em adereçar tendências emergentes, juntamente de correções de erros e melhorias nas ferramentas já existentes. Isso se alinha com o objetivo deste projeto de criar um sistema relevante e moderno, que se mantenha eficaz enquanto o cenário digital evolui.

Além disso, o Flutter oferece uma gama de ferramentas de acessibilidade nativas, que permitem desenvolver aplicações inclusivas por padrão, provendo suporte a vários tamanhos de fonte, leitores de tela e contraste suficiente, essenciais para usuários com deficiência visual. Esse suporte nativo complementa de maneira chave as metas do protótipo, almejando incorporar elementos de acessibilidade básica como recursos centrais.

Talvez ainda mais crucial, a característica multi-plataforma do Flutter é uma vantagem significativa, particularmente quando desenvolvendo um *design system*. Dado que o sistema almeja assegurar consistência visual e funcional entre diferentes interfaces, a capacidade de aplicar essas regras e definições por múltiplos dispositivos é valiosa. Esse recurso proporciona, inclusive, capacidade de estender as aplicações além das plataformas móveis; enquanto o foco primário são celulares, a característica multi-plataforma do Flutter significa que esse *design system* poderia ser facilmente adaptado e utilizado em outros dispositivos, expandindo o alcance e o impacto.

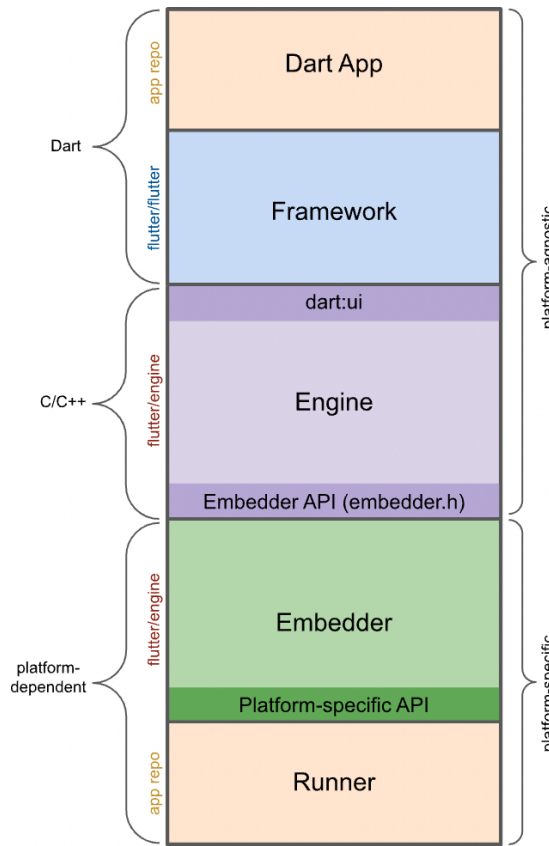


Figura 2 – Anatomia de uma aplicação (FLUTTER, 2023)

3.2 Pacote

No contexto de desenvolvimento de software, um pacote é um módulo reutilizável que contém código agrupado de maneira lógica. Pacotes encapsulam funcionalidades que podem ser facilmente importadas e usadas em diferentes projetos, promovendo a reutilização de código, permitindo que desenvolvedores compartilhem e utilizem códigos uns dos outros de forma padronizada. Pacotes são ferramentas cruciais para gerenciar ecossistemas de software complexos e ajudar na manutenção e evolução eficiente de código, sendo ferramentas importantes para o desenvolvimento de aplicações modernas (ESPINHA; ZAIDMAN; GROSS, 2014).

No âmbito do Flutter, um pacote é um trecho de código Dart reutilizável que pode ser integrado a um aplicativo; o pacote pode conter bibliotecas, componentes e outros recursos. O Flutter é extremamente eficaz em gerenciar pacotes, contando com uma vasta biblioteca de código aberto onde todos podem submeter seus recursos e contribuir para a expansão da gama de pacotes disponíveis. O sistema de pacotes do *framework* é reconhecido, em comparação de paradigmas de desenvolvimento de aplicativos móveis multi-plataforma, como um fator significativo em seu sucesso contra estruturas concorrentes como o React Native,

ressaltando sua capacidade de aprimorar o compartilhamento de código e melhorar a eficácia do desenvolvimento em geral (YOU; HU, 2021).

A criação do *design system* como um pacote Flutter se alinha perfeitamente com os objetivos do estudo. Desta forma, o protótipo pode ser facilmente integrado a vários aplicativos Flutter, provendo suas funcionalidades, componentes e recursos de acessibilidade. Isso permite que os desenvolvedores construam sobre uma base flexível e acessível, e customizem suas aplicações de acordo com seus requisitos específicos, promovendo a criação de produtos digitais mais inclusivos.

3.3 Arquitetura

A arquitetura do *design system* proposto é baseada na metodologia Design Atômico, uma abordagem amplamente reconhecida para a criação de componentes de UI, proposta inicialmente por Brad Frost em 2013, e constantemente trabalhada pelo autor desde então. O Design Atômico oferece uma estrutura sistemática e hierárquica que promove consistência, escalabilidade e capacidade de manutenção em projetos de *design system*. A metodologia divide os componentes de interface em cinco grupos: átomos, moléculas, organismos, modelos e páginas.

No nível atômico, o sistema abrange os blocos de construção fundamentais, incluindo cores, espaçamento, tipografia e raio de borda. Os átomos representam os menores e mais básicos elementos do sistema; ao defini-los, o protótipo garante que aspectos essenciais de acessibilidade como alto contraste de cor e textos legíveis, por exemplo, sejam levados em conta (FROST, 2016).

O próximo nível da hierarquia são as moléculas. Grupos de átomos combinados e manipulados para formar componentes de interface funcionais e reutilizáveis. Moléculas podem incluir elementos como botões, campos de texto ou fichas de informações. Neste protótipo, as moléculas serão construídas utilizando os átomos predefinidos como padrão, assegurando consistência visual e funcional para o usuário. Por exemplo, uma molécula de botão pode combinar átomos de texto, cor e raio de borda para criar um elemento interativo visual distinguível e, nesse caso, acessível.

Organismos são o terceiro nível da hierarquia e são compostos por múltiplas moléculas, para formar elementos maiores e mais complexos de interface. Esses organismos comumente representam cabeçalhos de página ou seções inteiras de uma aplicação.

Os modelos são o próximo, e quarto, nível da hierarquia do Design Atômico, e envolvem um arranjo de organismos, moléculas e átomos a fim de formar a estrutura de uma

página. Modelos definem a organização geral de uma aplicação, garantindo que suas regras e conceitos estabelecidos se mantenham prioritários. Ao criar modelos respeitando a hierarquia, um *design system* permite que desenvolvedores construam aplicações móveis que atendam a uma ampla gama de usuários, enquanto ainda mantêm consistência visual (NZONGO, 2022).

Páginas são o nível mais alto da hierarquia da metodologia, e representam o produto final, os modelos preenchidos com conteúdo real. No contexto do protótipo proposto, a criação de páginas - e, portanto, experiências - acessíveis é o resultado da combinação de átomos, moléculas, organismos e modelos ao longo de todo o processo de desenvolvimento, como ilustra a Figura 3.

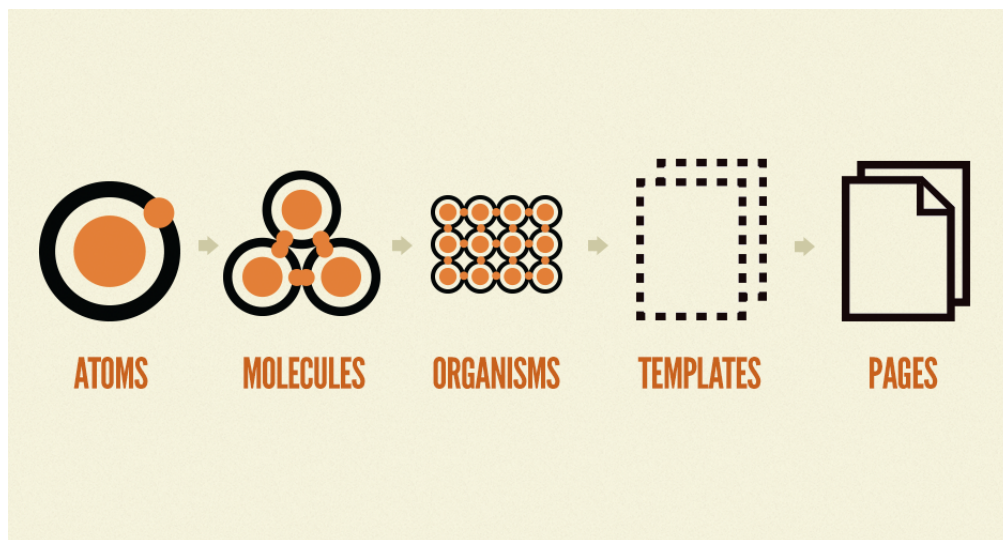


Figura 3 – Arquitetura da metodologia (FROST, 2016)

O *design system* desenvolvido foca especialmente nos dois primeiros níveis fundamentais do Design Atômico, átomos e moléculas. Esses blocos fundamentais estabelecem a base para qualquer aplicação, e a incorporação de recursos de acessibilidade no estágio inicial garante que esses recursos sejam levados adiante à medida que o sistema escala. Os componentes básicos criados no nível atômico e molecular servem como a base sobre a qual estruturas mais complexas podem ser construídas, imbuindo-as automaticamente de um nível inerente de acessibilidade.

