

Validador e Guidelines de Acessibilidade para o VoiceOver

Felipe K. Petersen¹, Paula T. Leite¹, Pedro H. C. Braga¹

¹Faculdade de Computação e Informática – Universidade Presbiteriana Mackenzie
Rua Consolação 930, CEP 01302-000, São Paulo – SP, Brasil

felipekpetersen@gmail.com, paula.t.leite@icloud.com, phcacique@gmail.com

Abstract. *With the popularization of mobile applications, many metrics for their development were created. However, when discussing accessibility, projects that don't pertain to mobile development are referenced. More recent studies show how accessibility should behave on a mobile systems, however, there are still few ways to validate and differentiate them from other systems. The following study addresses the validation of the VoiceOver system for accessibility of Apple devices. The end result are metrics for developers to follow and a validator that can be used to check these metrics.*

Resumo. *Com a popularização dos aplicativos mobile, muitas métricas para o desenvolvimento deles foram criadas. Porém, quando fala-se de acessibilidade, são referenciados trabalhos que não tratam do desenvolvimento mobile. Estudos mais recentes mostram como deve-se comportar a acessibilidade em um sistema mobile, porém, ainda existem poucas formas de validá-los e diferenciá-los de outros sistemas. O estudo a seguir aborda especificamente a validação do sistema VoiceOver de acessibilidade de dispositivos Apple. O resultado final são métricas para desenvolvedores seguirem e um validador que possa ser usado para a checagem dessas métricas.*

1. Introdução

Nos últimos tempos, o desenvolvimento de aplicações vem passando por uma crescente. Os dados apontam para um crescimento de 107,2% do uso pessoal de celulares por brasileiros nos últimos 10 anos [IBGE 2011]. Isso mostra uma necessidade do aumento do desenvolvimento de aplicativos móveis, sendo uma área que cresceu juntamente com o uso de celulares.

Para o desenvolvimento de *software*, existem várias etapas. Este projeto focou na etapa de avaliação. Essa etapa é de altíssima importância, sendo que ela busca evitar problemas futuros e, principalmente, contribui para garantir que o público alvo se sente confortável e satisfeito com a usabilidade da aplicação.

Dentro da usabilidade, existe a acessibilidade. Este artigo teve um enfoque nas formas com que a acessibilidade é avaliada dentro de dispositivos móveis, em relação ao *VoiceOver* da *Apple*, uma tecnologia de leitura de tela desenvolvida pela empresa. Atualmente existem especificações, criadas pela empresa, em relação a acessibilidade dentro de suas aplicações, mas elas não estão ao nível de, por exemplo, a WCAG [Caldwell B. 2008].

A WCAG [Caldwell B. 2008] é uma das principais coletâneas de métricas de acessibilidade a serem seguidas dentro da *Web*. Dentre elas, podemos também citar o eMAG

[eMAG 2021] e o WAI [WAI 2021]. Nota-se que essas diretrizes são muito bem trabalhadas para a *Web*, porém, elas não são o suficiente [Clegg-Vinell et al. 2014], necessitando ser reavaliadas para o contexto atual.

Notou-se que existe uma escassez de diretrizes de acessibilidade dentro do mundo *mobile*. Muitos desenvolvedores se baseiam nas diretrizes estabelecidas para a *Web* para avaliarem suas aplicações, porém, existe a necessidade do desenvolvimento de regras específicas. Isso tornou-se uma necessidade em consequência das grandes diferenças entre as plataformas e a necessidade de instruções para a realização de testes de acessibilidade.

Esses testes são de altíssima importância para as empresas e os seus desenvolvedores, porém, eles normalmente envolvem um alto custo e são trabalhosos. Existem diversas formas de realizar avaliações, sendo as principais [Sharp et al. 2007]: testes informais de usabilidade, testes formais de usabilidade, testes em campo e avaliações preditivas. Cada avaliação possui as suas próprias forças, porém, os testes de usabilidade são os que apresentam a maior quantidade de dados e são altamente custosos.

O custo e a rigidez dos testes de usabilidade são as duas falhas dessa avaliação. Sendo assim, diversas *frameworks* são desenvolvidas para contribuir nessas avaliações. Existem, por exemplo, *frameworks* automatizadas que fornecem uma forma mais rápida e eficiente de fazer esses testes com sistemas de formulários [Au et al. 2008]. Nota-se, então, que existem diversas sistemas que podem ser desenvolvidos para contribuir nesses testes.

Sendo assim, o desenvolvimento de diretrizes centradas no *VoiceOver* em dispositivos *mobile*, bem como um validador que ajudasse o desenvolvedor a seguir tais diretrizes, tornam-se importantes de serem desenvolvidos. Com isso, garante-se que a grande maioria das diretrizes estão sendo seguidas, levando a um menor custo e maior eficiência.

2. Objetivos

Neste projeto, teve-se o propósito de estudar a literatura que envolve acessibilidade em aplicações e testes de usabilidade da mesma.

