

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação



Tese

Um Arcabouço de Referência para Concepção de Soluções de Tecnologia Assistiva de Alto Nível

Rafael Cunha Cardoso

Pelotas, 2019

Rafael Cunha Cardoso

Um Arcabouço de Referência para Concepção de Soluções de Tecnologia Assistiva de Alto Nível

Tese apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Doutor em Ciência da Computação

Orientadora: Prof^a. Dr^a. Tatiana Aires Tavares

Pelotas, 2019

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

C268a Cardoso, Rafael Cunha

Um arcabouço de referência para concepção de soluções de tecnologia assistiva de alto nível / Rafael Cunha Cardoso ; Tatiana Aires Tavares, orientadora. — Pelotas, 2019.

175 f. : il.

Tese (Doutorado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2019.

1. Tecnologia assistiva. 2. Engenharia de software. 3. Interação humano-computador. 4. Arquitetura de software. 5. Iom. I. Tavares, Tatiana Aires, orient. II. Título.

CDD : 005

A todos que me inspiraram nesta jornada.

AGRADECIMENTOS

Agradeço primeiramente a minha família e meus amigos que me incentivaram de alguma forma durante toda essa caminhada. Vocês foram parte fundamental dessa jornada. O seu incentivo, apoio, paciência e compreensão e ajuda nos momentos mais difíceis, tornaram a trajetória desse trabalho mais amena. Em especial, obrigado a Bianca, minha revisora final da tese.

Aos colegas/amigos que acompanharam o "drama" do doutoramento bem de perto: Vini e Tauã, pela parceria, desabaços, churrascos e cervejas ao longo de todo esse árduo processo. Déia e Marcelo, que me ajudaram de forma crucial durante períodos de turbulência em nossa empreitada no velho continente. Obrigado Andréia, tanto pelas chamadas de atenção que ajustavam meu rumo, quanto pela parceria nos últimos meses agitados da tese.

Agradeço também às instituições que colaboraram para minha formação: Ao IFSul, por possibilitar o afastamento de minhas atividades docentes para me dedicar exclusivamente ao doutorado. Ao PPGC da UFPel, responsável direto por esse processo. E à Universidade de Aveiro, em Portugal, que me possibilitou uma experiência acadêmica e pessoal inesquecível durante o estágio sanduíche.

Obrigado a todos os professores que fizeram parte de toda este processo. Na UFPel, agradecimento especial aos professores Adenauer e Guilherem que contribuíram substancialmente com críticas e ideias para efetivação deste trabalho. Na UA, muito obrigado aos professores Telmo e Rita por terem me recebido de braços abertos em Portugal. E ao professor Fernando, meu orientador de mestrado na UFPE, por mais uma vez ter participado de uma parte tão importante da minha formação acadêmica.

A todo time WeTech pelo apoio nas atividades que desenvolvemos, com comprometimento, risadas e troca de conhecimento. Obrigado ao professor Márcio, que me possibilitou participar do projeto IOM, e a todos alunos que contribuíram no processo: Jamir, Adriano, Matheus, Krishna, Luan, Patrick e todos os demais. Em terras alémmar, um super agradecimento a toda equipe do DigiMedia, que me ajudou muito durante o meu período em Aveiro. Muito obrigado pela parceria.

Por fim, meu agradecimento especial para a minha orientadora, Tati, que me acompanhou nesta aventura, sempre me incentivando e acreditando no meu potencial! Obrigado pela paciência, empenho, carinho, confiança e dedicação ao longo de todos esses anos!

*"It's not a faith in technology.
It's faith in people."
— STEVE JOBS*

RESUMO

CARDOSO, Rafael Cunha. **Um Arcabouço de Referência para Concepção de Soluções de Tecnologia Assistiva de Alto Nível**. 2019. 175 f. Tese (Doutorado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2019.

Estima-se que em 2018 existam no mundo cerca de um bilhão de pessoas com algum tipo de limitação física ou cognitiva. Neste cenário, o provimento de soluções que simplifiquem e auxiliem a inclusão destas pessoas se torna uma questão muito relevante e desafiadora. Em especial, isto ocorre quando as soluções envolvem o desenvolvimento de software e hardware específicos, como é o caso da Tecnologia Assistiva (TA) de alto nível. Por exemplo, o dispositivo IOM (Interface Óculos Mouse) provê uma forma alternativa para que pessoas com movimentos da cabeça preservados utilizem computadores por meio de movimentos da cabeça. No entanto, a concepção desse tipo de solução acaba sendo dificultada, já que é preciso conhecer profundamente o funcionamento destas TA. Neste sentido, o desenvolvimento de estruturas de software reutilizáveis pode beneficiar este processo, pois, viabilizam a representação de diferentes aplicações, por meio da abstração das principais características e propriedades de um domínio particular. Dessa forma, é possível atender a demandas específicas presentes na construção de soluções assistivas. Este trabalho destaca o processo de concepção e um modelo reutilizável de software orientado a TA. Os elementos essenciais que compõem este modelo foram identificados a partir de duas etapas metodológicas. A primeira, de natureza teórica, consistiu em um mapeamento sistemático de literatura, o qual permitiu a caracterização do estado da arte da área de pesquisa, e por um estudo sobre dispositivos de interação consolidados no mercado. A etapa prática da metodologia, por sua vez, foi composta por quatro ciclos de desenvolvimento de aplicações utilizando o objeto de estudo (IOM) em diferentes contextos de uso. Os resultados oriundos destas etapas serviram de insumo para o projeto e desenvolvimento de um arcabouço de software para dar suporte a criação de soluções de TA de alto nível, mais especificamente, envolvendo o dispositivo IOM. Adicionalmente, a experiência construtiva permitiu evidenciar características inerentes ao processo de TA como um todo, possibilitando a ampliação do modelo desenvolvido para outros cenários.

