UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Uma abordagem para verificação de acessibilidade e usabilidade em aplicativos móveis

Marisa Helena da Silva Batista

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Marisa Helena da Silva Batista

Uma abordagem para verificação de acessibilidade e usabilidade em aplicativos móveis

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestra em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Renata Pontin de Mattos Fortes

USP – São Carlos
Novembro de 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

B333a   Batista, Marisa Helena da Silva
            Uma abordagem para verificação de acessibilidade
            e usabilidade em aplicativos móveis / Marisa Helena
            da Silva Batista; orientadora Renata Pontin de
            Mattos Fortes. -- São Carlos, 2018.
            205 p.

            Dissertação (Mestrado - Programa de Pós-Graduação
            em Ciências de Computação e Matemática
            Computacional) -- Instituto de Ciências Matemáticas
            e de Computação, Universidade de São Paulo, 2018.

            1. Acessibilidade. 2. Usabilidade . 3. Interação
            humano-computador. 4. Aplicativo móvel. 5. Avaliação.
            I. Fortes, Renata Pontin de Mattos, orient. II.
            Título.

Marisa Helena da Silva Batista

An approach to verifying accessibility and usability in mobile applications

Master dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Renata Pontin de Mattos Fortes

USP – São Carlos
November 2018

Este trabalho é dedicado aos meus pais Soeli e Laércio, minha irmã Gabriela, meus tios Maria e Dieter Werner Plaas, meu namorado Christian Lemos e amigos, que sempre me apoiaram e sem os quais não seria possível chegar onde hoje estou.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela oportunidade concedida.

Agradeço a professora Renata Pontin de Mattos Fortes pela orientação deste trabalho.

Agradeço também a meus pais Soeli Silva e Laércio Batista, minha irmã Gabriela Batista e a meus tios Dieter e Maria Plaas pelo apoio dedicado.

Agradeço ao meu namorado Christian Lemos, por todo o apoio e companheirismo.

Agradeço a Camila Oliveira, pelo auxílio durante a elaboração do *checklist*.

Agradeço a CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela concessão da bolsa durante uma parte do período de realização desta pesquisa.

Também agradeço a todos os professores em todos os níveis de escolaridade pelos quais passei, e todos os que contribuíram de alguma forma para que eu chegasse onde estou.

*“Uma grande viagem
começa com um único passo.”
(Lao-Tsé)*

RESUMO

BATISTA, M. H. S. **Uma abordagem para verificação de acessibilidade e usabilidade em aplicativos móveis**. 2018. 205 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Devido à evolução digital e popularização dos dispositivos de interação, diversas tarefas do cotidiano anteriormente realizadas fisicamente hoje são executadas por meio da tela de um computador ou de um celular. Com o grande fluxo de informações, as pessoas precisam manter-se mais conectadas aos serviços da Internet, por mais tempo no dia-a-dia, para se manterem atualizadas. Por serem mais práticos e compactos do que computadores pessoais, os dispositivos móveis são utilizados cada vez mais para esse fim, e intensificam a geração de aplicativos para atender as diversas demandas. Porém, para uma inserção completa da população nessa constante evolução digital, é fundamental que os aplicativos para *mobile* também ofereçam possibilidade de acesso aos diferentes perfis de usuários, independentemente de suas deficiências ou limitações. Em busca da qualidade, produtividade e rapidez na criação de aplicações, existe uma vasta gama de boas práticas de desenvolvimento. Apesar da existência de legislação específica, no panorama atual ainda existem muitos problemas de acessibilidade e usabilidade que precisam ser resolvidos no contexto de uso em dispositivo móvel. Neste projeto foi elaborada uma abordagem, que incorpora conceitos de boas práticas para desenvolvimento móvel associando-se a diretrizes e recomendações disseminadas na literatura, visando compor um *checklist* sobre acessibilidade e usabilidade em aplicativos móveis. Para tanto, nesta pesquisa foram realizadas uma revisão de literatura, estudos sobre as boas práticas de desenvolvimento de aplicativos em dispositivos móveis e sobre as recomendações de acessibilidade e de usabilidade existentes. O *checklist* desenvolvido, denominado *Acc_MobileCheck*, foi validado por especialistas e desenvolvedores.

Palavras-chave: Interação Humano-Computador, Acessibilidade, Usabilidade, Acessibilidade, Desenvolvimento, Boas Práticas, Aplicativos móveis, Tablets, Smartphones.

ABSTRACT

BATISTA, M. H. S. **An approach to verifying accessibility and usability in mobile applications**. 2018. 205 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Due to the digital revolution and popularization of interaction devices, various daily tasks previously accomplished through physical means are now being carried out by computers or smartphones. The huge flow of information requires people to be connected to internet services for longer on a day-to-day basis. Mobile devices are becoming more and more common to this end, because they are more practical and smaller than personal computers; this encourages the development of mobile applications to attend the demands. However, for a complete insertion of the population in this digital evolution, it is essential that mobile applications offer the possibility of access to different user profiles, regardless of their disabilities or limitations. A wide array of best practices of software development has been available seeking quality, productivity and quickness on the development of applications. While there is specific legislation, in the current panorama, there still exists many problems of accessibility and usability in the context of mobile applications that need to be solved. In this project, an approach incorporating the concepts of best practices for mobile development, associating guidelines and recommendations present in the literature, was developed aiming to compose a checklist about accessibility and usability in mobile applications. For this goal, the research followed along a bibliography review and a study of best practices for mobile applications development and existing recommendations for accessibility and usability. The complete checklist, denominated *Acc_MobileCheck*, was then validated by experts and developers.

Keywords: Human-Computer Interaction, Accessibility, Usability, Development, Good Practices, Mobile Apps, Tablets, Smartphones.

LISTA DE ILUSTRAÇÕES

Figura 1 – Dados de Internet móvel e fixa até 2015	25
Figura 2 – Acesso geral e móvel à Internet no mundo	26
Figura 3 – Percentual de pessoas com 10 anos ou mais de idade com posse de telefone móvel para uso pessoal no Brasil classificado por faixa etária	27
Figura 4 – Porcentagem de pessoas com ao menos um tipo de deficiência por grupo de faixa etária	28
Figura 5 – Etapas da metodologia de pesquisa	29
Figura 6 – Captura de tela mostrando técnica de melhor prática do MWBPWG	41
Figura 7 – Capturas de tela do aplicativo <i>Google Maps</i> nos diferentes métodos de entrega	53
Figura 8 – Exemplos do UIDP <i>Input Feedback</i>	58
Figura 9 – Aba “Usabilidade” na página oficial do <i>Material Design</i>	60
Figura 10 – O aplicativo <i>Travelocity</i> implementa exemplo típico do UIDP <i>mobile List of Links</i>	63
Figura 11 – <i>List of Links</i> implementado no <i>Google Plus</i> com informações de atualização	64
Figura 12 – O aplicativo <i>Southwest Airlines</i> mostra uma variação agrupada do padrão <i>List of Links</i>	65
Figura 13 – O padrão <i>List of Links</i> suporta múltiplas AIs	65
Figura 14 – Exemplo de implementação do UIDP <i>Voice Search</i>	67
Figura 15 – Exemplo de implementação do UIDP <i>Auto-Suggest</i>	68
Figura 16 – Exemplo de implementação do UIDP <i>Search from Action Bar</i>	69
Figura 17 – Exemplos de implementações do UIDP <i>Tabs</i> em aplicativos nativos Android	70
Figura 18 – Exemplos de implementações das duas versões UIDP <i>Multiple Select</i>	71
Figura 19 – Exemplo de implementação do UIDP <i>Textbox with input mask</i>	72
Figura 20 – Exemplo de implementação do UIDP <i>Inline error message</i>	73
Figura 21 – Exemplo de implementação do UIDP <i>Callback Validation</i>	73
Figura 22 – Exemplo do UIDP <i>Cancel/Ok</i> apresentado no modo de operação com uma mão do Android	74
Figura 23 – Exemplos de implementação do UIDP <i>Cancel/Ok</i>	75
Figura 24 – Exemplo de implementação do UIDP <i>Carousel</i> em um aplicativo fictício	76
Figura 25 – Componentes integrantes para desenvolvimento de <i>Acc_MobileCheck</i>	80
Figura 26 – Processo de desenvolvimento do <i>Acc_MobileCheck</i>	85
Figura 27 – Relação entre os tipos de problemas e de deficiências, observada nos itens de verificação do <i>Acc_MobileCheck</i>	87

Figura 28 – Implementações de UIDPs <i>mobile</i> diferentes na mesma tela	88
Figura 29 – Componentes analisados para proposta dos itens de verificação do <i>Acc_-MobileCheck</i>	89
Figura 30 – Nota média SUS atribuída por cada avaliador.	99
Figura 31 – Nota média SUS dos tipos de problema por tipo de avaliador	100
Figura 32 – Nota média SUS atribuída pelos desenvolvedores e especialistas.	101
Figura 33 – Criação de nova conta no aplicativo <i>Instagram</i>	178
Figura 34 – Erros de inserção possíveis na criação de conta no <i>Instagram</i> no passo exibido na Figura 33b	179
Figura 35 – Busca, bloqueio e desbloqueio de usuário no aplicativo <i>Instagram</i>	181
Figura 36 – Busca por cadeia de caracteres no Android 8.0.0	182
Figura 37 – Pesquisa genérica a partir da tela inicial do aplicativo <i>WhatsApp</i>	182
Figura 38 – Criação de nova conta de e-mail pelo aplicativo Gmail	185
Figura 39 – Chaveamento entre contas e mudança de marcador em múltiplos e-mails no aplicativo Gmail	186
Figura 40 – Digitação de destinatário de e-mail no aplicativo nativo Gmail	187
Figura 41 – Postagem de múltiplas fotos no aplicativo nativo <i>Instagram</i>	188
Figura 42 – Realização de pesquisa por voz utilizando o aplicativo nativo Google App Search	189
Figura 43 – Autenticação e registro de cuidador no aplicativo MobiCare	192
Figura 44 – Agendamento de lembrete para remédio no aplicativo MobiCare	193
Figura 45 – <i>Login</i> de usuário no aplicativo OCE	194
Figura 46 – Cadastro de usuário no aplicativo OCE	194
Figura 47 – Envio de relato de intensidade chuvosa no aplicativo OCE	195
Figura 48 – Visualização de relatos de altura de água no leito de rio no aplicativo OCE	196
Figura 49 – Quantidade real e proporcional de UIDPs web encontrados entre os sites, dividida por funcionalidade	202
Figura 50 – Número de <i>websites</i> que utilizam o UIDP	203
Figura 51 – Exemplo dos quatro UIDP com ocorrência em todos os <i>websites</i> analisados	204
Figura 52 – Número de <i>websites</i> dentro de uma categoria que utilizam o UIDP	205

LISTA DE QUADROS

Quadro 1 – Diretrizes WCAG 2.0 resumidas	40
Quadro 2 – Exemplo da estrutura de um dos itens de verificação do <i>Acc_MobileCheck</i>	90

LISTA DE TABELAS

Tabela 1 – Categorias dos 12 UIDPs <i>mobile</i> estudados	62
Tabela 2 – Os 12 UIDPs <i>mobile</i> analisados	77
Tabela 3 – Aplicativos móveis selecionados para investigação por UIDP	82
Tabela 4 – Distribuição das tarefas e potenciais tipos de problemas aos especialistas . .	94
Tabela 5 – Distribuição das tarefas e potenciais tipos de problemas aos desenvolvedores	95
Tabela 6 – Avaliação da TE2 pelo E2 usando o conjunto Operação (O) do <i>Acc_Mobile-Check</i>	97
Tabela 7 – Valores SUS na Avaliação por desenvolvedores (D_i) e especialistas (E_i) . . .	98
Tabela 8 – Média dos tipos de problema por desenvolvedor e especialista	100
Tabela 9 – Trabalhos relacionados	107
Tabela 10 – Relação entre itens de verificação de compreensão [C] e UIDPs <i>mobile</i> . . .	174
Tabela 11 – Relação entre itens de verificação de adaptação [A] e UIDPs <i>mobile</i>	174
Tabela 12 – Relação entre itens de verificação de operação [O] e UIDPs <i>mobile</i>	175
Tabela 13 – Relação entre itens de verificação de percepção [P] e UIDPs <i>mobile</i>	175
Tabela 14 – Tarefas atribuídas aos especialistas (TEs) para avaliação usando <i>Acc_MobileCheck</i> e UIDPs <i>mobile</i> existentes	177
Tabela 15 – Tarefas não utilizadas na validação, que foram elaboradas para os especialistas (TEs) com os respectivos UIDPs <i>mobile</i> existentes	183
Tabela 16 – UIDPs <i>mobile</i> e tarefas dos desenvolvedores nas quais aparecem	191
Tabela 17 – <i>websites</i> selecionados na lista do Alexa	200
Tabela 18 – UIDPs encontrados nos <i>sites</i> observados	201

LISTA DE ABREVIATURAS E SIGLAS

ASES	Avaliador e Simulador de Acessibilidade em Sítios
CSS	<i>Cascading Style Sheets</i>
eMAG	Modelo de Acessibilidade do Governo Eletrônico
EPD	Estatuto da Pessoa com Deficiência
IBGE	Instituto Brasileiro de Geografia e Estatística
ICMC	Instituto de Ciências de Matemáticas e de Computação
IHC	Interação Humano-Computador
MWABP	<i>Mobile Web Application Best Practices</i>
MWBP	<i>Mobile Web Best Practices</i>
MWBPWG	<i>Mobile Web Best Practices Working Group</i>
MWI	<i>Mobile Web Initiative</i>
PPG/CCMC	Programa Pós-Graduação em Ciências de Computação e Matemática Computacional
SO	Sistema Operacional
SUS	<i>System Usability Scale</i>
TIC	Tecnologia de Informação e Comunicação
UI	<i>User Interfaces</i>
UIDP	<i>User Interfaces Design Patterns</i>
USP	Universidade de São Paulo
UX	<i>User Experience</i>
W3C	<i>World Wide Web Consortium</i>
WAI	<i>Web Accessibility Initiative</i>

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Contextualização e Motivação	25
1.2	Objetivo	29
1.3	Metodologia de pesquisa	29
1.4	Organização do documento	30
2	FUNDAMENTOS	31
2.1	Considerações Iniciais	31
2.2	Qualidade de Software	32
2.3	Usabilidade	33
2.4	Acessibilidade	37
2.5	Avaliação de usabilidade e acessibilidade de UI	46
2.6	Considerações Finais	49
3	DESENVOLVIMENTO MÓVEL - PADRÕES DE PROJETO	51
3.1	Considerações Iniciais	51
3.2	Desenvolvimento de Aplicativos Móveis	51
3.3	<i>Design Patterns</i> como boa prática	54
3.4	Outras Boas Práticas	58
3.4.1	<i>Material Design</i>	58
3.4.2	<i>Google Accessibility e boas práticas Android</i>	61
3.5	<i>Mobile User Interface Design Patterns</i>	62
3.6	Considerações Finais	76
4	DESENVOLVIMENTO DO <i>CHECKLIST</i>	79
4.1	Considerações Iniciais	79
4.2	Decisões de projeto	81
4.3	<i>Acc_MobileCheck</i>	84
4.4	Validação da <i>Acc_MobileCheck</i>	92
4.5	Análise dos resultados da validação	99
4.6	Trabalhos Relacionados	103
4.7	Considerações Finais	107
5	CONCLUSÕES	109

5.1	Considerações Iniciais	109
5.2	Principais Contribuições	110
5.3	Limitações e trabalhos futuros	110
REFERÊNCIAS		113
APÊNDICE A	ITENS DE VERIFICAÇÃO DA <i>ACC_MOBILECHECK</i>	123
APÊNDICE B	RELAÇÃO UIDP <i>MOBILE</i> E ITENS DE VERIFICAÇÃO <i>ACC_MOBILECHECK</i>	173
APÊNDICE C	TAREFAS DOS ESPECIALISTAS	177
C.1	Tarefas realizadas pelos especialistas	177
C.2	Tarefas não realizadas	183
APÊNDICE D	TAREFAS DOS DESENVOLVEDORES	191
APÊNDICE E	TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO	197
APÊNDICE F	LEVANTAMENTO DE UIDP NA <i>WEB</i>	199
F.1	Estudo preliminar de utilização de UIDP no desenvolvimento web	199

INTRODUÇÃO

1.1 Contextualização e Motivação

Com o constante aumento do volume de informações e popularização dos computadores pessoais, o acesso à Internet tem se tornado cada vez mais presente e essencial para os cidadãos. O número de pessoas conectadas à rede está em constante ascensão; segundo [IBGE \(2015\)](#) a quantidade de internautas no Brasil praticamente quintuplicou em 9 anos. Esse cenário não ocorre apenas nacionalmente, como pode ser observado na [Figura 1](#), que exibe o constante crescimento do uso da Internet mundialmente, com destaque para utilização de telefonia móvel, chegando a 96,8% habitantes/famílias com assinaturas de telefonia móvel.

Figura 1 – Dados de Internet móvel e fixa até 2015

Fonte: Adaptada de [ITU \(2015\)](#). Tradução nossa.

Paralelamente à popularização da Internet, houve um crescimento na quantidade de aparelhos que pudessem acessá-la, contribuindo para que os dispositivos móveis evoluíssem tecnologicamente (NETO, 2013). Assim, tornaram-se o meio de acesso preferencial dos usuários em relação a computadores pessoais (vide Figura 1). Devido à constante evolução e criação de novos dispositivos de interação, a computação tem se tornado cada vez mais presente no cotidiano das pessoas, sendo inserida nas mais diversas atividades, antes realizadas sem auxílio tecnológico (MONTEIRO, 2006).

Segundo dados de 2016 (vide Figura 2), cerca de 47,1% da população mundial já possuía acesso à Internet (ITU, 2016). Além disso, o rápido crescimento desses acessos por meio de dispositivos móveis e suas tecnologias em evolução tem sido impressionante (vide Figura 2b).

Figura 2 – Acesso geral e móvel à Internet no mundo

Fonte: Adaptada de ITU (2016). Tradução nossa.

Montoliu e Gatica-Perez (2010) afirmam que dispositivos móveis vêm se tornando ferramentas multimídia naturais, dada a quantidade de funcionalidades que possibilitam. Segundo Poore-Pariseau (2010) e Neto (2013), para que o usuário possa ter uma boa interação com o software é necessário que sua interface possua uma boa **usabilidade**. Um software que possua boa usabilidade facilita o uso de uma grande parcela da população, porém existem diferentes perfis de usuários para os quais não é suficiente que o sistema seja apenas “usável” para que consigam utilizá-lo, mas também **acessível**, que é o caso das pessoas com deficiência^{1 2}.

¹ Nomenclatura atual convencionada pela ONU e adotada pelo Brasil - Resolução nº 1 de 15 de outubro de 2010.

² “Considera-se pessoa com deficiência aquela que tem impedimento de longo prazo de natureza física, mental, intelectual ou sensorial, o qual, em interação com uma ou mais barreiras, pode obstruir sua participação plena e efetiva na sociedade em igualdade de condições com as demais pessoas.” Capítulo 1, Art. 2º do Estatuto da Pessoa com Deficiência (BRASIL, 2015)

A [Figura 3](#) mostra dados de uma pesquisa realizada nacionalmente, indicando o aumento do percentual de pessoas que possuíam telefone móvel (celular), nos nove anos em que ocorreu a coleta de dados ([IBGE, 2015](#)). Os dados ressaltam os grupos que mais cresceram em porcentagem, como o de indivíduos com 55 a 59 anos (aumento de 155%) e de 60 anos ou mais (aumento de 230%) no número de usuários, o maior observado na pesquisa.

Figura 3 – Percentual de pessoas com 10 anos ou mais de idade com posse de telefone móvel para uso pessoal no Brasil classificado por faixa etária

Fonte: Adaptada de [IBGE \(2015\)](#).

A relação entre faixa etária e ocorrência de algum tipo de deficiência foi analisada no Brasil por [Oliveira \(2012\)](#), que afirma que a partir dos 60 anos alguma deficiência ocorre em mais da metade da população (vide [Figura 4](#)). Segundo [Oliveira \(2012\)](#), a partir de 55 anos de idade, mais de 45% da população possui algum tipo de deficiência, seja ela auditiva, visual, motora, cognitiva ou deficiência múltipla. No âmbito mundial, a Organização Mundial da Saúde ([WHO Press, 2011](#)) destaca que cerca de 650 milhões de pessoas vivem com alguma deficiência, o que equivale a aproximadamente 10% da população mundial. Esse cenário já demonstra a atual necessidade de que aplicações móveis sejam acessíveis, para garantir a inclusão desses usuários, além de manter boa usabilidade para todos os perfis.

A relação entre acessibilidade e usabilidade é pouco analisada explicitamente, tanto em relação à Web como nos demais sistemas computacionais ([POWER; PETRIE, 2007](#); [AMARAL, 2014](#)). No sentido de auxiliar o desenvolvimento de *sites* e aplicativos mais acessíveis várias soluções têm sido propostas, como a disponibilização de diretrizes de acessibilidade pela W3C (*World Wide Web Consortium*) com foco em Web, porém são pouco adotadas na prática

Figura 4 – Porcentagem de pessoas com ao menos um tipo de deficiência por grupo de faixa etária

Fonte: Adaptada de Oliveira (2012).

(GILBERTSON; MACHIN, 2012). No contexto de aplicativos móveis também foram criadas diretrizes, tanto pela W3C, como por outros pesquisadores/desenvolvedores, como é o caso do manual do desenvolvedor Android^{3,4}.

No entanto, como ressaltam Díaz-Bossini e Moreno (2014), não há um conjunto de recomendações que abranja todas as questões de acessibilidade em *smartphones* e *tablets*. Segundo Lazar, Dudley-Sponaule e Greenidge (2004), a relutância na adoção de recomendações, pelos desenvolvedores, ocorre porque requer mais tempo, trabalho e conhecimentos, além de ofuscar os benefícios que a utilização de boas práticas para desenvolvimento de aplicativos acessíveis possibilita.

Torna-se assim fundamental um apoio mais efetivo para o desenvolvimento de aplicativos móveis com acessibilidade. Desse modo, esta pesquisa de mestrado visa atender a lacuna existente de métodos de apoio ao desenvolvedor que visem a verificação de acessibilidade e usabilidade de modo integrado. Sobretudo, foi desenvolvida uma abordagem para facilitar a identificação de barreiras de acessibilidade e/ou problemas de usabilidade, objetivamente, durante o desenvolvimento de aplicativos móveis, a partir da verificação dos recursos que os desenvolvedores mais utilizam como boas práticas.

³ <https://developer.android.com/develop/index.html>

⁴ <https://developer.android.com/guide/topics/ui/accessibility/>

1.2 Objetivo

O objetivo geral deste trabalho foi pesquisar e identificar as boas práticas de desenvolvimento de *software* para dispositivos móveis, além de estudar as características de qualidade relativas a interfaces de aplicativos móveis. Em especial, dentre as boas práticas, foram enfatizados os *Design Patterns* mais adotados nos aplicativos móveis mais popularmente utilizados.

O objetivo específico foi definir uma abordagem para verificação de questões relacionadas com as boas práticas estudadas, adotadas no processo de desenvolvimento de aplicativos móveis, quanto aos potenciais barreiras e/ou problemas de acessibilidade e usabilidade.

1.3 Metodologia de pesquisa

A [Figura 5](#) esquematiza o fluxo dos procedimentos científicos realizados nesta pesquisa. As principais questões nesta pesquisa foram assim estabelecidas:

1. Quais questões de acessibilidade e usabilidade podem ser verificadas em *Design Patterns* existentes para desenvolvimento de aplicativos móveis?
2. Como integrar a verificação das questões de acessibilidade e usabilidade em um instrumento que agilize a identificação de barreiras de acessibilidade e/ou problemas de usabilidade durante o desenvolvimento de aplicativos móveis?

Figura 5 – Etapas da metodologia de pesquisa

Fonte: Elaborada pelo autor.

Para responder a essas questões de pesquisa, como pode ser visto na [Figura 5](#), inicialmente foram realizados estudos sobre os conceitos de **acessibilidade** e **usabilidade** (definições, diretrizes e legislação vigentes), os quais embasaram a caracterização dos **requisitos** de acessibilidade e usabilidade voltados para aplicativos móveis. Estudos sobre as **boas práticas** para

desenvolvimento de *User Interfaces* (UI) em especial os *Design Patterns* foram realizados também para embasar os levantamentos sobre *User Interfaces Design Patterns* (UIDP) no contexto web ([Apêndice F](#)) e de aplicativos móveis.

A partir desses estudos, foi realizado um **mapeamento** entre as perspectivas mais abstratas (requisitos, definições e diretrizes) sobre acessibilidade e usabilidade, e as perspectivas mais práticas (UIDP) adotadas no âmbito do desenvolvimento de aplicativos móveis.

Como um dos resultados desta pesquisa, foi desenvolvido um *checklist* com questões de acessibilidade e usabilidade para apoio ao desenvolvimento de aplicativos para dispositivos móveis. A seguir, esse *checklist* (denominado **Acc_MobileCheck**) foi submetido a uma validação. A proposta desse *checklist* é de que possa ser integrado ao processo de desenvolvimento que os desenvolvedores utilizem, em especial quando adotam boas práticas (*Design Patterns*). Assim, disseminar esse *checklist* para que possa continuamente evoluir de modo a atender as exigências atuais e das futuras gerações de usuários de diferentes perfis, incluindo pessoas com deficiência específica, ainda se apresenta como meta para evolução desta pesquisa.

1.4 Organização do documento

Neste capítulo foi apresentada uma visão geral sobre o projeto do mestrado, e introduzido o contexto da pesquisa. Nos próximos capítulos do documento, tem-se a seguinte organização: no [Capítulo 2](#) são apresentados os principais conceitos relacionados ao domínio no qual este trabalho se insere, foram estudadas definições, diretrizes sobre Qualidade de Software de modo abrangente, Acessibilidade e Usabilidade, procurando alinhar ao escopo de aplicativos móveis, bem como os métodos de avaliação consolidados na literatura científica; no [Capítulo 3](#), são descritos os trabalhos da literatura, relacionados com o desenvolvimento de aplicativos móveis, além de ser apresentada uma breve análise de Padrões de Projeto de Interface de Usuário em páginas web, por serem adotados preferencialmente pelos desenvolvedores profissionais. A seguir, no [Capítulo 4](#) é apresentado o *checklist Acc_MobileCheck*, como foi desenvolvido para integrar as questões de verificação que o compõe, bem como sua validação. Ao final, o [Capítulo 5](#) conclui esta dissertação, indicando as principais contribuições, limitações e possibilidades de pesquisas futuras.

FUNDAMENTOS

2.1 Considerações Iniciais

Usabilidade e acessibilidade são conceitos importantes no escopo de interfaces de *software* em geral, por estarem relacionados com questões inerentes à interação requerida dos usuários finais, ao utilizarem os *softwares* (LARA, 2013). Portanto, usabilidade e acessibilidade são considerados como atributos de qualidade de software. Segundo Coutinho *et al.* (2011), acessibilidade se apresenta como requisito não funcional. Thatcher *et al.* (2003) propõem que acessibilidade é um subconjunto de usabilidade, sugerindo que problemas de acessibilidade são tipos particulares de problemas de usabilidade. Argumentam que a diferença dos conceitos é que os problemas de usabilidade afetam todos os usuários igualmente, independentemente da capacidade ou deficiência, e os problemas de acessibilidade afetam pessoas com deficiência e idosos.

Para Chi (2013), há uma tênue diferença entre os objetivos de acessibilidade e usabilidade, embora não sejam diagonalmente opostos. O autor discute sobre essa falsa dicotomia, reforçada pela ideia de que existe um usuário “modelo” para o qual se deve projetar, quando na maioria das vezes o projeto e os testes são realizados com um conjunto de pessoas diferentes, por meio dos casos de uso e com níveis de habilidade distintos. Desse modo, ao insistir em projetos universais “sem barreiras” (acessíveis), a pesquisa sobre acessibilidade incide, geralmente, sobre questões de *design* teóricas e filosóficas que podem levar a soluções utópicas, ao invés de focar em projetos para beneficiar tanto pessoas com deficiências, como o público em geral (CHI, 2013).

Lara (2013) observa que ambos os conceitos (acessibilidade e usabilidade) possuem objetivos em comum, mas ressalta que, por vezes, características específicas de usabilidade prejudicam a acessibilidade de um sistema.

Pesquisas sobre acessibilidade e usabilidade constituem desafios científicos, na área de Interação Humano-Computador (IHC), e estão presentes no escopo deste trabalho de pesquisa,

além da área sobre qualidade de *software* de modo geral. Nas próximas seções são descritos os fundamentos, que foram estudados, para o desenvolvimento desta pesquisa, especialmente visando a compreensão dos requisitos de qualidade para compor uma abordagem para verificação de acessibilidade e usabilidade em aplicativos móveis. São descritas, ao final, as principais maneiras de avaliação de *User Interfaces* (UI), em termos de acessibilidade e usabilidade.

2.2 Qualidade de Software

Qualidade de *software* pode ser definida como um conjunto de características que um *software* precisa estar de acordo para que atenda às necessidades dos usuários (ROCHA; MALDONADO; WEBER, 2001). Para Guerra e Colombo (2009), qualidade de um *software* corresponde a conformidade em relação aos requisitos pré-definidos, de modo que possam ser observados com uso de medições. A série de normas ISO/IEC 25000, também conhecida como SQuaRE (*System and software Quality Requirements and Evaluation*), foi publicada em 2014 e tem como objetivo a criação de um *framework* para a avaliação da qualidade do produto de software. Esta ISO é um modelo de referência e constitui-se de um “guia” que orienta para adoção de outras normas, a partir da revisão de normas já publicadas, bem como visa a criação de novas normas que atendam requisitos do mercado (ISO, 2014).

A ISO/IEC 25010 é uma das divisões da ISO/IEC 25000, disponibilizada em 2011, e é resultado da evolução de vários outros padrões, em especial da ISO/IEC 9126, que define um modelo de qualidade para avaliação de produtos de *software*, e da ISO/IEC 14598, que define o processo de avaliação de produto de *software*. A norma ISO/IEC 25010 define as características e respectivas subcaracterísticas de qualidade de *software*, que constituem-se como requisitos de qualidade. As oito características de qualidade definidas na ISO/IEC 25010, são:

- (i) Adequação da funcionalidade - que engloba completude, corretude e adequação funcional;
- (ii) Eficiência de desempenho - que engloba comportamento em relação ao tempo, utilização de recursos e capacidade;
- (iii) Compatibilidade - que engloba coexistência e interoperabilidade,
- (iv) **Usabilidade** - que engloba adequação do reconhecimento, apreensibilidade, proteção contra erros do usuário, operabilidade, estética da interface e **acessibilidade**;
- (v) Confiabilidade - que engloba maturidade, tolerância a falhas, disponibilidade e recuperabilidade;
- (vi) Segurança - que engloba confidencialidade, integridade, não repúdio, responsabilização e autenticidade;

- (vii) Manutenibilidade - que engloba modularidade, reusabilidade, analisabilidade, modificabilidade e testabilidade; e
- (viii) Portabilidade - que engloba adaptabilidade, instabilidade e substitutibilidade.

Essas características de qualidade são divulgadas amplamente e constituem conceitos reconhecidamente importantes pelos desenvolvedores, mas que necessitam de detalhamento para serem adequadamente atendidas. Nesta pesquisa, focamos em usabilidade e acessibilidade, que são investigadas no âmbito das boas práticas, geralmente utilizadas por desenvolvedores, para o desenvolvimento de aplicativos web no contexto *mobile*. Foram, portanto, estudadas essas duas características de qualidade, como requisitos de qualidade de *software*, as quais têm sido alvo de inovações tecnológicas, no desenvolvimento de aplicativos web no contexto *mobile*. Assim, as próximas seções descrevem as definições e aspectos relativos a Usabilidade (Seção 2.3) e a Acessibilidade (Seção 2.4). Em especial, os estudos relativos a acessibilidade foram direcionados pela busca de quais propriedades são consideradas no escopo de interfaces de aplicativos móveis.

2.3 Usabilidade

Usabilidade é um conceito relativo a um conjunto independente de atributos de qualidade de software, como eficiência, apreensibilidade, satisfação do usuário, bem como a combinação de todos eles ao mesmo tempo (BEVAN; CURSON, 1999). Para Abran *et al.* (2003), as propriedades atribuídas à definição de usabilidade de um *software* podem variar de prioridade, conforme a perspectiva do público alvo do sistema, as quais podem ser de:

- (1) **Usuários finais**, para os quais a usabilidade do *software* é essencial, pois é um determinante de desempenho, ou seja, uma aplicação que apresenta boa usabilidade permitirá que o usuário execute a tarefa desejada de modo mais rápido e eficiente;
- (2) **Gestores**, para os quais a usabilidade é um importante ponto de decisão na escolha de um produto, pois terá uma influência direta sobre apreensibilidade do sistema escolhido, e por conseguinte, sobre a produtividade de quem utilizá-lo; e
- (3) **Desenvolvedores**, para os quais ela descreve os atributos internos do sistema, incluindo questões como a qualidade do *design* e manutenção de documentação.

Abran *et al.* (2003) ainda afirmam que esta diversidade de pontos de vista e respectivos requisitos de usabilidade levaram a diferentes concepções nos vários modelos ISO, implicando no uso de diferentes termos e rótulos para as mesmas características de usabilidade. Assim, é necessário contínuo trabalho na busca por um consenso visando integrar todos os pontos de vista em um modelo unificado que resista a constante sofisticação de novos aplicativos (SEFFAH; METZKER, 2004; ABRAN *et al.*, 2003; LARA, 2013).

A norma ISO/IEC 25010 possui duas definições de usabilidade. A definição em “qualidade em uso” para usabilidade é a mesma contida na norma ISO/IEC 9241-100 (ISO, 2010), descrevendo o critério de qualidade como:

“Usabilidade é o grau em que um produto ou sistema pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação, em um contexto específico de uso” (ISO, 2011, 4.1, tradução nossa).

Nessa definição, o contexto de uso abrange usuários, tarefas, equipamentos (*hardware*, *software* e pessoal) e ambiente no qual o produto é utilizado (ISO, 2010).

A ISO/IEC 25010 define os 3 requisitos necessários para atingir a usabilidade de um sistema em uso, como (ISO, 2011, 4.1, tradução nossa):

- Eficácia: precisão e integridade com que os usuários atingem objetivos específicos.
- Eficiência: recursos gastos (como tempo para concluir a tarefa, materiais ou o custo financeiro) em relação à precisão e completude com que os usuários atingem os objetivos.
- Satisfação: grau em que as necessidades dos usuários são satisfeitas quando um produto ou sistema é utilizado em um contexto específico de uso.

Outra definição da norma é a de “qualidade interna e externa do software”, que descreve usabilidade como:

“Atributo de qualidade de produto de software que pode ser especificado ou medido como uma característica de qualidade do produto em termos de suas subcaracterísticas”.

As seis subcaracterísticas de usabilidade, nesses termos, são definidas como a seguir (ISO, 2011, 4.2, tradução nossa):

- (i) Reconhecibilidade de adequação: grau em que os utilizadores podem reconhecer (com base em informações fornecidas pelo produto ou sistema como demonstrações, tutoriais, documentação ou, para um site, as informações na página inicial) se um produto ou sistema é adequado às suas necessidades.
- (ii) Apreensibilidade: grau em que um produto ou sistema pode ser usado por usuários específicos para atingir objetivos específicos de aprendizagem ao usar o produto ou sistema com eficácia, eficiência, liberdade e satisfação em um contexto específico de uso.
- (iii) Operabilidade: grau em que um produto ou sistema tem atributos que facilitam a operação e o controle.

- (iv) Proteção a erros do usuário: grau em que um sistema previne os usuários de cometerem erros.
- (v) Estética da interface do usuário: grau em que uma interface de usuário permite uma interação agradável e satisfatória para o usuário, com propriedades do produto ou sistema que aumentem o prazer e a satisfação do usuário, como o uso da cor e a natureza do desenho gráfico.
- (vi) **Acessibilidade**: grau em que um produto ou sistema pode ser usado por pessoas com a mais ampla gama de características e capacidades para atingir um objetivo específico em um contexto específico de uso.

Vale destacar que quanto a característica **acessibilidade**, a [ISO \(2011\)](#) define que esse requisito deve ser obedecido quando um produto ou sistema pode ser utilizado por usuários com incapacidades específicas para atingir seus objetivos específicos com eficácia, eficiência ou com a presença de propriedades do produto que suportem acessibilidade.

Normas e boas práticas podem ser utilizadas como guia de desenvolvimento e avaliação do software. Por exemplo, para realizar uma avaliação utilizando a [ISO \(2014\)](#), o avaliador define métricas com base nos requisitos de qualidade de produto que considera relevantes ao software analisado. Na avaliação são atribuídos valores (limitados a um intervalo definido e com a devida semântica associada) a cada métrica e com o resultado final é analisada a conformidade do software com relação aos requisitos de qualidade avaliados.

Por exemplo, em uma avaliação utilizando as métricas definidas com base nos requisitos de usabilidade para qualidade em produto ([ISO, 2014](#)), o avaliador poderia conferir se:

- *"O contraste das cores de plano de fundo e letras são adequados para a leitura?"*
- *"O software fornece feedback das ações realizadas ao usuário?"*

Essas métricas, quando avaliadas ainda durante o desenvolvimento, propiciam que o desenvolvedor busque implementar, por exemplo, a exibição de caixa de aviso para notificar cada ação importante solicitada pelo usuário, bem como utilizar cores contrastantes, possivelmente personalizáveis, facilitando a percepção dos elementos e legibilidade do texto. Essas soluções práticas, de implementação, são mais efetivas durante o desenvolvimento do que após o término do desenvolvimento. Portanto, as abordagens de avaliação quando se considera o conjunto de elementos para análise das interfaces do software para os usuários, segundo [Lucinda \(2010\)](#), [Barbosa e Silva \(2010\)](#), devem sempre ser aplicadas repetidamente, durante todo o ciclo de desenvolvimento.

Além das normas que fornecem orientações para desenvolvimento e avaliação de *software*, [Nielsen e Molich \(1990\)](#) propuseram recomendações de usabilidade, na forma de heurísticas (baseadas em estudos práticos sobre os problemas que mais acarretam dificuldades de uso

pelos usuários) com características desejadas em uma UI (NIELSEN, 1994a). Esse conjunto de 10 heurísticas de usabilidade, que orientam avaliações por especialistas em usabilidade, é assim constituído (NIELSEN, 1995):

1. **Visibilidade do status do sistema:** o sistema deve manter os usuários sempre informados sobre o que está acontecendo, por meio de *feedback* em tempo apropriado;
2. **Compatibilidade do sistema com o mundo real:** o sistema deve utilizar palavras, frases e conceitos familiares ao usuário em vez de termos técnicos do sistema. Devem ser seguidas convenções com o mundo real, apresentando as informações em uma ordem natural e lógica;
3. **Controle do usuário e liberdade:** algumas vezes os usuários realizam ações no sistema por engano e necessitam de uma “saída de emergência” apropriada sem precisarem passar por um diálogo extenso. A UI deve permitir ao usuário a opção de desfazer e refazer suas ações;
4. **Consistência e padrões:** o sistema deve seguir as convenções da plataforma, apresentando informações de modo consistente e padronizado. Palavras, situações ou ações diferentes não devem ser usadas para o mesmo significado;
5. **Prevenção de erros:** o sistema deve eliminar condições de propensão à ocorrência de erros, como botões que realizam ações opostas demasiadamente próximos, por exemplo, e requerer confirmação do usuário antes de concretizar uma ação selecionada. Melhor que boas mensagens de erro é um *design* cuidadoso que previne a ocorrência de problemas;
6. **Reconhecimento ao invés de relembração:** o sistema deve minimizar a carga de memória do usuário mantendo objetos, ações e opções visíveis. O usuário não deve precisar se lembrar de informações entre diálogos. As instruções de uso do sistema devem estar visíveis ou facilmente acessáveis sempre que houver necessidade;
7. **Flexibilidade e eficiência de uso:** o sistema pode fornecer mecanismos para que usuários realizem mais facilmente ações frequentes, e também aceleradores, que geralmente não são vistos por usuários novatos, podem acelerar a interação do usuário experiente de forma que atenda usuários com diferentes graus de conhecimento do sistema;
8. **Estética e design minimalista:** os diálogos do sistema não devem conter informações irrelevantes ou raramente necessárias. Toda unidade extra de informação na UI compete com unidades relevantes de informação e diminuem sua visibilidade relativa;
9. **Ajuda aos usuários para reconhecer, diagnosticar e corrigir erros:** as mensagens de erro do sistema devem estar em linguagem simples (sem códigos), indicar precisamente o problema e sugerir uma solução construtiva;

10. **Ajuda e documentação:** embora seja melhor que o sistema possa ser usado sem a necessidade de documentação, é preciso fornecê-la bem como a opção de ajuda. Essas informações devem ser fáceis de encontrar, focadas na tarefa do usuário, listar passos concretos a serem executados e não serem demasiadamente longas.

Nielsen (2012) descreve usabilidade como critério de qualidade que mede prioritariamente a facilidade de uso de um sistema, mas a partir do conjunto das suas 10 heurísticas de usabilidade, observa-se uma preocupação mais ampla sobre todos os elementos que afetam esse requisito.

Para Barbosa e Silva (2010), usabilidade foca no modo que características do usuário, como capacidades cognitivas e de interação com a interface, influenciam no uso de um sistema de interação. Essa definição está, portanto, alinhada com o conceito de “experiência do usuário” (ou *User Experience* (UX)¹) que tem sido atualmente muito utilizado por *designers* de interfaces e de interações (GEREA; HERSKOVIC, 2015). Assim, contribui com a inserção de características do usuário que possam influenciar no modo como o sistema será percebido e utilizado por ele/a.

Em conjunto, as definições de usabilidade sugerem que para um *software* apresentar boa usabilidade, precisa ser fácil de usar pelos usuários **independente de suas capacidades cognitivas, bem como suas habilidades na interação com a interface**. Destaca-se a percepção de que para um sistema ser usável, foi incluída a característica de **acessibilidade**, o que não era considerado nas normas obsoletas ISO/IEC 9126 e ISO/IEC 14598. Desse modo, a acessibilidade de *software* passou a ser considerada como parte necessária para obter a usabilidade como qualidade de um sistema.

2.4 Acessibilidade

Acessibilidade tem sido uma importante preocupação da sociedade atual. Uma das razões é de que nos últimos anos, houve um grande aumento da população idosa, de 21 milhões (10% da população brasileira em 2015) com previsão de aumento para 64 milhões em 2050 (cerca de 29% da população) (UNFPA-FUNDO, 2015). Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE) (IBGE, 2012), paralelamente ao fenômeno de envelhecimento populacional mundial, houve um aumento no acesso à Internet e uso de dispositivos móveis.

Outra razão para considerar acessibilidade como um critério de qualidade importante em aplicações móveis, é que ela nem sempre está relacionada com deficiências persistentes ou um grupo específico de usuários. Um exemplo levantado por Barbosa e Silva (2010) é o GPS (Sistema de Posicionamento Global), no qual o usuário não pode utilizar as mãos para interagir com o dispositivo enquanto dirige e necessita que o sistema forneça instruções por voz

---

¹ UX são as percepções e reações de uma pessoa que resultam do uso ou utilização prevista de um produto, sistema ou serviço (ISO, 2010)

no percurso. No exemplo, a aplicação precisou ser adaptada a limitações temporárias impostas apenas pelo contexto de uso. Assim, construir softwares visando acessibilidade pode beneficiar todos os grupos de usuários.

Devido à demanda por tecnologia acessível, diversos estudos vem sendo feitos para auxiliar nesse objetivo, conforme apresentados a seguir.

Web Accessibility Initiative (WAI)

Com o intuito de fornecer uma tecnologia para disponibilização de conteúdo em um padrão simples e poderoso por meio informações hipertexto em HTML, em 1990 Tim Berners-Lee criou a *World Wide Web* (Web) (RAGGETT *et al.*, 1999). Em 1994, com o aumento do conteúdo disponibilizado na Web, seu criador foi fundador do *World Wide Web Consortium* (W3C). O W3C foi criado com o objetivo de desenvolver especificações técnicas e orientações de alta qualidade em tecnologias web, com base nas contribuições dos seus filiados, da equipe de especialistas e da comunidade em geral (W3C BRASIL, 2011). Pouco tempo depois, observando a necessidade de páginas acessíveis para possibilitar a inclusão de diferentes grupos de usuários, alguns voluntários do W3C deram origem à *Web Accessibility Initiative* (WAI) em outubro de 1996 (DARDAILLER, 2009).

A WAI agrega uma série de diretrizes e padrões para a elaboração de *websites* com acessibilidade. No contexto móvel, o W3C (2017) informa que não foram elaboradas diretrizes específicas, pois a acessibilidade móvel está coberta nos padrões e diretrizes de acessibilidade já existentes na WAI. Apesar disso, um dos projetos atuais da iniciativa é o desenvolvimento do *Mobile Accessibility*², um guia com orientação para *designers* e desenvolvedores *mobile*, com conclusão prevista até o final de 2018 (W3C, 2017). Os três seguintes conjuntos de documentos estão disponíveis, sob responsabilidade da WAI:

- (1.) ***Authoring Tool Accessibility Guidelines (ATAG 2.0)*** - diretrizes para os desenvolvedores de ferramentas de autoria de conteúdo web, seja para *desktop* ou sistemas operacionais móveis (W3C, 2015b). Tem o objetivo de fornecer um modo para que autores produzam conteúdo mais acessível por pessoas com deficiência. Abrange *softwares* usados para criação de páginas e aplicativos web, inclusive para *mobile* (W3C, 2017).
- (2.) ***User Agent Accessibility Guidelines (UAAG 2.0)*** - criado para os desenvolvedores de agentes de usuário (*browsers*), seja para *desktop* ou sistemas operacionais de dispositivos móveis (W3C, 2015a). O intuito é melhorar a acessibilidade do agente por meio de sua própria interface, utilizando as opções que fornece para a renderização e interação com o conteúdo, e também por meio da sua capacidade de se comunicar com outras tecnologias.

² <<https://www.w3.org/TR/mobile-accessibility-mapping/>>

O documento UAAG 2.0 disponibiliza a orientação denominada “*Applying UAAG to Mobile Phones*”³, que consiste de uma lista relacionando as recomendações da UAAG ao contexto móvel. Outro documento fornecido para apoio aos desenvolvedores de aplicativos móveis é o *Mobile Accessibility Examples from UAAG 2.0*⁴, que contém exemplos de como a adoção das diretrizes nos navegadores pode beneficiar usuários com deficiência que acessam a Web em dispositivos móveis (W3C, 2017). Se o propósito da aplicação móvel é renderizar, recuperar e facilitar a interação do usuário web com conteúdo de sua escolha, as diretrizes UAAG 2.0 se aplicam a ela, diferente do que ocorre em aplicações que exibem apenas um conjunto restrito de conteúdo especificado pelo desenvolvedor.

(3.) Web Content Accessibility Guidelines (WCAG 2.0) - é um conjunto de diretrizes desenvolvido por meio do processo da W3C, ou seja, em colaboração com indivíduos e organizações em todo o mundo. Seu objetivo é oferecer um único padrão comum para a acessibilidade dos conteúdos web que atenda às necessidades de indivíduos, organizações e governos em nível mundial (W3C, 2008a).

Os documentos da WCAG visam esclarecer como tornar o conteúdo web acessível para pessoas com deficiência, incluindo: informações em forma de texto, imagens, e sons e; código ou marcação que define estrutura, apresentação das informações, etc.

A WCAG 2.0 (lançada em dezembro de 2008) já contava com uma tradução em língua portuguesa, cujas descrições das *guidelines* relacionadas a cada um dos princípios podem ser conferidas no [Quadro 1](#). Essas diretrizes (do inglês, *guidelines*) foram fundamentais para o desenvolvimento da abordagem proposta neste mestrado (vide [Capítulo 4](#)).