Além disso, desejou-se levantar métricas para a validação do *VoiceOver* em aplicações *emphApple*. Foi, também, desenvolvido um validador que ajuda desenvolvedores a validar essas métricas em suas aplicações. Posteriormente, foi-se desenvolvido um aplicativo para testar o funcionamento das métricas e do validador.

Também foi realizado, ao final do projeto, uma análise sobre as métricas e o validador desenvolvidos, de forma que o projeto se tornou uma referência para o desenvolvimento de novos validadores e métricas bem como para o uso deles.

3. Referencial Teórico

3.1. Materiais

Para a aquisição dos artigos levantados para o projeto, as bases de dados ACM, *Semantic Scholar* e *Springer* foram utilizadas como fontes de pesquisa por serem bases de grande número de artigos publicados em periódicos relevantes e por apresentarem mecanismos de filtragem.

Foram testadas algumas palavras para consulta relacionadas ao tema desejado, separados em três grandes grupos: *emphUsability Testing*, *Accessibility*, e *Usability Evaluation*.

Entretanto, os resultados se apresentaram bastante genéricos. Assim, mantendo o domínio do problema, acrescentou-se ainda um novo termo para filtragem: *emphAutomated*. As combinações passaram a ser:

- *Automated Usability Testing*
- *User Requirements*
- *VoiceOver*
- *Accessibility Guidelines*

Para a primeira consulta, foram encontrados um grande número de artigos e livros. Foi-se também reduzido para publicações a partir de 2002 e foram analisadas as principais publicações fornecidas pela ACM. Foi-se assim que quatro artigos e um livro relevantes para a pesquisa foram encontrados.

Para a segunda e terceira consulta, foram encontrados, novamente, um grande número de artigos de uma forma semelhante ao ocorrido no levantamento anterior. Dessa forma, encontrou-se mais quatro artigos relevantes. Dentre esses artigos, houve-se uma consulta pelas referências dentre eles, assim encontrando-se o nono artigo.

Por fim, buscou-se analisar as diretrizes de acessibilidade existentes na atualidade. Para isso, foram analisadas as levantadas pela própria *emphApple*, bem como, o WCAG [Caldwell B. 2008], eMAG [eMAG 2021] e WAI [WAI 2021].

3.2. Método

Para compreendermos os conhecimentos tirados desses artigos e livros, eles foram categorizados:

- Testes de Usabilidade
- *Frameworks* de Validação
- Diretrizes de Acessibilidade

A seguir, encontra-se o referencial teórico desenvolvido neste projeto.

3.2.1. Testes de Usabilidade

Os testes de usabilidade são de altíssima importância. Sendo assim, torna-se uma prioridade entender as diferentes formas de conduzi-los. Existem três métodos básicos [Scholtz 2001]: avaliações centradas nos usuários, avaliações baseadas em profissionais e avaliações baseadas em modelos.

As avaliações com usuários são divididas entre formativas e de usabilidade. Sendo a primeira realizada nos momentos iniciais, rapidamente e não formal. Já a segunda é feita oficialmente, seguindo as regras de testes de usabilidade e é feita no final do desenvolvimento. Já as avaliações profissionais são divididas em revisões de diretrizes e padrões, orientações cognitivas e avaliações heurísticas.

Para a escolha de quais avaliações devem ser realizadas e em que momentos, cada empresa deve realizar as suas próprias análises. Durante o desenvolvimento de uma

aplicação, são realizados diversos testes. Porém, isso não é o suficiente para garantir que o usuário estará satisfeito [Sharp et al. 2007]. Além disso, o investimento em testes de usabilidade pode resolver problemas antes da aplicação ser distribuída.

Essas avaliações podem ser divididas em dois momentos, durante o desenvolvimento do *design* da aplicação ou após o seu desenvolvimento.

3.2.2. Frameworks de Validação

O uso de dispositivos móveis é algo que se tornou muito comum na atualidade. Sendo assim, o desenvolvimento de aplicações para eles passou a ser constante. Um dos maiores problemas disso decorre do fato de que essas aplicações passam por testes de usabilidade infrequentes [Au et al. 2008]. Sendo assim, o desenvolvimento de *frameworks* que contribuam nesse processo torna-se uma necessidade.

Para que essas *frameworks* sejam implementadas corretamente, necessita-se que elas realizem uma avaliação da Interface Gráfica do Usuário, ou seja, das fontes, botões, disposição e menus de acordo com Au et al (2008). Afirma-se também que é necessário ocorrer uma análise de quanto esforço o usuário deve fazer para localizar e utilizar informações e funcionalidades. A ferramenta também deve apresentar modelos e diretrizes que descrevam soluções para os desenvolvedores e isso deve ser apresentado de uma forma clara e objetiva.