No contexto deste projeto, que é implementado como um pacote Flutter, o *design system* pode ser integrado a diferentes aplicações com objetivos diversos. O pacote fornece os componentes atômicos e moleculares; entretanto, a responsabilidade pela construção de organismos, modelos e páginas é dos aplicativos individuais que importam o sistema. Essa arquitetura facilita a flexibilidade, permitindo que as plataformas criem componentes mais complexos adequados aos seus requisitos específicos e, ao mesmo tempo, aproveitem os re-

cursos fundamentais de acessibilidade fornecidos.

Esse modelo, no qual o *design system* se concentra nos componentes fundamentais e deixa a construção dos mais complexos para os programas individuais, é uma prática comum do desenvolvimento de software moderno. Muitos *design systems* importantes, como o Material Design do Google e o Human Interface Guidelines da Apple, oferecem uma estrutura semelhante.

O Material Design, por exemplo, fornece diretrizes para escolhas de estilo como cores, tipografia e ícones - semelhantes aos nossos átomos - e componentes simples - nossas moléculas - como botões, cartões e controles de quantidade (GOOGLE, 2023). Dessa forma, permite que desenvolvedores construam suas aplicações utilizando e alavancando as estruturas providas.

De maneira similar, o Human Interface Guidelines da Apple fornece os elementos essenciais, como rótulos e botões, bem como orientações para seu uso (APPLE, 2023). Os desenvolvedores trabalham empregando esses recursos fornecidos para criar interfaces mais complexas adequadas às necessidades específicas de suas aplicações. Ao fornecer os elementos essenciais, esses *design systems* garantem que suas regras fundamentais sejam incluídas da melhor maneira possível, desprendendo-se da complexidade final de uma tela ou recurso complexo.

As pesquisas acadêmicas também corroboram essa abordagem: em seu livro *Design Systems: A practical guide to creating design languages for digital products*, Alla Kholmatova enfatiza que os sistemas devem focar em oferecer um conjunto compartilhado e integrado de componentes, como átomos e moléculas (KHOLMATOVA, 2017).

Dessa forma, o pacote proposto se alinha aos padrões da indústria, fornecendo uma camada fundamental de componentes, juntamente com recursos de acessibilidade a eles incorporados, dos quais aplicações podem utilizar e expandir.

3.3.1 Átomos

Na raiz da arquitetura do *design system* desenvolvido estão os átomos; formando os blocos mais simples, esses elementos estabelecem uma base sólida para escalar a componentes mais complexos. Ao introduzir recursos chave de acessibilidade nesses átomos, o sistema estabelece um padrão para a criação de experiências inclusivas e visualmente atrativas. Esta seção descreve os quatro átomos primários desenvolvidos no presente protótipo: Cores, Tipografia, Raio de Borda e Espaçamento.

3.3.1.1 Cores

Serve como a paleta primária de cores, oferecendo uma gama de cores de alto contraste e visualmente distinguíveis. A paleta inclui cor primária, secundária, de fundação, de fundo e de erro, bem como suas contrapartes, como demonstra a Figura 4. O foco no contraste permite que usuários com diferentes graus de acuidade visual possam discernir claramente os elementos da interface do usuário, melhorando a acessibilidade geral. Entretanto, embora essas definições de cores forneçam uma base sólida para o sistema, os projetos que importarem o pacote não estão limitados a ela; o sistema oferece e encoraja a flexibilidade, permitindo que o desenvolvedor modifique ou adicione essas definições para melhor atender às necessidades de suas aplicações.

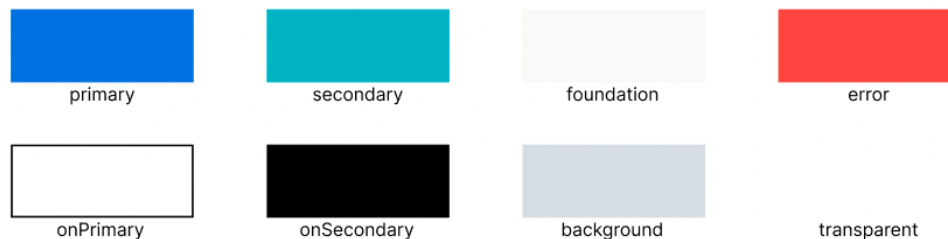


Figura 4 – Cores do sistema proposto (reprodução do autor)

3.3.1.2 Tipografia

Responsável inicial por garantir que o conteúdo apresentado é legível, esse átomo estabelece uma série de estilos de texto, demonstrados na Figura 5, com tamanhos e pesos de fonte adaptáveis e diferentes espaçamentos, a fim de melhorar a legibilidade. A tipografia almeja não apenas tornar o conteúdo facilmente compreensível, mas também criar uma experiência visualmente agradável ao estabelecer padrões de texto, evitando inconsistências. Reconhecendo as diferentes necessidades tipográficas de diferentes aplicativos, o sistema permite alterações, redefinições e ajustes a esses estilos de texto predefinidos.

title **subtitle**
highlight
body normal caption

Figura 5 – Tipografia do sistema proposto (reprodução do autor)

3.3.1.3 Raio de borda

A uniformidade e o apelo visual são as preocupações centrais desse átomo. A Figura 6 exibe o conjunto de valores de raio de borda fornecidos que podem ser aplicados a vários elementos da interface do usuário. A adoção desses valores consistentes garante uma aparência uniforme em todo aplicativo, criando familiaridade visual ao usuário com as funcionalidades.



Figura 6 – Raios de borda do sistema proposto (reprodução do autor)

3.3.1.4 Espaçamento

O átomo de espaçamento se concentra em promover uma interface bem organizada e navegável. A Figura 7 exibe o conjunto de valores de espaçamento padrão que podem ser aplicados a vários elementos da interface do usuário garantindo consistência e usabilidade geral.



Figura 7 – Espaçamentos do sistema proposto (reprodução do autor)

3.3.2 Calculadora de Contraste de Cor

O desenvolvimento de uma aplicação móvel acessível requer foco afiado na legibilidade dos elementos da interface. Um dos aspectos mais importantes se refere ao contraste de cores entre elementos e como eles interagem visualmente. A WCAG, na edição 2.1 de seu guia, sugere um contraste mínimo de 4,5:1 para texto de tamanho normal e de 3:1 para textos maiores (W3C, 2018), a fim de garantir acessibilidade para usuários com diversos tipos de dificuldades de visão.

No *design system* proposto, uma calculadora de contraste entre cores foi desenvolvida, seguindo diligentemente as recomendações da WCAG. A calculadora executa a tarefa crítica de ajustar o contraste entre duas cores de modo que atenda ou ultrapasse a taxa de contraste

estabelecida (W3C, 2018); esse mecanismo desempenha um papel significativo em melhorar a legibilidade e acessibilidade geral para usuários com acuidade visual variável.

O algoritmo da calculadora calcula a luminância relativa de cada cor. A luminância relativa adota a definição fotométrica de luminância do ISO/CIE 23539:2023 (CIE, 2023), mas normaliza os valores de 1 a 100, para um branco de referência; similar à definição fotométrica, está associada à densidade do fluxo luminoso em uma determinada direção, que é a densidade do fluxo radiante pela função de luminosidade λ do *Commission Internationale de l'Eclairage* (CIE). O uso de valores relativos é conveniente em sistemas onde a reprodução absoluta de uma cor é impraticável, como no caso de telas de dispositivos eletrônicos.

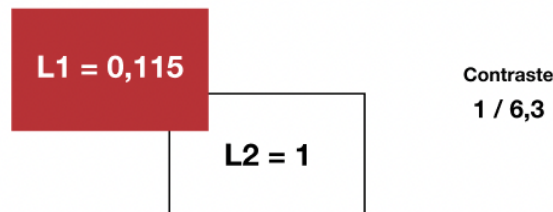


Figura 8 – Cálculo do contraste baseado em luminâncias (reprodução do autor)

Os componentes de cada cor no padrão RGB - vermelho, verde e azul - são transformados de acordo com as recomendações da WCAG para levar em consideração a relação não-linear entre a intensidade física de uma cor e a percebida pelo olho humano (STOKES et al., 1996). Esse processo de transformação ajuda a obter uma representação mais precisa de como os seres humanos percebem o brilho das cores.