Desde sua criação (maio de 1999) já foram disponibilizadas três versões da WCAG, e a última delas, mais recente, é a **WCAG 2.1**, uma norma técnica referenciável, lançada em 5 junho 2018. Possui diretrizes organizadas em princípios, e para cada diretriz, existem critérios de sucesso testáveis, divididos em três níveis de conformidade: A (nível mínimo de conformidade), AA e AAA (nível mais elevado de conformidade). Um aspecto relevante desta mais recente versão é destacada no próprio documento, que menciona que WCAG 2.1 teve como meta melhorar as orientações de acessibilidade para 3 principais grupos: usuários com deficiências de aprendizado e motora, usuários com baixa visão e usuários com deficiências que utilizam dispositivos móveis. Essa inclusão de orientações torna mais evidente que a proposta de um *checklist* voltado para aplicativos móveis, objetivo deste mestrado, é um tópico contemporâneo e necessário para o desenvolvimento de softwares da atualidade.

Em fevereiro de 2015, a WAI disponibilizou o documento *Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile*⁵, um guia que não possui novas

³ <https://www.w3.org/WAI/UA/work/wiki/Applying_UAAG_to_Mobile_Phones>

⁴ <<https://www.w3.org/TR/IMPLEMENTING-UAAG20/mobile>>

⁵ <<https://www.w3.org/TR/mobile-accessibility-mapping/#uaag-2.0-and-accessible-mobile-browsers>>

Quadro 1 – Diretrizes WCAG 2.0 resumidas

Princípio	Descrição das Diretrizes
1. Perceptível	1.1. Fornecer alternativas em texto para todo o conteúdo não textual de modo a que o mesmo possa ser apresentado de outras formas, de acordo com as necessidades dos utilizadores, como por exemplo: caracteres ampliados, <i>Braille</i> , fala, símbolos ou uma linguagem mais simples.
	1.2. Fornecer alternativas para multimídia dinâmica ou temporal.
	1.3. Criar conteúdo que possa ser apresentado de diferentes formas (por ex., um esquema de página mais simples) sem perder informação ou estrutura.
	1.4. Facilitar aos utilizadores a audição e a visão dos conteúdos nomeadamente através da separação do primeiro plano do plano de fundo.
2. Operável	2.1. Fazer com que toda a funcionalidade fique disponível a partir do teclado.
	2.2. Proporcionar aos utilizadores tempo suficiente para lerem e utilizarem o conteúdo.
	2.3. Não criar conteúdo de uma forma que se sabe que pode causar convulsões.
	2.4. Fornecer formas de ajudar os utilizadores a navegar, localizar conteúdos e determinar o local onde estão.
3. Compreensível	3.1. Tornar o conteúdo textual legível e compreensível.
	3.2. Fazer com que as páginas web apareçam e funcionem de forma previsível.
	3.3. Ajudar os utilizadores a evitar e a corrigir os erros.
4. Robusto	4.1. Maximizar a compatibilidade com os agentes de utilizador atuais e futuros, incluindo as tecnologias de apoio.

Fonte: W3C (2014).

recomendações, mas sim descreve como as recomendações existentes do WCAG 2.0 podem ser utilizadas no conteúdo e aplicações web móveis, aplicações nativas de apresentação de conteúdo web e também enfatiza a relevância do UAAG 2.0 no contexto móvel.

Mobile Web Initiative (MWI)

Após a criação da WAI e com a popularização de dispositivos móveis com acesso à Internet, aumentaram os problemas relacionados à interoperabilidade e usabilidade. Em meados de 2009, a W3C iniciou a *Mobile Web Initiative* (MWI) (W3C, 2009), para abordar tais problemas, por meio de um esforço dos principais intervenientes na cadeia de produção móvel, incluindo provedores de conteúdo, fabricantes de aparelhos, fornecedores de navegadores e operadores móveis (W3C, 2009).

Em 2005, a W3C criou o *Mobile Web Best Practices Working Group* (MWBPWG) (W3C, 2005) a fim de agrupar as melhores práticas, técnicas e materiais de apoio ao desenvolvimento de *websites* para proporcionar uma experiência de uso apropriada em dispositivos móveis. Para orientar os participantes do grupo sobre como as melhores práticas podem ser implementadas, foi criado o *website* “*Shared Techniques wiki for the W3C Mobile Web Initiative Best Practices*”⁶, um

⁶ <<https://www.w3.org/2005/MWI/BPWG/techs/TechniquesIntro.html>>

espaço online editável e publicamente acessível, onde membros compartilhavam técnicas para desenvolvimento com boas práticas de programação *mobile*. Uma técnica foi caracterizada pelo conjunto de medidas práticas para implementar uma **melhor prática** recomendada, podendo incluir até mesmo a configuração do servidor web (W3C, 2010b). As técnicas são organizadas em categorias, referenciando a(s) melhor(es) prática(s) a que se referem; por exemplo, *CategoryCss* que inclui as técnicas para implementar as melhores práticas referentes a CSS⁷. São disponibilizadas 64 melhores práticas, com suas respectivas técnicas existentes, como mostra a Figura 6.

Figura 6 – Captura de tela mostrando técnica de melhor prática do MWBPWG

Fonte: Elaborada pelo autor.

Outra iniciativa do grupo foi a criação da ferramenta *W3C mobileOK Checker*, um verificador online que a partir da inserção da URL permitia testar a compatibilidade do documento web em dispositivos móveis com base nas diretrizes da MWBP 1.0.

O grupo ficou ativo no período de 2005 a 2010, mas as edições da *wiki* não foram atualizadas após seu encerramento, embora as recomendações com as técnicas já existentes, atualmente, ainda possam ser acessadas. A ferramenta *W3C mobileOK Checker* permaneceu ativa depois do encerramento do MWBPWG⁸, mas seu funcionamento foi suspenso em 2016.

Atualmente, a MWI está centrada no desenvolvimento de tecnologias⁹ que aumentam os recursos de aplicativos web quando aplicados especificamente ao contexto móvel. Existem

⁷ *Cascading Style Sheets (CSS)* - é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML.

⁸ <https://www.w3.org/2008/06/MWBP-WG-charter.html>

⁹ <https://www.w3.org/2018/04/web-roadmaps/mobile/>

diversos projetos iniciados, para auxiliar avaliadores, desenvolvedores e por conseguinte, usuários. Dentre eles, destacam-se dois, conforme descritos a seguir.

1. MWBP - Mobile Web Best Practices

O **Mobile Web Best Practices** (MWBP) é um documento que especifica as melhores práticas para entrega de **conteúdo** web em dispositivos móveis; foi parcialmente derivado do WCAG e teve sua primeira versão disponibilizada em julho de 2008 (W3C, 2008b). O foco de suas recomendações é apenas entrega de conteúdo, não abordando sobre processos de criação desse conteúdo ou sobre dispositivos que o exibem. O documento possui 60 recomendações, as quais são descritas conforme exemplificado a seguir (W3C, 2008c).

Título - a área funcional abordada pela(s) declaração(ões) da(s) melhor(es) prática(s).

Ex: "Mecanismos de navegação".

Declarações - uma ou mais declarações de melhor prática, identificada(s) por "[IDENTIFICADOR DA MELHOR PRÁTICA] Descrição sucinta da melhor prática".

Ex: "[NAVEGAÇÃO] Forneça mecanismos consistentes de navegação".

Significado - uma introdução acerca do problema específico seguido do significado da(s) recomendação(ões) e explicações de como o uso pode melhorar a interação/funcionamento do software.

Como fazer - uma discussão de técnicas e sugestões de implementação da(s) declaração(ões).

O que testar - define se a recomendação é passível de ser testada de modo automatizado (testável por máquinas) ou se o teste requer interação humana. Para cada caso existente estão os aspectos que um validador externo poderia examinar para avaliar a conformidade com a(s) melhor(es) prática(s).

Ex: "Teste automatizado - testar a presença do atributo *accesskey*"

"Teste humano - verificar a presença do atributo *accesskey* em *links* como o de atalho para a página inicial".

Referências - nem sempre são inseridas. São colocadas apenas quando a(s) melhor(es) prática(s) está(ão) relacionada(s) a pontos específicos da WCAG 1.0.

As recomendações MWBP são direcionadas para melhoria na entrega de conteúdo de aplicativos web móveis e não para o agente de usuário utilizado para apresentação do conteúdo dessa aplicação. Em outras palavras, as melhores práticas não consideram o navegador (aplicação nativa) utilizado para acessar o *website* (aplicativo web móvel) em dispositivo móvel.

2. MWABP - Mobile Web Application Best Practices

O **Mobile Web Application Best Practices** (MWABP) consiste de um conjunto de recomendações pela W3C, disponibilizadas em dezembro de 2010, cujo objetivo é auxiliar o desenvolvimento de aplicações web ricas e dinâmicas para dispositivos móveis (W3C, 2010). O

documento reúne as práticas de engenharia mais relevantes, destacando as que direcionam a melhor UX e adverte em relação àquelas que possam ser prejudiciais (W3C, 2010). No documento são apresentadas 48 recomendações de boas práticas, classificadas nas categorias:

- (i) Dados de aplicação,
- (ii) Segurança e privacidade,
- (iii) Informação e controle do usuário,
- (iv) Uso conservador de recursos,
- (v) *User Experience*, e
- (vi) Manipulação da variação no contexto de entrega.

Essas são categorias que remetem aos requisitos de qualidade da norma ISO/IEC 25010 mencionada nas Seção 2.2 e Seção 2.3, porém se referem apenas a aplicativos web. Cada recomendação está definida, descrita conforme a seguir.

Título - nome da Melhor Prática (recomendação) quase sempre auto-explicativo. Ex: “3.6.5¹⁰ Oferecer aos usuários escolha de interfaces” (W3C, 2010, tradução nossa).

Significado - uma explicação mais detalhada, porém sucinta da Melhor Prática. Ex: no caso da recomendação 3.6.5:

“A detecção de característica de dispositivo é imperfeita, e nem sempre leva em conta os diferentes casos de uso de uma aplicação. Portanto a detecção automática não é suficiente como o único mecanismo para determinar qual a versão apropriada”(W3C, 2010, tradução nossa).

Como fazer - algumas sugestões de como implementar e características que a aplicação final deve possuir nesse quesito. Por vezes, possui links para documentos mais completos sobre o assunto em pauta. Ex: no caso da recomendação 3.6.5:

“– Onde existirem várias versões de uma aplicação, sempre oferecer ao usuário a oportunidade para selecionar a opção de sua preferência.”

“– Tente sempre fornecer a interface do usuário mais adequada ao padrão, na primeira utilização.”

“– Sempre lembrar a preferência do usuário para futuras visitas em um cookie ou armazenamento de dados local” (W3C, 2010, tradução nossa).

A W3C alerta que esse documento foi elaborado com base nas tecnologias da época, e conforme sua evolução algumas seções podem se tornar obsoletas.

Em resumo, dentre as diferenças entre MWBP e MWABP, tem-se que a primeira visa auxiliar no desenvolvimento de conteúdo web para dispositivos móveis, enquanto a segunda se preocupa em auxiliar no desenvolvimento de aplicativos web móveis e agentes de usuário

---

¹⁰ No documento MWABP de W3C (2010), seção *Best Practices*: <https://www.w3.org/TR/mwabp/#bp>

(*browsers*). Ambos documentos são de relevância para o desenvolvimento deste trabalho, pois, ao apresentarem pontos em comum com a WCAG, serviram de referência para o processo de relacionar e mapear *guidelines* de acessibilidade no desenvolvimento de *websites* (WCAG) com as recomendações que se referem ao desenvolvimento web móvel (MWBP e MWABP). Ou seja, o estudo desses documentos permitiu compreender quais aspectos e detalhes de aplicativos (web) móveis devem ser analisados quando o interesse é avaliar seu grau de acessibilidade e usabilidade.

Legislação vigente sobre acessibilidade

Com a crescente rapidez de compartilhamento de conteúdos e a maior utilização de meios eletrônicos portáteis para esse fim, foi observada a necessidade da criação de leis para padronizar e garantir acesso facilitado a todos os grupos de usuários.

Existem leis referentes à produção de conteúdo web acessível, sendo Estados Unidos, Canadá e Austrália os primeiros países a definir parâmetros de acessibilidade em 1997. Entrou em vigor nos Estados Unidos a *Section 508*¹¹, uma lei cujo objetivo era eliminar barreiras nas Tecnologia de Informação e Comunicação (TIC) de órgãos federais, determinando que fossem acessíveis a pessoas com deficiência. A *Section 508* possui uma série de documentos que auxiliam no desenvolvimento de *sites* em conformidade com as diretrizes da WCAG 2.0 em nível de conformidade AA, visando superar as mais frequentes barreiras encontradas por usuários com deficiência no ambiente eletrônico (OGP, 2015). Entre esses documentos estão também Boas Práticas¹² para desenvolvimento e procedimentos para testes de conformidade de *softwares*¹³. Os padrões para desenvolvimento também incluem computadores portáteis, porém o processo de testes pode ser realizado apenas em sistemas operacionais, navegadores e plataformas específicos, não incluindo *tablets* e aparelhos móveis (DHS, 2015).

Em 1999, mesmo ano do lançamento da WCAG 1.0, Portugal foi o primeiro país europeu a regulamentar regras de acessibilidade web com a *Resolução do Conselho de Ministros nº 97/99 - Acessibilidade dos sítios da Administração Pública na Internet pelos Cidadãos com Necessidades Especiais*¹⁴, que apresenta apenas a resolução de modo sucinto, sem diretrizes ou métodos de atingir a acessibilidade na Web. Em seguida, o Conselho Europeu aprovou o plano de ação “e-Europe 2002”, estendendo a iniciativa portuguesa a todos os 15 países pertencentes à União Europeia (TANGARIFE; MONT’ALVÃO, 2005).

No Brasil, a legislação de acessibilidade na Web foi regulamentada em 2004 no Decreto nº 5.296, de 2 de dezembro de 2004, que tornou obrigatória a acessibilidade em portais e *sites* governamentais e quaisquer documentos eletrônicos da administração pública, com o objetivo de

¹¹ <<https://www.section508.gov/>>

¹² <<https://www.section508.gov/content/learn/best-practice-library>>

¹³ <https://www.dhs.gov/sites/default/files/publications/DHS_Section_508_Compliance_Test_Process_for_Applications_0.pdf>

¹⁴ <http://www.acessibilidade.gov.pt/acesso/res97_99.htm>

torná-los acessíveis e permitir que pessoas com deficiência tivessem pleno acesso às informações (LARA, 2013; BRASIL, 2004). Esse Decreto nº 5.296/2004 regulamentava as leis brasileiras nº 10.048/2000 e nº 10.098/2000, que não haviam sido criadas pensando em acessibilidade web mas que serviram de base para promovê-la.

A partir da regulamentação brasileira sobre a acessibilidade na web, no mesmo ano foi elaborado o Modelo de Acessibilidade do Governo Eletrônico (eMAG). O eMAG é um conjunto de recomendações criado para padronizar a implementação da acessibilidade de *sites* e portais do Governo Brasileiro, coerente com as necessidades brasileiras e em conformidade com os padrões internacionais (E-MAG, 2014). As bases para proposição do eMAG foram a *Section 508* e as regras e pontos de verificação presentes na WCAG 1.0, versão mais recente na época. A primeira versão (1.4) do eMAG foi disponibilizada para consulta pública em 2005, e a segunda (2.0) em dezembro do mesmo ano. Em 2007, por meio da Portaria nº 3, de 7 de maio, a observância do eMAG se tornou obrigatória para sítios e portais do governo brasileiro (E-MAG, 2014).

Em 2011, foi desenvolvido o eMAG 3.0, que passou a adotar a WCAG 2.0 como principal referência, desconsiderando os níveis de prioridade A, AA e AAA, além de não permitir exceções com relação ao cumprimento das diretrizes. Em abril de 2014 foi disponibilizado o eMAG 3.1, versão mais atual que conta com 45 recomendações agrupadas em seções e associadas aos Critérios de Sucesso da WCAG 2.0. Essa versão apresenta melhorias no conteúdo do texto para torná-lo mais compreensível e novos exemplos para algumas recomendações (E-MAG, 2014).

A legislação brasileira avançou desde a regulamentação de acessibilidade. Em 6 de julho de 2015, com a Lei nº 13.146/2015 foi criado o Estatuto da Pessoa com Deficiência (EPD), que institui a lei brasileira de inclusão da pessoa com deficiência. O EPD visa assegurar e promover, em condições de igualdade, o exercício dos direitos e liberdades fundamentais à pessoa com deficiência, e traz, para fins de aplicação da lei, a seguinte definição de acessibilidade:

“[...] possibilidade e condição de alcance para utilização, com segurança e autonomia, de espaços, mobiliários, equipamentos urbanos, edificações, transportes, **informação e comunicação, inclusive seus sistemas e tecnologias**, bem como de outros serviços e instalações abertos ao público, de uso público ou privados de uso coletivo, tanto na zona urbana como na rural, por pessoa com deficiência ou com mobilidade reduzida”.
Artigo 3º, Capítulo I - DISPOSIÇÕES GERAIS (BRASIL, 2015)

A lei mantém a obrigatoriedade da implementação de acessibilidade nos *sites* governamentais do Decreto nº 5.296/2004, estendendo sua aplicação também a *sites* mantidos por empresas com sede ou representação comercial no País (BRASIL, 2015). O EPD define barreiras, entre outros, como qualquer obstáculo que limite ou impeça a participação social da pessoa ao acesso à informação. Nesse contexto, barreiras tecnológicas são quaisquer obstáculos que dificultam ou impedem o acesso da pessoa com deficiência às tecnologias (BRASIL, 2015). Para diminuição desses entraves aplicáveis a tecnologia móvel, a Lei define que:

“Cabe ao poder público incentivar a oferta de aparelhos de telefonia fixa e móvel celular com acessibilidade que, entre outras tecnologias assistivas, possuam possibilidade de indicação e de ampliação sonoras de todas as operações e funções disponíveis.”

Artigo 66º, Capítulo II - DO ACESSO À INFORMAÇÃO E À COMUNICAÇÃO (BRASIL, 2015)

Na próxima seção, são descritas as principais maneiras de avaliação de UI, em termos de acessibilidade e usabilidade, encontradas na literatura atual.

2.5 Avaliação de usabilidade e acessibilidade de UI

Para garantir a qualidade de um software interativo, é fundamental que sejam realizadas avaliações durante o seu desenvolvimento, ressaltando que é mais difícil e custoso corrigir um software depois de todos seus componentes terem sido implementados. Essas avaliações devem considerar critérios de qualidade em uso, tais como acessibilidade e usabilidade, que analisem não somente a visão de quem o desenvolve, mas também a do projeto e a do usuário final, segundo Barbosa e Silva (2010), e conforme observamos nas seções anteriores.

Um recurso útil para avaliações de interface durante o processo de desenvolvimento é a realização de **testes automáticos** (BLOMKVIST, 2005). Esses testes são realizados por um avaliador utilizando uma ferramenta de avaliação automática, que pode ser na forma de um programa para testar uma página web, por exemplo, quanto ao cumprimento de um conjunto definido de critérios ou características de qualidade, e gera um relatório dos problemas (falhas ou erros, por descumprimento aos critérios) encontrados (BARBOSA; SILVA, 2010; E-MAG, 2014). Segundo Ivory e Hearst (2001), o uso de testes automáticos auxilia no processo de avaliação, tornando-o menos demorado, mais eficiente e com custo reduzido.

A W3C disponibiliza diversas ferramentas¹⁵ de avaliação automática de acessibilidade na web com base nos padrões da WAI. Alguns exemplos de ferramentas existentes para avaliação automatizada de acessibilidade são a Avaliador e Simulador de Acessibilidade em Sítios (ASES)¹⁶, que avalia de acordo com diretrizes do eMAG, sem especificação de versão (ASES, 2016); *Cryptzone Cynthia Says*¹⁷, que avalia com base nas diretrizes da *Section 508* e as diretrizes WCAG 2.0 (CRYPTZONE; ICDRI; DISABILITY, 2003); *AChecker*¹⁸, que busca por problemas de acessibilidade, seguindo as recomendações da *Section 508* e do WCAG 1.0 e 2.0 e; daSilva¹⁹, que utiliza recomendações do WCAG 1.0 e 2.0 e eMAG 2.0. Todas essas ferramentas são disponibilizadas *online*, e avaliam a conformidade do aplicativo web, incluindo

¹⁵ <<https://www.w3.org/WAI/ER/tools/>>

¹⁶ <<http://asesweb.governoeletronico.gov.br/ases/>>

¹⁷ <<http://www.cynthiasays.com/>>

¹⁸ <<http://achecker.ca/>>

¹⁹ <<http://dasilva.org.br>>

web móvel, com base em padrões utilizados, basta inserir a URL da página a ser testada, arquivo ou código-fonte da página a ser avaliada.

Entretanto, apesar dessas ferramentas se enquadrarem como validadores automáticos de acessibilidade em relação a um conjunto de critérios reconhecidos e importantes para a comunidade, quando utilizados como único método de avaliação, não conseguem determinar se o que foi avaliado está ou não acessível sob a perspectiva de pessoas interagindo efetivamente com o aplicativo. Observa-se que nem todos os problemas de acessibilidade podem ser detectados mecanicamente (E-MAG, 2014).

Vigo, Brown e Conway (2013) investigaram 6 ferramentas automáticas para avaliação de acessibilidade, analisando sua cobertura, integridade e exatidão com relação à conformidade com as WCAG 2.0. Como resultado, o trabalho atingiu resultados quantitativos que comprovaram que a avaliação de acessibilidade apenas por meio de ferramentas automáticas não é suficiente, somente 50% dos critérios de sucesso das WCAG 2.0 são cobertos por essas ferramentas. Portanto, fica evidente a necessidade da verificação humana nas avaliações web.

Do mesmo modo que ocorre com acessibilidade, existem ferramentas para apoio à avaliação automática de usabilidade na web, porém a maioria delas não faz análise ergonômica em relação à percepção, cognição e capacidade motora do usuário, ficando limitada à validação da sintaxe do HTML (AMARAL, 2014). Assim, testes automáticos são, sem dúvida, adequados e eficientes para manter a qualidade do código, entretanto, não são apropriados para avaliar a qualidade de uso (BLOMKVIST, 2005; BARBOSA; SILVA, 2010).

Existem vários métodos para avaliar a “qualidade de uso”, propostos na literatura. Segundo (BARBOSA; SILVA, 2010), os métodos para avaliação de UI podem ser classificados em:

- **investigação**, que envolve o uso de questionários, a realização de entrevistas, grupos de foco e estudos de campo;
- **observação de uso**, no qual o avaliador observa os usuários em seu contexto habitual ou em laboratório, e coleta dados sobre situações em que os usuários realizam suas atividades, com ou sem apoio de sistemas interativos e;
- **inspeção**, no qual avaliadores buscam identificar problemas que os usuários podem vir a ter quando interagirem com o sistema.

Métodos de avaliação por inspeção costumam ser mais rápidos e de custo de execução inferior ao dos demais métodos, pois nem sempre necessitam recrutar usuários para sessões de coleta de opiniões ou de observação. Na avaliação por inspeção, os avaliadores examinam discrepâncias entre a interface e diretrizes, na tentativa de prever as experiências de uso que os usuários poderão vir a vivenciar no uso da interface avaliada (PAZ; POW-SANG, 2016).

Em geral, a inspeção pode ainda abordar questões relacionadas à gravidade dos problemas encontrados. Como exemplo, tem-se o método de inspeção denominado Avaliação Heurística (Seção 2.3) de Nielsen e Molich (1990), no qual vários avaliadores (especialistas em usabilidade) interagem com a interface e julgam sua adequação comparando-a com um conjunto de recomendações guia para a avaliação (heurísticas de usabilidade) e podem tecer comentários e opiniões sobre o que está bom ou ruim no sistema (WINCKLER; PIMENTA, 2002; NIELSEN, 1994b). Outro exemplo é o método de inspeção por recomendações ergonômicas, seja, por meio de *checklists* ou com base em diretrizes definidas, consiste em ter um ou mais avaliadores investigando a interface, e os problemas identificados são associados a uma ou mais recomendações que foram violadas (WINCKLER; PIMENTA, 2002).

Apesar da Avaliação Heurística ser muito adotada por ser de rápida utilização, ela direciona apenas a problemas relacionados às heurísticas e resulta em uma inspeção menos aprofundada se comparada à avaliação utilizando recomendações ergonômicas (LAZAR; FENG; HOCHHEISER, 2010). Trata-se de um método simples, contudo, Winckler e Pimenta (2002) ressaltam a existência de outros conjuntos de recomendações sugeridos, por Nielsen e outros autores, em substituição às 10 heurísticas clássicas, o que pode acarretar níveis de maior complexidade e alguma demora para sua adequada utilização pelos especialistas (DIAS; FORTES; MASIERO, 2014).

A elaboração de recomendações ergonômicas é resultado de pesquisas nas áreas de ciência cognitiva, psicologia e ergonomia. Podem ser consideradas recomendações de “bom senso”, compreendendo um conhecimento prático acumulado por meio do desenvolvimento de vários projetos. De modo geral, não existe um padrão/modelo para descrição dessas recomendações, que podem exigir interpretação por parte do avaliador para serem aplicáveis diretamente ao projeto (WINCKLER; PIMENTA, 2002). Na maioria das vezes, o conjunto de diretrizes utilizado nesse método de inspeção é bastante extenso, o que exige experiência dos avaliadores. Desse modo, o processo de avaliação com base em inspeção por recomendações ergonômicas pode ser exaustivo, caso o avaliador não tenha conhecimentos prévios sobre tais recomendações (ROCHA; BARANAUSKAS, 2003; LAZAR; FENG; HOCHHEISER, 2010; WINCKLER; PIMENTA, 2002).

Uma alternativa a esse inconveniente é utilizar um instrumento mais objetivo para pautar a inspeção, ou seja, *checklists*. *Checklists* são um conjunto mínimo de recomendações, pré-definidas, diretamente aplicáveis e que possuam interpretação facilitada (LAZAR; FENG; HOCHHEISER, 2010). Os *checklists* concentram-se em aspectos importantes da interface e que, possivelmente, apresentam os problemas mais graves de usabilidade (WINCKLER; PIMENTA, 2002). Pela facilidade de aplicação e análise pontual de recomendações específicas, também são recomendadas por e-MAG (2014) para validação manual de acessibilidade.

A inspeção por meio de *checklists* possibilita a realização de avaliações mais rápidas em diferentes contextos, bastando para tanto selecionar as recomendações ergonômicas adequadas.

Além disso, tem custo relativamente baixo e podem ser usadas com o duplo propósito de auxiliar durante o desenvolvimento ou guiar a avaliação, no decorrer do processo de um software (WINCKLER; PIMENTA, 2002).

2.6 Considerações Finais

Neste capítulo, foram introduzidos os conceitos essenciais para o desenvolvimento deste mestrado. As definições de Qualidade de Software, Acessibilidade e Usabilidade foram apresentadas. As principais características que aplicativos interativos devem apresentar, as quais podem ser consideradas como requisitos não funcionais, foram descritas. Esses conceitos foram estudados de modo a fundamentar o desenvolvimento realizado ao longo deste projeto de mestrado.

Os métodos para avaliação de UIs (Seção 2.5), estudados da literatura, foram estudados com o intuito de prover o referencial para a abordagem de verificação desenvolvida, que foi decidida ser na forma de um *checklist*, descrito no Capítulo 4.

Existem pesquisas sobre trabalhos que focam em “boas práticas”²⁰ e que auxiliam no desenvolvimento de aplicativos móveis em geral. Assim, um estudo da literatura, sobre tais propostas e suas características, considerando-se as questões de acessibilidade e usabilidade, será apresentado no Capítulo 3.

---

²⁰ “Boas Práticas” é uma expressão derivada do inglês *best practice*. Se refere a utilização de técnicas identificadas como as melhores para realizar determinada tarefa.

DESENVOLVIMENTO MÓVEL - PADRÕES DE PROJETO

3.1 Considerações Iniciais

Em especial, nesta pesquisa são considerados *Design Patterns* (Padrões de Projeto) e outras boas práticas de desenvolvimento de *software* para dispositivos móveis. Todos esses elementos foram estudados em detalhe, visando obter uma visão geral do contexto prático que desenvolvedores se deparam quando envolvidos com os recursos inerentes aos aplicativos móveis.

Nas próximas seções, são descritos os trabalhos estudados da literatura, referentes ao diferencial do desenvolvimento de aplicativos móveis ([Seção 3.2](#)), sobre as boas práticas que têm sido utilizadas no desenvolvimento de aplicativos móveis ([Seção 3.4](#)) e com foco em *Design Patterns* ([Seção 3.3](#)). A seguir, um estudo mais detalhado sobre um conjunto de *Design Patterns* aplicados a interfaces com usuários, em aplicativos móveis, é descrito na [Seção 3.5](#).

3.2 Desenvolvimento de Aplicativos Móveis

O número de dispositivos móveis inteligentes está em crescente ascensão, sendo o maior crescimento em relação a plataformas computacionais (cerca de 20% a mais de usuários de dispositivos móveis em relação a computadores pessoais só em 2013) ([PETTEY; MEULEN, 2008](#)). Um dos fatores para essa diferença é a maior praticidade e portabilidade que os dispositivos móveis possuem, além do poder computacional semelhante ao de um computador pessoal comum, bem como serem equipados com sensores e outros dispositivos (como lanterna e câmera, por exemplo).

Esses diferenciais em relação aos *desktops* permitem o desenvolvimento de aplicativos

capazes de fornecer aos usuários um conteúdo rico, com alto grau de localização e em tempo real (OULASVIRTA; WAHLSTRÖM; ERICSSON, 2011). Essa maior utilização tem modificado drasticamente o modo como as pessoas interagem com aplicativos de *software* (DEHLINGER; DIXON, 2011). Essas novas funcionalidades geraram novos desafios para desenvolvedores de aplicativos, bem como o aparecimento de requisitos não encontrados no desenvolvimento tradicional de *software* (WASSERMAN, 2010).

De acordo com Dehlinger e Dixon (2011), a combinação do poder computacional, com diferentes sensores e rapidez na transferência de softwares *desktop* em uma versão *mobile* tem tornado aplicativos para dispositivos móveis a nova plataforma computacional de desenvolvimento TICs.

Para Wasserman (2010), a novidade de uma plataforma de computação verdadeiramente móvel oferece oportunidades e desafios únicos. Contudo, abordagens de engenharia de *software* podem não se aplicar diretamente ao contexto *mobile*, devido a alguns fatores. Um deles é o surgimento de diferentes paradigmas de interação humano-computador, como multitoque e reconhecimento de imagens, não explorados anteriormente em pesquisas, e para os quais existem diretrizes de interface do usuário que não foram oficialmente estabelecidas (OULASVIRTA; WAHLSTRÖM; ERICSSON, 2011; BALAGTAS-FERNANDEZ; FORRAI; HUSSMANN, 2009). Outro fator é a existência de diferentes tipos de Sistema Operacional (SO) (iOS, Android, Windows Mobile, etc.), fabricantes de hardware (Apple, Samsung, Google, etc.) e plataformas de interação (ou seja, *smartphone*, *tablet*). Com isso, o desenvolvedor enfrenta o problema de ter que desenvolver versões distintas de um mesmo aplicativo, que sejam compatíveis com as diferentes combinações de hardware e *software* existentes no mercado (WASSERMAN, 2010).

Desse modo, cada uma dessas opções devem ser consideradas durante o desenvolvimento, influenciando diretamente sobre os requisitos de *software* do aplicativo. Além dos fatores mencionados, a abundância de aplicativos, plataformas e suas combinações, surgem desafios de segurança, pelo fato de que muitas vezes esses *softwares* são desenvolvidos por pequenas empresas com programadores nem sempre experientes (DYE; SCARFONE, 2014).

Dehlinger e Dixon (2011) ressaltam ainda a existência de diferentes métodos de entrega como um dos principais desafios no desenvolvimento de aplicativos para dispositivos móveis. Método de entrega é o modo como o aplicativo móvel é apresentado, um fator importante, pois deve ser definido pelo desenvolvedor antes mesmo de iniciar o desenvolvimento. Assim, em termos de métodos de entrega, um aplicativo móvel pode ser:

- **nativo:** instalado no dispositivo móvel, podendo ser de fábrica ou obtido via *download* de um repositório. Nesse caso, o usuário utilizará o aplicativo para um objetivo específico. A medida que são disponibilizadas novas versões do aplicativo, o usuário precisa permitir a atualização automática das versões instaladas ou realizá-las manualmente.
- **web móvel:** acessado por meio do navegador web de um dispositivo móvel. Não há necessidade de instalar um novo *software*, do mesmo modo que não há necessidade de

usuário realizar qualquer atualização. As atualizações são realizadas via servidor.

Embora aplicativos web móveis possuam uma base de código comum em todas as plataformas - o que diminui o custo de desenvolvimento e manutenção, os usuários podem utilizar diversos tipos de navegadores móveis, trazendo outras dificuldades para oferecimento de manutenção e suporte ao aplicativo. Os aplicativos nativos geralmente apresentam melhor desempenho, em termos de rapidez e acesso aos recursos do dispositivo (câmera, GPS, etc.).

O exemplo na Figura 7 mostra as versões web móvel¹ e nativa² do aplicativo *Google Maps*. Pode-se observar que o conteúdo é apresentado de forma similar nos dois métodos de entrega (Figura 7a e Figura 7c), porém, algumas funcionalidades, como navegação por voz, estão disponíveis apenas no aplicativo nativo (Figura 7b e Figura 7d). Desse modo, o foco de pesquisa nesse trabalho se volta para aplicativos nativos, que têm apresentado maior interesse pelos usuários, e melhor interação com recursos de hardware e software do dispositivo ou suas API's (Interface de Programação de Aplicação).

Figura 7 – Capturas de tela do aplicativo *Google Maps* nos diferentes métodos de entrega

Fonte: Elaborada pelo autor.

Com tantas variáveis a serem consideradas para a criação de um mesmo aplicativo, muitas vezes a qualidade do *software* não é amplamente atingida. Geralmente, os desenvolvedores visam uma única plataforma e, em vez de considerarem a construção de uma versão para cada uma, constroem um único aplicativo desconsiderando o risco de inconsistências funcionais (WASSERMAN, 2010; FLING, 2009). Esse cenário vem mudando com a popularização dos dispositivos móveis, mas ainda existem dificuldades inerentes ao processo de desenvolvimento

¹ <www.maps.google.com.br>

² <https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=pt_BR>

de *software* em relação a qualidade. Segundo [Lucinda \(2010\)](#), procedimentos de controle são feitos apenas no código ignorando os requisitos de *design*, e nem sempre estabelecem avaliações com clareza a padrões de qualidade consolidados.

3.3 *Design Patterns* como boa prática

As diretrizes de acessibilidade e usabilidade expostas anteriormente ([Capítulo 2](#)) representam os conhecimentos prescritivos, de modo abstrato, e servem como orientação para os desenvolvedores. No entanto, diante da variedade de combinações possíveis para criar *designs* de UIs, e a velocidade das demandas para construção de novos aplicativos, é reconhecidamente útil e eficaz a técnica utilizada para representar e comunicar conhecimento na área de Interação Humano-Computador (IHC), conhecida como Padrões de Projeto (*Design Patterns*), que pode ser aplicada durante o processo de desenvolvimento do software ([BITTAR, 2013](#)).

Definido no final da década de 1970, o conceito de *Design Patterns* foi concebido inicialmente pelo arquiteto e urbanista Christopher Alexander no domínio da arquitetura ([ALEXANDER; ISHIKAWA; SILVERSTEIN, 1979](#)), tratando-se de uma abordagem de *design* que incide sobre as interações entre a forma física do ambiente contruído e como isso impacta no comportamento pessoal e social dentro dele ([BAYLE et al., 1998](#)). Nessa abordagem, foram desenvolvidos documentos, organizados com formatos similares, para tratar problemas em construções urbanas, definindo padrão como uma solução de sucesso para um problema recorrente em um determinado contexto ([ALEXANDER; ISHIKAWA; SILVERSTEIN, 1979](#)).

Esses documentos serviram de inspiração para desenvolvedores de software, que observaram a possibilidade da extensão do conceito para a área da computação, criando em 1995 o primeiro catálogo de soluções para projeto de software, o *Gang of Four*³ (GoF). Esse catálogo, desenvolvido para orientar projetos que utilizam o paradigma de orientação a objetos, divide-se em três categorias de padrões ([GAMMA et al., 1994](#)): (1) Padrões de criação, direcionados no auxílio de como criar objetos; (2) Padrões de estrutura, referentes a como compor objetos; e (3) Padrões de comportamento, que tratam sobre como os objetos devem interagir.

Além do conceito ser bastante utilizado por desenvolvedores de software, *Design Patterns* têm sido adotados no domínio de IHC, e são conhecidos como *User Interface Design Patterns* (UIDP). Como há um grande número de pesquisas sobre *Design Patterns* em IHC, o conceito é conhecido também por outras denominações, por exemplo, “padrões de interação”, “padrões de interface de usuário”, “padrões de *design web*”, entre outros ([FOLMER, 2002](#)). Um desafio recorrente em IHC está em desenvolver técnicas eficientes de documentação e compartilhamento de conhecimento em relação à interação do usuário com uma interface, e

³ *Gang of Four*, ou GoF, é a denominação do 1. catálogo de *Design Patterns* de software e possui esse nome devido a ter sido escrito por seus 4 (*four*) autores: Erich Gamma, Ralph Johnson, Richard Helm e John Vlissides.

aplicá-las em determinadas circunstâncias (FINCHER *et al.*, 2003; ANACLETO *et al.*, 2004).

Existem diferentes opiniões sobre os elementos necessários para compor a estrutura dos UIDP. Welie e Veer (2003) propuseram a inclusão de um tópico de nome “Raciocínio” na seção “Solução”, por exemplo. Essa modificação tem a finalidade de explicar como a adoção do padrão melhoraria a usabilidade do produto de software, considerando atributos como aprendizagem e satisfação do usuário. Apesar das divergências, existe uma estrutura básica definida para os UIDP, semelhante a do GoF (GAMMA *et al.*, 1994). Segundo Folmer (2002), um UIDP é uma solução geral e repetível para um problema de usabilidade frequente no *design* de uma interface, e consiste dos seguintes elementos:

- **Nome do padrão:** uma referência geralmente com uma ou duas palavras. Usada para a descrição do problema, suas soluções e consequências, em um alto nível de abstração (GAMMA *et al.*, 1994).
- **Definição do problema:** descreve em que situação deve-se aplicar o padrão, informando como o problema está relacionado com o uso do sistema e é relevante para o usuário ou qualquer outro interessado em usabilidade.
- **Solução:** descreve uma solução comprovada para o problema, com os elementos que compõem o UIDP, seus relacionamentos, suas responsabilidades e colaborações. Uma solução não descreve um projeto concreto ou uma implementação em particular porque um padrão é como um “gabarito” que pode ser aplicado em muitas situações diferentes. Desse modo, a solução apenas fornece uma descrição abstrata de um problema de projeto dando ao *designer* a liberdade de implementá-la de diversas maneiras.
- **Consequências:** os resultados e análises das vantagens e desvantagens da adoção do padrão sobre flexibilidade, portabilidade e extensibilidade de um sistema, abordando aspectos sobre linguagens e implementação (GAMMA *et al.*, 1994). Semelhante à seção proposta por Welie e Veer (2003), porém é voltada para características relevantes a padrões de projeto orientados a objeto, ou seja, durante a implementação e não a UIDP. No caso dos UIDP, deve fornecer um argumento razoável para o impacto na usabilidade do software quando o padrão é aplicado, deve descrever quais aspectos de usabilidade deveriam ter sido melhorados ou quais outros aspectos poderiam ser influenciados pelo seu uso.

Na definição de Folmer (2002) também é considerada necessária a inclusão de exemplos de uso do padrão.

- **Exemplos:** exhibe como o padrão foi adotado com sucesso em um sistema real. Como se trata de interação do usuário com a parte gráfica do software, geralmente é acompanhado por uma captura de tela e descrição sucinta.

Neste projeto de mestrado foi dada ênfase aos dois tipos de UIDP, para aplicativos web e móveis. Embora ambos sejam soluções propostas para problemas de interface, apresentam determinadas definições diferentes, mesmo entre padrões similares, pois tratam de plataformas de software distintas e os requisitos mudam de um software apresentado em *desktop* de um software a ser entregue/ disponibilizado em dispositivo móvel. A definição para UIDP web segue a estrutura proposta por [Folmer \(2002\)](#) e inclui a seção “Raciocínio” de [Welie e Veer \(2003\)](#). A seguir, tem-se um exemplo de padrão com essa estrutura ([TOXBOE; HOBSON, 2015a](#)).

Nome do Padrão: *Input Feedback*.

Definição do problema: o usuário inseriu dados no sistema e espera receber *feedback* sobre o resultado desse envio.

O desenvolvedor deve utilizar esse padrão quando deseja informar o usuário sobre o status do envio de conteúdo para o site em questão, durante e depois da operação, e também possíveis erros no envio.

Solução: quando o site possibilita ao usuário envio de conteúdo por meio de formulários podem ocorrer erros de vez em quando. O objetivo desse padrão é melhorar a experiência de uso minimizando os erros de entrada. Um paradigma chamado validação de dados é adequado para capturar erros no momento do envio por meio de regras configuradas para cada campo de entrada no formulário. Essas regras podem ser:

- Validação da presença de conteúdo: pelo menos algum conteúdo deve ser inserido.
- Validação de exclusão de conteúdo: inibindo a inserção de valores proibidos, como por exemplo, inserir “admin” como nome de usuário.
- Validação da inclusão de conteúdo: os dados devem conter determinados dados ou devem estar dentro de um determinado intervalo, como idades maiores que as possíveis de um ser humano, por exemplo.
- Validação de aceitação: por exemplo, aceitação de termos de serviço, geralmente por meio de uma caixa de seleção.
- Validação de confirmação: dois campos de entrada precisam ser combinados, como cadastro de e-mail ou senhas, por exemplo.
- Validação de formato: um e-mail, por exemplo, precisa de um sinal “@” e um número de pontos.
- Validação de comprimento: por exemplo, número mínimo de caracteres em uma senha.
- Validação de unicidade: muitos sistemas permitem o cadastro de apenas um usuário com um determinado nome de usuário.

A validação dos dados enviados pelo usuário é uma boa prática para deixá-lo saber que tudo ocorreu como planejado. Recomenda-se ainda que após a operação o usuário seja redirecionado para uma página na qual possa ver o conteúdo recém-submetido em um contexto. Caso os dados não sejam validados, uma mensagem de erro deve ser apresentada ao usuário explicando como corrigir os dados e solicitar um reenvio. Tal mensagem de erro deve explicar:

- O tipo de erro: caixa de exibição na parte superior da página (para que o usuário não precise rolar a página para descobrir que ocorreu um erro), de preferência em cor vermelha.
- Onde o erro ocorreu: pode ser feito listando e destacando os campos que causaram o erro na mensagem de erro.
- Como o erro pode ser reparado: fornecer informações sobre o que precisa ser diferente para que seja validado.

A representação visual do *feedback* de entrada deve corresponder à mensagem que se deseja passar. Se houve êxito no envio, considere informar o usuário com uso de uma caixa verde. Se a mensagem for neutra, frequentemente a cor é amarela. Se algo deu errado, vermelho é frequentemente usado.

OBS: neste UIDP, vale aqui destacar a recomendação sobre uso de cores para distinção semântica, podendo acarretar problemas quanto a acessibilidade para pessoas daltônicas.

Consequências (ou Raciocínio): ao preencher um formulário na web, o usuário está conduzindo o processo de conversão de dados mentais estruturados para uma forma escrita estruturada de outra maneira. Como os usuários são seres humanos, podem inserir os dados de diferentes maneiras à medida que tentamos converter nossos dados estruturados individualmente para uma estrutura compartilhada definida pelo sistema. Os dados inseridos em formulários da web são propensos a conter erros, para os quais o *design* deve estar preparado. O usuário deve ter conhecimento do fato de que os dados inseridos não correspondem à estrutura projetada. Por meio de *feedbacks* visualmente distintos, o usuário será informado de tais erros e como corrigí-los.

Exemplos: dois exemplos de aplicação do UIDP *Input Feedback* são exibidos na [Figura 8](#). A [Figura 8a](#) mostra o cadastro de um usuário no *site Twitter*, no qual o usuário recebe *feedback* instantaneamente sobre o texto digitado. A [Figura 8b](#) se refere à situação em que o usuário não está logado no sistema e tenta acessar uma página que necessita de autenticação. Ele então é redirecionado para a tela de *login* com uma mensagem de *feedback* positivo informando a necessidade de autenticação, e logo em seguida, um *feedback* negativo sobre a inserção de um nome de usuário incompatível.

Figura 8 – Exemplos do UIDP *Input Feedback*

(a)

(b)

Fonte: Adaptada de [Toxboe e Hobson \(2015a\)](#).

Para a definição de UIDP de aplicativos móveis, a estrutura proposta por [Nudelman \(2013\)](#) foi estudada e adotada durante este trabalho. Cada definição de UIDP possui os elementos principais dos padrões web: definição do problema, solução e consequências, porém em ordem e com nomes diferentes. Os UIDPs de aplicativos móveis utilizados neste trabalho estão resumidos na [Seção 3.5](#). Nesta dissertação, para facilitar a identificação mais rápida, adotamos a expressão **UIDP *mobile*** para referenciar, de agora em diante, os UIDPs de aplicativos móveis.

3.4 Outras Boas Práticas

Nesta seção são descritas outras duas “boas práticas” que são bastante conhecidas, atualmente, por desenvolvedores de aplicativos móveis. Essas boas práticas referem-se a informações e recomendações que definem fatores de qualidade que podem ser manifestadas nas regiões de UIs, em sua tela e seus componentes. Assim, essas boas práticas podem ser adotadas para auxiliar as decisões de como um aplicativo móvel deve ser construído, com base em fundamentos que tratam de perspectivas da interface tanto de modo amplo, como nos seus detalhes. De modo geral, essas boas práticas visam proporcionar usabilidade e acessibilidade ([TSAKNAKI; FERNAEUS, 2016](#); [LEITNER, 2012](#)); ambas são da empresa Google: *Material Design* para usabilidade, e *Google Accessibility* para acessibilidade.

3.4.1 *Material Design*

Com o passar do tempo, novas boas práticas surgem a fim de auxiliar no desenvolvimento de *software* com maior qualidade. Em julho de 2014, a empresa Google criou um novo padrão de

interface chamado *Material Design* (GOOGLE, 2014). Segundo Google (2018), esse padrão “[...] é um guia abrangente para design visual, de movimento e de interação para diversas plataformas e dispositivos”, também compatível com componentes e funcionalidades disponíveis no Android a partir da versão 5.0.

O *Material Design* fornece um conjunto de orientações e ferramentas para auxílio ao desenvolvedor, com base em boas práticas de *design* de interface do usuário (GOOGLE, 2014). Suas diretrizes padronizam toda parte gráfica do aplicativo, como ícones, cores e animações, mas também trazem recomendações quanto ao código, abrangendo a definição de padrão de projeto, dada por Gamma *et al.* (1994).

A padronização no *design* da interface no *Material Design* visa permitir melhorias na experiência do usuário (UX), uma vez que aumenta a apreensibilidade e reconhecibilidade de adequação do *software*. Isso ocorre devido a um conjunto de regras para cores e localização comum de botões e ícones que possuem funcionalidades similares, que uniformiza e padroniza suas características visuais, e assim, o usuário não necessita reaprender ou adivinhar as funcionalidades no aplicativo.