3.2.3. Diretrizes de Acessibilidade

Existem diversas diretrizes de acessibilidade as quais, em sua grande maioria, são voltadas para ferramentas *Web*. Dentre as principais, podemos citar: WCAG (Caldwell, Reid e Vanderheiden, 2008), eMAG (2021) e WAI (2021).

As Diretrizes de Acessibilidade de Conteúdo da Web [Caldwell B. 2008] abrangem uma ampla gama de recomendações para tornar o conteúdo da *Web* mais acessível. Com o uso dessas diretrizes, o conteúdo do site se tornará mais acessível. As seguintes deficiências foram consideradas para o seu desenvolvimento: cegueira e baixa visão, surdez e perda auditiva, dificuldades de aprendizagem, limitações cognitivas, movimento limitado, deficiência de fala, fotossensibilidade e combinações destes. Existe uma versão do WCAG que está sendo desenvolvida com o enfoque em *mobile*, porém, ela ainda está em uma versão de rascunho.

As diretrizes anteriores foram desenvolvidas no exterior, porém, o Governo Brasileiro também possui suas diretrizes: o Modelo de Acessibilidade em Governo Eletrônico [eMAG 2021]. Elas pretendem ser um norte para o desenvolvimento de conteúdos digitais. As suas recomendações buscam padronizar o desenvolvimento de sites e facilitar sua implementação de forma acessível.

Por fim, a Iniciativa de Acessibilidade da *Web* [WAI 2021] é um esforço para melhorar a acessibilidade da *Web* para pessoas com deficiência. Pessoas com deficiência podem encontrar dificuldades ao usar computadores em geral, mas também na *web*. Essa iniciativa é de onde surge o WCAG (2008) e diversas outras diretrizes para o universo

mobile.

Existe uma escassez de diretrizes de acessibilidade voltadas para o mundo *mobile*. Por conta disso, existem autores que começaram a trazerem suas próprias *guidelines*. Com elas, é possível realizar uma avaliação mais concreta dos sistemas.

Existem pesquisas que mostram falhas dentro das diretrizes atuais de acessibilidade. Nelas, foram encontrados 1214 problemas de acessibilidade não encontrados nas métricas, [Calvo et al. 2016].

Sendo assim existe uma necessidade de re-avaliação das métricas estipuladas para a acessibilidade na *Web*, bem como a a criação de novas métricas e novos padrões [Clegg-Vinell et al. 2014].

Além disso, o desenvolvimento de métricas específicas para a acessibilidade em aplicativos tornou-se algo essencial. Isso por conta das características específicas de funcionamento dessa plataforma. Além disso, é importante que as métricas desenvolvidas não sejam generalizadas, elas devem ter um enfoque nas diferentes acessibilidades [Siebra et al. 2015].

4. Metodologia

No que diz respeito à Metodologia empregada, o trabalho teve início com uma revisão da literatura específica sobre o tema da pesquisa, envolvendo também um estudo teórico do problema. O foco de tal estudo envolveu os conceitos fundamentais e o referencial teórico relativo às métricas existentes para acessibilidade, bem como as melhores formas de validação das mesmas e suas vantagens e desvantagens.

A base teórica foi obtida através de pesquisas em bibliografias, documentos e livros. A leitura e análise dos fundamentos teóricos encontrados tiveram como objetivo a análise de formas de validação da acessibilidade da framework da *Apple* de *emphVoiceOver*, para que fosse possível o desenvolvimento de métricas para a validação do mesmo e um validador que percorra essas métricas levantadas.

A estruturação desses dados foi o próximo passo, na qual foi possível unificar as informações obtidas da revisão bibliográfica. Com isso, foi possível o desenvolvimento das métricas de acessibilidade do *emphVoiceOver*, assim como o seu validador, sendo então testados em situações reais. Com isso, foi possível discernir as vantagens de utilizar uma ferramenta avaliativa alternativa em contrapartida dos métodos tradicionais.

O próximo passo da pesquisa foi a análise dos resultados obtidos, para que fosse possível encontrar os ganhos e as perdas das métricas e do validador.

Por fim, os relatórios das etapas realizadas e as conclusões finais dessa pesquisa foram documentados no texto final, cuja redação consta como a última etapa do projeto.

5. Resultados

Tendo em vista as limitações das diretrizes de acessibilidade da atuais, principalmente em relação a sua aplicação em dispositivos *mobile* e sua generalização das diversas deficiências, foram propostas novas diretrizes, voltadas para o *VoiceOver*, em dispositivos *Apple*. Além disso, também houve a construção de um validador, baseado nessas diretrizes, o qual o desenvolvedor pode adicionar no seu processo de desenvolvimento.

Com a construção de ambos, pode-se realizar um teste de sua efetividade. Para isso, construiu-se um aplicativo, seguindo as *guidelines*, com o apoio do validador. Os resultados obtidos desse teste levaram a mudanças em ambos os produtos desenvolvidos.

