



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

DiaVision: Aplicação Móvel para o Autocuidado de Diabéticos com Acuidade Visual Prejudicada

Trabalho de Conclusão de Curso

Jonathan Kelvin de Jesus Santos



São Cristóvão – Sergipe

2022

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Jonathan Kelvin de Jesus Santos

**DiaVision: Aplicação Móvel para o Autocuidado de Diabéticos
com Acuidade Visual Prejudicada**

Trabalho de Conclusão de Curso submetido ao Departamento de Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador(a): Profa. Dra. Adicinéia Aparecida de Oliveira

São Cristóvão – Sergipe

2022

*Este trabalho é dedicado a todos que acreditam na educação
como meio essencial para o desenvolvimento sustentável de qualquer sociedade.*

Agradecimentos

Não haveria como não falar da minha mãe, Diana, em primeiro lugar neste espaço, esta que abriu mão de diversas coisas para lutar pelo bem de seus 3 filhos, que sempre quis o melhor pra mim e fez o que pôde para evitar e alertar sobre meus erros. E mesmo não entendendo minhas escolhas, não as barrou e sempre esteve lá para quando eu quebrasse a cara.

Mãe, obrigado por tudo!

Agradeço também à Sergiane e Walesson, que me confiaram a honra de ser padrinho de Wallace. Aos meus irmãos Daiane e David que, em meio à tantas brigas, sempre nos apoiamos e defendemos uns aos outros. Ao meu primo, Lucas, que foi uma referência durante a graduação. E, ao restante dessa grande família que, embora não os nomeie aqui, quero que saibam que foram e são muito importantes para mim.

As amizades que fiz foi o que me manteve firme para seguir até o final dessa jornada. Assim, não posso deixar de mencionar Abayomi com quem tenho quilômetros de mensagens trocadas no Telegram e que está em constante mudança e revolta (*pelu idi*). Brunna que talvez não saiba, mas nos momentos em que eu estava cheio de ansiedade, fez-me sentir em paz, enquanto lia livros no silêncio de sua sala.

Igor e seus argumentos socráticos que foram interpretados como sofistas e Geovanne que está trabalhando feito louco e me deixando preocupado com um *burnout*. Mayara que sempre notava quando estava desanimado e estava lá para quando eu precisasse desabafar. Raul de quem eu tinha que esconder a preocupação para evitar que infartasse e, por fim devido à ordem alfabética, Roberto, o tiozão do pavê.

Embora eu consciência de minhas dificuldades para me expressar, principalmente com relação a sentimentos, não sei o por que, mas nunca consegui dizer “Eu te amo” para ninguém depois que cresci, nem mesmo para minha mãe. Assim, queria aproveitar esse momento para aqui deixar registrado que embora nunca tenha-os dito:

Amo vocês!

Não posso esquecer dos professores do DComp que apresentaram os caminhos que possibilitaram que eu seguisse nessa jornada. Também aos professores de outros departamentos como o DPS e DCS que mostraram a importância do pensamento crítico. E, em especial, a Gilton sempre animado e com novas ideias, Leonardo por seus conselhos e desafios, e Ricardo que nos ouvia e parecia entender nossas frustrações.

E, por fim, à minha orientadora Adicinéia que, além da orientação e paciência comigo neste trabalho, foi a única que, transmitindo conhecimento durante as aulas, conseguiu manter minha concentração por tanto tempo, horas, sem me dar sono.

*Num processo de construção, de capital
Esquecemos de algo vital, a inclusão
E promovemos a exclusão, digital*

Listas de ilustrações

Figura 1 – Quantidade de artigos encontrados por base.	29
Figura 2 – Quantidade de artigos aceitos, rejeitados ou duplicados na seleção.	30
Figura 3 – Quantidade de artigos rejeitados por critério de exclusão.	31
Figura 4 – Artigos aceitos ou rejeitados na fase de extração.	32
Figura 5 – Artigos rejeitados na fase de extração por critério exclusão.	33
Figura 6 – Diagrama de casos de uso.	74
Figura 7 – Diagrama de Classes de Domínio.	75
Figura 8 – Telas iniciais do protótipo.	76
Figura 9 – Demais telas do protótipo.	77
Figura 10 – Estrutura de Pastas do Projeto.	82
Figura 11 – Classe Principal do Módulo.	83
Figura 12 – Recuperar Injeção do <i>Repository</i> no <i>Controller</i> .	83
Figura 13 – Recuperar <i>Controller</i> na <i>Page</i> .	83
Figura 14 – Esquema do Banco de Dados.	84
Figura 15 – Exemplo de Classe no Banco.	85
Figura 16 – Tela de <i>Login</i> do <i>Dashboard</i> .	86
Figura 17 – <i>Dashboard</i> do Administrador do Sistema.	87
Figura 18 – Página para Edição de Objeto.	88
Figura 19 – Página para Cadastro de Objeto.	88
Figura 20 – Introdução ao Aplicativo.	89
Figura 21 – Botão de Acessibilidade.	90
Figura 22 – Menu de Acessibilidade.	90
Figura 23 – Demonstração de Personalização.	91
Figura 24 – Intro Botão de Acessibilidade.	91
Figura 25 – Tela de Autenticação.	92
Figura 26 – Tela de Registro.	93
Figura 27 – Tela para Recuperação de Senha.	93
Figura 28 – Tela Principal Tema Escuro.	94
Figura 29 – Restante da Tela Principal.	94
Figura 30 – Tela de Configurações.	95
Figura 31 – Telas de Preferências e Dados.	95
Figura 32 – Notificação de Medicação.	96
Figura 33 – Notificação de Glicemia.	96
Figura 34 – Tela Inicial do Módulo Alimentação.	97
Figura 35 – Registro de Alimentação.	98
Figura 36 – Adicionando Alimentos.	98

Figura 37 – Registro de Alimento.	99
Figura 38 – Buscar Alimento.	99
Figura 39 – Listagem de <i>Apps</i> de Visão.	100
Figura 40 – Sugerir <i>App</i> de Visão.	100
Figura 41 – Listagem de Atividades Física.	101
Figura 42 – Registro de Atividade Física.	101
Figura 43 – Listagem de Autocuidados.	102
Figura 44 – Tela de Autocuidado.	102
Figura 45 – Listagem de Centros de Saúde.	103
Figura 46 – Tela do Centro de Saúde.	103
Figura 47 – Listagem de Registros de Glicemia.	104
Figura 48 – Alerta de Glicemia.	104
Figura 49 – Listagem de Medicações.	105
Figura 50 – Notificar de Medicação.	105
Figura 51 – Registro de Medicação.	106
Figura 52 – <i>Autocomplete</i> de Medicamentos.	106
Figura 53 – Listagem de Avaliações dos Pés.	107
Figura 54 – Avaliar Pés.	107
Figura 55 – Listagem de Diurese.	108
Figura 56 – Avaliar Diurese.	108

Lista de quadros

Quadro 1 – Palavras-chave e Sinônimos.	27
Quadro 2 – <i>String</i> genérica.	27
Quadro 3 – <i>Strings</i> específicas para busca em cada base.	28
Quadro 4 – Artigos aceitos na fase de extração.	34
Quadro 5 – Estudos relacionados identificados no processo de MSL.	50
Quadro 6 – Priorização de diretrizes de acessibilidade para usuários com DV.	57
Quadro 7 – Descrição do problema.	70
Quadro 8 – Relação de estórias de usuários.	71
Quadro 9 – Requisitos Funcionais da aplicação.	72
Quadro 10 – Requisitos Não-Funcionais da aplicação.	73
Quadro 11 – Requisitos Inversos da aplicação.	73
Quadro 12 – Máquinas de Desenvolvimento.	79
Quadro 13 – <i>Smartphones</i> utilizados no Desenvolvimento.	80

Lista de tabelas

Tabela 1 – Categorias dos problemas identificados.	38
Tabela 2 – Problemas de acessibilidade encontrados por categoria.	39
Tabela 3 – Categorias dos tipos de problemas mais identificados.	51
Tabela 4 – Problemas mais frequentes encontrados pelos usuários por tipo de DV.	51
Tabela 5 – Aspectos de acessibilidade à DV discutidos por <i>devs Android no Stack Overflow</i> .	52
Tabela 6 – Diretrizes do UXDG por categoria.	53
Tabela 7 – Categorias dos problemas mapeados na literatura.	53
Tabela 8 – Problemas relacionados às categorias CPM1, CPM2 e CPM6.	54
Tabela 9 – Problemas relacionados à entrada de dados (CPM3).	54
Tabela 10 – Problemas relacionados à interação por gestos (CPM4).	55
Tabela 11 – Problemas relacionados a leitores de tela (CPM5).	55
Tabela 12 – Categorias dos requisitos encontrados.	56
Tabela 13 – Requisitos essenciais e desejáveis focados em DV.	56
Tabela 14 – Categorias das diretrizes de acessibilidade <i>mobile</i> baseadas no eMAG.	57
Tabela 15 – Diretrizes de acessibilidade <i>mobile</i> baseadas no eMAG.	58
Tabela 16 – Tecnologias utilizadas no desenvolvimento e plataforma alvo das aplicações.	59
Tabela 17 – Técnicas utilizadas no desenvolvimento das soluções de acessibilidade do MSL.	60
Tabela 18 – Principais problemas identificados pelos estudos relacionados.	61
Tabela 19 – Diretrizes e técnicas relacionadas à cada tipo de problema.	62
Tabela 20 – Relação de técnicas adotadas pelos artigos e propostas para o DiaVision.	67
Tabela 21 – Relação de funcionalidades dos <i>apps</i> encontrados nas lojas de aplicativos.	69

List of abbreviations and acronyms

ACM	<i>Association for Computing Machinery</i>
ADA	<i>American Diabetes Association</i>
ADS	Administrador do Sistema
API	<i>Application Programming Interface</i>
APM	Aplicativo Móvel
<i>App</i>	<i>Application</i>
AR	Artigo Relacionado
ARIA	<i>Accessible Rich Internet Applications</i>
AM	Artigo Mapeado
CETIC	Centro de Estudos sobre as Tecnologias da Informação e da Comunicação
CNN	<i>Convolutional Neural Network</i>
CPER	Categoria de Problemas de Estudos Relacionados
DCOMP	Departamento de Computação
DM	<i>Diabetes Mellitus</i>
DSMES	<i>Diabetes Self-Management Education and Support</i>
DV	Deficiência Visual
eMAG	Modelo de Acessibilidade em Governo Eletrônico
IBGE	Instituto Brasileiro de Geografia e Estatística
IBM	<i>International Business Machines Corporation</i>
ICF	<i>International Classification of Functioning, Disability and Health</i>
IDF	<i>International Diabetes Federation</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
INPI	Instituto Nacional de Propriedade Industrial
LOC	<i>Lines of code</i>

MAC	<i>Mobile Accessibility Checker</i>
MATE	<i>Mobile Accessibility Testing</i>
MSL	Mapeamento Sistemático da Literatura
MWBP	<i>Mobile Web Best Practices</i>
OMS	Organização Mundial da Saúde
PDV	Pessoa(s) com Deficiência Visual
PPGITS	Programa de Pós-graduação Profissional em Gestão e Inovação Tecnológica em Saúde
PO	<i>Product Owner</i>
TA	Tecnologia Assistiva
TAM	Técnica de Artigo Mapeado
TTS	<i>Text-to-speech</i>
SO	Sistema Operacional
SDK	<i>Software Development Kit</i>
SUS	<i>System Usability Scale</i>
UFS	Universidade Federal de Sergipe
UI	<i>User Interface</i>
URL	<i>Uniform Resource Locator</i>
UX	<i>User Experience</i>
UXDG	<i>UX Design Guideline</i>
W3C	<i>World Wide Web Consortium</i>
WAI	<i>Web Accessibility Initiative</i>
WCAG	<i>Web Content Accessibility Guidelines</i>
WHO	<i>World Health Organization</i>

Sumário

1	Introdução	15
1.1	Contextualização e Motivação	15
1.2	Objetivos	16
1.2.1	Geral	16
1.2.2	Específicos	17
1.3	Metodologia	17
1.4	Organização do Documento	17
2	Fundamentação Teórica	18
2.1	Diabetes Mellitus	18
2.2	Deficiência Visual	18
2.3	Acessibilidade e Tecnologia Assistiva	19
2.4	Diretrizes de Acessibilidade	20
2.5	Ferramentas Relacionadas à Acessibilidade	21
2.5.1	Leitores de tela	21
2.5.2	Testes Automatizados	22
2.6	Desenvolvimento de aplicações móveis multiplataforma	23
2.6.1	Flutter	24
2.7	Considerações Finais	25
3	Mapeamento Sistemático	26
3.1	Protocolo de Mapeamento Sistemático	26
3.1.1	Bases de Dados	27
3.1.2	<i>String</i> de Busca	27
3.1.3	Critérios de Inclusão e Exclusão	29
3.1.4	Fase de Extração	32
3.2	Resultados Encontrados	35
3.2.1	<i>A Mobile Educational Game Accessible to All, Including Screen Reading Users on a Touch-Screen Device</i>	35
3.2.2	<i>A Model-Driven Approach to Cross-Platform Development of Accessible Business Apps</i>	36
3.2.3	<i>An Accessible Roller Coaster Simulator for Touchscreen Devices: An Educational Game for the Visually Impaired</i>	37
3.2.4	<i>Application for the Configuration and Adaptation of the Android Operating System for the Visually Impaired</i>	37
3.2.5	<i>Blind and visually impaired user interface to solve accessibility problems</i>	38

3.2.6	<i>Design and development of a mobile app of drug information for people with visual impairment</i>	39
3.2.7	<i>Designing multimodal mobile interaction for a text messaging application for visually impaired users</i>	40
3.2.8	<i>Do You like My Outfit? Cromnia, a Mobile Assistant for Blind Users</i>	41
3.2.9	<i>Improved and Accessible E-Book Reader Application for Visually Impaired People</i>	42
3.2.10	<i>MathMelodies 2: A Mobile Assistive Application for People with Visual Impairments Developed with React Native</i>	43
3.2.11	<i>Object Recognition and Hearing Assistive Technology Mobile Application Using Convolutional Neural Network</i>	44
3.2.12	<i>QUIMIVOX MOBILE 2.0: Application for Helping Visually Impaired People in Learning Periodic Table and Electron Configuration</i>	45
3.2.13	<i>“Talkin’ about the weather”: Incorporating TalkBack functionality and sonifications for accessible app design</i>	46
3.2.14	<i>Users’ perception on usability aspects of a braille learning mobile application ‘mBRAILLE’</i>	47
3.2.15	<i>WordMelodies: Supporting Children with Visual Impairment in Learning Literacy</i>	48
3.3	Estudos Relacionados	50
3.3.1	<i>Accessibility of Mobile Applications: Evaluation by Users with Visual Impairment and by Automated Tools</i>	50
3.3.2	<i>Can Everyone use my app? An Empirical Study on Accessibility in Android Apps</i>	51
3.3.3	<i>Effect of UX Design Guideline on the information accessibility for the visually impaired in the mobile health apps</i>	52
3.3.4	<i>Mobile Device Accessibility for the Visually Impaired: Problems Mapping and Empirical Study of Touch Screen Gestures</i>	53
3.3.5	<i>Observation Based Analysis on the Use of Mobile Applications for Visually Impaired Users</i>	56
3.3.6	<i>Prioritization of mobile accessibility guidelines for visual impaired users</i>	57
3.4	Análise dos Resultados	59
3.4.1	<i>CPER1: Feedback auditivo não é suficiente para a interação</i>	62
3.4.2	<i>CPER2: Apresentação dos conteúdos</i>	62
3.4.3	<i>CPER3 e CPER8: Problemas relacionados às descrições dos elementos e leitores de tela</i>	63
3.4.4	<i>CPER4: Dificuldades ao navegar pela aplicação</i>	63
3.4.5	<i>CPER5: Dificuldades com a utilização do teclado virtual padrão</i>	64
3.4.6	<i>CPER6: Dificuldades relacionadas a botões virtuais</i>	64

3.4.7	CPER7: Dificuldades na utilização de gestos	64
3.4.8	CPER9: Funcionalidades confusas ou não claras	64
3.4.9	CPER10: Obstáculos relacionados ao reconhecimento de voz	65
3.5	Respostas das Questões do Protocolo	66
3.6	Técnicas Propostas para o DiaVision	67
3.7	Considerações Finais	67
4	Planejamento do DiaVision	68
4.1	Descrição do Projeto	68
4.2	Busca de Anterioridade	68
4.3	Visão e Análise	69
4.3.1	Descrição do Problema	69
4.3.2	Riscos e Impedimentos	70
4.4	Requisitos	70
4.4.1	Estórias de usuários	71
4.4.2	Requisitos Funcionais	72
4.4.3	Requisitos Não-Funcionais	73
4.4.4	Requisitos Inversos	73
4.4.5	Casos de Uso	74
4.4.6	Diagrama de Classes	75
4.4.7	Protótipo de Telas	76
4.5	Considerações Finais	77
5	Desenvolvimento do DiaVision	78
5.1	Tecnologias Utilizadas	78
5.1.1	Visual Studio Code	78
5.1.2	Github	79
5.1.3	Ambiente de Desenvolvimento	79
5.1.4	Dispositivos de Teste	80
5.1.5	Back4app	80
5.1.6	Flutter	81
5.2	Implementação	81
5.2.1	Arquitetura	81
5.2.2	<i>Backend</i>	84
5.2.2.1	Classes do Banco de Dados	85
5.2.2.2	<i>Dashboard</i> do Administrador do Sistema	86
5.2.3	Aplicativo	89
5.2.3.1	Introdução	89
5.2.3.2	Autenticação	92
5.2.3.3	Tela Principal	94

5.2.3.4	Configurações	95
5.2.3.5	Notificações	96
5.2.3.6	Módulo Alimentação	97
5.2.3.7	Módulo <i>Apps</i> de Visão	100
5.2.3.8	Módulo Atividade Física	101
5.2.3.9	Módulo Autocuidado	102
5.2.3.10	Módulo Centros de Saúde	103
5.2.3.11	Módulo Glicemia	104
5.2.3.12	Módulo Medicações	105
5.2.3.13	Módulo Pés	107
5.2.3.14	Módulo Rins	108
5.3	Considerações Finais	109
6	Considerações Finais e Trabalhos Futuros	110
	Referências	111

1

Introdução

1.1 Contextualização e Motivação

O *Diabetes Mellitus* (DM) vem tornando-se um desafio global de saúde pública cada vez maior devido ao rápido aumento no número de casos nos últimos 20 anos ([ADA, 2019](#)). Estimativas da Federação Internacional de Diabetes (IDF), do Atlas do Diabetes de 2019, apontaram que 463 milhões de pessoas no mundo viviam com DM, o que representa cerca de 9,3% da população global adulta, e é esperado um aumento para 10,2% (578 milhões) em 2030 e 10,9% (700 milhões) em 2045 ([SAEEDI et al., 2019](#)).

O Brasil é o 5º país com mais diabéticos no mundo com 16,8 milhões em 2019, na faixa etária de 20 a 79 anos, e estimativas de 21,5 e 26 milhões de casos para 2030 e 2045, respectivamente ([SAEEDI et al., 2019](#)). Os custos totais de hipertensão, diabetes e obesidade no Sistema Único de Saúde (SUS) alcançaram 3,45 bilhões de reais em 2018, sendo 30% desse custo relacionado ao DM ([NILSON et al., 2020](#)).

Já a retinopatia diabética é uma complicação vascular do diabetes, cuja prevalência está diretamente relacionada à duração do diabetes e ao controle do nível de glicemia ([SOLOMON et al., 2017](#)). Essa complicação é a maior causa de novos casos de cegueira em adultos, na faixa etária de 20 a 74 anos, em países desenvolvidos ([ADA, 2019](#)). Além disso, outros distúrbios oculares como o glaucoma e a catarata ocorrem mais cedo e com maior frequência em diabéticos ([ADA, 2019](#)).

Diante disso, atualmente o mundo enfrenta um sério problema com relação à saúde da visão. Segundo a Organização Mundial da Saúde (OMS), pelo menos 2,2 bilhões de pessoas no mundo vivem com deficiência visual (DV) em algum grau, com isso, a necessidade de cuidados com os olhos tende a crescer drasticamente nas próximas décadas ([WHO, 2019b](#)).

Segundo a Associação Americana de Diabetes (ADA), serviços de educação e apoio para

o autogerenciamento do diabetes (DSMES, do inglês *diabetes self-management education and support*) facilitam na aquisição de conhecimento e habilidades necessárias para o autocuidado, incorporando as necessidades, objetivos e experiências de diabéticos (ADA, 2019). Assim, DSMES visam auxiliar o empoderamento dos pacientes com diabetes na tomada de decisões informadas de autogerenciamento (MARRERO et al., 2013).

Além disso, conforme estimativas de 2019, mais de 5 bilhões de pessoas no mundo possuem dispositivos móveis, sendo mais da metade destes, *smartphones* (TAYLOR; SILVER, 2019). Embora tenha sido apontando em Morris, Sweatman e Jones (2017) que cerca de 84% da população estadunidense com DV possua ou utilize telefone celular, a taxa média de indivíduos que os possuem nos países menos desenvolvidos é de apenas 61% (ITU, 2021).

Já no Brasil, a taxa de adultos que relataram possuir dispositivos móveis foi de 83% no total e 60% para *smartphones*, sendo que, na faixa etária entre 18 e 34 anos, houve um aumento no número de proprietários de *smartphones* de 61% em 2015 para 85% em 2018 (TAYLOR; SILVER, 2019).

Contudo, Yan e Ramachandran (2019) elaboraram um estudo abrangente, realizado com 479 *apps* de 23 categorias da Google Play, utilizando uma ferramenta automatizada, o IBM *Mobile Accessibility Checker* (MAC), para encontrar possíveis problemas com acessibilidade à pessoas com DV (PDV) nesses *apps*, categorizando-os em V (Violação), PV (Potencial Violação) e A (Alerta). Os resultados mostraram que 94.8%, 97.5% e 66.4% dos *apps* continham problemas associados à V, PV e A, respectivamente (YAN; RAMACHANDRAN, 2019).

Diante da popularização dos *smartphones* e das problemáticas mencionadas, o presente trabalho visou desenvolver o DiaVision, um aplicativo móvel com DSMES para o autocuidado de pacientes diabéticos, considerando a solução dos principais problemas de acessibilidade à PDV.

1.2 Objetivos

Nesta seção são apresentados os objetivos, divididos em geral, que traz uma visão mais ampla sobre o objeto de estudo, e específicos, que visam aprofundar as intenções expressas no geral (CERVO; BERVIAN; SILVA, 2006).

1.2.1 Geral

O objetivo deste trabalho foi desenvolver um aplicativo móvel (APM) multiplataforma e assistivo, por meio da implementação de soluções para os principais problemas de acessibilidade à PDV na utilização de APMs, voltado ao autocuidado de pessoas com diabetes e acuidade visual prejudicada.

1.2.2 Específicos

Para atingir objetivo geral deste trabalho, os seguintes objetivos específicos foram definidos:

- Identificar os principais problemas enfrentados por PDV na utilização de APMs;
- Identificar as principais técnicas e soluções de acessibilidade à PDV para APMs;
- Relacionar as principais técnicas e soluções aos principais problemas identificados;
- Realizar o desenvolvimento do APM com os principais requisitos levantados;
- Aplicar as soluções identificadas para os principais problemas de acessibilidade em APMs.

1.3 Metodologia

A metodologia adotada neste estudo foi a quantitativa exploratória, com o propósito de responder as questões de pesquisa apresentadas no início do capítulo 3, obtendo assim maior familiaridade com o problema e possíveis soluções.

Para isso, foi realizado um levantamento bibliográfico por meio de um Mapeamento Sistemático da Literatura (MSL), no qual foi possível analisar os resultados dos estudos selecionados (de acordo com o protocolo) e dos relacionados (outros estudos de mapeamento).

1.4 Organização do Documento

Neste capítulo foram apresentados a contextualização e a motivação deste trabalho, os objetivos e a metodologia adotada.

Para facilitar a navegação e melhor entendimento, este documento está organizado em capítulos, cujas descrições são listadas a seguir:

- Capítulo 2 - Fundamentação Teórica: aborda os principais conceitos relacionados ao trabalho realizado;
- Capítulo 3 - Mapeamento Sistemático da Literatura: descreve os estudos selecionados e os relacionados, e apresenta uma análise dos resultados e as respostas às questões de pesquisa;
- Capítulo 4 - Planejamento do DiaVision: descreve as etapas de planejamento do projeto;
- Capítulo 5 - Desenvolvimento do DiaVision: detalha o processo de desenvolvimento e as principais tecnologias e soluções utilizadas;
- Capítulo 6 - Considerações Finais e Trabalhos Futuros: apresenta as considerações finais deste trabalho e as melhorias que podem ser realizadas em trabalhos futuros.

2

Fundamentação Teórica

Este capítulo apresenta os conceitos básicos relevantes para compreensão deste trabalho, com o objetivo de contextualizar e situar o leitor no tema abordado.

2.1 Diabetes Mellitus

O DM é um grupo de doenças endocrinológicas crônicas caracterizado pela elevação da glicose no sangue que ocorre devido à deficiência de ação do hormônio insulina, e requer cuidados médicos contínuos para redução de risco e controle glicêmico ([ADA, 2019](#)).

Esse grupo divide-se em dois tipos: o tipo 1 afeta a produção de insulina devido a uma reação autoimune às proteínas das células das ilhotas do pâncreas, enquanto que o tipo 2 afeta o processamento do açúcar no sangue, e é causado por fatores genéticos relacionados à secreção prejudicada e resistência à insulina, além de fatores ambientais como obesidade, alimentação excessiva, falta de exercício, estresse e o envelhecimento ([OZOUGWU, 2013](#)).

Segundo a [ADA \(2019\)](#), estudos mostraram que DSMES estão associados ao aumento no conhecimento e comportamentos de autocuidado sobre o diabetes e no número de auto-relatos de redução de peso, à melhoria de qualidade de vida, à redução de risco de mortalidade em todas as causas e à redução nos custos de cuidados com a saúde.

Resultados mais recentes, de uma metanálise realizada em [Aminuddin et al. \(2021\)](#), também apontaram que as intervenções de autogerenciamento baseadas em *smartphones* teriam efeitos benéficos sobre atividades de autocuidado, principalmente para pacientes com DM tipo 2.

2.2 Deficiência Visual

De acordo com a Classificação Internacional de Funcionalidade, Incapacidade e Saúde (ICF), a incapacidade enfrentada por pessoas com deficiência visual (PDV) não é determinada

apenas pela condição ocular, mas também pelo ambiente físico e social em que a pessoa vive, bem como as dificuldades que pode enfrentar para realização de atividades de autocuidado, os problemas que sofrem cotidianamente e o acesso a cuidados, produtos e serviços oftalmológicos ([WHO, 2019b](#)).

Segundo a [WHO \(2019b\)](#), mais de 1 bilhão dos casos de pessoas com deficiência visual (DV) poderiam ser prevenidos ou tratados. A publicação ainda aponta que os principais motivos para esses casos são:

- O tempo despendido em ambientes fechados e aumento das atividades “*near work*” (ler, escrever, assistir TV, jogar videogames, etc);
- O aumento no número de pessoas vivendo com diabetes, principalmente o tipo 2;
- Muitas pessoas não terem acesso a serviços oftalmológicos e exames de rotina.

No Brasil, conforme o último censo do Instituto Brasileiro de Geografia e Estatística (IBGE), realizado em 2010, cerca de 18,6% da população era afetada por algum tipo de DV, sendo 3,46% por DV severa ([IBGE, 2012](#)). Embora o próximo censo esteja previsto para 2022, outra pesquisa foi realizada pelo Ministério da Saúde em 2019, a Pesquisa Nacional de Saúde (PNS), e apontou que 3,4% da população brasileira, com 2 ou mais anos de idade, possui muita dificuldade para enxergar ou não enxerga ([STOPA et al., 2020](#)).

2.3 Acessibilidade e Tecnologia Assistiva

Segundo o Art. 3º da Lei Brasileira de Inclusão da Pessoa com Deficiência, acessibilidade refere-se à:

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 ([BRASIL, 2015](#)).

Assim, visando essa inclusão, tecnologias conhecidas como Tecnologias Assistivas (TA) se tornam cada vez mais presentes. [Cook e Polgar \(2014\)](#) adotam em seu livro uma definição de TA criada por uma *Public Law* dos Estados Unidos da América (EUA) que é mundialmente utilizada, pois ela contempla os pontos mais importantes a respeito de TA, como diz a seguir:

Qualquer item, parte de equipamento ou **sistema** adquirido comercialmente, modificado ou customizado que é utilizado para aumentar, manter ou melhorar as capacidades funcionais de pessoas com deficiência ([COOK; POLGAR, 2014](#)).

De acordo com a OMS, o acesso à TAs adequadas e de qualidade por um preço acessível melhora no desenvolvimento das funções e na independência de indivíduos com deficiências, ao mesmo tempo que facilita a participação e integração dos mesmos na sociedade (WHO, 2019a).

2.4 Diretrizes de Acessibilidade

Para que as TAs funcionassem adequadamente na *web*, a *World Wide Web Consortium* (W3C) definiu, por meio da *Web Accessibility Initiative* (WAI), um conjunto de diretrizes e recomendações de acessibilidade, chamado *Web Content Accessibility Guidelines* (WCAG) que deveriam ser seguidas no desenvolvimento de aplicações *web* (W3C, 2019).

Com o advento da navegação na *web* em dispositivos móveis, a W3C lançou, em novembro de 2006, o *Mobile Web Best Practices*(MWBP)¹, com o objetivo de melhorar a experiência do usuário ao acessar a *web* nesses dispositivos.

Além da W3C, órgãos de governos também desenvolveram diretrizes e recomendações de acessibilidade, baseados no WCAG e em suas próprias legislações, como foi o caso dos EUA com a Seção 508 da Lei de Reabilitação de 1973, que exige acessibilidade de todas as agências federais quando desenvolvem, adquirem, mantêm ou usam tecnologia eletrônica e de informação (JAEGER, 2006). Contudo, estudos apontam que diversos sites governamentais não estão em total conformidade com a Seção 508 (KING; YOUNGBLOOD, 2016; YI, 2015).

Já no Brasil, o Decreto 5.296/04 estabeleceu o cumprimento dos requisitos de acessibilidade pelos órgãos da administração pública direta, indireta e fundacional, pelas empresas prestadoras de serviços públicos e instituições financeiras, resultando na criação do Modelo de Acessibilidade em Governo Eletrônico (eMAG) (EMAG, 2014).

Lançada em 2005, a primeira versão do eMAG também foi desenvolvida com base no WCAG 1.0. Estudos avaliaram as diferenças entre as diretrizes do eMAG (versões 1.0 e 3.0) e as do WCAG (versões 1.0 e 2.0), observando-se que as versões possuíam poucas diferenças nas recomendações, sendo as maiores diferenças estruturais, demonstrando vantagens na adoção do eMAG para o contexto brasileiro (BACH et al., 2009; ROCHA; DUARTE, 2013).

Entretanto, para aplicações nativas *mobile*, não existe uma entidade como a W3C para definir essas diretrizes e, embora ela tenha proposto o MWBP, este refere-se à criação de aplicações *web* para o *mobile*, não contemplando todas as necessidades das aplicações nativas (W3C, 2008).

Assim, as próprias empresas responsáveis pelos Sistemas Operacionais (SO) *mobile*, Google (Android) e Apple (iOS), criaram suas diretrizes e recomendações para o desenvolvimento de aplicativos acessíveis para suas plataformas. Apesar disso, muitos desenvolvedores não possuem

¹ O Guia de Boas Práticas em Web Móvel reúne padrões web do W3C para ajudar no desenvolvimento de conteúdos web para que funcionem adequadamente em dispositivos móveis. <<http://www.w3.org/TR/mobile-bp/>>

conhecimento sobre essas técnicas e recomendações, ou sobre a necessidade delas para o suporte das aplicações à pessoas com deficiências ([QUISPE; SCATALON; ELER, 2020](#); [BI et al., 2021](#)).

Um estudo realizado por [Ballantyne et al. \(2018\)](#) compila um conjunto de diretrizes para acessibilidade *mobile* e realiza testes em 25 dos *apps* mais populares da Google Play. Os resultados do estudo revelaram que apenas 8 dos 25 selecionados possuíam taxa de conformidade com as diretrizes acima de 75%. O estudo ainda revela que 63% das violações encontradas estavam relacionadas ao *design* (componentes de tela).

Para [Quispe, Scatalon e Eler \(2020\)](#) os principais fatores para a baixa priorização da acessibilidade de aplicações *mobile* são o desconhecimento, a alta demanda e a falta de tempo das equipes de desenvolvimento, fazendo com que se concentrem nos requisitos funcionais em detrimento de requisitos não funcionais de usabilidade como o de acessibilidade.

2.5 Ferramentas Relacionadas à Acessibilidade

Nesta seção são apresentadas ferramentas que podem ser utilizadas durante o desenvolvimento do projeto, com o objetivo de melhorar e validar a usabilidade da aplicação quanto à acessibilidade.

2.5.1 Leitores de tela

Com o objetivo de possibilitar a utilização dos *smartphones* por usuários cegos e auxiliar os com DV parcial, o Android e o iOS fornecem nativamente os leitores de tela chamados TalkBack e VoiceOver, respectivamente.

A descrição do VoiceOver pela Apple:

Com o VoiceOver – um leitor de tela baseado em gestos – você pode usar o iPhone mesmo que você não possa ver a tela. O VoiceOver fornece descrições audíveis do que está na tela — desde o nível da bateria até quem está ligando e em qual app o seu dedo está. Você também pode ajustar a velocidade da fala e o tom de voz conforme as suas necessidades ([APPLE, 2021](#)).

Descrição do TalkBack pelo Google:

O TalkBack é o leitor de tela do Google incluído em dispositivos Android. Ele permite que você controle o dispositivo sem usar os olhos ([GOOGLE, 2021](#)).

Enquanto apenas a Apple fabrica os *smartphones* que utilizam o iOS, o Google mantém o Android, mas cada fabricante pode escolher se deseja personalizá-lo com as características da marca ou se mantém o sistema fiel às definições padrões. Diante disso, a documentação ressalta:

A configuração depende do fabricante do dispositivo, da versão do Android e da versão do TalkBack. Estas páginas de ajuda se aplicam à maioria dos dispositivos, mas pode haver algumas diferenças (GOOGLE, 2021).

Esses leitores de tela utilizam uma técnica conhecida como *text-to-speech* (TTS, “texto para discurso” em tradução livre) para narrar as descrições para os usuários. Dessa forma, tanto o Android quanto o iOS disponibilizam Interfaces de Programação de Aplicações (APIs, do inglês “Application Programming Interface”) para que os desenvolvedores possam integrar essa tecnologia diretamente aos próprios *apps*, não sendo necessário habilitar os leitores nos dispositivos para funcionar nessas aplicações específicas (SHIN et al., 2017; BIASE et al., 2018; OLIVEIRA et al., 2019; CABALLERO; CATLI; BABIERRA, 2020).

Para o funcionamento adequado desses leitores de tela nas aplicações, é necessário que haja um tratamento com relação aos componentes de *interface*, com o fornecimento de descrições que possam ser narradas para o usuário. Por conta disso, problemas associados à falta de descrições desses componentes costumam ser os mais frequentes enfrentados pelos usuários com DV (VENDOME et al., 2019; RIEGER et al., 2020; SHERA et al., 2021).

Uma boa prática para mitigação desses problemas pode ser a utilização constante dos leitores de tela para validação do fluxo das aplicações durante o processo de desenvolvimento, visando diminuir a quantidade, visto que podem ser identificados e corrigidos mais cedo (TOMLINSON et al., 2016).

2.5.2 Testes Automatizados

A utilização de ferramentas automatizadas para realização de testes de usabilidade e acessibilidade pode reduzir o esforço e o retrabalho, pois identifica diversos problemas ainda em tempo de desenvolvimento, diminuindo os esforços e custos para realização das correções (RIEGER et al., 2020).

Uma dessas ferramentas é a *Mobile Accessibility Testing* (MATE), desenvolvida por brasileiros, que automaticamente explora os *apps* aplicando diferentes tipos de checagem por problemas de acessibilidade relacionadas à DV, gerando um relatório detalhado para auxiliar os desenvolvedores na resolução desses problemas (ELER et al., 2018).

O Google também possui duas ferramentas que permitem a realização desses testes, o *Accessibility Scanner*, uma aplicação instalável no dispositivo, disponível na Google Play, que sobrepõe o *app* a ser testado e faz sugestões de melhorias de acessibilidade, e o Test Lab, no qual é possível realizar o *upload* do *app* para realização dos testes pela ferramenta, disponível no Firebase².