Um dos objetivos desse padrão foi criar uma linguagem que unisse os princípios de *design* com novos recursos gerados pelo avanço tecnológico de dispositivos e *softwares*, bem como permitir uma experiência unificada em diferentes plataformas (GOOGLE, 2014). Para englobar os diferentes requisitos dessas plataformas e as variações entre elas, os componentes da interface de usuário da linguagem foram organizados em *Material Components*⁴ específicos para Android, iOS, web e Flutter⁵.

A maioria dos aplicativos da Google para Android foram implementados ou já utilizam o *Material Design*, como *Gmail*, *YouTube*, *Google Drive*, *Google Maps*, todos os aplicativos da *Google Play* e até mesmo alguns *softwares* para *desktop*, como o *Google Drive Desktop*, *Sheets*, *Slides* e *Inbox*. Todos são aplicativos muito populares e com um grande número de usuários ativos, o que possibilita a evolução da linguagem por meio de um grande número de *feedbacks* de usuários dos aplicativos que a tem como base. Esse fator também pode atrair maior interesse dos desenvolvedores na utilização do *Material Design* em seus aplicativos.

Nesse padrão os conceitos **usabilidade** e **acessibilidade** são considerados interdependentes. Na Figura 9 estão as telas das recomendações do *Material Design* em dois períodos espaçados: a Figura 9a em 2015 e a Figura 9b em 2018; em ambos, acessibilidade é considerada um critério de usabilidade. Em *Material Design* tem-se que: “Um produto bem projetado é acessível aos usuários de todos os níveis, incluindo aqueles com baixa visão, cegueira, deficiências auditivas, deficiências cognitivas ou deficiências motoras. Melhorar a acessibilidade do seu

⁴ União de *design* e engenharia com uma biblioteca de componentes específica para a plataforma a fim de obter consistência entre aplicativos iOS, Android e sites.

⁵ Flutter é um *framework* para construção de aplicativos móveis em Android e iOS, disponível em: <<https://flutter.io/>>

produto melhora a usabilidade para todos que o utilizarem” (GOOGLE, 2014, tradução nossa).

Figura 9 – Aba “Usabilidade” na página oficial do *Material Design*

(a) Em 2015

(b) Em 2018

Fonte: Adaptada de Google (2014), Google (2018a).

Em maio de 2018 houve uma reorganização do *website* do *Material Design*, muitas recomendações e tópicos foram realocados e modificados, mas a relação entre acessibilidade e usabilidade não apenas permaneceu (Figura 9b) como suas recomendações e tópicos foram incrementados. Na nova versão do *website*, a estrutura foi dividida em: (1) Desenvolvedores, com documentação, recomendações de *design* e códigos dos *Material Components* para as diferentes plataformas; (2) *Design*, com os fundamentos de boas práticas e recomendações de *design* em componentes, temas e diferentes ambientes de interação do usuário com o aplicativo e; (3) Ferramentas de auxílio ao desenvolvedor, como galeria de ícones, teste manual de cores visando acessibilidade no âmbito visual (aplicativos nativos) e visualização de uma interface em diferentes plataformas (web ou móvel).

O *Material Design* oferece recomendações sobre uso de cores em elementos da interface, e destaca como as cores influenciam na usabilidade da aplicação. Nesse contexto, algumas recomendações⁶ também são referentes a acessibilidade da aplicação. Geralmente o uso de cores está associado apenas a acessibilidade visual, porém existem outros pontos a serem considerados. Como por exemplo, o modo que o uso de cores diferentes auxilia a compreender informações e o uso de cores de ênfase possibilita a percepção do status de tarefas e elementos clicáveis. Essas recomendações auxiliam na interação com a interface, em especial, por pessoas com deficiência intelectual.

Nas recomendações sobre legibilidade, o *Material Design* faz referência a WCAG 2.0, que determina o contraste de cores entre plano de fundo e texto para garantir a acessibilidade em

⁶ <<https://material.io/design/color/color-usage.html>>

sites, e recomenda a leitura das diretrizes de acessibilidade do próprio *Material Design*⁷.

Como são muitos os documentos e estão dispersos é muitas vezes requerido um trabalho custoso do desenvolvedor, por ter que buscar e conferir o atendimento a essas recomendações. Mesmo com diretrizes expressas que direcionam a valores de cor e contraste, também pode ser difícil testar opções de combinações na interface. Para auxiliar nesse contexto, por exemplo, o *Material Design* disponibiliza a ferramenta *Color Tool*⁸.

3.4.2 Google Accessibility e boas práticas Android

O objetivo do *Google Accessibility*⁹ é promover criação e adaptação de produtos, aplicativos e ferramentas com acessibilidade. A iniciativa disponibiliza uma série de informações e recursos¹⁰ para auxílio no desenvolvimento de aplicativos acessíveis, alguns com foco *mobile*, explicados a seguir.

O “Guia Android para acessibilidade” (GOOGLE, 2017) é um dos recursos disponibilizados e contém recomendações para construção de aplicativos nativos em Android utilizando APIs, com disponibilização de bibliotecas e amostras com exemplos de código. Um dos recursos desse guia é o documento “Boas práticas de acessibilidade para o desenvolvimento de aplicativos”¹¹, com recomendações básicas de acessibilidade, como rotulagem de conteúdo, tamanho do alvo de toque, contraste de cores e atributos de visualização, com exemplos na etapa de codificação. O guia também disponibiliza recomendações para testes¹². Contudo, os testes apenas encontram erros simples, como ausência de descrições de conteúdo, contraste insuficiente e ícones de toque muito pequenos. Em relação a boas práticas de *design* acessível, são utilizadas as recomendações de usabilidade/acessibilidade¹³ do *Material Design*.

Um recurso de teste disponibilizado como aplicativo nativo para Android é a *Accessibility Scanner app*¹⁴, uma ferramenta que sugere melhorias de acessibilidade, como ampliar alvos de toque, aumentar o contraste e fornecer descrições de conteúdo, em aplicativos nativos no Android.

Segundo Google (2017), acessibilidade é requisito importante de qualquer aplicativo; é preciso assegurar que os componentes serão acessíveis. Ao integrar componentes e serviços com acessibilidade, o desenvolvedor melhora a usabilidade do software desenvolvido, especialmente para usuários com deficiência.

Algumas recomendações do *Google Accessibility* fazem parte do portfólio de boas

⁷ <<https://material.io/design/usability/accessibility.html#color-contrast>>

⁸ <<https://material.io/tools/color>>

⁹ <<https://www.google.com/accessibility>>

¹⁰ <<https://www.google.com/accessibility/for-developers.html>>

¹¹ <<https://developer.android.com/guide/topics/ui/accessibility/apps>>

¹² <<https://developer.android.com/training/accessibility/testing>>

¹³ <<https://material.io/design/usability/accessibility.html>>

¹⁴ <<https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor>>

práticas para desenvolvedores Android (*Android Developer* da [Google \(2017a\)](#)). Desse modo, no mesmo ambiente, encontram-se diretrizes de desenvolvimento para outras características do aplicativo, como dispositivos e telas, no formato de artigos não padronizados como diretrizes, mas na forma de explicação mais elaborada sobre os temas, incluindo pequenos trechos de código como exemplos.

3.5 Mobile User Interface Design Patterns

Nesta seção, é descrito o estudo mais aprofundado, realizado neste mestrado, sobre os UIDPs *mobile*. Para este estudo foram escolhidos 12 UIDPs *mobile*, dos 77 UIDPs da obra de [Nudelman \(2013\)](#). Na realidade, dos 77 UIDPs, 10 deles são *Antipatterns*, ou seja, devem ser evitados. Os critérios e procedimento para escolha dos 12 UIDPs são explicados na [Seção 4.2](#).

[Nudelman \(2013\)](#) agrupa UIDPs *mobile* em 14 categorias, que discriminam local de uso e função. Dentre elas, as 6 nas quais os UIDPs selecionados estão distribuídos são: (1) **Tela Inicial**, que agrega padrões de apresentação das funções e conteúdo do aplicativo; (2) **Busca**, que agrupa os padrões relacionados a ferramentas de pesquisa, inserção e sugestão de dados por texto e voz; (3) **Ordenando e Filtrando**, reunindo os padrões relacionados tratam de organização e filtro de informações personalizáveis pelos usuários; (4) **Entrada de dados**, que agrupa padrões relativos a inserção de dados e arquivos por parte dos usuários, bem como possíveis interações do usuário durante a seleção desses arquivos; (5) **Formulários**, que inclui padrões relativos a funções de confirmação e validação de dados e (6) **Navegação**, que trata de meios de visualização de conteúdo do aplicativo.

Os estudos sobre os 12 UIDPs *mobile* utilizados nesse trabalho estão resumidos ao longo desta seção (listados na [Tabela 1](#)).

Tabela 1 – Categorias dos 12 UIDPs *mobile* estudados

Classificação	UIDP <i>mobile</i>	Sigla
Tela Inicial	1. <i>List of Links</i>	LL
Busca	2. <i>Voice Search</i>	VS
	3. <i>Auto-Complete</i>	AC
	4. <i>Auto-Suggest</i>	AS
Ordenando e filtrando	5. <i>Search from Action Bar</i>	SAB
	6. <i>Tabs</i>	TB
	7. <i>Multiple select</i>	MS
Entrada de dados	8. <i>Textbox with Input Mask</i>	TBIM
	9. <i>Inline Error Message</i>	IEM
	10. <i>Callback Validation</i>	CV
Formulários	11. <i>Cancel/OK</i>	CO
	12. <i>Carousel</i>	CRS

Fonte: Elaborada pelo autor.

A título de exemplo, a seguir está a definição na íntegra, de um dos UIDPs *mobile* utilizados, o primeiro deles, *List of Links* (NUDELMAN, 2013, tradução nossa).

Tela Inicial

List of Links (LL)

Também chamado de *Hub-and-Spoke* (documentado pela primeira vez em 2011 pela especialista em *User Experience* (UX) Jennifer Tidwell, em seu livro *Designing Interfaces* de O'Reilly), o *List of Links* é um UIDP *mobile* respeitável e popular, do mundo *mobile*, em todas as plataformas e aplicativos.

- **Como funciona** - na tela inicial, serve como um centro de exibição para uma variedade de *links* e ícones de funções primárias ou telas populares que podem ser obtidas no aplicativo.
- **Exemplo** - o aplicativo *Travelocity* (Figura 10) mostra um exemplo do UIDP LL e destaca a questão-chave desse padrão: apresenta aos usuários sobre as informações do aplicativo, de modo estático (sem informações conforme o usuário as utiliza, no decorrer do tempo).

Figura 10 – O aplicativo *Travelocity* implementa exemplo típico do UIDP *mobile List of Links*

Fonte: Nudelman (2013).

Na tela, as informações que podem ser obtidas e quais ações podem ser executadas para interagir com o aplicativo estão claras, com os ícones claros. No entanto, apesar da clareza na exibição, não há nenhuma informação que diz respeito ao usuário: todos os usuários recebem o mesmo conjunto de *links*. Como fazer essa tela estática exibir algo a mais, a respeito do usuário? Uma maneira seria usando *notification badges* em algum lugar da página ou nos ícones individuais e *links*. Por exemplo, na versão antiga do *Google Plus* (Figura 11), as notificações aparecem em destaque no topo da tela (o número 3 na caixa vermelha).

Uma variação simples também poderia ser obtida colocando notificações na forma de pequenos *badges* em ícones individuais - por exemplo, inserir (1) no ícone *Photos* se alguém compartilhou

uma nova foto e um (2) no ícone *Circles* quando duas pessoas forem adicionadas aos seus círculos, e assim por diante. Essas notificações adicionais enriqueceriam o *List of Links* a oferecer um pouco mais de detalhes da informações que apresenta, dedicadas ao usuário.

Figura 11 – *List of Links* implementado no *Google Plus* com informações de atualização

Fonte: [Nudelman \(2013\)](#).

– **Onde e quando usar** - *List of Links* é o padrão tradicional que a maioria dos desenvolvedores deveria considerar como primeiro *design* para a tela inicial, pois ele cataloga os vários aspectos da funcionalidade do aplicativo. Mesmo que você acabe não utilizando esse padrão em sua tela inicial, elaborar uma *List of Links* do seu aplicativo é um ótimo exercício de organização da Arquitetura de Informação (AI) e de catalogação de casos de uso do aplicativo, e os dos concorrentes.

– **Por que usar?** - o padrão *List of Links* é um dos mais fáceis e intuitivos para um novato navegar. Seu *design* e construção são simples (dado que você tenha um *designer* de ícones experiente), e seus resultados são instantâneos já que ele não requer uma chamada de servidor para recuperar qualquer informação que já não esteja disponível no dispositivo móvel. Mesmo que você decida utilizar o servidor para obter as *badges*, como no caso do aplicativo *Google Plus* ([Figura 11](#)), essas atualizações tipicamente são compostas de dígitos únicos que requerem um tempo baixíssimo de *download*.

– **Outros usos** - uma variação popular desse padrão básico é o *Grouped List of Links*, exibido no aplicativo *Southwest Airlines* ([Figura 12](#)). Ótima variação para aplicativos que possuem muitos *links*, porque permite um agrupamento lógico que ajuda a reduzir a carga cognitiva do usuário.

– **Aplicação em Pet Shop** - o *List of Links* possibilita que *designers* obtenham sucesso no uso de uma grande variedade de AIs. Por exemplo, a [Figura 13](#) apresenta dois AIs: a AI na esquerda permite a exploração por tipo de objeto (cachorros, gatos, pássaros, peixes, répteis e outros), enquanto que a AI na direita é mais centrada no que você pode fazer (Encontrar animais, Cuidados e Alimentação, Notícias Locais e Seu Perfil). Obviamente a AI da esquerda é indicada para aplicativos de *e-commerce*, enquanto a outra é mais adequada para aplicativos de rede social. A escolha de qual AI usar depende do propósito do aplicativo.

Figura 12 – O aplicativo Southwest Airlines mostra uma variação agrupada do padrão *List of Links*

Fonte: [Nudelman \(2013\)](#).

O mecanismo de atualização por *notification badge* mencionado anteriormente também está incluso de forma que alguns *links* (por exemplo, Notícias Locais no *design* da direita) exibam novas atualizações que aconteceram desde que essa área em particular do aplicativo foi visitada pela última vez.

Figura 13 – O padrão *List of Links* suporta múltiplas AIs

Fonte: [Nudelman \(2013\)](#).

- **Aplicativos de tablet** - esse padrão se mostra inadequado para o espaço de *display* disponível ao usuário no *tablet*. Se você usar o *List of Links* mesmo assim, considere inserir uma exibição dividida com outro painel contendo um dos outros padrões apresentados.
- **Cuidados** - nenhum.
- **Padrões relacionados** - 6.4 Padrão: *Browse*.

Os demais 11 UIDPs *mobile* estudados são descritos a seguir, de modo mais resumido, enfatizando as características que orientaram as análises para compor o *checklist Acc_Mobile-Check* desenvolvido, conforme apresentado no [Capítulo 4](#). Algumas características dos UIDPs que sugerem atenção quanto à usabilidade / acessibilidade estão sublinhadas.

Vale destacar que, nos estudos e análises realizados, os UIDPs *Auto-Complete* e *Auto-Suggest* foram pesquisados separadamente neste trabalho, mas, seguindo [Nudelman \(2013\)](#), eles são descritos conjuntamente no item 3.

Busca

2. Voice Search (VS)

Em *Voice Search* “uma consulta de áudio inserida por meio de um microfone on-board é usada como entrada para pesquisa, em vez de uma consulta por palavras-chave digitadas (...)” ([NUDELMAN, 2013](#), tradução nossa).

Na descrição de [Nudelman \(2013\)](#), o UIDP VS geralmente aparece como um ícone clicável de microfone, que ao ser tocado faz com que o dispositivo entre no modo de escuta. Assim, não é possível uso de pontuação (como vírgulas por ex.) nem acionamento manual para submissão ao término da “inserção” de informação. No caso, o dispositivo interpreta uma pausa no fluxo de áudio como o final da inserção para consulta, transcrevendo o áudio capturado em palavras chave e retornando o resultado da pesquisa. Um exemplo desse padrão pode ser observado na [Figura 14](#).

– **Características analisadas de VS:** como a busca é uma funcionalidade fundamental, devido ao tamanho reduzido da tela e proximidade das teclas, escrever no dispositivo móvel torna a inserção propensa a erros. Esse cenário agrava-se quando o usuário possui deficiência motora ou visual. Assim, a entrada por áudio é uma ótima alternativa à digitação de texto. O uso do UIDP *Voice Search* está em ascensão em países desenvolvidos. Segundo [GSMA \(2017\)](#), mais de 20% das buscas do Google nos EUA, em 2017, foram realizadas por voz, e é esperado um número ainda maior em 2018.

3. Auto-Complete (AC) and Auto-Suggest (AS)

No padrão *Auto-Suggest* “(...) quando o usuário coloca um ou mais caracteres no campo de pesquisa, o sistema exibe uma série de sugestões adicionais contendo uma ou mais combinações correspondentes às palavras-chave inseridas. Nesse ponto, o usuário pode selecionar uma das sugestões ou continuar digitando” ([NUDELMAN, 2013](#), tradução nossa).

O padrão *Auto-Complete* “(...) utiliza parte da busca digitada pelo usuário como uma semente para fornecer sugestões (para que as sugestões incluam a palavra-chave ou o fragmento original). Nem sempre funciona perfeitamente em um dispositivo móvel

Figura 14 – Exemplo de implementação do UIDP Voice Search

Fonte: Nudelman (2013).

porque muitas vezes um pequeno trecho com erros ortográficos devido a fat-finger¹⁵” (NUDELMAN, 2013, tradução nossa).

O uso do UIDP *Auto-Complete* é possível quando utiliza-se de sugestões dadas pelo sistema, que completa automaticamente o texto digitado pelo usuário. Assim, esse padrão fornece menos “liberdade de movimento” que o *Auto-Suggest*. Entretanto, as sugestões funcionam melhor quando são uma combinação dos dois UIDP, na qual o sistema utiliza ideias de várias fontes.

O uso do UIDP *Auto-Suggest* pode auxiliar no funcionamento de implementações do padrão *Voice Search*. Apesar de erros de reconhecimento de fala serem muitas vezes piores do que por *fat-finger*, pesquisas por voz geralmente não fornecem sugestões automáticas. Nesse contexto, o uso conjunto de *Auto-Suggest* e *Voice Search* pode ser adequado, particularmente em aplicativos nos quais o sistema não tem a necessidade de ler as informações (sugestões) da tela para o usuário. Um exemplo desse padrão pode ser observado na [Figura 15](#), durante uma pesquisa realizada no aplicativo de troca de mensagens Messenger¹⁶, nativo para Android.

– **Características analisadas de AC e AS:** digitar no dispositivo móvel pode ser demorado e propenso a erros. Qualquer interação com teclas de tamanho reduzido e muito próximas, como é o caso de teclado em dispositivos móveis, pode tornar-se um desafio para usuários com deficiência motora ou visual. Desse modo, qualquer padrão que possa ajudar o usuário a inserir informações é de grande utilidade. No caso do *Auto-Complete*, tem-se auxílio na redução de erros de ortografia e no *Auto-Suggest*, é possível melhorar a especificidade da consulta, fornecendo opções mais precisas com menor número de caracteres digitados, bem como a redução de resultados nulos de pesquisa.

¹⁵ Erro de digitação / toque causado durante apontamento do dedo (gordo) na interação em área num dispositivo no qual as teclas ou tela são muito pequenas (NIELSEN, 2011, tradução nossa).

¹⁶ <<https://play.google.com/store/apps/details?id=com.facebook.orca>>

Figura 15 – Exemplo de implementação do UIDP Auto-Suggest

Fonte: Elaborada pelo autor.

4. *Search from Action Bar (SAB)*

A implementação do *Search from Action Bar* se caracteriza “(...) quando o usuário pode acessar a pesquisa por meio de um elemento clicável dedicado na barra de ação do aplicativo que quando tocado exibe pesquisas salvas, opções de refinamento de pesquisa, pesquisas populares, locais próximos, entre outros.” (NUDELMAN, 2013, tradução nossa).

Utilizado em aplicativos que possuem barra de ação nas quais a função de pesquisa é importante para o usuário, SAB escolhido pela Google para substituir o *Search from menu*, que não vem sendo utilizado em aplicativos nativos da Google desde a versão Android 4.0. A justificativa para essa troca é que forçar o usuário a realizar a pesquisa dentro de um menu pode levar a uma separação estranha da consulta de palavras-chave com as ferramentas de refinamento (NUDELMAN, 2013). Um exemplo de uso do UIDP *Search from Action Bar* pode ser visto na Figura 16, que mostra o elemento clicável para pesquisa na barra de ação e as opções de filtro no aplicativo Google+¹⁷.

– **Características analisadas de SAB:** O excesso de passos para acesso a ferramentas básicas em conjunto com a falta de agrupamento de funções similares interferem negativamente na acessibilidade do aplicativo. Assim, podem dificultar a compreensão de elementos do aplicativo, em especial por pessoas com deficiência intelectual, bem como dificultar a interação, principalmente por usuários com deficiência visual e/ou motora.

¹⁷ <<https://play.google.com/store/apps/details?id=com.google.android.apps.plus>>

Figura 16 – Exemplo de implementação do UIDP *Search from Action Bar*

Fonte: [Nudelman \(2013\)](#).

Ordenando e Filtrando

5. *Tabs* (TB)

A implementação de *Tabs* se caracteriza quando “(...) as guias na parte superior da página permitem que os usuários alterem as telas ou apliquem opções populares de classificação e filtro.” (NUDELMAN, 2013, tradução nossa).

Esse UIDP pode ser utilizado para acesso rápido a diferentes funcionalidades ou quando dados podem ser vistos de maneiras diferentes trazendo a necessidade de separação em telas. Por exemplo, os resultados de pesquisas mostrados aos usuários podem ser segregados em duas ou mais exibições no formato de guias na parte superior da página. A primeira guia geralmente é o resultado padrão e há a opção de tocar em uma das outras guias para alternar para uma visão diferente dos dados, como entre listas, galerias e vistas de mapa do mesmo conjunto de resultados.

– **Características analisadas de TB:** é um padrão bastante difundido, encontrado em muitos aplicativos populares, como o Youtube¹⁸. As [Figura 17a](#) e [Figura 17b](#) exemplificam o uso de *tabs* em dois aplicativos distintos. Em ambos, o padrão é usado para acesso rápido a diferentes conteúdos distribuídos entre abas do aplicativo.

¹⁸ Youtube é um dos aplicativos Android com maior número de usuários ativos no mundo, que segundo a [GSMA \(2017\)](#) obteve cerca de 1,5 bilhões de acessos somente no mês de julho de 2017.

Figura 17 – Exemplos de implementações do UIDP *Tabs* em aplicativos nativos Android

(a) Play Store

(b) Youtube

Fonte: Nudelman (2013).

Fonte: Elaborada pelo autor.

Entrada de Dados

6. Multiple select (MS)

Em *Multiple Select* “(...) quando o usuário toca em um ou mais itens (da galeria), a tela é alternada para o modo de seleção múltipla possibilitando alterações em massa, ou também quando o usuário toca em um ou mais valores (de uma lista exibida) e pode aplicar/descartar as seleções.” (NUDELMAN, 2013, tradução nossa). O padrão pode ser implementado em duas versões:

- Versão *drop-down*:** quando o usuário toca no controle, uma caixa *popover* é aberta com um conjunto de caixas de seleção que permitem a seleção de vários elementos da lista mostrada no *popover*. O usuário define o término da seleção clicando em um botão, ou pode descartar fechando a lista.
- Versão *Gallery*:** quando o usuário pode selecionar um ou mais itens da galeria ou lista, tocando e mantendo pressionado um dos itens. Isso alterna a galeria para o modo de seleção múltipla, que permite ao usuário escolher um ou mais itens e executar ações nos itens selecionados.

– **Características analisadas de MS:** existem tarefas nas quais o usuário precisa selecionar muitos valores de uma lista e, selecionar um item de cada vez é muito lento. A ausência desse UIDP pode diminuir a usabilidade do aplicativo (pois dificulta a execução da tarefa) e pode dificultar a interação de pessoas com deficiência motora e/ou visual. O *Multiple Select* é um padrão recomendado em aplicativos que possibilitam entrada de dados do usuário, pois fornece flexibilidade para operações em massa e pode ser usado em quaisquer aplicativos nos quais os usuários possam/necessitem selecionar mais de um item de uma longa lista.

A [Figura 18](#) mostra exemplos das duas versões do *Multiple Select* em aplicativos distintos: implementação do UIDP *drop-down* na inserção de um novo contato no aplicativo de agenda do SO ([Figura 18a](#)) e implementação da versão *gallery* do UIDP ([Figura 18b](#)) na seleção de diversas fotos na galeria do dispositivo.

Figura 18 – Exemplos de implementações das duas versões UIDP *Multiple Select*

Fonte: [Nudelman \(2013\)](#).

Fonte: Elaborada pelo autor.

7. Textbox with input mask (TBIM)

No UIDP TBIM “quando o campo aceita apenas dados específicos, como e-mail, número de telefone, RG ou CEP, o sistema pode fornecer o tipo certo de teclado para facilitar a entrada de dados. Além disso, dependendo da localização do rótulo, o campo pode opcionalmente mostrar a máscara de entrada (Input Mask) em sua parte interna” ([NUDELMAN, 2013](#), tradução nossa).

– **Características analisadas de TBIM:** o UIDP *Textbox with input mask* reduz os erros de entrada de dados e, portanto deve ser utilizado sempre que possível para produzir uma melhor experiência móvel. Também levamos em consideração sua frequência observada nos aplicativos escolhidos para as tarefas de validação com os especialistas, nos quais o UIDP foi implementado em telas de 78% das 9 tarefas definidas inicialmente, e apareceu em 100% das que ofereciam algum campo para inserção.

Um exemplo de seu uso é visto ao adicionar um novo contato na agenda do Android ([Figura 19](#)). Nessa tarefa, o campo de inserção do nome fornece o rótulo interno “Nome”, exibido em fonte de cor cinza clara, que permanece visível até o início da digitação. Para pessoas com deficiência visual, este UIDP deve ser adequadamente adaptado.

Figura 19 – Exemplo de implementação do UIDP Textbox with input mask

Fonte: [Nudelman \(2013\)](#).

Formulários

8. *Inline error message (IEM)*

No UIDP IEM “(...) quando ocorre um erro de entrada, o sistema notifica ao usuário quais campos precisam ser corrigidos. Geralmente, você pode reconhecer dois componentes desse UIDP: um indicador de erro no campo ou em torno dele e uma mensagem de erro geral (geralmente na parte superior da tela).” ([NUDELMAN, 2013](#), tradução nossa).

– **Características analisadas de IEM:** geralmente, é cansativo para o usuário, em especial àquele com alguma deficiência, procurar em formulários grandes por campos que necessitam de correção na inserção. Nos aplicativos analisados para elaboração das tarefas de validação, o padrão apareceu em 100% das vezes em que erros de digitação foram induzidos. Segundo [Nudelman \(2013\)](#), esse padrão resistiu ao teste do tempo, sendo ainda considerado a forma mais útil de se exibir erros de preenchimento de formulário.

Na [Figura 20](#) o UIDP IEM aparece durante a realização de um cadastro de usuário em um aplicativo nativo Android; ocorre que o usuário tenta avançar na tarefa deixando um campo obrigatório não preenchido.

9. *Callback Validation (CV)*

No UIDP *Callback Validation* “(...) quando uma parte dos dados de entrada do usuário precisa ser validado no servidor, o sistema detecta quando a entrada é concluída e emite uma chamada de servidor assíncrona para validar os dados, retornando com um de dois estados, OK ou Falha. Estados de falha são frequentemente acompanhados de sugestões dinâmicas que visam auxiliar o usuário na correção da condição de erro.” ([NUDELMAN, 2013](#), tradução nossa).

Às vezes, é impossível validar entradas em um formulário usando apenas o código do lado do usuário, como quando a validação não é referente a erros de digitação ou formatos

Figura 20 – Exemplo de implementação do UIDP *Inline error message*

Fonte: Nudelman (2013).

de dados, mas sim à disponibilidade de valores chave de identificação para cadastro de usuário. Por exemplo, ao inserir um nome de usuário já cadastrado, o sistema informa que o nome está indisponível e oferece sugestões baseadas nos caracteres originalmente inseridos, como exibido na Figura 21 durante a criação de uma conta de e-mail no *Gmail* nativo Android.

Figura 21 – Exemplo de implementação do UIDP *Callback Validation*

Fonte: Elaborada pelo autor.

– **Características analisadas de CV:** o aumento da velocidade e da disponibilidade de redes móveis também vem acompanhado do aumento nas expectativas dos usuários por um *feedback* contextual quase instantâneo. O *Callback Validation* auxilia no atendimento dessa exigência, pois observa diferenças no padrão de digitação esperado com relação ao texto inserido e informa ao usuário, em vez de esperar pelo envio de todo o formulário para relatar o erro (NUDELMAN, 2013).

10. *Cancel/OK (CO)*

Para projeto de formulários, é importante considerar que os usuários precisam confirmar ou cancelar algum envio de informação. Desse modo, “(...) *os botões de ação são posicionados como Cancel/OK na parte superior ou inferior do formulário. O botão principal está à direita e às vezes é maior ou implementado com uma cor mais saturada e/ou ícone. Frequentemente, o botão de ação primária está desativado no início do processo antes de um valor válido ser inserido no campo de texto*”. (NUDELMAN, 2013, tradução nossa).

A razão para implementar os botões de ação nessa ordem é que, ergonomicamente falando, o botão à esquerda é mais fácil de se tocar durante o uso do dispositivo móvel com uma mão. Nudelman (2013) argumenta que esta é uma convenção natural que decorre dos primeiros *designs* de interface de aplicativos móveis, que se basearam na tradição ocidental de leitura da esquerda para direita, sendo assim desnecessário alterar essa ordem.

Essa visão não considera personalização da interface para uma possível melhor experiência em uso por pessoas canhotas. De certo modo, alterar a ordem já consolidada de comandos de um UIDP tão utilizado poderia piorar drasticamente a usabilidade do aplicativo, obrigando o usuário a reacostumar com a nova organização. Assim, versões recentes do Android fornecem uma alternativa possibilitando que o usuário escolha o lado da tela preferido para interação. A Figura 22 exemplifica a apresentação do *Cancel/OK* no modo de operação com uma mão do Android 7.1. No caso, a ordem da opção cancelar (Cancel) e prosseguir (OK) não muda, apenas é possibilitado ao usuário utilizar uma versão reduzida na tela ao alcance dos dedos na sua mão preferida para interação com a interface.

Figura 22 – Exemplo do UIDP *Cancel/Ok* apresentado no modo de operação com uma mão do Android

Fonte: Elaborada pelo autor.

– **Características analisadas de CO:** em algum momento, o usuário necessitará confirmar ou cancelar uma ação, mesmo que não seja em uma inserção de dados. [Nudelman \(2013\)](#) ressalta que o padrão CO pode ser aplicado mesmo quando há apenas um botão, que deve estar do lado direito da tela para obter a mesma ergonomia de quando ambos estão presentes na interface. Nesse caso, o botão físico do dispositivo móvel ou o retorno da barra de ação são utilizados para a função de cancelamento. Para pessoas com determinada deficiência motora / física, este UIDP deve ser adequadamente adaptado.

A [Figura 23](#) exibe dois exemplos de implementação de *Cancel/Ok*. Nas [Figura 23c](#) e [Figura 23a](#), o padrão conta com ambos os botões e está localizado na parte inferior de um *pop-up* ou na barra de ação do aplicativo, respectivamente. A [Figura 23b](#) mostra a versão do padrão que exibe apenas o botão de conclusão da ação no aplicativo, deixando a função de retorno a cargo da barra de navegação do dispositivo.

Figura 23 – Exemplos de implementação do UIDP *Cancel/Ok*

Fonte: [Nudelman \(2013\)](#).

Navegação

11. *Carousel (CRS)*

Para navegação de produtos em destaque ou mesmo entre imagens em um aplicativo de galeria, o UIDP *Carousel* se destaca como uma das melhores soluções práticas de usabilidade para navegação por uma pequena coleção de produtos. Nesse UIDP “(...) o usuário vê várias imagens de produtos ao longo de uma fila horizontal. Para explorar mais produtos, o usuário pode deslizar pela fila para navegar horizontalmente até o próximo conjunto de produtos. Uma seta indicando a direção do movimento do *Carousel* geralmente é fornecida como uma dica para a interação no sentido desejado. Como alternativa, um dos produtos pode estar parcialmente escondido, criando uma prévia, indicando que mais conteúdo pode ser visualizado ao deslizar com o dedo ”(NUDELMAN, 2013, tradução nossa).

– **Características analisadas de CRS:** *Carousel* é um controle prático e intuitivo, que pode ser utilizado sempre que houver um pequeno conjunto de itens (8 a 20) que sejam facilmente reconhecíveis pela sua imagem; um exemplo de sua implementação está na [Figura 24](#). Além disso, [Nudelman \(2013\)](#) destaca que o UIDP se adapta bem em uma ampla variedade de tamanhos de dispositivos e resoluções de tela, aproveitando bem as funções e atalhos gestuais do dispositivo como *multitouch* e *swipping*. Esses atalhos e funções podem ser úteis como auxílio para pessoas com deficiência motora ou dificuldades de destreza interagirem com a interface.

Figura 24 – Exemplo de implementação do UIDP *Carousel* em um aplicativo fictício

Fonte: Elaborada pelo autor.

3.6 Considerações Finais

Neste capítulo foram apresentadas as pesquisas existentes sobre trabalhos que focam em boas práticas de projeto que auxiliam no desenvolvimento de aplicativos móveis em geral. Foram considerados, nesta pesquisa, especialmente os *Design Patterns* e outras boas práticas

para desenvolvimento de aplicativos para dispositivos móveis. Os UIDPs foram estudados em detalhe, visando obter uma visão geral do contexto prático que desenvolvedores de aplicativos móveis procuram adotar. Assim, foram descritos os trabalhos estudados da literatura, sobre o que é característico no desenvolvimento de aplicativos móveis (Seção 3.2) e sobre as boas práticas que têm sido utilizadas nesse tipo de desenvolvimento (Seção 3.3).

Em função do objetivo de pesquisar e identificar as boas práticas de desenvolvimento de software para dispositivos móveis, considerando acessibilidade e usabilidade, neste capítulo também foram descritos 12 UIDPs *mobile*, os quais representam os mais típicos exemplos (como indicados na Seção 3.5) que fundamentaram os critérios para compor o *checklist Acc_Mobile-Check*, desenvolvido neste mestrado. A análise das características dos UIDPs estudados levou em conta as possíveis situações em que problemas e soluções de acessibilidade e usabilidade puderam ser identificadas. No entanto, a partir da variedade e informalidade das descrições, é evidenciado que para os desenvolvedores, observar questões de usabilidade e acessibilidade torna-se um trabalho oneroso e difícil, pois requer conhecimentos mais especializados sobre como ocorrem as dificuldades de uso pelos usuários em geral.

A partir dos estudos sobre os 12 UIDPs *mobile* utilizados nesse trabalho, descritos na Seção 3.5, foram analisadas suas características e identificadas questões de acessibilidade / usabilidade (indicadas na Tabela 2). Porém, torna-se essencial que os requisitos de acessibilidade e usabilidade sejam revisados, para que sejam evidenciadas as questões importantes que auxiliem os desenvolvedores durante o seu trabalho.

Tabela 2 – Os 12 UIDPs *mobile* analisados

Classificação	UIDP <i>mobile</i>	Acessibilidade / Usabilidade
Tela Inicial	1. <i>List of Links</i>	
Busca	2. <i>Voice Search</i>	x
	3. <i>Auto-Complete</i>	x
	4. <i>Auto-Suggest</i>	x
	5. <i>Search from Action Bar</i>	x
Ordenando e filtrando	6. <i>Tabs</i>	
Entrada de dados	7. <i>Multiple select</i>	x
	8. <i>Textbox with Input Mask</i>	x
Formulários	9. <i>Inline Error Message</i>	x
	10. <i>Callback Validation</i>	
	11. <i>Cancel/OK</i>	x
Navegação	12. <i>Carousel</i>	x

Fonte: Elaborada pelo autor.

No próximo capítulo (Capítulo 4), são descritos os procedimentos realizados para desenvolvimento do *Acc_MobileCheck*.

DESENVOLVIMENTO DO *CHECKLIST*

4.1 Considerações Iniciais

Para obter softwares interativos com alta qualidade em uso, [Barbosa e Silva \(2010\)](#) ressaltam a importância da realização de avaliações, observando critérios de uso como usabilidade e acessibilidade, durante o desenvolvimento. Porém, existe uma lacuna em validações de qualidade ao longo do processo, pois em geral, quando são realizadas, não seguem padrões consolidados e não analisam qualidade em uso, por darem mais foco à codificação ([LUCINDA, 2010](#)).

Na etapa inicial deste trabalho, foi realizada uma investigação de boas práticas existentes para desenvolvimento e avaliação de aplicativos móveis ([Capítulo 3](#)). Com base nesse levantamento e na fundamentação dos conceitos estudados ([Capítulo 2](#)), foi possível constatar alguns fatores que dificultam a incorporação de avaliações de usabilidade e acessibilidade na criação de aplicativos móveis, tais como:

- espalhamento de informações em diversos documentos,
- falta de especificidade e/ou duplicidade de recomendações,
- documentos extensos,
- complexidade da definição e aplicação de diretrizes, exigindo conhecimento especializado do avaliador acerca do assunto,
- ausência de documentos com foco nos critérios de qualidade em uso, entre outros.

De fato, [Miñón et al. \(2014\)](#) concluíram que os desenvolvedores relatam possuir dificuldades tanto para compreender detalhes das recomendações existentes, quanto para acompanhar todas as exigências técnicas que possibilitem atingir níveis de qualidade que satisfaçam as expectativas para uso e acesso por todas as pessoas, incluindo as pessoas com deficiências.

Desse modo, métodos de validação que possam ser utilizados com maior frequência e com efetividade são uma abordagem que propicia uma alternativa objetiva para as dificuldades enfrentadas pelos desenvolvedores, uma vez que pode indicar objetivamente problemas específicos sobre a qualidade em uso, além de orientar os desenvolvedores para informações mais relacionadas ao trabalho que estão realizando.

Neste mestrado, com o objetivo de buscar por uma abordagem para verificação de boas práticas adotadas no processo de desenvolvimento de aplicativos móveis, quanto às questões de acessibilidade e usabilidade, foi escolhida a criação de um *checklist*. Esta escolha é justificada porque o instrumento *checklist* pode ser utilizado em inspeções, considerando-se um conjunto mínimo de recomendações diretamente aplicáveis, de baixo custo e reduzida complexidade de interpretação (WINCKLER; PIMENTA, 2002).

Neste capítulo são descritos os procedimentos realizados para desenvolvimento do *checklist* *Acc_MobileCheck*, que tem por objetivo auxiliar desenvolvedores e especialistas a conferirem o atendimento de critérios e padrões de usabilidade e acessibilidade em UI, no contexto *mobile*, de um modo simples e adaptável a diferentes objetivos de avaliação.

A Figura 25 esquematiza os componentes considerados durante as análises no desenvolvimento do *Acc_MobileCheck* para identificação das questões do *checklist*. Assim, inicialmente os estudos sobre acessibilidade e usabilidade propiciaram a identificação dos tipos de **problemas** que os usuários enfrentam, em especial as pessoas com deficiência. Esses problemas, bem como as diretrizes e recomendações de acessibilidade e usabilidade, foram então analisados, mapeando-os em um conjunto selecionado de **aplicativos móveis** e respectivos **UIDPs** que os aplicativos implementavam.

Figura 25 – Componentes integrantes para desenvolvimento de *Acc_MobileCheck*.

Fonte: Elaborada pelo autor.

O mapeamento foi realizado (definido como procedimento de pesquisa neste mestrado - vide Figura 5) entre as perspectivas mais abstratas (requisitos, definições e diretrizes) sobre acessibilidade e usabilidade, e as perspectivas mais práticas (UIDPs) adotadas no âmbito do

desenvolvimento de aplicativos móveis. Com base nesse cenário, foram obtidas questões candidatas, as quais passaram por um meticuloso procedimento de análise, descrito na [Seção 4.3](#), confrontando-se aplicativos móveis reais (e seus UIDPs) com as diretrizes atuais, para comporem o *checklist*.

4.2 Decisões de projeto

A necessidade de um método de avaliação que pudesse indicar de modo mais objetivo os problemas de usabilidade e de acessibilidade, foi a principal motivação para a elaboração do *checklist Acc_MobileCheck*.

A decisão por estudar as características de UIDPs é devido a eles serem padrões que descrevem projetos de soluções para problemas recorrentes, de usabilidade na interação do usuário com a interface, ainda que possibilitando implementações diferentes. Desse modo, ao identificá-los na interface, apenas comparando com sua definição no UIDP, pode-se verificar se foi realizada implementação adequada. Além disso, o *Material Design*, foi também estudado, por ser boa prática de *design mobile* atual, que traz grande parte das recomendações de *design* dos elementos da interface (tamanho, espaçamento, cor, localização, função de toque, etc.) aplicáveis a diferentes situações na UI, definidos em uma estrutura similar a um UIDP *mobile*.

O usuário interage com o conteúdo de um aplicativo de diferentes modos, seja em *mobile* ou em *desktop*. Assim, existem diferenças nos UIDPs de cada dispositivo de interação, e mesmo entre diferentes sistemas operacionais para o mesmo dispositivo. Portanto, a análise dos UIDPs para todas variáveis se tornaria inviável; foi conferido foco em aplicativos nativos para dispositivos móveis, com um determinado sistema operacional, conforme discutido a seguir.

Escolha do sistema operacional do dispositivo móvel

Uma das características de *checklists* é a possibilidade de adaptação para diversas situações de avaliação ([WINCKLER; PIMENTA, 2002](#)). De fato, os itens de verificação do *Acc_MobileCheck* podem ser aplicados em avaliações de usabilidade e acessibilidade em UI de diferentes plataformas móveis. Entretanto, alguns itens de verificação possuem recomendações técnicas específicas e não há garantias de que funcionarão em plataforma diferente da qual as recomendações foram propostas.

Como o *Acc_MobileCheck* visava auxiliar a avaliação de usabilidade e acessibilidade nos aplicativos móveis, possibilitando a diminuição de barreiras de interação para o maior número de usuários possível, a abrangência do resultado deste trabalho deveria estar diretamente proporcional ao número de usuários do SO escolhido. Assim, foi escolhido o SO com maior alcance pelos usuários, atualmente.

Segundo dados reportados de pesquisas, em Janeiro de 2017, haviam aproximadamente

4,92 bilhões de usuários *mobile* no mundo (66% da população), sendo que 72% acessavam a internet por meio de dispositivos com Android instalado ¹. No Brasil, dos cerca de 157 milhões de *smartphones* ativos com acesso à internet em 2016, 90% possuíam o sistema operacional Android instalado (AVAZU, 2016). Desse modo, levando em conta também que as boas práticas estudadas são direcionadas ou aplicáveis a esse SO, foram considerados nesta pesquisa os UIDPs relacionados a aplicativos nativos para **Android**.

Escolha dos UIDP mobile

Como UIDPs são padrões que descrevem soluções de *design* para problemas recorrentes de usabilidade na interação do usuário com a interface, um estudo mais aprofundado foi realizado, investigando particularmente como a acessibilidade deveria ser considerada. Para realização desse estudo aprofundado, em abril de 2017 foram selecionados 10 aplicativos móveis disponibilizados no mercado à época, na busca por UIDPs que desenvolvedores e usuários mais utilizam. Na Tabela 3, estão listados os aplicativos selecionados, discriminando-se suas origens (*download* ou fábrica) e categorias.

Tabela 3 – Aplicativos móveis selecionados para investigação por UIDP

Categoria	Origem	Download		Instalado	
Comunicação		<i>WhatsApp</i>	<i>Messenger</i>		
Social		<i>Facebook</i>	<i>Instagram</i>		
Reprodução áudio/ vídeo		Palco MP3		<i>Youtube</i>	
Compras e busca		Mercado Livre		<i>Google Search</i>	
Armazenamento arquivos				<i>Google Photos</i>	<i>Google Drive</i>

Fonte: Elaborada pelo autor.

A seleção dos aplicativos considerou inicialmente os 6 mais pesquisados do repositório da loja de aplicativos *Google Play*² (obtidos por meio de *download*), e os demais 4 aplicativos selecionados, entre os originais do Android (instalados pelo fabricante). Em ordem decrescente de popularidade entre os usuários, foram selecionados aplicativos com as seguintes características:

- não repetidos, ou seja, foram descartadas versões diferentes de um mesmo software,
- que ofereciam maior número de funcionalidades para análise acerca dos UIDPs implementados.

¹ <<https://wearesocial.com/special-reports/digital-in-2017-global-overview>>

² *Google Play* é um serviço de distribuição digital de aplicativos, jogos, filmes, programas de televisão, músicas e livros, desenvolvido e operado pela empresa *Google*.

Os dez aplicativos selecionados (*WhatsApp*³, *Messenger*⁴, *Facebook*⁵, *Instagram*⁶, *Youtube*⁷, *Google Search*⁸, *Google Drive*⁹, *Google Photos*¹⁰, *Mercado Livre*¹¹ e *Palco MP3*¹²) foram observados criteriosamente, examinando-se os UIDPs neles implementados. Foram utilizadas as definições de [Nudelman \(2013\)](#) sobre cada UIDP; os UIDPs encontrados foram analisados com base nos seguintes critérios de inclusão (CI) e critérios de exclusão (CE) definidos:

- ✓ [CI 1]: UIDP *mobile* observado com fácil evidência e muita frequência durante a interação;
- ✓ [CI 2]: UIDP *mobile* cuja ausência ou implementação ruim dificultariam ou impediriam o uso do aplicativo considerando também o contexto de acessibilidade.

- ✗ [CE 1]: UIDP *mobile* com presença reduzida na interface e direcionado a situação de uso muito pontual;
- ✗ [CE 2]: UIDP *mobile* observado com pouca frequência e pouco evidente durante a interação;
- ✗ [CE 3]: UIDP *mobile* cuja implementação dificultaria ou impediria o uso do aplicativo considerando também o contexto de acessibilidade.

Os UIDPs, ordenados por frequência de observação, após a aplicação dos critérios CI e CE foram então selecionados como componente de estudo nesta pesquisa. São eles:

1. *List of Links*;
2. *Voice Search*;
3. *Auto-Complete*;
4. *Auto-Suggest*;
5. *Search From Action Bar*;
6. *Tabs*;
7. *Multiple Select*;
8. *Textbox with input Mask*;
9. *Inline Error Message*;
10. *Callback Validation*;
11. *Cancel/Ok*;
12. *Carousel*

---

³ <[whatsapp.com/dl](https://www.whatsapp.com/dl)>

⁴ <<https://play.google.com/store/apps/details?id=com.facebook.orca>>

⁵ <<https://play.google.com/store/apps/details?id=com.facebook.katana>>

⁶ <<https://play.google.com/store/apps/details?id=com.instagram.android>>

⁷ <<https://play.google.com/store/apps/details?id=com.google.android.youtube>>

⁸ <<https://play.google.com/store/apps/details?id=com.google.android.googlequicksearchbox>>

⁹ <<https://play.google.com/store/apps/details?id=com.google.android.apps.docs>>

¹⁰ <<https://play.google.com/store/apps/details?id=com.google.android.apps.photos>>

¹¹ <<http://www.mercadolivre.com.br/appmobile>>

¹² <<https://play.google.com/store/apps/details?id=com.studiosol.palcomp3>>

As descrições sucintas baseadas na obra utilizada e algumas discussões sobre as características estudadas em cada um desses 12 UIDPs foram apresentadas na [Seção 3.5](#).