5.1. Diretrizes de Acessibilidade do *VoiceOver*

Para o desenvolvimento das diretrizes de Acessibilidade do *VoiceOver*, foi necessário voltar para a análise das *guidelines* pesquisadas durante este projeto. Dentre elas: WCAG (Caldwell, Reid e Vanderheiden, 2008), eMAG (2021), WAI (2021) e *Apple*.

Essa análise foi dividida em três etapas:

- Levantamento das funções do *VoiceOver* e suas aplicabilidades;
- Levantamento das diretrizes de cada documento que podem ser relacionadas com o *VoiceOver*;
- Desenvolvimento das diretrizes que garantam os dois itens anteriores.

5.1.1. Funções do *VoiceOver*

Para que fosse possível construir diretrizes voltadas para a tecnologia do *VoiceOver*, foi necessário garantir uma extensa compreensão dela.

O *VoiceOver* é baseado em certas configurações que podem ser adicionadas a elementos da interface de uma aplicação. Para que isso seja possível, existem funções previamente implementadas pela *Apple*, as quais tornam a configuração da acessibilidade no aplicativo algo lúdico e de grande facilidade.

Como é possível enxergar na Figura 1, o *VoiceOver* se encontra no meio do caminho de comunicação entre o aplicativo e o usuário.

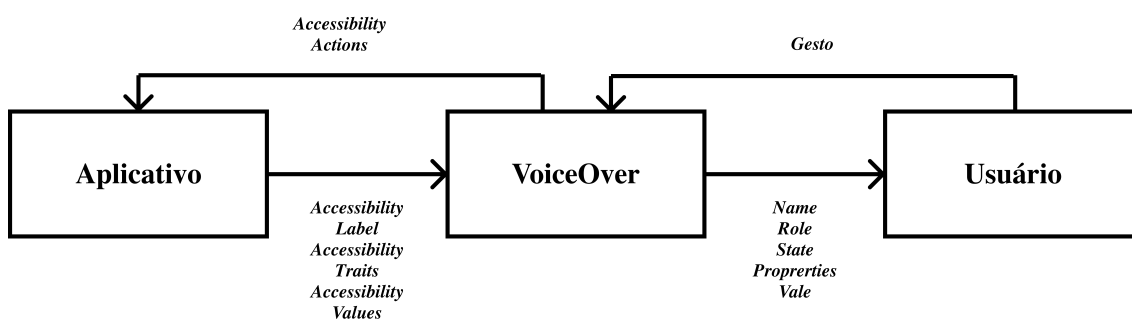


Figura 1. Descrição do funcionamento do *VoiceOver*.

Fonte: autores

Quando o usuário foca em um elemento da tela, o *VoiceOver* anuncia o nome, função, estado, propriedade e valor dele. Para isso, ele recebe essas informações de três propriedades disponíveis:

- *Accessibility Label*: a descrição da ação do elemento, idealmente utilizando apenas uma palavra.

- *Accessibility Traits*: cada elemento possui seu próprio tipo, é possível configurá-lo aqui.
- *Accessibility Value*: elementos com valores variáveis, como o volume, devem ter essa propriedade configurada para que o *VoiceOver* informe as suas variações.

Essas são as principais propriedades disponíveis nos elementos, porém existem outras que tornam a experiência para um usuário mais interessante ainda:

- *Accessibility Hints*: propriedade que pode ser adicionada a cada elemento se for necessário uma maior explicação dele, porém, os usuários podem desconfigurar.
- *Accessibility Element*: todo elemento deve ter essa propriedade como verdadeira, principalmente elementos que não são nativos; pode ser utilizado para adaptar a interface da aplicação para que o fluxo do leitor de tela seja mais lógico.
- *Accessibility Frame*: elementos não nativos podem necessitar dessa configuração para desenhar o cursor do *VoiceOver* ou para o cálculo do local de foco do elemento.
- *Accessibility Actions*: fornece uma maneira para que os elementos de acessibilidade suportem ações específicas.

Todas essas propriedades contribuem para a organização e a configuração da leitura da tela por parte do *VoiceOver*. Para que tudo funcione corretamente, é necessário seguir certas regras, desde começar com a fonte maiúscula, até terminar ou não com pontuação. A *Apple* fornece essas informações, porém elas não constam em um único local, o que acaba dificultando o processo.

5.1.2. Diretrizes e o *VoiceOver*

Além dos problemas levantados anteriormente, não existe uma documentação da empresa que mostre como a implementação de cada elemento se relaciona com diretrizes específicas de acessibilidade. Com isso em mente, foi desenvolvido um documento com diretrizes próprias para o *VoiceOver*, baseado nos três documentos de acessibilidade estudados durante esse artigo e nas especificações da própria *Apple*.

Foi feita uma análise das diretrizes buscando encontrar as que se relacionassem com a tecnologia em questão. Na tabela 1, pode-se ver todas as métricas analisadas.