Em seguida, o algoritmo computa a taxa de contraste comparando os valores de luminância obtidos das cores (W3C, 2018), conforme demonstra a Figura 8. Se o valor calculado for insuficiente, a função inicia um processo iterativo em que ajusta sistematicamente a luminância da cor principal. Isso continua até que a taxa de contraste seja igual ou superior ao padrão mínimo imposto pela WCAG, ou que a luminância da cor alterada atinja os valores-limite, 0 para preto e 1 para branco.

Essa decisão de ajustar automaticamente a luminância da cor é uma escolha de implementação feita com a intenção de maximizar a usabilidade e o impacto da calculadora de contraste desenvolvida. No entanto, o sistema pode ser facilmente modificado para alterar esse comportamento, se assim for desejado. Mudanças simples possíveis podem, ao invés de alterar a cor, acionar um erro de compilação ou somente registrar um aviso quando uma das cores calculadas não atender aos requisitos mínimos. Essa adaptabilidade oferece aos usuários do pacote a flexibilidade de usar o sistema proposto da maneira que melhor se adapte às suas necessidades.

3.3.3 Semântica

O conceito de semântica se refere ao significado ou à interpretação de algo; na computação, diz respeito a como uma aplicação transmite informações de significado e contexto para tecnologias (PRASAD; CHOPELLA, 2017); no caso deste projeto, assistivas. Para o Flutter, a semântica fornece uma maneira de anotar componentes com uma descrição de seu significado ou finalidade, além de manipular a maneira padrão com a qual o *framework* trabalha, que pode ser interpretada por tecnologias como leitores de tela, motores de busca, telas em braille ou outros softwares projetados para isso (FLUTTER, 2023).

A semântica é essencial para aprimorar a acessibilidade em aplicativos móveis. Ela permite que as tecnologias assistivas apresentem informações significativas aos usuários que, possivelmente, não consigam perceber ou interagir diretamente com os elementos visuais; sejam esses usuários deficientes visuais, com dificuldades motoras ou outras necessidades de acessibilidade.

O *design system* proposto aplica metodicamente anotações de semântica, a fim de garantir que cada componente seja o mais acessível e significativo possível para todos os usuários, fazendo valer das tecnologias presentes no Flutter. Por exemplo, na definição dos botões do protótipo, o componente de semântica é usado para prover um texto alternativo de descrição, que será lido pelo leitor de tela do sistema operacional rodando a aplicação; a semântica também informa o sistema de que o componente é - e deve se portar como - um botão, deixando clara sua interação desejada.

Além disso, o sistema semântico do Flutter foi criado para ser extensível e personalizado, permitindo ações como a união ou exclusão de grupos semânticos. Os desenvolvedores podem adicionar anotações conforme necessário ou sobrescrever as existentes, permitindo que sejam criados aplicativos altamente acessíveis e adaptados às necessidades específicas de seus usuários.

Além de fornecer anotações para componentes, o *framework* também organiza essas anotações em uma estrutura conhecida como Árvore Semântica. A árvore é uma representação lógica de um aplicativo Flutter, em que cada nodo corresponde a um componente com suas devidas informações semânticas; essa estrutura é utilizada por tecnologias assistivas para reunir informações e, acima de tudo, estruturar de maneira organizada a hierarquia de componentes da interface do usuário para leitores de tela.

3.3.4 Moléculas

Essa seção foca nos componentes específicos - ou moléculas, no modelo de Design Atômico - do protótipo desenvolvido. Cada componente é examinado, detalhando sua composição estrutural, como faz uso dos átomos nele contido, e como contribui para a experiência de acessibilidade dentro de uma aplicação que o utiliza.

Um grupo de componentes chave foi construído com o intuito de manter a consistência da interface de usuário; esses são elementos essenciais encontrados comumente na maioria das aplicações móveis (SHIRAZI et al., 2013), assegurando que, mesmo funcionando como um protótipo experimental, muitas das interações que usuários fazem nos fluxos usuais de aplicativos estejam cobertas.

3.3.4.1 Botão de Ícone

O componente customizável Botão de Ícone, ilustrado na Figura 9, se destaca como um componente chave da interface de usuários, comumente utilizado para uma gama de ações, e como um dos elementos onde a falta de acessibilidade é mais comum, segundo o projeto WebAIM Million 2023. Ao aceitar múltiplos parâmetros pelo seu construtor, o ícone oferece alto grau de personalização, tanto em termos de aparência como funcionalidade.

Uma de suas características pivotais é a incorporação do elemento de Semântica, que encapsula um botão padrão do Material Design (APPLE, 2023), facilitando a identificação certa do papel do botão pelos leitores de tela. Isso é feito aceitando um campo de texto alternativo como parâmetro, que é passado pela estrutura e será lido pelas tecnologias, levando ao usuário final informações essenciais sobre o papel do elemento na tela.

Visualmente, os parâmetros do construtor do objeto abrangem detalhes sobre as cores e tamanhos do ícone, do fundo e das bordas; todos opcionais com valores predefinidos dos átomos já estabelecidos. Essas opções permitem que os desenvolvedores modifiquem a aparência do botão, respeitando os limites estabelecidos pelo *design system* e pelas diretrizes de acessibilidade padrão; o tamanho mínimo é de 48x48 pixels, respeitando as diretrizes de ambos Android e iOS para elementos clicáveis. Como é padrão com elementos de toque, o componente também dá a liberdade de definir a função e o próprio ícone por meio de parâmetros, permitindo a flexibilidade necessária para implementações específicas distintas (NEIL, 2014).

No cerne do Botão de Ícone, encontramos a Calculadora de Contraste de Cor, que garante que o contraste entre o ícone e o fundo esteja alinhado, da melhor maneira possível, com as recomendações da WCAG, ajustando dinamicamente a luminância das cores até atingir uma relação de contraste aceitável ou limite.

Por baixo da superfície, o Botão de Ícone é um conjunto de componentes básicos do Flutter. O elemento de botão do Material Design forma o ícone, e o elemento de Texto do Flutter cria o rótulo a ser lido pelo leitor de telas. Ao abstrair a complexidade desses componentes de nível mais baixo, o *design system* oferece aos desenvolvedores um objeto simples, reutilizável e totalmente acessível.



Figura 9 – Botão de Ícone — visibilidade normal e semântica (reprodução do autor)

3.3.4.2 Botão de Texto

De maneira similar ao Botão de Ícone, o Botão de Texto é um componente básico extremamente utilizado, representado de maneira a levar em consideração acessibilidade, além de funcionalidade e estética.

Diretamente, o Botão de Texto emprega o componente nativo do Flutter de mesma função. Com um grupo de parâmetros customizados em seu construtor, os projetos que adotarem o sistema proposto podem facilmente modificar o funcionamento e a aparência do Botão de Texto para encaixar-se em suas necessidades. Todos os parâmetros opcionais recebem, por padrão, valores predefinidos nos átomos, resultando em uma experiência familiar.

A Figura 10 mostra que, novamente, o elemento de Semântica é empregado, recebendo um texto alternativo obrigatório, que será lido pelo leitor de telas em uso, informando usuários sobre sua posição e função.

As cores do texto e do fundo podem ser definidas, e passam pela Calculadora de Contraste de Cor, que ajusta as cores buscando aderir às diretrizes da WCAG. O estilo do texto, assim como da borda, pode ser manipulado para oferecer maior liberdade e adaptabilidade visual ao componente.

Em termos de interatividade, o botão recebe por parâmetro a função a ser executada quando é clicado, permitindo que essa molécula execute uma multitude de ações dentro de uma mesma aplicação.



Figura 10 – Botão de Texto — visibilidade normal e semântica (reprodução do autor)

3.3.4.3 Campo de Entrada de Texto

A molécula Campo de Entrada de Texto, apresentada na Figura 11 ressalta a abordagem simples e direta do sistema proposto. Essa molécula é uma representação de um campo de entrada padrão do *toolkit* Flutter, um elemento de interface comum em aplicativos, utilizado para entrada de dados textuais dos usuários.

Por ser construído utilizando elementos já conhecidos, o componente é inerentemente familiar para desenvolvedores que já atuaram com objetos similares. Seu construtor oferece uma série de parâmetros, permitindo personalização de função e aparência, enquanto mantém as características essenciais de acessibilidade; por padrão, todos os campos visuais recebem valores dos átomos já definidos.

O componente de Semântica é utilizado aqui para descrever a finalidade do Campo de Entrada de Texto. O texto alternativo obrigatório, quando lido em voz alta pelas tecnologias assistivas, informa aos usuários quais informações o Campo de Entrada de Texto selecionado espera receber (NEIL, 2014).