4.3 *Acc_MobileCheck*

Como boas práticas estudadas, foram encontrados documentos relevantes para auxiliar na proposta de questões para compor o *checklist*, alvo desta pesquisa. Dentre eles, conforme indicados no [Capítulo 2](#) e no [Capítulo 3](#), os que serviram como base para o desenvolvimento do *Acc_MobileCheck* foram: WCAG 2.0 ([W3C, 2008a](#)), *Mobile Web Best Practices* ([W3C, 2008b](#)), *Mobile Web Application Best Practices* ([W3C, 2010](#)); Heurísticas de usabilidade ([NIELSEN, 1995](#)); definições de Qualidade em Uso ([ISO, 2011](#)); *Google Developer e Accessibility* ([GOOGLE, 2017](#)) e UIDPs *mobile* ([NUDELMAN, 2013](#)).

A partir do estudo dessas boas práticas, foram elaboradas perguntas de avaliação (**itens de verificação**) considerando-se como parâmetros, questões relacionadas a acessibilidade e usabilidade, sempre que indicassem recomendações. Assim, a primeira versão do *Acc_MobileCheck* foi desenvolvida em setembro de 2017, com **241 itens de verificação**. Inicialmente, o projeto contava com duas especialistas, mas ainda não havia sido dado foco em UIDP's. O processo de desenvolvimento do *Acc_MobileCheck*, desde a elaboração dos itens de verificação, com base nos documentos estudados, até a sua última versão está ilustrado na [Figura 26](#).

Acc_MobileCheck - 1a. Etapa

A primeira etapa de melhoria iniciou pela remoção de itens de verificação duplicados. Além disso, foi incluído um terceiro pesquisador, especialista em usabilidade e acessibilidade, visando obter mais um parecer sobre a clareza das questões. Durante essa etapa (setembro/2017 a fevereiro/2018), foram realizadas as seguintes atividades:

- **Remoção de não relação ao contexto móvel** - foram removidas questões que abordavam situações exclusivas de ambiente ou elementos *desktop*, por exemplo:
  - recomendações de uso e atalhos por combinações do teclado e mouse, que não consideram o funcionamento por *touch-screen*,
  - questões que não teriam influência no contexto móvel analisado, como por exemplo, recomendações acerca da compatibilidade do aplicativo (web) com outro SO.
- **Remoção de não avaliabilidade** - foram removidas questões cuja verificação exigia mais recursos, por exemplo:
  - perguntas nas quais era necessário analisar o código para conferir o atendimento da recomendação relacionada, como por exemplo o uso de *tags* em *HTML* e diretrizes de navegação e alocação de elementos exclusivas de aplicativos web, como *breakpoints*,

Figura 26 – Processo de desenvolvimento do *Acc_MobileCheck*.

Fonte: Elaborada pelo autor.

- perguntas consideradas muito subjetivas, como por exemplo “*Os menus são bonitos?*”. De fato, a estética da interface é importante para uma interação agradável, inclusive tratada como atributo de usabilidade em qualidade interna e externa na ISO (2011), porém o *checklist* deve priorizar parâmetros mensuráveis, nos quais o avaliador prevê possíveis problemas que o usuário possa vir a ter utilizando a interface (BARBOSA; SILVA, 2010).
- **Remoção de não relação a UIDPs mobile** - todos os 66 itens de verificação foram analisados em relação aos 12 UIDPs mobile selecionados. Como espera-se que a *Acc_MobileCheck* seja útil para aplicação em boas práticas bastante utilizadas em desenvolvimento (UIDP mobile), os itens de verificação que não se aplicavam a nenhum deles foram removidos, por exemplos: questão que não depende do aplicativo móvel, mas do gerenciamento do sistema (“*As mensagens relativas a informações críticas da aplicação, como status da bateria, são exibidas prioritariamente em relação às demais?*”).

Durante essa primeira etapa, os itens de verificação foram validados em relação à objetividade, aplicabilidade e relevância das perguntas, com discussão e análises pelas três especialistas em usabilidade, resultando na segunda versão.

Acc_MobileCheck - 2a. Etapa

A partir da segunda versão do ***Acc_MobileCheck*** (com 59 itens de verificação), fez-se necessário estudar maneiras de tornar a avaliação mais prática e específica em relação a barreiras de acessibilidade. Essa segunda etapa, que resultou na terceira versão do ***Acc_MobileCheck***, foi realizada nos dois meses consecutivos e foi dedicada a especificar parâmetros que servissem de suporte informacional relativo a requisitos de acessibilidade. Assim, foram definidos quatro tipos de problemas, visando auxiliar a identificação dos diferentes tipos de barreiras de acesso que pessoas com deficiência pudessem se deparar. Esses tipos de problemas representam as potenciais dificuldades na interação atribuídas a cada um deles. Os problemas foram derivados dos 4 princípios de acessibilidade de WCAG 2.0 (vide [Seção 2.4](#) - perceptível, operável, compreensível e robusto). Os quatro 4 tipos de problemas (TP) são identificados conforme a seguir:

1. Problemas de Compreensão (C) - relacionados a dificuldades do usuário para realizar cognitivamente as seguintes tarefas:

- entender para o que serve um item clicável ou qualquer elemento não textual;
- entender onde ele está localizado atualmente no aplicativo;
- entender o vocabulário e/ou idioma utilizado no aplicativo;
- entender as mensagens do aplicativo;
- saber quais alternativas existem para corrigir um erro/ engano;
- aprender como utilizar o aplicativo;
- entender as consequências de uma determinada interação;
- inserir um texto correto para a busca desejada e compreender o contexto do resultado obtido.

2. Problemas de Operação (O) - relacionados a dificuldades do usuário para realizar operações nas seguintes tarefas:

- alcançar algum recurso do aplicativo de modo eficiente ou facilitado;
- apontar no elemento clicável;
- usar o teclado do aplicativo para inserção de dados;
- obter a partir do contexto visual, textual, sonoro ou tátil o status atual de uma tarefa em andamento;
- interagir com o aplicativo sem o uso de Tecnologia Assistiva¹³;
- desabilitar algum recurso indesejado no aplicativo;
- cancelar, desfazer uma ação ou retomar o ponto onde parou durante o uso do aplicativo;

¹³ “(..) produtos, equipamentos, dispositivos, recursos, metodologias, estratégias, práticas e serviços que objetivem promover a funcionalidade, relacionada à atividade e à participação da pessoa com deficiência ou com mobilidade reduzida, visando à sua autonomia, independência, qualidade de vida e inclusão social” (BRASIL, 2015).

- operar e interagir com o aplicativo e seus recursos sem uso de voz ou gestos.

3. Problemas de Percepção (P) - relacionados a dificuldades do usuário para perceber as seguintes tarefas:

- identificar no contexto visual, textual, sonoro ou tátil notificações/sugestões dadas pelo aplicativo, inclusive durante o uso de tecnologia(s) assistiva(s);
- reparar a existência de conteúdo no aplicativo além do exibido no momento;
- visualizar em relação a cor, tamanho, forma, contraste e organização o conteúdo do aplicativo;
- identificar o resultado visual, textual, sonoro ou tátil de uma interação de seleção sobre um elemento do aplicativo;
- notar no contexto visual, textual, sonoro ou tátil o conteúdo do aplicativo, inclusive durante o uso de alguma Tecnologia Assistiva.

4. Problemas de Adaptação (A) - relacionados a dificuldades do usuário para enfrentar as seguintes situações:

- necessita de auxílio personalizado/customizado para o uso do aplicativo;
- possui dificuldade em utilizar o aplicativo em uma só orientação específica (vertical/horizontal) ou em diferentes tamanho de tela.

Com base nos itens do *Acc_MobileCheck*, visando reduzir as dificuldades de acesso para cada um dos 4 tipos de deficiência considerados neste trabalho (visual, auditiva, motora e intelectual), foi possível relacionar cada tipo de problema a um tipo de deficiência. Essa relação entre os tipos de problemas e de deficiências, que foi identificada durante o desenvolvimento do *Acc_MobileCheck*, é representada na [Figura 27](#).

Figura 27 – Relação entre os tipos de problemas e de deficiências, observada nos itens de verificação do *Acc_MobileCheck*.

Fonte: Elaborada pelo autor.

Nessa segunda etapa (março e abril/2018), foram realizadas as seguintes atividades:

- **Separação UIDP e TP** - os itens de verificação foram agrupados por similaridades, considerando-se as definições dos tipos de problemas (TP) e da relação com os tipos de deficiência. Desse modo, foi possível examinar as interfaces dos aplicativos móveis, guiando-se pelo tipo de problema ou deficiência analisada. Por exemplo, para analisar a interface em relação a critérios direcionados a deficiência auditiva, seria útil e prático que o avaliador pudesse selecionar o tipo de problema “Percepção”. Assim, seriam exibidos os itens de verificação relacionados a esse contexto. Vale destacar que um item de verificação está relacionado a apenas um tipo de problema, mas pode conferir se relacionar a mais de um tipo de deficiência.

Nessa etapa, cada um dos itens de verificação já discriminados nos grupos por TP, foram relacionados ao UIDP que lhe deu origem e aos demais cujo item de verificação possa analisar (a relação completa pode ser vista no [Apêndice B](#)).

A [Figura 28](#) mostra um exemplo de como UIDPs analisados em um item de verificação (C14 - “São fornecidas sugestões para o texto inserido em uma busca?”) aparecem concomitantemente: (1) *Voice Search*, no elemento destacado como 2, em ambas as telas; (2) *Auto-Suggest*, no elemento destacado 3, em ambas as telas implementado no aplicativo do teclado e, na [Figura 28b](#) implementado no aplicativo *Facebook*; (3) *Search from Action Bar*, no elemento destacado 1, em ambas as telas e (4) *Textbox With Input Mask*, aparecendo parcialmente (apenas o teclado mais apropriado para inserção) na [Figura 28b](#) e completo, com a máscara de entrada na [Figura 28a](#).

Figura 28 – Implementações de UIDPs *mobile* diferentes na mesma tela

Fonte: Elaborada pelo autor.

No início da elaboração do *checklist*, um 13º UIDP selecionado foi incluído ([Seção 4.2](#)), o *Swiss-Army-Knife Navigation*. Porém, foi removido pois os itens de verificação relacio-

nados exclusivamente a ele analisavam características já atendidas pelos demais; assim, foram considerados redundantes.

- **Melhoramento após aplicação em avaliação** - foram realizadas três avaliações, utilizando o *Acc_MobileCheck* versão 2, em um aplicativo *m-learning* nativo *Android*. Os avaliadores são especialistas em avaliação de qualidade de *software*, com mais de 2 anos de experiência na área. Durante a aplicação dos itens de verificação de compreensão foram observadas possíveis barreiras de interação não consideradas, em especial para pessoas com deficiência cognitiva. Assim, foi incluído o item de verificação “[C16] Os elementos não textuais clicáveis da aplicação são compreensíveis para o usuário?”, resultando na terceira versão do *Acc_MobileCheck*.

A Figura 29 resume os componentes analisados para o desenvolvimento dos itens de verificação do *Acc_MobileCheck*. A versão atual (terceira) conta com 47 itens de verificação apresentados como enunciados de questões. Essas questões estão relacionadas a cada UIDP cujos elementos podem ser verificados/questionados quanto a acessibilidade/usabilidade, e por tipo de problema de acessibilidade, totalizando: 16 questões sobre problemas de Compreensão, 15 sobre problemas de Operação, 12 sobre problemas de Percepção e 4 sobre problemas de Adaptação¹⁴.

Figura 29 – Componentes analisados para proposta dos itens de verificação do *Acc_MobileCheck*.

Fonte: Elaborada pelo autor.

Na definição de cada um dos itens de verificação, tem-se o seu título/ enunciado da

¹⁴ Doravante, vamos nos referir a esses 4 tipos de problemas por meio das suas letras iniciais: COPA.

questão, seguido das seguintes informações: (1) **Contexto** no qual a questão se aplica; (2) **Exemplo** de situação na qual seria necessário verificar na interface sob avaliação, qual é a resposta a ser dada diante do questionamento no enunciado; (3) **Motivação** em termos dos ganhos de acessibilidade na interface, levantando para qual(is) tipo(s) de deficiência(s) o adequado atendimento à questão traria melhorias na interação; (4) **Recomendação** fornecendo informações mais abrangentes do escopo de origem da questão; (5) **Fonte** que indica os documentos utilizados para fundamentar a questão e (6) **Respostas** com as possíveis opções de respostas, além de observações pelo avaliador.

O **Quadro 2** apresenta um exemplo de item de verificação do *Acc_MobileCheck* conforme a estrutura descrita, para o item de verificação relativo a problema de Compreensão, nomeado como C5.

Quadro 2 – Exemplo da estrutura de um dos itens de verificação do *Acc_MobileCheck*

[C5] A linguagem utilizada nos elementos textuais é legível e compreensível?				
Contexto: Quando a aplicação apresenta conteúdo textual.				
Exemplo: Quando o texto não possui uma linguagem clara ou o conteúdo não está em um mesmo idioma, ou mesmo, utiliza abreviações, termos técnicos, jargões.				
Motivação: Auxiliar usuários com deficiência VISUAL que utilizam leitor de tela, a compreender o contexto da aplicação, a fim de entender a mensagem que o conteúdo da aplicação quer dizer. Auxiliar usuários com dificuldades de entendimento do significado da linguagem da aplicação, a fim de atender necessidades de usuários com deficiência INTELECTUAL.				
Recomendação: De acordo com a 2ª Heurística de Nielsen [3], “o sistema deve falar a linguagem do usuário ao invés de usar termos orientados ao sistema”. Desse modo, de acordo também com as recomendações da WCAG 2.0 [2], o usuário precisa conseguir ler e entender o que está lendo, independente se estiver sendo lido pelo próprio usuário ou com auxílio de leitor de tela. Partes textuais com termos técnicos ou distantes da linguagem comum ao usuário prejudicam a compreensão do conteúdo da tela para navegação, compreensão para uso das ferramentas, bem como a correção de possíveis erros informados pelo software. Escolha palavras comuns que sejam claras e facilmente compreensíveis para leitores iniciantes e avançados, o texto deve ser compreensível por qualquer pessoa, em qualquer lugar, independentemente da sua cultura ou idioma [1].				
Fontes: [1] “Writing”. <i>Material Design</i>, Google Inc., 2014. [2] <i>Guideline “3.1 Readable”</i>. WCAG 2.0, W3C, 2008. [3] Nielsen, J., (1994). “<i>Heuristic evaluation. Usability Inspection Methods</i>”. New York: John Wiley & Sons, Inc.				
Resposta:	☐ Sim	☐ Parcialmente	☐ Não	☐ Não se aplica
Observações:				

Fonte: Elaborada pelo autor.

A utilização do *checkist*, com os 47 itens de verificação do *Acc_MobileCheck* foi validada por especialistas e desenvolvedores, conforme descrito na [Seção 4.4](#). No [Apêndice A](#), encontra-se o *Acc_MobileCheck* na íntegra, a seguir estão os enunciados dos itens de verificação de cada tipo de problema.

Problemas de Compreensão:

- [C1] Existe uma descrição textual sucinta para todos os elementos clicáveis?
- [C2] Existe descrição para todos os elementos não textuais que possibilite leitura por T.A.?
- [C3] São fornecidos rótulos e títulos de seção descritivos em todas as telas da aplicação?
- [C4] As telas com o mesmo tipo de conteúdo possuem o mesmo título?
- [C5] A linguagem utilizada nos elementos textuais é legível e compreensível?
- [C6] Existem descrições de fácil acesso para todas as abreviaturas utilizadas?
- [C7] O idioma utilizado na aplicação é o mesmo em todo seu conteúdo?
- [C8] As mensagens de erro ao usuário são claras?
- [C9] É fornecida uma alternativa de correção com as etapas e ações necessárias para o usuário corrigir o erro?
- [C10] As mensagens da aplicação são compreensíveis para o usuário?
- [C11] Existe ajuda ou documentação sobre como utilizar a aplicação?
- [C12] Na primeira interação do usuário ou com o surgimento de atualização nas funcionalidades, a aplicação oferece instruções básicas de uso?
- [C13] Quando um usuário invoca uma ação importante, é questionado se ele deseja realmente prosseguir com a ação?
- [C14] São fornecidas sugestões para o texto inserido em uma busca?
- [C15] Em resposta à inserção de um texto para busca, o texto é novamente apresentado além do resultado da busca?
- [C16] Os elementos não-textuais clicáveis da aplicação são compreensíveis para o usuário?

Problemas de Percepção:

- [P1] As notificações de atualização são perceptíveis?
- [P2] As mensagens da aplicação são perceptíveis ao usuário?
- [P3] Os elementos ficam perceptíveis em diferentes orientações de tela do aparelho móvel?
- [P4] Quando todos os elementos a serem exibidos não cabem na tela da aplicação, é perceptível ao usuário que existem outros elementos a serem exibidos?
- [P5] O contraste entre as cores do plano de fundo com elementos textuais e não textuais possibilita a percepção e visualização desses elementos?
- [P6] O tamanho da fonte dos elementos textuais e o tamanho dos elementos não textuais possibilitam a percepção e visualização desses elementos?
- [P7] A aplicação fornece alternativas acessíveis para elementos não textuais?
- [P8] Existe *feedback* visual ao usuário quando um elemento clicável é selecionado?

- [P9] Existe *feedback* tátil ou sonoro ao usuário quando um elemento clicável é selecionado?
  - [P10] É informado ao usuário o número de itens selecionados no momento atual?
  - [P11] O usuário é avisado quando o áudio da inserção por voz não é compreendido?
  - [P12] É fornecido *feedback* perceptível da existência de sugestões de busca ao usuário durante a utilização T.A.?
-

Problemas de Adaptação:

- [A1] É possível a personalização das funcionalidades para acesso mais rápido?
  - [A2] É possível alterar tamanho e fonte dos elementos textuais da interface?
  - [A3] A aplicação suporta ambas as orientações?
  - [A4] O *design* dos elementos se ajusta aos diferentes tamanhos de tela do aparelho móvel?
-

Problemas de Operação:

- [O1] A aplicação permite realizar atividades básicas utilizando interação por Gestos?
  - [O2] A reprodução de sons ocorre somente quando configurada pelo usuário?
  - [O3] O usuário pode sair da aplicação e retornar do ponto onde parou?
  - [O4] O usuário pode cancelar uma ação (um processamento) em progresso?
  - [O5] O usuário pode desfazer uma ação?
  - [O6] É possível fazer alterações importantes, como modificação das configurações da aplicação, via comando de voz?
  - [O7] Existe *feedback* visual ao usuário acerca do status atual da tarefa?
  - [O8] Existe *feedback* tátil ou sonoro ao usuário acerca do status atual da tarefa?
  - [O9] A ajuda é de fácil acesso?
  - [O10] A troca de idioma é de fácil acesso?
  - [O11] É possibilitada a utilização de todas as funcionalidades da aplicação por T.A.?
  - [O12] É possibilitado o preenchimento dos campos de inserção pelo usuário com entradas de voz?
  - [O13] A área clicável de todos os elementos clicáveis ocupa toda e apenas a extensão dos mesmos?
  - [O14] A aplicação possui a função de busca por voz?
  - [O15] As funcionalidades mais comumente utilizadas estão disponibilizadas para acesso mais rápido?
-

4.4 Validação da *Acc_MobileCheck*

Boehm (1984) define verificação como a conferência de que o objeto avaliado foi construído de maneira correta a atender seus objetivos, e validação como a conferência de que foi construído o objeto correto para alcançar um objetivo.

Com o intuito de validar o *Acc_MobileCheck* para ser utilizado como apoio à atividade de avaliação de conformidade a requisitos de usabilidade e acessibilidade em aplicativos móveis, foram convidados especialistas em acessibilidade e usabilidade, bem como desenvolvedores de aplicativos móveis, para experimentarem o *checklist* desenvolvido. Participaram da validação cinco especialistas em acessibilidade e usabilidade, além de dois desenvolvedores de aplicativos móveis mais experientes em avaliação e um desenvolvedor novato em avaliação. Todos os 8 participantes são alunos de mestrado e doutorado do PPG/CCMC do ICMC/USP da área de IHC, com mais de 3 anos de experiência na área.

Os especialistas e desenvolvedores foram convidados por e-mail, e foram selecionados 8 (oito) deles, pois essa quantidade de participantes facilitava a separação por grupos de tarefas para abranger diversos aspectos do *checklist*, além de possibilitar uma análise sobre as visões dos diferentes perfis de avaliadores. O convite apresentava as informações sobre a pesquisa e uma visão geral do que seria solicitado durante a avaliação. Solicitava-se que cada avaliador aplicasse o *Acc_MobileCheck* durante a realização de duas tarefas em um (ou dois, pois as tarefas poderiam ser em aplicativos diferentes) aplicativo móvel e, ao final de cada tarefa, respondesse ao questionário *System Usability Scale* (SUS) (Seção 4.4) acerca da sua experiência no uso do *checklist*. Além disso, ao final de cada questionário, o avaliador poderia fornecer quaisquer comentários pertinentes.

Outro e-mail foi enviado aos participantes (que aceitaram cooperar), com instruções sobre como realizar a avaliação, informando o *link* da versão online do *Acc_MobileCheck* (na íntegra, vide Apêndice A). O procedimento requeria do avaliador que executasse duas tarefas, cada uma utilizando um grupo de itens de verificação de um tipo de problema diferente, a fim de que o mesmo avaliador não tivesse viés por conhecer melhor o grupo de questões na segunda avaliação.

Todos os participantes assinaram o Termo de Consentimento Livre e Esclarecido (TCLE) (Apêndice E) antes de iniciarem a validação. Tanto os 5 avaliadores “especialistas” quanto os 3 “desenvolvedores” realizaram avaliações usando o *Acc_MobileCheck* como instrumento de apoio, com a seguinte diferença:

- aos especialistas foram atribuídas as tarefas no aplicativo móvel e o tipo de problema (dentre os COPA¹⁵);
- aos desenvolvedores foi solicitado que realizassem a avaliação em seus próprios protótipos de aplicativo, cujas tarefas eles mesmos definiram a partir de recomendações informadas previamente. Também foram determinados os tipos de problemas que os desenvolvedores iriam analisar, mas como eles definiram suas próprias tarefas, foi dada a liberdade de escolha da tarefa na qual iriam utiliza-los (dentre os COPA).

As tarefas atribuídas aos especialistas foram definidas em aplicativos móveis populares,

¹⁵ sigla com as letras iniciais dos 4 tipos problemas: Compreensão, Operação, Percepção e Adaptação

nos quais era possível analisar a interface para conferir quais UIDPs *mobile* apareceriam. Como os desenvolvedores realizaram as tarefas em seus próprios protótipos, que não estavam disponíveis para *download*, não foi possível a identificação dos UIDPs *mobile* presentes nas tarefas definidas por eles. Desse modo, para permitir as mesmas variáveis obtidas de ambos os grupos de avaliadores, foi pedido aos desenvolvedores que realizassem capturas das telas que obtinham durante a realização de cada tarefa.

As tarefas para a validação do *Acc_MobileCheck* atribuídas aos especialistas (TEs) estão descritas no [Apêndice C](#), e as realizadas pelos desenvolvedores (TDs) estão descritas no [Apêndice D](#). Nas subseções seguintes, são resumidas informações acerca da execução das avaliações realizadas com os 8 participantes, seus resultados e análises.

Execução das avaliações

Cada avaliador realizou duas tarefas **diferentes**, cada uma analisando **um** tipo de problema **diferente**, para execução da avaliação usando o *Acc_MobileCheck*. Cada tipo de problema no *Acc_MobileCheck* é analisado por itens de verificação exclusivos. Assim, cada avaliador aplicou somente um conjunto de itens de verificação do *Acc_MobileCheck*.

As tarefas dos especialistas (TE) foram pré-definidas e distribuídas entre eles com os tipos de problemas (TP: Compreensão (C), Operação (O), Percepção (P) ou Adaptação (A)) designados para avaliação. Desse modo, foi possível analisar uma mesma tarefa por dois avaliadores e com diferentes itens de verificação. A distribuição das tarefas (TE1 a TE4), com os respectivos potenciais tipos de problemas analisados pelos 5 especialistas (E1, E2, E3, E4 e E5) estão resumidos na [Tabela 4](#).

Tabela 4 – Distribuição das tarefas e potenciais tipos de problemas aos especialistas

TP \ Tarefas	TE1	TE2	TE3	TE4	Qtdd. Especialistas / TP
C	-	-	E1	E5	2
O	E4	E2	-	E1	3
P	E5	E3		E2	3
A	E3	-	E4	-	2
Nr. Especialistas / Tarefa	3	2	2	3	

Fonte: Elaborada pelo autor.

As tarefas dos desenvolvedores (TD) foram elaboradas por eles em seus protótipos de aplicativo móvel, porém, os tipos de problemas analisados (TP: Compreensão (C), Operação (O), Percepção (P) ou Adaptação (A)) foram designados para cada avaliador. De modo análogo à distribuição das tarefas dos especialistas, cada desenvolvedor avaliou um tipo de problema diferente, porém, cada tarefa do desenvolvedor foi realizada uma única vez. A distribuição das

tarefas (TD1 a TD6) com os respectivos potenciais tipos de problemas (TP) analisados pelos desenvolvedores (D1, D2 e D3) estão resumidos na [Tabela 5](#).

Tabela 5 – Distribuição das tarefas e potenciais tipos de problemas aos desenvolvedores

TP \ Tarefas	TD1	TD2	TD3	TD4	TD5	TD6	Qtdd. Desenvolvedores
C	D1	-	D2	-	D3	-	3
O	-	-	-	D2	-	-	1
P	-	D1	-	-	-	-	1
A	-	-	-	-	-	D3	1

Fonte: Elaborada pelo autor.

Pode-se notar que houve uma distribuição mais proporcional na alocação de TP (C, O, P e A) aos especialistas (2, 3, 3, 2), do que na distribuição aos desenvolvedores (3, 1, 1, 1), na qual houve concentração no tipo de problema “Compreensão” (C). Porém, somando as quantidades de especialistas e desenvolvedores que analisaram cada TP, foi obtida a seguinte distribuição:

- 5 participantes avaliaram Compreensão (C), que conta com 16 itens de verificação (o maior conjunto do *Acc_MobileCheck*);
- 4 participantes avaliaram Operação (O), que conta com 15 itens de verificação;
- 4 participantes avaliaram Percepção (P), que conta com 12 itens de verificação e;
- 3 participantes avaliaram Adaptação (A), que conta com 4 itens de verificação.

As avaliações foram realizadas utilizando todos os itens de verificação pertencentes aos TPs do *Acc_MobileCheck*, desse modo o número de UIDPs *mobile* cobertos na validação serviram para conferir a consistência da relação de cada item atribuída a alguma(s) dessas boas práticas de desenvolvimento *mobile*.

A partir das respostas no *Acc_MobileCheck* fornecidas pelos avaliadores, nas realizações das tarefas, foi possível conferir se a relação dos UIDP *mobile* estava adequadamente construída, para que, quando a avaliação fosse realizada filtrando os itens de verificação por UIDP *mobile* nenhum item de verificação (IV) relevante fosse excluído.

No [Apêndice B](#) são relacionados os itens de verificação do *Acc_MobileCheck* com qual(is) UIDP(s) *mobile*, dentre os 12 selecionados, o item avalia. Cada tarefa definida cobriu alguns desses 12 UIDPs (vide [Tabela 14](#) e [Tabela 16](#) no [Apêndice C](#)).

Assim, apenas os itens de verificação relacionados a pelo menos um dos UIDPs cobertos na tarefa sejam aplicáveis para avaliação. Espera-se que:

Caso 1 - Itens de verificação **não cobertos** na tarefa sejam respondidos pelo avaliador com “**não se aplica**” (NA) na realização dessa tarefa, e;

Caso 2 - Itens de verificação **cobertos** na tarefa sejam respondidos pelo avaliador com algum nível de conformidade (“**Sim**” (S), “**Parcialmente**” (P) ou “**Não**” (N)) na realização dessa tarefa.

No Caso 1, uma resposta diferente de NA é considerada um problema na construção do *checklist*, apontando uma lacuna no *Acc_MobileCheck* ao ignorar pontos que deveriam ser observados para conferir a acessibilidade e usabilidade do aplicativo. Durante as avaliações pelos 8 participantes, essa situação não ocorreu, mas para uma análise mais confiável desse caso é recomendado um maior número de avaliações.

No Caso 2, uma resposta NA não é considerada um problema, apenas é esperado que não sejam frequentes pois apontam que o avaliador foi obrigado a ler itens desnecessários (que não se aplicavam àquela tarefa). Esse caso pode ocorrer pois muitas vezes UIDPs *mobile* possuem elementos em comum em suas definições, ou também podem ser implementados em uma mesma tela (vide exemplo na [Figura 28](#), da [Seção 4.3](#)). Houve 4 ocorrências de respostas NA para o Caso 2 durante as avaliações.

Uma das ocorrências do Caso 2 aconteceu na tarefa TE2 pelo E2. Essa tarefa passa por implementações dos UIDPs *mobile* LL, AS, SAB, TBIM e CO e foi analisada utilizando os 15 itens de verificação de problemas de Operação ([O_i]). A [Tabela 6](#) resume os UIDPs *mobile* que aparecem na tarefa TE2, os itens de verificação (IV) do tipo de problema analisado e as respostas do E2 a cada um dos IVs.

Na [Tabela 6](#), em cada coluna correspondente ao UIDP *mobile* encontram-se assinalados com “x” cada item de verificação que analisa algum elemento pertencente àquele UIDP *mobile*. Assim, os itens de verificação não assinalados em ao menos um dos UIDPs *mobile* apresentados não se aplicam, pela definição do *Acc_MobileCheck*, à tarefa TE2. Nesse caso, apenas o item de verificação **O2** não se aplica a nenhum UIDP *mobile* presente na tarefa TE2. Porém, durante a avaliação de TE2, o avaliador E2 respondeu “NA” ao item de verificação “[O4] – O usuário pode cancelar uma ação (um processamento) em progresso?”.

O objetivo de TE2 se refere à realização da busca, bloqueio e desbloqueio de usuário no aplicativo *Instagram*, que não se encaixa no contexto de uso do item de verificação [O4]: “– Quando a aplicação conta com a execução de alguma ação interna, que não pode ser executada instantaneamente, ou seja, demora um tempo para processamento”. Isso também foi observado pelo avaliador, cujo comentário na questão para justificar sua resposta foi: “– As ações realizadas duram menos que 5 segundos. Considerando que é solicitado confirmação para bloquear o usuário, assemelha-se a questão”

Assim, mesmo o avaliador encontrando implementações dos UIDPs *mobile* relacionados durante a tarefa, isso não a torna analisável pelo item de verificação [O4].

Tabela 6 – Avaliação da TE2 pelo E2 usando o conjunto Operação (O) do *Acc_MobileCheck*.

IV	LL	AS	SAB	TB	TBIM	CO	Resposta
[O1]	x	x		x		x	N
[O2]							NA
[O3]				x			S
[O4]						x	NA
[O5]	x				x		S
[O6]	x	x		x			N
[O7]	x	x	x	x	x	x	S
[O8]	x	x	x	x	x	x	N
[O9]			x		x	x	N
[O10]	x	x			x	x	P
[O11]	x	x	x	x	x	x	P
[O12]					x		N
[O13]	x		x	x	x	x	S
[O14]	x						N
[O15]	x			x	x		S

Fonte: Elaborada pelo autor.

Aplicação do questionário SUS

Aos avaliadores que participaram da validação do *Acc_MobileCheck* foi solicitado que respondessem 10 questões referentes ao questionário *System Usability Scale* (SUS). O SUS é um método de análise quantitativa criado por [Brooke et al. \(1996\)](#) para avaliar a usabilidade de produtos, serviços, *hardware*, *software*, *websites* e qualquer outro tipo de interface. A análise considera os critérios de usabilidade sugeridos pela ISO 9241-11 ([ISO, 2018](#)):

- Efetividade, ou seja, se os usuários conseguem completar seus objetivos;
- Eficiência, ou quanto esforço e recursos são necessários para isso e;
- Satisfação, ou seja, se a experiência foi satisfatória.

Assim, o objetivo do SUS é fornecer uma indicação geral do nível de usabilidade subjetiva de um sistema, com métricas que não exijam grande esforço e custos para coletar e analisar dados, para que possibilite ser usada para comparação com seus concorrentes ou suas outras versões ([BROOKE et al., 1996](#)). Essa indicação é obtida por meio de um questionário com 10 afirmativas, para as quais o avaliador tem possibilidade de escolher entre 5 níveis de concordância, de 1, que significa “discordo completamente” a 5, que significa “concordo completamente”. As afirmativas são as seguintes ([BROOKE et al., 1996](#), tradução nossa):

- Q1 - Eu acho que gostaria de usar este sistema frequentemente;
- Q2 - Achei o sistema desnecessariamente complexo;
- Q3 - Achei que o sistema foi fácil de utilizar;
- Q4 - Eu acho que eu precisaria do apoio de uma pessoa técnica para poder usar este sistema;
- Q5 - Eu avaliei que as várias funções neste sistema foram bem integradas;

- Q6 - Acho que havia muita inconsistência neste sistema;
 Q7 - Acredito que a maioria das pessoas aprenderia a usar este sistema rapidamente;
 Q8 - Eu achei o sistema muito complicado de usar;
 Q9 - Eu me senti muito confiante usando o sistema;
 Q10 - Eu precisei aprender muitas coisas novas antes de poder utilizar este sistema.

O questionário SUS foi desenvolvido para permitir a comparação de diferentes sistemas; desse modo, embora a escala pareça ser fácil e simples, interpretar os resultados implica nas seguintes etapas adicionais:

Etapla 1. Nas questões ímpares (1, 3, 5, 7 e 9) a nota será a nota atribuída pelo avaliador, subtraída em 1 (nota do avaliador - 1);

Etapla 2. Nas questões pares (2, 4, 6, 8 e 10) a nota será o valor 5 subtraído da nota atribuída pelo avaliador (5 - nota do avaliador);

Etapla 3. Todas as notas convertidas nas etapas anteriores devem ser somadas;

Etapla 4. O valor da soma deve ser multiplicado por 2.5, resultando na nota final SUS.

O período total de validação do *Acc_MobileCheck* foi de 12 dias (08 a 20 junho/2018). As respostas dos avaliadores para cada questão (Q1 a Q10) SUS, a nota SUS calculada a partir da nota indicada por avaliador (SUSav) analisando um tipo de problema (TP), e as médias SUS de cada avaliador (M (SUSav)) estão organizadas na [Tabela 7](#).

Tabela 7 – Valores SUS na Avaliação por desenvolvedores (D_i) e especialistas (E_i)

Questão	Desenvolvedores						Especialistas									
	D1	D1	D2	D2	D3	D3	E1	E1	E2	E2	E3	E3	E4	E4	E5	E5
Q1	5	5	5	4	5	5	3	3	4	4	4	4	5	2	5	5
Q2	2	1	1	1	1	1	5	4	1	1	2	2	2	2	1	1
Q3	5	5	5	5	5	5	2	2	5	5	4	5	4	4	5	3
Q4	1	1	2	1	1	1	2	2	1	1	2	1	1	1	1	1
Q5	5	5	4	5	5	5	3	2	5	5	5	5	4	3	5	3
Q6	2	1	1	1	1	1	1	2	1	1	1	1	1	1	2	1
Q7	4	4	4	4	5	5	1	2	3	3	4	3	3	3	3	3
Q8	2	2	1	1	1	1	4	2	1	1	1	1	2	1	1	3
Q9	4	5	3	3	5	5	5	3	5	5	4	3	4	5	2	3
Q10	4	4	1	1	1	1	1	1	2	2	1	1	1	1	1	2
SUSav	80	87,5	87,5	90	100	100	52,5	52,5	90	90	85	85	82,5	77,5	85	72,5
TP	C	P	C	O	A	C	C	O	O	P	A	P	O	A	P	C
M(SUSav)	83,75		88,75		100		52,5		90		85		80		78,75	

Fonte: Dados da pesquisa.

4.5 Análise dos resultados da validação

Segundo [Wainer et al. \(2007\)](#), a forma mais comum de utilização de dados de questionários é apenas reportá-los, descrevendo a distribuição das respostas de cada questão ou das mais relevantes. As estatísticas descritivas devem ser apropriadas. No caso de medidas categóricas, é preciso exibir a distribuição de frequência das respostas, a fim de resumir o resultado reporta-se a resposta mais frequente.

Em questionários com medidas ordinais, como no caso de medidores de usabilidade em sistemas, reporta-se a média das respostas para resumir o resultado ([WAINER et al., 2007](#)). Um exemplo é o questionário SUS, utilizado para **validar** o *Acc_MobileCheck*. Assim, podemos analisar os dados obtidos da seguinte maneira:

- A nota média SUS de avaliador (SUSav na [Figura 30](#)) mais baixa foi dada por um especialista (E1, com 52,5) e a mais alta foi dada por um desenvolvedor (D3, com 100).

Figura 30 – Nota média SUS atribuída por cada avaliador.

Fonte: Dados da pesquisa.

- Os tipos de problema (TP) com pior nota SUS de avaliador (SUSav na [Tabela 7](#)) foram avaliados pelo mesmo especialista, ambos com a mesma nota (E1, com 52,5 em Compreensão e Operação). Essa pior nota difere da média SUS dos demais avaliadores especialistas ([Tabela 8](#)) fornecidas, sobre problemas de Compreensão (62,5) e de Operação (75).

Uma outra análise sobre SUS foi proposta por ([BANGOR; KORTUM; MILLER, 2009](#)). Os autores coletaram mais de 2.300 respostas ao SUS sobre 7 diferentes tipos de UI, em pesquisas realizadas durante 10 anos. Nessas pesquisas, além das afirmativas SUS foi inserida uma 11a. que questionava qualitativamente o quão a interface é amigável ao usuário, como sinônimo de usável, em uma escala com 7 níveis. Relacionando o resultado dessa 11a. questão com a

Tabela 8 – Média dos tipos de problema por desenvolvedor e especialista

Tipo de avaliador \ TP	C	O	P	A
Desenvolvedor	89,2	90	87,5	100
Especialista	62,5	75	86,7	81,3
M(TP)	75,85	82,5	87,1	90,65

Fonte: Dados da pesquisa.

pontuação geral SUS dos sistemas analisados, foi considerado que de acordo com os critérios avaliados, o sistema possui uma usabilidade “média” quando apresenta pontuação SUS superior a 50,9 e uma **boa** usabilidade, quando a pontuação SUS é superior a 71,4.

Considerando essas notas referências, propostas por Bangor, Kortum e Miller (2009), na Figura 31 podemos ver que **todos** os tipos de problema (TPs) foram classificados acima da nota de referência para **média usabilidade** (50,9), e três deles estão classificados acima da nota para **boa usabilidade** (71,4) por ambos os grupos de avaliadores, durante a validação do *Acc_MobileCheck*.

Figura 31 – Nota média SUS dos tipos de problema por tipo de avaliador

Fonte: Dados da pesquisa.

Diante das pontuações obtidas, observa-se que a maior diferença entre as notas de desenvolvedor e especialista foi dada aos itens do *Acc_MobileCheck* relativos ao problema C, diferindo em 26,7; a menor diferença foi dada ao problema P, diferindo em apenas 0,8. No geral,

a nota média SUS (considerando todos os tipos de problema) dos especialistas foi menor do que a dos desenvolvedores (Figura 32).

Figura 32 – Nota média SUS atribuída pelos desenvolvedores e especialistas.

Fonte: Dados da pesquisa.

Brooke *et al.* (1996) destacam ainda que a nota SUS é uma medida subjetiva de usabilidade de um sistema, utilizada para obter uma visão geral de sua qualidade nesse contexto. O modo de obtenção de cada parte dessa medida depende de percepções pessoais de cada avaliador envolvido, e podem ocorrer resultados diversos. Ao considerarmos a nota média (visão geral) de uma grande amostra, possivelmente um resultado muito diferente não seria determinante. Independente do número de avaliadores, é interessante analisar o motivo de uma experiência ruim com a interface avaliada.

A seguir, alguns comentários fornecidos pelos avaliadores acerca de sua experiência com o *Acc_MobileCheck*:

E1 - “...., informações excessivas. Creio que seja melhor separar descrição do conteúdo de seu conjunto de diretrizes da ferramenta de avaliação. As opções de interação (*checklist* e caixa de texto) ficavam “espremidas” por textos acima e abaixo, os quais nem sempre são lidos. Eu preferiria que fossem telas separadas, umas para descrição das diretrizes e outra para verificação das mesmas.”

E2 - “Acredito que seria interessante poder especificar a tarefa que foi avaliada no *Acc_MobileCheck*.”

D3 - “O questionário no geral é simples de ser respondido, com perguntas diretas sobre o uso do sistema como um todo. Entretanto, as tarefas solicitadas não abrangem todas as questões.”

Analisando o comentário fornecido por E1, o argumento é que de fato, para especialistas em usabilidade e acessibilidade a grande quantidade de explicações em cada item de verificação pode ser exaustiva (dado que a plataforma utilizada para disponibilização *online* do *Acc-MobileCheck* não possibilita separação da ajuda). Esse argumento é reforçado diante da diferença observada considerando-se os perfis dos avaliadores D2 e E1. O avaliador D2 é o único não especialista em usabilidade e acessibilidade entre os desenvolvedores, e avaliou os mesmos tipos de problemas que o E1, atribuindo notas diferentes (vide [Tabela 7](#)). Entre as observações, apenas o D3 ressaltou como problema a existência de questões que não foram utilizadas na tarefa. Assim, não consideramos os UIDPs *mobile* presentes, que determinam as questões úteis, como relevantes na nota de E1.

Justificamos, portanto a importância dos textos de ajuda para avaliadores que não tenham conhecimento especializado sobre o assunto abordado nos itens de verificação.

[Winckler e Pimenta \(2002\)](#) mencionam que entre as vantagens de um *checklist* está a facilidade e baixa complexidade de interpretação das questões, permitindo sua aplicação por avaliadores com conhecimentos prévios sobre as recomendações. O *Acc-MobileCheck* é um *checklist* que visa ser um auxílio na avaliação de usabilidade e acessibilidade no processo de desenvolvimento de aplicativos móveis. Conta com perguntas simples e textos de ajuda para possíveis dúvidas acerca de cada item de verificação, para que avaliadores leigos no assunto tratado também possam utilizá-lo.

Nesse contexto, a observação sobre a experiência com o *Acc-MobileCheck* por avaliadores com diferentes níveis de conhecimento sobre usabilidade e acessibilidade mostra um progresso no cumprimento dos objetivos de sua criação.

Em resumo, segundo o critério de [Bangor, Kortum e Miller \(2009\)](#) 87,5% dos avaliadores (30) avaliaram o *checklist* como tendo uma boa usabilidade; as verificações sobre todos os tipos de problema, na média entre avaliadores, foram classificadas com boa usabilidade ([Tabela 8](#)), assim como a média geral ([Figura 32](#)). Ainda que o espaço amostral seja pequeno, composto por 8 avaliadores, as avaliações evidenciam um resultado positivo.

Finalmente, a média total de SUS obtida entre todos os avaliadores foi 84, considerada como **boa usabilidade**, com um desvio padrão de aproximadamente 16,7.

A partir dos resultados e comentários/sugestões obtidos na validação com os especialistas e desenvolvedores, foi realizada uma nova revisão detalhada de todos os pontos de verificação do *Acc-MobileCheck*, gerando uma nova versão do *checklist*, em que foram incluídos: legendas em abreviaturas, explicações para termos técnicos e específicos da área (Tecnologia Assistiva), definições para os tipos de deficiência (Visual, Motora, Intelectual e Auditiva) mais relacionadas às barreiras dos tipos de problema (C, O, P, A). Foram realizados ajustes no texto, para manter somente informações relevantes para o entendimento da questão nos textos de ajuda.

4.6 Trabalhos Relacionados

Diversos trabalhos têm discutido e proposto métodos de auxílio tanto na inspeção, quanto no desenvolvimento de UIs, em relação a usabilidade e acessibilidade. Dentre eles, alguns relacionam-se ao contexto móvel. A seguir, são descritos trabalhos da literatura, com suas propostas para avaliação e guias de desenvolvimento com usabilidade e acessibilidade. Dentre tais trabalhos, relacionados com a temática desta dissertação, alguns serviram de inspiração e outros são comparados à proposta do *Acc_MobileCheck*.

Estudos pioneiros foram realizados no sentido de explorar diferenças na interação por cegos e pessoas com visão ao usar dispositivos *touchscreen*. Kane, Wobbrock e Ladner (2011) observaram diferenças do uso de gestos para executar tarefas entre usuários cegos e com visão em dispositivos *touchscreen*. A partir do levantamento dos gestos comuns de preferência de cada grupo de participantes, foram analisados seus desempenhos e proposto um conjunto de gestos de referência, ressaltando diferenças significativas na velocidade, tamanho e forma dos gestos realizados por pessoas cegas e pessoas com visão. A partir desse levantamento, foram sugeridas 5 novas diretrizes de *design* para interfaces *touchscreen* acessíveis. O trabalho realizado por Kane, Wobbrock e Ladner (2011) foi importante para realçar as diferenças dos tipos de problemas que são tratados no *Acc_MobileCheck*, em especial quando os itens de verificação questionam as barreiras de pessoas com deficiência visual.

Piccolo, Menezes e Buccolo (2011) realizaram pesquisas acerca das soluções de acessibilidade visual para dispositivos móveis, por meio de entrevistas e observações de usuários cegos e com baixa visão utilizando dispositivos móveis com leitores de tela. Com base nos dados obtidos, foi definido um conjunto de 9 diretrizes de *design* para desenvolvimento de aplicativos móveis considerando as necessidades desse público alvo. Esse estudo possibilitou mais detalhamento sobre as reais necessidades dos usuários que usam TA em dispositivos móveis, as quais orientaram alguns dos itens de verificação do *Acc_MobileCheck*.

Investigando como as pessoas com deficiência visual interagem com dispositivos móveis, Kim *et al.* (2016) também propuseram um conjunto de implicações (no formato de heurísticas) para guiar o desenvolvimento de aplicativos móveis acessíveis. Foram realizados testes de usabilidade em um aplicativo nativo móvel com usuários em diferentes níveis de perda visual. Entre os problemas de usabilidade encontrados, foram identificadas 5 implicações relevantes no *design* de interfaces de recursos de Tecnologia Assistiva para pessoas com deficiência visual. Essas implicações não são incomuns para usuários com visão normal, mas podem ser implementadas no dispositivo móvel como configurações adicionais da UI e aplicadas a muitos tipos de diretrizes de acessibilidade.

Assim, o estudo de Kim *et al.* (2016) proporcionou um reconhecimento sobre os tipos de uso dos recursos de TA pelos usuários com deficiência visual, que guiaram as questões do *Acc_MobileCheck*, quando existe a necessidade de se considerar esse cenário.

Embora a maioria dos trabalhos encontrados estejam relacionados a deficiência visual, também existem pesquisas abrangendo outros tipos de deficiências. Nessa direção, [Mi et al. \(2014\)](#) propuseram um *checklist* de heurísticas para o desenvolvimento de interfaces acessíveis para dispositivos móveis, considerando deficiências motora e visual. O processo de desenvolvimento do *checklist* foi dividido em 3 fases: (a) elaborados 59 requisitos a partir de diretrizes e padrões de desenvolvimento existentes; (b) cada requisito passou por uma validação qualitativa com 6 usuários, sendo 3 com deficiência visual severa, 1 participante cego e 1 com deficiência severa de destreza (motora). Com base nos resultados da validação, foi definido um conjunto de recomendações para orientação do projeto de protótipos *mobile* com *touchscreen*; (c) a etapa final, de teste em protótipos, foi realizada avaliação de acessibilidade usável, conduzido com outros 9 participantes com deficiência visual severa - nesta etapa foi proposta uma tarefa simples e questionado acerca da facilidade ou dificuldade na realização da tarefa. O objetivo era utilizar os resultados para complementar as diretrizes de *design*, porém, nenhuma das diretrizes de *design* acessível foi adicionada ou modificada. As diretrizes foram modificadas como itens de verificação, permitindo resposta “sim” ou “não”.