Tabela 1. Diretrizes estudadas nas listas de recomendações de acessibilidade.
Fonte: autores

WCAG	eMAG	WAI
Diretriz 1.1	Recomendação 1.5	Diretriz 1
Diretriz 1.2	Recomendação 1.8	Diretriz 2
Diretriz 2.4	Recomendação 2.2	Diretriz 8
Diretriz 3.1	Recomendação 3.3	Diretriz 12
Diretriz 3.3	Recomendação 3.5	Diretriz 13
Diretriz 4.1	Recomendação 3.6	-
-	Recomendação 6.1	-
-	Recomendação 6.3	-

5.1.3. Desenvolvimento das diretrizes

Com isso, foi possível construir as diretrizes do *VoiceOver*. No total, foram desenvolvidas quatro principais: Perceptível, Operável, Compreensível e Multimídia. Dentro de cada uma, existem sub-diretrizes que buscam abranger todos os tópicos das requisições da *Apple* e das outras diretrizes. Cada diretriz está descrita nas seguintes tabelas: Tabela 2, Tabela 3, Tabela 4, Tabela 5.

Tabela 2. Primeira Diretriz, chamada Perceptível.

Fonte: autores

Perceptível	
1.1 <i>Accessibility Label</i>	Essa propriedade identifica o elemento de uma forma sucinta e clara. Para tal, existem certas regras a serem seguidas: <ul style="list-style-type: none">• O tipo do elemento não deve ser incluído;• Deve-se descrever o elemento com apenas uma palavra;• A palavra deve começar em maiúsculo;• O uso de ponto final é desencorajado;• A sua palavra deve ser localizada.
1.2 <i>Accessibility Trait</i>	Existe uma variedade de tipos de elementos. Com o uso de elemento não-nativos, é de suma importância o uso dessa propriedade, já que a partir dessa escolha, o usuário entenderá melhor a funcionalidade do elemento. As opções existentes no <i>VoiceOver</i> são: <i>none, button, link, searchField, image, selected, playsSound, keyboardKey, staticText, summaryElement, notEnabled, updatesFrequently, startsMediaSession, adjustable, allowsDirectInteraction, causesPageTurn, header</i> . Elementos mutuamente excludentes não devem ser combinados.
1.3 <i>Accessibility Hint</i>	Essa propriedade não deve ser utilizada se o restante das propriedades já informam o suficiente para a explicação do elemento. Ao estruturar a frase, deve-se considerar o resultado da ação do elemento. Caso ela precisa ser utilizada, deve-se seguir as seguintes regras: <ul style="list-style-type: none">• A frase deve começar com um verbo que nomeia o resultado da ação do elemento;• O tipo do elemento não deve ser incluído;• A frase deve começar com uma palavra em maiúsculo para trazer a inflexão correta;• A frase deve ter, no máximo, 8 (oito) palavras e terminar com ponto final.
1.4 Braille	O <i>emphVoiceOver</i> também oferece descrições em braille. Pode-se utilizar um display em braille ou escrever diretamente na tela touch com a Entrada de Braille. O braille é convertido automaticamente para texto e aparece em um painel na tela.

Tabela 3. Segunda Diretriz, chamada Operável.
Fonte: autores

Operável	
2.1 <i>Accessibility Element</i>	Os elementos no decorrer da aplicação devem ter a propriedade essa propriedade ativa. O uso de <i>Accessibility Traits</i> (1.2) contribui na especificação do objetivo de cada elemento. Deve-se considerar o uso de agrupamento de elementos (2.3) para tornar a experiência para usuários do mais fluida.
2.2 Páginas com títulos	O uso de páginas com títulos é essencial. Toda página deve ter um cabeçalho que vá descrevê-la com o uso de, no máximo, três palavras. Torna-se muito importante que o desenvolvedor garanta essa informação em cada <i>view</i> . Isso permite que o usuário entenda aonde se encontra dentro da aplicação. O uso do <i>Accessibility Trait</i> (1.2) chamado header é o melhor caminho para essa especificação.
2.3 Ordem de foco	O <i>VoiceOver</i> pode ser configurado para que a sequência de navegação faça sentido e seja fácil de operar. Ao navegar, o <i>VoiceOver</i> visita cada elemento na ordem em que aparece na tela. Algo possível, é a especificação da ordem de leitura dos elementos utilizando o <i>Accessibility Element</i> (2.1). Além disso, é possível agrupar elementos de acessibilidade, com a propriedade <i>accessibilityFrameInContainerSpace</i> .
2.4 <i>Used Links</i>	Cada <i>emphlink</i> deve ser identificável pelo seu texto ou pelo texto com o contexto dele. Com o uso do <i>emphAccessibility Trait</i> (1.2), pode-se especificar que o elemento é um link. Além disso, ao invés de permitir que o <i>emphVoiceOver</i> leia o nome do link, pode-se especificar ele melhor com o uso da <i>emphAccessibility Label</i> (1.1) e <i>emphHint</i> (1.3). Quando um link é apenas um elemento dentro de outro elemento, necessita especificar qual parte do texto deve ser considerada como um link. Para isso, utiliza-se o <i>UIAccessibilityCustomRotor</i>