² O Firebase é uma plataforma do Google que oferece serviços de backend, monitoramento e engajamento que facilitam o processo de desenvolvimento de aplicações móveis e web. <<https://firebase.google.com/>>

A ferramenta MAC da IBM possibilita a realização de testes automatizados em aplicações móveis nativas e em conteúdos *web mobile*, gerando alertas sobre problemas de acessibilidade com recomendações de correções baseadas em diretrizes padrões da indústria e regulamentações governamentais (PATIL; BHOLE; SHETE, 2016; YAN; RAMACHANDRAN, 2019).

2.6 Desenvolvimento de aplicações móveis multiplataforma

Conforme pesquisa realizada pelo Centro de Estudos sobre as Tecnologias da Informação e da Comunicação (CETIC), 98% dos usuários de Internet brasileiros, com 16 anos ou mais, utilizavam telefone celular para acessar a Internet em 2020, essa taxa foi de 97% em 2018 e 99% em 2019 (CETIC.BR, 2021). A pesquisa ainda mostrou que 40% desses usuários buscaram por informações ou realizaram serviços públicos *online* relacionados aos direitos do trabalhador ou previdência social em 2019, e esse número aumentou para 72% em 2020, durante a pandemia de COVID-19 (CETIC.BR, 2021).

Quanto à realização de atividades remotas, 87% dos usuários que declararam frequentar escola ou universidade, no momento da coleta dos dados dessa pesquisa, afirmaram que a instituição na qual estudaram ofereceu atividades educacionais remotas, com o telefone celular sendo o dispositivo mais utilizado para o acompanhamento dessas atividades pelas classes D e E (CETIC.BR, 2021). Os resultados dessa pesquisa também mostraram que 84% dos usuários das classes D e E utilizaram principalmente o celular para realização de atividades profissionais de forma remota durante a pandemia.

Embora aplicativos móveis possam ser desenvolvidos até por desenvolvedores amadores, plataformas móveis são muito complexas, tanto o SO Android quanto iOS, os principais para *smartphones* da atualidade, contém mais de 12 milhões de linhas de código (LOC, do inglês “*lines of code*”) (PRESSMAN; MAXIM, 2014).

O alto número de *apps* disponíveis faz com que o processo de desenvolvimento de *software* para esses dispositivos pareça ter sido bem compreendido, porém ainda existe um grande número de questões que precisam ser resolvidas (PRESSMAN; MAXIM, 2014; WASSERMAN, 2010).

Os requisitos não funcionais de aplicações móveis como a usabilidade, por exemplo, são diferentes de aplicações *web* ou *desktop* (PRESSMAN; MAXIM, 2014). Existe um “ângulo” *mobile* para praticamente todo aspecto de engenharia de *software*, em que as características das aplicações e seus SOs apresentam um novo ou diferente conjunto de questões que precisam ser consideradas (WASSERMAN, 2010).

Como mencionado por Pressman e Maxim (2014), diferentes dispositivos móveis utilizam diferentes SOs e, consequentemente, diferentes ambientes e ferramentas de desenvolvimento para cada plataforma, destacando a importância de se considerar portabilidade ao desenvolver

aplicações para esses dispositivos.

Devido à tais questões, o tempo e o esforço de desenvolvimento necessários para oferecer suporte à múltiplas plataformas aumenta significativamente, elevando os custos do projeto (HEITKÖTTER; HANSCHKE; MAJCHRZAK, 2013; WASSERMAN, 2010). Nesse sentido, diversos kits de ferramentas de desenvolvimento (mais conhecidos pelo termo em inglês: *framework*), foram propostos, visando simplificar esse processo, diminuindo os custos de desenvolvimento e manutenção (MARTINEZ; LECOMTE, 2017; FRANCSESE et al., 2015).

2.6.1 Flutter

Com a primeira versão estável sendo lançada em dezembro de 2018, o Flutter é um kit de ferramentas de Interface de Usuário (UI, do inglês *User Interface*) de código aberto criado pelo Google para construção de aplicações compiladas nativamente para múltiplas plataformas, tais como *mobile* (Android e iOS), *web* e *desktop* (Linux, MacOS e Windows) (KUZMIN; IGNATIEV; GRAFOV, 2020).

O *framework* foi construído com base na linguagem de programação Dart, também criado pelo Google e de código aberto. A linguagem foi lançada em 2011 com o objetivo inicial de “substituir” o JavaScript no desenvolvimento *web*, com diferenças como mecanismos de abstrações e semântica mais clara, visando a coesão e elegância de código e rodando tanto no lado do cliente (navegadores) quanto no lado servidor (*backend*) (WALRATH; LADD, 2012).

Um dos diferenciais do Flutter com relação a maioria dos *frameworks* para desenvolvimento multiplataforma é que ele renderiza os próprios componentes de *interface*, sem utilizar os componentes nativos de cada plataforma como os demais, ou seja, quando você constrói um botão com Flutter, ele mesmo renderiza, sem a necessidade de uma “ponte” com os SOs para solicitar a renderização do componente nativo (ZAMMETTI, 2019; BOUKHARY; COLMENARES, 2019).

O motor (do inglês, *engine*) do Flutter é baseado, em sua maior parte, no C++, linguagem nativa utilizada pelos SOs (Android, iOS, Windows etc), assim, possibilitando alcançar desempenho próximo ao de aplicações nativas (ZAMMETTI, 2019; KUZMIN; IGNATIEV; GRAFOV, 2020). Já o motor gráfico utilizado por essa base de código é o Skia, uma compacta biblioteca gráfica de código aberto que apresenta excelente desempenho nas plataformas suportadas (ZAMMETTI, 2019; BOUKHARY; COLMENARES, 2019).

O Flutter também fornece uma *interface* sobre os kits de desenvolvimento de *software* (SDKs, do inglês “Software Development Kit”) nativos de ambas as plataformas (Android e iOS), esta que tem como objetivo eliminar as diferenças entre as APIs nativas de cada plataforma (ZAMMETTI, 2019). Dessa forma, não é necessário que os desenvolvedores se preocupem com o funcionamento nativo das APIs como a de câmera de cada plataforma, podendo utilizar a do Flutter diretamente, pois ela abstrairá toda essa parte, ficando responsável por fazer a chamada à API adequada (ZAMMETTI, 2019).

No trabalho de mestrado de Gonsalves (2019), um protótipo de aplicação móvel foi desenvolvido para Android e iOS nativamente, e com os *framework* de desenvolvimento multiplataforma Flutter e Cordova, visando realizar um comparativo de aspectos do processo de desenvolvimento e dos resultados das aplicações.

Os resultados do estudo mostraram que, no geral, o Flutter oferece uma melhor experiência de desenvolvimento que o Cordova e as plataformas nativas (considerando desenvolver para iOS e Android). Ambos os *frameworks* multiplataforma apresentaram menos LOC que os *apps* nativos separados, sendo pouco mais de 1/3 do total de LOC dos *apps* nativos (GONSALVES, 2019).

Um ponto negativo do Flutter foi o tamanho significativamente maior do *app* em megabytes (MB) e maior consumo de RAM³ que os *apps* em Cordova e nativos, porém o Flutter também mostrou maior velocidade na inicialização do *app* e na navegação entre as telas que os demais (GONSALVES, 2019). O maior tamanho e consumo de RAM no *app* desenvolvido com Flutter dá-se pela inclusão do seu motor, das bibliotecas utilizadas e de outros recursos no *app* (GONSALVES, 2019; ZAMMETTI, 2019).

2.7 Considerações Finais

Este capítulo buscou trazer elucidações sobre o DM e a relação com DV, acessibilidade, diretrizes e ferramentas, finalizando sobre o desenvolvimento de aplicativos móveis multiplataforma. O próximo capítulo abordará o processo de revisão da literatura, no qual foi realizado um levantamento das principais soluções utilizadas na implementação de acessibilidade em aplicações móveis.

³ A memória de Acesso Randômico (do inglês *Random Access Memory*) é onde ficam armazenados os programas básicos operacionais de sistemas eletrônicos digitais.

3

Mapeamento Sistemático

Este capítulo abordará o processo de Mapeamento Sistemático adotado neste estudo, bem como os resultados e a análise dos mesmos.

3.1 Protocolo de Mapeamento Sistemático

O termo Mapeamento Sistemático da Literatura (MSL) refere-se a um estudo secundário que realiza uma revisão ampla de estudos primários existentes em um tema específico e visa identificar as evidências disponíveis nessa área (KITCHENHAM; CHARTERS, 2007). O método de Mapeamento Sistemático adotado foi o de Kitchenham, descrito por Silva (2009). Seguindo o método, foi desenvolvido um protocolo de revisão com intuito de responder as seguintes questões:

1. Quais são as principais soluções de acessibilidade para PDV utilizadas no desenvolvimento de aplicações móveis?
2. Quais foram as principais tecnologias utilizadas no desenvolvimento dessas soluções?
3. Para quais plataformas as soluções foram propostas?
4. Quem são os públicos alvos dessas soluções?

O *Parfisal*¹, ferramenta *online* que auxilia no desenvolvimento de Revisões Sistemáticas da Literatura, foi utilizado neste estudo. Com ele foi possível importar os resultados das buscas nas bases, identificar os artigos duplicados, definir os critérios para inclusão e exclusão, realizar a seleção dos estudos e, por fim, obter os relatórios para construção dos artefatos que apresentam o processo e os resultados desse MSL.

¹ <<https://parsif.al/>>

3.1.1 Bases de Dados

Cinco bases de dados científicos foram escolhidas neste trabalho, a *IEEE Xplore*², na qual estão disponíveis conteúdos técnicos e científicos publicados pelo *Institute of Electrical and Electronics Engineers (IEEE)* e seus parceiros, a *Scopus*³, que é mantida pela *Elsevier* e combina um abrangente banco de dados de resumos e citações de literatura acadêmica em diversas áreas, a *ScienceDirect*⁴, principal plataforma da *Elsevier* de literatura revisada por pares, a *ACM Digital Library*⁵, uma plataforma de pesquisa que conta com textos completos de todas as publicações da *Association for Computing Machinery (ACM)* e de uma curada coleção de publicações de editoras selecionadas e a *PubMed*⁶, plataforma gratuita que conta com uma base de dados com mais de 33 milhões de citações e resumos da literatura biomédica.

3.1.2 String de Busca

Para realização da busca pelos artigos, um conjunto de palavras-chave e sinônimos foi definido de acordo com o tema deste trabalho, como é mostrado no [Quadro 1](#).

Quadro 1 – Palavras-chave e Sinônimos.

Palavra-chave	Sinônimos (Inglês)
Acessibilidade	<i>Accessibility</i>
Aplicativo	<i>Application, App</i>
Deficiência visual	<i>Visual impairment, Visually impaired</i>
Móvel	<i>Mobile, Smartphone</i>

Fonte: Autor.

Uma *string* genérica de busca foi definida a partir das palavras-chave, com o objetivo de manter a consistência da busca nas diferentes bases e pode ser vista no [Quadro 2](#). As palavras-chave que possuem sinônimos aparecem na *string* entre parêntesis com o operador *OR*, visando incluir os resultados que contenham pelo menos um dos termos, para indicar que são sinônimos no contexto do tema abordado neste trabalho.

Quadro 2 – *String* genérica.

String genérica
<i>accessibility AND (“visual impairment” OR “visually impaired”) AND (mobile OR smartphone) AND (app OR application)</i>

Fonte: Autor.

² <<https://ieeexplore.ieee.org>>

³ <<https://www.scopus.com>>

⁴ <<https://www.sciencedirect.com>>

⁵ <<https://dl.acm.org>>

⁶ <<https://pubmed.ncbi.nlm.nih.gov>>

A *string* genérica serviu de modelo para criação das *strings* específicas, indicadas no [Quadro 3](#), de acordo com as particularidades de cada base de busca. As *strings* específicas buscam pelas palavras-chave nos títulos e resumos dos artigos, afim de encontrar os que focam no tema proposto neste estudo, desconsiderando os que apenas citam as palavras-chave ao longo texto.

Quadro 3 – *Strings* específicas para busca em cada base.

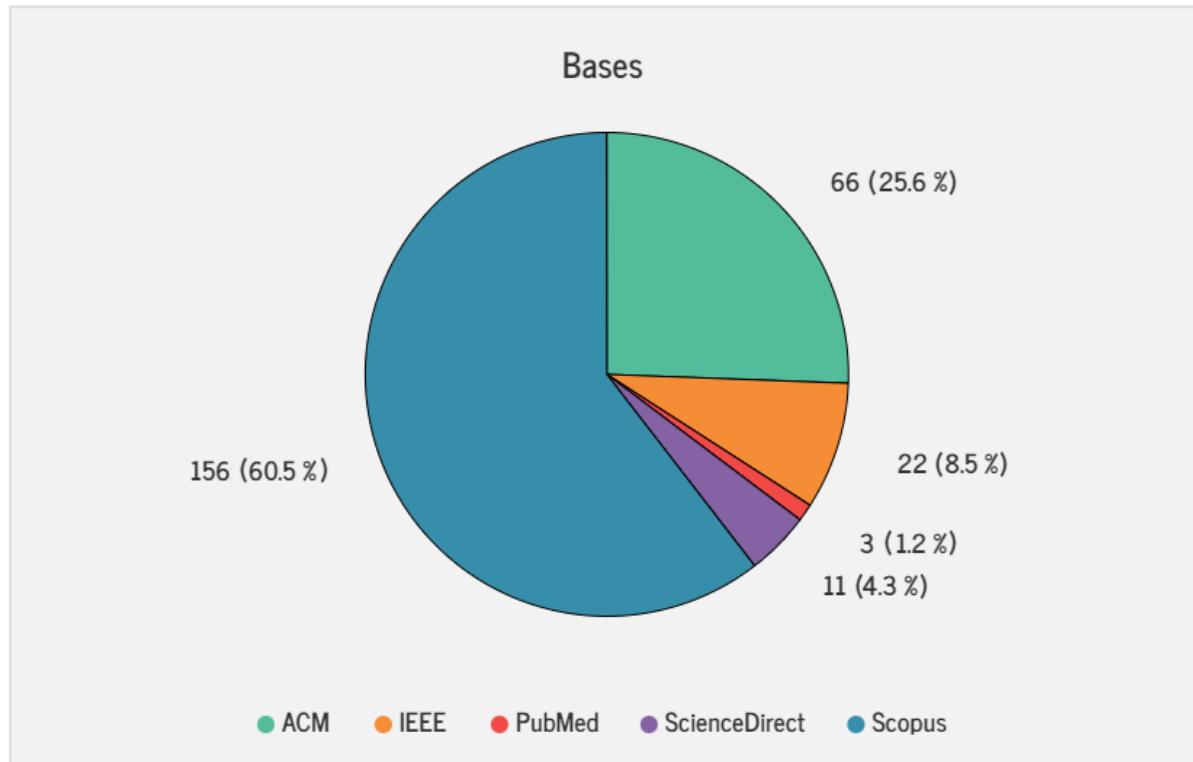
Base	String de busca
<i>ACM Digital Library</i>	(Abstract:(accessibility) OR Title:(accessibility)) AND (Abstract:(“visual impairment”OR “visually impaired”) OR Title:(“visual impairment”OR “visually impaired”)) AND (Abstract:(smartphone OR mobile) OR Title:(smartphone OR mobile)) AND (Abstract:(app OR application) OR Title:(app OR application))
<i>IEEE Xplore</i>	((“Abstract”:accessibility OR “Document Title”:accessibility) AND (“Document Title”:mobile OR “Document Title”:smartphone OR “Abstract”:mobile OR “Abstract”:smartphone) AND (“Document Title”:“visual impairment”OR “Abstract”:“visual impairment”OR “Document Title”:“visually impaired”OR “Abstract”:“visually impaired”) AND (“Document Title”:app OR “Document Title”:application OR “Abstract”:app OR “Abstract”:application))
<i>PubMed</i>	(accessibility[Abstract] OR accessibility[Title]) AND (“visually impaired”[Abstract] OR “visual impairment”[Abstract] OR “visually impaired”[Title] OR “visual impairment”[Title]) AND (mobile[Abstract] OR smartphone[Abstract] OR mobile[Title] OR smartphone[Title]) AND (app[Title] OR app[Abstract] OR application[Title] OR application[Abstract])
<i>ScienceDirect</i>	Title, abstract, keywords: accessibility AND (“visual impairment”OR “visually impaired”) AND (mobile OR smartphone) AND (app OR application)
<i>Scopus</i>	TITLE-ABS (accessibility AND (“visual impairment”OR “visually impaired”) AND (smartphone OR mobile) AND (app OR application))

Fonte: Autor.

A busca nas bases de dados selecionadas foi realizada no dia 04 de outubro de 2021 e retornou um total de 258 resultados. Como mostra a [Figura 1](#), a maior quantidade de resultados foi encontrada na *Scopus*, isso acontece porque ela possui, de acordo com seu site⁷, o maior banco de dados de resumos e citações da literatura com revisão por pares. Em seguida, aparecem a *ACM Digital Library* e a *IEEE Xplore* cujas principais publicações são na área de computação, e elétrica no caso da IEEE. Por fim, com as menores quantidades, a *ScienceDirect*, com publicações nas mais diversas áreas da ciência e a *PubMed* com publicações na área biomédica, justificando os menores resultados, visto que a busca foi realizada visando encontrar técnicas, métodos e padrões para o desenvolvimento de aplicações móveis acessíveis.

⁷ <<https://www.elsevier.com/pt-br/solutions/scopus>>

Figura 1 – Quantidade de artigos encontrados por base.



Fonte: Autor.

3.1.3 Critérios de Inclusão e Exclusão

Critérios de inclusão e exclusão foram definidos visando a coerência dos artigos selecionados com o tema deste trabalho e a remoção de artigos incompletos ou indisponíveis. Os critérios são:

Critérios de Inclusão

- O artigo deve propor método, técnica ou padrão para o desenvolvimento de aplicações móveis com acessibilidade para deficientes visuais;
- O artigo deve estar disponível na *web*;
- O artigo deve apresentar texto completo em formato eletrônico;
- O artigo deve estar escrito em português ou inglês.

Critérios de Exclusão

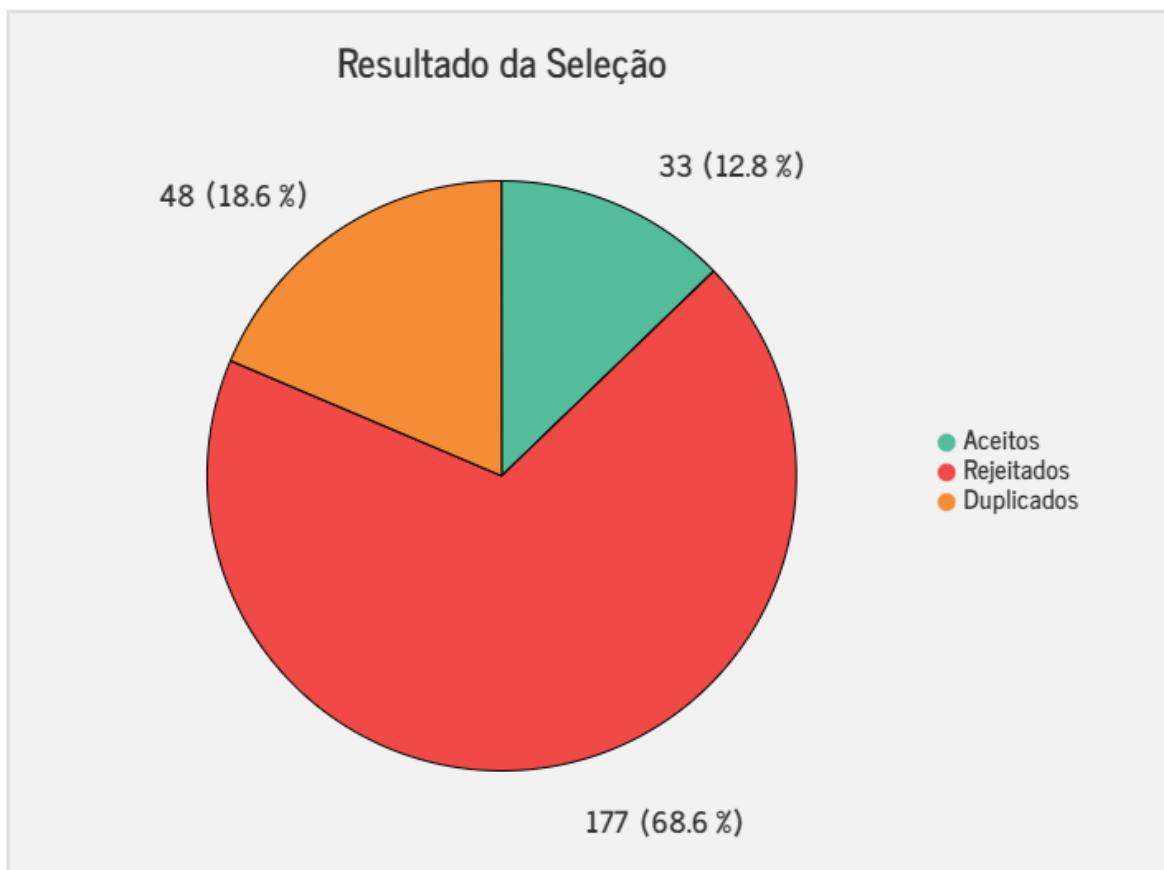
- O artigo não apresenta proposta de aplicação móvel como solução;
- O artigo não apresenta aplicativo desenvolvido no contexto do tema deste trabalho;

- O artigo é um livro ou parte de um;
- O artigo foi publicado antes de 2016;
- O artigo está incompleto, indisponível ou duplicado.

No processo de seleção dos estudos, inicialmente, foi aplicado o critério de exclusão de artigos duplicados e, em seguida, o de artigos publicados antes de 2016, rejeitando 48 e 65 artigos, respectivamente. Esses critérios foram priorizados por não haver necessidade de leitura dos títulos e resumos dos artigos para serem aplicados.

Por fim, após a leitura dos títulos e resumos, mais 112 artigos foram rejeitados, totalizando 225 artigos. Assim, sendo aceitos 33 artigos para leitura completa e análise. A [Figura 2](#) apresenta o resultado dessa seleção.

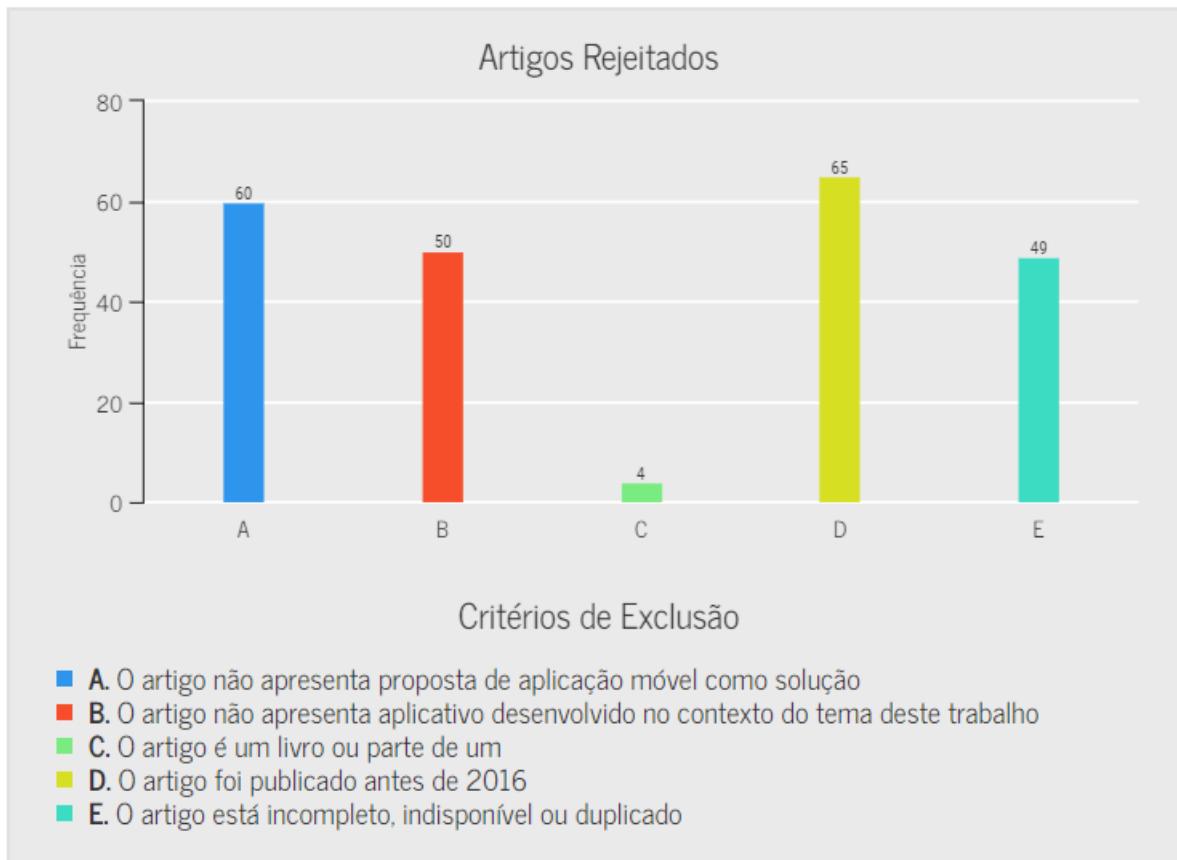
Figura 2 – Quantidade de artigos aceitos, rejeitados ou duplicados na seleção.



Fonte: Autor.

A frequência dos artigos rejeitados para cada critério de exclusão, sendo que para rejeição o artigo deveria atender a pelo menos um desses critérios, pode ser observada no gráfico da [Figura 3](#).

Figura 3 – Quantidade de artigos rejeitados por critério de exclusão.



Fonte: Autor.

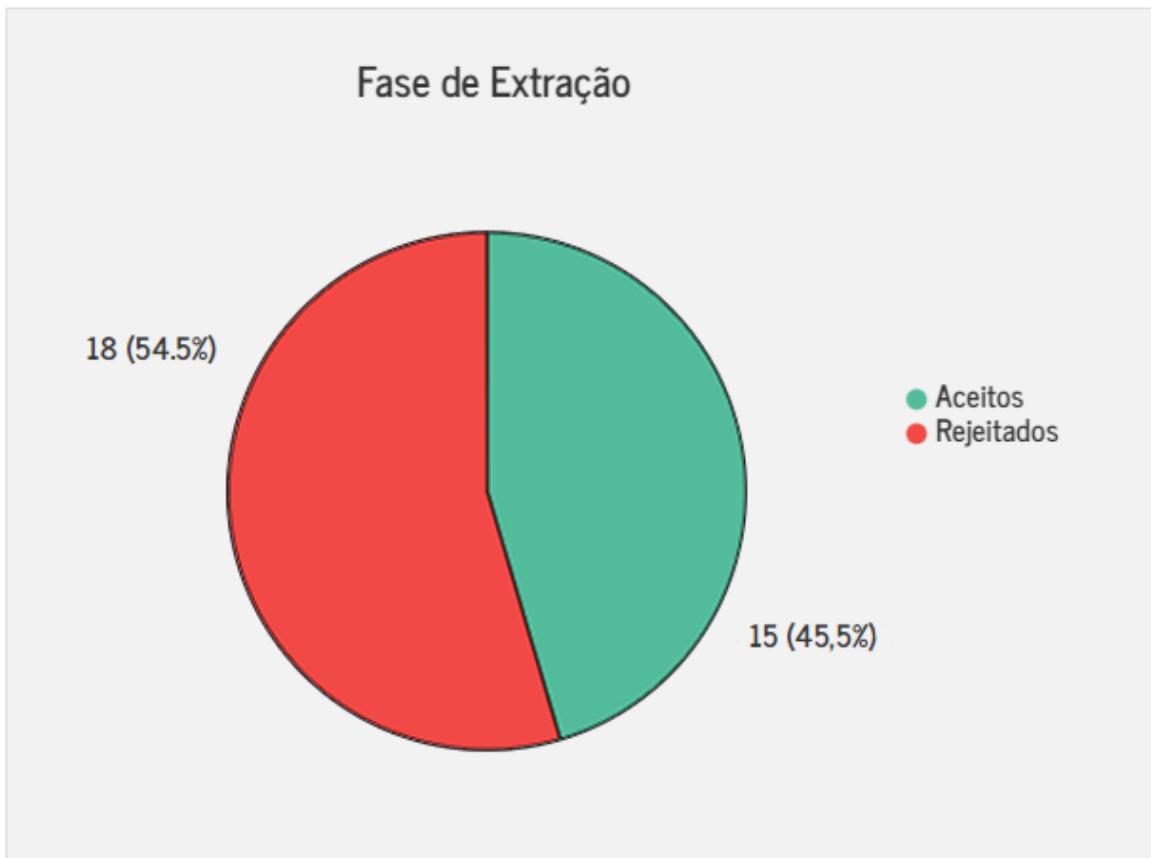
Os critérios de exclusão levaram em consideração, principalmente, aspectos como divergência com o tema deste trabalho e não apresentação de aplicação desenvolvida para dispositivos móveis. Os critérios D e E aparecem com grande frequência na [Figura 3](#), valendo ressaltar a ordem na avaliação dos critérios (E, D, A, B e C), na qual os artigos que se enquadram em um dos critérios foram rejeitados, não sendo considerados para avaliação nos demais.

É comum que o critério D seja utilizado no próprio processo de busca dos artigos, filtrando apenas os anos de interesse, porém, como não foi possível adicionar esse filtro às *strings* de busca para todas as bases, foi optado por não utilizá-lo, para manter a consistência nos resultados das buscas.

3.1.4 Fase de Extração

Durante a fase de extração, uma análise mais aprofundada dos artigos foi realizada, inicialmente com intuito de reaplicar os critérios já definidos e utilizados na fase anterior. O resultado dessa última filtragem pode ser visto na [Figura 4](#).

Figura 4 – Artigos aceitos ou rejeitados na fase de extração.

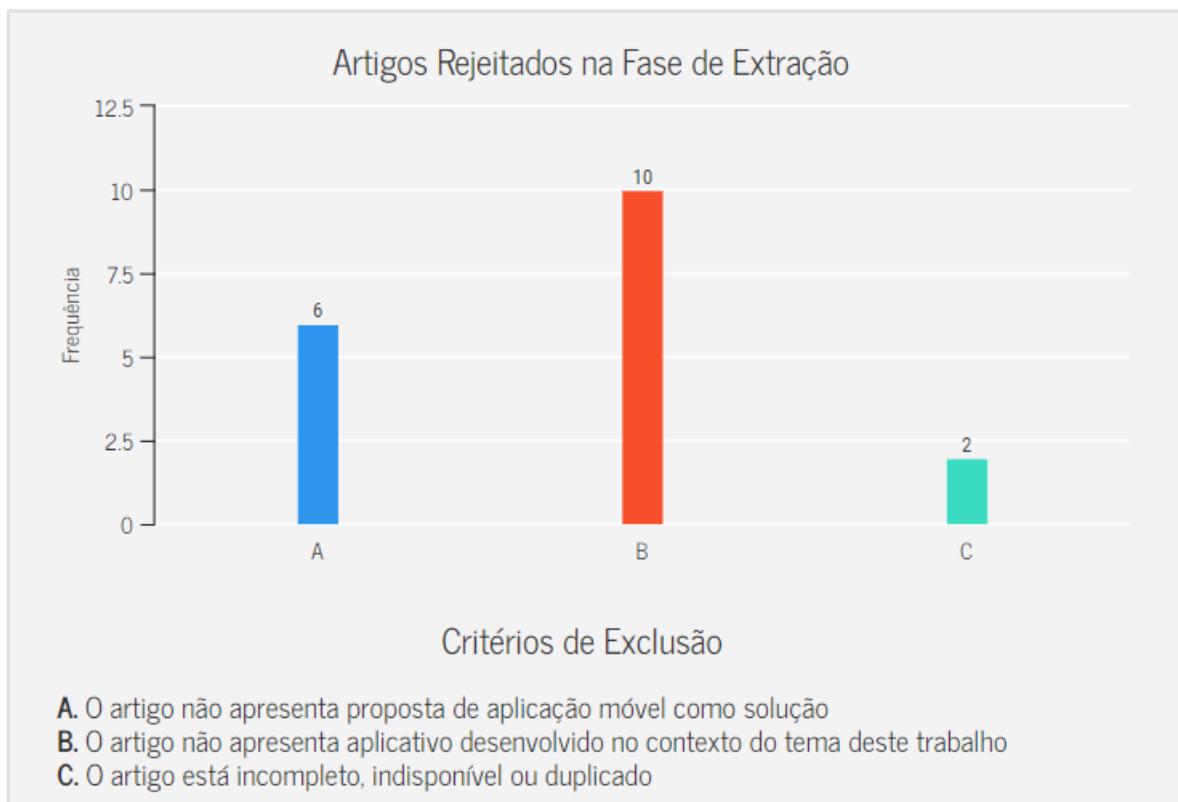


Fonte: Autor.

Como a aplicação inicial dos critérios foi realizada com base apenas na leitura dos títulos e resumos dos estudos, não foi possível garantir que os artigos aceitos realmente não se enquadram nos critérios de exclusão. Assim, com a análise mais aprofundada e leitura completa dos textos, foi possível identificar 18 artigos que se enquadram em algum desses critérios, como mostrou a [Figura 4](#).

A Figura 5 mostra a frequência de artigos que foram rejeitados por cada critério de exclusão. Apenas os critérios com frequência maior que 0 foram considerados na figura. O principal motivo para rejeição foi o B, no qual os trabalhos apresentavam aplicações móveis que não haviam sido desenvolvidas com foco na acessibilidade da aplicação em si. O segundo foi o A, este que refere-se aos estudos que não apresentavam uma aplicação móvel com acessibilidade à PDV como solução.

Figura 5 – Artigos rejeitados na fase de extração por critério exclusão.



Fonte: Autor.

Os 15 estudos aceitos na fase de extração foram reunidos no Quadro 4 com a listagem de informações como o título do estudo, referência e o nome da base de dados em que o artigo foi encontrado.

Quadro 4 – Artigos aceitos na fase de extração.

Código	Título	Referência	Base de dados
AM1	<i>A Mobile Educational Game Accessible to All, Including Screen Reading Users on a Touch-Screen Device</i>	(LEPORINI; PALMUCCI, 2017)	ACM Digital Library
AM2	<i>A model-driven approach to cross-platform development of accessible business apps</i>	(RIEGER et al., 2020)	ACM Digital Library
AM3	<i>An Accessible Roller Coaster Simulator for Touchscreen Devices: An Educational Game for the Visually Impaired</i>	(BIASE et al., 2018)	IEEE Xplore
AM4	<i>Application for the Configuration and Adaptation of the Android Operating System for the Visually Impaired</i>	(OLIVEIRA; BRAGA; DAMACENO, 2018)	ACM Digital Library
AM5	<i>Blind and visually impaired user interface to solve accessibility problems</i>	(SHERA et al., 2021)	Scopus
AM6	<i>Design and development of a mobile app of drug information for people with visual impairment</i>	(MADRIGAL-CADAVID et al., 2020)	ScienceDirect
AM7	<i>Designing multimodal mobile interaction for a text messaging application for visually impaired users</i>	(DUARTE et al., 2017)	Scopus
AM8	<i>Do You like My Outfit? Cromnia, a Mobile Assistant for Blind Users</i>	(VITIELLO et al., 2018)	ACM Digital Library
AM9	<i>Improved and Accessible E-Book Reader Application for Visually Impaired People</i>	(SHIN et al., 2017)	ACM Digital Library
AM10	<i>MathMelodies 2: A Mobile Assistive Application for People with Visual Impairments Developed with React Native</i>	(CANTÙ et al., 2018)	ACM Digital Library
AM11	<i>Object Recognition and Hearing Assistive Technology Mobile Application Using Convolutional Neural Network</i>	(CABALLERO; CATLI; BABIERRA, 2020)	ACM Digital Library
AM12	<i>QUIMIVOX MOBILE 2.0: Application for Helping Visually Impaired People in Learning Periodic Table and Electron Configuration</i>	(OLIVEIRA et al., 2019)	ACM Digital Library
AM13	<i>“Talkin’ about the weather”: Incorporating Talk-Back functionality and sonifications for accessible app design</i>	(TOMLINSON et al., 2016)	Scopus
AM14	<i>Users’ perception on usability aspects of a braille learning mobile application ‘mBRAILLE’</i>	(NAHAR; SULAIMAN; JA-AFAR, 2019)	Scopus
AM15	<i>WordMelodies: Supporting Children with Visual Impairment in Learning Literacy</i>	(MASCETTI et al., 2019)	ACM Digital Library

Fonte: Autor.

3.2 Resultados Encontrados

Nesta seção são apresentados os resumos com as principais características relacionadas ao tema deste trabalho dos artigos selecionados na fase de extração, visando encontrar respostas para as questões levantadas na definição do protocolo de MSL.