Assim como o *Acc_MobileCheck*, os 44 itens de verificação do *checklist* de heurísticas proposto são divididos em categorias por semelhança, relativas a elementos e funções do dispositivo/interface, todos no contexto de deficiência visual e motora. [Mi et al. \(2014\)](#) distribuíram as questões em: (1) controles mecânicos, (2) tela, (3) voz e operações gerais de controle, (4) controles de feedback sonoro, (5) controles de operações de toque e (6) outros.

Destaca-se o importante empenho do trabalho de [Mi et al. \(2014\)](#), por ter incluído pessoas com deficiência para participarem da etapa de avaliação, porém uma diferença em relação a *Acc_MobileCheck* é que existem questões do *checklist* de heurísticas subjetivas, como é o caso da Questão 43 ([Mi et al., 2014](#), tradução nossa): “O *layout* do aplicativo é consistente e familiar?”. Outra diferença é que os demais tipos de deficiência não foram estudados.

[Nathan et al. \(2017\)](#) desenvolveram um modelo para avaliação de aplicativos móveis para pessoas com deficiência auditiva. Inicialmente foram realizadas entrevistas com esses usuários enquanto usavam um aplicativo móvel, para coletar informações sobre suas necessidades e desafios. Os resultados obtidos com os usuários foram cruzados com requisitos mencionados em estudos sobre deficiência auditiva encontrados em uma revisão sistemática. Assim, foi definido um modelo de qualidade com 6 dimensões de usabilidade para avaliações de aplicativos para pessoas com deficiência auditiva.

O trabalho realizado por [Nathan et al. \(2017\)](#) foi relevante por evidenciar as diferenças dos tipos de problemas que são tratados no *Acc_MobileCheck*, em especial quando os itens de verificação questionam as barreiras de pessoas com deficiência auditiva.

Pesquisas considerando apenas usabilidade no contexto móvel também foram encontradas. [Neto \(2013\)](#) propôs um conjunto de 11 heurísticas para avaliação de **usabilidade** em interfaces de aplicativos móveis e definiu um conjunto de 25 diretrizes para o desenvolvimento

dessas interfaces. Para isso, foram realizadas inspeções de usabilidade em aplicativos móveis utilizando simulação de uso em 4 aplicativos móveis. Após verificar que as heurísticas de Nielsen (1995) não eram facilmente associadas a determinadas categorias de problemas de usabilidade encontrados, um novo conjunto de heurísticas foi definido, com base nas tais categorias de problemas e nas heurísticas já existentes. Foram realizadas avaliações com ambos os conjuntos de heurísticas, e ao confrontar os problemas encontrados em cada um, verificou-se que as novas heurísticas são mais eficientes em identificar problemas de usabilidade no contexto móvel.

Neto (2013) também definiu um conjunto de diretrizes utilizando-se as documentações oficiais presentes nas páginas web das empresas detentoras dos sistemas operacionais *Android*, *Blackberry*, *iOS* e *WindowsPhone*. Cada uma delas indica o uso adequado dos componentes de interface existentes nos respectivos sistemas operacionais, a fim de que o usuário disponha do máximo de usabilidade possível (NETO, 2013). A preocupação com os diferentes sistemas operacionais dos dispositivos móveis foi também analisada durante o desenvolvimento do *Acc-MobileCheck*. Embora as 25 diretrizes para usabilidade tenham sido propostas por Neto (2013), a questão de acessibilidade não foi tratada.

Em um estudo acerca de estratégias de desenvolvimento de aplicativos móveis nativos baseado em *websites* existentes, Filho e Junior (2015) realizaram uma revisão sistemática para identificar boas práticas e experiências reportadas na literatura sobre a criação de aplicativos móveis. Os 9 artigos selecionados indicam grande preocupação em relação a usabilidade na construção de UI e ressaltam a importância de avaliações de usabilidade considerando requisitos do usuário. Como na revisão foi notada ausência de estratégias específicas para a criação de aplicativos móveis baseados em aplicativos web existentes, os autores propuseram um processo para a criação desses aplicativos, chamado Metamorfose. Um dos desafios da criação da *Acc-MobileCheck* foi adaptar recomendações e documentos voltados para web ao contexto móvel. Filho e Junior (2015) trataram das restrições intrínsecas do ambiente móvel que geralmente não são observadas quando se busca adaptar documentos criados para ambientes *desktop*. Reforçam, assim, a necessidade da criação de artefatos pensando especialmente nos elementos do contexto móvel (como tamanho reduzido de tela, interação *touchscreen*), pois não é possível obter uma compatibilidade direta sem adaptação.

Por meio de revisão sistemática, Damaceno, Braga e Mena-Chalco (2018) identificaram problemas de acessibilidade em aplicativos móveis, por pessoas com deficiência visual, a partir de 42 trabalhos da literatura. Entre seus resultados, destaca-se que mais de 70% dos trabalhos envolveram pessoas com deficiência, a maioria em testes de usabilidade. Foram identificados 68 problemas de acessibilidade classificados em 7 categorias de problemas: botões (falta de botões físicos), entrada de dados, interação baseada em gestos, leitor de tela, tamanho da tela, *feedback* ao usuário e comandos de voz. A partir dos estudos e dessa categorização, foram propostas 28 recomendações para melhoria na acessibilidade de aplicativos móveis, em especial para pessoas com deficiência visual. Embora o estudo seja uma revisão de literatura relevante, com grande

número de pesquisas analisadas, o artigo não discorre sobre as diferenças entre a acessibilidade de aplicativos nativos e web móveis.

O trabalho de [Rodrigues \(2016\)](#) investigou barreiras e dificuldades de interação enfrentadas pelos senescentes (idosos) ao interagirem com a Web, por meio de testes com usuários e revisão da literatura. A partir dos resultados obtidos, foi proposto um *checklist*, denominado Sene-Check, com 52 itens de verificação para avaliação de acessibilidade e usabilidade em aplicativo/ conteúdo web, considerando esse perfil de usuários. Os itens de verificação foram agrupados em 5 categorias de dificuldades (identificadas como os principais problemas reportados pelo público senescente na interação com a Web): leitura e compreensão de textos; navegação; realização de tarefas específicas; reconhecimento e acesso aos *links* e dificuldade na busca e localização de informações.

Assim como o *Acc_MobileCheck*, as categorias do Sene-Check são relacionadas a dificuldades dos usuários. A diferença é que as barreiras do *Acc_MobileCheck* foram baseadas nos princípios da WCAG 2.0 e as do Sene-Check são de um público específico. No Sene-check foram consideradas especialmente as deficiências relacionadas à diminuição das capacidades natural do processo de envelhecimento.

Uma iniciativa empresarial é a da *British Broadcasting Corporation* ([BBC, 2013](#)), que publicou um conjunto de 45 diretrizes (práticas agnósticas) em tecnologia para conteúdo web para dispositivos móveis, aplicativos híbridos e nativos para oferecer suporte a desenvolvedores, com acessibilidade, baseadas nos requisitos do conteúdo da empresa. Essas diretrizes foram elaboradas para uso por funcionários e fornecedores da BBC e desenvolvidas para uso com a tecnologia comumente disponível no Reino Unido. Embora seja relevante essa iniciativa, ela é particularizada para esse tipo de empresa, e possui regras específicas de uso.

Um trabalho recente, de [Carvalho et al. \(2018\)](#) relata sobre testes de usabilidade com 10 participantes, sendo 6 cegos e 4 usuários com visão normal, utilizando 4 *sites* e seus respectivos 4 aplicativos nativos móveis. A avaliação resultou em um total de 514 problemas encontrados, 105 por usuários com visão normal e 409 por usuários cegos, sendo os mais frequentes e mais graves encontrados pelo último grupo. Os resultados mostram problemas críticos que precisam ser abordados no desenvolvimento de aplicativos e *sites* móveis. Assim, o trabalho levantou pontos importantes que guiaram na criação do *Acc_MobileCheck* para conferir se as questões e definições elaboradas atendiam problemas de acessibilidade levantados por usuários com deficiência visual (alternativas para imagens, identificação de botões (elementos clicáveis), etc.).

A [Tabela 9](#) resume um conjunto de características identificadas em cada trabalho relacionado estudado. Assim, tem-se uma visão comparativa, e na última linha na tabela, contrastamos o *Acc_MobileCheck*.

Tabela 9 – Trabalhos relacionados

Ref.	Método	Base/Metodologia	Critério	Contexto
(PICCOLO; MENEZES; BUCCOLO, 2011)	9 diretrizes	testes com usuários + estudos da literatura	acessibilidade (def. visual)	desenvolv. móvel
(KIM <i>et al.</i> , 2016)	5 heurísticas	testes com usuários + estudos da literatura	acessibilidade (def. visual)	avaliação móvel
(MI <i>et al.</i> , 2014)	<i>Checklist</i> (44)	testes e validação com usuários + estudos da literatura	acessibilidade (def. visual e motora)	avaliação móvel
(NATHAN <i>et al.</i> , 2017)	modelo qualidade (6 dimensões)	entrevistas e observação de usuários + estudos da literatura	acessibilidade (def. auditiva)	avaliação móvel
(NETO, 2013)	11 heurísticas	testes com usuários + estudos da literatura + especialistas	usabilidade	avaliação móvel
(FILHO; JUNIOR, 2015)	boas práticas (4 etapas)	revisão sistemática da literatura	usabilidade/ adaptação funcionalidades	desenvolv. móvel de versão web
(DAMACENO; BRAGA; MENCHALCO, 2018)	28 recomendações	revisão sistemática da literatura	usabilidade + acessibilidade (def. visual)	desenvolv. móvel
(RODRIGUES, 2016)	<i>Checklist</i> (52)	testes com usuários + estudos da literatura	acessibilidade (idosos)	avaliação web
<i>Acc_MobileCheck</i>	<i>Checklist</i> (47)	testes e validação com especialistas + estudos da literatura + recomendações de fabricante de SO móvel	acessibilidade (def. visual, motora, intelectual e auditiva) + usabilidade	avaliação móvel

Fonte: Elaborada pelo autor.

4.7 Considerações Finais

Neste capítulo foram descritos os procedimentos para o desenvolvimento do *Acc_MobileCheck*. As decisões tomadas para compor os itens de verificação do *checklist* foram apresentadas. Assim, discutimos sobre a escolha do sistema operacional do dispositivo móvel, bem como a escolha dos UIDPs relacionados aos aplicativos móveis selecionados.

O processo de inclusão, remoção de duplicatas, validação, avaliabilidade e especificação dos itens de verificação para compor o *checklist* foi também descrito. Desde o início, quando 241 questões foram consideradas (oriundas dos diversos estudos referenciais) até a versão final do *Acc_MobileCheck* que contem 47 itens de verificação. Em especial, as etapas e versões do *checklist* foram justificadas, com as respectivas considerações sobre as categorias relacionadas

com os tipos de problemas identificados: C (Compreensão), O (Operação), P (Percepção) e A (Adaptação), que representavam as dificuldades enfrentadas por pessoas com diferentes tipos de deficiência (visual, auditiva, intelectual e motora).

A versão final do *Acc_MobileCheck* está integralmente apresentada no [Apêndice A](#).

Trabalhos da literatura, relacionados com métodos de avaliação de aplicativos móveis, quanto a acessibilidade e/ou usabilidade foram descritos. Um quadro comparativo foi apresentado mostrando os procedimentos, critérios de atender aos quesitos de acessibilidade / usabilidade e respectivas deficiências estudadas ([Tabela 9](#)).

Finalmente, a validação do *Acc_MobileCheck* foi apresentada e discutida na [Seção 4.4](#). Foi possível constatar que o *checklist* apresenta boa usabilidade, e responde às questões de pesquisa deste mestrado:

- Por meio de itens de verificação do *Acc_MobileCheck* tem-se questões de acessibilidade e usabilidade que podem ser verificadas em *Design Patterns* de aplicativos móveis.
- O *Acc_MobileCheck* é um instrumento que agiliza a identificação de barreiras de acessibilidade e problemas de usabilidade no desenvolvimento de aplicativos móveis.

CONCLUSÕES

5.1 Considerações Iniciais

Devido à praticidade em relação a computadores pessoais, dispositivos móveis são cada vez mais utilizados para realizar tarefas do cotidiano. Para atender essa demanda a geração de softwares para esses dispositivos está em constante crescimento. No cenário brasileiro, o uso de dispositivos móveis se intensificou inclusive em faixas etárias com ocorrência maior de um dos tipos de deficiência (visual, motora, mental ou intelectual) devido a idade. No âmbito mundial, uma grande parcela da população tem algum tipo de deficiência. Porém, ainda existem muitos problemas de acessibilidade e usabilidade que precisam ser resolvidos no contexto de uso de aplicativos em dispositivo móvel para incluir a diversidade de usuários.

Finalmente, destaca-se que este trabalho de mestrado respondeu às questões de pesquisa previamente definidas, conforme a seguir:

1. *Quais questões de acessibilidade e usabilidade podem ser verificadas em Design Patterns existentes para desenvolvimento de aplicativos móveis?*

Para isso, foram realizados estudos sobre usabilidade e acessibilidade e métodos de avaliação, padrões e normas, bem como boas práticas de desenvolvimento móvel com foco na investigação de padrões de projeto de interface (UIDP) dos aplicativos. Como resultado, foi elaborado um conjunto de itens de verificação para aplicativos móveis, considerando barreiras enfrentadas por usuários com deficiência visual, motora, mental e intelectual na interação com esses dispositivos.

2. *Como integrar a verificação das questões de acessibilidade e usabilidade em um instrumento que agilize a identificação de barreiras de acessibilidade e/ou problemas de usabilidade durante o desenvolvimento de aplicativos móveis?*

Na busca por responder a essa questão, a proposta deste mestrado foi por meio da elaboração de um *checklist* para apoiar avaliações de acessibilidade e usabilidade no desenvolvimento de aplicativos móveis, a fim de auxiliar na criação de softwares usáveis e acessíveis.

5.2 Principais Contribuições

A principal contribuição deste trabalho foi o levantamento de UIDPs como boa prática de projeto, atualmente utilizados em aplicativos móveis e *websites* populares, e a relação com as diretrizes e recomendações sobre acessibilidade e usabilidade, visando a elaboração do *checklist* ***Acc_MobileCheck***.

Durante a realização do mestrado, foi realizado um **mapeamento** entre as perspectivas mais abstratas (requisitos, definições e diretrizes) sobre acessibilidade e usabilidade, e as perspectivas mais práticas (UIDP) adotadas no âmbito do desenvolvimento de aplicativos móveis. Esse estudo representa uma contribuição significativa para evoluir conforme surjam novos tipos de boas práticas em desenvolvimento de aplicativos móveis. Um resultado desta pesquisa se consolidou por meio do desenvolvimento do *checklist* ***Acc_MobileCheck***, que foi submetido a uma validação.

O ***Acc_MobileCheck*** se apresenta como um instrumento de fácil adaptação ao contexto de tipos de problemas que usuários com diferentes deficiências podem enfrentar quando interagem com as UI móveis. A validação do ***Acc_MobileCheck*** mostrou que trata-se de um instrumento de fácil aplicação, por desenvolvedores e especialistas, para auxiliar tanto na avaliação de usabilidade e acessibilidade quanto no desenvolvimento de aplicativos móveis, a fim de criar softwares usáveis e acessíveis.

5.3 Limitações e trabalhos futuros

Durante o desenvolvimento deste trabalho foram observadas algumas limitações decorrentes do escopo da proposta. Dentre as limitações, tem-se: o número de voluntários avaliadores disponíveis e o período de tempo para realização das atividades, que poderiam ter sido maiores. Esses fatores podem ser motivação para continuidade da pesquisa realizada com o ***Acc_MobileCheck*** até o momento.

Como perspectiva de trabalhos futuros, consideramos ainda as possibilidades, conforme descritas a seguir.

- **Levantamento de mais UIDPs *mobile*:** nesta pesquisa, foram investigados os 10 aplicativos móveis gratuitos mais utilizados pelos usuários do sistema operacional Android em busca de UIDPs *mobile* implementados. Assim, poderiam ser investigar mais aplicativos

móveis, de modo a analisar a importância desses UIDPs *mobile*. Desse modo, mais UIDPs *mobile* poderiam ser selecionados, a fim de aumentar o escopo das análises realizadas, e melhorias poderiam ser refletidas em versões futuras do *Acc_MobileCheck*;

- **Validação por especialistas:** o número de tarefas realizadas pelos especialistas foi menor do que o número de tarefas elaboradas. Com um maior número de especialistas, poderiam ser realizadas todas as tarefas elaboradas no escopo de validação do *Acc_MobileCheck*. Além disso, seria alocado mais tempo para análise e novas tarefas também poderiam ser elaboradas;
- **Validação por desenvolvedores:** cada desenvolvedor elaborou e realizou duas tarefas em seu protótipo. Dois desenvolvedores que participam do mesmo projeto elaboraram tarefas no mesmo protótipo. Com maior tempo disponível, os desenvolvedores do mesmo protótipo poderiam realizar as tarefas do outro desenvolvedor, para possibilitar melhor comparação. Além disso, com mais desenvolvedores, poderiam ser realizadas as mesmas tarefas por mais de um desenvolvedor do mesmo protótipo;
- **Validação por usuários:** neste projeto, como a perspectiva do processo para desenvolvimento do *checklist* considerou o foco em UIDPs, que são mais conhecidos por desenvolvedores, não foi realizada validação com usuários finais de aplicativos móveis. No entanto, uma pesquisa muito importante seria a realização desse tipo de validação, que sem dúvida, traria uma confirmação realista dos problemas que foram estudados nessa dissertação;
- **Categorias (TP) da *Acc_MobileCheck* utilizadas na validação:** cada tarefa foi analisada apenas uma vez por tipo de problema. Com mais avaliadores disponíveis, seria possível realizar avaliações de uma mesma tarefa pelo mesmo grupo de itens de verificação. Assim, possibilitaria melhor análise da experiência do avaliador com o *Acc_MobileCheck*, retirando o viés de problemas em relação a tarefa que o *checklist* está avaliando ou mesmo o perfil de avaliador que o utiliza nessa avaliação;
- **Uso no desenvolvimento de aplicativo real:** o escopo inicial desse mestrado previa a utilização da abordagem proposta no desenvolvimento de aplicativo móvel. Desse modo, consideramos que sua utilização no processo de desenvolvimento real seria importante para melhorias, bem como valeria como uma validação mais confiável do *Acc_MobileCheck*. A partir disso, seria de suma importância a realização de testes com usuários com deficiência, visando analisar possíveis barreiras que não foram observadas e proporcionariam melhorias ao *Acc_MobileCheck* durante o desenvolvimento do aplicativo.

A fim de incentivar seu uso e possíveis pesquisas decorrentes do estudo realizado, o *checklist Acc_MobileCheck* está disponibilizado *online*¹ e gratuito. Desse modo, desenvolvedores e especialistas podem contribuir na evolução do projeto.

---

¹ <<http://limesurvey.labes.icmc.usp.br/index.php/783492>>

REFERÊNCIAS

ABRAN, A.; KHELIFI, A.; SURYN, W.; SEFFAH, A. Usability meanings and interpretations in iso standards. **Software Quality Journal**, Springer, v. 11, n. 4, p. 325–338, 2003. Citado na página 33.

ALEXA, I. I. **Alexa - About Us. Information, Insight, Advantage**. 2016. Disponível em: <<http://www.alexacom/about>>. Acesso em: 11/07/2016. Citado na página 199.

ALEXANDER, C.; ISHIKAWA, S.; SILVERSTEIN, M. **A pattern language**. [S.l.]: New York: Oxford University Press, 1979. Citado na página 54.

AMARAL, L. A. do. **Um ambiente para análise de resultados de avaliações de acessibilidade e usabilidade na Web**. 145 f. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação de São Carlos (ICMC/USP), Universidade de São Paulo, São Carlos - SP, 2014. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-08082014-144655/>>. Acesso em: 09 ago. 2014. Citado nas páginas 27 e 47.

ANACLETO, J. C.; PENTEADO, R. A. D.; SILVA, A. C.; NETO, A. T. Padrões de interação para projetos web uma visão geral. In: TEIXEIRA, C. A. C.; GOULARTE, R.; (ORG.). **Tópicos em Tecnologias Web e Multimídia**. [S.l.]: Ribeirão Preto: SBC, 2004. v. 1, p. 197–221. Citado na página 55.

AVAZU INC. **Global Internet Industry Research (FULL EDITION) - Brazil**. 2016. Acesso em: 06 set. 2017. Citado na página 82.

BALAGTAS-FERNANDEZ, F.; FORRAI, J.; HUSSMANN, H. Evaluation of user interface design and input methods for applications on mobile touch screen devices. In: **Human-Computer Interaction–INTERACT 2009**. [S.l.]: Springer, 2009. p. 243–246. Citado na página 52.

BANGOR, A.; KORTUM, P.; MILLER, J. Determining what individual sus scores mean: Adding an adjective rating scale. **Journal of usability studies**, Usability Professionals' Association, v. 4, n. 3, p. 114–123, 2009. Disponível em: <http://uxpajournal.org/wp-content/uploads/pdf/JUS_Bangor_May2009.pdf>. Acesso em: 22 jul. 2016. Citado nas páginas 99, 100 e 102.

BARBOSA, S.; SILVA, B. **Interação humano-computador**. [S.l.]: Elsevier Brasil, 2010. Citado nas páginas 35, 37, 46, 47, 79 e 85.

BAYLE, E.; BELLAMY, R.; CASADAY, G.; ERICKSON, T.; FINCHER, S.; GRINTER, B.; GROSS, B.; LEHDER, D.; MARMOLIN, H.; MOORE, B. *et al.* Putting it all together: towards a pattern language for interaction design: A chi 97 workshop. **ACM SIGCHI Bulletin**, ACM, v. 30, n. 1, p. 17–23, 1998. Citado na página 54.

BBC Standards and Guidelines for Mobile Accessibility. 2013. Disponível em: <<http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile>>. Acesso em: 2 ago. 2018. Citado na página 106.

- BEVAN, N.; CURSON, I. Planning and implementing user-centred design. In: ACM. **CHI'99 Extended Abstracts on Human Factors in Computing Systems**. [S.l.], 1999. p. 137–138. Citado na página 33.
- BITTAR, T. J. **Uma abordagem de apoio a boas práticas para desenvolvimento de aplicações Web acessíveis**. Tese (Doutorado) — Universidade de São Paulo, 2013. Citado na página 54.
- BLOMKVIST, S. Towards a model for bridging agile development and user-centered design. In: **Human-centered software engineering—integrating usability in the software development lifecycle**. [S.l.]: Springer, 2005. p. 219–244. Citado nas páginas 46 e 47.
- BOEHM, B. W. Verifying and validating software requirements and design specifications. **IEEE software**, IEEE Computer Society, v. 1, n. 1, p. 75, 1984. Citado na página 92.
- BRASIL. **Decreto nº 5.296 de 2 de dezembro de 2004**. Brasília, DF, 2004. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/decreto/d5296.htm>. Acesso em: 22 jul. 2016. Citado na página 45.
- _____. **Lei nº 13.146, de 6 de julho de 2015 (Estatuto da Pessoa com Deficiência)**. Brasília, DF, 2015. Disponível em: <<http://www2.camara.leg.br/legin/fed/lei/2015/lei-13146-6-julho-2015-781174-norma-actualizada-pl.pdf>>. Acesso em: 10 jul. 2018. Citado nas páginas 26, 45, 46, 86 e 141.
- BROOKE, J. *et al.* Sus - a quick and dirty usability scale. In: _____. **Usability Evaluation In Industry**. Londres: Taylor & Francis Ltd., 1996. cap. 21, p. 189–194. Citado nas páginas 97 e 101.
- CARVALHO, M. C. N.; DIAS, F. S.; REIS, A. G. S.; FREIRE, A. P. Accessibility and usability problems encountered on websites and applications in mobile devices by blind and normal-vision users. In: ACM. **Proceedings of the 33rd Annual ACM Symposium on Applied Computing**. [S.l.], 2018. p. 2022–2029. Citado na página 106.
- CHI, E. H. The false dichotomy between accessibility and usability. In: **Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility**. New York, NY, USA: ACM, 2013. (W4A '13), p. 19:1–19:2. ISBN 978-1-4503-1844-0. Citado na página 31.
- COUTINHO, E.; MOREIRA, C.; ANDRADE, R. *et al.* Requisitos de qualidade: Um padrão para identificação de requisitos não funcionais conflitantes. **MiniPlop Brasil**, São Paulo - SP, set. 2011. Disponível em: <http://www.miniplop.ita.br/ArtigosAceitos/MiniPloP_Artigo_1.pdf>. Citado na página 31.
- CRYPTZONE, I.; ICDRI; DISABILITY, I. S. **About the Cryptzone Cynthia Says Portal**. 2003. Disponível em: <<http://www.cynthiasays.com/Pages/About.aspx>>. Acesso em: 30 jul. 2016. Citado na página 46.
- DAMACENO, R. J. P.; BRAGA, J. C.; MENA-CHALCO, J. P. Mobile device accessibility for the visually impaired: problems mapping and recommendations. **Universal Access in the Information Society**, Springer, v. 17, n. 2, p. 421–435, 2018. Citado nas páginas 105 e 107.
- DARDAILLER, D. **WAI early days**. 2009. Disponível em: <<https://www.w3.org/WAI/history>>. Acesso em: 20 mai. 2018. Citado na página 38.

DEHLINGER, J.; DIXON, J. Mobile application software engineering: Challenges and research directions. In: **Workshop on Mobile Software Engineering**. [S.l.: s.n.], 2011. v. 2, p. 2–2. Citado na página 52.

DIAS, A. L.; FORTES, R. P. de M.; MASIERO, P. C. HEUA: A Heuristic Evaluation with Usability and Accessibility Requirements to Assess Web Systems. In: **Proceedings of the 11th Web for All Conference**. New York, NY, USA: ACM, 2014. (W4A '14), p. 18:1–18:4. ISBN 978-1-4503-2651-3. Disponível em: <<http://doi.acm.org/10.1145/2596695.2596706>>. Citado na página 48.

DÍAZ-BOSSINI, J.-M.; MORENO, L. Accessibility to mobile interfaces for older people. **Procedia Computer Science**, Elsevier, v. 27, p. 57–66, 2014. Citado na página 28.

DYE, S. M.; SCARFONE, K. A standard for developing secure mobile applications. **Computer Standards & Interfaces**, Elsevier, v. 36, n. 3, p. 524–530, 2014. Citado na página 52.

FILHO, I. d. M. B.; JUNIOR, G. S. A. Development of mobile applications from existing web-based enterprise systems. **International Journal of Web Information Systems**, Emerald Group Publishing Limited, v. 11, n. 2, p. 162–182, 2015. Citado nas páginas 105 e 107.

FINCHER, S.; FINLAY, J.; GREENE, S.; JONES, L.; MATCHEN, P.; THOMAS, J.; MOLINA, P. J. Perspectives on hci patterns: concepts and tools. In: ACM. **CHI'03 extended abstracts on Human factors in computing systems**. [S.l.], 2003. p. 1044–1045. Citado na página 55.

FLING, B. **Mobile Design and Development: Practical concepts and techniques for creating mobile sites and web apps**. [S.l.]: "O'Reilly Media, Inc.", 2009. Citado na página 53.

FOLMER, E. **Interaction Design Foundation: Interaction design patterns**. 2002. Disponível em: <<https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/interaction-design-patterns>>. Citado nas páginas 54, 55 e 56.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. Padrões de projeto. **Soluções reutilizáveis de software**, 1994. Citado nas páginas 54, 55 e 59.

GEREA, C.; HERSKOVIC, V. Measuring user experience in latin america: An exploratory survey. In: **Proceedings of the Latin American Conference on Human Computer Interaction**. New York, NY, USA: ACM, 2015. (CLIHC '15), p. 19:1–19:4. ISBN 978-1-4503-3960-5. Disponível em: <<http://doi.acm.org/10.1145/2824893.2824914>>. Citado na página 37.

GILBERTSON, T. D.; MACHIN, C. H. Guidelines, icons and marketable skills: an accessibility evaluation of 100 web development company homepages. In: ACM. **Proceedings of the international cross-disciplinary conference on web accessibility**. [S.l.], 2012. p. 17. Citado na página 28.

GOOGLE, I. **Material Design**. 2014. Disponível em: <<https://material.google.com/#>>. Acesso em: 31 jul. 2017. Citado nas páginas 59 e 60.

_____. **GMail**. 2017. Disponível em: <<https://play.google.com/store/apps/details?id=com.whatsapp&hl=pt>>. Acesso em: 11 ago. 2017. Citado na página 184.

- _____. **Google Search Bar - Google App**. 2017. Disponível em: <<https://play.google.com/store/apps/details?id=com.google.android.googlequicksearchbox>>. Acesso em: 11 ago. 2017. Citado na página 188.
- _____. **Material Design**. 2018a. Disponível em: <<https://material.io/design/>>. Acesso em: 20 jun. 2018. Citado na página 60.
- GOOGLE INC., ANDROID DEVELOPERS. **API Guides - Accessibility**. 2017. Disponível em: <<https://developer.android.com/guide/topics/ui/accessibility/index.html>>. Acesso em: 18 mar. 2017. Citado nas páginas 61 e 84.
- _____. **Componentes da Arquitetura do Android - Documentos e ferramentas de apoio ao desenvolvedor**. 2017a. Disponível em: <<https://developer.android.com/develop/index.html>>. Acesso em: 18 mar. 2017. Citado na página 62.
- _____. **Material Design para Android**. 2018. Disponível em: <<https://developer.android.com/design/material/>>. Acesso em: 25 jul. 2018. Citado na página 59.
- GSA'S OFFICE OF GOVERNMENT-WIDE POLICY. **WCAG 2.0 Conformance**. 2015. Disponível em: <<https://www.section508.gov/content/build/website-accessibility-improvement/WCAG-conformance>>. Acesso em: 30 nov. 2016. Citado na página 44.
- GSMA INTELLIGENCE. **Global Mobile Trends 2017**. 2017. Disponível em: <<https://www.gsmainelligence.com/research/?file=3df1b7d57b1e63a0cbc3d585feb82dc2&download>>. Acesso em: 06 set. 2017. Citado nas páginas 66 e 69.
- GUERRA, A. C.; COLOMBO, R. M. T. Tecnologia da informação: qualidade de produto de software. Brasília: Ministério da Ciência e Tecnologia – Secretaria Política de Informática, p. 432, 2009. Citado na página 32.
- IBGE, D. d. p. e. c. d. t. e. r. I. B. d. G. e. E. **Pesquisa nacional por amostra de domicílios 2005-2011**. [s.n.], 2012. Disponível em: <ftp://ftp.ibge.gov.br/Acesso_a_internet_e_posse_celular/2011/PNAD_Inter_2011.pdf>. Citado na página 37.
- INSTAGRAM. **Instagram**. 2017. Disponível em: <<https://play.google.com/store/apps/details?id=com.instagram.android&hl=pt>>. Acesso em: 11 ago. 2017. Citado na página 178.
- INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Pesquisa nacional por amostra de domicílios 2005-2014**. 2015. Disponível em: <<https://biblioteca.ibge.gov.br/visualizacao/livros/liv95753.pdf>>. Citado nas páginas 25 e 27.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 9241-210: 2010 Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems**. 2010. Citado nas páginas 34 e 37.
- _____. **Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuARE) — System and software quality models**. 2011. Disponível em: <<https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>>. Acesso em: 13 nov. 2016. Citado nas páginas 34, 35, 84 e 85.
- _____. **ISO/IEC 25000 SQuARE - System and Software Quality Requirements and Evaluation**. 2014. Disponível em: <<http://iso25000.com/index.php/en/iso-25000-standards>>. Acesso em: 22 jun. 2016. Citado nas páginas 32 e 35.

_____. **ISO 9241-11: 2018 Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts**. 2018. Citado na página 97.

INTERNATIONAL TELECOMMUNICATION UNION. **ICT Facts and Figures - The world in 2015**. 2015. Disponível em: <<http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf>>. Acesso em: 2 ago. 2016. Citado na página 25.

_____. **ICT Facts and Figures 2016**. 2016. Disponível em: <<http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2016.pdf>>. Acesso em: 1 ago. 2016. Citado na página 26.

IVORY, M. Y.; HEARST, M. A. The state of the art in automating usability evaluation of user interfaces. **ACM Computing Surveys (CSUR)**, ACM, v. 33, n. 4, p. 470–516, 2001. Disponível em: <<http://webtango.berkeley.edu/papers/ue-survey/p470-ivory.pdf>>. Acesso em: 10 jul. 2018. Citado na página 46.

KANE, S. K.; WOBBROCK, J. O.; LADNER, R. E. Usable gestures for blind people: understanding preference and performance. In: ACM. **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. [S.l.], 2011. p. 413–422. Citado na página 103.

KIM, H. K.; HAN, S. H.; PARK, J.; PARK, J. The interaction experiences of visually impaired people with assistive technology: A case study of smartphones. **International Journal of Industrial Ergonomics**, Elsevier, v. 55, p. 22–33, 2016. Citado nas páginas 103 e 107.

LARA, S. M. A. de. **Mecanismos de apoio para usabilidade e acessibilidade na interação de adultos mais velhos na Web**. 278 f. Tese (Doutorado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação de São Carlos (ICMC/USP), Universidade de São Paulo, São Carlos - SP, 2013. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-14022013-163940/>>. Acesso em: 22 abr. 2016. Citado nas páginas 31, 33 e 45.

LAZAR, J.; DUDLEY-SPONAUGLE, A.; GREENIDGE, K.-D. Improving web accessibility: a study of webmaster perceptions. **Computers in Human Behavior**, Elsevier, v. 20, n. 2, p. 269–288, 2004. Citado na página 28.

LAZAR, J.; FENG, J. H.; HOCHHEISER, H. **Research methods in human-computer interaction**. 1. ed. United Kingdom: John Wiley & Sons, 2010. 426 p. ISBN 978-0-470-72337-1. Citado na página 48.

LEITNER, M. Supporting design for mobile people: A materialistic approach. In: **CHI '12 Extended Abstracts on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2012. (CHI EA '12), p. 935–938. ISBN 978-1-4503-1016-1. Disponível em: <<http://doi.acm.org/10.1145/2212776.2212869>>. Citado na página 58.

LUCINDA, M. A. **Qualidade-Fundamentos e Práticas**. [S.l.]: Brasport, 2010. Citado nas páginas 35, 54 e 79.

MI, N.; CAVUOTO, L. A.; BENSON, K.; SMITH-JACKSON, T.; NUSSBAUM, M. A. A heuristic checklist for an accessible smartphone interface design. **Universal access in the information society**, Springer, v. 13, n. 4, p. 351–365, 2014. Citado nas páginas 104 e 107.

MINISTÉRIO DO PLANEJAMENTO, ORÇAMENTO E GESTÃO, SECRETARIA DE LOGÍSTICA E TECNOLOGIA DA INFORMAÇÃO. **eMAG - Modelo de Acessibilidade de Governo Eletrônico**. 2014. Disponível em: <<http://emag.governoeletronico.gov.br/>>. Acesso em: 02 jul. 2018. Citado nas páginas 45, 46, 47 e 48.

MINISTÉRIO DO PLANEJAMENTO, DESENVOLVIMENTO E GESTÃO, PORTAL DO SOFTWARE PÚBLICO BRASILEIRO. **Avaliador e Simulador de Acessibilidade em Sítios**. 2016. Disponível em: <<https://softwarepublico.gov.br/social/ases/>>. Acesso em: 29 jul. 2016. Citado na página 46.

MIÑÓN, R.; MORENO, L.; MARTÍNEZ, P.; ABASCAL, J. An approach to the integration of accessibility requirements into a user interface development method. **Science of Computer Programming**, v. 86, p. 58 – 73, 2014. ISSN 0167-6423. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167642313001032>>. Citado na página 79.

MONTEIRO, J. D. A. **Desenvolvimento de aplicações multi-plataformas para dispositivos móveis**. 136 f. Tese (Mestrado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação de São Carlos (ICMC/USP), Universidade de São Paulo, São Carlos - SP, 2006. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-24072006-103336/publico/dissertacao_JaneMonteiro.pdf>. Acesso em: 30 jul. 2016. Citado na página 26.

MONTOLIU, R.; GATICA-PEREZ, D. Discovering human places of interest from multimodal mobile phone data. In: ACM. **Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia**. [S.l.], 2010. p. 12. Citado na página 26.

NATHAN, S. S.; HUSSAIN, A.; HASHIM, N. L.; OMAR, M. A. Dimensions for hearing-impaired mobile application usability model. In: AIP PUBLISHING. **AIP Conference Proceedings**. [S.l.], 2017. v. 1891, n. 1, p. 020108. Citado nas páginas 104 e 107.

NETO, O. J. M. **Usabilidade da interface de dispositivos móveis: heurísticas e diretrizes para o design**. 117 f. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação de São Carlos, Universidade de São Paulo, São Carlos - SP, 2013. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-07012014-110754/publico/dissertacaoOlibario.pdf>>. Acesso em: 21 jul. 2016. Citado nas páginas 26, 104, 105 e 107.

NIELSEN, J. Heuristic evaluation. In: NIELSEN, J AND MACK, R. **Usability inspection methods**. New York, NY, USA: John Wiley & Sons, 1994. p. 25–62. Citado na página 36.

_____. Heuristic evaluation. **Usability inspection methods**, v. 17, n. 1, p. 25–62, 1994. Citado na página 48.

_____. 10 usability heuristics for user interface design. **Nielsen Norman Group**, v. 1, n. 1, 1995. Citado nas páginas 36, 84 e 105.

_____. Fat-finger problem makes mobile sites superior. **Nielsen Norman Group**, 2011. Disponível em: <<https://www.nngroup.com/articles/kindle-fire-usability-findings/>>. Citado na página 67.

_____. Usability 101: Introduction to usability. **Nielsen Norman Group**, p. 1–10, 2012. Disponível em: <<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>>. Citado na página 37.

NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: ACM. **Proceedings of the SIGCHI conference on Human factors in computing systems**. [S.l.], 1990. p. 249–256. Citado nas páginas 35 e 48.

NUDELMAN, G. **Android design patterns: interaction design solutions for developers**. [S.l.]: John Wiley & Sons, 2013. Citado nas páginas 58, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 83, 84, 182, 184, 186, 187, 196 e 203.

OLIVEIRA, L. M. B. Cartilha do censo 2010–pessoas com deficiência. **Brasília: Secretaria de Direitos Humanos da Presidência da República (SDH/PR), Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência (SNPD), Coordenação-Geral do Sistema de Informações sobre a Pessoa com Deficiência**, 2012. Citado nas páginas 27 e 28.

OULASVIRTA, A.; WAHLSTRÖM, M.; ERICSSON, K. A. What does it mean to be good at using a mobile device? an investigation of three levels of experience and skill. **International journal of human-computer studies**, Elsevier, v. 69, n. 3, p. 155–169, 2011. Citado na página 52.

PAZ, F.; POW-SANG, J. A. A systematic mapping review of usability evaluation methods for software development process. **Int. J. of Software Engineering and Its Applications**, v. 10, n. 1, p. 165–178, 2016. Disponível em: <http://www.sersc.org/journals/IJSEIA/vol10_no1_2016/16.pdf>. Citado na página 47.

PETTEY, C.; MEULEN, R. **Gartner says more than 1 billion PCs in use worldwide and headed to 2 billion units by 2014**. Gartner Group, 2008. Disponível em: <<http://www.gartner.com/it/page.jsp?id=703807>>. Acesso em: 21 abr. 2016. Citado na página 51.

PICCOLO, L. S. G.; MENEZES, E. M. D.; BUCCOLO, B. D. C. Developing an accessible interaction model for touch screen mobile devices: preliminary results. In: BRAZILIAN COMPUTER SOCIETY. **Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction**. [S.l.], 2011. p. 222–226. Citado nas páginas 103 e 107.

POORE-PARISEAU, C. Online learning: Designing for all users. **J. Usability Studies**, Usability Professionals' Association, Bloomington, IL, v. 5, n. 4, p. 15:147–15:156, ago. 2010. ISSN 1931-3357. Disponível em: <<http://dl.acm.org/citation.cfm?id=2019116.2019119>>. Citado na página 26.

POWER, C.; PETRIE, H. Accessibility in non-professional web authoring tools: a missed web 2.0 opportunity? In: ACM. **Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)**. [S.l.], 2007. p. 116–119. Citado na página 27.

RAGGETT, D.; HORS, A. L.; JACOBS, I. *et al.* Html 4.01 specification. **W3C recommendation**, v. 24, 1999. Disponível em: <<https://www.w3.org/TR/html401/>>. Acesso em: 16 mai. 2018. Citado na página 38.

ROCHA, A. R. C. d.; MALDONADO, J. C.; WEBER, K. C. **Qualidade de Software: Teoria e Prática**. [S.l.]: São Paulo: Prentice Hall, 2001. 303 p. ISBN 85.346.1322-2. Citado na página 32.

ROCHA, H. da; BARANAUSKAS, M. **Design e avaliação de interfaces humano-computador**. Unicamp, 2003. 244p. p. ISBN 9788588833043. Disponível em: <<http://www.nied.unicamp.br/?q=content/download-heloisa-cecilia-2003>>. Citado na página 48.

RODRIGUES, S. S. **Avaliação de acessibilidade e usabilidade na Web: um apoio com foco nos usuários senescentes**. 243 f. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação de São Carlos, Universidade de São Paulo, São Carlos - SP, 2016. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-17012017-095657/pt-br.php>>. Acesso em: 11 jul. 2018. Citado nas páginas 106 e 107.

SEFFAH, A.; METZKER, E. The obstacles and myths of usability and software engineering. **Communications of the ACM**, ACM, v. 47, n. 12, p. 71–76, 2004. Citado na página 33.

TANGARIFE, T. M.; MONT'ALVÃO, C. Acessibilidade web: Um estudo exploratório do conhecimento do desenvolvedor web brasileiro. **ATIID'05 - III Seminário e II Oficina "Acessibilidade TI e Inclusão Digital"**, São Paulo - SP, Setembro 2005. Disponível em: <http://www.prodam.sp.gov.br/multimidia/midia/cd_atiid/conteudo/atiid2005/mr2/01/acessibilidadeweb.doc>. Citado na página 44.

THATCHER, J.; WADDELL, C.; REGAN, B.; HENRY, S. L.; BURKS, M. R.; URBAN, M. D.; BOHMAN, P. **Constructing accessible web sites**. [S.l.]: Glasshaus Birmingham, 2003. v. 34. 415 p. ISBN 978-159059-148-2. Citado na página 31.

TOXBOE, A.; HOBSON, N. *Input Feedback*. **User Interface Design Pattern Library**, 2015. Disponível em: <ui-patterns.com/patterns/InputFeedback>. Citado nas páginas 56 e 58.

_____. *User Interface Design Patterns*. **User Interface Design Pattern Library**, 2015. Disponível em: <ui-patterns.com/patterns>. Citado na página 200.

TSAKNAKI, V.; FERNAEUS, Y. Expanding on wabi-sabi as a design resource in hci. In: **Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2016. (CHI '16), p. 5970–5983. ISBN 978-1-4503-3362-7. Disponível em: <<http://doi.acm.org/10.1145/2858036.2858459>>. Citado na página 58.

UNFPA-FUNDO, D. P. D. N. Unidas. **Relatório sobre a situação da população mundial**, 2015. Citado na página 37.

U.S. DEPARTMENT OF HOMELAND SECURITY. **DHS Section 508 Compliance Test Process for Applications**. 2015. Disponível em: <https://www.dhs.gov/sites/default/files/publications/DHS_Section_508_Compliance_Test_Process_for_Applications_0.pdf>. Acesso em: 2 jun. 2016. Citado na página 44.

VIGO, M.; BROWN, J.; CONWAY, V. Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests. In: **Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility**. New York, NY, USA: ACM, 2013. p. 1:1–1:10. Disponível em: <<http://doi.acm.org/10.1145/2461121.2461124>>. Citado na página 47.

WAINER, J. *et al.* Métodos de pesquisa quantitativa e qualitativa para a ciência da computação. **Atualização em informática**, Sociedade Brasileira de Computação/Editora PUC Rio Rio de Janeiro, v. 1, p. 221–262, 2007. Citado na página 99.

WASSERMAN, A. I. Software engineering issues for mobile application development. In: **ACM. Proceedings of the FSE/SDP workshop on Future of software engineering research**. [S.l.], 2010. p. 397–400. Citado nas páginas 52 e 53.

WELIE, M. V.; VEER, G. C. Van der. Pattern languages in interaction design: Structure and organization. In: **Proceedings of interact**. [S.l.: s.n.], 2003. v. 3, p. 1–5. Citado nas páginas 55 e 56.

WHATSAPP, I. **WhatsApp Messenger**. 2017. Disponível em: <<https://play.google.com/store/apps/details?id=com.google.android.gm>>. Acesso em: 11 ago. 2017. Citado na página 181.

WHO Press. World Report on Disability 2011. Geneva: World Health Organization, 2011. Citado na página 27.

WINCKLER, M.; PIMENTA, M. S. Avaliação de usabilidade de sites web. **Escola de Informática da SBC SUL (ERI 2002) ed. Porto Alegre: Sociedade Brasileira de Computação (SBC)**, v. 1, p. 85–137, 2002. Citado nas páginas 48, 49, 80, 81 e 102.

WORLD WIDE WEB CONSORTIUM. **Web Content Accessibility Guidelines 2.0**. 2008a. Disponível em: <<http://www.w3.org/TR/WCAG20/>>. Acesso em: 03 mai. 2016. Citado nas páginas 39 e 84.

_____. **Web Content Accessibility Guidelines 2.0 - Tradução Autorizada em Português Europeu**. 2014. Disponível em: <<https://www.w3.org/Translations/WCAG20-pt-PT/>>. Acesso em: 01 jul. 2016. Citado na página 40.

_____. **User Agent Accessibility Guidelines (UAAG) 2.0**. 2015a. Disponível em: <<http://www.w3.org/TR/UAAG20/#intro-wcag>>. Acesso em: 9 jul. 2018. Citado na página 38.

WORLD WIDE WEB CONSORTIUM BRASIL. **Mobile Web Best Practices Working Group**. 2005. Disponível em: <<https://www.w3.org/2005/MWI/BPWG/>>. Acesso em: 2 jun. 2016. Citado na página 40.

_____. **Mobile Web Best Practices 1 (MWBP)**. 2008b. Disponível em: <<https://www.w3.org/TR/mobile-bp/>>. Acesso em: 2 jun. 2016. Citado nas páginas 42 e 84.

_____. **Extrutura de Melhores Práticas e Statements**. 2008c. Disponível em: <<https://www.w3.org/TR/mobile-bp/#bpstructure>>. Acesso em: 2 jun. 2016. Citado na página 42.

_____. **Mobile Web Initiative (MWI)**. 2009. Disponível em: <<https://www.w3.org/Mobile/About>>. Acesso em: 2 jun. 2016. Citado na página 40.

_____. **Mobile Web Application Best Practices (MWABP)**. 2010. Disponível em: <<http://www.w3.org/TR/mwabp/>>. Acesso em: 20 mai. 2016. Citado nas páginas 42, 43 e 84.

_____. **Shared Techniques wiki for the W3C Mobile Web Initiative Best Practices**. 2010b. Disponível em: <<https://www.w3.org/2005/MWI/BPWG/techs/TechniquesIntro>>. Acesso em: 25 mai. 2016. Citado na página 41.

_____. **Conhecendo o W3C - História**. 2011. Disponível em: <<http://www.w3c.br/Sobre/ConhecendoW3C>>. Acesso em: 24 abr. 2016. Citado na página 38.