Tabela 4. Terceira Diretriz, chamada Compreensível.
Fonte: autores

Compreensível	
3.1 Legibilidade	Todo conteúdo de texto na aplicação deve ser legível e compreensível. Para garantir isso na aplicação, necessita-se do <i>Accessibility Label</i> (1.1) e <i>Hint</i> (1.3) bem implementados.
3.2 Rótulos ou Instruções	Tendo entrada de dados, deve-se especificar rótulos ou instruções para que o objetivo desse elemento seja evidente e compreensível. Com o uso do <i>Accessibility Traits</i> (1.2) e <i>Label</i> (1.1), isso pode ser atingido.
3.3 <i>Accessibility Value</i>	Quando ocorre uma alteração no elemento, o <i>VoiceOver</i> indicará essa alteração através dessa propriedade. Geralmente, essa propriedade é utilizada para elementos que possuem algum tipo de variação. Um bom exemplo seria o volume do som.

Tabela 5. Quarta Diretriz, chamada Multimídia.
Fonte: autores

Multimídia	
4.1 Gravação e reprodução de mídias	A experiência com o <code>emphVoiceOver</code> para a gravação e reprodução de mídias pode perder fluidez quando não é devidamente configurada. Isso ocorre em consequência da sequência lógica de uso dele. Sendo assim, utilizando a <i>Accessibility Trait</i> (1.2) correta pode melhorar a experiência. A melhor opção para esses casos seria o <i>startsMediaSession</i> .
4.2 Descrição de mídias	As imagens devem ter uma descrição detalhada. Com a atualização do <i>iOS 14</i> , a tecnologia da <i>emphApple</i> que fornece a descrição das imagens sofreu uma atualização. Porém, a descrição pode não ser o suficiente. O uso da <i>Accessibility Label</i> (1.1) e <i>Hint</i> (1.3) são indicadas. Deve-se dar preferência <i>Label</i> já que as <i>Hints</i> podem ser desativadas. Assim, as regras da <i>emphAccessibility Label</i> devem ser consideradas.

Com isso, foi possível passar para a próxima etapa do desenvolvimento: a construção do validador.

5.2. Validador de Acessibilidade

Para a criação do validador, foi criado um projeto na plataforma *Xcode*, disponibilizada pela *Apple* para desenvolvimento de *softwares*. Para validação, testes unitários foram aplicados para garantir que as diretrizes fossem seguidas. Cada teste unitário se refere a uma diretriz, podendo ser escalável para abordar novas diretrizes.

O teste unitário é uma solução aplicada para separar e testar pequenos trechos de código, geralmente utilizado em funções, validando erros de código e ações inesperadas de entrada de dados [Runeson 2006]. Esta solução se encaixa no validador por serem testados dados e propriedades específicas dos elementos em tela, sendo que cada uma das diretrizes é separada em um único teste unitário, validando cada atributo.

Tratando-se dos testes, foi possível testar valores dentro de variáveis específicas. Por conta disso, nem todas as diretrizes puderam ser abordadas por essa estratégia, já que muitas dependem da interpretação e do contexto da aplicação. A Tabela 6 apresenta a diretriz e o código feito para o respectivo teste.

Todos os testes foram feitos usando uma função própria disponibilizada pelo *Swift*, linguagem de programação usada no desenvolvimento. A função de testes *XCTAssertTrue* verifica se o valor interno retorna como verdadeiro, e, caso afirmativo, o elemento passa no teste.

A usabilidade do validador foi feita para ser a mais simples possível, não precisando importar nenhum pacote dentro do arquivo da tela. Tudo foi criado dentro da parte de testes unitários, onde foi referenciada a tela para teste.

Dentro do contexto de aplicativos nativos *iOS*, para a utilização foi utilizado a inicialização da *ViewController* dentro do arquivo de testes unitários. Isso se dá por conta da necessidade da existência da *ViewController* antes dos testes serem executados. Para melhores práticas, cada *ViewController* deve ter seu próprio arquivo de testes unitários, fornecidos pela plataforma *Xcode*.

Tabela 6. Código para cada tipo de variável

Fonte: autores

Diretriz	Teste Unitário
Accessibility Label	XCTAssertTrue(!(aView.accessibilityLabel?.isEmpty ?? true))
Accessibility Trait	XCTAssertTrue(!(aView.accessibilityTraits.isEmpty))
Accessibility Hint	XCTAssertTrue(!(aView.accessibilityHint?.isEmpty ?? true))
Accessibility Element	XCTAssertTrue((aView.isAccessibilityElement))
Página com Títulos	XCTAssertTrue(!(vc.title?.isEmpty ?? true))
Accessibility Value	XCTAssertTrue(!(aView.accessibilityValue?.isEmpty ?? true))

5.3. Validação

Foi desenvolvido um aplicativo com o único intuito de realizar os testes unitários. Porém, ele também funciona como uma lista com as diretrizes desenvolvidas, vide Figura 2.