3.2.1 *A Mobile Educational Game Accessible to All, Including Screen Reading Users on a Touch-Screen Device*

O estudo realizado por Leporini e Palmucci (2017) teve como objetivo levantar informações e possíveis soluções para as dificuldades levantadas por um grupo composto por 6 pessoas cegas ao responder questões de tarefas interativas. E investigou, por meio de tarefas interativas como exercícios e questionários, a acessibilidade e usabilidade de gestos e leitores de tela em dispositivos móveis com *touch-screen*.

No artigo é apresentado um *game* que envolveu duas pessoas cegas com experiência na utilização de *smartphones* na fase inicial do planejamento do protótipo. O jogo funciona como se fosse um “sistema solar” com oito planetas, com cada planeta representando um conjunto de questões e exercícios. O jogador recebe determinada pontuação cada vez que joga de acordo com os acertos e erros. As principais funcionalidades do *app* relativas à acessibilidade identificadas foram:

1. Contraste de cor para garantir diferentes níveis de acessibilidade;
2. Apresentações de conteúdos de forma auditiva e visual;
3. Interação via gestos ou toques;
4. Suporte auditivo com descrições dos elementos.

Por meio da avaliação desse protótipo, por cegos, o estudo investigou o suporte de acessibilidade *mobile* multiplataforma do conjunto de especificações técnicas WAI-Aria⁸, observando problemas na detecção de elementos devido às suas posições na tela e conteúdos difíceis de identificar na interação com leitores de tela. Notando também que houve alguma dificuldade por conta de gestos implementados no *app* diferirem dos habituais utilizados pelos usuários no *VoiceOver* do iOS.

Apesar dos problemas encontrados, o artigo aponta que o *feedback* foi positivo e os resultados mostraram que os exercícios puderam ser realizados facilmente, por pessoas cegas, fazendo uso de simples gestos com auxílio dos leitores de tela.

⁸ A Iniciativa de Acessibilidade na Web (WAI, do inglês *Web Accessibility Initiative*) define uma maneira de tornar aplicações e conteúdos web mais acessíveis às pessoas com deficiência por meio do conjunto Aplicações para a Internet Ricas em Acessibilidade (ARIA, do inglês *Accessible Rich Internet Applications Suite*).

Tecnologia utilizada para desenvolvimento: *Cordova Framework.*

Plataforma alvo do app desenvolvido: multiplataforma (Android e iOS).

Público alvo da aplicação: PDV.

3.2.2 A Model-Driven Approach to Cross-Platform Development of Accessible Business Apps

Um procedimento comum no processo de desenvolvimento de *software* é considerar a acessibilidade para PDV apenas na etapa final. Além disso, muitos desenvolvedores não estão cientes de técnicas de software para atender esse grupo, pois o domínio de aplicativos móveis multiplataforma tem recebido uma atenção limitada por pesquisadores. Foi nesse sentido, que o estudo de Rieger et al. (2020) buscou identificar desafios, requisitos e soluções técnicas de acessibilidade, selecionando 28 requisitos a respeito de acessibilidade para aplicações móveis por meio de uma RSL.

O artigo apresenta uma abordagem orientada a modelos que integra conceitos de acessibilidade no desenvolvimento de aplicações móveis multiplataforma em conjunto com protótipos acessíveis à PDV. Com isso, apresenta também uma aplicação com foco em fornecer informações sobre chuvas fortes e inundações, na qual os usuários podem ter uma visão de eventos de inundações próximos e compartilhar novos incidentes.

O estudo comparou uma versão da aplicação desenvolvida nativamente que necessitou de 3,400 linhas de código *Java* e 3,200 linhas de código *XML* (gerado de forma semiautomática) com outra versão, com um conjunto similar de funcionalidades. A nova versão do *app* consistiu em 445 linhas de código *MD²*, *framework* baseado na abordagem orientada a modelos para desenvolvimento móvel multiplataforma com a linguagem de alto nível *Xtend*⁹. Principais funcionalidades sobre acessibilidade identificadas:

1. Adaptação da *interface* de acordo com as necessidades do usuário;
2. Integração com os leitores de tela graças ao fornecimento de descrições em texto para elementos não textuais;
3. Personalização do contorno de foco do *TalkBack*.

Segundo o artigo, o estudo de caso mostrou que *apps* acessíveis podem ser gerados a partir do modelo de alto nível *MD²*, implementando as técnicas de integração adequadas em cada ponto. Embora o autor afirme isso, o estudo também deixa claro que ainda havia uma pendência de validação centrada no usuário, visto que o trabalho não implementou todas as técnicas e a solução proposta não foi testada com PDV.

⁹ Linguagem de programação de alto nível para a JVM (*Java Virtual Machine*) com foco em injeção de dependências e geração de código Java. <<https://www.eclipse.org/xtend/>>

Tecnologia utilizada para desenvolvimento: *Xtend, Java e Eclipse.*

Plataforma alvo do app desenvolvido: multiplataforma (Android e iOS).

Público alvo da aplicação: PDV interessadas em saber sobre eventos climáticos locais como chuvas fortes e inundações.

3.2.3 An Accessible Roller Coaster Simulator for Touchscreen Devices: An Educational Game for the Visually Impaired

O trabalho de [Biase et al. \(2018\)](#) apresenta um *app* simulador de montanha russa, baseado em simuladores educacionais já existentes e adaptado para *smartphones*, para ser utilizado em disciplinas de Educação Física por pessoas com e sem DV. A aplicação foi desenvolvida para auxiliar no estudo de Energia Mecânica e trás as interações por áudio e tátil como alternativas à visual. As principais funcionalidades sobre acessibilidade identificadas no *app* foram:

1. Os elementos visuais possuem descrições textuais para integração com leitores de tela;
2. *Feedback* por meio de “texto para voz” (TTS, do inglês *text-to-speech*) e vibração ao clicar em determinados elementos na tela, mesmo com o modo de acessibilidade desativado;
3. Efeitos sonoros característicos que ilustram os resultados da simulação ao longo do percurso.

Com taxas de 73% eficácia, 77% de eficiência e 66% satisfação do usuário com relação a aplicação desenvolvida, os testes de usabilidade demonstraram que as estratégias de interação propostas são viáveis, com grande potencial para serem utilizadas em propósitos educacionais.

Contudo, alguns problemas de acessibilidade afetaram a taxa de satisfação dos usuários, a mantendo em 66%, tais como dificuldades em seguir a trilha da montanha com apenas um dedo, não ser possível detectar quando o carro está voltando no trilho e falha no comando que altera o foco dos elementos, alterando para o elemento errado.

Tecnologia utilizada para desenvolvimento: *Unity 3D engine.*

Plataforma alvo do app desenvolvido: Android.

Público alvo da aplicação: Pessoas com e sem DV.

3.2.4 Application for the Configuration and Adaptation of the Android Operating System for the Visually Impaired

Apesar das vantagens dos dispositivos móveis, alguns desafios da interação de PDV com os sistemas operacionais (SOs) desses dispositivos precisam ser superados, para que a tecnologia alcance um número significativo nesse grupo. Assim, o estudo de [Oliveira, Braga e Damaceno](#)

(2018) visou planejar e desenvolver uma aplicação que automatize as configurações do SO Android de acordo com as preferências de acessibilidade de cada PDV, por meio de comandos de voz. O artigo apresenta algumas funcionalidades e técnicas relacionadas a acessibilidade que são listadas a seguir:

1. Escala de Usabilidade do Sistema (SUS, do inglês *System Usability Scale*) para avaliação de usabilidade da aplicação;
2. *SpeechRecognizer* do Android para reconhecimento de voz;
3. Eurísticas de Usabilidade de Nielsen (do inglês, *Nielsen Usability Heuristics*) para evitar problemas de acessibilidade já mapeados.

Um protótipo foi desenvolvido e mostrou potencial para ser utilizado como ferramenta para PDV, trazendo benefícios com a possibilidade do uso de comando de voz. Os testes foram realizados com seis voluntárias com DV, sendo duas parcial e quatro total. Das quais três já possuíam experiência com comandos de voz e apenas duas das seis pessoas já haviam realizado a configuração do dispositivo alguma vez.

Por fim, as voluntárias expressaram avaliações positivas quanto à autonomia, satisfação e usabilidade da aplicação. E o tempo gasto para realizar as configurações de acessibilidade foi mais curto no *app* desenvolvido que na aplicação padrão do Android.

Tecnologia utilizada para desenvolvimento: *Android Studio 2.0*.

Plataforma alvo do *app* desenvolvido: Android.

Público alvo da aplicação: PDV.

3.2.5 *Blind and visually impaired user interface to solve accessibility problems*

Este estudo realizou uma RSL e testes em várias aplicações móveis para PDV, e dividiu os problemas encontrados em três categorias: organização, apresentação e comportamento (OAC). Uma aplicação móvel, chamada “*Read Master*”, também foi desenvolvida no trabalho de [Shera et al. \(2021\)](#), incorporando soluções para os principais problemas de OAC.

Tabela 1 – Categorias dos problemas identificados.

Código	Categoria
CRR1	Apresentação
CRR2	Comportamento
CRR3	Organizacional

Fonte: [Shera et al. \(2021\)](#).

Na tabela [Tabela 2](#) estão listados os problemas identificados pelo estudo. Os códigos na coluna “Categoria” referem-se às categorias da [Tabela 1](#).

Tabela 2 – Problemas de acessibilidade encontrados por categoria.

Código	Problema	Categoria
AM1P1	Falta de consistência no <i>layout</i> e terminologias	CRR1
AM1P2	Leitor de tela fornecendo <i>feedbacks</i> confusos	CRR1
AM1P3	Leitor de tela quebrando	CRR2
AM1P4	Ouvir cabeçalhos e títulos das páginas de forma redundante antes de detectar o conteúdo das telas	CRR2
AM1P5	Conteúdos na tela da aplicação móvel	CRR3
AM1P6	Fluxo de tarefas	CRR3
AM1P7	Não entendimento da sequência natural de leitura	CRR3
AM1P8	Não entendimento do fluxo natural de tarefas	CRR3
AM1P9	Problemas de navegação	CRR3
AM1P10	Sobrecarga de informações	CRR3

Fonte: [Shera et al. \(2021\)](#).

O *app* desenvolvido consistiu em duas funcionalidades principais: fornecer informações científicas e *quizzes* de múltipla escolha. As principais técnicas e funcionalidades identificadas no estudo para o suporte de acessibilidade foram:

1. *SUS* para avaliação de usabilidade da aplicação;
2. Leitor de tela embutido por meio de TTS.
3. Levantamento e categorização dos principais problemas de acessibilidade em *apps* móveis.

Uma avaliação de usabilidade do *app*, com 56 PDV, foi conduzida e validada com foco na experiência de usuários com DV. Os resultados mostraram que a organização da aplicação estava 100% efetiva tanto para os usuários cegos quanto para os com DV parcial. Já com relação à eficiência dos usuários, a dos com DV parcial mostrou-se maior que a dos cegos.

O nível mais alto de satisfação quanto às 3 categorias de problemas avaliados estava na apresentação com 87,62% para usuários com DV total, enquanto para os com visão parcial estava tanto na organização quanto na apresentação com 89,21%. No geral, o estudo indica que a aplicação reduziu a gravidade dos problemas de OPB, oferecendo alta usabilidade.

Tecnologia utilizada para desenvolvimento: Não informado.

Plataforma alvo do *app* desenvolvido: Android.

Público alvo da aplicação: PDV.

3.2.6 *Design and development of a mobile app of drug information for people with visual impairment*

Esse trabalho foi desenvolvido na Colômbia, onde a falta de acesso à informações acessíveis de rótulos de medicamentos como contraindicações, armazenamento, data de validade

e dosagem foi identificada como uma das principais barreiras no uso de medicamentos por PDV ([MADRIGAL-CADAVID et al., 2020](#)).

Nesse contexto, uma aplicação *mobile* chamada *FarmaceuticApp* foi desenvolvida no estudo. A principal funcionalidade do *app* é a de busca por informações de medicamentos, apresentando-as ao usuário de forma acessível e podendo ser realizada por vários meios.

As principais técnicas e funcionalidades identificadas, relacionadas à acessibilidade e utilizadas no desenvolvimento dessa solução, foram:

1. Tamanho da fonte das letras personalizável;
2. Vibração e sons para alertar o usuário do resultado da busca;
3. *Tutorial* com possibilidade de ser visto novamente;
4. Possibilidade de busca por *barcode* e *qrcode*, foto, comando de voz e texto;
5. Possibilidade de ativar e desativar o assistente de voz do *app*.

Tecnologia utilizada no desenvolvimento: *Java, Android Studio, Accessibility Scanner App, e o Test Lab do Firebase.*

Plataforma alvo do *app* desenvolvido: *Android.*

Público alvo da aplicação: PDV que buscam obter informações de rótulos de medicamentos.

O estudo envolveu 48 PDV, das quais 69% necessitavam de assistência para o uso de medicamentos e 90% possuíam celulares, sendo 93% deles com o SO *Android*. Na avaliação final, 100% dos usuários disseram utilizariam o *app* e o avaliaram entre 4 e 5 estrelas (bom e muito bom).

3.2.7 *Designing multimodal mobile interaction for a text messaging application for visually impaired users*

Apesar da inclusão de opções de acessibilidade, os SOs móveis ainda enfrentam uma falta de suporte adequado para alguns tipos de atividades e contextos, como é o exemplo da escrita de textos para PDV, uma tarefa que acaba consumindo muito tempo. Além disso, os usuários geralmente necessitam utilizar as duas mãos para escrever mensagens, o que mostra ser um problema para cegos que necessitam carregar bengala ou possuem cão guia, assim restando apenas uma mão livre.

Nesse contexto, a abordagem proposta no estudo de [Duarte et al. \(2017\)](#), por meio do protótipo de um *app* para envio de mensagens, visou uma interação com o *smartphone* com as

mãos livres, por meio de técnicas multimodais, especialmente o uso de gestos em combinação com comandos de voz.

Os gestos são utilizados como gatilhos para ações, logo quando um gesto é reconhecido, alguma função como o “reconhecedor de fala” ou o TTS é acionada. Por exemplo, existe um gesto para a ação de adicionar uma nova mensagem, ao reconhecer-lo, o *app* ativa o reconhecedor de fala para que o usuário dite o que deve ser escrito na mensagem.

Um outro gesto aciona a função para revisão da mensagem escrita por meio do TTS, em que a mensagem é lida palavra a palavra. As principais características relacionadas à acessibilidade identificadas nessa solução foram:

1. Reconhecimento de voz;
2. Reconhecimento de gestos;
3. Síntese de fala.
4. Possibilidade de revisar as mensagens escritas de maneira acessível;
5. Possibilidade de parar a narração durante a revisão da mensagem e editar palavras específicas;
6. Aplicação de questionário da escala SUS.

Tecnologia utilizada no desenvolvimento: Java, Android Studio, Accessibility Scanner App, e o Test Lab do Firebase.

Plataforma alvo do *app* desenvolvido: Android.

Público alvo da aplicação: PDV.

Uma pesquisa foi realizada com 9 usuários com DV e resultou em *feedbacks* positivos, principalmente a respeito da interação por gestos. O estudo também trouxe comparativo de performance dos usuários na realização de tarefas no *app* de envio de mensagem padrão com o *app* desenvolvido. Na avaliação da usabilidade das aplicações, fazendo uso da escala SUS, ambas atingiram 74 pontos, considerada uma alta pontuação.

Os resultados mostraram que na realização de tarefas fáceis a performance do *app* era pouco superior a alternativa padrão do sistema. Porém, passa-se a notar grandes diferenças a favor do *app* desenvolvido em tarefas consideradas normais e difíceis, com cerca de 30% e 50% mais performance, respectivamente, para a solução desenvolvida em relação ao *app* padrão.

3.2.8 *Do You like My Outfit? Cromnia, a Mobile Assistant for Blind Users*

O objetivo do estudo de Vitiello et al. (2018) foi projetar uma solução assistiva que pudesse prover autonomia à pessoas cegas em suas atividades diárias. Especialistas na área de

deficiência visual, de clínicos à profissionais de reabilitação vocacional e operadores do campo de cuidados sociais, participaram do estudo.

O processo de análise e projeto envolveu, desde o início, a participação de 4 pessoas cegas da *Italian Blind Union*, que se voluntariaram para colaborar com a equipe de *design* de usabilidade. Entre as tarefas diárias que mais se esperava autonomia a de vestir-se com uma combinação de cores e roupas adequadas mostrou-se ser o maior interesse para as PDV, essas que geralmente dependem de ajudantes para isso.

O estudo levantou que já existiam soluções no mercado para esse problema, porém a ideia de uma ferramenta paga não foi bem aceita pelos entrevistados, observando que muitos não poderiam pagar. Diante disso, uma aplicação *mobile* foi projetada visando a autonomia de PDV total ou parcial nesse ato cotidiano de vestir-se. O *app* é bem simples e consiste em uma única *interface*, parecida com a padrão da câmera do sistema iOS.

As principais soluções de acessibilidade utilizadas no desenvolvimento foram:

1. Integração com leitores de tela;
2. Tamanho de fontes e *labels* adaptáveis de acordo com o tipo de deficiência;
3. Sistema de notificações simples e imediato;
4. Resposta em tempo real.

Tecnologia utilizada no desenvolvimento: Não informado.

Plataforma alvo do *app* desenvolvido: iOS.

Público alvo da aplicação: PDV.

Como resultado do estudo uma aplicação chamada *Cromnia* foi desenvolvida, esta que possibilita que os usuários reconheçam cores, padrões e combinações de cores, considerando a iluminação do ambiente.

Os testes envolveram 6 PDV com parcial e 6 com DV total. Os participantes gostaram dos benefícios do *app* e mostraram-se ansiosos para experimentar novas versões, pensando em quando poderão utilizar o aplicativo de fato no dia-a-dia. O *app* está disponível na *AppStore* e conta com alto número de *downloads*.

3.2.9 Improved and Accessible E-Book Reader Application for Visually Impaired People

Embora livros digitais já estejam estabelecidos internacionalmente, não são satisfatórios em termos de acessibilidade e *interface*. Por conta disso, o estudo de [Shin et al. \(2017\)](#) apresenta um aplicativo leitor de *e-book* acessível à PDV, que tem o objetivo de suprimir limitações como

falta de novos livros, ausência de textos alternativos e navegação desconfortável dos atuais formatos acessíveis (áudio e *Braille*).

Um levantamento de requisitos de usuário foi realizado por meio de questionário e cerca 70% dos requisitos foram implementados. O *app* possibilita a realização de busca, *download* e leitura de conteúdos no formato *EPUB3* e possui controles para iniciar, parar, avançar e retroceder a leitura. Quanto à acessibilidade, foram identificadas as seguintes soluções:

1. Suporte para comandos de voz;
2. Configurações de alto contraste;
3. Síntese de voz para leitura dos *e-books*;
4. Tamanho dos botões e espaçamentos adequados à PDV.

Tecnologia utilizada no desenvolvimento: Não informado.

Plataforma alvo do *app* desenvolvido: iOS.

Público alvo da aplicação: PDV que gostam de livros.

Nos resultados dos testes, realizados com 12 PDV (7 experientes e 5 sem experiência), o estudo mostrou que a média de satisfação dos usuários foi de aproximadamente 75% nos testes de usabilidade, realizados em 3 fases, com usuários com e sem experiência. E o tempo médio de execução das tarefas foi de 92 segundos para usuários não experientes e 82 segundos para experientes.

Entretanto, usuários experientes acabaram enfrentando erros relacionados a *login*, configuração e busca por tentarem utilizar suas próprias abordagens baseadas nas interações com outras aplicações.

3.2.10 *MathMelodies 2: A Mobile Assistive Application for People with Visual Impairments Developed with React Native*

Esse artigo apresenta a experiência do desenvolvimento do *MathMelodies 2*, uma aplicação para ajudar crianças de 1 a 5 anos com DV no estudo de matemática. A aplicação apresenta 13 tipos de exercícios e diferentes níveis de dificuldade. Esses exercícios passam-se dentro de contos de fantasia, os quais a criança tem que resolver para avançar na história.

A primeira versão foi desenvolvida em 2013, financiada por uma campanha de *crowdfunding* e lançada para *iPad* de forma gratuita. O *design* do novo *app* seguiu princípios derivados da experiência e do *feedback* dos usuários da versão anterior, dos quais uma das demandas mais frequentes foi a de disponibilização do *app* para outras plataformas, Android e iOS.

Assim, nesse trabalho, [Cantù et al. \(2018\)](#), desenvolve essa nova versão como um protótipo, utilizando *React Native* para reduzir o esforço de desenvolvimento. As principais técnicas e funcionalidades para acessibilidade, utilizadas nesse estudo, são listadas a seguir:

1. Implementação nativa para iOS e Android de componentes não acessíveis no *React Native*;
2. Elementos chave de interação sempre posicionados na mesma parte da tela, em locais de fácil acesso;
3. Tamanho dos ícones e componentes adaptáveis de acordo com tamanho da tela;
4. Todos os elementos visíveis na tela sem necessidade de rolagem;
5. Cores de fundo uniformes e neutras;
6. Interações por gestos simples.

Tecnologia utilizada para desenvolvimento: *React Native*.

Plataforma alvo do app desenvolvido: multiplataforma (Android e iOS).

Público alvo da aplicação: Crianças com DV.

Embora as funcionalidades básicas tenham sido contempladas pelo *framework* utilizado, uma funcionalidade avançada que foi requerida não era suportada. Por conta disso, foi necessário desenvolver componentes adicionais nativamente, isto é, utilizando as tecnologias específicas para cada plataforma.

Por fim, o artigo conclui que *React Native* é uma escolha válida para o desenvolvimento de aplicações acessíveis, com base nos resultados de testes preliminares, com participação de duas pessoas (uma com DV parcial e outra total), sugerindo que a aplicação estava totalmente acessível.

3.2.11 *Object Recognition and Hearing Assistive Technology Mobile Application Using Convolutional Neural Network*

A falta de aplicações móveis que atendam pelo menos as necessidades mais comuns de PDV motivou a realização do trabalho de [Caballero, Catli e Babierra \(2020\)](#), que desenvolveu uma aplicação com objetivo de atender as necessidades desse grupo utilizando tecnologias de Reconhecimento de Objetos (RO) e TTS.

O app utiliza algoritmos de *Convolutional Neural Network* (CNN), solução de aprendizado de máquina reconhecida como um poderoso método para reconhecimento de imagens, para identificar detalhes em imagens e narrá-los para o usuário por meio do TTS.

O artigo concentra-se mais na apresentação da API utilizada para o RO, mostrando pouco sobre a aplicação *mobile*, ainda assim, foram identificadas as seguintes características de acessibilidade no *app*:

1. Reconhecimento de detalhes de imagens;
2. Síntese dos resultados do RO por voz.

Tecnologia utilizada para desenvolvimento: Não informado.

Plataforma alvo do *app* desenvolvido: Android.

Público alvo da aplicação: PDV.

O estudo realizou a revisão de diferentes estudos e tecnologias que utilizam CNN, um dos principais estudos citados foi publicado em 2015 na Conferência Brasileira de Sistemas Inteligentes (BRACIS), este que utiliza RO para um sistema de navegação inteligente que possibilita que robôs interajam e determinem o comportamento de objetos. Fazendo uso de informações identificadas nos trabalhos relacionados, o trabalho aplicou o RO direcionado à inclusão social de PDV.

Os resultados mostraram que CNN tem potencial para classificar coisas vivas e objetos em ambientes interiores e exteriores com alta precisão, fazendo uso de imagens públicas como base de treinamento, possibilitando um desempenho funcional e confiável do sistema em benefício das PDV graças ao *app* desenvolvido.

3.2.12 *QUIMIVOX MOBILE 2.0: Application for Helping Visually Impaired People in Learning Periodic Table and Electron Configuration*

Muito ainda precisa ser feito quanto à inclusão de PDV no processo de ensino e aprendizagem de química, por requerer de muitos recursos visuais. E, embora exista uma quantidade significativa de *apps* que auxiliam no ensino de química, os mesmos não são acessíveis aos DV, mesmo com o uso de leitores de tela.

Foi nesse sentido que o estudo de Oliveira et al. (2019) introduziu uma nova versão do “*Quimivox Mobile 2.0*”, aplicativo que apresenta informações acessíveis à PDV sobre a tabela periódica e, na nova versão, a configuração eletrônica dos elementos químicos. A interação do *app* é baseada em gestos e comandos de voz, com as informações sendo apresentadas graficamente e por síntese de voz, graças ao *TalkBack*.

A aplicação utiliza de técnicas de gestos já utilizadas em outras ferramentas que consistem em deslizar com os dedos em quatro direções. Esses gestos foram complementados com outros específicos para a realização de ações na aplicação, tais como a ativação do reconhecimento de voz e uma opção para retornar à tela anterior. Segue abaixo as principais técnicas e funcionalidades para acessibilidade identificadas no estudo:

1. Interação por reconhecimento de voz e gestos;
2. Tamanhos de fontes de letras ampliados;
3. Alto contraste (fundos pretos e textos brancos);
4. Possibilidade de escolha de cores do *app* para melhorar a legibilidade para pessoas daltônicas;
5. *Feedback* sonoro mesmo com *Talkback* desativado.

Tecnologia utilizada para desenvolvimento: Java, *Android Studio* e *API Airy*.

Plataforma alvo do *app* desenvolvido: Android 4.0 ou superior.

Público alvo da aplicação: PDV interessadas no aprendizado de Química.

Os usuários apontaram o comando de voz como a funcionalidade que mais facilitou na utilização da *app*. Na avaliação de uma das PDV, participante dos testes, o desenvolvimento de manual poderia contribuir com melhor entendimento do funcionamento do aplicativo. Outras sugestões foram a ampliação dos tipos de toques na tela e o aumento na velocidade da voz sintetizada.

O artigo conclui que os participantes aprovaram a nova versão, avaliando positivamente o *app*, indicando que a maior dificuldade estava na pouca prática no uso de dispositivos móveis por parte de alguns DV. E relata que essa dificuldade estava relacionada aos gestos, com a maioria fazendo algum comentário negativo, citando 5 desses participantes.

Porém, o autor supõe que, com a prática no uso dos gestos, essa dificuldade poderia ser diminuída significativamente, citando o reconhecimento da falta de experiência na utilização de dispositivos móveis por 4 participantes como justificativa, sendo que apenas um deles, chamado P10, fazia parte dos 5 participantes citados pelos comentários negativos.

3.2.13 “*Talkin’ about the weather*”: Incorporating TalkBack functionality and sonifications for accessible app design

Informações a respeito do clima atual e previsões são especialmente importantes para PDV, visto que podem afetar suas decisões do cotidiano, como escolhas de rotas, roupas e tecnologias assistivas que impactam significativamente seu trajeto. Mesmo assim, essas pessoas enfrentam péssimas experiências tentando buscar informações sobre o clima nos dispositivos móveis.

Esses problemas costumam ocorrer devido a erros entre as informações na tela e a ordem em que os leitores de tela as apresentam, além dos *apps* serem cheios de imagens e ícones que não apresentam descrições para o usuário, a menos que possa enxergá-las.

Diante disso, em [Tomlinson et al. \(2016\)](#), foi projetado um *app* de clima para ser acessível a usuários que dependem de leitores de tela. No qual o estudo realizou uma análise das necessidades dos usuários com DV, levantando quais eram as informações importantes e em qual ordem eles gostariam de consumi-las.

As principais soluções quanto à acessibilidade identificadas foram:

1. Alternativa aos ícones padrões utilizados com os chamados “Ícones audíveis”;
2. Utilização constante do *TalkBack* durante o processo de desenvolvimento;
3. Interface com alto contraste (textos brancos em fundo preto), visando a experiência de usuário (UX) de PDV;
4. Integração com *Talkback* seguindo as Diretrizes de Acessibilidade do Google.

“Ícones audíveis” emitem sons breves, baseados nos sons reais do cotidiano, e servem alternativa para representação dos ícones visuais de clima, como o ícone de chuva, representado por sons que remetem ao evento.

Tecnologia utilizada para desenvolvimento: Não informado.

Plataforma alvo do *app* desenvolvido: Android.

Público alvo da aplicação: PDV que necessitam saber sobre o clima.

Nos testes de usabilidade, 7 participantes responderam que utilizaram o *app* por pelo menos seis dias durante a semana e, no geral, reportaram terem obtido experiência tão boa ou melhor que nos *apps* de clima que já utilizaram anteriormente.

3.2.14 *Users' perception on usability aspects of a braille learning mobile application 'mBRAILLE'*

Estudantes com DV enfrentam dificuldades ou incapacidade, a depender do nível de DV, para obter informações visuais, o que torna o processo de aprendizagem deles mais difícil que o dos outros. Nesse artigo, [Nahar, Sulaiman e Jaafar \(2019\)](#), apresenta o *mBRAILLE*, *app* que foi desenvolvido em *Bangladesh* para auxiliar PDV no processo de autoaprendizagem de *Braille*, sem ou com dependência mínima de outras pessoas.

Embora a publicação não apresente muitos detalhes do processo de desenvolvimento, sequer mencionam leitores de tela, algumas características relacionadas à acessibilidade utilizadas na solução foram identificadas, seguem:

1. *Tutorial* para auxiliar o usuário na utilização do *app*;
2. *Feedback* por vibração e áudio;

Tecnologia utilizada para desenvolvimento: Não informado.

Plataforma alvo do app desenvolvido: Android.

Público alvo da aplicação: Estudantes de *Bangladesh* com DV.

O trabalho avaliou 4 aspectos de usabilidade (aprendizagem, interface e funcionalidades, acessibilidade e auto descriptividade) do *app* por meio de testes com 5 usuários com DV, que realizaram a avaliação após utilizarem a aplicação por 2 semanas, mostrando resultados de avaliação média satisfatórios, de 6 ou acima, numa escala de 0 a 7.

O estudo teve uma limitação de apenas 5 participantes, sendo todos experientes em *Braille*. Contudo, o artigo menciona que trabalhos futuros irão concentrar-se na avaliação e testes da efetividade do aprendizado de *Braille* por meio do *app* com um grande número de participantes de diferentes escolas.

3.2.15 WordMelodies: Supporting Children with Visual Impairment in Learning Literacy

As ferramentas educacionais de escolas primárias frequentemente não são acessíveis para crianças com DV. Além disso, os livros costumam ser ricos em conteúdos gráficos com o intuito de engajar os alunos, impactando na acessibilidade mesmo quando estão disponíveis no formato digital. Da mesma forma, *apps* educacionais constantemente apresentam conteúdos gráficos interativos de maneira inacessível à PDV.

Visando amenizar esses problemas, o artigo de [Mascetti et al. \(2019\)](#) apresenta o *WordMelodies*, uma aplicação *mobile* inclusiva e multiplataforma que teve o objetivo de ajudar crianças com DV na aquisição de habilidades básicas de literatura com 8 tipos de exercícios.

A aplicação foi projetada e avaliada por 3 especialistas no domínio de tecnologias assistivas e educação para crianças com DV. As principais características relativas à acessibilidade encontradas no artigo foram:

1. Elementos chave de interação sempre posicionados na mesma parte da tela, priorizando os cantos da tela;
2. Interações por gestos como “arrastar e soltar” com descrição auditiva;
3. Descrição alternativa em texto dos elementos de tela para integração com leitores de tela.

Tecnologia utilizada para desenvolvimento: *React Native*.

Plataforma alvo do app desenvolvido: multiplataforma (Android e iOS).

Público alvo da aplicação: Crianças com DV.

Exceto por um problema que afetou a utilização do usuário ao navegar entre os elementos utilizando leitores de tela, o *app* mostrou-se totalmente acessível na avaliação dos especialistas. Nessa navegação, a ordem dos elementos não corresponde com a ordem lógica apresentada na tela, problema que ocorreu por uma limitação do *kit* de ferramentas da plataforma de desenvolvimento utilizada, o *React Native*.

Um dos principais desafios no desenvolvimento foi alcançar uma funcionalidade de “arrastar e soltar” acessível e fácil de utilizar, pois no *React Native* esse componente não fornece suporte à acessibilidade, sendo necessário o desenvolvimento de um componente nativo tanto no iOS como no Android, para prover informações auditivas ao usuário durante o uso do componente.

3.3 Estudos Relacionados

Durante o processo de seleção de artigos do MSL, foram encontrados alguns estudos secundários, tipo de estudo que realiza revisão de estudos primários relacionados a um tema específico (KITCHENHAM; CHARTERS, 2007). Embora tenham sido rejeitados no MSL, por se enquadarem em algum dos critérios definidos na seção anterior, os estudos que realizaram essas revisões dentro do tema abordado neste trabalho foram considerados como estudos relacionados.

Assim, esta seção apresenta os principais problemas e propostas de soluções relacionados à acessibilidade de aplicações para dispositivos móveis identificados por esses estudos. No [Quadro 5](#) estão listadas as informações de cada um desses estudos secundários.

Quadro 5 – Estudos relacionados identificados no processo de MSL.

Código	Título	Referência	Base de dados
AR1	<i>Accessibility of Mobile Applications: Evaluation by Users with Visual Impairment and by Automated Tools</i>	(MATEUS et al., 2020)	ACM Digital Library
AR2	<i>Can Everyone use my app? An Empirical Study on Accessibility in Android Apps</i>	(VENDOME et al., 2019)	Scopus
AR3	<i>Effect of UX Design Guideline on the information accessibility for the visually impaired in the mobile health apps</i>	(KIM et al., 2019)	Scopus
AR4	<i>Mobile Device Accessibility for the Visually Impaired: Problems Mapping and Empirical Study of Touch Screen Gestures</i>	(DAMACENO; BRAGA; CHALCO, 2016)	ACM Digital Library
AR5	<i>Observation Based Analysis on the Use of Mobile Applications for Visually Impaired Users</i>	(SIEBRA et al., 2016)	ACM Digital Library
AR6	<i>Prioritization of mobile accessibility guidelines for visual impaired users</i>	(QUISPE; SCATALON; ELER, 2020)	Scopus

Fonte: Autor.

3.3.1 *Accessibility of Mobile Applications: Evaluation by Users with Visual Impairment and by Automated Tools*

O artigo apresenta um estudo comparativo de problemas de acessibilidade encontrados pelas ferramentas automatizadas MATE e *Accessibility Scanner*, com os problemas encontrados em um estudo anterior envolvendo 11 usuários com DV. Além disso, o trabalho sumarizou e categorizou os problemas mais encontrados pelos usuários. As principais categorias são listadas na [Tabela 3](#).

Na [Tabela 4](#) são listados os principais tipos de problemas, que apresentaram um total de pelo menos 10 observações. As categorias, de acordo com a [Tabela 3](#), e o número total de

Tabela 3 – Categorias dos tipos de problemas mais identificados.

Código	Categoria
CPF1	Botões
CPF2	Características do Sistema
CPF3	Conteúdo e Significado
CPF4	Controles, formulários e funcionalidades
CPF5	Imagen

Fonte: [Rieger et al. \(2020\)](#).

observações para cada tipo de DV (total ou parcial) também são relacionados à cada tipo de problema. Como o artigo só menciona os tipos problemas encontrados com maior frequência por cada tipo de usuário, o número de observações de alguns não estão presentes na [Tabela 4](#).

Tabela 4 – Problemas mais frequentes encontrados pelos usuários por tipo de DV.

Código	Problema	Categoria	DVT	DVP	Total
AR1P1	Feedback inapropriado.	CPF4	34	15	49
AR1P2	Falta de informações.	CPF1	22	8	30
AR1P3	Usuários presumiram que era uma funcionalidade.	CPF4	18	9	27
AR1P4	Funcionalidades confusas ou não claras.	CPF4	25	-	25
AR1P5	Apresentação padrão de elementos de controle ou formulário não adequada.	CPF4	11	12	23
AR1P6	Sequências de interação confusas ou não claras.	CPF4	15	6	21
AR1P7	Usuários não entenderam sentido do conteúdo.	CPF3	15	5	20
AR1P8	Organização do conteúdo inconsistente.	CPF3	12	6	18
AR1P9	Funcionalidade não funciona como esperado.	CPF4	6	10	16
AR1P10	Funcionalidades dos botões confusas ou não claras.	CPF1	15	-	15
AR1P11	Expectativa de funcionalidade que não existe.	CPF4	10	5	15
AR1P12	Sem alternativa textual.	CPF5	14	-	14
AR1P13	Sistema muito lento.	CPF2	-	11	11
AR1P14	Significado no conteúdo está perdido.	CPF3	6	4	10

Fonte: [Rieger et al. \(2020\)](#).

Os resultados do estudo mostraram que 36 tipos de problemas foram encontrados somente pelos usuários, 11 somente pelas ferramentas e 3 por ambos os métodos. Evidenciando assim a necessidade de utilização de mais de um método para identificação dos problemas de acessibilidade.

Além disso, o estudo mostrou a importância da utilização dessas ferramentas automatizadas, visto que parte significativa dos problemas podem ser identificados ainda no processo de desenvolvimento, reduzindo o esforço e, consequentemente, o custo para solucioná-los.