Em termos de estética e funcionalidade, os desenvolvedores podem controlar diversos aspectos. O tipo de teclado exibido quando o Campo de Entrada está em foco é passado por parâmetro, assim como o estilo do texto interno à caixa de entrada e a ação associada ao botão de concluído dos teclados e sistemas operacionais.

Há ainda parâmetros opcionais que reforçam o tipo de informação a ser digitada no componente, que podem ser exibidos de acordo com a necessidade, criando dicas textuais do tipo de dado esperado. A borda do Campo de Entrada de Texto muda de cor quando o mesmo é selecionado, mantendo as diretrizes de contraste, resultando numa sutil sugestão visual que visa aprimorar a experiência do usuário, indicando claramente a área de entrada de texto ativa.

3.3.4.4 Imagem

A molécula de Imagem é um componente simples que foi criado com base no componente nativo do Flutter, particularmente o construtor de imagens da *web*, que busca uma

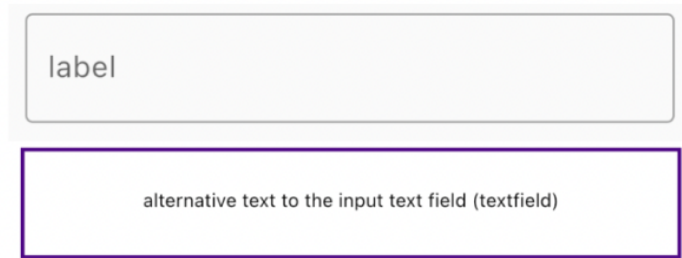


Figura 11 – Campo de Entrada de Texto — visibilidade normal e semântica (reprodução do autor)

gravura de um URL fornecido como parâmetro ao objeto. Isso possibilita a incorporação de conteúdo de imagem dinâmico e diversificado, podendo, inclusive, ser alterado em tempo de execução, se o URL de hospedagem for próprio.

A Figura 12 demonstra o principal recurso de acessibilidade incorporado nessa molécula, outra vez, a Semântica. Ao fornecer um texto alternativo obrigatório para a imagem, não há possibilidade de haver uma imagem sem descrição, segundo erro de acessibilidade mais comum nas telas iniciais das aplicações investigadas pelo projeto WebAIM Million, garantindo que o leitor de tela em uso reconheça e descreva a figura.

Além disso, as dimensões - altura e largura - da Imagem são controladas por meio de parâmetros opcionais, o que permite a adaptação visual do componente; se nenhuma dimensão for especificada, o componente assume o maior tamanho que pode ocupar, de acordo com a hierarquia de elementos onde está localizado.



Figura 12 – Imagem — visibilidade normal e semântica (reprodução do autor)

3.3.4.5 Link de Texto

Como elemento do cenário digital, os hipertextos desempenham um papel vital na conexão de conteúdo e direcionamento de fluxo de usuário (COSCARELLI, 2012). A molécula Link de Texto simplifica o processo de incorporação de acessibilidade a hiperlinks.

O Link de Texto aceita vários parâmetros, permitindo que os desenvolvedores personalizem a aparência e a função dos *links*. Outra vez, o elemento de Semântica é empregado, fornecendo texto alternativo ao endereço sendo clicado. Em termos de estilo, a aparência pode ser modificada utilizando parâmetros opcionais; por padrão, o hiperlink é estilizado com o convencional texto azul sublinhado, como apresenta a Figura 13. No entanto, isso pode ser modificado conforme necessário.

Um recurso de acessibilidade único do Link de Texto é a função de verificação de URL. Essa função realiza a tarefa essencial de validar a disponibilidade do endereço a ser acessado, terceira maior falha de acessibilidade apontada pelo projeto WebAIM Million.

Para fazer a verificação, o componente faz uma solicitação GET, o método HTTP padrão que solicita dados para o URL desejado. Quando um código de status afirmativo é recebido em resposta, isso significa uma interação bem-sucedida com o recurso da *web*. Em outras palavras, o servidor está operante e o URL fornecido aponta para uma página válida; esse é, inclusive, o processo padrão subjacente quando qualquer recurso da internet é acessado via navegador. Em caso de falha, uma mensagem de erro é devolvida à aplicação que implementa o Link de Texto, cabendo ao desenvolvedor decidir o que fazer com essa informação.

De maneira fundamental, o Link de Texto é construído utilizando componentes primários essenciais do Flutter: o detector de gestos - responsável por lançar a função de abertura do URL quando o texto é clicado - e o próprio texto, estilizado à maneira convencional de um hiperlink.

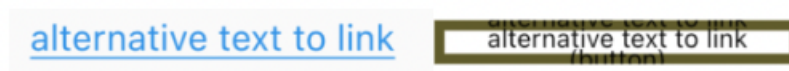


Figura 13 – Link de Texto — visibilidade normal e semântica (reprodução do autor)

3.3.4.6 SnackBar

Um desafio de acessibilidade frequentemente negligenciado é garantir que tecnologias assistivas entendam adequadamente quando elementos da interface de usuário aparecem dinamicamente na tela (NGUYEN et al., 2021). A Figura 14 demonstra que o SnackBar desenvolvido oferece uma solução para esse problema, fornecendo uma versão modificada do componente padrão do Flutter, usado para exibir mensagens breves e contextuais na parte inferior da tela.

Ao utilizar o atributo de região ativa do elemento de Semântica, o leitor de telas é instruído a priorizar o anúncio do SnackBar. Após a construção de um objeto SnackBar, uma função de anúncio da Semântica é imediatamente invocada, comunicando proativamente as informações presentes no componente, usualmente textos. Isso significa que, assim que o elemento é renderizado, a Árvore de Semântica o captura e transmite sua mensagem, fornecendo suporte consistente e instantâneo.

O construtor aprimora as funcionalidades com parâmetros opcionais, que oferecem a opção de botões acessíveis no SnackBar, permitindo que os usuários interajam com o componente, e alteração do tempo de exibição da mensagem. Utilizar uma ação obrigatória em um componente dinâmico não é recomendado: é encorajado o aumento do tempo de exibição ao utilizar esta metodologia de trabalho.

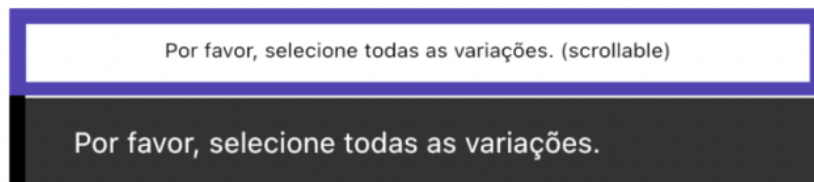


Figura 14 – SnackBar — visibilidade normal e semântica (reprodução do autor)

4 Validação

Validação é uma etapa essencial no desenvolvimento de qualquer sistema, oferecendo percepções acerca da funcionalidade e usabilidade do produto; sendo possível avaliar a efetividade da proposta e identificar potenciais falhas e pontos de melhoria. No contexto de aplicações móveis, particularmente voltadas à acessibilidade, validação é ainda mais crucial, uma vez que apenas a interação com o usuário final pode, de maneira assertiva, passar a capacidade do sistema ao incorporar as funcionalidades de acessibilidade desejadas.

O processo de validação no campo de acessibilidade é amplamente reconhecido como fundamental para o sucesso de qualquer produto de engenharia (MAROPOULOS; CEGLA-REK, 2010). Acessibilidade não deve ser levada como um processo separado, mas como uma qualidade inerente de todos os aspectos de um sistema (ABASCAL; NICOLLE, 2005), fator que só pode ser assegurado com testes e validações rigorosas.

Para validar o *design system* proposto neste estudo, uma aplicação demonstrativa foi desenvolvida: um aplicativo Flutter, que importa e utiliza os componentes e átomos do pacote *design system* para todas as suas funções, contendo alguns dos principais fluxos de um aplicativo de *e-commerce*. Esse gênero de aplicação foi escolhido devido sua popularidade e por oferecer uma excelente plataforma para testar uma variedade das moléculas implementadas em diferentes contextos.

Ao navegar pela aplicação, valendo-se da visibilidade de Árvore de Semântica que o Flutter oferece em modo de depuração, é possível avaliar o que vai ser exibido, fazendo uma primeira análise de usuário. Além disso, os sistemas operacionais nativos oferecem ferramentas de depuração de acessibilidade, que serão utilizadas para buscar erros e possíveis melhorias.