O desenvolvimento do aplicativo com as diretrizes em foco tornou o processo de acessibilidade mais simples, já que a categorização faz o desenvolvedor passar por cada uma das etapas antes de terminar o código. Além disso, com a garantia dos testes unitários, pode-se ter a certeza de que tudo está dentro dos padrões esperados. O validador pode se tornar muito útil em empresas de desenvolvimento, onde são criados padrões para garantir a qualidade do código, e, conseqüentemente, da acessibilidade.

A lista faz referência as diretrizes elaboradas para este trabalho, e foi aplicado em seus elementos os nossos testes de acessibilidade, fazendo dele um aplicativo acessível de acordo com os tópicos levantados.

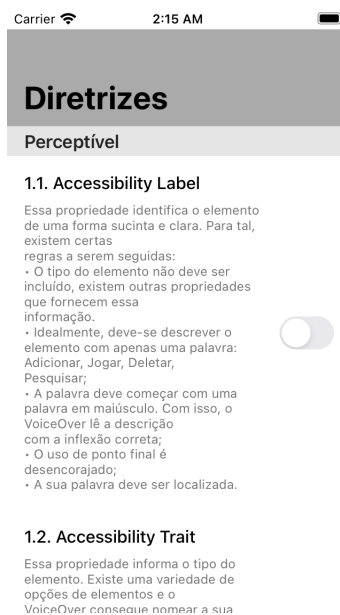


Figura 2. Aplicativo desenvolvido para o validador.

Fonte: autores

5.4. Mudanças

Após a validação do processo proposto, pode-se analisar o que era necessário mudar dentro do validador e das *guidelines*.

O validador sofreu mudanças principalmente para torná-lo mais genérico para qualquer tipo de tela, tornando-o mais útil para qualquer situação. Antes do aplicativo existir, não era possível fazer a busca pelos elementos em toda a tela, sendo necessário a passagem da referência para cada um dos elementos que desejasse testar.

Já dentro das métricas, houve uma análise da permanência ou retirada das *guidelines* que não se adequam ao validador desenvolvido. Decidiu-se por manter todas as diretrizes atuais, e realizar apenas a adição na diretriz 1.2 para uma melhor especificação dos tipos de *Accessibility Traits*. Segue essa adição:

- *none*: Usado para remover qualquer funcionalidade do *Accessibility Trait*;
- *button*: Objetos chamados de botão, ou seja, que aceitam eventos de entrada básicos;
- *link*: Aplicado apenas quando a interação do usuário com o controle os tire do aplicativo;
- *searchField*: Elementos de busca;
- *image*: Aplicado quando a aparência visual do objeto apresenta informações importantes, podendo ser associado com outros *traits*;
- *selected*: Elementos que estão selecionados, como botões que estão em foco;
- *playsSound*: Um elemento que acionará o som uma vez ativado. Diz ao *VoiceOver* para parar de anunciar assim que o usuário ativar o elemento;
- *keyboardKey*: Um elemento que atua como uma tecla em um teclado;
- *staticText*: Texto que não varia ao longo do ciclo de vida do aplicativo. Isso informa ao leitor de tela que não é necessário verificar se o valor deste elemento mudou;
- *summaryElement*: Cada *view* pode ter apenas um *summaryElement*, devendo ser um elemento que terá foco mais a diante porém merece destaque. Ele será lido logo que a *view* carregar;
- *notEnabled*: Quando deseja desativar um botão temporariamente;
- *updatesFrequently*: Ajuda o leitor a lidar melhor com os controles com conteúdo dinâmico. Nesses casos, ele pesquisará o controle enquanto ele está focado, em vez de tentar compartilhar todo o texto que já existiu nele. Também deve ser usado em circunstâncias que o usuário deve ser notificado sobre alterações de conteúdo, sem ter foco no elemento;
- *startsMediaSession*: Elementos que tocam ou gravam sons, não tocando o som automaticamente e não repetindo a *label*;
- *adjustable*: Utilizado com *UISlider* ou elementos que conseguem alterar por si só;
- *allowsDirectInteraction*: Informa ao *emphVoiceOver* que não deve haver nenhum desvio do controle de toque padrão para esta visualização;
- *causesPageTurn*: Indica que o conteúdo representa uma página de um conjunto de páginas;
- *header*: Adiciona a informação de que o elemento é um cabeçalho e torna a navegação mais fácil, devendo existir pelo menos um cabeçalho por visualização.

6. Conclusões

O objetivo principal deste projeto foi construir *guidelines* específicas de acessibilidade que ajudassem desenvolvedores a configurar a tecnologia de leitura de tela da *Apple*, o *VoiceOver*. Através do sistema desenvolvido, pode-se exemplificar como a criação dessas diretrizes e de um validador para acompanhá-las, facilita esse processo.