3.3.2 *Can Everyone use my app? An Empirical Study on Accessibility in Android Apps*

Esse trabalho realizou um estudo piloto no qual foi observado que desenvolvedores de aplicativos móveis raramente utilizam as APIs de Acessibilidade e que o uso de descrições alter-

nativas para elementos de *interface* também é limitado. Ademais, visando entender a perspectiva desses desenvolvedores, o estudo também realizou uma investigação de postagens no *Stack Overflow*, identificando os aspectos de acessibilidade que os desenvolvedores implementavam e os que experienciavam dificuldades.

O estudo investigou aspectos de acessibilidade no geral, baseado em 336 discussões de desenvolvedores Android no *Stack Overflow*, sendo 159 dessas sobre acessibilidade à DV. Dessa 159 discussões, os principais aspectos discutidos foram sobre *feedbacks* sonoros e legibilidade (114 e 24 postagens, respectivamente) como mostra a [Tabela 5](#).

Tabela 5 – Aspectos de acessibilidade à DV discutidos por *devs Android* no *Stack Overflow*.

Código	Aspecto	Categoria
AR2P1	Alertas de acessibilidade	<i>Feedbacks</i> sonoros
AR2P2	Ampliação da tela	Legibilidade
AR2P3	Aspectos não funcionais	<i>Feedbacks</i> sonoros
AR2P4	Consciência de contexto	<i>Feedbacks</i> sonoros
AR2P5	Conteúdos, ações e gestos customizados	<i>Feedbacks</i> sonoros
AR2P6	<i>Frameworks</i> de terceiros	<i>Feedbacks</i> sonoros
AR2P7	<i>Mobile web apps</i>	<i>Feedbacks</i> sonoros
AR2P8	Problemas com serviços	<i>Feedbacks</i> sonoros
AR2P9	Sons e vibrações	<i>Feedbacks</i> sonoros
AR2P10	Suporte à <i>Braille</i>	Teclados alternativos
AR2P11	Tamanho de fonte	Legibilidade
AR2P12	Teclado customizado	Teclados alternativos
AR2P13	Transformações de cores	Transformações de cores

Fonte: [Vendome et al. \(2019\)](#).

O trabalho de [Vendome et al. \(2019\)](#) analisou 13.817 *apps Android* de código aberto, descobrindo que cerca de 50% deles tinham descrições alternativas para todos os elementos, enquanto cerca de 37% não tinha nenhuma. Além disso, o artigo apontou que apenas cerca de 2% desses *apps* utilizavam alguma API de acessibilidade no projeto.

3.3.3 *Effect of UX Design Guideline on the information accessibility for the visually impaired in the mobile health apps*

Acessibilidade de informações visuais para DV raramente é considerada ao projetar aplicações móveis para saúde ([KIM et al., 2019](#)). Diante disso, o artigo propõe um guia de diretrizes de acessibilidade à DV, chamado UXDG (*UX Design Guideline*), para resolver esse problema. A [Tabela 6](#) lista as diretrizes do UXDG de acordo com as categorias.

Como parte da validação do guia, 120 *apps* da área de saúde foram analisados quanto à taxa de conformidade com o guia. Na análise desses *apps*, a média da taxa de conformidade com o guia foi de 39,24%, com a diretriz AR3D7 apresentando a maior taxa, com 71,67%, enquanto a AR3D9 apresentou a menor, com 5%.

Tabela 6 – Diretrizes do UXDG por categoria.

Código	Diretriz	Categoria
AR3D1	Destacar as mídias que disparam ação.	Aquisição de informação
AR3D2	Destacar as principais imagens que o usuário pode acessar.	Aquisição de informação
AR3D3	Navegação intuitiva.	Acessibilidade dos dados
AR3D4	Posicionar a caixa de pesquisa sempre no mesmo local.	Busca de dados
AR3D5	Posicionar resultados de buscas logo após a caixa de texto.	Busca de dados
AR3D6	Reconhecimento de voz para entrada de texto.	Busca de dados
AR3D7	Resposta intuitiva do <i>menu</i> de acordo com intenção do usuário.	Acessibilidade dos dados
AR3D8	Suporte à esquemas de cores alternativos.	Melhora na exposição dos dados
AR3D9	Suporte de <i>zoom in/out</i> para os principais conteúdos.	Melhora na exposição dos dados
AR3D10	Supporte para outros métodos entrada além do toque.	Acessibilidade dos dados
AR3D11	Uso de fontes com alta legibilidade.	Aquisição de informação

Fonte: [Kim et al. \(2019\)](#).

O estudo realizou testes, conduzidos com 23 PDV e 23 sem DV, comparando *apps* selecionados da área da saúde antes e depois da aplicação do UXDG. Os resultados apontaram que houve um aumento na velocidade de reconhecimento das informações depois de aplicar as diretrizes. De acordo com o experimento, esse aumento aconteceu tanto para usuários com DV, aumento de 13,68%, quanto para os sem, de 32,41%.

3.3.4 *Mobile Device Accessibility for the Visually Impaired: Problems Mapping and Empirical Study of Touch Screen Gestures*

Esse artigo, mediante um MSL, apresenta os problemas de acessibilidade enfrentados na utilização de dispositivos móveis por PDV encontrados na literatura. A [Tabela 7](#) mostra, como categorias, 6 dos 7 grupos de problemas identificados no estudo, desconsiderando o de “borda não sensível ao toque”, visto que esse é um problema relativo aos dispositivos físicos.

Tabela 7 – Categorias dos problemas mapeados na literatura.

Código	Categoria
CPM1	Botões
CPM2	Comandos de voz
CPM3	Entrada de dados
CPM4	Interação por gestos
CPM5	Leitor de tela
CPM6	Retorno ao usuário

Fonte: [Damaceno, Braga e Chalco \(2016\)](#).

Na [Tabela 8](#) são listados os problemas relacionados à botões (CPM1), comandos de voz (CPM2) e retorno do usuário (CPM6), e o número de citações, este último que corresponde ao número de estudos nos quais o problema foi identificado, sendo que os problemas relacionados aos botões físicos dos dispositivos foram desconsiderados, por estarem fora do controle da aplicação.

Tabela 8 – Problemas relacionados às categorias CPM1, CPM2 e CPM6.

Código	Problema	Categoria	Citações
AR4P1	A grande proximidade entre os botões virtuais dificulta a interação.	CPM1	1
AR4P2	Os botões virtuais acarretam menor sensibilidade tátil.	CPM1	1
AR4P3	Apenas um comando de voz é reconhecido por vez.	CPM2	2
AR4P4	Há baixa privacidade ao emitir comandos de voz.	CPM2	1
AR4P5	Há diminuição do desempenho do reconhecimento em condições de ruído.	CPM2	1
AR4P6	Há diminuição do desempenho do reconhecimento devido à entonação e à acentuação.	CPM2	1
AR4P7	Há dificuldade para ativar comando de voz.	CPM2	1
AR4P8	Há necessidade de mentalizar instrução por voz, aumentando carga de memória do indivíduo.	CPM2	1
AR4P9	O reconhecimento de voz funciona apenas em alguns aplicativos.	CPM2	1
AR4P10	O uso de comandos de voz é computacionalmente custoso.	CPM2	1
AR4P11	Há ausência de retorno ao usuário, ao interagir com alguns elementos de interface.	CPM6	1
AR4P12	Há dificuldade para compreender diferentes padrões vibratórios.	CPM6	1
AR4P13	Há dificuldade para compreender a orientação da interface, utilizando apenas o retorno auditivo.	CPM6	1
AR4P14	Retorno auditivo é prejudicado em ambientes ruidosos.	CPM6	2
AR4P15	Usar apenas o retorno auditivo não é o suficiente para a interação.	CPM6	1

Fonte: [Damaceno, Braga e Chalco \(2016\)](#).

A [Tabela 9](#) mostra os problemas relacionados à entrada de dados (CPM3) com o número de citações para cada problema. Os problemas que mencionavam teclado físico de dispositivos móveis foram desconsiderados, pois a aplicação desenvolvida suporta apenas *smartphones*.

Tabela 9 – Problemas relacionados à entrada de dados (CPM3).

Código	Problema	Citações
AR4P16	A digitação de textos é lenta em teclados QWERTY virtuais.	2
AR4P17	As teclas mais distantes das bordas são mais difíceis de encontrar do que as mais próximas das bordas, em teclados virtuais QWERTY.	1
AR4P18	É preciso conhecer previamente Braille para ter bom desempenho de digitação utilizando esta modalidade.	2
AR4P19	É preciso trocar o modo do teclado virtual, para acessar determinados caracteres.	1
AR4P20	Há ausência de marca tátil para o número 5, no teclado numérico virtual, e para as letras “F” e “J” no teclado QWERTY virtual.	2
AR4P21	Há erros ao corrigir caracteres digitados equivocadamente, substituindo por fonemas semelhantes, em teclados virtuais.	1
AR4P22	Há erros de omissão de caracteres, faltando um ou mais ao digitar palavras em teclados virtuais.	1
AR4P23	Há necessidade de confirmação de cada caractere digitado em teclados virtuais.	1
AR4P24	Há necessidade de navegar pelo teclado virtual para localizar os caracteres desejados.	1
AR4P25	Há um segundo de espera para entrar com cada tecla em teclados virtuais.	1
AR4P26	O teclado numérico virtual é denso dificultando, a interação.	1

Fonte: [Damaceno, Braga e Chalco \(2016\)](#).

A [Tabela 10](#) lista os problemas relacionados à interação por gestos (CPM4) com o número de citações para cada problema encontrado.

Tabela 10 – Problemas relacionados à interação por gestos (CPM4).

Código	Problema	Citações
AR4P27	Baixa flexibilidade de ângulo e velocidade dos gestos dificultam o reconhecimento.	1
AR4P28	Gestos com forma da letra ‘L’ são difíceis de fazer.	2
AR4P29	Gestos com formas geométricas fechadas (círculo e triângulo) são difíceis de fazer.	1
AR4P30	Gestos com formas geométricas são lentos de se fazer.	1
AR4P31	Conflito na desambiguação entre dois toques com um dedo e três toques com um dedo.	1
AR4P32	Dificuldade para fazer gestos estando em movimento.	1
AR4P33	Dificuldade para fazer gestos próximos à barra superior de sistemas.	1
AR4P34	Dificuldade para fazer gestos representados por símbolos.	2
AR4P35	Dificuldade para fazer o gesto de dois toques com um dedo.	1
AR4P36	Dificuldade para se localizar na tela para realizar gestos.	1
AR4P37	Erros na detecção de gestos multitoque.	1
AR4P38	Falha de interpretação de gestos em geral, pelo sistema.	4
AR4P39	Mudança indevida de foco ao tentar fazer o gesto dois toques com um dedo.	1
AR4P40	Não é possível alterar mapeamento dos gestos às funções do sistema.	1
AR4P41	Não há consistência de gestos entre diferentes sistemas.	1
AR4P42	Não há gestos que acionam as principais funções do sistema.	1
AR4P43	O toque acidental na tela, com outro dedo, prejudica o reconhecimento de gestos.	1
AR4P44	Os manuais de explicação de como fazer gestos de toque não são eficientes.	3
AR4P45	Conflito entre do aplicativo gestos e os do leitor de tela do sistema.	1

Fonte: [Damaceno, Braga e Chalco \(2016\)](#).

Por fim, são listados, na [Tabela 11](#), os problemas relacionados a leitores de tela (CPM5) com o número de citações.

Tabela 11 – Problemas relacionados a leitores de tela (CPM5).

Código	Problema	Citações
AR4P46	A leitura é linear, demorando para se ter noção global da interface.	2
AR4P47	A pronúncia de algumas palavras é problemática.	1
AR4P48	A voz do leitor de tela é artificial.	1
AR4P49	Alguns elementos de interface não são lidos.	3
AR4P50	Há baixa familiaridade com o leitor de tela de dispositivos móveis.	1
AR4P51	Há conflito ao usar o leitor de tela do sistema em conjunto com o leitor embutido em aplicativos.	2
AR4P52	Há desconforto ao ouvir o leitor de tela em ambientes ruidosos.	2
AR4P53	Há leitura de apenas o que está em foco.	1
AR4P54	Não há controle de velocidade de leitura.	2
AR4P55	Não há um botão para interromper a leitura imediatamente.	1
AR4P56	O foco do leitor de tela muda indevidamente.	2
AR4P57	O foco do leitor de tela não possui uma ordem de navegação lógica.	2
AR4P58	O leitor de tela é lento.	1
AR4P59	O texto lido é, por vezes, inadequado.	1

Fonte: [Damaceno, Braga e Chalco \(2016\)](#).

3.3.5 Observation Based Analysis on the Use of Mobile Applications for Visually Impaired Users

O estudo realizou uma análise, envolvendo 5 PDV, com o objetivo de validar se a falta dos requisitos de acessibilidade levantados em um trabalho anterior realmente impactavam na utilização de *apps* móveis por PDV.

Tabela 12 – Categorias dos requisitos encontrados.

Código	Categoria
CRED1	Feedbacks audíveis
CRED2	Adaptação das informações visuais
CRED3	Navegação

Fonte: [Siebra et al. \(2016\)](#).

Os requisitos foram divididos em 3 categorias, como mostra a [Tabela 12](#). Baseados na análise dos resultados, o estudo qualificou os requisitos em 3 níveis (Essencial, Desejável e Não observado).

Como os requisitos “não observados”, de acordo com o artigo, não foram mencionados pelos participantes dos testes, apenas os requisitos essenciais e desejáveis são listados na [Tabela 13](#). Somente um requisito foi classificado como desejável pelo estudo, o AR5R7, o restante foi classificado como essencial.

Tabela 13 – Requisitos essenciais e desejáveis focados em DV.

Código	Requisito	Categoria
AR5R1	O nome do caractere que está sendo digitado deve ser ouvido.	CRED1
AR5R2	Nomes de elementos e imagens na tela devem ser ouvidos ao serem tocados ou selecionados.	CRED1
AR5R3	Feedback de ações/interações devem ser claros e fornecidos de forma tátil, voz ou eventos sonoros.	CRED1
AR5R4	Estratégias para o uso de leitores de tela (ex.: atalhos para navegar na tela de forma mais eficiente.)	CRED1
AR5R5	Prover uma chave “home” tátil de acesso fácil e rápido para que um usuário possa retornar a um lugar conhecido.	CRED2
AR5R6	Prover documentação em formatos alternativos, utilizando fontes grandes.	CRED3
AR5R7	Permitir customizações pelo usuário e evitar que essas preferências sejam perdidas.	CRED3
AR5R8	Apresentar amplificador com zoom ajustável.	CRED3
AR5R9	Prover equivalências textuais claras para evitar erros quando os textos são lidos na tela.	CRED3
AR5R10	Brilho, contraste e cores ajustáveis.	CRED3
AR5R11	Prover alertas informativos por outros canais além do visual (ex.: voz.)	CRED3

Fonte: [Siebra et al. \(2016\)](#).

3.3.6 Prioritization of mobile accessibility guidelines for visual impaired users

O artigo apresenta uma proposta de priorização de diretrizes de acessibilidade que resultaram de estudos anteriores. Essas diretrizes foram baseadas no eMAG, entretanto, diretrizes como as da BCC (*BBC Mobile Accessibility Guidelines*) e recomendações da plataforma Android também foram consideradas. Para criação do *ranking*, o estudo utilizou um questionário que foi respondido 103 vezes, sendo 66 dessas respostas de PDV, nas quais a análise se concentrou.

O estudo dividiu as diretrizes em 6 categorias que podem ser visualizadas na [Tabela 14](#).

Tabela 14 – Categorias das diretrizes de acessibilidade *mobile* baseadas no eMAG.

Código	Categoria
AR6CE	Estrutura
AR6CC	Comportamento
AR6CCI	Conteúdo/Informação
AR6CAD	Apresentação/ <i>Design</i>
AR6CM	Multimídia
AR6CF	Formulários

Fonte: [Quispe, Scatalon e Eler \(2020\)](#).

O estudo considerou a priorização para 4 grupos diferentes, baseados no tipo de DV (baixa visão, visão parcial e os 2 tipos de cegueira: legal e total). E os resultados mostraram que existiam diferenças notáveis na percepção das diretrizes entre os grupos.

Quadro 6 – Priorização de diretrizes de acessibilidade para usuários com DV.

Id	Visão parcial	Baixa visão	Cegueira legal	Cegueira total	Todas as DV
1	AR6D7, AR6D28	AR6D9	AR6D24, AR6D28	AR6D8, AR6D22, AR6D23	AR6D8
2	AR6D25	AR6D11, AR6D28	AR6D7, AR6D11, AR6D22, AR6D25	AR6D1, AR6D6, AR6D25, AR6D28	AR6D25
3	AR6D24	AR6D25	AR6D1, AR6D27	AR6D7, AR6D9, AR6D15, AR6D24, AR6D27	AR6D7, AR6D9, AR6D22
4	AR6D9, AR6D22	AR6D7, AR6D22	AR6D9	AR6D11	AR6D11, AR6D24
5	AR6D11	AR6D10	AR6D10	AR6D10	AR6D27
6	AR6D10, AR6D27	AR6D15	AR6D8, AR6D23		AR6D1
7	AR6D8, AR6D15	AR6D1, AR6D6, AR6D27	AR6D6		AR6D8
8	AR6D6	AR6D8	AR6D15		AR6D6, AR6D23
9	AR6D1	AR6D23, AR6D24			AR6D10, AR6D15
10	AR6D23				

Fonte: [Quispe, Scatalon e Eler \(2020\)](#).

A partir desses resultados, o trabalho relacionou as diretrizes com as percepções de cada grupo e criou a lista de priorização que pode ser vista no [Quadro 6](#). No qual a coluna “Id” informa a ordem de priorização e os códigos que estão nas demais são listados na [Tabela 15](#) junto com as diretrizes.

Tabela 15 – Diretrizes de acessibilidade *mobile* baseadas no eMAG.

Código	Diretriz	Categoria
AR6D1	Elementos de tela devem ser organizados de maneira lógica e semântica.	AR6CE
AR6D2	As telas devem apresentar sequência lógica de leitura para navegação entre <i>links</i> , controles de formulário e outros elementos.	AR6CE
AR6D3	<i>Links</i> na tela devem ser organizados para evitar confusão.	AR6CE
AR6D4	Informações devem ser divididas em grupos específicos para facilitar a procura e leitura dos conteúdos.	AR6CE
AR6D5	Usuários devem ser informados se <i>links</i> abrem novas telas para poderem decidir se querem ou não sair da tela atual.	AR6CE
AR6D6	Todas as funcionalidades na tela devem estar disponíveis a partir do teclado.	AR6CC
AR6D7	Todos os elementos de <i>interface</i> na tela devem ser acessíveis.	AR6CC
AR6D8	Redirecionamento automático de telas não deve acontecer.	AR6CC
AR6D9	Em telas com limite de tempo, deve haver opções para desligar ou ajustar o tempo.	AR6CC
AR6D10	Não deve haver efeitos visuais piscantes, intermitentes ou cintilantes na tela.	AR6CC
AR6D11	Conteúdos animados não devem iniciar automaticamente.	AR6CC
AR6D12	A linguagem utilizada na tela deve ser especificada.	AR6CCI
AR6D13	Mudanças na linguagem dos conteúdos sempre devem ser especificadas.	AR6CCI
AR6D14	Títulos de telas devem ser descritivos, informativos e representativos com relação ao conteúdo principal.	AR6CCI
AR6D15	Deve haver algum mecanismo para indicar ao usuário onde ele está no momento, no conjunto de telas.	AR6CCI
AR6D16	Alvos de <i>links</i> devem ser identificados claramente, incluindo informações sobre se estão funcionando ou se direcionam para outra tela.	AR6CCI
AR6D17	Todas as imagens devem possuir descrição textual.	AR6CCI
AR6D18	Documentos em formatos acessíveis devem estar disponíveis.	AR6CCI
AR6D19	Quando uma tabela é utilizada na tela, título e sumário apropriados devem ser fornecidos.	AR6CCI
AR6D20	Os textos nas telas devem ser fáceis de ler e entender.	AR6CCI
AR6D21	Totais as siglas, abreviações e palavras incomuns na tela devem possuir explicação.	AR6CCI
AR6D22	Deve haver uma taxa mínima de contraste entre as cores de fundo e as de frente.	AR6CAD
AR6D23	Características sensoriais (ex. cores, formas e sons) não podem ser o único significado para distinguir elementos de tela.	AR6CAD
AR6D24	O elemento ou área em foco deve ser evidente visualmente.	AR6CAD
AR6D25	Vídeos que não incluem áudio devem fornecer alternativas como legendas.	AR6CM
AR6D26	Deve haver alternativas a conteúdo de áudio (ex. transcrição ou linguagem de sinais).	AR6CM
AR6D27	Conteúdos visuais que não estão disponíveis como áudio devem ser descritos.	AR6CM
AR6D28	Devem haver mecanismos para controlar áudios da aplicação.	AR6CM
AR6D29	Devem haver mecanismos para controlar animações que iniciam automaticamente.	AR6CM
AR6D30	Botões de imagem ou conteúdos de áudio em formulários devem possuir alternativas textuais.	AR6CF
AR6D31	Todos os campos do formulário devem ser identificados.	AR6CF
AR6D32	Uma ordem lógica na navegação pelo formulário deve ser garantida.	AR6CF
AR6D33	Não devem haver mudanças automáticas quando um elemento do formulário é focado, para não confundir ou desorientar o usuário.	AR6CF
AR6D34	Formulários devem possuir instruções de preenchimento.	AR6CF
AR6D35	Erros de entrada devem sempre ser descritos e as submissões de dados confirmadas.	AR6CF

Fonte: [Quispe, Scatalon e Eler \(2020\)](#).

3.4 Análise dos Resultados

Os resultados do MSL mostram que o Android foi a principal plataforma dos *apps* desenvolvidos pelos estudos. Sendo 9 aplicativos desenvolvidos somente para Android, 4 multiplataforma (Android e iOS) e 2 apenas para iOS, como mostra a [Tabela 16](#).

Tabela 16 – Tecnologias utilizadas no desenvolvimento e plataforma alvo das aplicações.

Artigo	Tecnologias	Plataforma
AM1	<i>Cordova Framework</i>	<i>Android e iOS</i>
AM2	<i>MD², Xtend, Java, Eclipse</i>	<i>Android e iOS</i>
AM3	<i>Unity 3D engine, Java</i>	Android
AM4	<i>Android Studio 2.0</i>	Android
AM5	Não informado	Android
AM6	<i>Java, Android Studio, Accessibility Scanner App, Test Lab</i>	Android
AM7	Não informado	Android
AM8	Não informado	iOS
AM9	Não informado	iOS
AM10	<i>React Native</i>	<i>Android e iOS</i>
AM11	Não informado	Android
AM12	<i>Java, Android Studio, API Airy</i>	Android
AM13	Não informado	Android
AM14	Não informado	Android
AM15	<i>React Native</i>	<i>Android e iOS</i>

Fonte: Autor.

Embora todos os estudos tenham mencionado a plataforma para qual o *app* foi desenvolvido, como pode ser observado na [Tabela 16](#), boa parte deles, 7 no total, não mencionam as tecnologias utilizadas. Com isso, *Java* destacou-se como a principal linguagem, utilizada em pelo menos 4 estudos, para o desenvolvimento das soluções. Enquanto o *React Native* apareceu como principal o *framework* para desenvolvimento multiplataforma com duas aplicações.

Quanto às técnicas relacionadas à acessibilidade utilizadas no desenvolvimento das soluções apresentadas nos artigos, são listadas na [Tabela 17](#) as principais identificadas no MSL. Nessa tabela foi atribuído um código de referência para cada técnica listada.

Tabela 17 – Técnicas utilizadas no desenvolvimento das soluções de acessibilidade do MSL.

Código	Técnicas	Artigos
TAM1	Contraste de cor para garantir diferentes níveis de acessibilidade	AM1, AM9, AM10, AM12, AM13
TAM2	Descrição textual dos elementos visuais	AM1, AM2, AM3, AM5, AM6, AM7, AM8, AM9, AM10, AM11, AM12, AM13, AM15
TAM3	Escala SUS para avaliação da usabilidade da aplicação	AM4, AM5, AM7
TAM4	Elementos chave de interação sempre posicionados na mesma parte da tela, em locais de fácil acesso	AM10, AM15
TAM5	<i>Feedback</i> por vibração	AM3, AM14
TAM6	<i>Feedback</i> por voz por meio de TTS	AM3, AM5, AM6, AM7, AM9, AM11, AM12, AM13, AM14
TAM7	Interação alternativa fazendo uso de gestos	AM1, AM7, AM10, AM12, AM15
TAM8	Personalização de pontos da <i>interface</i> que afetam a acessibilidade	AM2, AM6, AM8, AM9, AM12
TAM9	Possibilidade de revisar as mensagens escritas por meio de TTS	AM7
TAM10	Reconhecimento de voz	AM4, AM6, AM7, AM9, AM12
TAM11	Tamanho da fonte das letras ampliado ou personalizável	AM6, AM8, A12
TAM12	Tamanho dos botões e espaçamentos adequados à PDV	AM9
TAM13	Tamanho dos ícones e componentes adaptáveis de acordo com tamanho da tela	AM10
TAM14	Todos os elementos visíveis na tela sem necessidade de rolagem	AM10
TAM15	Utilização de efeitos sonoros para contextualizar o usuário	AM3, AM6, AM13

Fonte: Autor.

A coluna “Artigos” na [Tabela 17](#) indica em quais artigos (representados por código) cada técnica foi identificada. Como as descrições das técnicas identificadas variaram de acordo com os artigos, elas foram representadas nessa tabela com uma descrição genérica, possibilitando a identificação dos artigos que utilizaram as mesmas técnicas.

Os problemas mapeados pelos estudos relacionados, apresentados na [Tabela 18](#), foram generalizados e divididos de acordo com as categorias da [Tabela 14](#), definidas por [Quispe, Scatalon e Eler \(2020\)](#). O autor utilizou o eMAG como base, alterando apenas a primeira categoria chamada “Marcação”, que refere-se à linguagens de marcação da *web*, para “Estrutura”, ainda mantendo o mesmo sentido de organização e estrutura dos elementos de *interface*.

Tabela 18 – Principais problemas identificados pelos estudos relacionados.

Código	Problema	Categoria	Códigos de referência dos problemas
CPER1	<i>Feedback</i> auditivo não é suficiente para a interação	Conteúdo / Informação	AR4P14, AR4P15, AR4P47, AR4P50, AR4P52, AM1P3
CPER2	Apresentação dos conteúdos	Apresentação / Design	AM1P1, AM1P5, AM1P7, AM1P10, AR1P5, AR1P6, AR1P7, AR1P8, AR1P9, AR1P11, AR1P14
CPER3	Descrição textual inadequada/inexistente dos elementos de tela	Conteúdo / Informação	AR1P8, AM1P4, AR1P1, AR1P2, AR1P4, AR1P7, AR1P10, AR4P11, AR1P14, AR4P49, AR4P59, AM1P4
CPER4	Dificuldades ao navegar pela aplicação	Estrutura	AM1P1, AM1P5, AM1P6, AM1P8, AM1P9, AR1P5, AR1P8, AR4P46
CPER5	Dificuldades com a utilização do teclado virtual padrão	Formulários	AR4P16, AR4P17, AR4P19, AR4P20, AR4P21, AR4P22, AR4P23, AR4P24, AR4P25, AR4P26
CPER6	Dificuldades relacionadas a botões virtuais	Apresentação / Design	AR4P1, AR1P2, AR4P2, AR1P10
CPER7	Dificuldades na utilização de gestos	Comportamento	AR4P27, AR4P28, AR4P29, AR4P30, AR4P31, AR4P32, AR4P33, AR4P34, AR4P35, AR4P36, AR4P37, AR4P38, AR4P39, AR4P40, AR4P41, AR4P42, AR4P43, AR4P44, AR4P45
CPER8	Dificuldades com leitor de tela	Comportamento	AM1P7, AM1P2, AR4P46, AR4P47, AR4P48, AR4P50, AR4P51, AR4P53, AR4P54, AR4P55, AR4P56, AR4P57, AR4P58, AM1P3
CPER9	Funcionalidades confusas ou não claras	Estrutura	AR1P3, AR1P4, AR1P6, AR1P9, AR1P10, AR1P11, AM1P6, AM1P8
CPER10	Obstáculos relacionados ao reconhecimento de voz	Comportamento	AR4P3, AR4P4, AR4P5, AR4P6, AR4P7, AR4P8, AR4P9, AR4P10

Fonte: Autor.

A [Tabela 19](#) relaciona os códigos de referência dos problemas listados na [Tabela 18](#) com os códigos das diretrizes definidas pelos estudos relacionados e das principais técnicas identificadas nos artigos do MSL, reunidas na [Tabela 17](#).

Tabela 19 – Diretrizes e técnicas relacionadas à cada tipo de problema.

Código do problema	Diretrizes/Técnicas
CPER1	AR3D8, AR3D9, AR3D11, AR5R10, AR6D22, AR6D24, TAM5, TAM11, TAM13, TAM14
CPER2	AR3D1, AR3D2, AR3D5, AR3D8, AR3D9, AR3D11, AR5R6, AR5R8, AR5R10, AR6D3, AR6D4, AR6D10, AR6D14, AR6D24, TAM1, TAM8, TAM11
CPER3	AR5R2, AR5R9, AR6D5, AR6D7, AR6D12, AR6D13, AR6D14, AR6D16, AR6D17, AR6D19, AR6D20, AR6D21, AR6D23, AR6D27, TAM2, TAM14
CPER4	AR3D3, AR3D4, AR3D5, AR3D7, AR3D9, AR5R4, AR5R5, AR6D1, AR6D2, AR6D3, AR6D5, AR6D8, AR6D15, AR6D32, TAM4
CPER5	AR3D6, AR5R1, TAM9, TAM10
CPER6	AR3D11, AR6D30, TAM12, TAM13
CPER7	AR3D6, AR3D10, TAM6, TAM7, TAM9
CPER8	AR5R1, AR5R2, AR5R3, AR5R4, AR5R6, AR5R8, AR5R9, AR6D1, AR6D2, AR6D3, AR6D12, AR6D13, AR6D14, AR6D17, TAM2, TAM4, TAM6, TAM14
CPER9	AR3D4, AR3D5, AR3D7, AR5R9, AR6D1, AR6D2, AR6D3, AR6D4, AR6D16, TAM4
CPER10	AR5R1, TAM10

Fonte: Autor.

Como pode ser observado na [Tabela 19](#), todos os tipos de problemas listados possuem alguma técnica da [Tabela 17](#), das técnicas identificadas no MSL, relacionada.

3.4.1 CPER1: *Feedback auditivo não é suficiente para a interação*

Embora o retorno auditivo tenha sido o principal recurso utilizado para acessibilidade de PDV, como pode ser visto na [Tabela 17](#), existem várias limitações quanto à essa solução, mesmo considerando que todos os elementos de tela tenham as descrições adequadas para narração pelos leitores de telas.

Tais limitações referem-se a: entendimento da orientação e organização da *interface*, utilização em ambientes ruidosos, falta privacidade e pronúncia das palavras ([DAMACENO; BRAGA; CHALCO, 2016](#)). Com isso, faz-se necessário o suporte à outras formas de retorno acessível para o usuário, mediante a solução de outros problemas listados, em destaque o CPER2 relacionado à *interface*.

3.4.2 CPER2: Apresentação dos conteúdos

Problemas como a falta de consistência no *layout*, a má organização dos conteúdos das telas e funcionalidades e sequências de interação confusas podem levar a um não entendimento

da sequência de leitura e gerar sobrecarga de informações para o usuário (SHERA et al., 2021; RIEGER et al., 2020).

Diante desses problemas, técnicas como posicionar os elementos chave de interação sempre na mesma posição, adaptar o tamanho dos componentes de acordo com o tamanho da tela e evitar colocar muito conteúdo na tela para evitar a rolagem podem ser bastante úteis (MASCETTI et al., 2019; CANTÙ et al., 2018). Suporte ao alto contraste, esquemas alternativos de cores, ampliação da tela e fontes maiores e mais legíveis também podem ajudar nesse sentido (KIM et al., 2019; OLIVEIRA et al., 2019).

3.4.3 CPER3 e CPER8: Problemas relacionados às descrições dos elementos e leitores de tela

Problemas relacionados a leitores de tela e descrições textuais dos elementos apareceram com alta frequência na Tabela 18, justificando a técnica TAM2, da Tabela 17, ter sido a mais utilizada nos estudos e a mais discutida por desenvolvedores no *Stack Overflow* (VENDOME et al., 2019).

A técnica TAM2 refere-se à descrição textual dos elementos de tela e sua utilização foi mencionada explicitamente em 13 dos 15 artigos selecionados no MSL. Além dela, a técnica TAM6 que fornece o *feedback* auditivo nativamente nas aplicações, por meio TTS também foi uma das técnicas mais utilizadas, como pode ser visto na Tabela 17. Assim, mostrando-se uma possível alternativa para o problema com *crash* do leitor de telas citado por Shera et al. (2021).

3.4.4 CPER4: Dificuldades ao navegar pela aplicação

Diversos problemas de navegação associados também ao CPER2, de apresentação dos conteúdos, foram encontrados, particularmente: o não entendimento do fluxo de tarefas e a falta de consistência na organização dos conteúdos/*layout* (SHERA et al., 2021; RIEGER et al., 2020; QUISPE; SCATALON; ELER, 2020). Outro problema dá-se pela forma de navegação entre os elementos dos leitores de tela, que é linear, demorando para ter-se uma noção geral da *interface* (DAMACENO; BRAGA; CHALCO, 2016).

Portanto, a preocupação com uma navegação intuitiva, posicionando elementos chave como caixas de pesquisa sempre no mesmo local mostra-se fundamental para resolver alguns desses problemas (KIM et al., 2019; MASCETTI et al., 2019; CANTÙ et al., 2018). A utilização de estratégias para o uso de leitores de tela, como atalhos para facilitar a navegação do usuário com DV pela aplicação, principalmente um que possibilite que o usuário retorne a um lugar conhecido, como a tela inicial do *app*, foi definida como essencial por Siebra et al. (2018), a partir de uma análise realizada com envolvimento de 5 PDV.

3.4.5 CPER5: Dificuldades com a utilização do teclado virtual padrão

Vários problemas relacionados à utilização do teclado virtual padrão foram identificados, como pode ser visto na [Tabela 18](#). Tornando ainda mais evidente a necessidade de possibilitar várias alternativas de interação com as aplicações para usuários com DV, tanto de entrada de dados quanto de saída.

Algumas soluções como a possibilidade de *feedback* auditivo dos caracteres que estão sendo digitados e a de revisão do conteúdo escrito ao finalizar a digitação foram identificadas e validadas ([SIEBRA et al., 2016](#); [DUARTE et al., 2017](#)). Além disso, o reconhecimento de voz, definido como diretriz por [Kim et al. \(2019\)](#), também foi utilizado como alternativa ao uso de teclados virtuais por um terço dos estudos do MSL, como pode ser visto na [Tabela 17](#).

3.4.6 CPER6: Dificuldades relacionadas a botões virtuais

No conjunto de problemas relacionados a botões levantado por [Rieger et al. \(2020\)](#), são mencionados a falta de informações e as funcionalidades confusas dos mesmos, porém, esses problemas estão associados ao CPER3, de descrições textuais inadequadas/inexistentes das quais foram mencionadas outras soluções. Outros problemas com relação a esses botões são: a grande proximidade entre eles e a menor sensibilidade tática, em comparação com os botões físicos ([DAMACENO; BRAGA; CHALCO, 2016](#)).

O uso de fontes legíveis com tamanho ampliado ou personalizável e o tamanho e espaçamento adequado para os botões virtuais são essenciais, visto que podem dificultar a visualização para PDV parcial quando não estão utilizando leitores de tela ([SHIN et al., 2017](#); [KIM et al., 2019](#)).

3.4.7 CPER7: Dificuldades na utilização de gestos

Como pôde-se visualizar na [Tabela 18](#), problemas relacionados a gestos foram os mais frequentes, com a maioria dos usuários que participaram do estudo de [Oliveira et al. \(2019\)](#) relatando ter enfrentado dificuldades, deixando algum comentário negativo a respeito. Outro motivo para a dificuldade está na diferença entre os gestos das aplicações desenvolvidas e os habituais de cada plataforma ([LEPORINI; PALMUCCI, 2017](#)).

Contudo, resultados de outros estudos indicam que a utilização de gestos simples gerou *feedbacks* positivos ([DUARTE et al., 2017](#); [CANTÙ et al., 2018](#)).

3.4.8 CPER9: Funcionalidades confusas ou não claras

Funcionalidades com apresentações não claras ou confusas, tanto pela descrição textual audível quanto pela visual, acabam causando confusão nos usuários, os fazendo pensar que uma funcionalidade faz uma coisa quando na verdade faz outra ([RIEGER et al., 2020](#)).