_____. **Authoring Tool Accessibility Guidelines (ATAG 2.0)**. 2015b. Disponível em: <<https://www.w3.org/TR/ATAG20/>>. Acesso em: 24 abr. 2016. Citado na página 38.

_____. **Mobile Web Best Practices Working Group**. 2017. Disponível em: <<https://www.w3.org/WAI/standards-guidelines/mobile/>>. Acesso em: 2 jun. 2018. Citado nas páginas 38 e 39.

ITENS DE VERIFICAÇÃO DA *ACC_MOBILECHECK*

O *checklist Acc_MobileCheck* completo, com os filtros de avaliação por tipo de problema de acessibilidade e por *UIDP mobile*, encontra-se disponível em: <<http://limesurvey.labes.icmc.usp.br/index.php/783492>>.

O *Acc_MobileCheck* consiste em um conjunto de 47 itens de verificação, elaborados no formato de perguntas objetivas. Essas perguntas, formuladas a partir de estudos em documentos relacionados a recomendações e critérios de usabilidade, acessibilidade e boas práticas de desenvolvimento *mobile*.

Para facilitar o uso do *Acc_MobileCheck* os itens de verificação foram agrupados por:

- **Tipo de problema relacionado** - podendo ser Problemas de Compreensão, Problemas de Percepção, Problemas de Operação ou Problemas de Adaptação.
- **User Interface Design Pattern (UIDP) mobile relacionado à pergunta** - entre os seguintes: *List of Links*, *Voice Search*, *Auto-Complete*, *Auto-Suggest*, *Search From Action Bar*, *Tabs*, *Multiple Select*, *Textbox with input Mask*, *Inline Error Message*, *Callback Validation*, *Cancel/Ok* e *Carousel*.

O avaliador pode escolher combinações entre *UIDP(s) mobile* e *Tipo(s) de problema*. Por exemplo, é possível analisar apenas Problemas de Operação relacionados ao *UIDP Cancel/Ok*.

Para cada questão, as opções de respostas correspondem aos seguintes níveis de conformidade:

- **“Sim”** - está totalmente em conformidade;
- **“Parcialmente”** - atende ao menos 50% dos critérios de usabilidade/acessibilidade questionados;
- **“Não”** - atende menos de 50% critérios de usabilidade/acessibilidade questionados no item;
- **“Não se aplica”** - não é possível avaliar os critérios de usabilidade/acessibilidade do item naquela tarefa.

Sugestões e dúvidas, por favor entre em contato (marisa.batista@usp.br).

Problemas de Compreensão

São relacionados a possíveis dificuldades do usuário:

- Em entender para o que serve um item clicável ou qualquer elemento não textual.
- Em entender onde ele está localizado atualmente no aplicativo;
- Em entender o vocabulário e/ou idioma utilizado no aplicativo;
- Para entender as mensagens do aplicativo;
- Em saber quais alternativas existem para corrigir um erro;
- Para aprender como utilizar o aplicativo;
- Em entender as consequências de uma determinada interação;
- Para inserir um texto correto para a busca desejada e compreender o contexto do resultado obtido.

[C1] Existe uma descrição textual sucinta para todos os elementos clicáveis?

Contexto: Quando o aplicativo apresenta imagens ou figuras ou mesmo ícones clicáveis.

Exemplo: Um botão clicável dentro da aplicação que não possua descrição textual de seu funcionamento.

Motivação: Auxiliar usuários de leitor de tela a compreenderem o conteúdo da aplicação, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de entendimento de ícones e símbolos de interação, a fim de atender necessidades de usuários com deficiência INTELECTUAL.

Recomendação: A aplicação deve fornecer alternativas curtas de texto para todo conteúdo não textual que expressem seu propósito desse conteúdo [2] e, também alternativas curtas com uma breve descrição do conteúdo [3] com contraste suficiente entre elementos (para isso o desenvolvedor pode utilizar a ferramenta *Color Tool*, do *Material Design*) [6][7] e tamanho de fonte e fontes legíveis. Todo campo de texto clicável deve ter um rótulo que exiba o tipo de entrada que o campo requer [1], os rótulos em menus, devem refletir de forma concisa e precisa os itens dentro do menu. As barras de menus normalmente usam palavras únicas como rótulos, como "arquivo", "formato" e "editar". Outros contextos podem exigir rótulos mais longos [5]. O mesmo para botões e outros elementos clicáveis da aplicação.

Quando o conteúdo não textual é muito pequeno ou não existe espaço para colocação de texto, existem recursos no Android que podem ser utilizados para apresentar essa informação. Por exemplo, surge um texto com uma explicação curta e de fácil entendimento a respeito do propósito do elemento clicável, incluído dentro de um frame com bordas limitantes, sobrepondo parte da tela com destaque e indicando claramente com um conector ou linha que se refere ao respectivo item clicável. Essa ação pode ocorrer com um toque longo simples ou por uso de Gestos [8]. Todos elementos da aplicação devem ser intuitivos e fornecer meios do usuário compreender seu propósito.

Fontes:

[1] "Label - Layout". *Material Design*, Google Inc., 2014. URL: <https://material.io/guidelines/components/text-fields.html#text-fields-principles>

[2] Success Criterion "1.1.1 Non-Text Content". Technique: "G94: Providing short text alternative for non-text content that serves the same purpose and presents the same information as the non-text content". WCAG 2.0, W3C, 2008. URL: <https://www.w3.org/TR/WCAG20-TECHS/G94.html>

[3] Success Criterion "1.1.1 Non-Text Content". Technique: "G95: Providing short text alternatives that provide a brief description of the non-text content". WCAG 2.0, W3C, 2008. URL: <https://www.w3.org/TR/WCAG20-TECHS/G95.html>

[4] Success Criterion "2.4.6 Headings and Labels". WCAG 2.0, W3C, 2008. URL: <https://www.w3.org/WAI/WCAG21/quickref/#headings-and-labels>

[5] "Menu labels - Components-Menus". *Material Design*, Google Inc., 2014. URL: <https://material.io/guidelines/components/menus.html#menus-usage>

[6] "Style- Color". *Material Design*, Google Inc., 2014. URL: <https://material.io/guidelines/style/color.html#>

[7] "Color Tool". *Material Design*, Google Inc., 2014. URL: <https://material.io/color/#!>>

[8] "Gestures". *Material Design*, Google Inc., 2014. URL: <https://material.io/guidelines/patterns/Gestures.html#>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C2] Existe descrição para todos os elementos não textuais que possibilite leitura por TA¹?

Contexto: Quando o aplicativo possui elementos gráficos (imagens, figuras, ícones, gráficos, etc.) que precisam estar descritos para acesso por usuários que utilizam TA.

Exemplo: Uma aplicação possui uma imagem e o leitor de tela não consegue descrever (falar) o que a imagem representa.

Motivação: Auxiliar usuários de leitor de tela a compreenderem o conteúdo da aplicação, a fim de atender as necessidades de usuários com deficiência VISUAL.

Recomendação: O suporte a tecnologias assistivas é específico da plataforma Android, mas é preciso que o desenvolvedor assegure que sua aplicação funcionará com métodos de entrada por toque e tipos de teclado. Por exemplo, conferir se a aplicação permitirá ser utilizada com o leitor de tela do *TalkBack* [3][5].

É importante fornecer descrições úteis e significativas que explicam o significado e a finalidade de cada elemento interativo para os usuários. Esses rótulos permitem que leitores de tela, como o *TalkBack*, expliquem em corretamente a função de um controle específico para os usuários que dependem desses serviços [7]. Esses rótulos são chamados por textos de acessibilidade, que quando são escritos de modo claro e significativo são uma das principais formas de tornar as interfaces de usuário mais acessíveis. Usuários com pouca ou nenhuma visão se beneficiam de descrições verbais explícitas. O texto de acessibilidade refere-se ao texto usado pelo *software* de acessibilidade do leitor de tela, como o *TalkBack* no Android, o *VoiceOver* no iOS e o *JAWS* no computador. Os leitores de tela lêem todo o texto na tela em voz alta, incluindo texto alternativo visível e não visível [5].

As interfaces de toque devem maximizar a compatibilidade com tecnologias assistivas permitindo que leitores de tela e outros dispositivos de tecnologia assistiva leiam todas as partes da sua interface, e o texto lido em voz alta deve ser significativo e útil [5][4]. Para que isso ocorra todos os componentes da interface do usuário, seu nome e a função, estados, propriedades e valores precisam ser determinados por programação [6] (para permitir que leitores de tela leiam nomes de componentes, utilize o “contentDescriptionattribute” nos itens clicáveis, como botões, ícones, e *Tabs* que contenham ícones com texto não visível [5]), de modo que não apenas o que está escrito na tela é lido pelo assistente de voz, mas seu significado. O leitor de tela precisa saber o que é clicável na UI.

Fontes:

[1] Success Criterion “2.4.6 Headings and Labels”. WCAG 2.0, World Wide Web Consortium, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#headings-and-labels>>

[2] Success Criterion “2.4.8 Location”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#location>>

[3] Guideline “1.4 Compatible”. WCAG 2.0, World Wide Web Consortium, 2008. <<https://www.w3.org/WAI/WCAG20/quickref/#ensure-compat>>

[4] “Screen Readers - Usability- Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#accessibility-hierarchy-focus>>

[5] “Usability- Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

[6] Success Criterion “4.1.2 Name, Role, Value”. WCAG 2.0, World Wide Web Consortium, 2008. <<https://www.w3.org/WAI/WCAG20/quickref/?showtechniques=111%2C411%2C412#ensure-compat-rsv>>

[7] “Making aplicativos More Accessible: Labeling UI Elements”. Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <<https://developer.android.com/guide/topics/ui/accessibility/aplicativos.html#label-elements>>

[8] BRASIL. SDHPR “Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência”. SNPD. 2009. <<http://www.pessoacomdeficiencia.gov.br/aplicativo/publicacoes/tecnologia-assistiva>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

¹TA ou Tecnologia Assistiva se define por “(..) uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social” [8].

[C3] São fornecidos rótulos e títulos de seção descritivos em todas as telas da aplicação?

Contexto: Quando o aplicativo possui diversas telas com conteúdos e funcionalidades variadas.

Exemplo: Quando a aplicação não exibe no topo da tela da aplicação um rótulo descritivo acerca do que existe na página, e não exibe títulos de seção para subgrupos do assunto principal que identifiquem seu conteúdo.

Motivação: Auxiliar usuários com deficiência VISUAL que utilizam leitor de tela, a compreender o contexto da aplicação, a fim de entender o *status* de execução de uma determinada tarefa da aplicação. Auxiliar usuários com dificuldades de entendimento do *status* atual de interação com a aplicação, a fim de atender necessidades de usuários com deficiência INTELECTUAL.

Recomendação: O aplicativo deve fornecer rótulos e títulos que precisam identificar o tópico e propósito da tela atual, serem curtos e fazerem sentido quando lidos fora de contexto, como em uma lista de links que redirecionam para a tela/seção com esse título ou rótulo [1][3][4].

Controles de navegação devem ser fáceis de localizar e escritos com clareza [5]. Subtítulos (ou títulos de seção) podem ser exibidos em linha com blocos ou associados ao conteúdo. Eles estão normalmente relacionados a critérios de filtragem ou classificação e permanecem fixos na parte superior da tela após a rolagem, até serem empurrados para fora da tela pelo próximo subtítulo [7].

O uso de rótulos e títulos de seção possibilita o usuário identificar mais rapidamente o assunto do qual se trata a tela atual [6] sem a necessidade de ler ou interpretar o conteúdo da página onde está. Desse modo também auxilia o usuário a encontrar um determinado conteúdo ou funcionalidade na aplicação, se orienta sobre eles e a navegar na aplicação[2]. Os rótulos e títulos de seção precisam estar com contraste entre cor de fonte e plano de fundo suficientes e de tamanho de fonte e fonte legíveis e perceptíveis [6].

Fontes:

[1] Success Criterion “2.4.6 Headings and Labels”. WCAG 2.0, World Wide Web Consortium, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#headings-and-labels>>

[2] Success Criterion “2.4.8 Location”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#location>>

[3] Success Criterion “2.4.2 Page Titled”. Technique “G88: Providing descriptive titles for *Web* pages”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/G88.html>>

[4] Success Criterion “2.4.6 Headings and Labels”. Technique: “G130: Providing descriptive headings”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/G130.html>>

[5] “Usability– Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

[6] “aplicativo bar Layout– Structure”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/layout/structure.html#structure-toolbars>>

[7] “Subheaders - Components”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/components/subheaders.html#>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C4] As telas com o mesmo tipo de conteúdo possuem o mesmo título?

Contexto: Quando existem diversas telas na aplicação relacionadas a um mesmo assunto ou tarefa.

Exemplo: A tarefa está sendo executada e possui várias etapas com diferentes títulos das telas para uma mesma tarefa.

Motivação: Auxiliar principalmente usuários com deficiência VISUAL que utilizem leitores de tela e usuários com deficiência INTELECTUAL, que podem ter dificuldades na compreensão dos elementos da aplicação. Porém, por se tratar de um critério de usabilidade mesmo antes de acessibilidade, sua aplicação influencia em todos os usuários a localizarem, compreenderem e navegarem pelo conteúdo da aplicação.

Recomendação: Os mecanismos de navegação que são repetidos em várias páginas da *Web* dentro de um conjunto de páginas da *Web* devem ocorrer na mesma ordem relativa cada vez que são repetidos, a menos que uma alteração seja iniciada pelo usuário. Os componentes que possuem a mesma funcionalidade dentro de um conjunto de páginas da *Web* devem ser identificados de forma consistente [3]. A aplicação deve utilizar palavras consistentes em todas as funcionalidades [1], títulos, títulos de seção, botões, descrição de ações, etc.

Em relação ao corpo do texto, o texto visível e não visível (para possibilitar uma melhor utilização por usuários de leitor de tela) deve ser útil, descritivo e independentemente significativo, pois alguns usuários navegam usando todos os títulos ou links em uma página. Teste seu aplicativo com um leitor de tela para identificar áreas que estão faltando ou que precisam de um melhor texto de acessibilidade [5].

Quando independente do local de partida do usuário para acessar uma tela da aplicação, o título se mantém consistente em relação ao seu conteúdo.

Fontes:

[1] “Writing”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/writing.html>>

[2] Success Criterion “2.4.10 Section Headings”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#section-headings>>

[3] Success Criterion “3.2.3 Consistent Navigation”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#consistent-navigation>>

[4] Success Criterion “3.2.4 Consistent Identification”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#consistent-identification>>

[5] “Usability– Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C5] A linguagem utilizada nos elementos textuais é legível e compreensível?

Contexto: Quando a aplicação apresenta conteúdo textual.

Exemplo: Quando o texto não possui uma linguagem clara ou o conteúdo não está em um mesmo idioma, ou mesmo, utiliza abreviações, termos técnicos, jargões.

Motivação: Auxiliar usuários com deficiência VISUAL que utilizam leitor de tela, a compreender o contexto da aplicação, a fim de entender a mensagem que o conteúdo da aplicação quer dizer. Auxiliar usuários com dificuldades de entendimento do significado da linguagem da aplicação, a fim de atender necessidades de usuários com deficiência INTELECTUAL.

Recomendação: De acordo com a 2ª Heurística de Nielsen [3], “o sistema deve falar a linguagem do usuário ao invés de usar termos orientados ao sistema”. Desse modo, de acordo também com as recomendações da WCAG 2.0 [2], o usuário precisa conseguir ler e entender o que está lendo, independente se estiver sendo lido pelo próprio usuário ou com auxílio de leitor de tela.

Partes textuais com termos técnicos ou distantes da linguagem comum ao usuário prejudicam a compreensão do conteúdo da tela para navegação, compreensão para uso das ferramentas, bem como a correção de possíveis erros informados pelo *software*. Escolha palavras comuns que sejam claras e facilmente compreensíveis para leitores iniciantes e avançados, o texto deve ser compreensível por qualquer pessoa, em qualquer lugar, independentemente da sua cultura ou idioma [1].

Fontes:

[1] “Writing”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/writing.html>>

[2] Guideline “3.1 Readable”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#language-of-page>>

[3] Nilsen, J., (1994). “Heuristic evaluation”. *Usability Inspection Methods*. New York: John Wiley & Sons, Inc. <http://www.useit.com/papers/heuristic/heuristic_evaluation.html>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C6] Existem descrições de fácil acesso para todas as abreviaturas utilizadas?

Contexto: Quando a aplicação conta com abreviações em suas partes textuais.

Exemplo: Quando o texto possui abreviações que não são descritas ou são de difícil acesso ao usuário.

Motivação: Auxiliar usuários de leitor de tela a compreenderem e lerem o conteúdo da aplicação, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de entendimento de siglas e abreviações, a fim de atender necessidades de usuários com deficiência INTELECTUAL.

Recomendação: A aplicação deve utilizar palavras comuns que sejam claras e facilmente compreensíveis para leitores iniciantes e avançados, o texto deve ser compreensível por qualquer pessoa, em qualquer lugar, independentemente da sua cultura ou idioma[2]. Um mecanismo para identificação da abreviatura com seu significado ou de sua forma expandida deve estar disponível de fácil acesso [1].

Fontes:

[1] Success Criterion “3.1.4 Abbreviations”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#abbreviations>>

[2] “Writing”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/writing.html>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C7] O idioma utilizado na aplicação é o mesmo em todo seu conteúdo?

Contexto: Quando o conteúdo textual do aplicativo está em diferentes idiomas.

Exemplo: Um texto do aplicativo que contém palavras em português e palavras em inglês sem tradução.

Motivação: Auxiliar usuários de leitor de tela a compreenderem o conteúdo da aplicação, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de entendimento em outros idiomas, a fim de atender necessidades de usuários com deficiência INTELECTUAL.

Recomendação: Todas as partes textuais da aplicação devem estar no mesmo idioma, salvo exceções particulares como nomes pessoais, termos técnicos, e demais palavras ou frases com explicação ou legendas como: termos técnicos, palavras de linguagem indeterminada e palavras ou frases em outro idioma que se tornaram parte do vernáculo do texto circundante imediato[3]. O texto deve ser compreensível por qualquer pessoa, em qualquer lugar, independentemente da sua cultura ou idioma[1].

Manter uma uniformidade no idioma da aplicação auxilia na compreensão do usuário acerca das partes textuais da aplicação, para usuários que não compreendem um outro idioma. Neste último caso, uma sugestão é tratar como uma abreviatura, ou seja, incluir meios de o usuário compreender o significado daquela palavra e o objetivo (caso seja o rótulo de funcionalidade que ela representa no aplicativo).

Também, recomenda-se colocar tags durante o desenvolvimento compatível com leitores de tela, informando o idioma no qual foi escrito para uma correta leitura ou também colocar apenas seu significado, como por exemplo: em uma funcionalidade com o rótulo descritivo “*feedback*”, que já é uma palavra bastante utilizada na língua portuguesa mas que não teria sua pronúncia correta (talvez impedindo a compreensão pelo usuário), inserir uma tag que retornaria a leitura “ENVIAR SUGESTÃO OU RECLAMAÇÃO”.

Fontes:

[1] “Writing”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/writing.html>>

[2] Success Criterion “3.1.1 Language of Page”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#language-of-page>>

[3] Success Criterion “3.1.2 Language of Parts”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#language-of-parts>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C8] As mensagens de erro ao usuário são claras?

Contexto: Quando a aplicação exibe uma mensagem de erro ao usuário.

Exemplo: A aplicação exibe ao usuário uma mensagem de erro com termos técnicos ou em formato incompreensível.

Motivação: Auxiliar usuários de leitor de tela a identificarem o propósito da mensagem de erro exibida, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de entendimento a respeito da mensagem de erro, a fim de atender necessidades de usuários com deficiência INTELECTUAL.

Recomendação: Existem três tipos de erros possíveis, para cada um deve-se tomar uma ação[2]:

- Erro de inserção do usuário: Em formulários, quando o usuário insere tipos de dados incompatíveis, números de caracteres excedente ou inferior ao requerido ou mesmo quando deixa em branco campos de preenchimento obrigatórios. Esclareça o motivo e origem do erro, destaque os campos nos quais o erro ocorreu com o texto auxiliar transformado em texto de erro com cores destacadas e contraste perceptível com o plano de fundo[2], informando ações tomadas para sua correção a menos que isso coloque em risco a segurança ou a finalidade do conteúdo[3].
- Erro por estados incompatíveis: Esclareça o motivo e origem do erro. Quando no caso de incompatibilidade das permissões de versões limitadas de uma aplicação, ou alguma configuração do usuário (offline por escolha), deixar um aviso sucinto e persistente acima da *Snackbar* na tela atual.
- Erro da aplicação: São erros que ocorrem independentemente da entrada do usuário. Como elementos clicáveis e recursos não disponíveis e são exibidos como desativados na UI. Sempre que possível, forneça ao usuário uma ação que ajudará a solucionar o erro[2].

De acordo com a 2ª Heurística de Nielsen [6], “o sistema deve falar a linguagem do usuário ao invés de usar termos orientados ao sistema”. Desse modo, de acordo também com as recomendações da WCAG 2.0 [5], o usuário precisa conseguir ler e entender o que está lendo, indiferentemente se isso está sendo realizado pelo próprio usuário ou com auxílio de tecnologia assistiva.

Partes textuais com termos técnicos ou distantes da linguagem comum ao usuário prejudicam a compreensão do conteúdo da tela para navegação, compreensão para uso das ferramentas, bem como a correção de possíveis erros informados pelo *software*. Escolha palavras comuns que sejam claras e facilmente compreensíveis para leitores iniciantes e avançados, o texto deve ser compreensível por qualquer pessoa, em qualquer lugar, independentemente da sua cultura ou idioma[1].

Fontes:

- [1] “Writing”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/writing.html>>
- [2] “Errors”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/errors.html#>>
- [3] Guideline “3.3 Input assistance”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#error-identification>>
- [4] Success Criterion “3.3.3 Error Suggestion”. WCAG 2.0, World Wide Web Consortium, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#error-suggestion>>
- [5] Guideline “3.1 Readable”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#language-of-page>>
- [6] NIELSEN, J., (1994). “Heuristic evaluation”. In J. Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods*. New York: John Wiley & Sons, Inc. <http://www.useit.com/papers/heuristic/heuristic_evaluation.html>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C9] É fornecida uma alternativa de correção com as etapas e ações necessárias para o usuário corrigir o erro?

Contexto: Quando o aplicativo exibe mensagem de erro ao usuário.

Exemplo: Ao informar a ocorrência de um erro ao usuário a aplicação não informa os passos necessários para corrigí-lo.

Motivação: Auxiliar usuários de leitor de tela a corrigirem o erro informado pelo aplicativo, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades para compreenderem alternativas de solução de erros apresentados pelo aplicativo, a fim de atender necessidades de usuários com deficiência INTELECTUAL.

Recomendação: Quando um erro ocorre, a aplicação deve destacar as áreas de ocorrência e informar a razão do erro [3][4], informando ações tomadas para sua correção a menos que isso coloque em risco a segurança ou a finalidade do conteúdo[5].

Quando existirem sugestões conhecidas para a correção de um erro, a aplicação deve fornecê-las [6][4]. Em erros por incompatibilidade das permissões de versões limitadas de uma aplicação, ou alguma configuração do usuário (offline por escolha), deixar um aviso sucinto e persistente acima da *Snackbar* na tela atual com uma possível solução, ou a causa do problema que a inclua [5]. Em erros de formulários, a aplicação deve fornecer explicação do erro e possíveis sugestões logo na sequência, ou por padrão na parte inferior do próprio campo [1].

Fontes:

[1] “Help and feedback”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/help-feedback.html>>

[2] Success Criterion “3.3.6 Error Prevention (All)”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#error-prevention-all>>

[3] Success Criterion “3.3.5 Help”. Technique: “SCR18: Providing client-side validation and alert”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#help>>

[4] “Errors”. *Material Design*, Google Inc., 2014. <<http://https://material.io/guidelines/patterns/errors.html#>>

[5] Guideline “3.3 Input assistance”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#error-identification>>

[6] Success Criterion “3.3.3 Error Suggestion”. WCAG 2.0, World Wide Web Consortium, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#error-suggestion>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C10] As mensagens da aplicação são compreensíveis para o usuário?

Contexto: Quando a aplicação exibe mensagens ao usuário.

Exemplo: Uma mensagem é exibida ao usuário com códigos, números ou termos técnicos específicos.

Motivação: Auxiliar usuários de leitor de tela a identificarem o propósito da mensagem do aplicativo exibida, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de entendimento a respeito da mensagem do aplicativo, a fim de atender necessidades de usuários com deficiência INTELECTUAL.

Recomendação: De acordo com a 2ª Heurística de Nielsen [3], “o sistema deve falar a linguagem do usuário ao invés de usar termos orientados ao sistema”. Desse modo, de acordo também com as recomendações da WCAG 2.0 [2], o usuário precisa conseguir ler e entender o que está lendo, independente se estiver sendo lido pelo próprio usuário ou com auxílio de tecnologia assistiva.

Partes textuais com termos técnicos ou distantes da linguagem comum ao usuário prejudicam a compreensão do conteúdo da tela para navegação, compreensão para uso das ferramentas, bem como a correção de possíveis erros informados pelo *software*. Escolha palavras comuns que sejam claras e facilmente compreensíveis para leitores iniciantes e avançados, o texto deve ser compreensível por qualquer pessoa, em qualquer lugar, independentemente da sua cultura ou idioma[1].

Fontes:

[1] “Writing”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/writing.html>>

[2] Guideline “3.1 Readable”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#language-of-page>>

[3] NIELSEN, J., (1994). “Heuristic evaluation”. In J. Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods*. New York: John Wiley & Sons, Inc. <http://www.useit.com/papers/heuristic/heuristic_evaluation.html>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C11] Existe ajuda ou documentação de como utilizar a aplicação?

Contexto: Quando a aplicação oferece ao usuário um tutorial, documento de texto ou quaisquer elementos que auxiliem o usuário a compreender o aplicativo.

Exemplo: O aplicativo conta com diversas funcionalidades e não oferece ajuda ao usuário.

Motivação: Auxiliar usuários a sanarem dúvidas da aplicação e obterem ajuda de como utilizá-la, a fim de atender as necessidades de usuários com deficiência VISUAL e INTELECTUAL.

Recomendação: A aplicação deve conter ajuda que possibilite sua utilização pelo usuário iniciante ou que possua dificuldades com o entendimento de seu funcionamento.

O conteúdo da ajuda fornece respostas para perguntas comuns de usuários sobre seu aplicativo. Os usuários podem enviar comentários, relatar bugs e fazer perguntas que ainda não foram respondidas pelo conteúdo da ajuda. Ele deve ser facilmente acessado e, se possível, disponibilizado a partir de vários pontos na navegação do aplicativo. Geralmente é colocado na gaveta de navegação (ou no menu flutuante) sob o rótulo “Ajuda” ou “Enviar *feedback*” [1].

Além dos rótulos em itens clicáveis, recomenda-se que a aplicação forneça ilustrações visuais além do texto, que ajudem os usuários com deficiências de leitura a entender textos difíceis que descrevam conceitos ou processos [3].

Cada tela da aplicação deve conter ao menos uma opção com informação de ajuda específicas ao contexto da tela. Em elementos clicáveis, uma opção é inserir opção de ajuda de modo adjacente ao elemento [4], uma opção é utilizar *Gestures* ou textos de acessibilidade.

Fontes:

[1] "Help and *feedback*". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/help-feedback.html>>

[2] Guideline "3.3.5 Help". WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#help>>

[3] Success Criterion "3.3.5 Help". Technique: "G103: Providing visual illustrations, pictures, and symbols to help explain ideas, events, and processes". WCAG 2.0, W3C, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/G103.html>>

[4] Success Criterion "3.3.5 Help". Technique: "G71: Providing a help *link* on every *Web* page". WCAG 2.0, W3C, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/G71.html>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C12] Na primeira interação do usuário ou com o surgimento de atualização nas funcionalidades, a aplicação oferece instruções básicas de uso?

Contexto: Quando o aplicativo conta com uma atualização ou inclusão de uma nova funcionalidade.

Exemplo: O usuário utiliza uma aplicação pela primeira vez ou uma nova versão dela e, nesse primeiro uso, a aplicação não fornece uma visão geral das funcionalidades disponíveis ou dos controles básicos da aplicação.

Motivação: Auxiliar usuários de leitor de tela a compreenderem a utilização de novas funcionalidades do aplicativo, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldade de entendimento sobre as funcionalidades do aplicativo, a fim de atender necessidades de usuários com deficiência INTELECTUAL.

Recomendação: Ocasionalmente, o conteúdo da ajuda não solicitada, ou instruções básicas como tutorial, aparece quando os usuários interagem com um aplicativo pela primeira vez, mesmo que o usuário não tenha solicitado ajuda. Recomenda-se que a aplicação forneça uma visão geral de seus recursos e funcionamento para que os usuários não precisem pesquisar no menu "Ajuda" para obter respostas sobre como usar os novos recursos. Esse tipo de ajuda geralmente aparece [1]:

- Para promover a descoberta de recursos, quando um recurso não pode ser facilmente ativado em um menu ou botão,
- Quando é introduzido um novo gesto/recurso que não estava na aplicação antes,
- Esse tipo de ajuda deve ser limitado a novos recursos e educação por gestos, pois pode ser uma interrupção e uma distração.

Fontes:

[1] "Unsolicited help - Help and feedback". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/help-feedback.html#help-feedback-behavior>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C13] Quando um usuário invoca uma ação importante, é questionado expondo as consequências de sua ação se ele deseja realmente prosseguir com a ação?

Contexto: Quando a aplicação permite ações importantes que precisam da aprovação do usuário.

Exemplo: O aplicativo executa uma ação importante sem questionar ao usuário se ele deseja concluí-la.

Motivação: Auxiliar usuários com deficiência visual, intelectual e motora a compreenderem as ações que estão sendo realizadas pelo aplicativo.

Recomendação: É recomendado que para cada ação importante solicitada pelo usuário a aplicação peça confirmação ao usuário de que ele realmente deseja prosseguir com a ação. Principalmente, avise sobre as consequências da ação quando ela for irreversível [3].

Os títulos de diálogo modal para confirmações devem ser significativos replicar novamente na pergunta a ação solicitada. Uma confirmação pede ao usuário que verifique se eles realmente quer prosseguir com a ação que acaba de invocar, confirmar e reconhecer ações pode ajudar a aliviar a incerteza sobre coisas que aconteceram ou acontecerão. Também evita que os usuários cometam erros que possam se arrepender. Essa confirmação não é necessária quando a ação é reversível ou insignificante. Por exemplo, se em um *Multiple Select* mostrar que uma imagem foi selecionada, a confirmação adicional é desnecessária[2].

Para todas as ações que apresentam confirmação recomenda-se o oferecimento, quando possível, da ação de desfazer a ação, de modo emparelhado geralmente em *Snackbars* [2][1]. *Snackbars* são caixas de diálogo exibidas logo após uma ação, com informações opcionais bem como confirmando a descarte de um rascunho e, geralmente permitem que um usuário desfaz uma ação que acaba de ser tomada. [1]

Fontes:

[1] "Alerts - Components, Dialogs". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/components/dialogs.html#dialogs-alerts>>

[2] "Acknowledgement - Confirmation & acknowledgement". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/confirmation-acknowledgement.html#confirmation-acknowledgement-confirmation>>

[3] Success Criterion "3.3.4 Error Prevention (Legal, Financial, Data)". WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#error-prevention-legal-financial-data>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C14] São fornecidas sugestões para o texto inserido em uma busca?

Contexto: Quando é fornecida a ferramenta de busca na aplicação e esta oferece sugestões de pesquisa ao usuário.

Exemplo: Durante uma pesquisa, não são exibidas ao usuário sugestões de palavras com base no histórico da aplicação ou inserção do termo pesquisado.

Motivação: Auxiliar usuários de leitor de tela a completarem uma palavra a ser pesquisada, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de escrita ou leitura, a fim de atender necessidades de usuários com deficiência INTELECTUAL e MOTORA.

Recomendação: Frequentemente, as pessoas com deficiência cognitiva, assim como pessoas com baixa visão ou cegos têm dificuldade em digitar uma palavra, mas podem conseguir que a ortografia seja aproximadamente correta. Um programa de verificação ortográfica irá poupá-los de ter que digitar a palavra corretamente. Ele também ajudará pessoas com deficiência motora fina [3].

A pesquisa básica envolve: abrir um campo de texto de pesquisa, iniciar e enviar uma consulta e então exibir um conjunto de resultados de pesquisa. A experiência do usuário durante uma pesquisa/busca pode ser aprimorada com alguns recursos, dentre eles a sugestão de texto para inserção. O texto de entrada pode ser também utilizar a função de autocomplete para ajudar os usuários que têm alfabetização limitada ou que escrevem em uma língua estrangeira. As sugestões de entrada podem ser baseadas em consultas recentes do usuário, antes que uma consulta seja concluída ou, sugestões de pesquisa concluídas automaticamente que correspondem aos resultados reais dos dados da sua aplicação. As sugestões são atualizadas com cada pressionamento de tecla e preenchimento com o texto de entrada padrão [3][2][1].

Fontes:

[1] "Search". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/search.html>>

[2] "Input text Components– Text fields". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/components/text-fields.html#text-fields-layout>>

[3] Success Criterion "3.3.5 Help". Technique: "G194: Providing spell checking and suggestions for text input". WCAG 2.0, W3C, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/G194.html>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C15] Em resposta à inserção de um texto para busca, o texto é novamente apresentado além do resultado da busca?

Contexto: Quando o aplicativo oferece a ferramenta de pesquisa/busca.

Exemplo: O usuário realiza uma pesquisa e a resposta não se inicia com os termos utilizados na pergunta/busca.

Motivação: Auxiliar usuários de leitor de tela a compreenderem os resultados da busca realizada no aplicativo, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de entendimento e memorização a identificarem os resultados da busca realizada sem a necessidade de memorização ou releitura, visando atender necessidades de usuários com deficiência INTELECTUAL. Pode ser um grande auxílio para usuários com deficiência relacionada a fala, que não usam a busca por voz (que geralmente fala a pergunta na resposta automaticamente).

Recomendação: Mesmo parecendo redundante, o auxílio da inserção do texto da busca no resultado, deve ser inserido.

Assim como em alertas de confirmação de ações importantes, nos quais os títulos de diálogo modal devem ser significativos replicar novamente na pergunta a ação solicitada. Compreende-se que o usuário entende melhor com a replicação de conteúdo, sendo melhor o excesso que a falta de explicação [1].

Sugere-se que: quando na busca tiver o trecho “Qual a previsão do clima para São Carlos hoje?” a resposta contenha “A previsão do clima hoje para São Carlos é de tempo nublado e pancadas de chuva” ou, “Os resultados da busca para clima em São Carlos hoje estão a seguir”.

Fontes:

[1] “Acknowledgement - Confirmation & acknowledgement”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/confirmation-acknowledgement.html#confirmation-acknowledgement-confirmation>>

[2] “Search”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/search.html#search-in-aplicativo-search>>

[3] “Usability– Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[C16] Os elementos não textuais clicáveis da aplicação são compreensíveis para o usuário?

Contexto: Quando a aplicação possui ícones e/ou imagens clicáveis.

Exemplo: O design de um elemento clicável não deixa claro qual sua função.

Motivação: Auxiliar usuários com dificuldades visuais e dificuldades de entendimento a identificarem o propósito do elemento clicável do aplicativo, a fim de atender as necessidades de usuários com deficiência VISUAL e INTELECTUAL.

Recomendação: De acordo com a 2ª Heurística de Nielsen [3], “O designer deve seguir as convenções do mundo real (...)”, com formas e cores correspondentes ao esperado e lógico para o usuário, facilitando a inteligibilidade dos elementos da interface em relação a seu propósito. Desse modo, de acordo também com o Princípio 3 da WCAG 2.0 [2], as informações e operações da interface do usuário precisam ser compreensíveis. Embora as recomendações expressas sejam referentes a parte textual da interface, compreende-se que, exista a necessidade de uma boa compreensibilidade se estenda todos os elementos existentes incluindo os não textuais para uma utilização satisfatória pelo usuário.

Para o *Material Design* [5], ícones de produtos devem comunicar a ideia central e a intenção de um produto de uma forma simples, arrojada e amigável. Embora sejam visualmente distintos, todos os ícones de produtos de uma marca devem ser unificados por meio de conceito e execução. Desse modo, existem diretrizes de em relação a cores, layout e disposição dos ícones para auxiliar o desenvolvedor nesse objetivo.

Do mesmo modo, ícones do sistema devem ser simples, modernos e amigáveis, cada um reduzido à sua forma mínima, expressando características essenciais e garantindo a legibilidade e clareza, mesmo em tamanhos pequenos. Os ícones do sistema Android são predefinidos, porém, o desenvolvedor pode personalizar três atributos para criar ícones do sistema diferenciados: traçado e preenchimento, raio do canto e cor [1].

Para esses e demais tipos de ícones o *Material Design* disponibiliza exemplos de ícones para serem utilizados pelo desenvolvedor, e demais recomendações de cores e layout [4].

Fontes:

[1] “System icons – Iconography”. *Material Design*, Google Inc., 2014. <<https://material.io/design/iconography/system-icons.html>>

[2] Principle “3 Understandable”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#language-of-page>>

[3] NIELSEN, J., (1994). “Heuristic evaluation”. In J. Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods*. New York: John Wiley & Sons, Inc. <http://www.useit.com/papers/heuristic/heuristic_evaluation.html>

[4] “Icon Themes - Icons”. *Material Design*, Google Inc., 2014. <https://material.io/tools/icons/?icon=tab_unselected&style=baseline>

[5] “Product icons – Iconography”. *Material Design*, Google Inc., 2014. <<https://material.io/design/iconography/product-icons.html>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

Problemas de Operação

Se relacionam a situações nas quais o usuário possa ter dificuldade:

- De alcançar algum recurso do aplicativo de modo eficiente ou facilitado;
- Para apontar no elemento clicável;
- Para usar o teclado do aplicativo para inserção de dados;
- Em saber no contexto visual, textual, sonoro ou tátil o *status* atual de uma tarefa em andamento;
- Em interagir com o aplicativo sem o uso de tecnologia(s) assistiva(s)¹.
- Em desabilitar algum recurso indesejado no aplicativo;
- Para cancelar, desfazer uma ação ou retomar o ponto onde parou durante o uso do aplicativo;
- Em operar e interagir com o aplicativo e seus recursos sem uso de voz ou gestos.

---

¹ "(..) produtos, equipamentos, dispositivos, recursos, metodologias, estratégias, práticas e serviços que objetivem promover a funcionalidade, relacionada à atividade e à participação da pessoa com deficiência ou com mobilidade reduzida, visando à sua autonomia, independência, qualidade de vida e inclusão social"(BRASIL, 2015).

[O1] A aplicação permite realizar atividades básicas utilizando interação por Gestos?

Contexto: Quando a aplicação possibilita interação por meio de gestos.

Exemplo: O usuário não consegue aplicar o zoom, usar o scroll, mudar de abas ou telas, obter informações do conteúdo ou deslizar a tela por meio de toques ou gestos.

Motivação: Auxiliar usuários de leitor de tela a deslizarem pela tela e acessarem o conteúdo, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de manipulação do dispositivo e interação manual com a aplicação, a fim de atender necessidades de usuários com deficiência MOTORA.

Recomendação: A inclusão de ações por Gestos possíveis na aplicação torna a utilização mais facilitada e eficiente para o usuário, principalmente para os que possuem dificuldades nos movimentos. Gestos são como atalhos para operações sem a necessidade de muitos passos. Por exemplo, o usuário pode remover um e-mail da caixa de entrada para a lixeira apenas deslizando para a direita com apenas um dedo sobre o e-mail desejado, sem a necessidade de selecionar e soltar a seleção para então tocar na lixeira que surgirá (geralmente no canto superior direito) na tela. É possível realizar muitas atividades básicas com a implementação deste recurso, dentre elas inserção, organização de elementos na aplicação, deleção, multiseleção, aumento e diminuição de zoom, entre outros que podem ser conferidos na página correspondente do *Material Design Gestures* [1]. A WCAG 2.0 recomenda que todos os controles de um *software* (no caso sites, mas aqui aplicados no contexto móvel) sejam acessíveis também por teclado sem exigir temporizações específicas para pressionar teclas individuais[2], nesse caso, as possibilidades de operação fornecidas por *Gestures* não obrigam o usuário a retirar os dedos da tela para realizar ações que comumente necessitam sem seu uso, como deleção e múltipla seleção, por exemplo.

Fontes:

[1] “*Gestures*”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/Gestures.html#>>

[2] Guideline “2.1.1 Keyboard”. WCAG 2.0, W3C, 2008.<<https://www.w3.org/WAI/WCAG21/quickref/#keyboard>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O2] A reprodução de sons ocorre somente quando configurada pelo usuário?

Contexto: Quando a aplicação conta com sons e/ou efeitos sonoros.

Exemplo: A aplicação executa um áudio quando não solicitado ou configurado pelo usuário.

Motivação: Auxiliar usuários de leitor de tela a não confundirem o áudio do leitor com áudios do aplicativo, a fim de atender as necessidades de usuários com deficiência VISUAL.

Recomendação: É necessário prover legendas, transcrições ou outras alternativas visuais para elementos de áudio e alertas sonoros. Deve-se evitar [1]:

- A reprodução de sons desnecessários durante o uso de leitores de tela, como músicas automáticas em background quando entrar em uma página *Web*, ou aplicação. Se houver uma música em background a aplicação deve permitir que o usuário pause ou cancele sua reprodução de modo seguro.
- A adoção de sons extras a elementos nativos, para que leitores de tela consigam interpretar o conteúdo de modo correto.

Caso um áudio seja executado por mais de 3 segundos na aplicação, a aplicação deve prover um mecanismo para pausa-lo ou pará-lo, ou mecanismo independente de controle de volume deste áudio que seja independente do volume geral do sistema [2].

Todo conteúdo dinâmico (vídeo ou áudio) deve incluir controles que permitam ao usuário pausar, alterar o volume e alternar legendas [3].

Fontes:

[1] "Sound and motion. Usability– Accessibility". *Material Design*, Google Inc., 2014. <<http://protect.unhbox\voidb@x\penalty\@M\https://material.io/guidelines/usability/accessibility.html#accessibility-sound-motion>>

[2] Guideline "1.4.2 Audio Control" - WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#audio-control>>

[3] "Making aplicativos More Accessible: Presenting Media Content". Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <<https://developer.android.com/guide/topics/ui/accessibility/aplicativos.html#aplicativolying-resources>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O3] O usuário pode sair da aplicação e retornar do ponto onde parou?

Contexto: Quando a aplicação conta com *Tabs* ou outras telas para exibir seu conteúdo.

Exemplo: A aplicação não mantém o mesmo estado de conteúdo da tela ou aba quando o usuário sai e retorna novamente para a aplicação.

Motivação: Auxiliar usuários com dificuldades de digitação e navegação, a fim de atender as necessidades de usuários com deficiência MOTORA.

Recomendação: A aplicação deve permitir que os dados sejam conservados quando uma sessão é interrompida por expiração da autenticação ou intervenções do usuário, permitindo-o que continue as atividades anteriores sem perda de dados [2]. Todo conteúdo dinâmico (vídeo ou áudio) deve incluir controles que permitam ao usuário pausar, alterar o volume e alternar legendas [3]. No caso de saída da aplicação por parte do usuário com pausa na exibição de conteúdo de vídeo ou áudio, a aplicação deve manter o estado da execução da mídia para que ao voltar o usuário possa reiniciá-la do ponto que parou.

Fontes:

- [1] Guideline "2.2.2 Pause, Stop, Hide". WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#pause-stop-hide>>
[2] Guideline "2.2.5 Re-authenticating". WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#re-authenticating>>
[3] "Making aplicativos More Accessible: Presenting Media Content". Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <<https://developer.android.com/guide/topics/ui/accessibility/aplicativos.html#aplicativolying-resources>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O4] O usuário pode cancelar uma ação (um processamento) em progresso?

Contexto: Quando a aplicação conta com a execução de alguma ação interna, que não pode ser executada instantaneamente, ou seja, demora um tempo para processamento.

Exemplo: O usuário não consegue cancelar um download em andamento.

Motivação: Auxiliar usuários com dificuldades de obter controle do processamento de alguma parte do aplicativo, a fim de atender principalmente as necessidades de usuários com deficiência MOTORA.

Recomendação: Para qualquer ação em andamento ou que comece automaticamente, que dure mais de cinco segundos e/ou seja apresentada em paralelo com outro conteúdo, é preciso que a aplicação forneça um mecanismo para o usuário pausar, parar ou esconder, a menos que a ação não faça parte de uma atividade onde é essencial para o funcionamento da aplicação (como atualizações importantes que inviabilizam o uso até estar completa).

No caso de atualizações automáticas, recomenda-se que a aplicação permita o cancelamento ou a pausa, de modo que já estarão executando em segundo plano enquanto o usuário não está interagindo diretamente com a sua aplicação (o Android exige que os usuários estejam conscientes desses tipos de serviços, mostrando uma notificação não-desabilitada) [2]. As atualizações podem ser canceladas também, com exceção para as que forem essenciais para o funcionamento da plataforma [1], no caso, o sistema operacional.

Fontes:

[1] Guideline "2.2.2 Pause, Stop, Hide". WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#pause-stop-hide>>

[2] "Required notifications - Notifications". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/notifications.html#notifications-usage>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O5] O usuário pode desfazer uma ação?

Contexto: Quando a aplicação conta com uma ação que deva ser realizada pelo usuário.

Exemplo: A aplicação não permite que uma ação de deleção, inserção, seleção e/ou atualização seja desfeita.

Motivação: Auxiliar usuários com dificuldades de manipulação do dispositivo e interação manual com a aplicação a desfazerem uma ação, a fim de atender necessidades de usuários com deficiência MOTORA. **Recomendação:** É recomendado que para cada ação importante solicitada pelo usuário a aplicação peça confirmação ao usuário de que ele realmente deseja prosseguir com a ação. Essa confirmação não é necessária quando a ação é reversível ou insignificante. Por exemplo, se em um *Multiple Select* mostrar que uma imagem foi selecionada, a confirmação adicional é desnecessária [2].

Para todas as ações que apresentam confirmação recomenda-se o oferecimento, quando possível, da ação de desfazer a ação, de modo emparelhado geralmente em *Snackbars* [2][1].

Snackbars são caixas de diálogo exibidas logo após uma ação, com informações opcionais bem como confirmando a descarte de um rascunho e, geralmente permitem que um usuário desfaz uma ação que acaba de ser tomada [1].

Recomendação: É recomendado que para cada ação importante solicitada pelo usuário a aplicação peça confirmação ao usuário de que ele realmente deseja prosseguir com a ação. Essa confirmação não é necessária quando a ação é reversível ou insignificante. Por exemplo, se em um *Multiple Select* mostrar que uma imagem foi selecionada, a confirmação adicional é desnecessária [2].

Para todas as ações que apresentam confirmação recomenda-se o oferecimento, quando possível, da ação de desfazer a ação, de modo emparelhado geralmente em *Snackbars* [2][1]. *Snackbars* são caixas de diálogo exibidas logo após uma ação, com informações opcionais bem como confirmando a descarte de um rascunho e, geralmente permitem que um usuário desfaz uma ação que acaba de ser tomada [1].

Fontes:

[1] "Alerts - Components, Dialogs". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/components/dialogs.html#dialogs-alerts>>

[2] "Acknowledgement - Confirmation & acknowledgement". *Material Design*, Google Inc., 2014. <<http://protect.unhbox\voidb@x\penalty\@Mhttps://material.io/guidelines/patterns/confirmation-acknowledgement.html#confirmation-acknowledgement-confirmation>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O6] É possível fazer alterações importantes como modificação das configurações da aplicação, via comando de voz?

Contexto: Quando a aplicação conta com a funcionalidade de comando de voz.

Exemplo: O usuário não consegue realizar modificações importantes, como habilitar ou desabilitar funcionalidades, nas configurações da aplicação, por meio de comando de voz.