O uso de *guidelines* mais específicas para *mobile* trazem pontos específicos que são muito importantes de serem analisados dentro dos aplicativos. Além disso, a união dessas diretrizes com o validador facilitam a aplicação real delas. Por fim, notou-se a necessidade do versionamento dessas diretrizes para que elas se atualizem com as novas tecnologias.

A lista de diretrizes proposta apresenta uma visão objetiva e clara das necessidades de uso da tecnologia em questão para a melhor implementação dela. Para isso, a lista contém uma relação direta com os requerimentos de outras *guidelines* de acessibilidade. Nota-se, então, que isso contribui para o destaque e enfoque em uma acessibilidade específica, garantindo que ela seja trabalhada da melhor forma possível, algo destacado por Siebra et al (2015).

Já o desenvolvimento do validador transforma a experiência do desenvolvedor. A consulta e acesso das informações necessárias das listas de *guidelines* muitas vezes é escassa, sendo um trabalho realizado após o desenvolvimento. O validador passa a ser um passo que ajuda a detectar falhas no processo com maior antecedência, tornando o trabalho de validação da acessibilidade menos custoso e trabalhoso. Porém, deve-se salientar que ele não substitui testes de usabilidade da acessibilidade do produto, essa ainda é uma etapa necessária para qualquer projeto.

O mais importante objetivo deste projeto foi mostrar a necessidade da criação de métricas específicas para auxiliar desenvolvedores no processo de aplicação de tecnologias de acessibilidade em seus projetos, bem como uma facilitação desse processo com o uso do validador. Há muito ainda a ser desenvolvido a partir desse projeto, buscando garantir maior acessibilidade dentro do mundo da tecnologia *mobile*.

7. Trabalhos Futuros

Há muito ainda a ser estudado nessa área de métricas e validadores para a acessibilidade em dispositivos *mobile*. Sendo assim, uma análise de outras tecnologias de acessibilidade, com a criação de suas próprias métricas e validadores, poderá ser um próximo passo. Bem como um estudo mais aprimorado de *guidelines* mais práticas, para que possam ser implementadas no próprio validador.

Deseja-se, em trabalhos futuros, realizar maiores testes com desenvolvedores para validar as diretrizes e o validador. Bem como, pretende-se realizar uma melhoria no validador para que ele passe a ser uma ferramenta de maior apoio aos desenvolvedores.

Espera-se que ocorra uma menção ao presente trabalho em outros projetos, onde seja possível que desenvolvedores consigam usufruir dos produtos desenvolvidos para aprimorar a acessibilidade de seus produtos.

Referências

- Au, F. T. W., Baker, S., Warren, I., and Dobbie, G. (2008). Automated usability testing framework. In *Proceedings of the Ninth Conference on Australasian User Interface - Volume 76*, AUIC '08, page 55–64, AUS. Australian Computer Society, Inc.
- Caldwell B., Reid L., V. G. (2008). Web content accessibility guidelines (wcag). <https://www.w3.org/TR/WCAG20>.
- Calvo, R., Seyedarabi, F., and Savva, A. (2016). Beyond web content accessibility guidelines: Expert accessibility reviews. In *Proceedings of the 7th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-Exclusion*, DSAI 2016, page 77–84, New York, NY, USA. Association for Computing Machinery.
- Clegg-Vinell, R., Bailey, C., and Gkatzidou, V. (2014). Investigating the appropriateness and relevance of mobile web accessibility guidelines. In *Proceedings of the 11th Web for All Conference*, W4A '14, New York, NY, USA. Association for Computing Machinery.
- eMAG (2021). *Modelo de Acessibilidade em Governo Eletrônico*. <http://emag.governoeletronico.gov.br>, 3a edição.
- IBGE, B. (2011). Pnad: De 2005 para 2011, número de internautas cresce 143,8% e o de pessoas com celular, 107,2%. <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/14404-asi-pnad-de-2005-para-2011-numero-de-internautas-cresce-1438-e-o-de-pessoas-com-celular-1072>. Agência IBGE Notícias.
- Runeson, P. (2006). A survey of unit testing practices. *IEEE Software*, 23(4):22–29.
- Scholtz, J. (2001). Usability evaluation.
- Sharp, H., Rogers, Y., and Preece, J. (2007). *Interaction Design: Beyond Human Computer Interaction*. John Wiley Sons, Inc., Hoboken, NJ, USA.
- Siebra, C., Gouveia, T., Macedo, J., Correia, W., Penha, M., Silva, F., Santos, A., Anjos, M., and Florentin, F. (2015). Usability requirements for mobile accessibility: A study on the vision impairment. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, MUM '15, page 384–389, New York, NY, USA. Association for Computing Machinery.
- WAI (2021). *Web Accessibility Initiative*. <https://www.w3.org/WAI/>, 3a edição.