Esses problemas ressaltam ainda mais a importância de fornecer descrições textuais audíveis claras dos elementos de tela e funcionalidades da aplicação, como é apontado no tópico específico do CPER3. Ao mesmo tempo, reforça também a importância da apresentação visual dos conteúdos mencionada no tópico do CPER2.

3.4.9 CPER10: Obstáculos relacionados ao reconhecimento de voz

Diversas limitações com relação ao reconhecimento de voz foram levantadas por [Dama-ceno, Braga e Chalco \(2016\)](#), algumas delas são as mesmas citadas no tópico CPER1, sobre o *feedback auditivo* não ser suficiente, por questões de privacidade, ruídos de ambientes e pronúncia das palavras. Da mesma forma que o CPER1, faz-se necessário oferecer outros métodos de entrada alternativos para os usuários, como resolver os problemas do tópico CPER5, associados a teclados virtuais.

Os outros problemas citados pelo autor, no mesmo estudo, foram o de apenas um comando ser reconhecido por vez, a necessidade de mentalizar as instruções por voz, aumentando a carga de memória dos usuários, e o uso desses comandos ser computacionalmente custoso. Para esses problemas não foram identificadas soluções nos estudos do MSL ou nos estudos relacionados.

3.5 Respostas das Questões do Protocolo

Nesta seção são respondidas as questões definidas no Protocolo de MSL, apresentado no início deste capítulo, levando em consideração a análise dos resultados da seção anterior. Assim, seguem:

1. Quais são as principais soluções de acessibilidade para PDV utilizadas no desenvolvimento de aplicações móveis?
 - a) Foi observado o cumprimento de diretrizes de acessibilidade para aplicações móveis propostas por entidades como Google¹⁰, SIDI¹¹ e BBC¹²; não foi identificado, porém, o uso das diretrizes da Apple¹³;
 - b) Realização de estudos visando identificar problemas enfrentados por usuários com DV no uso de aplicativos móveis e definir diretrizes para solucionar cada problema;
 - c) Utilização de diretrizes de acessibilidade da W3C¹⁴ adaptadas da *web* para o contexto de aplicações móveis;
 - d) Utilização de ferramentas para realização de testes de acessibilidade automatizados.
2. Quais são as tecnologias utilizadas no desenvolvimento dessas soluções?
 - a) A linguagem Java no desenvolvimento de aplicações Android;
 - b) Os *frameworks* React Native, Cordova e MD² no desenvolvimento multiplataforma;
 - c) As IDEs Android Studio e Eclipse;
 - d) Unity 3D no desenvolvimento de jogos para Android;
 - e) As ferramentas Accessibility Scanner App, MATE e Test Lab para realização de testes de acessibilidade automatizados.
3. Para quais plataformas as soluções foram propostas?
 - a) 9 apenas para Android;
 - b) 2 apenas para iOS;
 - c) 4 multiplataforma, para Android e iOS.
4. Quem são os públicos alvos dessas soluções?

As soluções visaram atender diversas necessidades de PDV. Assim, foram criadas aplicações para diferentes públicos com DV, como crianças e estudantes, e com diversos tópicos específicos, como livros, medicamentos e clima.

¹⁰ <<https://developer.android.com/guide/topics/ui/accessibility/apps>>

¹¹ <<https://www.sidi.org.br/guiadeacessibilidade>>

¹² <<https://www.bbc.co.uk/accessibility/forproducts/guides/mobile>>

¹³ <<https://developer.apple.com/design/human-interface-guidelines/accessibility/overview/introduction/>>

¹⁴ <<https://www.w3.org/TR/mobile-bp/summary>>

Tabela 20 – Relação de técnicas adotadas pelos artigos e propostas para o DiaVision.

Aplicativo/ Técnicas	AM1	AM2	AM3	AM4	AM5	AM6	AM7	AM8	AM9	AM10	AM11	AM12	AM13	AM14	AM15	DiaVision
TAM1	✓	X	X	X	X	X	X	X	✓	✓	X	✓	✓	X	X	✓
TAM2	✓	✓	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	✓	✓
TAM3	X	X	X	✓	✓	X	✓	X	X	X	X	X	X	X	X	X
TAM4	X	X	X	X	X	X	X	X	X	✓	X	X	X	X	✓	✓
TAM5	X	X	✓	X	X	X	X	X	X	X	X	X	X	✓	X	X
TAM6	X	X	✓	X	✓	✓	✓	X	✓	X	✓	✓	✓	✓	X	✓
TAM7	✓	X	X	X	X	X	✓	X	X	✓	X	✓	X	X	X	X
TAM8	X	✓	X	X	X	✓	X	✓	✓	X	X	✓	X	X	X	✓
TAM9	X	X	X	X	X	X	✓	X	X	X	X	X	X	X	X	✓
TAM10	X	X	X	✓	X	✓	✓	X	✓	X	X	✓	X	X	X	X
TAM11	X	X	X	X	X	✓	X	✓	X	X	X	✓	X	X	X	✓
TAM12	X	X	X	X	X	X	X	X	✓	X	X	X	X	X	X	✓
TAM13	X	X	X	X	X	X	X	X	X	✓	X	X	X	X	X	✓
TAM14	X	X	X	X	X	X	X	X	X	✓	X	X	X	X	X	✓
TAM15	X	X	✓	X	X	✓	X	X	X	X	X	X	✓	X	X	X

Fonte: Autor.

3.6 Técnicas Propostas para o DiaVision

A Tabela 20 relaciona as técnicas para solução de problemas de acessibilidade à DV, listadas na Tabela 17, utilizadas no desenvolvimento das aplicações nos estudos do MSL, com as propostas para implementação no aplicativo desenvolvido no presente trabalho, o DiaVision.

3.7 Considerações Finais

Este capítulo buscou realizar um levantamento, por meio de um estudo de MSL, do estado da arte acerca dos problemas de acessibilidade enfrentados por PDV e das soluções para os mesmos. A partir da análise dos resultados, ficou evidente a necessidade de preocupação com acessibilidade no processo desenvolvimento de soluções para dispositivos móveis. Sendo identificadas 15 das principais soluções para resolver diferentes problemas relacionados à DV, das quais 10 foram consideradas no desenvolvimento da solução proposta, que será descrita no próximo capítulo.

4

Planejamento do DiaVision

Neste capítulo é apresentado o processo de planejamento do DiaVision, com os requisitos, estórias de usuário, casos de uso e demais artefatos que fazem parte do levantamento de requisitos e análise, oferecendo uma visão ampla da aplicação desenvolvida.

4.1 Descrição do Projeto

Este projeto foi desenvolvido em parceria com a mestrande, Débora Almeida Silveira Sobral, do Programa de Pós-graduação Profissional em Gestão e Inovação Tecnológica em Saúde (PPGITS), também orientanda da Profa. Dra. Adicinéia Aparecida de Oliveira. Com objetivo de desenvolver uma aplicação móvel como ferramenta de auxílio à educação e ao autocuidado de pacientes diabéticos com acuidade visual prejudicada, acessível à PDV.

Para tanto, em [Sobral \(2021\)](#) foi realizado um levantamento de referencial teórico e tecnológico sobre o DM, aplicativos móveis e deficiência visual. Assim, reunindo as principais funcionalidades e soluções que o aplicativo desenvolvido deveria adotar como requisitos para que atendesse às necessidades desse público-alvo.

4.2 Busca de Anterioridade

Em [Sobral \(2021\)](#), realizou-se uma busca de anterioridade na base do Instituto Nacional de Propriedade Industrial (INPI) e nas lojas de aplicativos Google Play (Android) e Apple Store (iOS), visando identificar os *softwares* e funcionalidades já existentes sobre DM e DV no mercado. Conforme esse trabalho, a [Tabela 21](#) relaciona os *apps* encontrados nas lojas de aplicativos às principais funcionalidades propostas para o DiaVision.

Tabela 21 – Relação de funcionalidades dos *apps* encontrados nas lojas de aplicativos.

LOJA VIRTUAL	APLICATIVO	GLICEMIA	MEDICAÇÃO	PROFISSIONAIS DA SAÚDE	CENTROS DE SAÚDE	AUTO CUIDADO	AUMENTAÇÃO	VISÃO	RINS	PÉS	ATIVIDADE FÍSICA	EMERGÊNCIA	ACESSIBILIDADE VISUAL
AppStore	mySugar	✓	PRO	± PRO	✗	±	PRO	✗	✗	✗	✗	✗	✗
AppStore	Índice Carga Glicêmica Comidas	✓	✗	✗	✗	±	✓	✗	✗	✗	✗	✗	✗
AppStore	Glicés Diabetes e Glicemia	✓	✗	±	✗	✗	✓	✗	✗	✗	✗	✗	✗
AppStore	Contagem de Carboidratos - SBD	✓	✗	±	✗	✗	✓	✗	✗	✗	✗	✗	✗
AppStore	Aplicativo ACCU-CHEK® Connect	✓	✗	✓	✗	✗	PRO	✗	✗	✗	PRO	✗	✗
AppStore	Diabetes Pro	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
AppStore	Glicemia (Os pacientes diabéticos)	✓	✗	PRO	✗	✗	✗	✗	✗	✗	PRO	✓	✗
AppStore	FreeStyle LibreLink – BR*	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
AppStore	OneTouch Reveal**	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗
AppStore	Glicose Companheiro	✓	✗	±	✗	✗	✗	✗	✗	✗	✗	✗	✗
PlayStore	BD Diabetes Care	✓	✗	±	✗	✓	✓	✗	✗	✗	✓	✗	✗
PlayStore	DiabetesM	✓	±	±	✗	✗	✓	✗	✗	✗	✗	✗	✗
PlayStore	Diabetes controle – diário de diabetes	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
PlayStore	Diabete – Diário Glucose	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
PlayStore	forDiabetes: aplicativo de diabetes	✓	±	±	✗	✗	✓	✗	✗	✗	✓	✗	✗
PlayStore	SocialDiabetes. Take control	✓	✓	✓	✗	✗	✓	✗	✗	✗	✓	✗	✗
PlayStore	Índice e Carga Glicêmica alimentos para diabetes	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
PlayStore	Blood Glucose Tracker	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
PlayStore	Diabetes Control APP	✓	±	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
	DiaVision	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Fonte: [Sobral \(2021\)](#).

4.3 Visão e Análise

Nesta seção são descritas as necessidades e características planejadas para o produto de *software* desenvolvido, identificadas a partir de reuniões com a dona do produto (PO, do inglês *product owner*), esta que identificou a problemática abordada neste trabalho e realizou o levantamento de funcionalidades e problemas das soluções já existentes no mercado em [Sobral \(2021\)](#).

4.3.1 Descrição do Problema

O [Quadro 7](#) apresenta, de forma resumida, o problema, seus impactos e a proposta de solução com seu diferencial.

Quadro 7 – Descrição do problema.

Problema	Dificuldade de acesso à informações de autocuidado com relação ao DM por deficientes visuais.
Afeta	Independência e qualidade de vida de diabéticos com DV.
Impacta	No autocuidado e, consequentemente, no controle do DM.
Solução	Desenvolvimento de aplicação móvel com conteúdos e funcionalidades que auxiliem diabéticos no gerenciamento do autocuidado com o DM.
Diferencial	Acessibilidade ao deficiente visual.

Fonte: Autor e [Sobral \(2021\)](#).

4.3.2 Riscos e Impedimentos

Os seguintes riscos e possíveis impedimentos com relação ao produto foram identificados:

- Não adesão por parte do público alvo;
- Dificuldades no manuseio do *smartphone* pelo público alvo;
- Dificuldade de localizar os possíveis participantes da pesquisa;
- Utilização incorreta do aplicativo ou não assimilação das informações adquiridas;
- Afastamento do paciente da assistência continuada na rede primária;
- Constrangimento do usuário por falta de entendimento das funcionalidades.

4.4 Requisitos

Antes de iniciar o desenvolvimento de qualquer tarefa técnica de engenharia de *software*, é interessante que seja criado um conjunto de requisitos. Isso porque as tarefas de levantamento de requisitos levam a um entendimento dos impactos da solução, necessidades do cliente e como os usuários finais vão interagir com o *software*, diminuindo as chances de erros por má interpretação das solicitações dos clientes ([PRESSMAN; MAXIM, 2014](#)).

Esses requisitos costumam ser classificados como funcionais, não-funcionais e inversos ([SOMMERVILLE et al., 2007](#)). E serão apresentados nesta seção, iniciando pelas estórias de usuários, parte inicial do processo de elicitação dos requisitos, passando pelos diagramas de casos de uso e classes, e finalizando com o protótipo de telas.

4.4.1 Estórias de usuários

Estórias de usuários são muito utilizadas em metodologias ágeis e descrevem um cenário geral no qual é possível visualizar quais ações são possíveis, os atores envolvidos e quais os valores dessas ações, servindo como lembrete de possíveis requisitos que precisam ser melhor detalhados com o cliente (NAWROCKI et al., 2014).

Assim, as estórias de usuários identificadas são listadas na [Quadro 8](#).

Quadro 8 – Relação de estórias de usuários.

Eu, enquanto	Quero	Para
Paciente	Encontrar o <i>app</i> nas lojas virtuais	Baixar o <i>app</i> no meu celular.
Paciente	Realizar cadastro no aplicativo	Ter acesso às funcionalidades do <i>app</i> .
Paciente	Realizar login de forma prática	Para acessar as funcionalidades do <i>app</i> .
Paciente	Poder alterar minha senha	Poder alterá-la e recuperar acesso ao <i>app</i> .
Paciente	Registrar informações das refeições	Acompanhar a quantidade de calorias consumidas por refeição.
Paciente	Ter acesso a aplicativos acessíveis para deficientes visuais	Ajudar a realizar atividades do dia a dia.
Paciente	Sugerir aplicativos acessíveis para deficientes visuais	Compartilhar aplicativos que possam ajudar outros usuários com DV.
Paciente	Registrar práticas de atividade física	Acompanhar a evolução da rotina de atividade física.
Paciente	Ter acesso à dicas de autocuidado	Melhorar a qualidade de vida e prevenir complicações do DM.
Paciente	Filtrar as dicas por categorias	Facilitar a busca das dicas sobre assuntos específicos.
Paciente	Consultar locais para acesso à serviços de saúde	Facilitar o acesso e contato com as principais clínicas, hospitais e consultórios da cidade.
Paciente	Registrar glicemia	Acompanhamento dos valores de glicemia e ser alertado quando estiver fora do limite.
Paciente	Registrar medicações que faço uso	Ter uma lista atualizada com todas as informações das medicações e ser alertado dos horários de uso das medicações.
Paciente	Realizar avaliação dos pés	Acompanhar a evolução dos pés e detectar quando surgir alterações.
Paciente	Registrar diurese diária	Acompanhar quando surgir alterações.
Paciente	Ter acesso a relatórios dos dados registrados	Visualizar e compartilhar esses dados registrados.
Paciente	Ter acesso aos dados pessoais	Editiar ou acrescentar dados pessoais durante o uso do aplicativo.
Paciente	Configurar notificações	Definir horários e quais ativar ou desativar.
Paciente	Configurar preferências	Personalizar os limites da glicemia.
Paciente	Realizar <i>logout</i>	Para desvincular minha conta do <i>app</i> .
Administrador do sistema	Adicionar dicas de autocuidado para os pacientes	Fornecer informações acerca de cuidados com a saúde.
Administrador do sistema	Cadastrar centros de saúde no sistema	Que o paciente possa conhecer os centros de saúde que atendem suas demandas.
Administrador do sistema	Cadastrar sugestões de aplicativos acessíveis no sistema	Que o paciente possa conhecer outros <i>apps</i> acessíveis que possam ajudá-lo no cotidiano.
Administrador do sistema	Aprovar/recusar as sugestões de centros de saúde e aplicativos	Assegurar credibilidade ao aplicativo.

Fonte: Autor e [Sobral \(2021\)](#).

4.4.2 Requisitos Funcionais

O [Quadro 9](#) mostra os requisitos funcionais da aplicação, estes que referem-se, principalmente, às funções e comportamentos do sistema.

Quadro 9 – Requisitos Funcionais da aplicação.

Código	Atores	Requisito	Prioridade	Descrição
RF01	Paciente	Manter paciente	Essencial	O paciente poderá gerenciar seus dados na aplicação.
RF02	Paciente	Resetar senha	Essencial	O paciente poderá solicitar a alteração de senha para recuperar acesso.
RF03	Paciente	Autenticação	Essencial	Será necessária autenticação com e-mail e senha para ter acesso às funcionalidades do <i>app</i> .
RF04	Administrador	Manter Dicas de Autocuidado	Essencial	O administrador do sistema poderá gerenciar as dicas de autocuidado no sistema.
RF05	Administrador	Manter Centros de Saúde	Desejável	O administrador do sistema poderá gerenciar os centros de saúde no sistema.
RF06	Administrador	Manter Apps de Visão	Importante	O administrador do sistema poderá gerenciar sugestões de aplicativos acessíveis à PDV.
RF07	Paciente	Manter Registros de Diurese	Importante	O paciente poderá registrar diurese e gerenciar esses registros.
RF08	Paciente	Manter Registros de Glicemia	Essencial	O paciente poderá registrar níveis de glicemia e gerenciar esses registros.
RF09	Paciente	Manter Registros de Medicção	Essencial	O paciente poderá registrar suas medicações e gerenciar esses registros.
RF10	Paciente	Manter Registros de Exercícios	Importante	O paciente poderá registrar atividades físicas realizadas e gerenciar esses registros.
RF11	Paciente	Manter Avaliações dos Pés	Essencial	O paciente poderá registrar avaliações do estado dos pés e gerenciar esses registros.
RF12	Paciente	Sugerir Apps de Visão	Importante	O paciente poderá sugerir de aplicativos acessíveis à PDV para avaliação do administrador.
RF13	Paciente	Manter Registros de Alimentação	Essencial	O paciente poderá registrar os alimentos que consumiu por refeição e gerenciar esses registros.
RF14	Paciente	Consultar Dicas de Autocuidado	Essencial	O paciente poderá consultar as dicas de autocuidado disponíveis no sistema.
RF15	Paciente	Consultar Centros de Saúde	Desejável	O paciente poderá consultar os centros de saúde disponíveis no sistema.
RF16	Paciente	Consultar Apps de Visão	Importante	O paciente poderá consultar sugestões de aplicativos acessíveis à PDV disponíveis no sistema.
RF17	Paciente	Sugerir Centros de Saúde	Desejável	O paciente poderá sugerir centros de saúde para avaliação do administrador.
RF18	Paciente	Configurar Notificações	Essencial	O paciente poderá configurar quais notificações deseja receber e os horários.
RF19	Paciente	Envio de Notificações	Essencial	Notificações deverão ser enviadas de acordo com as configurações definidas pelo paciente.

Fonte: Autor e [Sobral \(2021\)](#).

4.4.3 Requisitos Não-Funcionais

Requisitos não-funcionais em sistemas podem ser descritos como atributos de qualidade, desempenho, segurança ou gerais, estes que podem ser identificados a partir das necessidades do cliente mesmo que não tenham sido falados explicitamente (PRESSMAN; MAXIM, 2014).

Assim, na [Quadro 10](#), são listados esses requisitos identificados juntamente com o tipo e a prioridade de cada um deles para este projeto.

Quadro 10 – Requisitos Não-Funcionais da aplicação.

Código	Tipo	Requisito	Prioridade	Descrição
RNF01	Usabilidade	Acessibilidade	Essencial	Implementar as técnicas de acessibilidade para solucionar os principais problemas relacionados.
RNF02	Usabilidade	Simplicidade	Essencial	<i>Interface</i> simples e intuitiva, mantendo apenas as informações necessárias na tela.
RNF03	Usabilidade	Buscas Ágeis	Desejável	Facilitar buscas por meio de <i>auto complete</i> .
RNF04	Segurança	Compartilhamento de Informações	Essencial	Somente o próprio usuário terá acesso e poderá compartilhar suas informações.
RNF05	Tecnologia	Aplicação multiplataforma	Desejável	Deve-se utilizar de ferramentas que possibilitem a construção da aplicação para Android e iOS.

Fonte: Autor e [Sobral \(2021\)](#).

4.4.4 Requisitos Inversos

Os requisitos listados na [Quadro 11](#), chamados inversos, referem-se às restrições, condições que não devem ocorrer no sistema.

Quadro 11 – Requisitos Inversos da aplicação.

Código	Prioridade	Descrição
RI01	Essencial	Um usuário não deve poder acessar recursos de outros.
RI02	Essencial	Os usuários não devem ser notificados se não estiverem deslogados.
RI03	Essencial	O administrador do sistema não deve ter acesso aos dados dos usuários.

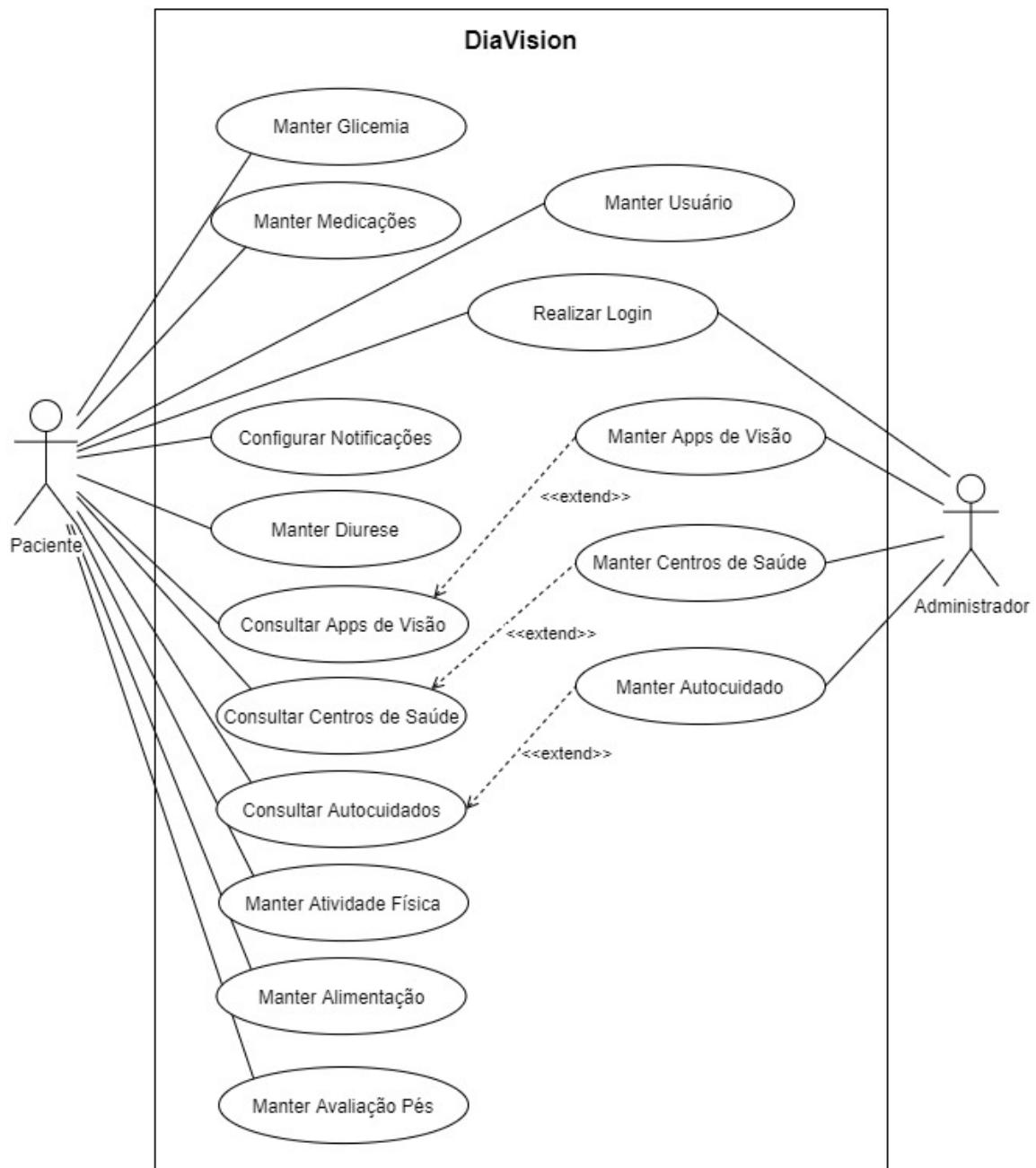
Fonte: Autor e [Sobral \(2021\)](#).

4.4.5 Casos de Uso

De acordo com Pressman e Maxim (2014), um caso de uso é caracterizado como um “contrato de comportamento” que define como um ator utiliza um sistema para alcançar algum objetivo e descreve um cenário de uso de forma simples do ponto de vista desse ator.

A partir dos requisitos e estórias de usuários identificados, o diagrama de casos de usos da Figura 6 foi elaborado.

Figura 6 – Diagrama de casos de uso.



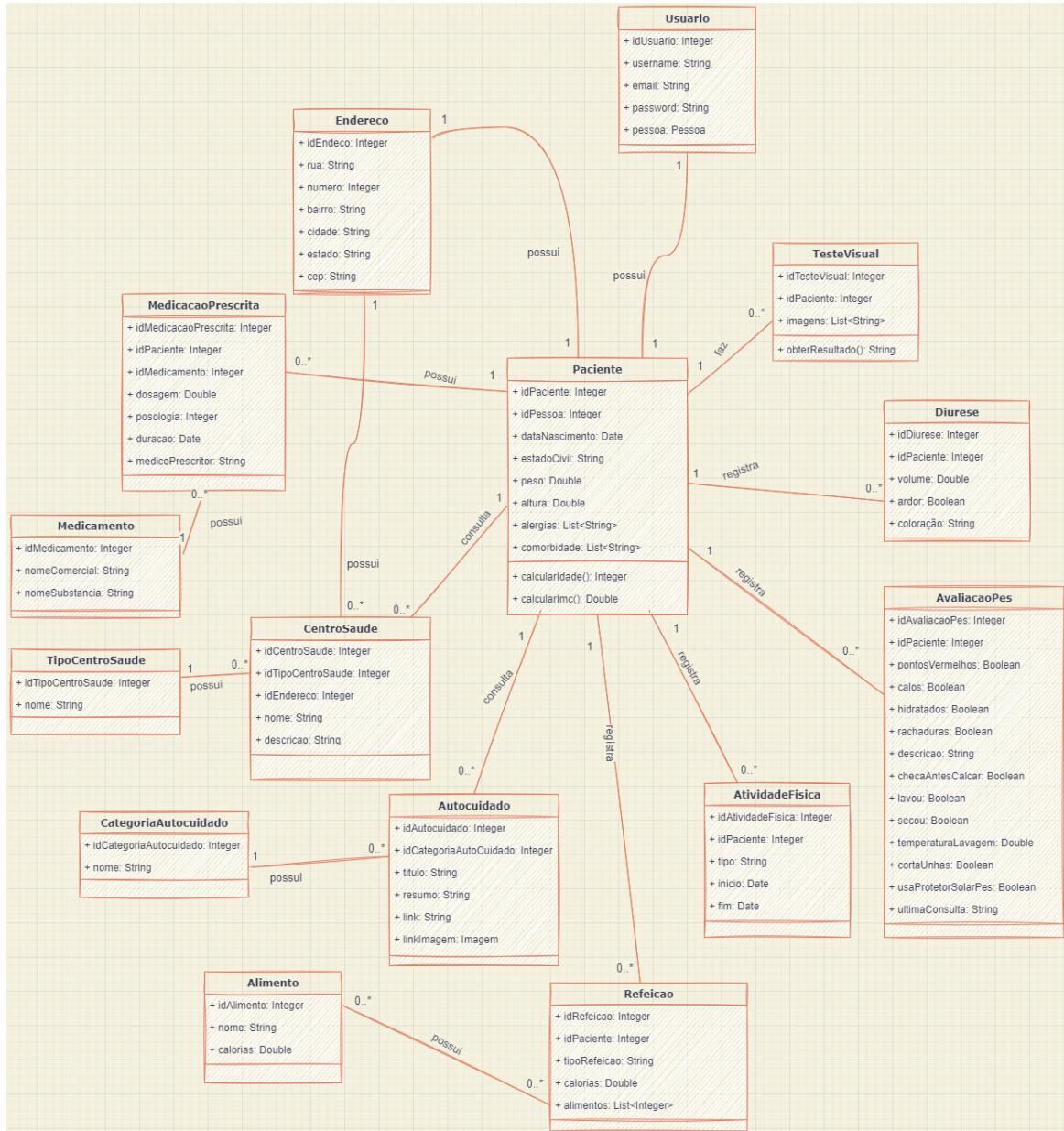
Fonte: Autor.

4.4.6 Diagrama de Classes

O Diagrama de Classes é um dos principais artefatos definidos pelo UML e tem como objetivo descrever a estrutura e as relações de um sistema por meio de classes que representam abstrações de objetos do mundo real (GUEDES, 2018).

Diante disso, a [Figura 7](#) ilustra as classes de domínio e as relações identificadas durante a análise e planejamento do sistema.

Figura 7 – Diagrama de Classes de Domínio.

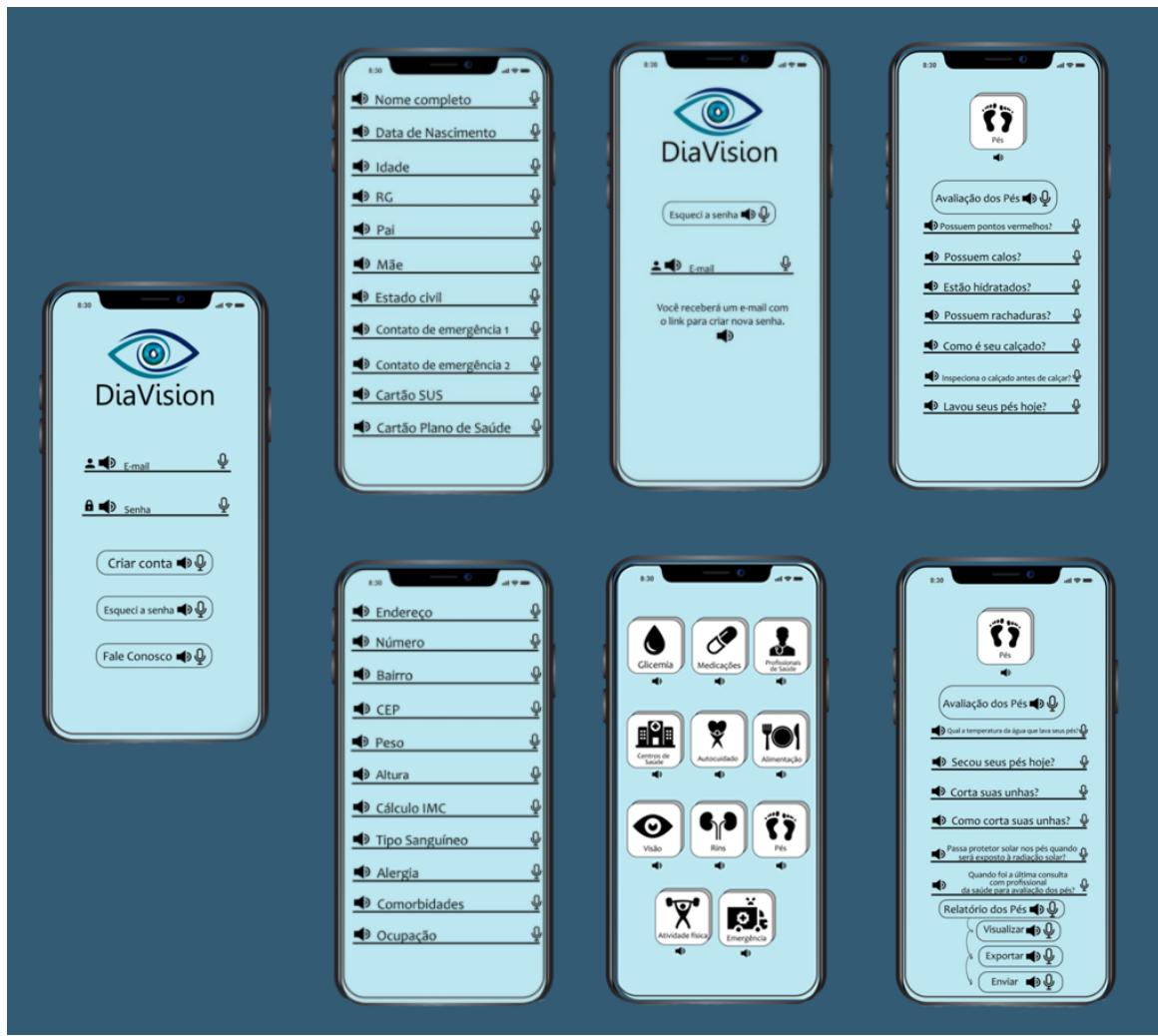


Fonte: Autor.

4.4.7 Protótipo de Telas

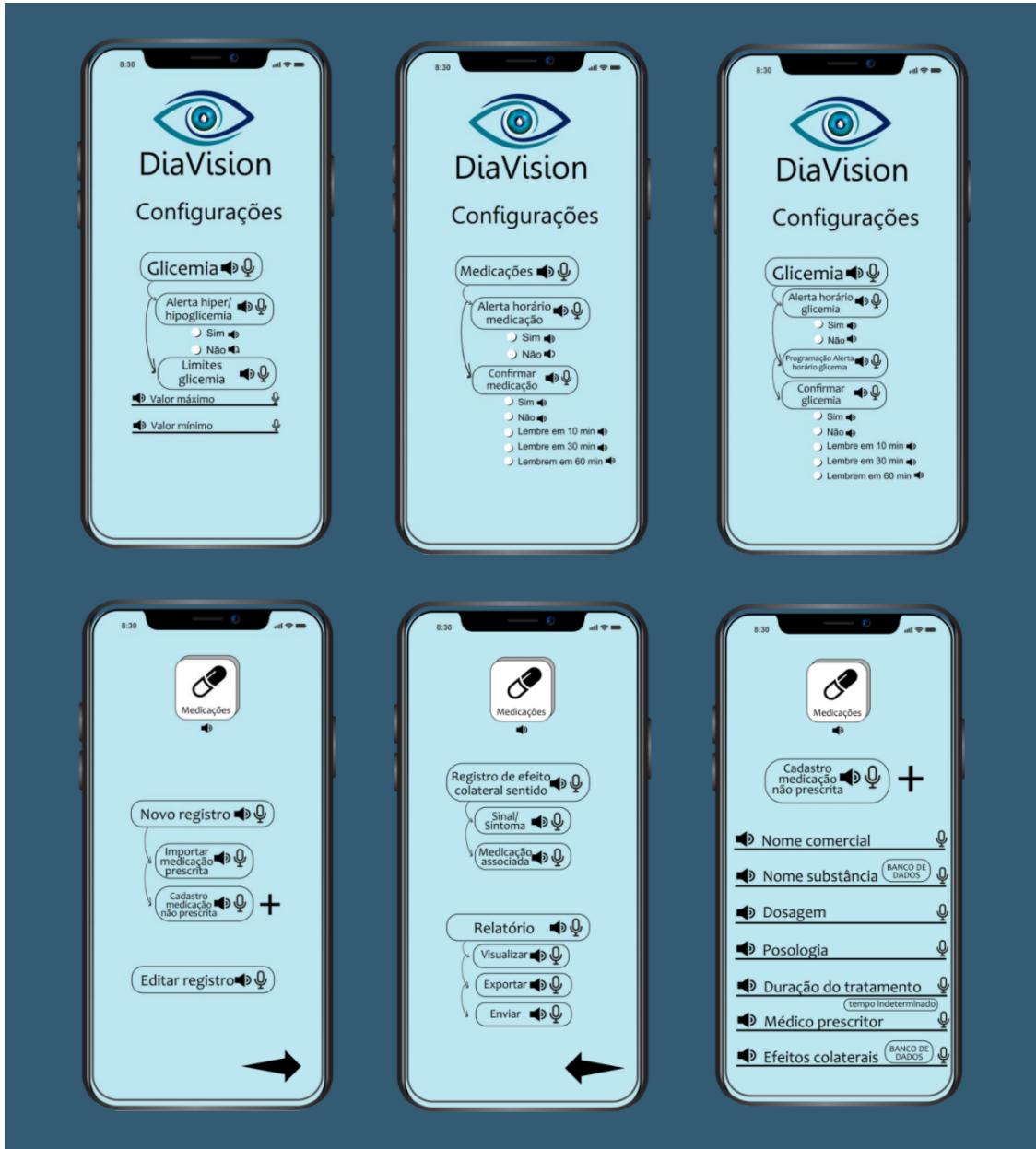
Um protótipo de telas foi elaborado por Sobral (2021) e teve como objetivo melhorar o entendimento dos requisitos e funcionalidades esperados do aplicativo desenvolvido. Assim, nas Figura 8 e Figura 9 são apresentadas as telas do protótipo.

Figura 8 – Telas iniciais do protótipo.