Na busca por validação, o teste definitivo do sistema vem da interação com um usuário muito especial: uma autoridade globalmente reconhecida em deficiências visuais, que dedicou sua vida a defender a acessibilidade e, mais importante ainda para esse teste, também é deficiente visual desde a infância. Volmir Raimondi é ex-presidente da União Latino-Americana de Cegos e ex-membro do Conselho Nacional de Saúde, fundador da Associação dos Deficientes Visuais de Bento Gonçalves, palestrante e participante de eventos ao redor de todo o mundo, que com sua experiência, tanto pessoal quanto profissional, traz uma profundidade inigualável de percepção para o processo de validação.

Ao entregar em mãos o aplicativo que utiliza rigorosamente as funcionalidades desenvolvidas no *design system*, foi possível obter um relato direto sobre o desempenho do sistema

para os usuários a que se destina, na forma de uma entrevista. Como alguém que dedicou sua vida à causa da acessibilidade, esse usuário pôde avaliar a aplicação de uma perspectiva excepcionalmente informada e relevante. O objetivo desse teste final é não apenas entender se o sistema é tecnicamente funcional, mas também se alcança seu objetivo geral de melhorar a usabilidade para pessoas com deficiência visual.

4.1 Aplicação Desenvolvida

O ramo de comércio eletrônico é um dos mais prevalentes domínios do cenário de aplicações móveis, e segue crescendo em franca expansão no Brasil (NIELSENIQ EBIT, 2023). No cerne desses aplicativos estão alguns momentos críticos de interface de usuário, com diferentes complexidades, mas com estrutura relativamente estabelecida.

Para validar o *design system* proposto, foi desenvolvido um protótipo de aplicação de comércio eletrônico que simula parte do fluxo padrão de uma compra digital. Essa abordagem foi motivada pelo objetivo de encaixar a maioria dos componentes desenvolvidos em um gênero de aplicativo comum, demonstrando sua compatibilidade e eficácia em cenários do mundo real (ZHANG; ADIPAT, 2005).

As seções a seguir discorrem acerca de cada estágio do fluxo do aplicativo explicando como os átomos e moléculas do sistema desenvolvido se encaixam na estrutura e contribuem para melhorar sua acessibilidade geral.

4.1.1 Páginas

No contexto da aplicação desenvolvida, as páginas são as entidades mais complexas e completam a representação da metodologia de Design Atômico. Cada página representa uma unidade funcional do aplicativo, gerenciando diversas interações do usuário e tarefas (FROST, 2016).

As páginas detalhadas a seguir são ilustrações válidas de como átomos, moléculas, organismos e modelos se unem para proporcionar uma experiência de usuário completa. Cada página é criada utilizando componentes reutilizáveis do *design system*, o que aumenta a consistência visual e a eficiência em todo o fluxo da aplicação.

4.1.1.1 Página de Detalhes de Produto (PDP)

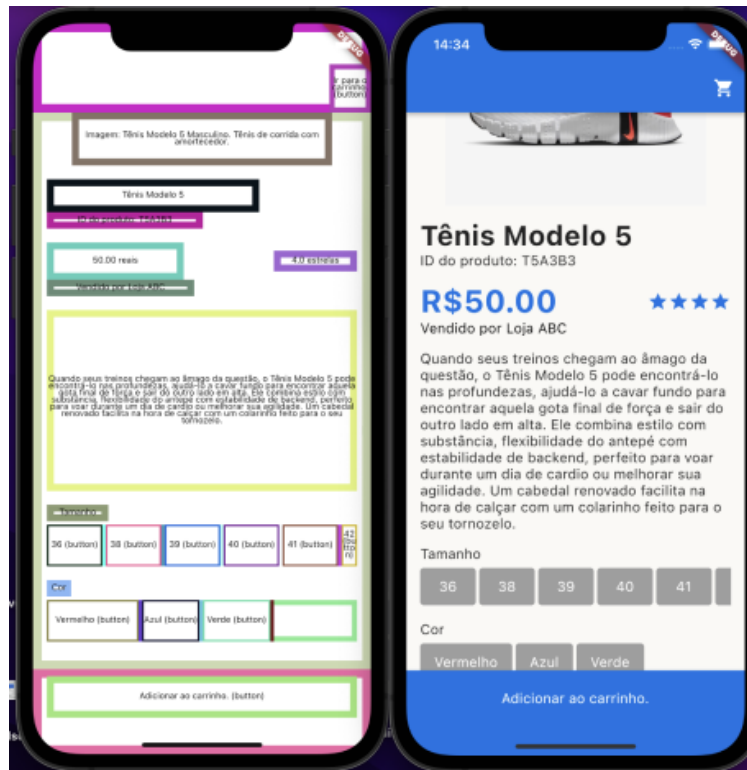


Figura 15 – PDP — visibilidade normal e semântica (reprodução do autor)

A PDP exerce função vital em qualquer aplicativo de comércio eletrônico. É onde o usuário recebe detalhes essenciais sobre um produto, permitindo-o tomar uma decisão informada de compra. De acordo com essa definição, a página desenvolvida foi pensada para ser altamente informativa e acessível, fazendo uso extensivo dos componentes do *design system*.

A Figura 15 demonstra que, em estilo tradicional, a tela inicia com uma imagem do produto no topo, fazendo uso do componente Imagem proposto; incluindo o campo de texto alternativo obrigatório, provendo informações essenciais sobre os aspectos visuais do produto.

Diretamente abaixo, há uma série de informações textuais relevantes: nome do produto, identificador, preço e descrição. Esses campos de texto se apoiam em átomos de Tipografia para garantir consistência e assegurar legibilidade. À direita dessas informações, existe um componente de avaliação, formado por estrelas que demonstram a nota do produto; esse componente foi envolto com Semântica, que transmite sua informação para as tecnologias assistivas.

Um componente chave de toda PDP é o seletor de variações, que permite ao usuário escolher detalhes específicos do produto, como, por exemplo, cor e tamanho. Esse componente é construído como um carrossel horizontal de Botões de Texto, automaticamente herdando

as qualidades acessíveis da molécula.

Finalmente, no final da tela, um Botão de Texto permite que usuários adicionem produtos ao seu carrinho de compras; a acessibilidade desse botão é garantida pelas características do componente já descritas. Adicionalmente, a ação do botão é validada para garantir que todas as variações tenham sido selecionadas antes de proceder, retornando a informação negativa com o uso do componente Snackbar proposto, que tem suas informações lidas proativamente pelo leitor de telas, demonstrando a preocupação da aplicação em garantir uma experiência agradável (NGUYEN et al., 2021).

Ao adicionar um produto ao carrinho de compras com sucesso, um diálogo é exibido e lido automaticamente pelas tecnologias assistivas, dando as opções de seguir o fluxo ou permanecer no produto.

4.1.1.2 Carrinho de Compras

A página do carrinho serve como um passo intermediário entre a exploração e seleção do produto e o processo de fechamento de compra, fornecendo informações gerais ao usuário acerca dos itens selecionados e permitindo fazer ajustes finais antes de proceder.

A estrutura da tela segue um estilo clássico de comércio eletrônico, conforme ilustra a Figura 16, e começa com uma barra superior que dispõe ao usuário informações essenciais sobre o estado da página atual. Na extrema direita, um Botão de Ícone que limpa o carrinho, com seu respectivo texto alternativo.

Subsequentemente, o corpo do carrinho apresenta uma lista dos produtos selecionados, construída com um componente próprio ao aplicativo de validação, o Cartão de Produto. Esse componente é relativo aos organismos, no Design Atômico, e combina as moléculas Botão de Ícone e Imagem para formar uma entidade mais complexa, a fim de demonstrar a flexibilidade do *design system* importado.

Seguindo a listagem de produtos, encontra-se um setor de aplicação de descontos, construído utilizando uma combinação das moléculas de Campo de Entrada de Texto e Botão de Ícone, outra vez análogo a um organismo. Para assegurar que esse organismo é acessível, os componentes recebem claras identificações e texto alternativo; além disso, a resposta da validação do cupom é gerenciada pelo componente de Semântica em tempo real, gerando uma resposta automática às tecnologias assistivas, seguindo as diretrizes de interfaces compreensíveis e operantes (JACOB; MACKENZIE, 2018).

Na base da página, um resumo dos valores gerais da compra até o momento é apresentado, utilizando os átomos de Tipografia, e semanticamente agrupados, aproveitando-se da característica de união de Semântica, provendo clareza e facilidade de entendimento para

usuários de leitores de tela. Por fim, um Botão de Texto permite aos usuários navegar à tela de seleção de endereço e entrega.