Motivação: Auxiliar usuários com deficiência VISUAL a modificarem as configurações da aplicação, sem a necessidade de uso de teclado e apontamento (sem ser manual). Auxiliar usuários com dificuldades de manipulação do dispositivo e interação manual com a aplicação a alterarem configurações da aplicação, a fim de atender necessidades de usuários com deficiência MOTORA.

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O7] Existe *feedback* visual ao usuário acerca do *status* atual da tarefa?

Contexto: Quando a aplicação possui etapas para execução das tarefas possíveis.

Exemplo: A aplicação não apresenta *feedback* visual ao usuário de arquivos que foram selecionados ou etapas restantes para o total preenchimento de formulários.

Motivação: Auxiliar usuários com dificuldades de compreensão quanto ao *status* atual de uma tarefa apresentada pela aplicação, a fim de auxiliar usuários com deficiência INTELECTUAL.

Recomendação: A aplicação deve fornecer alternativas textuais e táteis a informações transmitidas por sons, e sonoras a táteis e/ou textuais [1]. A aplicação deve fornecer ao usuário *feedback* e noções de localização atual na aplicação e tarefas realizadas[3]. Pode-se utilizar indicadores de progresso de atividade, que são representações visuais do carregamento de um conteúdo na aplicação, mídia ou progresso de ações requisitadas. Um único indicador visual deve ser usado para representar cada tipo de operação. Por exemplo, uma operação de atualização deve exibir uma barra de atualização ou um círculo de atividades, mas não ambas. O tipo de indicação de *status* pode variar conforme a tarefa realizada, se para uma tarefa com diversas etapas ou demorada geralmente utiliza-se indicadores determinados indicam quanto tempo uma operação irá demorar, para as demais, utiliza-se Indicadores indeterminados com tempo de espera não especificado [2].
Recomenda-se também que a aplicação permita o funcionamento de *feedback* por *flash*, para possibilitar a percepção do *status* da tarefa (geralmente conclusão) para usuários com deficiência auditiva.

Fontes:

[1] "Sound and motion - Usability- Accessibility". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#accessibility-sound-motion>>

[2] "Progress & activity, Components- Progress". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/components/progress-activity.html>>

[3] "Notifications". Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <<https://developer.android.com/guide/topics/ui/notifiers/notifications.html?hl=pt-br#Progress>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O8] Existe *feedback* tátil ou sonoro ao usuário acerca do *status* atual da tarefa?

Contexto: Quando a aplicação possui etapas para execução das tarefas possíveis.

Exemplo: A aplicação não apresenta *feedback* tátil ou sonoro ao usuário, sobre os arquivos que foram selecionados ou etapas restantes para o total preenchimento de formulários.

Motivação: Auxiliar usuários de ferramentas assistivas que tenham dificuldades de compreensão quanto ao *status* atual de uma tarefa apresentada pela aplicação, a fim de atender as necessidades usuários com deficiência VISUAL e AUDITIVA.

Recomendação: A aplicação deve fornecer alternativas a sons, e vice-versa [1]. A aplicação deve fornecer ao usuário *feedback* e noções de localização atual na aplicação e tarefas realizadas.

Para todas as informações contidas na tela (nesse caso, inclui as informações de *status* visual da tarefa), é necessário verificar se a referência contém informações que permitem que o item seja localizado e identificado sem qualquer conhecimento de características físicas do elemento [2]. O *feedback* do *status* pode ser a combinação de vibração conforme progressos específicos (como 50% , ou a conclusão) e um aviso sonoro ao tocar no elemento de progresso da ação (*Snackbar*) ou quando a ação tiver sido concluída.

Fontes:

[1] "Sound and motion - Usability- Accessibility". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#accessibility-sound-motion>>

[2] Success Criterion "1.3.3 Sensory Characteristics". Technique "G96: Providing textual identification of items that otherwise rely only on sensory information to be understood". WCAG 2.0, World Wide Web Consortium, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/G96.html>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O9] A ajuda é de fácil acesso?

Contexto: Quando o aplicativo oferece a ferramenta de ajuda.

Exemplo: O usuário não consegue acessar quaisquer mecanismos de ajuda, como documentos, tutoriais ou títulos de campos de texto com facilidade.

Motivação: Auxiliar usuários com dificuldades de compreensão da aplicação e dificuldades de manipulação do dispositivo e interação manual com a aplicação a acessarem facilmente a ajuda da aplicação, a fim de atender as necessidades de usuários com deficiência INTELECTUAL, VISUAL e MOTORA.

Recomendação: A aplicação deve possibilitar acesso imediato para o conteúdo de ajuda logo no primeiro clique sobre um *link* com a palavra “ajuda”, sem a necessidade de subcamadas de busca para uma ajuda específica. Esse conteúdo deve ser relevante para a localização atual do usuário na aplicação. Por exemplo, se um usuário estiver visualizando suas informações de conta, o conteúdo de ajuda exibido deve incluir informações sobre contas [1][2].

Fontes:

[1] "Help and feedback". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/help-feedback.html#help-feedback-behavior>>

[2] "Patterns – Help & feedback". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/help-feedback.html#help-feedback-placement>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O10] A troca de idioma é de fácil acesso?

Contexto: Quando o aplicativo possui uma versão em mais de um idioma.

Exemplo: O usuário não consegue trocar o idioma da aplicação sem precisar pesquisar nos menus para alterar configurações de linguagem.

Motivação: Auxiliar usuários com dificuldades de manipulação do dispositivo e interação manual com a aplicação a alterarem o idioma da aplicação sem recorrerem a muitas etapas, a fim de atender necessidades de usuários com deficiência MOTORA e VISUAL que são usuários de leitores de tela.

Recomendação: Identificar o idioma do documento é importante por vários motivos, dentre eles permitir que o *software* de tradução Braille substitua os códigos de controle por caracteres acentuados e que os sintetizadores (para leitores de tela) que suportam vários idiomas sejam capazes de se adaptarem à pronúncia e à sintaxe[2]. O aplicativo deve mostrar na página inicial do menu principal as configurações mais importantes ou mais frequentemente usadas, como troca de idiomas e métodos de entrada de teclado [1].

Todos os fragmentos de palavras e frases em outro idioma devem ser identificáveis por ferramentas assistivas como leitores de tela, palavras de origem estrangeira e afins, para que o usuário consiga compreender do que se trata o elemento textual sem a pronúncia no idioma errado[3].

Fontes:

[1] "Grouping settings - Help and *feedback*". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/settings.html#settings-grouping-settings>>

[2] Success Criterion "3.1.2 Language of Parts". WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG21/quickref/#language-of-parts>>

[3] Success Criterion "3.1.1 Language of Page". Technique "H58: Using language attributes to identify changes in the human language". WCAG 2.0, W3C, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/H58.html>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O11] É possibilitado a utilização de todas as funcionalidades da aplicação por TA¹?			
Contexto: Quando a aplicação possui funcionalidades disponíveis para utilização pelo usuário que precisa acesso por TA.			
Exemplo: Uma aplicação possui uma funcionalidade e o usuário de leitor de tela não consegue utilizá-la.			
Motivação: Auxiliar usuários de leitor de tela utilizarem funcionalidades da aplicação, a fim de atender as necessidades de usuários com deficiência VISUAL.			
Recomendação: O suporte a tecnologias assistivas é específico da plataforma Android, mas é preciso que o desenvolvedor assegure que sua aplicação funcionará com métodos de entrada por toque, tipos de teclado e mouse. Por exemplo, conferir se a aplicação permitirá ser utilizada com o leitor de tela do Google <i>TalkBack</i> [1].			
Fontes: [1] “Specific. Accessibility - Usability”. <i>Material Design</i> , Google Inc., 2014. < https://material.io/guidelines/patterns/settings.html#settings-grouping-settings > [2] “Usability– Accessibility”. <i>Material Design</i> , Google Inc., 2014. < ">https://material.io/guidelines/usability/accessibility.html#> [3] BRASIL. SDHPR “Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência”. SNPD. 2009. < http://www.pessoacomdeficiencia.gov.br/aplicativo/publicacoes/tecnologia-assistiva >			
Resposta:	☐ Sim	☐ Parcialmente	☐ Não ☐ Não se aplica
Observações:			

¹TA ou Tecnologia Assistiva se define por “(..) uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social” [3].

[O12] É possibilitado o preenchimento dos campos de inserção pelo usuário com entradas de voz?

Contexto: A aplicação possui campos de inserção para preenchimento pelo usuário.

Exemplo: O usuário não consegue preencher um campo de inserção em um formulário por meio de entrada de voz.

Motivação: Auxiliar usuários de leitor de tela a preencherem campos de inserção sem a necessidade de digitação manual, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de digitação na aplicação a preencherem campos de inserção, a fim de atender necessidades de usuários com deficiência MOTORA.

Recomendação: É recomendado quando possível a implementação de preenchimento de campos de inserção com o uso de voz representado como padrão por um ícone de microfone ao lado do campo, justificado corretamente e sempre ativado. Seu tamanho recomendado é de 24dp , com altura de 8dp acima da linha inferior do campo de inserção [1].

Fontes:

[1] "Voice input icon. Components - Text fields". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/settings.html#settings-grouping-settings>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O13] A área clicável de todos os elementos clicáveis ocupa toda e apenas a extensão dos mesmos?

Contexto: A aplicação possui elementos clicáveis.

Exemplo: O usuário clica além dos limites de borda de um botão clicável e o seleciona; ou clica dentro do botão clicável e não o seleciona.

Motivação: Auxiliar usuários de leitores de tela a clicarem corretamente em um elemento clicável “falado” pela ferramenta assistiva, a fim de atenderem as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de manipulação do dispositivo e interação manual com a aplicação a clicarem corretamente em elementos clicáveis e não realizarem um clique indesejado, a fim de atender necessidades de usuários com deficiência MOTORA.

Recomendação: Muitas pessoas têm dificuldade em interagir com pequenos alvos de toque na tela de um dispositivo, portanto, é recomendado adicionar áreas de toque maiores para facilitar a navegação no aplicativo. Como padrão a área de toque deve ter no mínimo de 48dp x 48dp [1].

Em alguns casos para possibilitar a acomodação de um espectro maior de usuários, a área de toque de um elemento clicável pode ultrapassar sua delimitância visual (por exemplo, um ícone pode parecer ser 24 x 24 dp, mas o preenchimento em torno dele compreende o alvo de toque total de 48 x 48 dp), mas sempre respeitando a separação de no mínimo 8dp entre áreas de toque de elementos próximos [2]. Essa abordagem pode ser considerada, porém, para atingir esse objetivo, é recomendado utilizar alvos de toque maiores com possível personalização de tamanho pelo usuário, adequando à sua necessidade.

Os alvos de toque devem ter pelo menos 48 x 48 dp (tamanho físico de aproximadamente 9 mm, independentemente do tamanho da tela), com tamanho de alvo recomendado para elementos touchscreen de 7-10mm [2].

Outro recurso é utilizar a API *TouchDelegate* para aumentar o tamanho da área de toque dos elementos clicáveis aumentar o tamanho dos próprios elementos (mas sem ultrapassar as margens nem abranger a área de outros elementos, mesmo não clicáveis) [1].

Na imagem abaixo, temos um exemplo de área de toque delimitada de elementos clicáveis de uma aplicação que redirecionam para outras aplicações.

--- Área de toque do elemento clicável

Siga @AndroidDev no Twitter

Siga os desenvolvedores do Android no Google+

Confira os desenvolvedores do Android no YouTube

Fontes:

[1] “Making aplicativos More Accessible: Making Touch Targets Large”. Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <<https://developer.android.com/guide/topics/ui/accessibility/aplicativos.html#touch-targets>>

[2] “Touch and Pointer targets – Layout and Typography”. *Material Design*, Google Inc., 2014. <<https://material.io/design/usability/accessibility.html#layout-typography>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O14] A aplicação possui a função de busca por voz?

Contexto: Quando a aplicação possui a funcionalidade de busca.

Exemplo: O usuário não consegue realizar uma pesquisa por meio de voz.

Motivação: Auxiliar usuários com deficiência VISUAL a realizarem uma pesquisa na aplicação, sem a necessidade de uso de teclado e toque. Auxiliar usuários com dificuldades de manipulação do dispositivo e interação manual com a aplicação a realizarem uma pesquisa sem a necessidade de interação manual para início e digitação da pesquisa, a fim de atender necessidades de usuários com deficiência MOTORA.

Recomendação: A pesquisa básica envolve abrir um campo de texto de pesquisa, iniciar e enviar uma consulta e então exibir um conjunto de resultados de pesquisa. A experiência do usuário durante uma pesquisa/busca pode ser aprimorada com alguns recursos, dentre eles a busca por voz [2]. É recomendado quando possível a implementação de preenchimento de campos de inserção com o uso de voz representado como padrão por um ícone de microfone ao lado do campo, justificado corretamente e sempre ativado. Seu tamanho recomendado é de 24dp, com altura de 8dp acima da linha inferior do campo de inserção [1].

Fontes:

[1] "Voice input icon. Components - Text fields". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/components/text-fields.html#text-fields-layout>>

[2] "In-aplicativo search. Patterns - Search". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/search.html#>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[O15] As funcionalidades mais comumente utilizadas estão disponibilizadas para acesso mais rápido?

Contexto: Quando a aplicação possui muitas funcionalidades.

Exemplo: O usuário não consegue acessar de modo mais rápido que as demais as funcionalidades mais frequentemente utilizadas na aplicação.

Motivação: Auxiliar usuários com dificuldades de manipulação do dispositivo e navegação com a aplicação a acessarem de modo mais rápido funcionalidades da aplicação que preferem que estejam com acesso facilitado (encurtado), a fim de atender necessidades de usuários com deficiência MOTORA.

Recomendação: O controle de foco, ou a capacidade de controlar teclado e foco de leitura, devem ser implementados para tarefas usadas com frequência [1].

A estrutura da aplicação deve ser organizada com foco de atenção em destinos importantes, exibindo-os em guias ou na navegação lateral exibindo o conteúdo inessencial em locais menos proeminentes. A navegação deve organizar o conteúdo para que seja mais fácil encontrar destinos importantes do aplicativo, incluindo lugares frequentemente visitados, e deve conter configurações ou encorajar ações específicas. Para determinar o tipo de navegação mais adequado do aplicativo, identifique os usuários do seu aplicativo, os caminhos típicos que eles possam ter no aplicativo [2].

Em update de dezembro de 2016, o *Material Design* [3] recomenda e possibilita a inserção de até quatro atalhos de funcionalidades do aplicativo para acesso rápido e facilitado, podendo adicioná-los à tela inicial e são exibidos conforme a localização do ícone da aplicação. Pode-se inserir um botão de ação flutuante para representar a ação principal ou mais comum naquela tela da aplicação. Em forma de ícone circundado flutuando acima da UI, ele muda de cor após o foco e levanta a seleção. Quando pressionado, pode conter mais ações relacionadas. É recomendado apenas um botão de ação flutuante por tela, com padrão de 56 x 56dp, podendo variar conforme o tamanho do dispositivo [4].

Fontes:

[1] "Hierarchy Focus - Accessibility". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#accessibility-hierarchy-focus>>

[2] "Navigation defining - Navigation". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/navigation.html#navigation-defining-your-navigation>>

[3] "aplicativo Shortcut icons - Style-Icons". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/icons.html#icons-system-icons>>

[4] "Components- Buttons: Floating Action Button". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/components/buttons-floating-action-button.html#buttons-floating-action-button-floating-action-button>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

Problemas de Percepção

Se relacionam a situações nas quais o usuário pode ter dificuldade:

- Em identificar no contexto visual, textual, sonoro ou tátil notificações/sugestões dadas pelo aplicativo, inclusive durante o uso de tecnologia(s) assistiva(s);
- Para perceber a existência de conteúdo no aplicativo além do exibido no momento;
- Em visualizar em relação a cor, tamanho, forma, contraste e organização o conteúdo do aplicativo;
- Para identificar o resultado visual, textual, sonoro ou tátil de uma interação de seleção sobre um elemento do aplicativo;
- Em notar no contexto visual, textual, sonoro ou tátil o conteúdo do aplicativo, inclusive durante o uso de tecnologia(s) assistiva(s).

[P1] As notificações de atualização são perceptíveis?

Contexto: Quando a aplicação fornece notificações de atualização ao usuário.

Exemplo: O aplicativo possui atualizações de mensagens, por meio de notificações por meio de ícones ou sons que não estão perceptíveis pelo usuário.

Motivação: Auxiliar usuários de leitor de tela a perceberem e verificarem as notificações de atualizações, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com deficiência AUDITIVA a visualizarem ou sentirem e identificarem as notificações de atualização apresentadas pelo aplicativo.

Recomendação: Para conferir acessibilidade é recomendada a combinação de acordo com configurações prévias do usuário de *feedback* tátil, visual e/ou sonoro ao receber uma nova notificação. Considerando as recomendações da WCAG 2.0 [1], a aplicação deve oferecer diferentes meios de transmissão de uma informação. Quando a informação de uma notificação for transmitida por diferenças de cor, é necessário que essa informação também esteja presente em texto, com contraste e tamanho de fontes perceptíveis [1][3][4]. Quando em texto, que a apresentação da notificação inclua o uso de diferentes formas ou tamanhos, fornecendo texto ou padrões visuais, adicionando *feedback* de áudio ou toque (*haptic*) ou compatíveis com o recurso de uso do *flash* do aparelho móvel [4], bem como o uso de *notification badges* [2].

Fontes:

- [1] Guideline 1.4.1 “Use of color”. WCAG 2.0, W3C, 2008. <<http://https://www.w3.org/WAI/WCAG20/quickref/#visual-audio-contrast-without-color>>
- [2] “Tabs - Component”. *Material Design*, Google Inc., 2014. <<https://material.io/components/ios/catalog/tabs/>>
- [3] “Notifications”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/notifications.html#notifications-usage>>
- [4] “Making aplicativos More Accessible: Using Cues Other Than Color”. Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <<https://developer.android.com/guide/topics/ui/accessibility/aplicativos.html#cues-other-than-color>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[P2] As mensagens da aplicação são perceptíveis ao usuário?

Contexto: Quando a aplicação apresenta mensagens textuais direcionadas ao usuário.

Exemplo: O aplicativo possui mensagens textuais direcionadas ao usuário, notificadas por meio de ícones ou sons, ou caixas de texto que não estão perceptíveis pelo usuário.

Motivação: Auxiliar usuários de leitor de tela a perceberem e verificarem as mensagens textuais da aplicação, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com deficiência AUDITIVA a identificarem as mensagens textuais direcionadas ao usuário apresentadas pelo aplicativo.

Recomendação: Para conferir acessibilidade é recomendada a combinação de acordo com configurações prévias do usuário de *feedback* tátil, visual e/ou sonoro ao receber uma nova notificação de mensagem. Considerando as recomendações da WCAG 2.0 [1], a aplicação deve oferecer diferentes meios de transmissão de uma informação. Quando uma notificação de mensagem textual da aplicação for transmitida por diferenças de cor, é necessário que a informação apresentada por ela também esteja presente em texto, com contraste e tamanho de fontes perceptíveis [1][3][4]. Na parte textual da mensagem direcionada ao usuário, é necessário que sejam utilizados diferentes formas ou tamanhos, fornecendo texto ou padrões visuais, adicionando *feedback* de áudio ou toque (*haptic*) ou compatíveis com o recurso de uso do *flash* do aparelho móvel [4], bem como o uso de *notification badges* [2].

Fontes:

[1] Success Criterion "1.4.1 Use of color". WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG20/quickref/#visual-audio-contrast-without-color>>

[2] "Tabs - Component". *Material Design*, Google Inc., 2014. <<https://material.io/components/ios/catalog/tabs/>>

[3] "Notifications". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/notifications.html#notifications-usage>>

[4] "Making aplicativos More Accessible: Using Cues Other Than Color". Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <<https://developer.android.com/guide/topics/ui/accessibility/aplicativos.html#cues-other-than-color>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[P3] Os elementos ficam perceptíveis em diferentes orientações de tela do aparelho móvel?

Contexto: Quando a aplicação permite a exibição na tela com orientação vertical e horizontal.

Exemplo: Os elementos da aplicação não permanecem visíveis e proporcionais entre si, seja em uma tela na horizontal quanto na vertical em um mesmo dispositivo móvel.

Motivação: Auxiliar todos os perfis de usuários a perceberem ou visualizarem e interagirem com os elementos da aplicação, principalmente, usuários com deficiência MOTORA e VISUAL.

Recomendação: Recomenda-se a criação de um “*layout líquido*”, ou seja, que os elementos se adaptem ao espaço horizontal disponível, com regiões de layout definidas que redimensionam com texto e refluxo conforme necessário para exibir a região na tela [1][3]. Embora o *layout* exato varie, a relação de elementos e a ordem de leitura devem permanecer iguais [3].

Fontes:

[1] “Layout–Responsive UI”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/layout/responsive-ui.html>>

[2] “Usability– Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

[3] Success Criterion “1.4.4 Resize Text”. Technique: “G146: Using liquid layout”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/G146.html>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[P4] Quando todos os elementos a serem exibidos não cabem na tela da aplicação, é perceptível ao usuário que existem outros elementos a serem exibidos?

Contexto: Quando o aplicativo conta com uma grande quantidade de elementos que não possibilitam a sua exibição em apenas uma tela.

Exemplo: A aplicação não informa ao usuário de modo perceptível que outros elementos não estão sendo exibidos na tela.

Motivação: Auxiliar usuários de leitor de tela a perceberem todos os elementos disponibilizados pelo aplicativo, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades para perceberem quando existem elementos além dos exibidos na tela da aplicação, a fim de atender necessidades de usuários com deficiência INTELECTUAL.

Recomendação: Quando existem elementos que podem ser agrupados em temas e tipos de ações diferentes, é recomendada a separação em “*Tabs*”, para uma melhor percepção de conteúdo e operação pelo usuário [1]. A interface do usuário da aplicação deve ter *layouts* claros e simples. Cada botão adicionado, imagem e linha de texto tornam a tela mais complicada [2].

Quando existe mais conteúdo além do exibido, a aplicação deve informar ao usuário de modo perceptível, por exemplo, com o surgimento de uma “*scroll bar*” ou uma seta indicando rolagem no meio da tela. Esses elementos clicáveis de rolagem precisam ser de tamanho, cor e contrastes perceptíveis, com *feedback* de clique visual, tátil e/ou sonoro[3].

Fontes:

[1] “*Tabs* - Component”. *Material Design*, Google Inc., 2014. <<https://material.io/components/ios/catalog/tabs/>>

[2] “Usability– Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

[3] “Making aplicativos More Accessible: Using Cues Other Than Color”. Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <<https://developer.android.com/guide/topics/ui/accessibility/aplicativos.html#cues-other-than-color>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[P5] O contraste entre as cores do plano de fundo com elementos textuais e não textuais possibilita a percepção e visualização desses elementos?

Contexto: Quando a aplicação possui elementos textuais e/ou não textuais que precisam ser percebidos e visualizados pelo usuário.

Exemplo: Os elementos textuais e não textuais da tela da aplicação não estão com contraste suficiente com o plano de fundo.

Motivação: Auxiliar usuários com deficiência VISUAL a perceberem e lerem elementos na tela. A estética e agradabilidade da interface para navegação e tema relacionado em documentos sobre usabilidade renomados, como as Heurísticas de Nielsen[5] e ISO 25010[6] do conjunto SQuaRe. Baixas taxas de contraste e elementos mal organizados e escondidos na tela da aplicação podem prejudicar a navegação, prejudicando ainda mais o usuários de leitores de tela e/ou que são daltônicos.

Recomendação: As áreas e os elementos de UI grandes devem ser coloridos com sua cor primária. Uma cor secundária pode ser usada para acentuar áreas menores. Caso a aplicação não possua uma cor secundária, pode ser utilizada a cor primária para essas áreas [2]. Certifique-se de que o uso de cores do seu aplicativo atende aos padrões de acessibilidade, com contraste suficiente entre elementos (para isso o desenvolvedor pode utilizar a ferramenta *Color Tool*, do *Material Design*) [2][3].

Fontes:

- [1] "Colors & Contrast - typography". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/typography.html#typography-other-typographic-guidelines>>
- [2] "Style- Color". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/color.html#>>
- [3] "*Color Tool*". *Material Design*, Google Inc., 2014. <<https://material.io/color/#!>>
- [4] "Making aplicativos More Accessible: Using Cues Other Than Color". Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <<https://developer.android.com/guide/topics/ui/accessibility/aplicativos.html#cues-other-than-color>>
- [5] NIELSEN, J., (1994). "Heuristic evaluation". In J. Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods*. New York: John Wiley & Sons, Inc. <http://www.useit.com/papers/heuristic/heuristic_evaluation.html>
- [6] "ISO/IEC 25000 SQuaRE - System and *software* Quality Requirements and Evaluation". ISO, 2014. <<http://iso25000.com/index.php/en/iso-25000-standards>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[P6] O tamanho da fonte dos elementos textuais e o tamanho dos elementos não textuais possibilitam a percepção e visualização desses elementos?

Contexto: Quando a aplicação possui elementos textuais e/ou não textuais que precisam ser percebidos e visualizados pelo usuário.

Exemplo: Um texto de contrato da aplicação está com letras muito pequenas. Um ícone sem legenda está com um tamanho reduzido para que o usuário possa reconhecer sua figura e entender seu significado.

Motivação: Auxiliar usuários com deficiência VISUAL a perceberem e lerem elementos na tela.

Recomendação: O tamanho do alvo recomendado para os elementos de telas touchscreen é 7-10 mm, que pode ser aumentado para pessoas com deficiência motora ou crianças em fase de desenvolvimento [2]. Para melhorar a legibilidade, os usuários podem aumentar o tamanho da fonte. A aplicação deve permitir aumento da fonte com legibilidade. Dispositivos móveis e navegadores incluem recursos para permitir aos usuários ajustar o tamanho da fonte em todo o sistema. Para habilitar o tamanho da fonte do sistema em um aplicativo Android, marque o texto e seus contêineres associados para serem medidos em pixels escaláveis [3] e certifique-se de atribuir espaço suficiente para o padrão fontes de idioma grande e estrangeiro [1].

Fontes:

[1] "Line-Height - Typography". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/typography.html#typography-line-height>>

[2] "Usability- Accessibility". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

[3] "Layout-Units & Measurements". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/layout/units-measurements.html#units-measurements-scaleable-pixels-sp>>

[4] NIELSEN, J., (1994). "Heuristic evaluation". In J. Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods*. New York: John Wiley & Sons, Inc. <http://www.useit.com/papers/heuristic/heuristic_evaluation.html>

[5] "ISO/IEC 25000 SQuaRE - System and software Quality Requirements and Evaluation". ISO, 2014. <<http://iso25000.com/index.php/en/iso-25000-standards>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica
Observações:

[P7] A aplicação fornece alternativas acessíveis para conteúdos elementos não textuais?	
Contexto: Quando o aplicativo apresenta conteúdo não textual.	
Exemplo: O usuário não consegue perceber e entender elementos não textuais da aplicação, como figuras, imagens, gráficos e ícones.	
Motivação: Auxiliar usuários de leitor de tela a perceberem e identificarem os elementos não textuais disponíveis pelo aplicativo, a fim de atender as necessidades de usuários com deficiência VISUAL.	
Recomendação: A aplicação deve oferecer diferentes meios de transmissão de uma informação [1]. Quando a informação de uma notificação for transmitida por diferenças de cor, imagem ou ícone é necessário que essa informação também esteja presente em texto, com contraste e tamanho de fontes perceptíveis [1][3][4], adicionando <i>feedback</i> de áudio ou toque (<i>haptic</i>) (quando clicáveis) ou compatíveis com o recurso de uso do <i>flash</i> do aparelho móvel [4] (quando mensagens do aplicativo) em quaisquer partes . O suporte a TA¹ é específico da plataforma Android, mas é preciso que o desenvolvedor assegure que sua aplicação funcionará com métodos de entrada por toque [6], como por exemplo, o leitor de tela do Google <i>TalkBack</i>. As interfaces de toque devem maximizar a compatibilidade com TA permitindo que leitores de tela e outros dispositivos de tecnologia assistiva leiam todas as partes da sua interface, inserindo tags com texto significativo e útil correspondentes aos elementos não textuais, para que seja lido em voz alta por TA [5][7]. Sugere-se que para uma figura, exista uma descrição dos elementos que a compõem - “paisagem de árvore desenhada em um campo de grama, dia ensolarado” -, ou seu objetivo (caso seja um elemento clicável) - “ferramenta edição de foto”. Para um ícone, “<”, a frase “seguir para a esquerda” é lida em voz alta após o toque; entre outros.	
Fontes: [1] Success Criterion "1.4.1 Use of color". WCAG 2.0, W3C, 2008. <https://www.w3.org/WAI/WCAG20/quickref/#visual-audio-contrast-without-color> [2] “Tabs - Component”. <i>Material Design</i>, Google Inc., 2014. <https://material.io/components/ios/catalog/tabs/> [3] “Notifications”. <i>Material Design</i>, Google Inc., 2014. <https://material.io/guidelines/patterns/notifications.html#notifications-usage> [4] “Making aplicativos More Accessible: Using Cues Other Than Color”. Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <https://developer.android.com/guide/topics/ui/accessibility/aplicativos.html#cues-other-than-color> [5] “Usability - Accessibility”. <i>Material Design</i>, Google Inc., 2014. <https://material.io/guidelines/usability/accessibility.html#> [6] Guideline "1.4 Compatible". WCAG 2.0, W3C, 2008. <https://www.w3.org/WAI/WCAG20/quickref/#ensure-compat> [7] “Screen Readers, Usability - Accessibility”. <i>Material Design</i>, Google Inc., 2014. <https://material.io/guidelines/usability/accessibility.html#accessibility-hierarchy-focus> [8] Success Criterion "1.1.1 Non-text content". WCAG 2.0, W3C, 2008. <https://www.w3.org/WAI/WCAG21/quickref/#non-text-content> [9] BRASIL. SDHPR "Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência". SNPD. 2009.<http://www.pessoacomdeficiencia.gov.br/app/publicacoes/tecnologia-assistiva>	
Resposta:	☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica
Observações:	

¹TA ou Tecnologia Assistiva se define por “(..) uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social” [9].

[P8] Existe *feedback* visual ao usuário quando um elemento clicável é selecionado?

Contexto: Quando o aplicativo conta com elementos clicáveis.

Exemplo: Uma ou mais características do texto como cor, tamanho, formatação, negrito ou contorno não se modifica(m) alertando o usuário que o elemento clicável foi selecionado.

Motivação: Auxiliar usuários a perceberem que um elemento foi selecionado, a fim de atender as necessidades de usuários com deficiência VISUAL, MOTORA e INTELECTUAL.

Recomendação: É preciso que ocorra alguma alteração no elemento clicável após ser selecionado [4]. A aplicação precisa conectar a resposta do usuário às reações na tela, como ondinhas no ponto de contato e mais fracas gradativamente conforme a distância, para representar o toque [1]. Alguns estados podem ser atribuídos ao elemento clicável para uma percepção mais rápida pelo usuário, tais como: estados normais, suspensos, focados, pressionados e inativos [2].

Fontes:

[1] “Radial Reaction”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/motion/choreography.html#choreography-radial-reaction>>

[2] “Icon Toggles”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/components/buttons.html#buttons-toggle-buttons>>

[3] “Customize Touch *feedback* - Defining Custom Animations”. Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <<https://developer.android.com/training/material/animations>>

[4] Success Criterion “2.4.7 Focus Visible”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG20/quickref/?showtechniques=141#navigation-mechanisms-focus-visible>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[P9] Existe *feedback* tátil ou sonoro ao usuário quando um elemento clicável é selecionado?

Contexto: Quando o aplicativo conta com elementos clicáveis.

Exemplo: O usuário seleciona um elemento clicável e não existe *feedback* sonoro/tátil de que a seleção foi realizada.

Motivação: Auxiliar usuários a perceberem que um elemento foi selecionado, a fim de atender as necessidades de usuários com deficiência VISUAL, INTELECTUAL, MOTORA e AUDITIVA.

Recomendação: A aplicação deve fornecer uma alternativa de *feedback* de toque além da visual (rótulos, cores e ícones) dessa interação [1]. Essa alternativa pode ser sonora e/ou tátil. O *feedback* visual e o *feedback* do toque mostram aos usuários o que está disponível na UI, assim como nas atualizações por parte do aplicativo, deve-se adicionar diferentes reações da aplicação para as interações por parte do usuário, tais como *feedback* de áudio ou toque (*haptic*) ou compatíveis com o recurso de uso do *flash* do aparelho móvel [2].

Fontes:

[1] Guideline 1.4.1 “Use of color”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/WAI/WCAG20/quickref/#visual-audio-contrast-without-color>>

[2] “Making aplicativos More Accessible: Using Cues Other Than Color”. Google Developer (API Guides – Accessibility), Google Inc., Android Developers, 2017. <<https://developer.android.com/guide/topics/ui/accessibility/aplicativos.html#cues-other-than-color>>

[3] “Usability– Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[P10] É informado ao usuário o número de itens selecionados no momento atual?

Contexto: Quando a aplicação possibilita a seleção de mais de um item concomitantemente.

Exemplo: O usuário seleciona várias imagens em uma galeria e o aplicativo não informa de modo perceptível o número de imagens selecionadas até o momento.

Motivação: Auxiliar usuários de leitor de tela a perceberem quantos elementos foram selecionados no aplicativo, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de entendimento para perceberem quantos elementos foram selecionados na aplicação, a fim de atender necessidades de usuários com deficiência INTELECTUAL.

Recomendação: A seleção deve ser ativada pelo padrão global de toque longo (sem uso de leitores de tela), e a definida pelo usuário caso ele utilize TA¹ [1]. O usuário deve ter controle e conhecimento do que se passa atualmente na aplicação, informações importantes expressadas apenas de modo visual devem estar também representadas em texto, como informações de número de itens selecionados para permitir a percepção por usuários que experimentam versões “text-only” do aplicativo [2].

Fontes:

[1] “Item Selection”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/selection.html#selection-item-selection>>

[2] “Usability– Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

[3] BRASIL. SDHPR “Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência”. SNPD. 2009.<<http://www.pessoacomdeficiencia.gov.br/aplicativo/publicacoes/tecnologia-assistiva>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

¹TA ou Tecnologia Assistiva se define por “(..) uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social” [3].

[P11] O usuário é avisado quando o áudio da inserção por voz não é compreendido?

Contexto: Quando o aplicativo oferece a ferramenta de preenchimento de campos de inserção por voz.

Exemplo: O aplicativo não fornece nenhum *feedback* quando não consegue compreender o áudio inserido pelo usuário no campo de inserção para pesquisa.

Motivação: Auxiliar usuários de leitor de tela a perceberem quando houve algum problema de compreensão do áudio inserido para preenchimento de um campo de inserção, a fim de atender as necessidades de usuários com deficiência VISUAL. Auxiliar usuários com dificuldades de entendimento a identificarem a ocorrência de problemas de compreensão por parte da aplicação, durante o preenchimento de campos de inserção por voz, visando atender necessidades de usuários com deficiência INTELECTUAL. Pode ser um grande auxílio para usuários com dificuldades de digitação na aplicação a preencherem campos de inserção, que precisam perceber quando este não é compreendido para sua plena utilização, a fim de atender necessidades de usuários com deficiência MOTORA.

Recomendação: É recomendado quando possível a implementação de preenchimento de campos de inserção com o uso de voz representado por padrão por um ícone de microfone ao lado do campo, justificado corretamente e sempre ativado. Seu tamanho recomendado é de 24dp, com altura de 8dp acima da linha inferior do campo de inserção [1]. Para seu pleno funcionamento, o usuário deve ser informado acerca do *status* de cada ação, assim como na ocorrência de erros.

Erros ocorrem quando o aplicativo falha em completar uma ação, podendo ser quando [2]:

- O sistema operacional ou o aplicativo falha;
- A entrada do usuário não é compreendida pela aplicação;

Em ambos os casos, a aplicação deve auxiliar o usuário a corrigir erros assim que forem detectados, desabilitando a submissão de um formulário, por exemplo, e informando claramente a razão da ocorrência do erro e como resolvê-lo [2].

Campos de inserção devem incluir textos de ajuda durante toda a interação do usuário na etapa em que ele aparece e quando ocorre um erro, o texto de ajuda se transforma em um texto de erro, com descrição sucinta e coloração e/ou tamanho de fonte modificadas para destaque.

Fontes:

[1] "Voice input icon. Components - Text fields". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/components/text-fields.html#text-fields-layout>>

[2] "Errors". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/errors.html#>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[P12] É fornecido *feedback* perceptível da existência de sugestões de busca ao usuário durante a utilização TA¹?

Contexto: Quando é fornecida a ferramenta de busca na aplicação e esta oferece sugestões de pesquisa ao usuário, que precisam ser perceptíveis para acesso por usuários que utilizam TA.

Exemplo: A aplicação fornece sugestões de busca e não apresenta *feedback* tátil, sonoro e/ou visual de modo que o usuário de TA perceba a existência dessas sugestões.

Motivação: Auxiliar usuários de ferramentas de TA que tenham dificuldades de percepção quanto a existência de sugestões de uma busca apresentadas pela aplicação, a fim de atender as necessidades usuários com deficiência VISUAL e AUDITIVA.

Recomendação: A aplicação deve fornecer alternativas textuais e táteis a informações transmitidas por sons, e sonoras a táteis e/ou textuais [1]. A aplicação deve fornecer ao usuário *feedback* e noções de ações da aplicação e tarefas realizadas, bem como a existência de sugestões ou quaisquer atalhos que facilitem a interação com a interface.

Frequentemente, as pessoas com deficiência cognitiva, assim como pessoas com baixa visão ou cegos têm dificuldade em digitar uma palavra, mas podem conseguir que a ortografia seja aproximadamente correta. Um programa de verificação ortográfica irá poupá-los de ter que digitar a palavra corretamente. Ele também ajudará pessoas com deficiência motora fina [5]. Analogamente, sugestões de pesquisa que não só verifiquem a ortografia, mas completem a intenção do que o usuário quer pesquisar apenas com o começo da inserção. As sugestões de entrada podem ser baseadas em consultas recentes do usuário, antes que uma consulta seja concluída ou, sugestões de pesquisa concluídas automaticamente que correspondem aos resultados reais dos dados da sua aplicação. As sugestões são atualizadas com cada pressionamento de tecla e preenchimento com o texto de entrada padrão [5][4][3].

As interfaces de toque devem maximizar a compatibilidade com TA, e todas as funcionalidades devem ser compatíveis também durante acesso à interface por meio de TA [6][7], isso inclui notificações da aplicação, como *feedback* de sugestões de busca.

Para todas as informações contidas na tela (como opções de texto sugeridos para uma pesquisa), é necessário verificar se a referência contém informações que permitem que o item seja localizado e identificado sem qualquer conhecimento de características físicas do elemento [2]. O *feedback* pode ser a combinação de vibração e um aviso sonoro que informe ao usuário de leitor de tela sem a necessidade de toque, apenas para facilitar sua percepção.

Fontes:

[1] "Sound and motion. Usability– Accessibility". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#accessibility-sound-motion>>

[2] Success Criterion "1.3.3 Sensory Characteristics". Technique "G96: Providing textual identification of items that otherwise rely only on sensory information to be understood". WCAG 2.0, World Wide Web Consortium, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/G96.html>>

[3] "Search". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/search.html>>

[4] "Input text Components – Text fields". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/components/text-fields.html#text-fields-layout>>

[5] Success Criterion "3.3.5 Help". Technique: "G194: Providing spell checking and suggestions for text input" - WCAG 2.0, W3C, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/G194.html>>

[6] "Screen Readers - Usability– Accessibility". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#accessibility-hierarchy-focus>>

[7] “Usability– Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

[8] BRASIL. SDHPR "Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência". SNPD. 2009. <<http://www.pessoacomdeficiencia.gov.br/aplicativo/publicacoes/tecnologia-assistiva>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

¹TA ou Tecnologia Assistiva se define por “(..) uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social” [8].

Problemas de Adaptação

Se relacionam a situações possíveis nas quais o usuário:

- Precisa de um auxílio personalizado/customizado para o uso do aplicativo, ou;
- Tem dificuldade em utilizar o aplicativo em uma orientação específica (vertical/horizontal) ou em diferentes tamanho de tela.

[A1] É possível a personalização das funcionalidades para acesso mais rápido?

Contexto: Quando o aplicativo conta com diversas funcionalidades.

Exemplo: O usuário deseja acesso rápido às funcionalidades do aplicativo e não consegue personalizar atalhos com as mais utilizadas.

Motivação: Auxiliar usuários com deficiência MOTORA a organizarem a aplicação facilitando o acesso de algumas funcionalidades.

Recomendação: A navegação na aplicação deve ter fluxos de tarefas claras com etapas mínimas. O controle de foco, ou a capacidade de controlar teclado e foco de leitura, devem ser implementados para tarefas usadas com frequência [1].

A estrutura da aplicação deve ser organizada com foco de atenção em destinos importantes, exibindo-os em guias ou na navegação lateral exibindo o conteúdo inessencial em locais menos proeminentes. A navegação deve organizar conteúdo para que seja mais fácil encontrar destinos importantes do aplicativo, incluindo lugares frequentemente visitados, e deve conter configurações ou encorajar ações específicas.

Para determinar o tipo de navegação mais adequado do aplicativo, identifique os usuários do seu aplicativo, os caminhos típicos que eles possam ter no aplicativo [2].

Em update de dezembro de 2016, o *Material Design* [3] recomenda e possibilita a inserção de até quatro atalhos de funcionalidades do aplicativo para acesso rápido e facilitado, podendo adicioná-los à tela inicial e são exibidos conforme a localização do ícone da aplicação.

Fontes:

[1] "Hierarchy Focus - Accessibility". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#accessibility-hierarchy-focus>>

[2] "Navigation defining - Navigation". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/patterns/navigation.html#navigation-defining-your-navigation>>

[3] "aplicativo Shortcut icons - Style-Icons". *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/icons.html#icons-system-icons>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[A2] É possível alterar tamanho e fonte dos elementos textuais da interface?

Contexto: Quando o aplicativo conta com elementos textuais.

Exemplo: O aplicativo não possibilita que o usuário altere o tamanho e fonte dos elementos textuais.

Motivação: Auxiliar usuários a visualizarem os elementos textuais na interface do aplicativo, a fim de atender as necessidades de usuários com deficiência VISUAL.

Recomendação: A aplicação deve permitir alteração do tamanho da fonte e a fonte com legibilidade. Dispositivos móveis e navegadores incluem recursos para permitir aos usuários ajustar o tamanho da fonte e fonte* em todo o sistema. Para habilitar o tamanho da fonte do sistema em um aplicativo Android, marque o texto e seus contêineres associados para serem medidos em pixels escaláveis [3] e certifique-se de atribuir espaço grande o suficiente para o padrão de fontes dos idiomas e palavras de origem estrangeira [1].

Fontes:

[1] “Line-Height. Estilo - Tipografia”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/style/typography.html#typography-line-height>>

[2] “Usability– Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

[3] “Layout–Units & measurements”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/layout/units-measurements.html#units-measurements-scaleable-pixels-sp>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

[A3] A aplicação suporta ambas as orientações?

Contexto: Quando a aplicação apresenta uma tela de conteúdo disponibilizado para visualização em enquadramento retangular.

Exemplo: O aplicativo não possibilita que o usuário consiga visualizar o aplicativo na horizontal e na vertical.

Motivação: Auxiliar usuários a visualizarem e interagirem de modo diferenciado com o mesmo conteúdo, a fim de atender as necessidades de usuário com deficiência VISUAL e MOTORA.

Recomendação: É recomendado que a aplicação suporte diferentes orientações de tela e necessário que o design seja perceptível e visível em ambas. O usuário pode conseguir utilizar, ou ter facilidade de uso em uma orientação de tela específica, preferencialmente com os elementos melhor organizados aproveitando o espaço horizontal disponível [1]. O *Material Design* [1] e WCAG 2.0 [3] recomendam o desenvolvimento de um “*Layout líquido*”, com os elementos da tela organizados com regiões de *layout* definidas que redimensionam com texto e refluxo conforme necessário para exibir a região na tela [1][3]. Alterar a orientação de um dispositivo não deve fazer com que a IU mude de forma inesperada. Por exemplo, um aplicativo exibindo um vídeo em uma das telas divididas (no modo retrato) não deve começar a reprodução no modo de tela cheia se o dispositivo girar para o modo paisagem [4]. Embora o *layout* exato varie, a relação de elementos e a ordem de leitura devem permanecer iguais [3].

Fontes:

[1] “Layout–Responsive UI”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/layout/responsive-ui.html>>

[2] “Usability– Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

[3] Success Criterion “1.4.4 Resize Text”. Technique: “G146: Using liquid layout”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/G146.html>>

[4] “Layout - Split screen”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/layout/split-screen.html#split-screen-behavior>>

[5] “Layout–Structure”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/layout/structure.html#>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica
Observações:

[A4] O design dos elementos se ajusta aos diferentes tamanhos de tela do aparelho móvel?

Contexto: Quando a aplicação deve ser executada em dispositivos de diferentes tamanhos de tela.

Exemplo: Os elementos da aplicação não permanecem proporcionais entre si tanto em uma tela de tamanho reduzido como em uma tela grande.

Motivação: Auxiliar usuários que utilizam diferentes tipos de dispositivos e diferentes tamanhos de telas, a fim de atender as necessidades de usuário com deficiência VISUAL e MOTORA.

Recomendação: À medida que o espaço da tela aumenta, pode aplicar alguns padrões de resposta, como ‘Refluxo’ para realocar elementos no espaço disponível [3]. Os itens na tela devem ser organizados de acordo com seu nível relativo de importância, decrescente da parte superior para a inferior da tela (acessível com atalhos) e itens relacionados de uma hierarquia similar um ao lado do outro [2].

Use texto escalável e um *layout* espaçoso para possibilitar uso de fontes de texto grandes, correção de cores, ampliação ou outras configurações de assistência ativadas [2][1].

Recomenda-se a criação de um “*layout* líquido”, ou seja, que os elementos se adaptem ao espaço horizontal disponível, com regiões de *layout* definidas que redimensionam com texto e refluxo conforme necessário para exibir a região na tela [1][4]. Embora o *layout* exato varie, a relação de elementos e a ordem de leitura devem permanecer iguais [4].

Fontes:

[1] “Layout–Responsive UI”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/layout/responsive-ui.html>>

[2] “Usability– Accessibility”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/usability/accessibility.html#>>

[3] “Tabs - Component”. *Material Design*, Google Inc., 2014. <<https://material.io/components/ios/catalog/tabs/>>

[4] Success Criterion “1.4.4 Resize Text”. Technique: “G146: Using liquid layout”. WCAG 2.0, W3C, 2008. <<https://www.w3.org/TR/WCAG20-TECHS/G146.html#>>

[5] “Layout–Structure”. *Material Design*, Google Inc., 2014. <<https://material.io/guidelines/layout/structure.html#>>

Resposta: ☐ Sim ☐ Parcialmente ☐ Não ☐ Não se aplica

Observações:

RELAÇÃO UIDP *MOBILE* E ITENS DE VERIFICAÇÃO *ACC_MOBILECHECK*

Neste apêndice estão tabelas que apresentam a relação dos itens de verificação da *Acc_MobileCheck* com os 12 UIDPs *mobile* estudados. As tabelas estão separadas por tipo de problema ao qual os itens de verificação pertencem. Para conferência das siglas nas tabelas, os UIDPs *mobile* são:

LL – *List of Links*;
VS – *Voice Search*;
AC – *Auto-Complete*;
AS – *Auto-Suggest*;
SAB – *Search From Action Bar*;
TB – *Tabs*;
MS – *Multiple Select*;
TBIM – *Textbox with input Mask*;
IEM – *Inline Error Message*;
CV – *Callback Validation*;
CO – *Cancel/Ok*;
CRS – *Carousel*

A relação dos itens de verificação e UIDPs *mobile* definidas na *Acc_MobileCheck* é representada por um “x”. Assim, as questões que não possuem nenhum “x” nos UIDPs *mobile* apresentados não se aplicam para sua avaliação.