Fonte: Sobral (2021).

Figura 9 – Demais telas do protótipo.



Fonte: Sobral (2021).

4.5 Considerações Finais

Este capítulo apresentou um resumo de todos os aspectos analisados durante o planejamento do sistema. Descrevendo o problema a ser resolvido e as motivações, e o projeto, seus riscos, restrições e requisitos. Assim como, passou a visão do sistema por meio de diagramas e protótipos de tela.

5

Desenvolvimento do DiaVision

Neste capítulo são apresentadas as tecnologias utilizadas, as soluções adotadas para os problemas de acessibilidade identificados no MSL e os resultados alcançados ao final do processo desenvolvimento.

5.1 Tecnologias Utilizadas

Esta seção lista as principais tecnologias e recursos utilizados no desenvolvimento do sistema DiaVision.

5.1.1 Visual Studio Code

O Visual Studio Code (VSCode) é um editor de código *open source* desenvolvido pela Microsoft para Windows, Linux e MacOS. Ele inclui suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe, IntelliSense - termo para um conjunto de funcionalidades de edição de código, tais como completar código e informações de parâmetros - e refatoração de código.

O VSCode também possibilita a instalação de extensões que adicionam funcionalidades ao editor, assim, aumentando a produtividade do desenvolvedor. Por conta dessas características, esse foi o editor utilizado neste projeto tanto para desenvolver o sistema DiaVision quanto para a parte escrita deste trabalho.

As extensões utilizadas para desenvolvimento da aplicação foram: Flutter e Dart, que dão suporte à criação de aplicações com o SDK do Flutter e fluttermobx, que oferece atalhos para adicionar componentes de gerenciamento do estado da aplicação. Já Para a parte escrita, foi utilizado o conjunto de extensões: bibtexLanguage, Code Spell Checker, LaTeX e LaTeX Workshop.

5.1.2 Github

O Github é uma plataforma de desenvolvimento e gestão de código-fonte, baseada no Git, que permite aos usuários compartilhar seus projetos e arquivos com controle de versionamento de código. Com isso é possível realizar *commits*, sendo cada *commit* um ponto de alteração no histórico do projeto, criar *branches*, ramificações que possibilitam trabalhar em diferentes funcionalidades ao mesmo tempo e realizar *merges*, que é processo de unir ramificações.

O versionamento de código é uma das funcionalidades mais importantes do Git, pois, manter o histórico de alterações facilita a investigação e resolução de problemas e, em caso de falhas em uma versão do projeto, é possível reverter para uma versão anterior.

O Github é gratuito e permite a criação de projetos públicos e privados, porém, os repositórios privados possuem limitações nas funcionalidades da plataforma. Ambos os repositórios, do desenvolvimento do DiaVision¹ e da parte escrita com LaTeX² subiram para o GitHub, inicialmente de forma privada e disponibilizados de forma pública após a finalização deste trabalho.

5.1.3 Ambiente de Desenvolvimento

A máquina principal utilizada para desenvolvimento do sistema DiaVision foi um computador de mesa (do inglês *Desktop*), com o SO Windows 11. Porém, trata-se do desenvolvimento de uma aplicação multiplataforma e não é possível testar a aplicação para o sistema operacional iOS, da Apple, em um computador que não seja esteja rodando o macOS. Assim, para validação do aplicativo, também foi utilizado um MacBook.

O Quadro 12 mostra as especificações das máquinas utilizadas no desenvolvimento do sistema.

Quadro 12 – Máquinas de Desenvolvimento.

	<i>Desktop</i>	<i>MacBook</i>
Sistema Operacional	Windows 11 Pro 22H2	macOS Big Sur 11.3.1
Disco Rígido	SDD NVMe 512GB e HD 1TB	256GB SDD
Memória RAM	32GB 3000MHz DDR4	8GB 1600MHz DDR3
Processador	AMD Ryzen 5 5600X 6-Core	Intel Core i5 Dual-Core

Fonte: Autor.

¹ <https://github.com/jnthonklnv/dia_vision>

² <https://github.com/jnthonklnv/tcc_dia_vision>

5.1.4 Dispositivos de Teste

Os testes e validações da aplicação, durante o processo de desenvolvimento, foram realizados em dispositivos físicos rodando o SO Android, porém, para o iOS os testes foram realizados por meio do iOS Simulator, disponível apenas em computadores com o SO macOS.

Assim, o [Quadro 13](#) apresenta as especificações dos *smartphones* utilizados durante o desenvolvimento do aplicativo.

Quadro 13 – *Smartphones* utilizados no Desenvolvimento.

	Galaxy S20	Redmi Note 8
Marca	Samsung	Xiaomi
Sistema Operacional	Android 12	Android 9.0
Armazenamento Interno	128GB	32GB
Memória RAM	8GB	3GB
Chipset	SAMSUNG Exynos 990	Snapdragon 665 Qualcomm

Fonte: Autor.

5.1.5 Back4app

O Back4app é uma plataforma gratuita (no plano básico) que segue o modelo de serviço em nuvem *Backend* como Serviço (BaaS, do inglês *Backend as a Service*) para desenvolvimento de *backend* e gerenciamento de infraestrutura com eficiência e facilidade. Com a plataforma é possível criar estruturas de dados e serviços de forma simples, sem a necessidade de codificação, possibilitando também a adição de código por meio de *Cloud Functions*³ para criar validações ou funcionalidades mais complexas.

A plataforma é baseada no ParsePlatform, um *framework* de código aberto que fornece um conjunto de SDKs que visam a criação rápida de *backends* com armazenamentos de objetos e arquivos, autenticação de usuário, *push notifications*, *dashboard* para gerenciamento e integração com aplicações móveis e *web*.

O Back4app também disponibiliza a criação, customização e hospedagem de uma aplicação *web* para administração do sistema. Sendo possível definir as classes e operações que estarão disponíveis para gerenciamento pelo Administrador do Sistema (ADS).

Devido a simplicidade que o Back4app oferece para criação, gerenciamento e hospedagem de *backend*, comparado à aplicações convencionais desenvolvidas por meio da codificação, e ao foco deste trabalho, que é o desenvolvimento da aplicação móvel com o foco na usabilidade, esse foi o serviço escolhido para o desenvolvimento do sistema.

³ Com Funções em Nuvem é possível escrever funções simples que são disparadas a partir de eventos emitidos pela infraestrutura e serviços de nuvem.

O plano utilizado para o serviço foi o gratuito, que possui limite de 10 mil requisições por mês, 250MB para armazenamento de dados, 1GB para transferência de dados, 1GB para armazenamento de arquivos e 1 *Cloud Code Job*⁴.

5.1.6 Flutter

O *framework* escolhido para desenvolvimento do aplicativo móvel foi o Flutter, apresentado no Capítulo 2, devido a sua facilidade no desenvolvimento móvel multiplataforma, possibilitando a geração de aplicações para Android e iOS a partir de uma base de código única, mantendo desempenho próximo ao de aplicações nativas.

5.2 Implementação

Esta seção detalha as soluções de implementação adotadas, tais como a arquitetura do sistema e as soluções para os problemas de acessibilidade.

5.2.1 Arquitetura

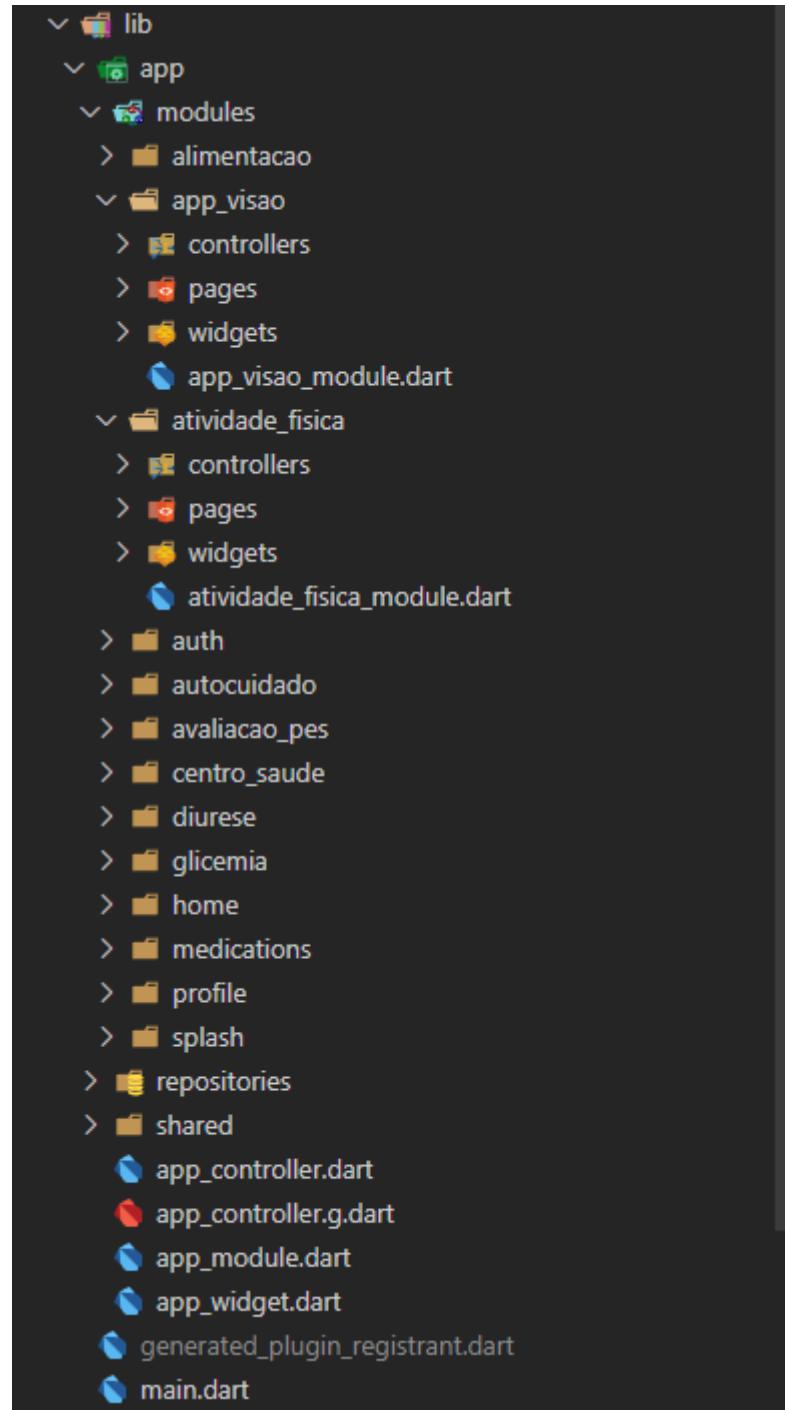
A estrutura de pastas do projeto está organizada em módulos, os quais não conhecem uns aos outros, apenas as classes relativas ao próprio módulo estão dentro de cada pasta. A exceção é para os *repositories* (repositórios) que estão numa camada separada e contém os modelos e as conexões com a fonte de dados. Essa camada não tem conhecimento a respeito das demais camadas da aplicação.

Dentro de cada módulo estão as outras duas camadas, os *controllers* (controladores) e a *view* (interface visual). Os *controllers* fazem a mediação entre a *view* e os *repositories*, eles controlam o estado da aplicação, devendo atualizar a *view* quando receber atualização dos *repositories* e chamar os *repositories* quando determinados eventos são disparados na *view*.

A Figura 10 mostra como está organizada essa estrutura de pastas, sendo que a *view* está dividida em *pages*, que são as telas e *widgets*, que são os componentes que fazem parte das telas.

⁴ *Cloud Jobs* são utilizados para execução de funções de longa duração, das quais não é necessário aguardar a resposta, no servidor. Em outras soluções é possível executar essas funções periodicamente, porém, essa funcionalidade ainda não está disponível no Back4app e elas precisam ser disparadas por meio da API Rest.

Figura 10 – Estrutura de Pastas do Projeto.



Fonte: Autor.

Ainda na Figura 10 percebe-se que dentro de cada módulo existe uma classe geral, é nela onde são definidas as rotas, injeções de dependências e *singletons*⁵ de cada módulo, como mostra a Figura 11.

⁵ Singleton é um padrão de projeto que garante a existência de apenas uma instância de uma classe.

Figura 11 – Classe Principal do Módulo.

```

11  class AppVisaoModule extends Module {
12    @override
13    final List<Bind> binds = [
14      Bind((i) => AppVisaoRepository()),
15      Bind((i) => AppVisaoController(i())),
16      Bind((i) => AppVisaoRegisterController(i())),
17    ];
18
19    @override
20    final List<ModularRoute> routes = [
21      ChildRoute('/', child: (_, args) => AppsVisaoPage()),
22      ChildRoute('/$registerStr/',
23        child: (_, args) => AppVisaoRegisterPage(args.data)), // ChildRoute
24    ];
25  }

```

Fonte: Autor.

Os *singletons* dos repositórios são injetados nos construtores dos controladores como mostrado nas linhas 15 e 16 da [Figura 11](#) e recuperados pelos controladores conforme mostrado na [Figura 12](#).

Figura 12 – Recuperar Injeção do *Repository* no *Controller*.

```

12  abstract class _AppVisaoControllerBase with Store {
13    final IAppVisaoRepository _avaliacaoPesRepository;
14
15    _AppVisaoControllerBase(this._avaliacaoPesRepository);

```

Fonte: Autor.

Já os *singletons* dos controladores, por sua vez, são recuperados nas telas, como nota-se na linha 24 da [Figura 13](#).

Figura 13 – Recuperar *Controller* na *Page*.

```

16  <> class AppsVisaoPage extends StatefulWidget with ScaffoldUtils {
17    AppsVisaoPage({Key? key}) : super(key: key);
18
19    @override
20    _AppsVisaoPageState createState() => _AppsVisaoPageState();
21  }
22
23  You, 4 weeks ago | 2 authors (Jonathan Kelvin and others)
24  <> class _AppsVisaoPageState
25  <>   extends ModularState<AppsVisaoPage, AppVisaoController> {

```

Fonte: Autor.

5.2.2 Backend

O Back4app foi escolhido para o *backend* do sistema e com ele foi possível definir a estrutura de dados, que pode ser vista na [Figura 14](#), graficamente por meio de sua interface de usuário.

Figura 14 – Esquema do Banco de Dados.



The screenshot shows a database browser interface with a blue header labeled "Database". Below the header, there is a "Browser" section containing a list of database classes and their corresponding counts. A "Create a class" button is located in the top right corner of the browser section. The list of classes is as follows:

Class	Count
Role	1
Session	30
User	19
Alimentacao	14
Alimento	40
AppVisao	16
AtividadeFisica	23
AutoCuidado	82
AvaliacaoPes	6
B4aCustomField	29
B4aMenuItem	24
B4aSetting	4
CentroSaude	17
Diurese	8
Endereco	16
Glicemia	94
Image	2
MedicacaoPrescrita	17
Medicamento	4.73k
Paciente	10

Fonte: Autor.

Embora a plataforma utilize um banco de dados NoSQL - banco de dados não relacional -, o esquema de dados é definido de forma relacional, onde classes representam tabelas e os relacionamentos são definidos com a utilização de ponteiros que apontam para objetos de outras classes. Assim, unindo o maior desempenho possibilitado por uma estrutura NoSQL com uma visão relacional do esquema de dados em alto nível.

5.2.2.1 Classes do Banco de Dados

A [Figura 15](#) apresenta a classe “Alimentacao” como exemplo, cada classe no banco, por padrão, possui os atributos objectId, que é utilizado para referenciar os objetos em outras classes por meio de ponteiros, createdAt, que é a data de criação do objeto, updatedAt, que é a data da última atualização do objeto, e ACL⁶, que define qual usuário tem permissão de acesso a esse objeto e quais são essas permissões (escrita, leitura ou ambos).

Com isso, visando a segurança dos dados dos usuários, todas as classes que registram dados dos pessoais possuem restrição de ACL, onde somente o próprio usuário possui permissão tanto de leitura quanto de escrita.

Figura 15 – Exemplo de Classe no Banco.

	objectId	createdAt	ACL	tipo	paciente	calorias
<input type="checkbox"/>	OWLEl5hJoG	13 Apr 2022 at 00:28...	TDJWTRG6RH	Café da manhã	h3UGDzTzeE	38
<input type="checkbox"/>	8NjpaKkK2J	15 Dec 2021 at 22:27...	ZC4C4blz6f	Lanche	4xY08RD10E	136
<input type="checkbox"/>	SeWb3tZEXS	11 Oct 2021 at 17:44...	Iyc40RjlvV	Almoço	LDqAIdmq3L	141
<input type="checkbox"/>	aYU1hfEAmw	3 Sept 2021 at 17:56...	ZC4C4blz6f	Almoço	4xY08RD10E	79927
<input type="checkbox"/>	syBiSgvlip	3 Sept 2021 at 17:53...	ZC4C4blz6f	Café da manhã	4xY08RD10E	8212
<input type="checkbox"/>	BcGfGXyZ2Q	2 Aug 2021 at 12:17...	ZC4C4blz6f	Café da manhã	4xY08RD10E	24405
<input type="checkbox"/>	R9FaIyWCy	29 July 2021 at 17:2...	ZC4C4blz6f	Almoço	4xY08RD10E	65445.5
<input type="checkbox"/>	mP3IOATTou	26 July 2021 at 22:5...	ZC4C4blz6f	Almoço	4xY08RD10E	61481.25
<input type="checkbox"/>	4EihW62WSl	26 July 2021 at 22:5...	ZC4C4blz6f	Café da manhã	4xY08RD10E	152
<input type="checkbox"/>	OCXoCxgZBe	22 July 2021 at 23:4...	vRubdf8zP1	Almoço	FeAAllXGAB	189
<input type="checkbox"/>	oUMHPwpWMY	22 July 2021 at 23:4...	vRubdf8zP1	Almoço	FeAAllXGAB	227
<input type="checkbox"/>	IQOGAygs8V	21 July 2021 at 17:4...	ZC4C4blz6f	Almoço	4xY08RD10E	300
<input type="checkbox"/>	TvixNLGShE	21 July 2021 at 17:2...	ZC4C4blz6f	Almoço	4xY08RD10E	10
<input type="checkbox"/>	Y6otEsFSVZ	17 July 2021 at 02:3...	vRubdf8zP1	Café da manhã	FeAAllXGAB	304

Fonte: Autor.

Ainda na [Figura 15](#), percebe-se que a classe “Alimentacao” possui um atributo chamado paciente do tipo *Pointer* (Ponteiro) que armazena o objectId que faz referência à classe Paciente como uma chave estrangeira em bancos de dados relacionais.

⁶ Lista de Controle de Acesso (ACL, do inglês *Access Control List*) são regras de acesso definidas para cada objeto na sua criação.

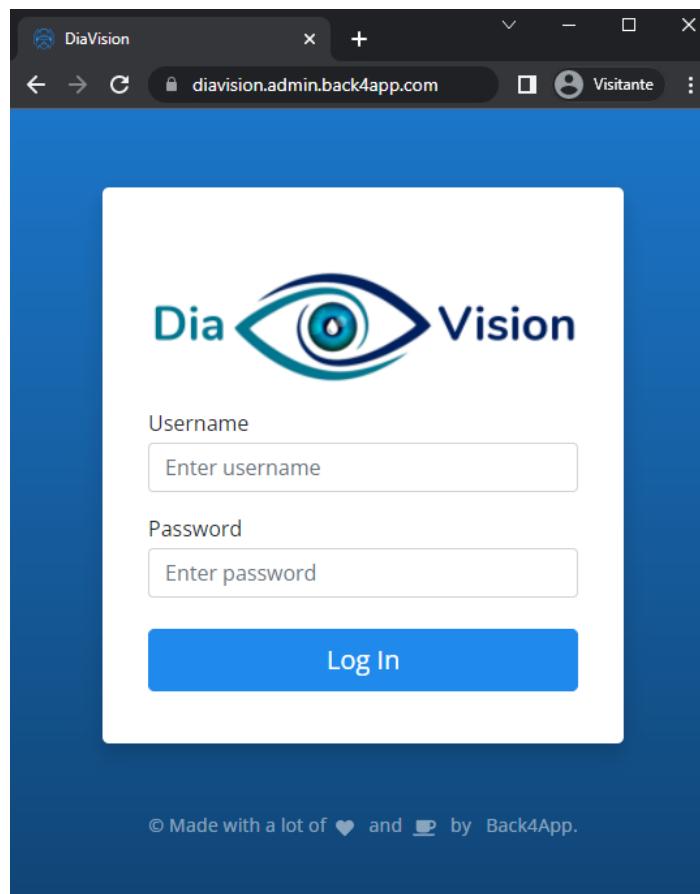
5.2.2.2 Dashboard do Administrador do Sistema

O Back4app também possibilitou a construção de um *dashboard web* para gerenciamento das informações que são disponibilizadas para os usuários no aplicativo, atendendo aos requisitos funcionais do Administrador do Sistema, mencionados no [Quadro 9](#).

A customização e disponibilização das funcionalidades nesse *dashboard* foi feita utilizando a própria interface da plataforma, onde foi possível definir as funcionalidades, campos e perfis acesso que estariam disponíveis. Ao habilitar o *dashboard* e realizar as customizações, são criadas novas classes no banco de dados, como pôde ser observado nas que iniciam com "B4a" na [Figura 15](#), que mantém salvas essas configurações.

A [Figura 16](#) apresenta a página de autenticação e a *URL*⁷ de acesso, sendo que as credenciais para autenticar-se são as mesmas utilizadas no aplicativo, porém, com um ACL específico de acesso ao *dashboard*.

Figura 16 – Tela de *Login* do *Dashboard*.



Fonte: Autor.

⁷ Localizador Uniforme de Recursos (URL, do inglês *Uniform Resource Locator*), se refere ao endereço que permite que um site ou recurso seja encontrado na rede.

Na [Figura 17](#) é possível visualizar a tela inicial do *dashboard* com as classes que estão disponíveis para gerenciamento no menu lateral esquerdo.

[Figura 17 – Dashboard](#) do Administrador do Sistema.

The screenshot shows the DiaVision system's administrative interface. On the left, there is a sidebar with the following navigation items: Apps de Visão, Autocuidados, Centros de Saúde, Endereço, Imagens, and Medicamentos. The main area is titled "Apps de Visão" and displays a table of applications. The table has columns for Id, descrição, linkAppleStore, and linkGooglePlay. There are 12 rows listed, each representing a different app. The first row is selected, showing details for "WeWALK". The table includes standard CRUD buttons (Edit, Delete, Add) and a filter/sort bar at the top. At the bottom, there is a pagination control showing page 1 of 2.

	Id	descrição	linkAppleStore	linkGooglePlay
<input checked="" type="checkbox"/>	KWBnjFH5ct	WeWALK é um aplicativo de navegação e explo...	https://apps.apple.com/br/app/wewalk-accessi...	https://play.google.com/store/apps/di...
<input type="checkbox"/>	ZIUv20YleN	O Sullivan + é um aplicativo de auxilio visual pa...	https://apps.apple.com/br/app/%EC%84%A4%EC%...	https://play.google.com/store/apps/di...
<input type="checkbox"/>	F4oU3vKdzK	Este aplicativo permite que deficientes visuais a...		https://play.google.com/store/apps/di...
<input type="checkbox"/>	9zbICodvB4	Este aplicativo permite com que usuários com ...		https://play.google.com/store/apps/di...
<input type="checkbox"/>	cGmbkDMXVF	Este joguinho é voltado para crianças com defici...	https://apps.apple.com/br/app/sherlockforms/i...	
<input type="checkbox"/>	WQMTe5mB0T	O Pay Voice é um aplicativo voltado para pesso...	https://apps.apple.com/br/app/pay-voice/id134...	https://play.google.com/store/apps/di...
<input type="checkbox"/>	634bGzXgqs	O Microsoft Soundscape colocará rótulos e indi...	https://apps.apple.com/br/app/microsoft-soun...	
<input type="checkbox"/>	G6otlAtmYb	O Veever é um aplicativo que utiliza-se de um si...	https://apps.apple.com/br/app/veever/id14888...	https://play.google.com/store/apps/di...
<input type="checkbox"/>	8ePXSBORvE	Com o CittaMobi você pode consultar a Previsâ...	https://apps.apple.com/br/app/acessibilidade-c...	https://play.google.com/store/apps/di...
<input type="checkbox"/>	fXFxwBFRGq	JoWalk visa ajudar pessoas cegas a detectar dif...	https://apps.apple.com/br/app/jowalk/id14937...	

Fonte: Autor.

Assim, a classe “Apps de Visão” trata-se de sugestões de aplicativos, de diversas categorias e finalidades, accessíveis à PDV, essas sugestões podem ser cadastradas pelo ADS ou sugeridas pelos usuários do DiaVision e aprovadas pelo ADS. A classe “Centros de Saúde” segue o mesmo modelo da anterior, porém, se refere à informações sobre centros de saúde, como nome, descrição, categoria, contatos e endereço.

A classe “Autocuidados” armazena dicas de autocuidado com relação à saúde para os usuários. Porém, diferentemente das duas classes anteriores, apenas o ADS pode cadastrar novas dicas, os usuários do aplicativo não podem criar sugestões.

Já a classe “Endereço”, armazena as informações de endereços, como nome, rua, CEP etc, e no momento é utilizada apenas pela classe “Centro de Saúde”. As imagens com o logotipo do DiaVision exibidas tanto na tela de autenticação quanto na tela principal do *dashboard* são arquivos cujas referências ficam na classe “Imagens”.

Por fim, a classe “Medicamentos” armazena informações sobre os nomes de medicamentos disponíveis no Brasil, especificamente o nome de substâncias utilizadas e o nome comercial de cada medicamento. Essas informações são utilizadas para facilitar o processo para o usuário informar os medicamentos que faz uso no aplicativo DiaVision.

A Figura 18 mostra o exemplo de edição de um objeto da classe “Apps de Visão”. O campo “verificado” é utilizado para aprovar as sugestões de *apps* feitas pelos usuários.

Figura 18 – Página para Edição de Objeto.

Edit a Apps de Visão Object	
descrição	WeWALK é um aplicativo de navegação e exploração
linkAppleStore	https://apps.apple.com/br/app/wewalk-accessible
linkGooglePlay	https://play.google.com/store/apps/details?id=tr.
título	WeWALK
verificado	<input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Undefined

[X Cancel](#) [✓ Save](#)

Fonte: Autor.

Na Figura 19, pode-se visualizar um exemplo de cadastro do objeto “Autocuidado”.

Figura 19 – Página para Cadastro de Objeto.

Add a Autocuidados Object	
categoria	Enter categoria
link	Enter link
linkImagem	Enter linkImagem
resumo	Enter resumo
título	Enter título

[X Cancel](#) [✓ Save](#)

Fonte: Autor.

5.2.3 Aplicativo

O aplicativo foi desenvolvido utilizando o *framework* de desenvolvimento Flutter, como mencionado no capítulo anterior, de acordo com os requisitos definidos no planejamento e as soluções de acessibilidade identificadas no MSL do Capítulo 3 e propostas para DiaVision.

5.2.3.1 Introdução

A página inicial do aplicativo pode ser vista na [Figura 20](#), é nela onde é apresentada uma das principais funcionalidades da aplicação, o *feedback* por voz dos componentes de interface.

Figura 20 – Introdução ao Aplicativo.



Fonte: Autor.

Ao pressionar e manter pressionados componentes como botões, textos, imagens etc,

uma descrição desses componentes é lida pelo assistente de voz do aparelho, sem a necessidade de ativar o modo acessibilidade que habilita os leitores de tela citados no Capítulo 2. Essa funcionalidade faz referência às soluções de acessibilidade TAM2 e TAM6, propostas para o *app* no final do Capítulo 3.

Ainda na tela inicial, da Figura 20, nota-se que existe um botão flutuante na parte inferior da tela, a criação dele foi inspirada no AssistiveTouch do iOS⁸. Já na Figura 21, o botão flutuante está em outro canto da tela após ser arrastado para lá pelo usuário.

Figura 21 – Botão de Acessibilidade.



Fonte: Autor.

Figura 22 – Menu de Acessibilidade.



Fonte: Autor.

Ao clicar no Botão de Acessibilidade, um menu com opções de personalização do aplicativo é aberto, como pode ser observado na Figura 22. As funcionalidades disponíveis, baseadas nas soluções de acessibilidade TAM1, TAM4, TAM8 e TAM11, são:

⁸ O AssistiveTouch é um botão flutuante, disponível em dispositivos Apple com iOS, que pode ser arrastado para qualquer borda da tela e que abre um menu no qual é possível ajustar o volume, bloquear a tela, usar gestos com vários dedos, reiniciar o dispositivo etc.

1. Aumentar e diminuir o tamanho da fonte de toda a aplicação;
2. Ocultar o botão de acessibilidade, sendo possível exibi-lo novamente na Tela de Configurações do app;
3. Acessar atalhos tanto para a Tela Inicial quanto para a de Configurações do app;
4. Alternar o tema do aplicativo entre claro e escuro;
5. Ativar e desativar o *feedback* por voz de toda a aplicação.

A [Figura 23](#) exibe o resultado do aumento no tamanho da fonte e da aplicação do tema escuro. Enquanto na [Figura 24](#) é exibida a segunda tela de introdução, com tema escuro e tamanho normal da fonte, explicando o funcionamento do botão de acessibilidade para o usuário.

Figura 23 – Demonstração de Personalização.



Fonte: Autor.

Figura 24 – Intro Botão de Acessibilidade.

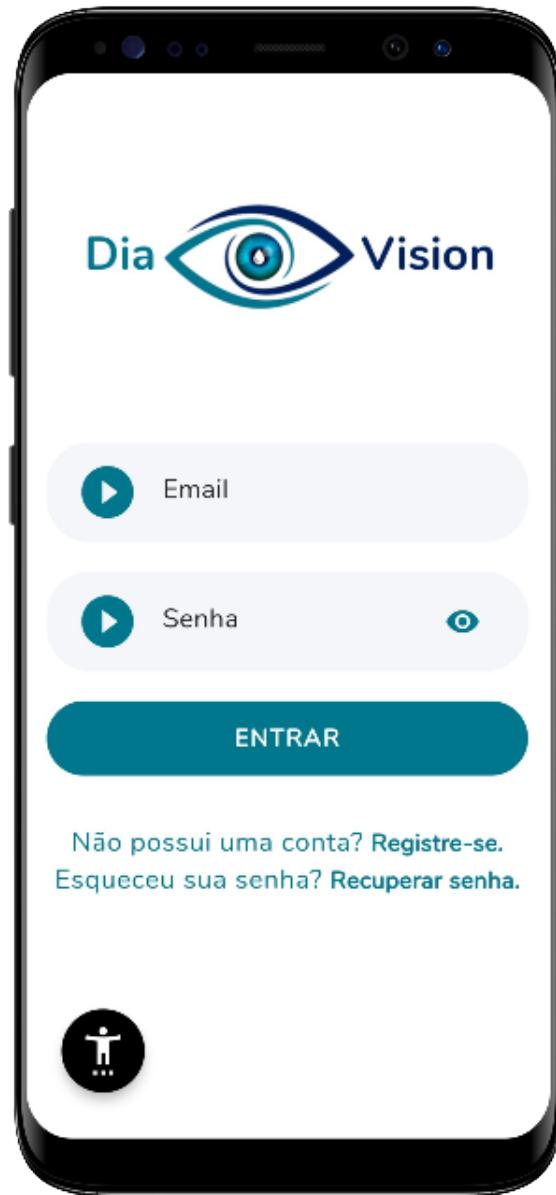


Fonte: Autor.

5.2.3.2 Autenticação

Ao clicar no botão “Pronto”, mostrado na [Figura 24](#), o aplicativo direciona para a tela de autenticação, apresentada na [Figura 25](#). A partir dessa tela o usuário pode realizar *login*, informando usuário e senha, para ser direcionado à tela principal do aplicativo.

Figura 25 – Tela de Autenticação.

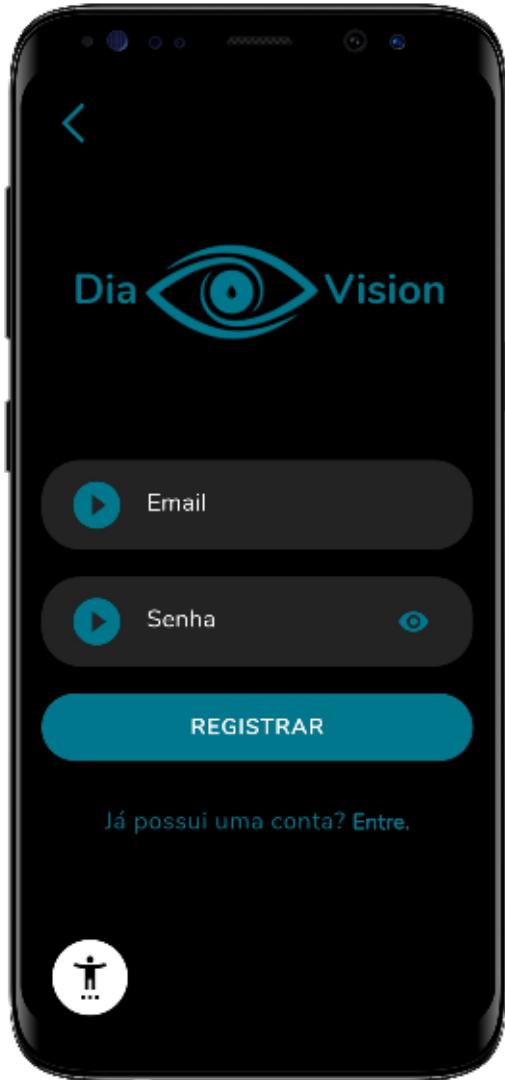


Fonte: Autor.

Além disso, a partir da tela de autenticação também é possível ir para tela de registro, caso ainda não tenha conta, ou para a tela de recuperação a senha, caso tenha-a esquecido, clicando nos textos correspondentes abaixo do botão “Entrar”, este que podê ser visto na [Figura 25](#).

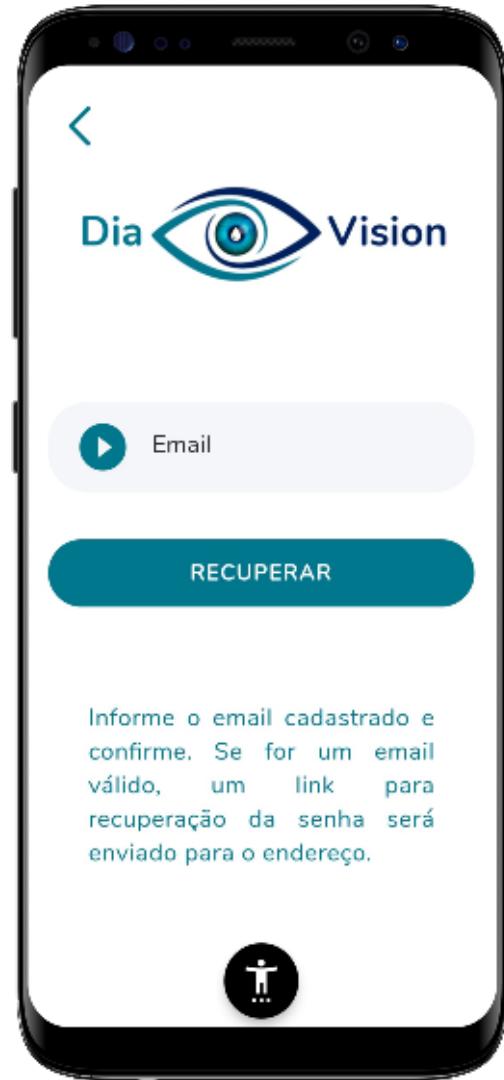
A tela de registro é apresentada na [Figura 26](#), nela é possível registrar um novo usuário, inicialmente, informando apenas e-mail e senha, mas podendo acrescentar mais informações na tela de Edição e Dados, após realizar a autenticação.

Figura 26 – Tela de Registro.



Fonte: Autor.

Figura 27 – Tela para Recuperação de Senha.



Fonte: Autor.

Já a [Figura 27](#), mostra a tela de recuperação de senha, bastando informar o e-mail da conta para receber um *link* no mesmo endereço para criação da nova senha.

5.2.3.3 Tela Principal

A tela principal, é apresentada na [Figura 28](#) e na [Figura 29](#), é a partir dela que pode-se acessar os módulos e outras funcionalidades do aplicativo. Na listagem há 9 módulos que podem ser acessados ao clicar em um dos quadrados e, assim como os demais componentes do *app*, ao mantê-los pressionados, suas descrições serão lidas pelo assistente de voz do aparelho.

Na primeira imagem o tema está escuro, enquanto que, na segunda, está claro e com o botão de acessibilidade oculto, mostrando que a personalização da aplicação é dinâmica e está acessível ao longo de todo o aplicativo.

Figura 28 – Tela Principal Tema Escuro.



Fonte: Autor.

Figura 29 – Restante da Tela Principal.