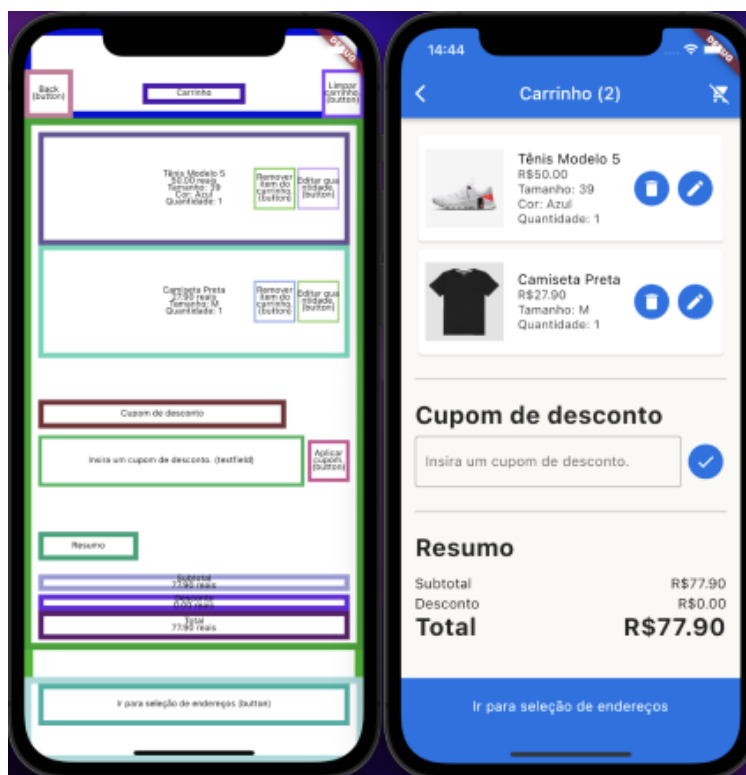


Figura 16 – Carrinho de Compras — visibilidade normal e semântica (reprodução do autor)

4.1.1.3 Endereço e Entrega

Páginas de seleção de endereço e entrega são, normalmente, pontos pivotais na experiência de compra do usuário por conterem uma gama maior de informações a serem fornecidas, funcionando como uma ponte entre o carrinho e o estágio final de pagamento.

A Figura 17 ilustra que a tela desenvolvida para a aplicação protótipo reflete os princípios gerais comuns de telas dessa funcionalidade, sendo essencialmente uma listagem de cartões de endereço do usuário e métodos de entrega disponíveis para o endereço selecionado.

Quando não há endereços cadastrados, o estado inicial da tela é vazio, e essa informação é passada via uma mensagem que utiliza os átomos de Tipografia para garantir acessibilidade. A seguir, encontra-se um Botão de Texto provendo ao usuário a ação de adicionar novos endereços; ao clicar nesse botão, é apresentada uma modal inferior composta por componentes de Campos de Entrada de Texto relativos às informações do endereço a ser cadastrado. O foco em acessibilidade é mais uma vez evidenciado com a resposta aos dados digitados (JACOB; MACKENZIE, 2018), trazendo uma reação programática prioritária na Árvore de Semântica se algum campo faltou ou se o endereço foi cadastrado com sucesso.

Cartões de endereço e de seleção de método de entrega são construídos, ambos centrais a funcionalidade desta página. Esses organismos foram implementados utilizando Botões de Ícones que se adaptam de acordo com a seleção do cartão; com foco em acessibilidade visual, a resposta de quais endereço e método estão selecionados é passada ao usuário via elementos de Semântica.

A barra de navegação inferior é construída com um Botão de Texto que fornece a informação final de preço. A informação é atualizada com o valor acrescido pela entrega, permitindo a completude do fluxo proposto pelo protótipo, oferecendo resposta negativa, na eventual não seleção de uma informação necessária, ou positiva, ao concluir o processo.

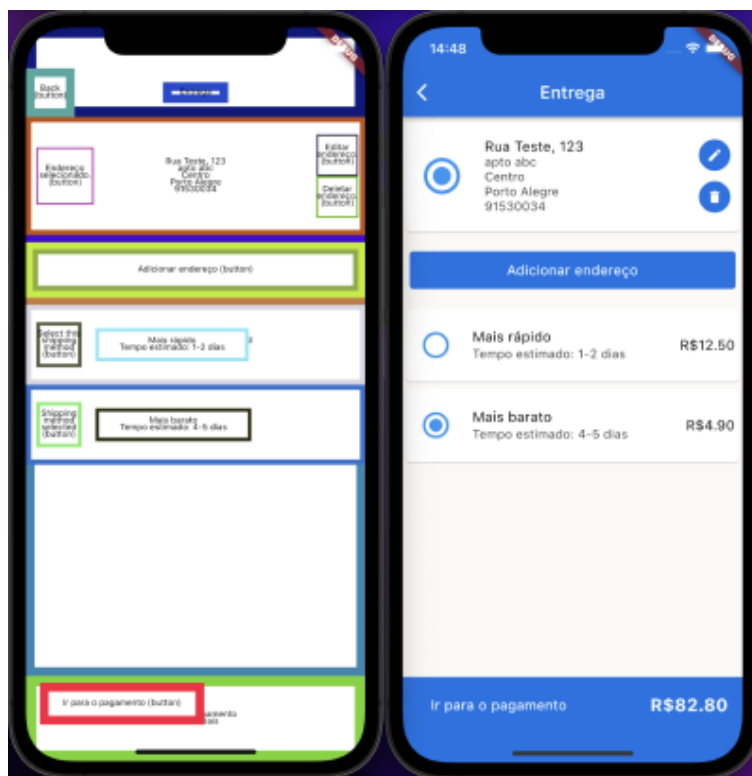


Figura 17 – Endereço e Entrega — visibilidade normal e semântica (reprodução do autor)

4.1.2 Validação Manual

A validação manual é um processo que exige uma análise cuidadosa do aplicativo em desenvolvimento (ZHANG; ADIPAT, 2005). Utilizando os modos de depuração, com ou sem visibilidade da Árvore de Semântica do Flutter, foi possível visualizar diretamente o comportamento do aplicativo, para usuários com ou sem tecnologias de leitores de tela, respectivamente.

Utilizando o depurador semântico, destacou-se cada bloco de construção da Árvore

Semântica, permitindo ver quais informações cada nó continha, como ações, estados, rótulos e estrutura.

Verificou-se a presença de textos alternativos em cada elemento interativo, para garantir que eles não apenas tivessem rótulos corretos, mas dicas significativas e informações de valor. Testou-se cada interação do aplicativo em formato regressivo, percorrendo os possíveis fluxos de diferentes maneiras, a fim de assegurar consistência e coesão na usabilidade.

Essa primeira etapa permitiu detectar e corrigir quaisquer inconsistências ou deficiências aparentes ainda durante o processo de desenvolvimento, facilitando o fornecimento de uma experiência de usuário satisfatória e, acima de tudo, acessível.

4.1.3 Validação de Ferramenta

Na segunda fase, utilizou-se o *Accessibility Scanner*, uma ferramenta automatizada do Google para aplicativos Android. Essa ferramenta realiza uma análise completa da interface do aplicativo, tela a tela, verificando possíveis problemas de acessibilidade (FLUTTER, 2023); inspecionando elementos como contraste de cores, tamanhos de alvos de toque, rotulagem de conteúdo e mais.

Com a ferramenta, foi possível analisar sistematicamente a aplicação em relação aos princípios-chave de acessibilidade. Alguns pontos iniciais foram levantados pela ferramenta, sobretudo acerca do tamanho de botões, que foram prontamente ajustados. Essas verificações automáticas ajudaram na identificação e solução de problemas mais sutis, que poderiam ter passado despercebidos durante a validação manual, garantindo assim um nível mais elevado de conformidade com as diretrizes de acessibilidade.

Vale ressaltar que há, na versão 3.7.9 do Flutter, utilizada no desenvolvimento do protótipo, uma incompatibilidade ainda não solucionada com o *Android Accessibility Scanner*, que acusa uma falha geral inexistente em todas as telas. Isso pode ser um impeditivo àqueles que desejam passar por um processo de validação oficial de acessibilidade, mas não atrapalhou o desenvolvimento da aplicação objeto deste estudo.

4.1.4 Validação de Usuário

O campo da interação humano-computador enfatiza fortemente a importância do teste de usuário no processo de desenvolvimento de qualquer produto digital. Isso é especialmente importante na criação de aplicativos focados em acessibilidade, pois os testes empíricos com usuários reais podem expor problemas que o ferramental ou os testes feitos pelos desenvolvedores podem não identificar (PETRIE; BEVAN, 2009). A eficácia de um produto, afinal,

só pode ser avaliada de fato por aqueles que são seus beneficiários pretendidos.

Nesse sentido, a acessibilidade não se refere apenas à adesão a diretrizes ou normas técnicas, mas às pessoas. Trata-se de criar um mundo digital em que todos, independentemente de suas capacidades físicas, possam participar e se envolver plenamente (STORY; MUELLER; MACE, 1998). Trata-se de expandir o direito de cada indivíduo de acesso à informação, de poder expressar seus pensamentos e se conectar com o mundo ao seu redor, cada vez mais digital.