Essa relação é utilizada no filtro por UIDP *mobile* na versão *online* da *checklist*, que está disponível no endereço <<http://limesurvey.labes.icmc.usp.br/index.php/783492>>.

Problemas de Compreensão

Na [Tabela 10](#) estão as relações de UIDPs *mobile* e os itens de verificação do tipo de problema compreensão.

Tabela 10 – Relação entre itens de verificação de compreensão [C] e UIDPs *mobile*

item de verificação	UIDP <i>mobile</i>											
	LL	VS	AC	AS	SAB	TB	MS	TBIM	IEM	CV	CO	CRS
[C1]	x	x			x	x		x	x	x	x	x
[C2]	x	x			x	x	x		x	x	x	x
[C3]						x						x
[C4]						x						x
[C5]	x	x	x	x	x	x	x	x	x	x	x	x
[C6]	x							x				
[C7]	x	x	x	x		x	x	x	x	x	x	x
[C8]		x					x		x	x		
[C9]		x					x		x	x		
[C10]		x			x	x	x		x	x	x	x
[C11]	x	x			x	x		x	x	x		x
[C12]	x	x				x	x	x				x
[C13]							x				x	
[C14]		x		x	x			x				
[C15]		x	x	x	x							
[C16]	x	x			x	x			x	x	x	x

Fonte: Elaborada pelo autor.

Problemas de Adaptação

Na [Tabela 11](#) estão as relações de UIDPs *mobile* e os itens de verificação do tipo de problema adaptação.

Tabela 11 – Relação entre itens de verificação de adaptação [A] e UIDPs *mobile*

item de verificação	UIDP <i>mobile</i>											
	LL	VS	AC	AS	SAB	TB	MS	TBIM	IEM	CV	CO	CRS
[A1]	x					x						
[A2]	x	x	x	x	x	x	x	x	x	x	x	x
[A3]	x	x	x	x	x	x	x	x	x	x	x	x
[A4]	x	x	x	x	x	x	x	x	x	x	x	x

Fonte: Elaborada pelo autor.

Problemas de Operação

Na [Tabela 12](#) estão as relações de UIDPs *mobile* e os itens de verificação do tipo de problema operação.

Tabela 12 – Relação entre itens de verificação de operação [O] e UIDPs *mobile*

item de verificação	UIDP <i>mobile</i>											
	LL	VS	AC	AS	SAB	TB	MS	TBIM	IEM	CV	CO	CRS
[O1]	x			x		x					x	x
[O2]		x							x	x		
[O3]						x	x					
[O4]		x									x	
[O5]	x						x	x				
[O6]	x		x	x		x						
[O7]	x	x	x	x	x	x	x	x	x	x	x	x
[O8]	x	x	x	x	x	x	x	x	x	x	x	x
[O9]		x			x			x	x	x	x	x
[O10]	x	x	x	x				x	x	x	x	x
[O11]	x	x	x	x	x	x	x	x	x	x	x	x
[O12]		x						x	x	x		
[O13]	x	x			x	x	x	x	x	x	x	x
[O14]	x	x										
[O15]	x					x	x	x				

Fonte: Elaborada pelo autor.

Problemas de Percepção

Na [Tabela 13](#) estão as relações de UIDPs *mobile* e os itens de verificação do tipo de problema percepção.

Tabela 13 – Relação entre itens de verificação de percepção [P] e UIDPs *mobile*

item de verificação	UIDP <i>mobile</i>											
	LL	VS	AC	AS	SAB	TB	MS	TBIM	IEM	CV	CO	CRS
[P1]	x					x						
[P2]	x	x	x	x	x		x	x	x	x	x	x
[P3]	x	x	x	x	x	x	x	x	x	x	x	x
[P4]	x	x		x	x	x	x		x	x		x
[P5]	x	x	x	x	x	x	x	x	x	x	x	x
[P6]	x	x	x	x	x	x	x	x	x	x	x	x
[P7]	x	x			x	x	x		x	x	x	x
[P8]	x	x	x	x	x	x	x	x	x	x	x	x
[P9]	x	x	x	x	x	x	x	x	x	x	x	x
[P10]							x					
[P11]		x										
[P12]		x		x	x							

Fonte: Elaborada pelo autor.

TAREFAS DOS ESPECIALISTAS

C.1 Tarefas realizadas pelos especialistas

Inicialmente foram especificadas nove tarefa para uso do *Acc_MobileCheck* pelos especialistas. As tarefas foram definidas em aplicativos populares para Android com grande número de usuários, visando atingir o maior número de *UIDP mobile* na avaliação.

Para realização das validações e análises, quatro tarefas foram consideradas (as demais se encontram na [Seção C.2](#) deste apêndice). Os *UIDPs mobile* que cada tarefa permite analisar quanto aos itens de verificação do *Acc_MobileCheck* estão relacionados na [Tabela 14](#).

Tabela 14 – Tarefas atribuídas aos especialistas (TEs) para avaliação usando *Acc_MobileCheck* e *UIDPs mobile* existentes

UIDP <i>mobile</i>	Tarefas			
	TE1	TE2	TE3	TE4
<i>List of Links</i> (LL)	X	X	X	X
<i>Voice Search</i> (VS)			X	
<i>Auto-Complete</i> (AC)				
<i>Auto-Suggest</i> (AS)		X	X	X
<i>Search from Action Bar</i> (SAB)		X	X	X
<i>Tabs</i> (TB)		X		
<i>Multiple Select</i> (MS)				
<i>Textbox with Input Mask</i> (TBIM)	X	X	X	X
<i>Inline Error Message</i> (IEM)	X			
<i>Callback Validation</i> (CV)	X			
<i>Cancel/OK</i> (CO)	X	X		X
<i>Carousel</i> (CRS)				

Fonte: Elaborada pelo autor.

Em alguns casos, são encontradas diferentes implementações para um *UIDP mobile* na mesma tela, como é o caso do AS nas tarefas dos especialistas 2, 3 e 4. Nas tarefas onde o teclado é utilizado

para inserção em uma busca, o teclado padrão do Android fornece sugestões de digitação com base nas palavras esperadas e mais frequentes para aquela sequência. Nesse caso, as sugestões não são elaboradas como parte de uma string de busca exibindo possíveis resultados finais para o elemento procurado. As sugestões de demais aplicativos geralmente não podem ser desabilitadas, ao contrário do que ocorre no teclado, no qual recurso pode ser ativado e desativado diretamente nas configurações do dispositivo móvel.

As tarefas dos especialistas 1 e 2 são realizadas no *Instagram*, um aplicativo de rede social com edição e compartilhamento e edição de fotos e vídeos entre usuários (INSTAGRAM, 2017).

Tarefa do especialista 1 (TE1)

Enunciado da TE1: Criar uma nova conta de usuário com foto de perfil, forçando uma inserção inválida ou com valores de identificação já sabidamente cadastrados nos campos de texto, como nome de usuário e/ou e-mail.

Nessa tarefa, caso sejam forçadas todas entradas de usuário e tipo de senha erradas, o avaliador passa por etapas onde foram implementados os UIDPs *mobile* TBIM, IEM, CV, CO e apenas uma exibição sem interação do LL. Na Figura 33 está um exemplo da execução dos passos descritos nessa tarefa.

Figura 33 – Criação de nova conta no aplicativo *Instagram*

Fonte: Elaborada pelo autor.

O UIDP *mobile* TBIM está implementado em todos os campos de inserção de texto, como nas Figuras 33a e 33d. A implementação do UIDP *mobile* IEM precisa uma inserção inválida pelo usuário para

ser exibida. No caso do aplicativo em questão, aparece ao inserir uma senha com número de caracteres insuficiente, visto na Figura 33c.

No aplicativo *Instagram*, foi realizada uma implementação do UIDP *mobile* CV diferente da encontrada em outros aplicativos, como o Gmail (TE5 e TE6, nas tarefas não realizadas da Seção C.2). Aqui, ele não necessita de uma inserção inválida do usuário para aparecer, sendo sempre visível mostrando o status da validação, seja o status da entrada "não validada por duplicata da inserção" ou "validada pelo servidor". Uma das possibilidades de aparecimento desse UIDP *mobile* é exibido na Figura 33e, na qual o aplicativo confirma no próprio campo que o nome de usuário está disponível após realizada uma consulta no lado do servidor.

Existem outras inserções pelo usuário que mostram a implementação dos IEM e CV. A Figura 34 traz exemplos de inserção de dados indisponíveis no cadastro de uma nova conta, em tentativa de recadastro do mesmo e-mail após a TE1, como visto na Figura 34a, e também a inserção de dados com formatos e caracteres inválidos, nas Figuras 34b e 34c ou a, tanto por endereço de e-mail como por número de telefone.

O UIDP *mobile* CO aparece de modo implícito nas etapas mostradas nas Figuras 33f, 33g e 33i. Sua implementação diferenciada mescla a ideia de aparecimento das opções em uma janela suspensa na qual tocar em uma das opções seleciona-a e tocar fora dessa janela cancela a ação, bem como utilizar o retorno do dispositivo móvel que confere o mesmo resultado.

Nessa tarefa espera-se que o especialista insira valores inválidos nos campos de cadastro. Na criação da conta fica a critério do avaliador passar por todas as etapas, pois não alteraria o número de UIDP *mobile* analisados nessa tarefa. Apenas reforçaria o CO em subetapas omitidas nas quais é possível convidar amigos para o *Instagram*. Para LL, exibido na Figura 33j não há como fazer uma análise aprofundada, dado que é o final da tarefa. Porém, é interessante a observação por parte do especialista da organização dos elementos na tela, que em implementação diferenciada cumpre as definições do UIDP *mobile*, exibindo as principais funcionalidades da ferramenta e ações permitidas ao usuário.

Figura 34 – Erros de inserção possíveis na criação de conta no *Instagram* no passo exibido na Figura 33b

Fonte: Elaborada pelo autor.

Tarefa do especialista 2 (TE2)

Enunciado da TE2: Buscar e bloquear um usuário no *Instagram*, sair do aplicativo e desbloquear o mesmo usuário.

Essa tarefa, cujo um exemplo de execução é definido na Figura 35 diferente da anterior que realizava ações pontuais, é composta de quatro sub-tarefas.

Em relação ao UIDP *mobile* TB, a interação foi imposta forçando situações nas quais precisa-se oscilar entre abas, como na troca da tela inicial para busca (Figura 35a a 35c), e na pesquisa entre as abas de opções (35c).

LL é analisado quanto à clareza da implementação no aplicativo para que o usuário compreenda e encontre as ferramentas necessárias, e principalmente, levando em conta possíveis deficiências desse usuário.

O UIDP *mobile* CO é exibido por meio de uma janela *pop-up*, de modo diferente da inserção de fotos no mesmo aplicativo (na barra de ação inferior). Nessa versão, a função cancelar do UIDP *mobile* pode ser executada pelo retorno do próprio dispositivo (toque nos botões padrão da barra de navegação do Android), porém, não possibilita retorno por deslize do dedo na tela, ou toque fora do *pop-up*, como no Gmail por exemplo.

Na Figura 35c, para realização da pesquisa é usado o UIDP *mobile* SAB, em conjunto com o TBIM para inserção. Ainda durante a pesquisa, a medida que o usuário insere uma cadeia de caracteres aparece a implementação do UIDP *mobile* AS, também na Figura 35c.

Na TE2, em especial, percebemos as diferenças da implementação dos mesmos UIDPs *mobile* em um mesmo aplicativo. Desse modo, a partir da realização do conjunto de tarefas definido, é esperado obter uma visão abrangente das barreiras encontradas por usuários com deficiência ao interagir com essas implementações de UIDP *mobile*.

Tarefa do especialista 3 (TE3)

Enunciado da TE3: Fazer uma busca por termo genérico no aparelho móvel.

O objetivo dessa tarefa é analisar a implementação de outro UIDP *mobile* não atingido pelas demais tarefas (*Voice Search*), atendendo o maior número de UIDPs *mobile* já observados, mas **sem a necessidade de instalar aplicativo algum no dispositivo** (a busca será realizada no aplicativo de fábrica do próprio SO). Para isso, o especialista deve inserir um termo genérico a ser pesquisado no campo designado na tela inicial, ou seja, não é necessário que seja um nome de usuário, palavra ou conjunto aleatório de caracteres, pois o avaliado será como será a interação com o UIDP *mobile* até que o usuário atinja o objetivo desejado. Recomenda-se, para uma melhor análise que insira mais de uma vez termos diferentes para observar a apresentação da ferramenta quando encontra resultados e quando não os encontra, bem como suas sugestões.

Nessa tarefa, a ordem de aparecimento dos UIDPs *mobile* pode mudar de acordo com a versão do Android e plataforma utilizada. Para sua elaboração, foram observadas interações a partir da versão 7.1 do Android. Foi certificado que os avaliadores continham uma das versões do SO avaliado que passariam pela implementação desses UIDPs *mobile*.

Figura 35 – Busca, bloqueio e desbloqueio de usuário no aplicativo *Instagram*

Fonte: Elaborada pelo autor.

A Figura 40 traz um exemplo de pesquisa no Android 8.0.0. Nela, o avaliador passa por etapas onde foram implementados os UIDP *mobile* LL em conjunto com o SAB e VS (Figura 40b). O UIDP *mobile* VS também aparece concomitantemente com os UIDPs *mobile* TBIM e CO (Figura 36b) e o aparecimento de sugestões, baseadas nas últimas pesquisas do dispositivo e não com base em inserção do usuário. Desse modo, as sugestões não se caracterizam como AS, cuja implementação só aparece nas sugestões com base na inserção “ca” na Figura 36c, juntamente com o CO.

Caso o usuário deseje prosseguir com a pesquisa dada a inserção (Figura 36c), o aplicativo de busca interno do Android mostra os resultados da busca (Figura 40a).

A tarefa do especialista 4 é realizada no *WhatsApp*, um aplicativo de troca de mensagens entre usuários, que utiliza conexão com a internet (conforme disponibilidade) para envio mensagens e realização de chamadas entre seus usuários (WHATSAPP, 2017).

Tarefa do especialista 4 (TE4)

Enunciado da TE4: Fazer uma busca por termo genérico a partir da tela inicial do *WhatsApp*.

Nessa tarefa, o avaliador passa por etapas onde foram implementados os UIDP *mobile* TB e SAB (Figura 37a). Na sequência TBIM, AS e uma versão por teclado do CO. Na Figura 37 está um exemplo da execução dos passos descritos nessa tarefa.

Figura 36 – Busca por cadeia de caracteres no Android 8.0.0

Fonte: Elaborada pelo autor.

O UIDP *mobile* TBIM aparece com o surgimento do teclado considerado pelo aplicativo como apropriado para a entrada (Figura 37b, 37c e 37d). A parte opcional do UIDP *mobile*, segundo Nudelman (2013), está apresentada no campo de inserção para o texto de pesquisa (*Input Mask*), visto na Figura 37b.

A implementação do UIDP *mobile* SAB é exibida em todas as etapas dessa tarefa, pois, quando não há a lupa indicando a opção de busca ao usuário (Figura 37a e 37e), o campo de pesquisa está aberto, tanto vazio (Figura 37b) como preenchido (Figuras 37c e 37d). Já os padrões TB e CO são pouco visados nessa tarefa. O primeiro aparece na tela inicial, na qual são disponibilizadas as telas de conversas, *status* e chamadas (Figura 37a). Já o segundo, aparece implicitamente na etapa de digitação da pesquisa nas Figuras 37c e 37d com o cancelar no canto superior direito e o “ok” como a lupa do teclado.

Figura 37 – Pesquisa genérica a partir da tela inicial do aplicativo WhatsApp

Fonte: Elaborada pelo autor.

C.2 Tarefas não realizadas

Dentre as 9 tarefas definidas para validação do *Acc_MobileCheck*, apenas 4 foram selecionadas e realizadas pelos especialistas; as 4 tarefas estão descritas na [Seção C.1](#). As demais tarefas, que **não** foram utilizadas (TE5 a TE9) na validação ([Seção 4.4](#)), são descritas a seguir pois exemplificam implementação de alguns dos 12 UIDPs *mobile* escolhidos durante os estudos neste mestrado. Os UIDPs *mobile* que cada uma das 5 tarefas permite analisar, quanto aos itens de verificação do *Acc_MobileCheck*, estão relacionados na Tabela 15.

Tabela 15 – Tarefas não utilizadas na validação, que foram elaboradas para os especialistas (TEs) com os respectivos UIDPs *mobile* existentes

UIDP <i>mobile</i>	Tarefas				
	TE5	TE6	TE7	TE8	TE9
<i>List of Links</i> (LL)				X	
<i>Voice Search</i> (VS)					X
<i>Auto-Complete</i> (AC)			X		
<i>Auto-Suggest</i> (AS)			X		X
<i>Search from Action Bar</i> (SAB)					
<i>Tabs</i> (TB)		X			
<i>Multiple Select</i> (MS)		X		X	
<i>Textbox with Input Mask</i> (TBIM)	X		X	X	
<i>Inline Error Message</i> (IEM)	X				
<i>Callback Validation</i> (CV)	X				
<i>Cancel/OK</i> (CO)	X	X		X	X
<i>Carousel</i> (CRS)				X	

Fonte: Elaborada pelo autor.

GMAIL

O Gmail é um aplicativo de e-mails que permite o usuário se manter informado das mensagens recebidas instantaneamente por meio de notificações push, lê-las e respondê-las on-line e off-line (GOOGLE, 2017a).

Tarefa do especialista 5 (TE5)

Enunciado da TE5: Criar uma nova conta, que em alguma das etapas na inserção dos dados seja forçada uma inserção inválida ou com valores de identificação os quais já é sabido estarem cadastrados anteriormente.

Nessa tarefa, o avaliador passa por etapas onde foram implementados os UIDPs *mobile* CO, TBIM, CV e IEM. Na Figura 38 está um exemplo de cadastro seguindo os passos descritos nessa tarefa.

Em todas as telas temos uma das implementações propostas por Nudelman (2013) para o UIDP *mobile* CO, na barra inferior do aplicativo, porém, não há opção de retorno para a tela inicial se não pela barra de navegação do dispositivo. Também não possibilita retorno por deslize do dedo na tela, deixando a função de cancelamento a cargo dos botões físicos do aparelho *mobile*.

Os UIDPs *mobile* CV e IEM não são necessariamente exibidos na tarefa de cadastro de usuário, para analisá-los é preciso uma entrada inválida pelo usuário. Desse modo, o procedimento para realização do cadastro pelo usuário solicitado inclui a inserção de dados errados por digitação ou previamente existentes.

As Figura 38e e 38g são exemplos de IEM, sem uso do CV, pois exibem erros que podem ser conferidos apenas do lado do usuário como número de caracteres e caracteres inválidos. O CV aparece nos casos no qual não é possível validação sem interação com o servidor como nos casos de dados de identificação para *login* já existentes, vistos nas Figura 38f e 38h. TBIM aparece em todas as telas nas quais são exibidos campos para inserção, como exemplo a Figura 38d e 38k.

Figura 38 – Criação de nova conta de e-mail pelo aplicativo Gmail

Fonte: Elaborada pelo autor.

Tarefa do especialista 6 (TE6)

Enunciado da TE6: Realizar troca entre contas no Gmail e mudar o marcador de múltiplos e-mails.

Para isso, o avaliador terá que entrar na aplicação e trocar de conta cadastrada entre as sincronizadas no dispositivo. Na conta acessada, ele deve selecionar dois ou mais e-mails da caixa de entrada e enviar para outro marcador.

Nessa tarefa, o avaliador passa por etapas onde foram implementados CO e MS. Na [Figura 39](#)

está um exemplo da execução dos passos descritos nessa tarefa, no qual MS é utilizado para selecionar os e-mails (Figura 39c e 39d) e CO aparece em *pop-up* na Figura 39e. De certo modo podemos considerar na tela inicial da aplicação uma implementação do UIDP *mobile* LL. As possibilidades de ação do usuário como busca e escrever novo e-mail ficam explícitas na tela. Porém, embora estejam disponíveis na mesma tela inicial, as ações visualizar, selecionar, arquivar e remover e-mail não estão claras para usuários novatos na aplicação.

Figura 39 – Chaveamento entre contas e mudança de marcador em múltiplos e-mails no aplicativo Gmail

Fonte: Elaborada pelo autor.

Tarefa do especialista 7 (TE7)

Enunciado da TE7: Acessar a funcionalidade para escrita de um novo e-mail e inserir dois ou mais destinatários, buscando não necessitar digitar todo o endereço.

Nessa tarefa, o avaliador passa por etapas onde foram implementados os UIDPs *mobile* AC e AS, bem como novamente o TBIM em todos os campos de inserção por parte do usuário. A implementação do UIDP *mobile* AS aparece a partir da inserção da primeira letra do endereço de e-mail, sem considerar a ordem que ela aparece dentro do texto das sugestões exibidas (Figura 40a). No caso do UIDP *mobile* AC, sua implementação na aplicação não se mostra muito consistente, podendo por erros de toque do usuário inserir apenas parte do campo digitado sem possibilidade de edição e assim, precisando remover o endereço errado e inserir novamente. O resultado da inserção independente da validade do e-mail está relacionado com o UIDP *mobile* Textbox with Atomic Entities¹, também definido na obra de Nudelman (2013) para entidades atômicas em campos de texto, mas que não foi tratado diretamente nesse trabalho.

¹ Alguns campos de texto são usados como uma caixa de pesquisa, para localizar e inserir entidades específicas, discretas e não divisíveis (NUDELMAN, 2013, tradução nossa).

Figura 40 – Digitação de destinatário de e-mail no aplicativo nativo Gmail

Fonte: Elaborada pelo autor.

INSTAGRAM

Utilizado nas TE1 e TE2.

Tarefa do especialista 8 (TE8)

Enunciado da TE8: Inserir um conjunto de imagens no aplicativo *Instagram* buscando imagens da galeria do dispositivo.

Nessa tarefa, o avaliador passa por etapas onde foram implementados os UIDPs *mobile CO*, *MS* e *CRS*.

Existem dois modos de realizar essa tarefa: (1) selecionando imagens já existentes no dispositivo, como exemplificado na Figura 41 o (2) tirando fotos, cancelando a publicação, retornando e procedendo igual o primeiro modo para selecionar as imagens recém tiradas no dispositivo. Em quaisquer um dos modos o usuário passará pela implementação do UIDP *mobile LL*, como mostra a Figura 41a, que se bem projetado auxilia a encontrar o caminho para a ação requerida na tarefa a partir das funcionalidades que a ferramenta lhe apresenta.

Nas telas de seleção das fotos no modo (1), bem como execução de uma nova fotografia como no modo (2), temos uma das implementações propostas por Nudelman (2013) para o UIDP *mobile CO*, na barra superior do aplicativo, como mostrado nas Figura 41b, 41c e 41d, com opção também de retorno ou cancelamento da ação atual pela barra de navegação. Não possibilita retorno por deslize do dedo na tela.

O UIDP *mobile MS* pode ser analisado na segunda etapa dessa tarefa, na qual o usuário seleciona as imagens que deseja inserir para realizar a postagem. Esse passo é exibido na Figura 41c. Na Figura 41d, 41e e 41f é possível observar a implementação do UIDP *mobile CRS*, exibido na mesma aplicação de modos muito semelhantes, com diferença apenas no *feedback* de quantidade total de fotos e localização da foto visualizada no momento. Não era necessário a seleção de múltiplas imagens para analisar a implementação desse padrão, dado que haveria interação do usuário *MS* no momento da seleção dos

filtros de edição.

Figura 41 – Postagem de múltiplas fotos no aplicativo nativo *Instagram*

Fonte: Elaborada pelo autor.

GOOGLE APP SEARCH

O Google App Search é um aplicativo de busca com respostas rápidas, que permite pesquisar por quaisquer assuntos existentes na *web*, receber notificações personalizadas conforme buscas anteriores e localização, com possibilidade de busca por voz (GOOGLE, 2017b).

Tarefa do especialista 9 (TE9)

Enunciado da TE9: Fazer uma busca por voz no Google App Search.

Nessa tarefa, o avaliador passa por etapas onde foram implementados os UIDPs *mobile* TBIM, AS, VS e implicitamente o CO. Na Figura 42 está um exemplo da execução dos passos descritos nessa tarefa.

Essa tarefa tem algumas variações possíveis de passos: a primeira versão (1) tocando diretamente no ícone do aplicativo e a segunda versão (2), na qual a busca ocorre sem toque no dispositivo mas apenas por comando de voz. Para ambas as versões a tarefa começa na tela inicial vista na Figura 42a, na qual já aparece a indicação de implementação do UIDP *mobile* VS.

Na primeira, o avaliador toca no ícone da aplicação, que exibe na mesma tela os UIDPs *mobile* AS e TBIM (Figura 42b), seguindo para a captação de fala (Figura 42c). Na sequência, a aplicação exibe os resultados da busca caso a captação de voz seja bem sucedida (Figura 42f) lendo em voz alta o conteúdo para o usuário.

Já na segunda versão, a partir da tela inicial e sem toque, o utilizador dá o comando por voz dizendo “Ok, Google!” e a aplicação dá um retorno sonoro curto. Logo em seguida, a tela de captação de voz é exibida (Figura 42g) e no decorrer da inserção por voz a tela exibe a sequência sonora compreendida por extenso (Figura 42h). Os resultados são apresentados como na última etapa da versão anterior.

Figura 42 – Realização de pesquisa por voz utilizando o aplicativo nativo Google App Search

Fonte: Elaborada pelo autor.

TAREFAS DOS DESENVOLVEDORES

Tarefas realizadas pelos desenvolvedores

O objetivo de avaliações em aplicativos desenvolvidos pelos próprios desenvolvedores é analisar a diferença de experiência com a *Acc_MobileCheck* por avaliadores que conhecem em profundidade o aplicativo avaliado, para observar se possíveis problemas no uso, como a compreensão das questões por exemplo, estão relacionados à *checklist* ou derivam do conhecimento do avaliador acerca da interface.

Foi pedido aos desenvolvedores avaliadores que enviassem as capturas de tela para análise dos UIDP *mobile* presentes em cada uma das seis tarefas. Essa relação está resumida na [Tabela 16](#).

Tabela 16 – UIDPs *mobile* e tarefas dos desenvolvedores nas quais aparecem

UIDP <i>mobile</i>	Tarefas					
	TD1	TD2	TD3	TD4	TD5	TD6
<i>List of Links</i> (LL)		X			X	X
<i>Voice Search</i> (VS)						
<i>Auto-Complete</i> (AC)						
<i>Auto-Suggest</i> (AS)						
<i>Search from Action Bar</i> (SAB)						
<i>Tabs</i> (TB)					X	X
<i>Multiple Select</i> (MS)						
<i>Textbox with Input Mask</i> (TBIM)	X	X	X	X		
<i>Inline Error Message</i> (IEM)	X					
<i>Callback Validation</i> (CV)	X		X	X		
<i>Cancel/OK</i> (CO)	X	X	X	X		
<i>Carousel</i> (CRS)						

Fonte: Elaborada pelo autor.

As tarefas 1 e 2 do desenvolvedor foram elaboradas no **MobiCare**, um aplicativo protótipo para cuidado de idosos, no qual é possível cadastrar cuidadores, alarmes para horário de remédios com instruções de aplicação, programas e atividades realizadas pelo idoso, entre outros.

Tarefa do desenvolvedor 1 (TD1)

Enunciado da TD1: Autenticar-se no aplicativo e registrar um cuidador.

Para realização dessa tarefa o avaliador precisa em ordem: autenticar-se no dispositivo do cuidador usando o número do dispositivo, confirmar o nome e número nos dados do cuidador e preencher os dados do idoso.

A [Figura 43](#) traz algumas telas da execução da TD1. Foram observadas as implementações de TBIM em todas as telas. Quando na [Figura 43a](#) foi forçado o prosseguimento da autenticação sem completar o número de telefone, o aplicativo informou com o CV por meio de um IEM a ocorrência de erro no número inserido, que não estava cadastrado. Também foi observada a implementação de CO na [Figura 43c](#).

Figura 43 – Autenticação e registro de cuidador no aplicativo MobiCare

Fonte: Elaborada pelo autor.

Tarefa do desenvolvedor 2 (TD2)

Enunciado da TD2: Agendar um lembrete de remédio.

Para realização dessa tarefa o avaliador precisa entrar como cuidador (existe a visão do aplicativo pelo cuidador e pelo idoso) e cadastrar um remédio para o idoso com as seguintes características: – Losartana 50 mg, 1 comprimido, a ser administrado diariamente; Selecionar o horário, registrar uma foto do medicamento e gravar um áudio de instrução para o idoso.

Alguns passos importantes dessa tarefa são exibidos na [Figura 44](#). Foi observada uma versão em menu de LL na [Figura 44a](#). O UIDP *mobile* CO aparece em diferentes versões, a primeira na [Figura 44b](#) na barra de ação superior do aplicativo e a segunda dentro de uma janela *pop-up* ([Figura 44c](#)). Ainda na [Figura 44b](#) é exibida a máscara de entrada do TBIM, que apareceu para inserção do campo “Nome do remédio” e é aberto novamente assim que o avaliador toca no campo de inserção.

Na [Figura 44d](#) podemos analisar uma adaptação do CO, na qual a função cancelar está explícita e a função de prosseguir com a ação pode ser interpretada como o elemento clicável “usar este”, que só é ativado para toque a partir de uma inserção de voz.

Figura 44 – Agendamento de lembrete para remédio no aplicativo MobiCare

Fonte: Elaborada pelo autor.

As tarefas do desenvolvedor 3, 4, 5 e 6 foram elaboradas no **Observatório Cidadão de Enchentes (OCE)**, um aplicativo protótipo parte de uma pesquisa desenvolvida pelo ICMC/USP, cujo principal objetivo é fornecer uma plataforma para que a população possa relatar sobre o nível da água no leito do rio e sobre a extensão de áreas alagadas. Além disso, esta plataforma objetiva guiar os voluntários ao contribuir com informações relacionadas a gestão de inundação.

Tarefa do desenvolvedor 3 (TD3)

Enunciado da TD3: Realizar *login* no aplicativo OCE.

Para realização dessa tarefa o avaliador precisa abrir o aplicativo OCE, inserir o nome e senha de usuário e clicar no botão clicável “entrar”.

Alguns passos importantes dessa tarefa são exibidos na [Figura 45](#).

O UIDP *mobile* TBIM aparece completo na [Figura 45b](#) e apenas as máscaras de entrada nas [Figura 45a](#) e [45c](#). A [Figura 45d](#) mostra a implementação de CV sem o uso de IEM para a tentativa de *login* de um usuário inválido. Em todas as telas podemos observar uma adaptação do CO, na qual a função cancelar está disponibilizada apenas na barra de navegação do Android e a função de prosseguir com a ação pode ser interpretada como o botão clicável “Entrar” ou a seta para a direita na [Figura 45b](#).

Tarefa do desenvolvedor 4 (TD4)

Enunciado da TD4: Cadastrar um usuário.

Para realização dessa tarefa o avaliador precisa abrir o aplicativo OCE, tocar no botão clicável “cadastre-se”; no inserir nome, e-mail, senha e confirmação de senha de usuário e clicar no botão clicável “salvar”.

Alguns passos importantes dessa tarefa são exibidos na [Figura 46](#).

O UIDP *mobile* TBIM aparece completo na [Figura 46b](#) e apenas as máscaras de entrada na

Figura 45 – Login de usuário no aplicativo OCE

Fonte: Elaborada pelo autor.

Figura 46a. CO está presente nas Figura 46b com o cancelamento pela barra de navegação do Android e 46c com a função no botão clicável “cancelar”. A Figura 46d mostra o aviso por um *Toast Alert* em vez do IEM para informar o erro na tentativa de cadastro com senha e confirmação de senha diferentes. Um exemplo de como usabilidade e acessibilidade pode ser melhorada utilizando IEM, para que o usuário perceba o erro, onde ele ocorreu e entenda como resolvê-lo. Também é exibido o erro, similar a Figura 46d para inserção de usuário já cadastrado, na qual é utilizado o UIDP *mobile CV*.

Figura 46 – Cadastro de usuário no aplicativo OCE

Fonte: Elaborada pelo autor.

Tarefa do desenvolvedor 5 (TD5)

Enunciado da TD5: Enviar um relato sobre a intensidade da chuva.

Para realizar essa tarefa, o avaliador terá que abrir a tela inicial do aplicativo, com todas as funcionalidades, selecionar a aba “*Inform*”, selecionar o elemento clicável “*Rain Intensity*” na tela de envio de relatos, em seguida clicar no elemento clicável dentre os exibidos (Figura 47b) que mais se encaixar com o nível de intensidade de chuva observado.

Na TD5 estão as implementações de LL na Figura 47a e TB em todas as telas da Figura 47 na barra de ação inferior. O avaliador irá utilizar TB logo após o *login* na mesma Figura 47a para chegar à tela de envio de relatos. o UIDP *mobile* foi utilizado uma vez pelo conhecimento do avaliador sobre o aplicativo, caso contrário, ele poderia precisar navegar entre as abas para encontrar as ferramentas desejadas.

As funcionalidades de cancelar e concluir a ação não estão implementadas de acordo com as definições do padrão. O cancelar é realizado pela barra de navegação do Android, e o concluir se dá apenas pela seleção do elemento clicável que corresponde às opções pré definidas para intensidade da chuva (Figura 47b).

Figura 47 – Envio de relato de intensidade chuvosa no aplicativo OCE

Fonte: Elaborada pelo autor.

Tarefa do desenvolvedor 6 (TD6)

Enunciado da TD6: Pesquisar relatos sobre altura da água no leito do rio.

Para realizar essa tarefa o avaliador abrirá o aplicativo em sua tela inicial com as opções disponíveis de filtro (“*street situation*”, “*rain intensity*”, “*water level*”, etc). Dentre elas, poderá selecionar o mecanismo “*Water level in riverbed*” para visualização dos relatos inseridos.

O avaliador passa pela implementação de LL na tela inicial (Figura 48a), e na sequência interage com TB (Figura 48b) para ir até a tela desejada com os filtros de pesquisa (Figura 48c).

Muitos UIDP *mobile* não foram atendidos pelas tarefas do desenvolvedor, em geral devido a

Figura 48 – Visualização de relatos de altura de água no leito de rio no aplicativo OCE

Fonte: Elaborada pelo autor.

natureza dos aplicativos nos quais por vezes não existiam ações que possibilitassem ou precisassem utilizá-los.

Embora o foco da comparação entre as avaliações de ambos os perfis de avaliadores não seja baseada no número de UIDPs *mobile* cobertos por cada um, as tarefas dos desenvolvedores possibilitaram uma melhor análise nesse sentido. Cobrindo itens de verificação em relação a TB e CV, pouco observados nas tarefas dos especialistas, foi possível comparar se as relações dos itens de verificação com um número maior de UIDPs *mobile* foi construída corretamente, analisando as respostas esperadas e as obtidas pelos avaliadores (Tabela 6 na Seção 4.4).

E também, foi reforçada a importância de TBIM e CO como UIDP *mobile* em aplicativos interativos, pois, mesmo sem interferência das criadoras da *Acc_MobileCheck* (as quais consideraram esses padrões importantes) na elaboração de novas tarefas, eles apareceram diversas vezes.

Em especial, foram exibidos exemplos de como a ausência do IEM para exibir erros em campos de inserção (em TD3 e TD4) prejudicam a percepção e entendimento do erro. Bem como a ausência de elementos considerados importantes (na *Acc_MobileCheck*) em UIDPs *mobile* prejudicam o entendimento e a própria interação do usuário. Um exemplo é a TD5, na qual IEM está implementado com todos os elementos sugeridos por Nudelman (2013), porém, títulos de campos de inserção (cujos IV foram relacionados a TBIM pela *checklist*) estão ausentes, sendo representados apenas pela “input mask” do UIDP, que desaparece após toque ou inserção no campo. Assim, caso o usuário troque os locais corretos para cada tipo de dado e ocorra um erro, terá de apagar o texto até encontrar onde inserir corretamente Figura 46c e 46d.

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Concordo em participar, como voluntário/a, da pesquisa intitulada **“Uma abordagem para incorporação de boas práticas de acessibilidade e usabilidade no desenvolvimento de aplicativos móveis”**, que tem como pesquisadora responsável a aluna Marisa Helena da Silva Batista, mestranda do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, orientada pela professora Dr^a Renata Pontin de Mattos Fortes, as quais podem ser contatadas pelos e-mails marisa.batista@usp.br e renata@icmc.usp.br. O presente trabalho tem por objetivos propor um apoio a avaliação de acessibilidade para aplicativos móveis e, viabilizar um instrumento que é alvo desta pesquisa. Minha participação consistirá em avaliar o instrumento AccMobile_Check (um checklist elaborado pela mestranda responsável). Compreendo que esse estudo possui finalidade exclusivamente de pesquisa, e que os dados obtidos serão divulgados seguindo as diretrizes éticas da pesquisa, assegurando, assim, minha privacidade. Sei que posso desistir dessa participação quando eu quiser, e que não receberei nenhum pagamento por essa participação.

São Carlos, ____ de junho de 2018

_____
Assinatura do participante

_____
Nome completo do participante

_____
Marisa Helena da Silva Batista

_____
Renata Pontin de Mattos Fortes

LEVANTAMENTO DE UIDP NA WEB

Nas primeiras etapas desse trabalho, foi realizado um levantamento de **boas práticas** para desenvolvimento de UI, em especial UIDPs, no contexto web e de aplicativos móveis.

Este apêndice apresenta uma breve análise de Padrões de Projeto de Interface de Usuário em páginas web, por serem adotados preferencialmente pelos desenvolvedores profissionais.

F.1 Estudo preliminar de utilização de UIDP no desenvolvimento web

Com o objetivo de obter uma visão mais abrangente do que está sendo utilizado em relação a UI em desenvolvimento de *websites* atualmente, foi conduzida uma avaliação de dez *websites* dentre os que possuem mais acessos no mundo segundo Alexa¹. Nessa ferramenta é possível encontrar listas com o *ranking* de visitação de páginas web com estimativas baseadas em dados globais realizados em diferentes extensões do navegador (ALEXA, 2016).

Foram escolhidos 10 dentre os 50 *websites* mais visitados do mundo, sendo utilizados como critérios de inclusão (In) e critérios de exclusão (Ex):

- ✓ [In1]: O *website* possui conteúdo relevante para análise posterior;
- ✓ [In2]: O *website* está entre os primeiros 50 do *ranking*;
- ✗ [Ex1]: O *website* foi desenvolvido em língua estrangeira não inglesa e não mantém sua estrutura ao ser traduzido automaticamente;
- ✗ [Ex2]: O *website* não possui conteúdo relevante para análise posterior, como *websites* de conteúdo adulto, por exemplo;
- ✗ [Ex3]: Já foram selecionados mais de três *websites* da mesma categoria de funcionalidade;
- ✗ [Ex4]: O *website* é apenas uma versão em outro idioma/oficial de um país de algum já selecionado (como Google.co.in, a versão indiana do Google.com).

---

¹ <http://www.alexa.com/topsites/global>

A [Tabela 17](#) resume os *websites* selecionados (URL), suas respectivas posições no *ranking* de visitação do Alexa, categorias de funcionalidade e ID para posterior referência na análise dos dados.

Tabela 17 – *websites* selecionados na lista do Alexa

ID	Posição no <i>ranking</i> do Alexa	URL	Funcionalidade
1	1	<i>Google.com</i>	Buscas
2	2	<i>Youtube.com</i>	Entretenimento
3	3	<i>Facebook.com</i>	Rede Social
4	4	<i>Baidu.com</i>	Buscas
5	6	<i>Amazon.com</i>	Compras
6	7	<i>Wikipedia.org</i>	Estudos/Informação
7	9	<i>Twitter.com</i>	Rede Social
8	17	<i>Linkedin.com</i>	Rede Social
9	19	<i>Instagram.com</i>	Rede Social
10	36	<i>Netflix.com</i>	Entretenimento

Fonte: Elaborada pelo autor.

A partir da seleção, foi iniciada a identificação de UIDPs presentes nos *websites*, utilizando as definições de UIDPs de [Toxboe e Hobson \(2015b\)](#) (exemplificadas na [Seção 3.3](#)). Para cada *website* foi conferido se existia uma versão web móvel² e nativa, ambas no mesmo dispositivo móvel³ instalado com mesmo sistema operacional⁴. Para todos os *websites* foram encontradas ambas versões de aplicativos móveis. Foram identificados 61 UIDPs, a relação dos que foram encontrados em cada um dos 10 *websites* está resumida na [Tabela 18](#).

² Utilizando o navegador móvel Google Chrome versão 51.0.2704.81

³ Samsung Galaxy S5 mini duo (SM-G800)

⁴ Android versão 5.1.1

Tabela 18 – UIDPs encontrados nos sites observados

UIDP	1	2	3	4	5	6	7	8	9	10
Password Strength Meter	x	x	x	x			x			
Input Feedback	x		x			x	x	x	x	x
Captcha			x			x			x	
Calendar Picker			x							
Structured Format			x		x	x	x	x		x
Settings	x	x	x	x	x	x	x	x	x	x
Inplace Editor			x					x		
Fill in the Blanks					x		x		x	
Drag and drop	x	x	x	x		x	x	x		
Keyboard Shortcuts	x	x	x	x		x	x	x	x	x
Morphing Controls	x		x		x	x	x	x	x	
Steps Left			x							x
Vote To Promote			x				x	x		
Pay To Promote			x		x		x	x		
Flagging e Reporting		x	x		x		x	x	x	x
Rate Content		x	x		x		x	x	x	x
Navigation Tabs		x	x							
Home link	x	x	x	x	x	x	x	x	x	x
Notifications		x	x		x	x	x	x	x	x
Modal			x		x		x		x	
Event Calendar			x					x		
Tagging			x				x	x	x	
Continuous Scrolling			x				x	x	x	x
Horizontal D. Menu			x	x	x			x		
Favorites			x	x	x	x	x			x
Thumbnail	x	x	x	x					x	
Gallery			x	x	x		x		x	
Carousel	x		x	x			x	x		x
Activity Stream			x				x	x		
Friend			x				x	x	x	
Auto-sharing		x	x				x	x		
Invite friends			x				x	x	x	
Friend list			x					x	x	
Chat			x				x	x	x	
Reaction			x							
Follow		x	x				x	x	x	
Walkthrough			x							

UIDP	1	2	3	4	5	6	7	8	9	10
Forgiving Format					x					x
Autocomplete	x	x	x	x	x	x	x	x	x	x
Isolation Effect	x	x		x						
Input Prompt	x			x		x	x	x	x	x
Inline Hintst				x	x	x	x	x		x
Fat footer	x			x	x	x		x		x
Shortcut Dropdown					x			x		
Progressive Disclosure		x			x			x		
Categorization					x	x				x
Pagination	x	x			x					
FAQ					x	x			x	x
Image Zoom					x					
Search Filters	x				x	x		x		x
Paywall					x		x	x		x
Product page					x					
Pricing table					x					
Coupon					x					
Shopping Cart					x					
Preview						x		x	x	
Good Defaults								x		x
Inline Help Box								x		x
Expandable Input		x						x		
Completeness Meter		x								
Adaptable View						x				
Progressive Disclosure								x		
Undo						x				
Dashboard									x	
Vertical D. Menu										x
Article List								x		
Module Tabs								x		
Autosave								x		
WYSIWYG								x		
Guided Tour								x		
Account Registration	x	x	x	x	x	x	x	x	x	x
Lazy Registration	x	x	x	x	x	x	x	x	x	
Archive			x			x				
Cards						x	x			x

(a) Primeira metade

(b) Segunda metade

Fonte: Dados da pesquisa.

De modo que o número de *websites* escolhidos por tipo de funcionalidade foi desigual, para possibilitar comparações as quantidades de padrões encontrados em cada uma foi multiplicada por: (1) 4

vezes nas funcionalidades **Compras** e **Estudos/ Informação**, que tiveram apenas um *website* selecionado de cada; (2) 2 nas funcionalidades **Buscas** e **Entretenimento**, que tiveram dois *websites* selecionados cada e (3) 1 para **Redes Sociais**, que teve quatro *websites* selecionados. Na **Figura 49** são apresentados os resultados absolutos (na parte inferior de cada coluna) e proporcionais (a coluna total já com a multiplicação), que apontaram Redes Sociais como a categoria com maior número de UIDPs utilizados (136).

O Facebook obteve a maior concentração de UIDPs, somando sozinho 40 padrões, entre eles alguns exclusivos como o *Reaction*. Todos os *websites* da categoria rede social possuem um número maior de UIDPs implementados que qualquer outro site analisado (**Tabela 18**).

Figura 49 – Quantidade real e proporcional de UIDPs web encontrados entre os sites, dividida por funcionalidade

Fonte: Elaborada pelo autor.

A **Figura 50** exibe para cada UIDP web o número de *websites* nos quais ele foi identificado. Dos 61 padrões observados, apenas 4 estavam presentes em 100% das análises, sendo eles: (1) *Autocomplete* (**Figura 51a**), no qual quando o usuário insere o início de uma palavra ou frase o próprio campo de inserção fornece sugestões; (2) *Account Registration* (**Figura 51d**), que permite que o usuário crie uma conta no site; (3) *Home Link* (**Figura 51b**) no qual existe um campo clicável e intuitivo que retorna o usuário de qualquer página para a página inicial e; (4) *Settings* (**Figura 51c**), que permite ao usuário personalizar especificidades da ferramenta utilizada.

Um ponto interessante é a diferença no significado de UIDP web *Autocomplete*⁵ e o UIDP mobile *Auto-Complete* (**Seção 3.5**). Ambos possuem quase o mesmo nome, porém a definição de cada um é bastante distinta. Enquanto o *Auto-Complete (mobile)* se refere a função de completar automaticamente à medida que os caracteres vão sendo digitados, o *Autocomplete (web)* se refere a função de fornecer

⁵ <http://ui-patterns.com/patterns/Autocomplete>

Figura 50 – Número de *websites* que utilizam o UIDP

Fonte: Elaborada pelo autor.

sugestões conforme o texto inserido pelo usuário em um campo de pesquisa. Essa função se encaixa na definição dada por [Nudelman \(2013\)](#) para o UIDP *mobile Auto-Suggest*.

Na [Figura 52](#) estão o número de sites dentro de cada categoria nos quais foi identificado cada UIDP. Foram considerados apenas UIDPs que apareceram em mais de um *website* e categorias com ao menos dois *websites* selecionados. Observando essa relação é possível notar uma tendência em categorias específicas. Um exemplo é o *Notifications*, encontrado em todos os *websites* de Redes Sociais e Entretenimento, mas em nenhum da categoria buscas. Também foi observado que existem UIDPs exclusivos de uma categoria, como o caso dos UIDPs: *Friend*, que permite se conectar mutuamente a outro usuário; *Chat*, que permite conversar privativamente com um ou mais usuários e, *Continuous Scrolling*, que exibe o conteúdo da tela continuamente ao deslizar a barra de rolagem. Todos foram observados apenas e em todas redes sociais analisadas ([Figura 52](#)).

Considerando que os 10 *websites* analisados possuem versão nativa e web móvel, e se tratam de *websites* com grande número de acessos, podemos considerar que empresas responsáveis por *websites* de

Figura 51 – Exemplo dos quatro UIDP com ocorrência em todos os *websites* analisados

Fonte: Elaborada pelo autor.

sucesso estão buscando se adaptar ao crescimento constante do uso de dispositivos móveis para realização das tarefas antes realizadas em um *desktop*.

Figura 52 – Número de *websites* dentro de uma categoria que utilizam o UIDP

Fonte: Elaborada pelo autor.