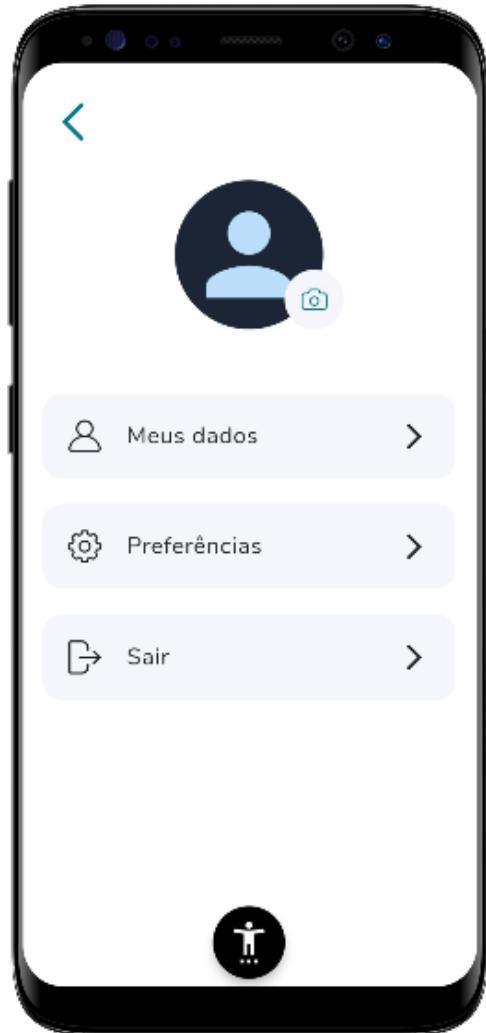
Fonte: Autor.

5.2.3.4 Configurações

O botão de engrenagem no topo da tela principal leva até a tela de configurações, mostrada na [Figura 30](#), onde é possível visualizar e alterar os dados pessoais, editar preferências e realizar logoff.

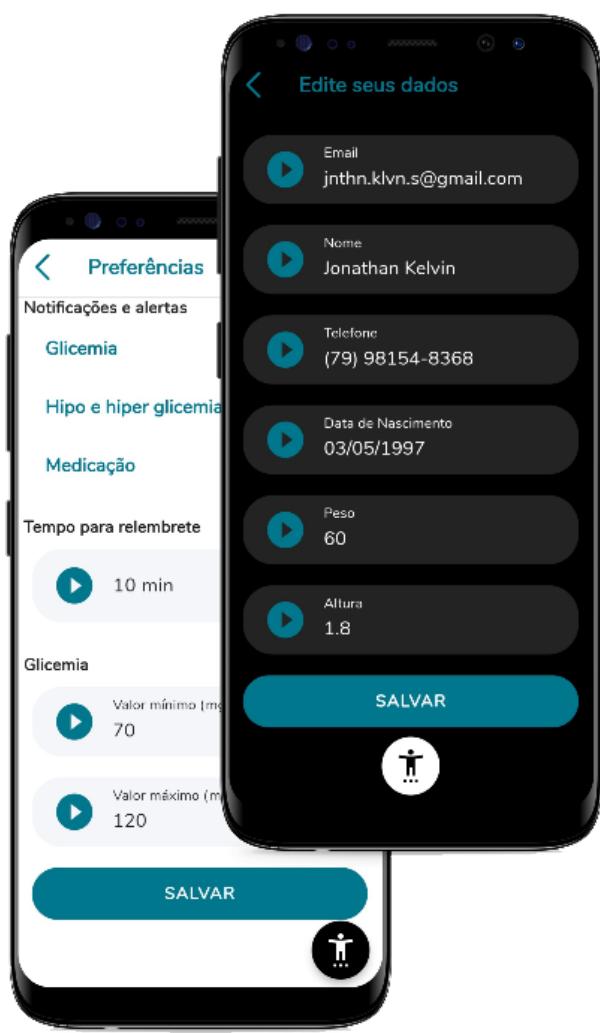
Na [Figura 31](#) são apresentadas as telas de edição dos dados do usuário e de preferências. Em “Meus Dados” o usuário pode alterar os dados de e-mail, nome, telefone, data nascimento, peso e altura.

Figura 30 – Tela de Configurações.



Fonte: Autor.

Figura 31 – Telas de Preferências e Dados.



Fonte: Autor.

Já em “Preferências”, como mostrado na [Figura 31](#), o usuário pode definir quais notificações e alertas deseja receber, o tempo para relembrete das notificações e os valores mínimo e máximo de glicemia, que por padrão são 70 e 120 mg/dL. Bem como, no caso da notificação de glicemia estar ativa, os horários em que deseja ser lembrado de registrá-la.

5.2.3.5 Notificações

As notificações enviadas pelo aplicativo podem ser visualizadas na barra de notificações e possuem um botão interativo para adiá-las, como mostram a [Figura 32](#) e a [Figura 33](#). Por enquanto, o tempo adiado se refere ao “Tempo para reembrete” definido nas “Preferências”, este que também define os minutos de antecedência do horário das notificações e registro glicemia.

Figura 32 – Notificação de Medicação.

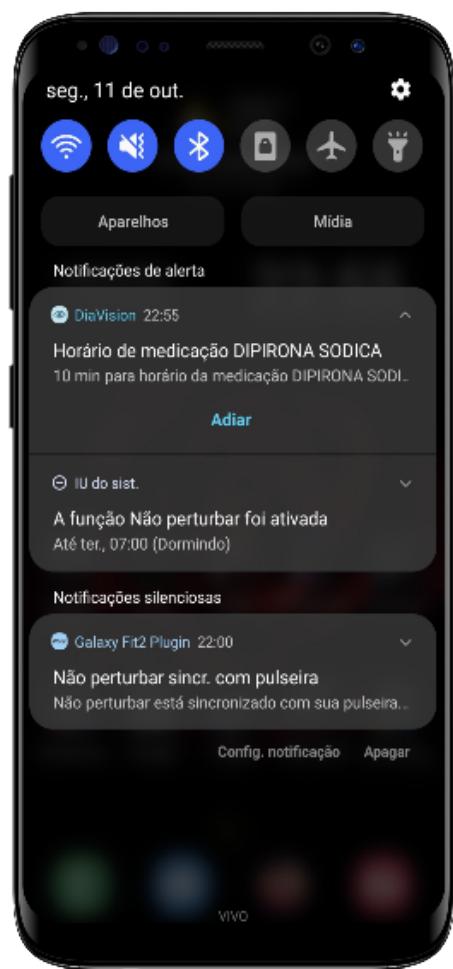
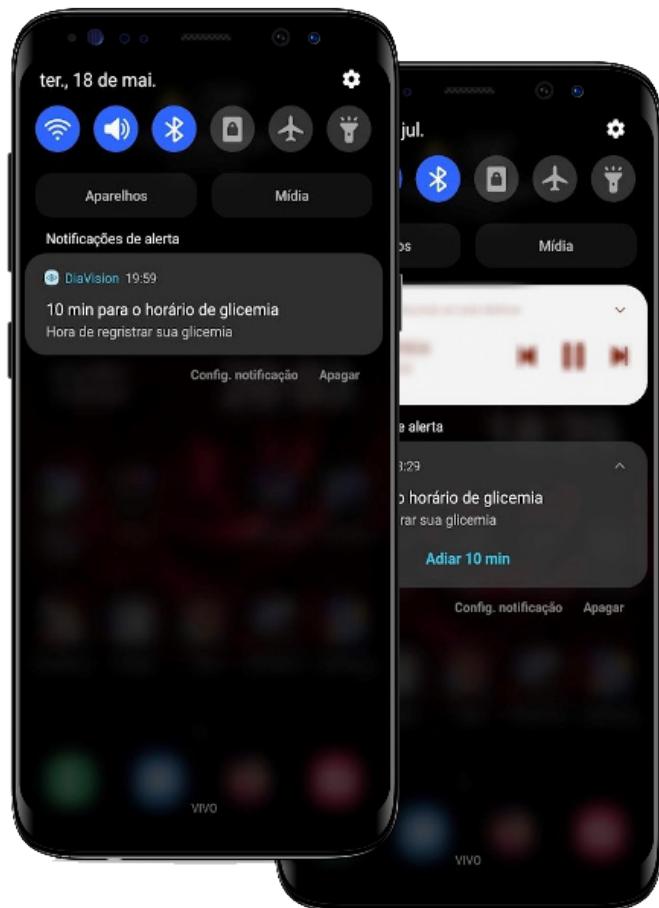


Figura 33 – Notificação de Glicemia.



Fonte: Autor.

Fonte: Autor.

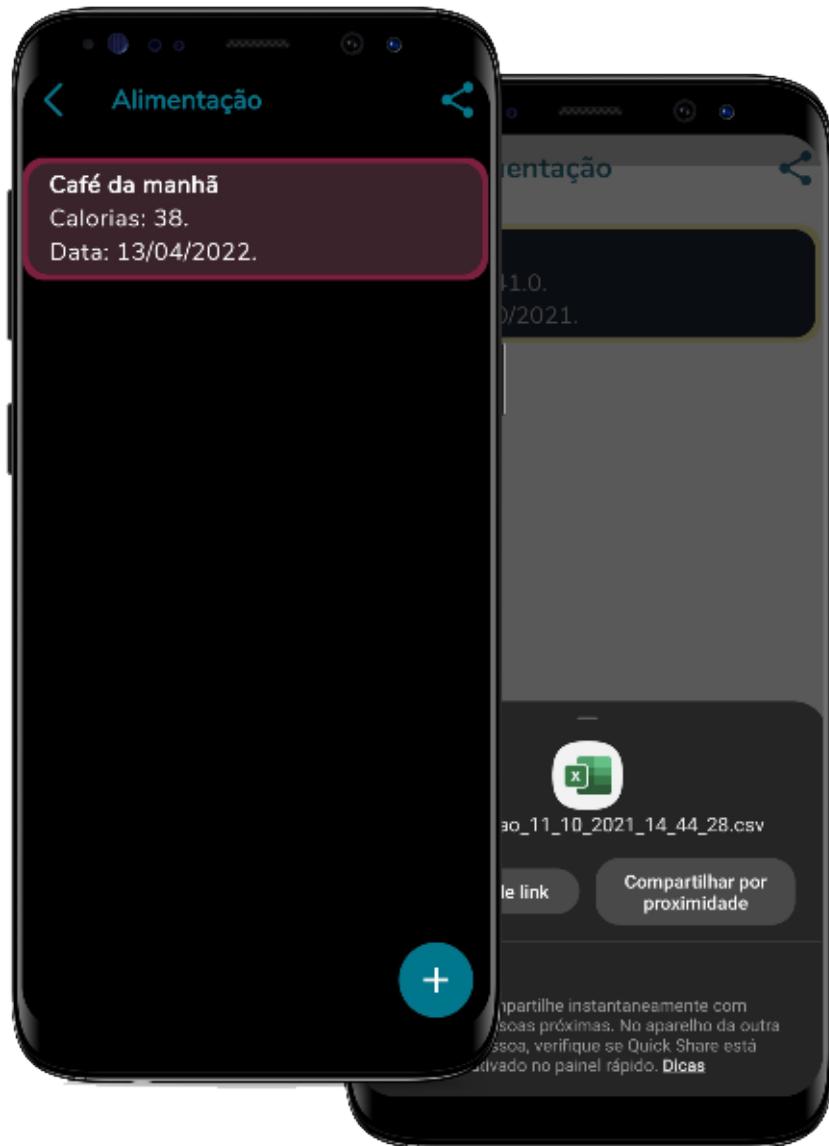
As notificações de medicações são enviadas para lembrar o usuário de que ele precisa tomar uma medicação nos horários pré-definidos pelo mesmo e para cada medicação. O cadastro de medicações a ativação dessas notificações será abordado mais à frente.

Já as notificações de glicemia, vistas na [Figura 33](#), são enviadas periodicamente, também nos horários definidos pelo usuário, para lembrá-lo de que precisa medir e registrar sua glicemia.

5.2.3.6 Módulo Alimentacão

Na primeira tela deste módulo, como pode-se observar na [Figura 34](#), são listados todos os registros de alimentação feitos pelo usuário. Também há um botão “compartilhar” no canto superior direito, por meio do qual é possível exportar esses registros, no formato CSV⁹, e compartilhá-los para outros *apps*.

Figura 34 – Tela Inicial do Módulo Alimentação.



Fonte: Autor.

Ao clicar no botão “+”, no canto inferior direito da [Figura 34](#), o usuário é direcionado para a tela de cadastro de alimentação.

⁹ Valores Separados por Vírgula (CSV, do inglês *Comma-separated values*) é um formato de arquivo de texto comumente utilizado por *softwares* de planilhas como o Google Planilhas, o Microsoft Excel e o *software* livre LibreOffice Calc.

Para registrar uma alimentação é necessário definir o tipo (Almoço, Jantar etc) e adicionar cada alimento, informando as calorias de cada porção consumida. Porém, também é possível não adicionar os alimentos e informar as calorias totais da refeição diretamente.

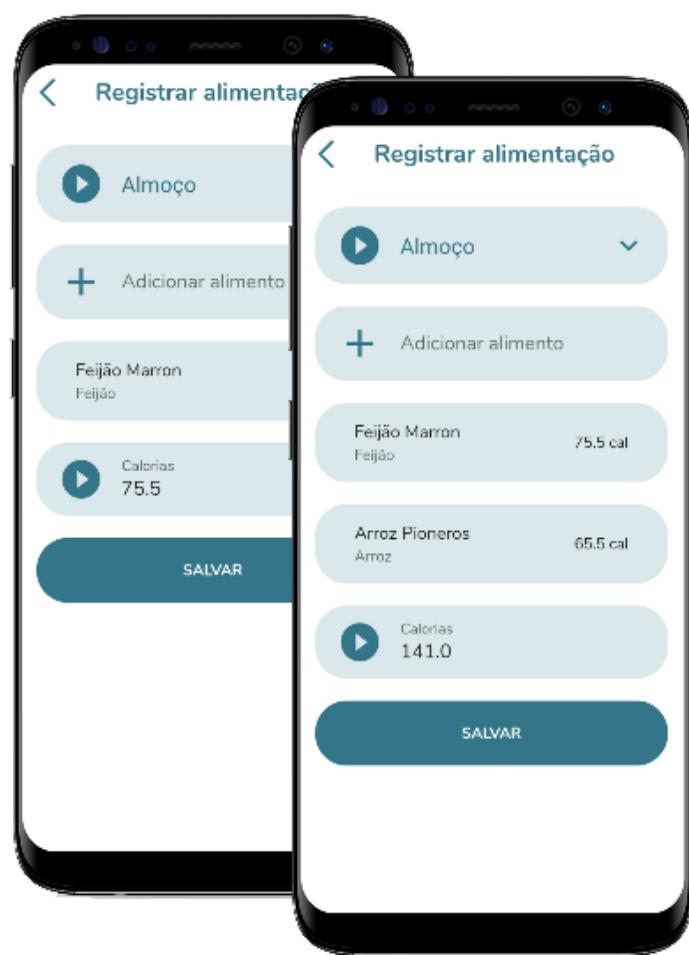
A [Figura 35](#) mostra a tela de registro de alimentação no estado inicial, sem nenhum alimento adicionado. A medida que alimentos vão sendo adicionados, esses vão aparecendo na tela e o valor das calorias também vai sendo atualizado automaticamente com os valores das calorias consumidas de cada alimento, como observa-se na [Figura 36](#).

Figura 35 – Registro de Alimentação.



Fonte: Autor.

Figura 36 – Adicionando Alimentos.



Fonte: Autor.

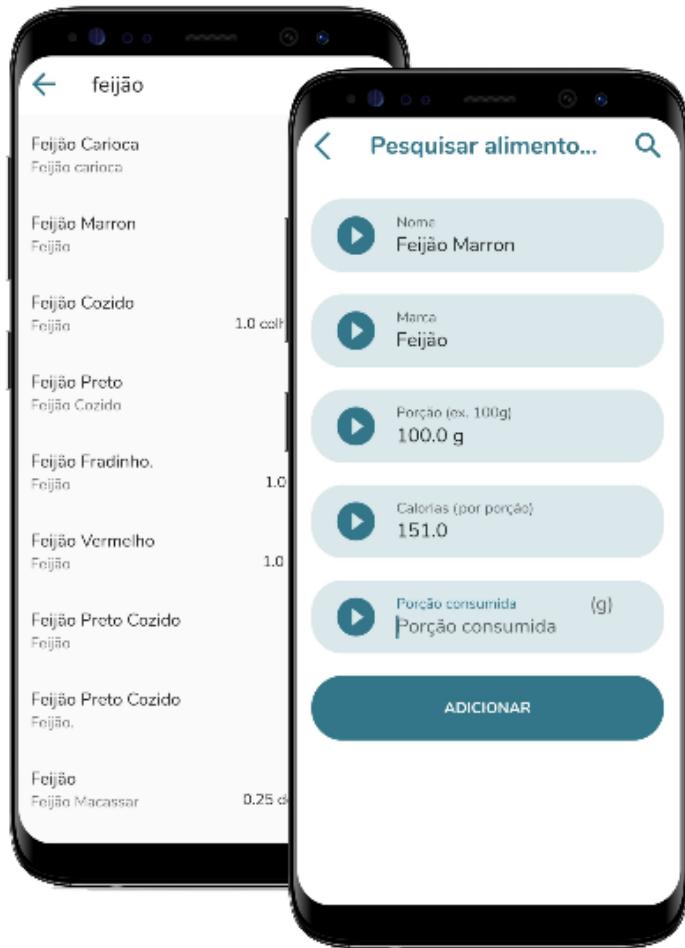
Para que o cálculo das calorias seja feito automaticamente a cada alimento adicionado, o usuário precisa informar os dados necessários, sendo estes: nome, porção, calorias por porção e porção consumida. O usuário não é obrigado a informar nenhum desses dados, porém, caso não informe, o cálculo não poderá ser feito.

Cada alimento pode ser adicionado manualmente ou por meio da opção “Pesquisar alimento...”, localizada na barra superior do aplicativo na [Figura 37](#), na qual o usuário informa o nome do alimento ou parte dele e o *app* faz uma busca pelo alimento referido.

Figura 37 – Registro de Alimento.



Figura 38 – Buscar Alimento.



Fonte: Autor.

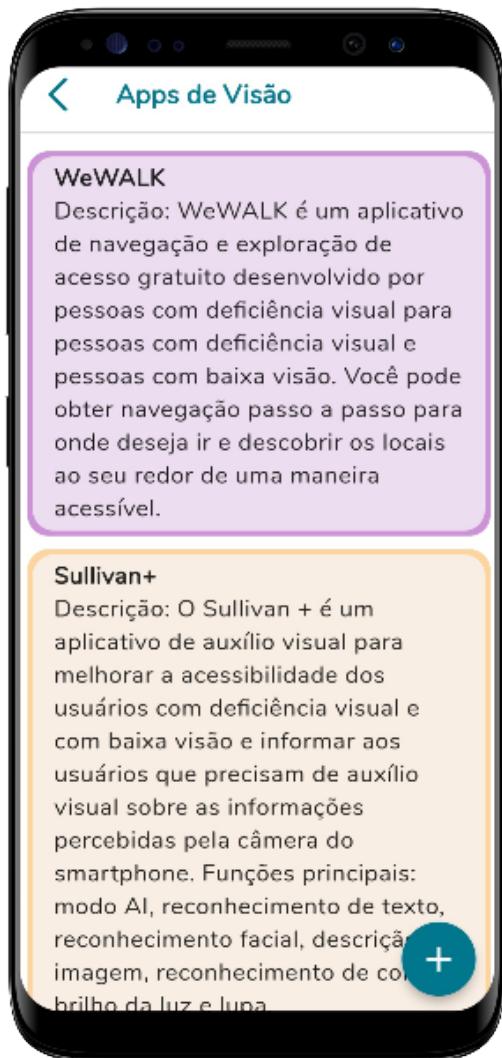
Fonte: Autor.

Ao realizar a busca, o *app* retorna uma lista de alimentos que possuem os nomes parecidos com o informado, como nota-se na [Figura 38](#). Os alimentos são retornados com os valores de porção, unidade de medida e calorias por porção, assim, ao selecionar um desses alimentos, resta ao usuário apenas informar a quantidade consumida na mesma unidade indicada para o alimento selecionado.

5.2.3.7 Módulo Apps de Visão

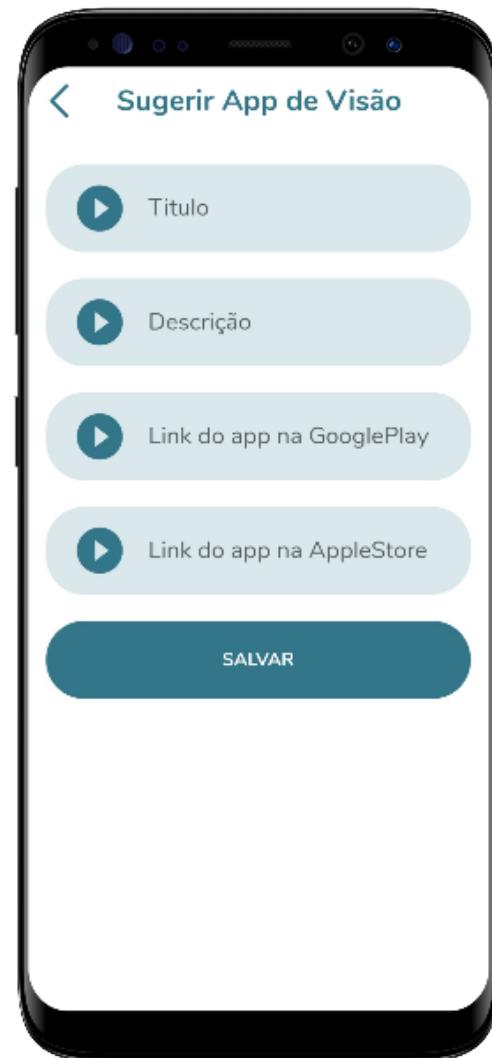
Neste módulo, são apresentados sugestões de aplicativos de diversos temas que auxiliam, de alguma forma, pessoas com deficiência visual, como constata-se na [Figura 39](#). Ao clicar em um deles, o usuário será direcionado à página do *app* na loja de aplicativos, de acordo com SO do *smartphone* utilizado.

Figura 39 – Listagem de Apps de Visão.



Fonte: Autor.

Figura 40 – Sugerir App de Visão.



Fonte: Autor.

Vale ressaltar que as sugestões podem ser diferentes de acordo com o SO, caso um *app* esteja disponível apenas para Android, não aparecerá na listagem ao acessar esse módulo em um dispositivo iOS e vice-versa.

O usuário também pode sugerir algum aplicativo que acredite ser útil para pessoas com deficiência visual, como visto na [Figura 40](#). As sugestões recebidas são aprovadas por um administrador do sistema, se estiverem de acordo com o esperado.

5.2.3.8 Módulo Atividade Física

Este módulo tem o objetivo de oferecer uma opção para gerenciamento das atividades físicas praticadas pelo usuário. Nesta tela inicial há uma listagem simples dos exercícios físicos registrados, como mostra a [Figura 41](#). Ao clicar em um deles, o usuário é direcionado a página de edição da atividade, para que possa alterar algum dado informado.

Figura 41 – Listagem de Atividades Física.



Fonte: Autor.

Figura 42 – Registro de Atividade Física.



Fonte: Autor.

Por enquanto pode-se registrar apenas exercícios considerando o tempo que foi praticado e a distância, sendo que, esta última se aplica somente nos dois tipos predefinidos (caminhada e corrida). Neste módulo também há a opção de compartilhar os registros no formato CSV.

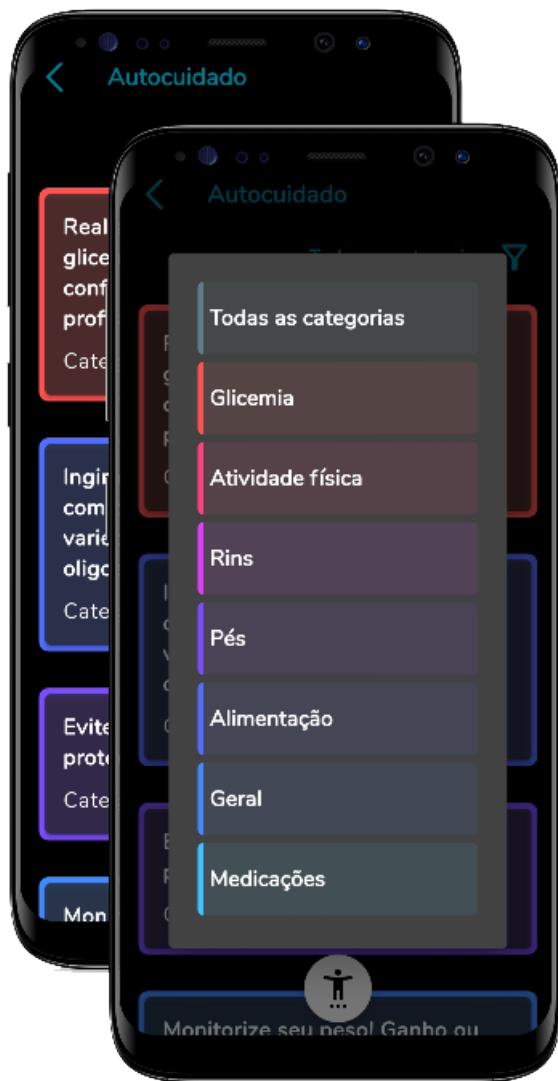
Para o registro mostrado na [Figura 42](#), o primeiro campo solicitado é o tipo, que pode ser um dos predefinidos ou “Outro”, este que pode ser informado pelo usuário no campo que surge quando essa opção é selecionada. O segundo campo varia de acordo com a opção selecionada no primeiro. E o terceiro é a duração, em minutos, do exercício.

5.2.3.9 Módulo Autocuidado

Na tela inicial deste módulo, como mostra a Figura 43, são listadas dicas de autocuidado divididas em diversas categorias, onde o usuário pode filtrar por alguma específica ou manter todas as categorias na listagem.

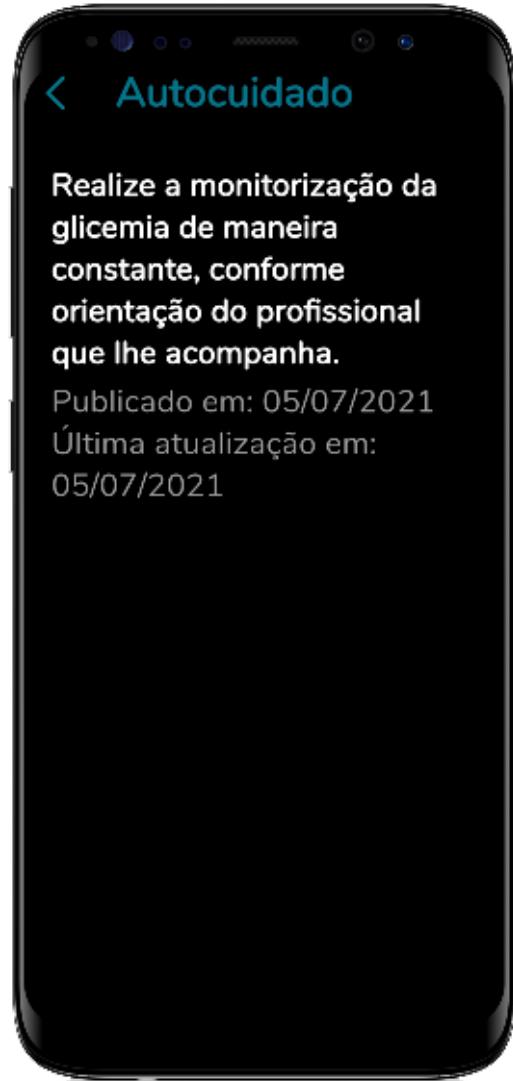
Essas dicas são cadastradas, exclusivamente, por um administrador do sistema de acordo com recomendações validadas por profissionais da saúde.

Figura 43 – Listagem de Autocuidados.



Fonte: Autor.

Figura 44 – Tela de Autocuidado.



Fonte: Autor.

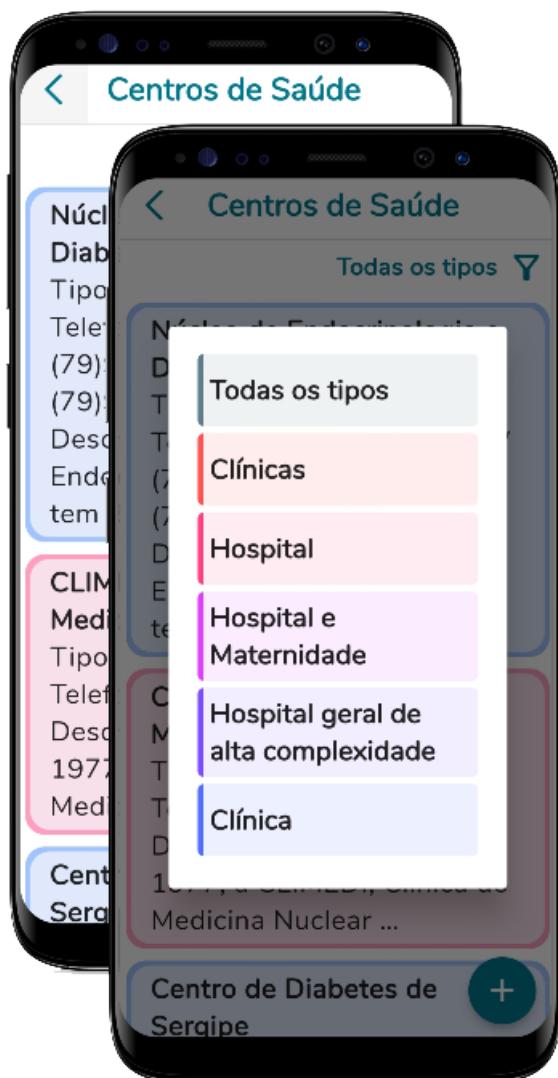
Ao clicar em um dos componentes de autocuidado, é aberta uma nova tela com os detalhes. Essa tela pode conter título, descrição e as datas de publicação e última atualização da dica, como foi mostrado na Figura 44.

5.2.3.10 Módulo Centros de Saúde

Neste módulo são listados, na tela principal, centros de saúde com informações sobre o tipo, nome, descrição, telefones para contato, endereço e localização no Google Maps¹⁰.

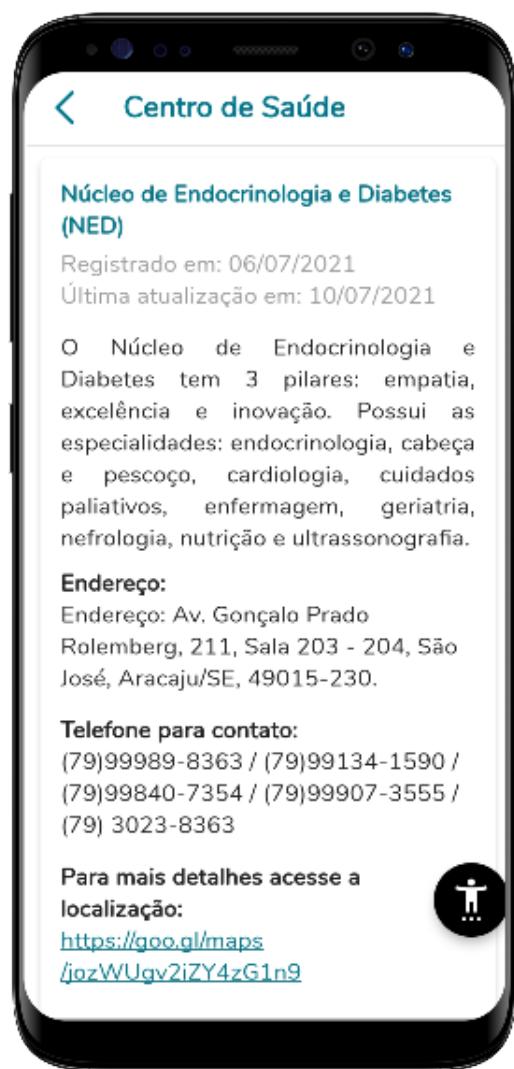
Como pode-se observar na [Figura 45](#), a listagem possui uma filtragem por tipo, assim, em conjunto com às funcionalidades de acessibilidade do *app*, o usuário pode encontrar por centros de saúde que atendam sua demanda de maneira mais acessível.

Figura 45 – Listagem de Centros de Saúde.



Fonte: Autor.

Figura 46 – Tela do Centro de Saúde.



Fonte: Autor.

Ao clicar em um dos componentes de centro de saúde, o usuário é direcionado para uma tela como a da [Figura 46](#), na qual ele pode visualizar os detalhes desse centro.

Além disso, o usuário também pode sugerir novos centros de saúde para o sistema, funcionando da mesma forma das sugestões de *apps* de visão.

¹⁰ O Google Maps é um serviço de busca e visualizar de mapas, localizações e rotas oferecido pelo Google de forma gratuita.

5.2.3.11 Módulo Glicemia

Na tela inicial do módulo de glicemia é exibida uma listagem dos registros de glicemia feitos pelo usuário. Esses registros também podem ser exportados da mesma forma que os registros de alimentação e atividade física.

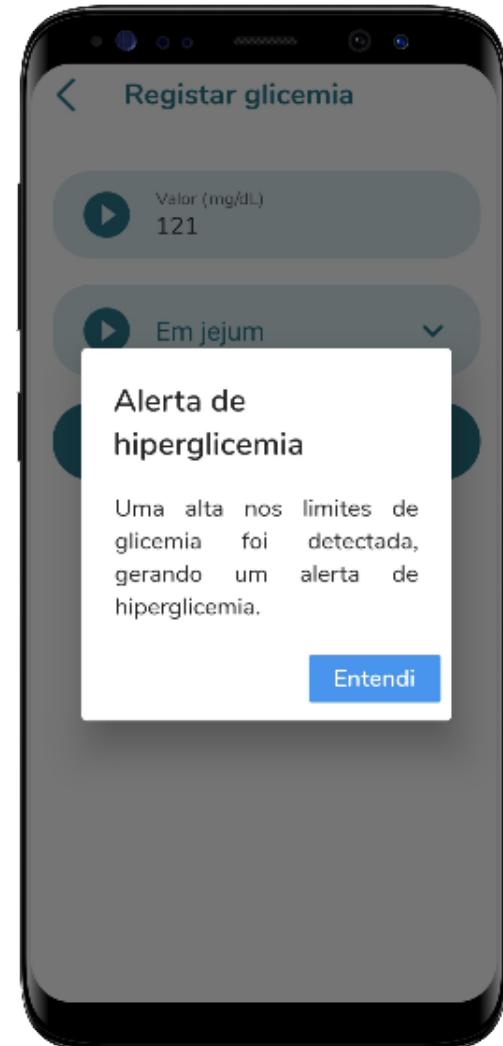
Ao clicar no botão “+” no fim da tela mostrada na [Figura 47](#), é possível inserir um novo registro. Porém, na primeira vez que o botão é pressionado, um dialogo é exibido perguntando se o usuário quer manter os limites padrões de glicemia ou definir os próprios, como mostra a figura.

Figura 47 – Listagem de Registros de Glicemia.



Fonte: Autor.

Figura 48 – Alerta de Glicemia.



Fonte: Autor.

Ao registrar um valor de glicemia, caso ele seja superior ao limite máximo ou inferior ao mínimo, um alerta de hiper ou hipoglicemia, respectivamente, será exibido, de acordo com os valores definidos nas “Preferências”.

5.2.3.12 Módulo Medicações

O objetivo geral deste módulo é que o usuário registre todas as medicações que faz uso para que possa ser notificado alguns minutos antes de cada horário de medicação.

Assim, como mostra a [Figura 49](#), na primeira tela do módulo são listadas as medicações que o usuário registrou. Para cada uma dessas medicações, é exibido um botão para ativar as notificações, caso ele esteja verde, as notificações para essa medicação estão desativadas, e ativadas caso contrário, como pode ser visto na [Figura 50](#).

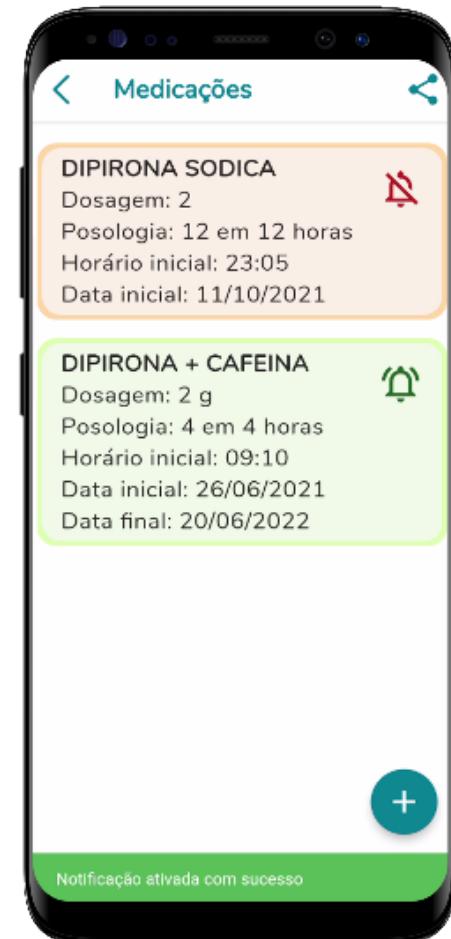
É importante ressaltar que o *feedback* por voz da aplicação, caso esteja ativo, se aplica também à mensagens de alerta como a exibida na [Figura 50](#).