A busca por um processo de validação abrangente leva ao teste definitivo: a interação com o usuário. Esse projeto conta com o apoio de Volmir Raimondi, deficiente visual desde a infância, dedicou a vida a defender a causa da acessibilidade, nas mais diversas esferas: como foi apresentado anteriormente, é ex-presidente da União Latino-Americana de Cegos e ex-membro do Conselho Nacional de Saúde, fundador da Associação dos Deficientes Visuais de Bento Gonçalves e palestrante reconhecido internacionalmente por seus esforços e participação ativa na comunidade assistiva.

Ao testar o aplicativo com o Sr. Raimondi, obteve-se *feedbacks* diretos sobre o desempenho do sistema para os usuários pretendidos. A tela do dispositivo foi gravada durante toda a interação, servindo como base para entender como um usuário deficiente visual navega no aplicativo proposto e interage com cada um dos elementos do *design system* construído. Sua experiência e resposta confiável ajudaram a avaliar o aplicativo de uma perspectiva excepcionalmente informada. Nesse processo, além de avaliar se o sistema era tecnicamente viável, foi possível aprofundar se a aplicação realmente atinge seu objetivo principal de melhorar a usabilidade para pessoas com acuidades visuais variadas.

Durante esse processo, um diálogo interativo foi mantido com o Sr. Raimondi, sobre suas experiências, como ele percebia o aplicativo e o que pensava da iniciativa como um todo. Essa discussão inspirou e foi seguida por uma série de cinco perguntas criadas para oficializar uma resposta mais estruturada acerca do projeto, dos recursos e da acessibilidade em geral. As respostas a essas perguntas, combinadas ao uso do aplicativo em mãos, tiveram um papel crucial no processo de validação, e discorrem a seguir:

1. Como usuário, quais foram suas impressões gerais ao usar o aplicativo? Em quesito de acessibilidade, como a estrutura e funcionalidade geral do protótipo se comportaram?

Um sentimento de liberdade por perceber que as ferramentas tecnológicas, quando são desenvolvidas com cuidado e a preocupação de incluir, embora gerem mais trabalho e desafiem o desenvolvedor a pensar fora da caixa, o resultado torna-se palpável com produtos de valor agregado. Nada é mais libertador do que você utilizar as facilidades do mundo digital como qualquer pessoa, eleger um produto, pagar e recebê-lo na sua casa.

Quando conseguimos esse feito, certamente é devido a acessibilidade em aplicações que foram projetadas para excluir barreiras e nunca pessoas.

2. Em relação aos componentes, como botões e campos de texto, funcionaram da maneira esperada?

O protótipo é de fácil navegação, intuitivo, simples e perfeito na interação com o leitor de tela, todos campos editáveis e botões totalmente funcionais. Atendeu minhas expectativas de acessibilidade e lógica aplicada na construção. Ao testar o protótipo encontrei um desenvolvedor sensível e preocupado em criar uma solução que proporcionasse a liberdade da qual falei.

3. Existe algum ponto particular do aplicativo onde as funcionalidades de acessibilidade faltaram ou poderiam ser melhoradas?

Apenas um: na opção entrega, sugiro incluir o botão na mesma linha com a descrição do frete.

4. Em comparação a outros aplicativos, de uso cotidiano ou esporádico, como as funcionalidades de acessibilidade se comportaram? E o uso em geral?

Em comparação com outros aplicativos, fica claro que quando o desenvolvedor procura conhecer as necessidades dos usuários do seu produto, este consegue desenvolver um aplicativo que realmente atende as expectativas. Na minha experiência diária é comum encontrar apps que me permitem determinadas funcionalidades, como visualizar um produto que eu gostaria de comprar e que no final das contas me deixa totalmente frustrado por não me permitir utilizar o campo pagamento ou escolha do tipo de entrega. No protótipo tudo funcionou com facilidade e de forma intuitiva; algumas vezes, os aplicativos são razoavelmente acessíveis, mas, na primeira atualização deixam de ser pois, alguém acaba esquecendo da tal acessibilidade, assim, acredito que para nós pessoas com deficiência, um app acessível assume a melhor forma de fidelização do cliente. Só compro em empresas que se preocuparam comigo e com acessibilidade.

5. Dada sua vasta experiência, tanto como usuário quanto como autoridade e especialista no assunto, qual seu sentimento acerca dessa iniciativa? Como você percebe o futuro da acessibilidade em aplicações digitais?

Considero uma iniciativa extremamente valiosa, considerando que a preocupação com acessibilidade, embora seja lei, ainda não é corrente entre a maioria das empresas de construção de soluções digitais. Faz-se necessário diminuir a brecha digital através de iniciativas como essa, que compartilha conhecimento e cria caminhos que levam a uma cultura de produtos com acessibilidade universal. Entendo que poderia ser altamente

benéfico que haja o desenvolvimento de uma carreira de especialização em acessibilidade e que empresas de desenvolvimento possuam grupos dedicados a testagem e validação de seus produtos. Creio que num futuro próximo, o mercado exigirá acessibilidade por padrão, assim, quem começar agora, estará apostando na autopreservação.

A vasta experiência pessoal e profissional do Sr. Volmir Raimondi no campo da deficiência visual garantiu uma validação robusta e abrangente do proposto. Essa resposta ajuda a afirmar que o *design system* desenvolvido realmente melhora a usabilidade para usuários com deficiência visual, atingindo seu objetivo geral.

Essa experiência foi um poderoso lembrete de que o esforço desse projeto não se limita apenas a criar um produto que funciona corretamente, mas que faz a diferença na vida das pessoas que dependem de recursos de acessibilidade para interagir com o mundo ao seu redor. No fim das contas, esse é o objetivo final da acessibilidade: construir um mundo, físico ou digital, em que todas as pessoas possam prosperar em suas interações.

5 Conclusão

A inspiração para construir um sistema que promova a cultura de inclusão nasceu do reconhecimento de que a revolução do mundo digital é incompleta sem a percepção da acessibilidade. Com base em vários campos, o projeto foi impulsionado por um profundo entendimento da necessidade de acesso universal, inspirado pela verdade inegável de que a perspectiva digital deve ser uma arena aberta a todos, cenário, infelizmente, diferente do encontrado hoje.

O desenvolvimento do *design system* proposto teve seus desafios: a mentalidade de trabalho precisou passar por uma mudança substancial, devido à notável falta de familiaridade e conforto ao navegar no domínio da acessibilidade; foi necessário aprender a analisar a verdadeira inclusão do sistema, e apreciar sua importância além dos - necessários - domínios das diretrizes e normas recomendadas, experimentando suas implicações no mundo real por meio do uso de tecnologias assistivas.

A escolha e envolvimento do Flutter como framework de trabalho foi uma prova do poder da tecnologia quando aproveitada com propósito e compreensão. Com arquitetura baseada na metodologia do Design Atômico, foi possível desenvolver um *design system* que serve como uma base robusta para o desenvolvimento de futuras aplicações, empregando elementos de acessibilidade em seus componentes de maneira direta e prática.

A manifestação do sistema desenvolvido tomou forma em uma aplicação de validação, no formato de um fluxo de compra padrão de um aplicativo de comércio eletrônico, fazendo valer de todos os componentes implementados. Embora o *design system* tenha estabelecido uma base sólida, a criação de fluxos e elementos mais complexos exigiu uma abordagem mais ativa, manipulando e criando novos componentes acessíveis, ressaltando a necessidade da atenção à acessibilidade em diferentes escopos.

No entanto, a validação derradeira veio da utilização do sistema no mundo real. A oportunidade de interagir com um usuário que dedicou sua vida à defesa da acessibilidade, recebendo uma resposta extremamente positiva e obtendo discernimento a partir de sua perspectiva, foi mais do que uma prova do sucesso do projeto, mas um lembrete do propósito por trás dele. A experiência satisfatória do usuário valida a hipótese de que a integração de considerações de acessibilidade desde os estágios iniciais, representada aqui por um *design system*, pode criar um ambiente digital mais inclusivo.

A tecnologia é parte significativa da vida no mundo atual, e garantir que todos possam

se beneficiar dela é uma missão que vale a pena ser defendida. Como observação final, ressalta-se que a acessibilidade é uma jornada contínua, não uma conquista única; é necessário buscar conhecimento, em contato com aqueles que fazem uso das tecnologias, mantendo os usuários e suas necessidades no centro dos esforços.

5.1 Trabalhos Futuros

A jornada deste trabalho de conclusão de curso trouxe à tona um *design system* destinado a integrar acessibilidade em seu centro, com o objetivo de aprimorar a experiência digital de usuários com acuidades visuais variadas. Entretanto, como em qualquer empreitada de pesquisa, a conclusão não marca o fim; ao invés disso, estabelece a base para possíveis trabalhos e aprimoramentos futuros.

Uma área que merece atenção futura é a Calculadora de Contraste de Cor. Em sua forma atual, a Calculadora só é capaz de computar e ajustar cores dentro do mesmo componente, com o conhecimento do contexto. Isso deixa uma lacuna na avaliação de problemas de contraste entre componentes diferentes ou entre um componente e o plano de fundo. Além disso, a Calculadora pode ser expandida visando levar em conta anomalias de percepção de cores, como o daltonismo; embora a ferramenta já ajude a obter um melhor contraste - e isso, certamente, ajuda - seria interessante entender os diversos espectros de daltonismo e suas abordagens.

Em termos de validação, embora a realizada tenha fornecido percepções valiosas, o tamanho da amostragem foi mínimo. Trabalhos futuros poderiam validar o *design system* e seus aplicativos resultantes com um grupo maior e mais diversificado de usuários com deficiência visual, identificando aprimoramentos ou problemas que não ficaram evidentes nos testes iniciais. Além disso, novas ferramentas de validação poderiam ser incorporadas.

Por fim, é possível aprofundar-se na exploração de recursos tecnológicos, tanto os já presentes no sistema proposto, como novos. Como a tecnologia continua avançando, o mesmo acontece com as oportunidades de tornar os espaços digitais mais acessíveis e inclusivos.

Referências

- ABASCAL, J.; NICOLLE, C. Moving towards inclusive design guidelines for socially and ethically aware hci. *Interacting with Computers*, v. 17, 2005. Citado na página 41.
- APPLE. *Human Interface Guidelines*. 2023. Disponível em: <<https://developer.apple.com/design/human-interface-guidelines>>. Citado 4 vezes nas páginas 21, 23, 30, and 35.
- BRASIL. ABNT. *ABNT NBR 17060: Acessibilidade em aplicativos de dispositivos móveis*. 2022. Citado na página 15.
- BROWN, T. *Design Thinking: Edição comemorativa 10 anos*. [S.l.]: Editora Alta Books, 2020. ISBN 9788550814377. Citado na página 19.
- COSCARELLI, C. V. *Hipertextos na teoria e na prática*. [S.l.]: Autêntica, 2012. 176 p. ISBN 9788565381512. Citado na página 39.
- DART. *Dart Documentation*. 2023. Disponível em: <<https://dart.dev/guides>>. Citado na página 26.
- ESPINHA, T.; ZAIDMAN, A.; GROSS, H. G. Web api growing pains: Stories from client developers and their code. *IEEE*, p. 84–93, 2014. Citado na página 27.
- ESTADOS UNIDOS DA AMÉRICA. U.S. Department of Justice, Civil Rights Division. *Americans with Disabilities Act*. 1990. Citado na página 15.
- FLUTTER. *Flutter Documentation*. 2023. Disponível em: <<https://docs.flutter.dev/>>. Citado 5 vezes nas páginas 9, 26, 27, 34, and 47.
- FROST, B. *Atomic Design*. [S.l.]: Brad Frost, 2016. ISBN 9780998296609. Citado 6 vezes nas páginas 9, 13, 20, 28, 29, and 42.
- GOOGLE. *Material Design*. 2023. Disponível em: <<https://m3.material.io/>>. Citado 3 vezes nas páginas 21, 22, and 30.
- IBM. *Carbon Design System*. 2023. Disponível em: <<https://carbondesignsystem.com/>>. Citado 2 vezes nas páginas 22 and 23.
- INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. *Censo Demográfico 2010*. 2010. 114-141 p. Citado na página 13.
- INTERNATIONAL COMMISSION ON ILLUMINATION. *ISO/CIE 23539:2023 | CIE TC 2-93: Photometry — the cie system of physical photometry*. 2023. Citado na página 33.
- JACOB, S. V.; MACKENZIE, I. S. Comparison of feedback modes for the visually impaired: Vibration vs. audio. In: *Universal Access in Human-Computer Interaction. Methods, Technologies, and Users: 12th International Conference, UAHCI 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15-20, 2018, Proceedings, Part I*. [S.l.]:

Springer-Verlag, 2018. p. 420–432. ISBN 9783319920481. Citado 2 vezes nas páginas 44 and 45.

KHOLMATOVA, A. *Design Systems: A practical guide to creating design languages for digital products*. [S.l.: s.n.], 2017. ISBN 9783945749586. Citado 2 vezes nas páginas 21 and 30.

MAROPOULOS, P.; CEGLAREK, D. Design verification and validation in product lifecycle. *CIRP Annals*, v. 59, n. 2, p. 740–759, 2010. ISSN 0007-8506. Citado na página 41.

MICROSOFT. *Fluent 2*. 2023. Disponível em: <<https://fluent2.microsoft.design/>>. Citado 2 vezes nas páginas 21 and 23.

MOVIMENTO WEB PARA TODOS, BIGDATACORP, W3C BRASIL, NIC.BR. *Estudo de acessibilidade em aplicativos Android (Brasil, 2021)*. [S.l.], 2021. Disponível em: <<https://mwpt.com.br/metodologia-utilizada-no-estudo-de-acessibilidade-em-sites-ativos-e-aplicativos-android-realizado-em-2021/>>. Citado na página 14.

NEIL, T. *Mobile Design Pattern Gallery: Ui patterns for smartphone apps*. [S.l.]: O’Reilly Media, 2014. 404 p. ISBN 9781449363635. Citado 2 vezes nas páginas 35 and 37.

NGUYEN, T. T. et al. Measuring user perception for detecting unexpected access to sensitive resource in mobile apps. In: *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. [S.l.]: Association for Computing Machinery, 2021. (ASIA CCS ’21), p. 578–592. ISBN 9781450382878. Citado 2 vezes nas páginas 40 and 44.

NIELSEN CONSUMER LLC. *Webshoppers 47*: Edição gratuita. [S.l.], 2023. Citado na página 42.

NZONGO, F. *Design em escala*. [S.l.]: Viver Cultural, 2022. ISBN 9786599081972. Citado 3 vezes nas páginas 9, 20, and 29.

PETRIE, H.; BEVAN, N. The evaluation of accessibility, usability, and user experience. *C Stepanidis*, 2009. Citado na página 47.

PRASAD, G. S.; CHOPELLA, V. Notion of semantics in computer science: A systematic literature review. 2017. Citado na página 34.

SAARINEN, K. Building a visual language: Behind the scenes of our airbnb design system. 2016. Disponível em: <<https://medium.com/airbnb-design/building-a-visual-language-behind-the-scenes-of-our-airbnb-design-system-224748775e4e>>. Citado na página 21.

SALESFORCE. *Lightning Design System*. 2023. Disponível em: <<https://www.lightningdesignsystem.com/>>. Citado 2 vezes nas páginas 22 and 23.

SHIRAZI, A. S. et al. Insights into layout patterns of mobile user interfaces by an automatic analysis of android apps. In: . [S.l.]: Association for Computing Machinery, 2013. ISBN 9781450321389. Citado na página 35.

SHOPIFY. *Polaris*. 2023. Disponível em: <<https://polaris.shopify.com/>>. Citado 2 vezes nas páginas 21 and 23.

STOKES, M. et al. A standard default color space for the internet - srgb. 1996. Citado na página 33.

STORY, M. F.; MUELLER, J. L.; MACE, R. L. *The Universal Design File*. [S.l.: s.n.], 1998. 170 p. Citado 3 vezes nas páginas 13, 25, and 48.

SUAREZ, M. et al. *Design Systems Handbook*. [S.l.]: DesignBetter.co, 2019. ISBN 9781987139617. Citado 2 vezes nas páginas 19 and 20.

UNIÃO EUROPEIA. Comissão Europeia. *European accessibility act: Union of equality: Strategy for the rights of persons with disabilities 2021-2030*. 2019. Citado na página 15.

WEBAIM. *The WebAIM Million*. [S.l.], 2023. Disponível em: <<https://webaim.org/projects/million/>>. Citado 2 vezes nas páginas 15 and 35.

WORLD HEALTH ORGANIZATION. *World Report on Vision*. 2019. Citado na página 13.

WORLD WIDE WEB CONSORTIUM. *Web Content Accessibility Guidelines (WCAG) 2.1*. 2018. Disponível em: <<https://www.w3.org/TR/2018/REC-WCAG21-20180605/>>. Citado 2 vezes nas páginas 32 and 33.

YOU, D.; HU, M. A comparative study of cross-platform mobile application development. *Citrenz* 2021, 2021. Citado na página 28.

ZHANG, D.; ADIPAT, B. Challenges, methodologies, and issues in the usability testing of mobile applications. *Int. J. Hum. Comput. Interaction*, v. 18, p. 293–308, 2005. Citado 2 vezes nas páginas 42 and 46.



Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br