Figura 49 – Listagem de Medicações.



Fonte: Autor.

Figura 50 – Notificar de Medicação.



Fonte: Autor.

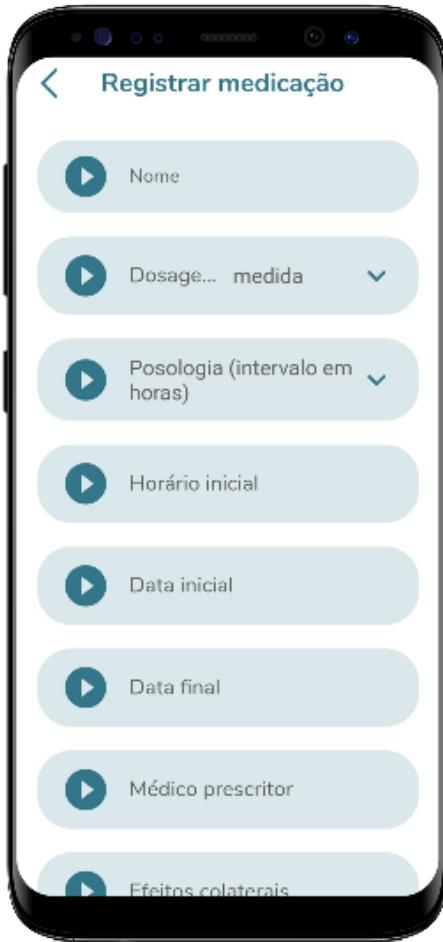
As notificações de medicações são baseadas no horário inicial e na posologia, e aparecem determinado tempo antes de cada horário, sendo 10 minutos por padrão e podendo ser alterado nas “Preferências”.

Para registro de medicação, alguns dados como nome, dosagem, posologia, horário inicial e data final, são importantes para que o usuário possa ser notificado. Isso porque a posologia

(intervalo entre as doses) e o horário inicial definem os horários de notificação. A data final define quando as notificações devem parar. E, o nome e a dosagem são exibidos na notificação de lembrete.

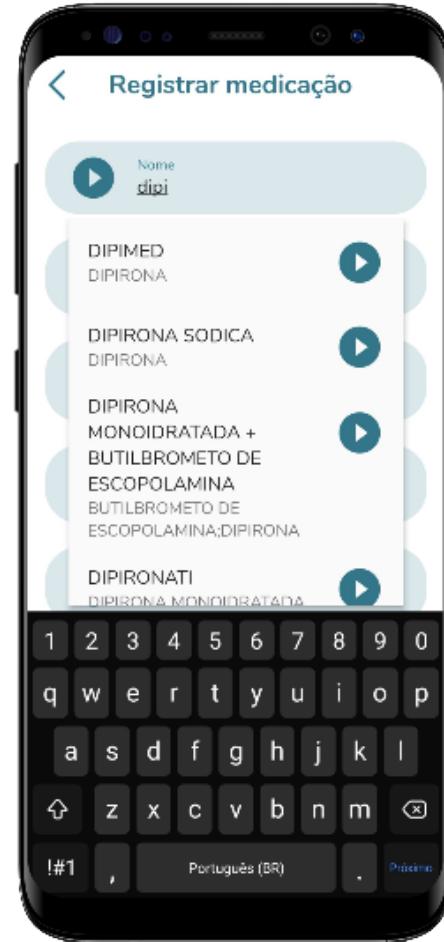
Porém, nenhum desses dados, exibidos na Figura 51, são obrigatórios. Assim, se não informados, o usuário não será notificado nos horários que deve tomar as medicações.

Figura 51 – Registro de Medicação.



Fonte: Autor.

Figura 52 – Autocomplete de Medicamentos.



Fonte: Autor.

Ao digitar o nome da medicação no campo “Nome”, após a quarta letra, o *app* começa a buscar e filtrar sugestões para a medicação que está sendo buscada, como mostra o exemplo na Figura 52.

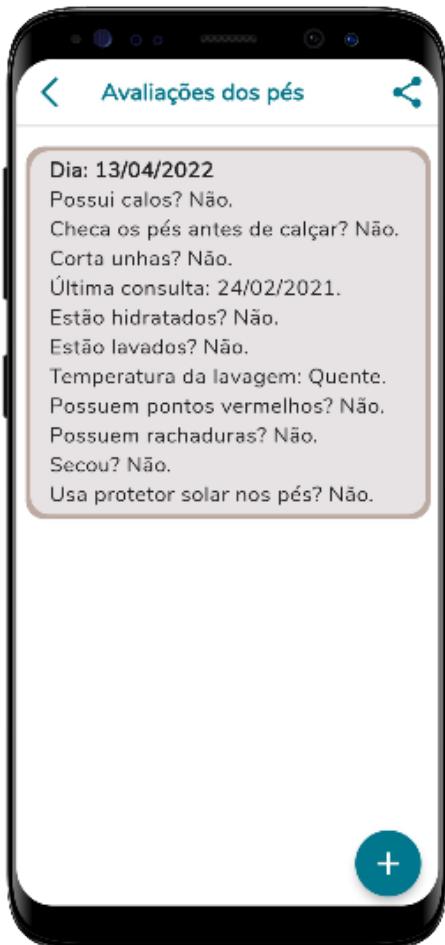
Contudo, nessa listagem de sugestões, diferente da maioria dos componentes do aplicativo, não é possível ouvir o *feedback* com as descrições ao manter os itens pressionados. Isso ocorre devido a uma limitação do componente utilizado e por esta razão existe o “ícone de play” ao lado de cada item para cumprir esse papel.

5.2.3.13 Módulo Pés

Este módulo serve para que o usuário possa acompanhar a evolução dos cuidados com os pés, com o registro diário de avaliação do estado dos mesmos.

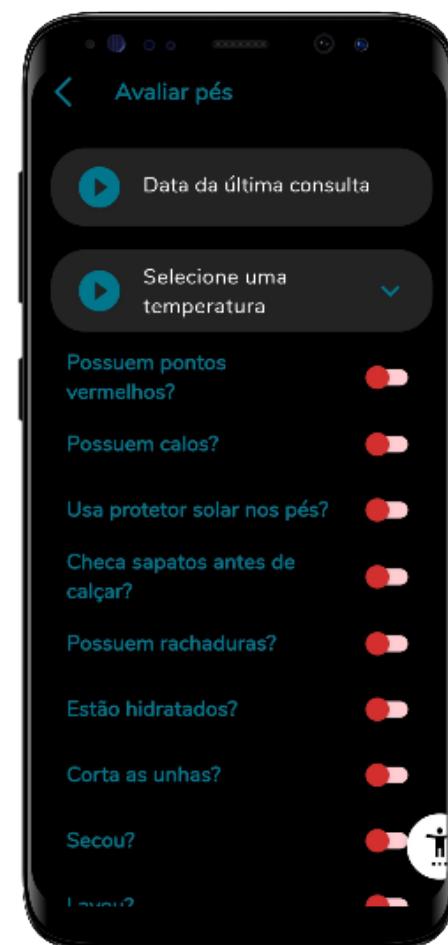
Como nos demais módulos de registros pessoais do usuário, no canto superior direito da Figura 53 tem um botão para compartilhar esses registros. Assim, o usuário pode utilizar esse recurso para compartilhar as avaliações com seu podologista ou com algum familiar responsável por acompanhá-lo na visita ao profissional.

Figura 53 – Listagem de Avaliações dos Pés.



Fonte: Autor.

Figura 54 – Avaliar Pés.



Fonte: Autor.

No registro de avaliação, o usuário pode informar a data da última consulta e a temperatura da água utilizada para lavagem dos pés. Além disso, há várias perguntas de “Sim” ou “Não” com detalhes que podem descrever o estado dos pés do usuário em cada registro, como mostrou a Figura 54.

5.2.3.14 Módulo Rins

Este é o último módulo e se refere ao acompanhamento dos rins e diurese - quantidade de urina produzida em determinado período -, listando os registros diários de diurese na tela inicial como pode-se observar na [Figura 55](#).

Essas informações podem auxiliar o usuário na visita ao profissional da saúde, pois, oferecem dados que possibilitam a avaliação da função renal. Assim, também há a funcionalidade para compartilhamento desses registros.

Figura 55 – Listagem de Diurese.



Fonte: Autor.

Figura 56 – Avaliar Diurese.



Fonte: Autor.

Na avaliação de diurese, conforme mostrado na [Figura 56](#), são solicitados o volume em mL (mililitros), uma avaliação da coloração da urina e se houve ardor.

Como esses registros sempre acrescentam a data automaticamente, o usuário pode fazer o registro a cada vez que urina ou somente uma vez somando todo o volume diário. A estratégia fica a cargo do usuário, devendo o mesmo relatar o método utilizado para o profissional responsável.

5.3 Considerações Finais

Este capítulo buscou elucidar o processo de desenvolvimento, relatando a estrutura do projeto, as tecnologias utilizadas, as soluções de acessibilidade adotas e o resultado final do sistema DiaVision. Este que incluiu o aplicativo móvel multiplataforma e a aplicação *backend* para gerenciamento dos dados, que também ofereceu suporte para criação de um *dashboard* para administração de dados do sistema.

6

Considerações Finais e Trabalhos Futuros

Este trabalho buscou contextualizar e fundamentar a problemática identificada pela dificuldade no acesso à informações sobre o autocuidado com o DM por pacientes com DV. Para isso, foram introduzidos o DM e suas complicações relacionadas à DV, bem como, foram apresentados dados que indicam o crescimento no número de casos de ambos.

Por meio de estudos anteriores, em um trabalho de mestrado que acarretou na parceira para desenvolvimento deste projeto, foi possível um aprofundamento sobre as necessidades e requisitos do público-alvo. Nos quais foram identificadas a importância do autocuidado no tratamento do DM e as principais funcionalidades utilizadas como solução no mercado ([SOBRAL, 2021](#)).

Outra problemática introduzida foi que, mesmo com o aumento da informatização e popularização dos *smartphones*, PDV ainda enfrentam sérias dificuldades devido à falta de acessibilidade à DV em aplicações móveis ([SHERA et al., 2021](#)). Diante dessa problemática, foi realizado um processo de MSL visando identificar as principais soluções que estão sendo adotadas. Além disso, foram apontadas as principais ferramentas e diretrizes relacionadas à acessibilidade disponibilizadas pelas plataformas móveis.

Nesse processo de mapeamento, foram extraídas informações relevantes de trabalhos publicados em bases acadêmicas que apresentaram técnicas para resolver esses problemas em aplicações móveis. A partir da análise dos resultados do MSL, estabeleceram-se as técnicas de acessibilidade que seriam utilizadas na solução proposta.

Por fim, aliando os benefícios do autocuidado na redução das complicações ocasionadas pelo DM ([ADA, 2019](#)) ao crescimento no acesso à Internet por meio de *smartphones* ([CETIC.BR, 2021](#)) e às principais técnicas para solução dos problemas de acessibilidade em *apps* móveis, propõe-se o desenvolvimento de uma aplicação com a combinação dessas características.

Referências

- ADA. 8. obesity management for the treatment of type 2 diabetes: Standards of medical care in diabetes—2019. *Diabetes Care*, American Diabetes Association, v. 42, n. Supplement 1, p. S81–S89, 2019. ISSN 0149-5992. Disponível em: <https://care.diabetesjournals.org/content/42/Supplement_1/S81>. Acesso em: 14 nov. 2021. Citado 4 vezes nas páginas 15, 16, 18 e 110.
- AMINUDDIN, H. B. et al. Effectiveness of smartphone-based self-management interventions on self-efficacy, self-care activities, health-related quality of life and clinical outcomes in patients with type 2 diabetes: A systematic review and meta-analysis. *International Journal of Nursing Studies*, v. 116, p. 103286, 2021. ISSN 0020-7489. Self-care in long term conditions. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0020748919300306>>. Acesso em: 14 nov. 2021. Citado na página 18.
- APPLE. Ative e treine o voiceover no iphone. 2021. Disponível em: <<https://support.apple.com/pt-br/guide/iphone/iph3e2e415f/ios>>. Acesso em: 14 nov. 2021. Citado na página 21.
- BACH, C. F. et al. *DIRETRIZES DE ACESSIBILIDADE: UMA ABORDAGEM COMPARATIVA ENTRE WCAG E E-MAG*. IBEPES (Instituto Brasileiro de Estudos e Pesquisas Sociais), 2009. Disponível em: <<http://dx.doi.org/10.5329/RESI.2009.0801001>>. Acesso em: 14 nov. 2021. Citado na página 20.
- BALLANTYNE, M. et al. Study of accessibility guidelines of mobile applications. In: . Association for Computing Machinery, 2018. p. 305–315. ISBN 9781450365949. Disponível em: <<https://dl.acm.org/doi/abs/10.1145/3282894.3282921>>. Acesso em: 14 nov. 2021. Citado na página 21.
- BI, T. et al. Accessibility in software practice: A practitioner's perspective. *CoRR*, abs/2103.08778, 2021. Disponível em: <<https://arxiv.org/abs/2103.08778>>. Acesso em: 14 nov. 2021. Citado na página 21.
- BIASE, L. C. D. et al. An accessible roller coaster simulator for touchscreen devices: An educational game for the visually impaired. In: *2018 IEEE Games, Entertainment, Media Conference (GEM)*. [s.n.], 2018. p. 101–105. Disponível em: <<https://ieeexplore.ieee.org/document/8516457>>. Acesso em: 14 nov. 2021. Citado 3 vezes nas páginas 22, 34 e 37.
- BOUKHARY, S.; COLMENARES, E. A clean approach to flutter development through the flutter clean architecture package. In: IEEE. *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2019. p. 1115–1120. Disponível em: <<https://ieeexplore.ieee.org/document/9071367>>. Acesso em: 14 nov. 2021. Citado na página 24.
- BRASIL. Lei brasileira de inclusão da pessoa com deficiência. 2015. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm>. Acesso em: 14 nov. 2021. Citado na página 19.

- CABALLERO, A. R.; CATLI, K. E. I.; BABIERRA, A. G. F. Object recognition and hearing assistive technology mobile application using convolutional neural network. In: *Proceedings of the International Conference on Wireless Communication and Sensor Networks*. New York, NY, USA: Association for Computing Machinery, 2020. (icWCSN 2020), p. 41–48. ISBN 9781450377638. Disponível em: <<https://dl.acm.org/doi/10.1145/3411201.3411208>>. Acesso em: 14 nov. 2021. Citado 3 vezes nas páginas 22, 34 e 44.
- CANTÙ, N. et al. Mathmelodies 2: A mobile assistive application for people with visual impairments developed with react native. In: *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. New York, NY, USA: Association for Computing Machinery, 2018. (ASSETS '18), p. 453–455. ISBN 9781450356503. Disponível em: <<https://dl.acm.org/doi/10.1145/3234695.3241006>>. Acesso em: 14 nov. 2021. Citado 4 vezes nas páginas 34, 44, 63 e 64.
- CERVO, A.; BERVIAN, P.; SILVA, R. da. *Metodologia científica*. Pearson Prentice Hall, 2006. ISBN 9788576050476. Disponível em: <<https://books.google.com.br/books?id=9SK2GQAACAAJ>>. Acesso em: 15 nov. 2021. Citado na página 16.
- CETIC.BR. Pesquisa web sobre o uso da internet no brasil durante a pandemia do novo coronavírus – painel tic covid-19. *Centro de Estudos sobre as Tecnologias da Informação e da Comunicação*, abr. 2021. Disponível em: <<https://cetic.br/pt/publicacao/painel-tic-covid-19/>>. Acesso em: 14 nov. 2021. Citado 2 vezes nas páginas 23 e 110.
- COOK, A. M.; POLGAR, J. M. *Assistive technologies: Principles and practice: Fourth edition*. [s.n.], 2014. Disponível em: <<https://books.google.com.br/books?id=ODWaBQAAQBAJ>>. Acesso em: 14 nov. 2021. Citado na página 19.
- DAMACENO, R. J. P.; BRAGA, J. C.; CHALCO, J. P. M. Mobile device accessibility for the visually impaired: Problems mapping and empirical study of touch screen gestures. In: *Proceedings of the 15th Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2016. (IHC '16). ISBN 9781450352352. Disponível em: <<https://dl.acm.org/doi/10.1145/3033701.3033703>>. Acesso em: 14 nov. 2021. Citado 8 vezes nas páginas 50, 53, 54, 55, 62, 63, 64 e 65.
- DUARTE, C. et al. Designing multimodal mobile interaction for a text messaging application for visually impaired users. *Frontiers in ICT*, Frontiers Media S.A., v. 4, n. DEC, 2017. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85061981298&doi=10.3389%2ffict.2017.00026&partnerID=40&md5=c6d6a2fcab00d15edc7f162bf576c772>>. Acesso em: 14 nov. 2021. Citado 3 vezes nas páginas 34, 40 e 64.
- ELER, M. M. et al. Automated accessibility testing of mobile apps. *2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)*, p. 116–126, 2018. Disponível em: <<https://ieeexplore.ieee.org/document/8367041>>. Acesso em: 14 nov. 2021. Citado na página 22.
- EMAG. emag - modelo de acessibilidade em governo eletrônico. 2014. Disponível em: <<http://emag.governoeletronico.gov.br/>>. Acesso em: 14 nov. 2021. Citado na página 20.
- FRANCESE, R. et al. Model-driven development for multi-platform mobile applications. In: ABRAHAMSSON, P. et al. (Ed.). *Product-Focused Software Process Improvement*. Cham: Springer International Publishing, 2015. p. 61–67. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-26844-6_5>. Acesso em: 14 nov. 2021. Citado na página 24.

- GONSALVES, M. *Evaluating the mobile development frameworks Apache Cordova and Flutter and their impact on the development process and application characteristics*. Tese (Doutorado), 2019. Disponível em: <<https://scholarworks.calstate.edu/concern/theses/kp78gg98g>>. Acesso em: 14 nov. 2021. Citado na página 25.
- GOOGLE. Primeiros passos no android com o talkback. 2021. Disponível em: <https://support.google.com/accessibility/android/answer/6283677?hl=pt-BR&ref_topic=10601571>. Acesso em: 14 nov. 2021. Citado 2 vezes nas páginas 21 e 22.
- GUEDES, G. T. *UML 2-Uma abordagem prática*. Novatec Editora, 2018. Disponível em: <<https://books.google.com.br/books?id=mJxMDwAAQBAJ>>. Acesso em: 05 may. 2022. Citado na página 75.
- HEITKÖTTER, H.; HANSCHKE, S.; MAJCHRZAK, T. A. Evaluating cross-platform development approaches for mobile applications. In: CORDEIRO, J.; KREMPELS, K.-H. (Ed.). *Web Information Systems and Technologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 120–138. ISBN 978-3-642-36608-6. Disponível em: <https://link.springer.com/chapter/10.1007%2F978-3-642-36608-6_8>. Acesso em: 14 nov. 2021. Citado na página 24.
- IBGE. Cartilha do censo 2010 - pessoas com deficiências. *Secretaria de Direitos Humanos da Presidência da República (SDH/PR), Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência (SNPD)*, p. 32, 2012. ISSN 1098-6596. Disponível em: <<https://inclusao.enap.gov.br/wp-content/uploads/2018/05/cartilha-censo-2010-pessoas-com-deficienciareduzido-original-eleitoral.pdf>>. Acesso em: 14 nov. 2021. Citado na página 19.
- ITU. Connectivity in the least developed countries: Status report 2021. *International Telecommunication Union*, set. 2021. Disponível em: <<https://www.itu.int/en/myitu/Publications/2021/09/17/11/46/Connectivity-in-the-Least-Developed-Countries-Status-report-2021>>. Acesso em: 14 nov. 2021. Citado na página 16.
- JAEGER, P. T. Assessing section 508 compliance on federal e-government web sites: A multi-method, user-centered evaluation of accessibility for persons with disabilities. *Government Information Quarterly*, v. 23, n. 2, p. 169–190, 2006. ISSN 0740-624X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0740624X06000487>>. Acesso em: 14 nov. 2021. Citado na página 20.
- KIM, W. et al. Effect of ux design guideline on the information accessibility for the visually impaired in the mobile health apps. In: . Institute of Electrical and Electronics Engineers Inc., 2019. p. 1103–1106. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85062552457&doi=10.1109%2fBIBM.2018.8621471&partnerID=40&md5=b0ac4a92a73fedbd9803f08ab427814e>>. Acesso em: 14 nov. 2021. Citado 5 vezes nas páginas 50, 52, 53, 63 e 64.
- KING, B. A.; YOUNGBLOOD, N. E. E-government in alabama: An analysis of county voting and election website content, usability, accessibility, and mobile readiness. *Government Information Quarterly*, v. 33, n. 4, p. 715–726, 2016. ISSN 0740-624X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0740624X16301691>>. Acesso em: 14 nov. 2021. Citado na página 20.

KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2007. Disponível em: <https://www.elsevier.com/__data/promis_misc/52544systemicreviewsguide.pdf>. Acesso em: 14 nov. 2021. Citado 2 vezes nas páginas 26 e 50.

KUZMIN, N.; IGNATIEV, K.; GRAFOV, D. Experience of developing a mobile application using flutter. In: *Information Science and Applications*. Springer, 2020. p. 571–575. Disponível em: <https://link.springer.com/chapter/10.1007/978-981-15-1465-4_56>. Acesso em: 14 nov. 2021. Citado na página 24.

LEPORINI, B.; PALMUCCI, E. A mobile educational game accessible to all, including screen reading users on a touch-screen device. In: *Proceedings of the 16th World Conference on Mobile and Contextual Learning*. New York, NY, USA: Association for Computing Machinery, 2017. (mLearn 2017). ISBN 9781450352550. Disponível em: <<https://dl.acm.org/doi/10.1145/3136907.3136941>>. Acesso em: 14 nov. 2021. Citado 3 vezes nas páginas 34, 35 e 64.

MADRIGAL-CADAVID, J. et al. Design and development of a mobile app of drug information for people with visual impairment. *Research in Social and Administrative Pharmacy*, v. 16, n. 1, p. 62–67, 2020. ISSN 1551-7411. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1551741119301317>>. Acesso em: 14 nov. 2021. Citado 2 vezes nas páginas 34 e 40.

MARRERO, D. G. et al. *Twenty-First Century Behavioral Medicine: A Context for Empowering Clinicians and Patients With Diabetes: A consensus report*. American Diabetes Association, 2013. 463–470 p. Disponível em: <<http://dx.doi.org/10.2337/dc12-2305>>. Acesso em: 14 nov. 2021. Citado na página 16.

MARTINEZ, M.; LECOMTE, S. Towards the quality improvement of cross-platform mobile applications. In: *2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*. [s.n.], 2017. p. 184–188. Disponível em: <<https://ieeexplore.ieee.org/document/7972737/>>. Acesso em: 14 nov. 2021. Citado na página 24.

MASCETTI, S. et al. Wordmelodies: Supporting children with visual impairment in learning literacy. In: *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. New York, NY, USA: Association for Computing Machinery, 2019. (ASSETS '19), p. 642–644. ISBN 9781450366762. Disponível em: <<https://dl.acm.org/doi/10.1145/3308561.3354587>>. Acesso em: 14 nov. 2021. Citado 3 vezes nas páginas 34, 48 e 63.

MATEUS, D. A. et al. Accessibility of mobile applications: Evaluation by users with visual impairment and by automated tools. In: *Proceedings of the 19th Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2020. (IHC '20). ISBN 9781450381727. Disponível em: <<https://dl.acm.org/doi/10.1145/3424953.3426633>>. Acesso em: 14 nov. 2021. Citado na página 50.

MORRIS, J. T.; SWEATMAN, M.; JONES, M. L. Smartphone use and activities by people with disabilities: user survey 2016. *J Technol Pers Disabil*, v. 5, p. 50–66, 2017. Disponível em: <<https://scholarworks.csun.edu/bitstream/handle/10211.3/190202/JTPD-2017-p50-66.pdf?sequence=1>>. Acesso em: 14 nov. 2021. Citado na página 16.

- NAHAR, L.; SULAIMAN, R.; JAAFAR, A. Users' perception on usability aspects of a braille learning mobile application 'mbraille'. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, v. 11870 LNCS, p. 100–109, 2019. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85077844210&doi=10.1007%2f978-3-030-34032-2_10&partnerID=40&md5=01d0716dda193cd19f0d3b46dc897e13>. Acesso em: 14 nov. 2021. Citado 2 vezes nas páginas 34 e 47.
- NAWROCKI, J. et al. Agile requirements engineering: A research perspective. In: SPRINGER. *International Conference on Current Trends in Theory and Practice of Informatics*. 2014. p. 40–51. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-04298-5_5>. Acesso em: 17 nov. 2021. Citado na página 71.
- NILSON, E. A. F. et al. *Custos atribuíveis a obesidade, hipertensão e diabetes no Sistema Único de Saúde, Brasil, 2018*. Pan American Health Organization, 2020. 1 p. Disponível em: <<http://dx.doi.org/10.26633/RPSP.2020.32>>. Acesso em: 14 nov. 2021. Citado na página 15.
- OLIVEIRA, A. S. de et al. Quimivox mobile 2.0: Application for helping visually impaired people in learning periodic table and electron configuration. In: *Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2019. (IHC '19). ISBN 9781450369718. Disponível em: <<https://dl.acm.org/doi/10.1145/3357155.3358436>>. Acesso em: 14 nov. 2021. Citado 5 vezes nas páginas 22, 34, 45, 63 e 64.
- OLIVEIRA, B. de; BRAGA, J. C.; DAMACENO, R. J. P. Application for the configuration and adaptation of the android operating system for the visually impaired. In: *Proceedings of the 15th International Web for All Conference*. New York, NY, USA: Association for Computing Machinery, 2018. (W4A '18). ISBN 9781450356510. Disponível em: <<https://dl.acm.org/doi/10.1145/3192714.3192838>>. Acesso em: 14 nov. 2021. Citado 2 vezes nas páginas 34 e 38.
- OZOUNGWU, O. *The pathogenesis and pathophysiology of type 1 and type 2 diabetes mellitus*. Academic Journals, 2013. 46–57 p. Disponível em: <<http://dx.doi.org/10.5897/JPAP2013.0001>>. Acesso em: 14 nov. 2021. Citado na página 18.
- PATIL, N.; BHOLE, D.; SHETE, P. Enhanced ui automator viewer with improved android accessibility evaluation features. In: IEEE. *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*. 2016. p. 977–983. Disponível em: <<https://ieeexplore.ieee.org/document/7877733>>. Acesso em: 14 nov. 2021. Citado na página 23.
- PRESSMAN, R.; MAXIM, D. B. R. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education, 2014. ISBN 9780078022128. Disponível em: <<https://books.google.com.br/books?id=i8NmAEACAAJ>>. Acesso em: 14 nov. 2021. Citado 4 vezes nas páginas 23, 70, 73 e 74.
- QUISPE, F.; SCATALON, L.; ELER, M. Prioritization of mobile accessibility guidelines for visual impaired users. In: . SciTePress, 2020. v. 2, p. 563–570. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091396826&partnerID=40&md5=f30643b9bc6f6f843d4c94328b592afc>>. Acesso em: 14 nov. 2021. Citado 6 vezes nas páginas 21, 50, 57, 58, 61 e 63.

- RIEGER, C. et al. A model-driven approach to cross-platform development of accessible business apps. In: . New York, NY, USA: Association for Computing Machinery, 2020. (SAC '20), p. 984–993. ISBN 9781450368667. Disponível em: <<https://dl.acm.org/doi/10.1145/3341105.3375765>>. Acesso em: 14 nov. 2021. Citado 6 vezes nas páginas 22, 34, 36, 51, 63 e 64.
- ROCHA, J. A. P.; DUARTE, A. B. S. Diretrizes de acessibilidade web: um estudo comparativo entre as wcag 2.0 e o e-mag 3.0. *Inclusão Social*, v. 5, n. 2, dez. 2013. Disponível em: <<http://revista.ibict.br/inclusao/article/view/1678>>. Acesso em: 14 nov. 2021. Citado na página 20.
- SAEEDI, P. et al. Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the international diabetes federation diabetes atlas, 9th edition. *Diabetes Research and Clinical Practice*, v. 157, p. 107843, 2019. ISSN 0168-8227. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168822719312306>>. Acesso em: 14 nov. 2021. Citado na página 15.
- SHERA, A. et al. Blind and visually impaired user interface to solve accessibility problems. *Intelligent Automation and Soft Computing*, Tech Science Press, v. 30, n. 1, p. 285–301, 2021. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85111705547&doi=10.32604%2fiasc.2021.018009&partnerID=40&md5=506f2304e2021f8d02726bdb342599fd>>. Acesso em: 14 nov. 2021. Citado 6 vezes nas páginas 22, 34, 38, 39, 63 e 110.
- SHIN, H. et al. Improved and accessible e-book reader application for visually impaired people. In: *SIGGRAPH Asia 2017 Posters*. New York, NY, USA: Association for Computing Machinery, 2017. (SA '17). ISBN 9781450354059. Disponível em: <<https://dl.acm.org/doi/10.1145/3145690.3145748>>. Acesso em: 14 nov. 2021. Citado 4 vezes nas páginas 22, 34, 42 e 64.
- SIEBRA, C. et al. An analysis on tools for accessibility evaluation in mobile applications. In: . Association for Computing Machinery, 2018. p. 172–177. Cited By 2. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85055787089&doi=10.1145%2f3266237.3266238&partnerID=40&md5=0976ca72671e6471f46f66a763c35771>>. Acesso em: 14 nov. 2021. Citado na página 63.
- SIEBRA, C. et al. Observation based analysis on the use of mobile applications for visually impaired users. In: *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. New York, NY, USA: Association for Computing Machinery, 2016. (MobileHCI '16), p. 807–814. ISBN 9781450344135. Disponível em: <<https://dl.acm.org/doi/10.1145/2957265.2961848>>. Acesso em: 14 nov. 2021. Citado 3 vezes nas páginas 50, 56 e 64.
- SILVA, E. R. P. da. *Métodos para Revisão e Mapeamento Sistemático da Literatura (Methods for Systematic Literature Reviews and Systematic Mapping Studies)*. Tese (Doutorado) — Federal University of Rio de Janeiro, 03 2009. Disponível em: <https://www.researchgate.net/publication/303497814_Metodos_para_Revisao_e_Mapeamento_Sistematico_da_Literatura_Methods_for_Systematic_Literature_Reviews_and_Systematic_Mapping_Studies>. Acesso em: 14 nov. 2021. Citado na página 26.
- SOBRAL, D. A. Diavision: Ferramenta educacional em saúde para o autocuidado do diabetes em pacientes com acuidade visual prejudicada. In: . [S.l.: s.n.], 2021. Citado 9 vezes nas páginas 68, 69, 70, 71, 72, 73, 76, 77 e 110.

- SOLOMON, S. D. et al. Diabetic retinopathy: A position statement by the american diabetes association. *Diabetes Care*, American Diabetes Association, v. 40, n. 3, p. 412–418, 2017. ISSN 0149-5992. Disponível em: <<https://care.diabetesjournals.org/content/archive/40/3/412/1>>. Acesso em: 14 nov. 2021. Citado na página 15.
- SOMMERVILLE, I. et al. *Engenharia de software*. Pearson Prentice Hall, 2007. ISBN 9788588639287. Disponível em: <<https://books.google.com.br/books?id=ifIYOGAACAAJ>>. Acesso em: 17 nov. 2021. Citado na página 70.
- STOPA, S. R. et al. Pesquisa nacional de saúde 2019: histórico, métodos e perspectivas. *Epidemiologia e Serviços de Saúde*, SciELO Brasil, v. 29, 2020. Disponível em: <<https://doi.org/10.1590/S1679-49742020000500004>>. Acesso em: 14 nov. 2021. Citado na página 19.
- TAYLOR, K.; SILVER, L. Smartphone ownership is growing rapidly around the world, but not always equally | pew research center. p. 47, 2019. Disponível em: <<https://www.pewresearch.org/global/2019/02/05/smartphone-ownership-is-growing-rapidly-around-the-world-but-not-always-equally/>>. Acesso em: 14 nov. 2021. Citado na página 16.
- TOMLINSON, B. et al. "talkin' about the weather": Incorporating talkback functionality and sonifications for accessible app design. In: . Association for Computing Machinery, Inc, 2016. p. 377–386. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84991387146&doi=10.1145%2f2935334.2935390&partnerID=40&md5=773b9f0cb6471ee5944fbe8c0a03cdc4>>. Acesso em: 14 nov. 2021. Citado 3 vezes nas páginas 22, 34 e 47.
- VENDOME, C. et al. Can everyone use my app? an empirical study on accessibility in android apps. In: . Institute of Electrical and Electronics Engineers Inc., 2019. p. 41–52. Cited By 13. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85077219554&doi=10.1109%2fICSME.2019.00014&partnerID=40&md5=aea0355325a633ad2b12030536471926>>. Acesso em: 14 nov. 2021. Citado 4 vezes nas páginas 22, 50, 52 e 63.
- VITIELLO, G. et al. Do you like my outfit? cromnia, a mobile assistant for blind users. In: *Proceedings of the 4th EAI International Conference on Smart Objects and Technologies for Social Good*. New York, NY, USA: Association for Computing Machinery, 2018. (Goodtechs '18), p. 249–254. ISBN 9781450365819. Disponível em: <<https://dl.acm.org/doi/10.1145/3284869.3284908>>. Acesso em: 14 nov. 2021. Citado 2 vezes nas páginas 34 e 41.
- W3C. Mobile web best practices 1.0. 2008. Disponível em: <<https://www.w3.org/TR/mobile-bp/>>. Acesso em: 14 nov. 2021. Citado na página 20.
- W3C, W. W. W. C. Web content accessibility guidelines (wcag) overview. *Web Accessibility Initiative (WAI)*, 2019. Disponível em: <<https://www.w3.org/WAI/standards-guidelines/wcag>>. Acesso em: 14 nov. 2021. Citado na página 20.
- WALRATH, K.; LADD, S. *Dart: Up and Running: A New, Tool-Friendly Language for Structured Web Apps*. O'Reilly Media, 2012. ISBN 9781449330859. Disponível em: <<https://books.google.com.br/books?id=w2jC1KYCzcoC>>. Acesso em: 14 nov. 2021. Citado na página 24.

WASSERMAN, A. I. Software engineering issues for mobile application development. In: *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*. New York, NY, USA: Association for Computing Machinery, 2010. (FoSER '10), p. 397–400. ISBN 9781450304276. Disponível em: <<https://doi.org/10.1145/1882362.1882443>>. Acesso em: 14 nov. 2021. Citado 2 vezes nas páginas 23 e 24.

WHO. Global perspectives on assistive technology: proceedings of the great consultation 2019, world health organization, geneva, switzerland, 22–23 august 2019. volume 1. World Health Organization, v. 2, 2019. Disponível em: <<https://apps.who.int/iris/handle/10665/330372>>. Acesso em: 14 nov. 2021. Citado na página 20.

WHO. *World report on vision*. World Health Organization, 2019. v. 214. Disponível em: <<https://www.who.int/publications/i/item/9789241516570>>. Acesso em: 14 nov. 2021. Citado 2 vezes nas páginas 15 e 19.

YAN, S.; RAMACHANDRAN, P. G. The current status of accessibility in mobile apps. *ACM Transactions on Accessible Computing*, Association for Computing Machinery, v. 12, 2 2019. ISSN 19367228. Disponível em: <<https://dl.acm.org/doi/10.1145/3300176>>. Acesso em: 14 nov. 2021. Citado 2 vezes nas páginas 16 e 23.

YI, Y. J. Compliance of section 508 in public library systems with the largest percentage of underserved populations. *Government Information Quarterly*, v. 32, n. 1, p. 75–81, 2015. ISSN 0740-624X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0740624X14001610>>. Acesso em: 14 nov. 2021. Citado na página 20.

ZAMMETTI, F. *Practical Flutter: Improve your Mobile Development with Google's Latest Open-Source SDK*. Apress, 2019. ISBN 9781484249727. Disponível em: <<https://link.springer.com/book/10.1007/978-1-4842-4972-7>>. Acesso em: 14 nov. 2021. Citado 2 vezes nas páginas 24 e 25.