Palavras-Chave: tecnologia assistiva; engenharia de software; interação humano-computador; arquitetura de software; processo de desenvolvimento; arcabouços; iom

ABSTRACT

CARDOSO, Rafael Cunha. ***A Reference Framework for Designing High-Level Assistive Technology Solutions***. 2019. 175 f. Tese (Doutorado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2019.

It is estimated that by 2018, there are around one billion people in the world with some physical or cognitive limitation. In this scenario, providing solutions that simplify and assist the inclusion of these people in the society becomes a very relevant and challenging issue. This challenge is especially real when solutions of this nature involve the development of specific software and hardware, as in the case of high-level Assistive Technology (AT). For example, the IOM (acronym for Interface Óculos Mouse, in Portuguese) device provides an alternative way for persons with preserved head movements to using computers through head movements. However, the process of developing this type of solution is complicated, since it is necessary to know the functioning of these ATs deeply. In this sense, the development of reusable software frameworks can benefit this process. They make it possible, through the abstraction of the main characteristics and properties of a specific domain, the representation of different applications. Thus, it is possible to meet the particular demands present in the construction of assistive solutions. This work highlights the design process and a reusable TA-oriented software model. The essential elements that make up this model were identified from two methodological steps. The first, with a theoretical nature, consisted of a systematic literature mapping, which allowed the characterization of the research area's state of the art, and a study on interaction devices consolidated in the market. The practical step of the methodology, in turn, was composed of four application development cycles using the object of study (IOM) in different contexts of use. The results from these two steps served as inputs for designing and developing a software framework to support the development of high-level AT solutions, more specifically involving the IOM device. Additionally, the valuable experience allowed us to highlight characteristics inherent to the AT process as a whole, providing the expansion of the developed model to other scenarios.

Key-words: *assistive technology; software engineering; human-computer interaction; software architecture; frameworks; iom*

LISTA DE FIGURAS

1	População brasileira que declarou algum tipo de deficiência física.	17
2	Etapas previstas na metodologia de pesquisa.	24
3	Metodologia de desenvolvimento.	25
4	Exemplo de diagrama de Contexto, segundo o modelo C4.	37
5	Exemplo de diagrama de Contêineres, segundo o modelo C4.	38
6	Exemplo de diagrama de Componentes, segundo o modelo C4.	39
7	Exemplo de diagrama de Classes, segundo o modelo C4.	40
8	Processo de software genérico.	43
9	Técnicas de reutilização.	44
10	Padrões para desenvolvimento de Arcabouços.	46
11	Etapas do processo de desenvolvimento de um produto de TA.	48
12	Fluxo de execução do MSL.	58
13	Gráficos sobre os artigos recuperados no MSL.	60
14	Artigos caracterizados de acordo com eixo de pesquisa.	61
15	Esquema conceitual do AsTeRICS.	62
16	Arquitetura do AsTeRICS.	63
17	PS3 adaptado.	64
18	<i>Switch Trainer</i> , jogo criado a partir do ATLab.	65
19	Jogos <i>Ladybugs</i> e <i>Invaders</i> , criados a partir do GNomon.	66
20	Arquitetura em camadas do STTK.	67
21	Arquitetura de contexto do FHCI.	67
22	Arquitetura do FHCI.	68
23	ETM.	69
24	Modelo conceitual da TA Universal.	69
25	Arquitetura em camadas da TA universal.	70
26	Componentes da arquitetura proposta.	71
27	Arquitetura do MWiMT.	72
28	Operação BCI com o microFenix.	74
29	Interface para controle de dispositivos em um ambiente assistivo <i>wireless</i>	75
30	Diagrama de componentes do SBC.	76
31	Arquitetura do hBCI.	77
32	Diagrama de implementação para controle do robô via sinais BCI.	78
33	Diagrama de contexto do sistema.	79
34	Visão geral do AID.	80
35	Placa de comunicação.	81

36	Diagrama de componentes do jogo.	82
37	Trabalhos do MSL, distribuídos por funcionalidades.	85
38	Ciclos de desenvolvimento previstos.	98
39	Diagrama de contexto do IOM.	99
40	Fluxograma de funcionamento do <i>firmware</i> do IOM.	101
41	GUI do SCCIOM.	102
42	Arquitetura do SCCIOM.	103
43	Interface de configuração do IOM.	104
44	Dispositivo tangível <i>Sphero</i> , versão 1.0.	105
45	Doce Labirinto.	106
46	Arquitetura do Doce Labirinto.	107
47	Painel de gerenciamento do Doce Labirinto.	108
48	Resultado da avaliação do Doce Labirinto.	109
49	Contexto de uso do IOM4Home.	110
50	Sprints de desenvolvimento do IOM4Home.	111
51	Arquitetura do IOM4Home.	111
52	Interface do <i>IOM4Home</i> , primeira versão.	112
53	Interface do <i>IOM4Home</i> , segunda versão.	113
54	Resultado da avaliação do IOM4Home.	113
55	Arquitetura da plataforma +TV4E.	115
56	Arquitetura do IOM4TV.	116
57	Protótipos do IOM4TV.	117
58	Resultados do SAM. Em laranja, as respostas do primeiro experi- mento. Em verde, as respostas referentes ao segundo experimento.	118
59	Resultado das avaliações do IOM4TV.	118
60	Interface do protótipo baseado em eventos	119
61	Modelagem corrente do arcabouço para o IOM.	128
62	Diagrama de componentes do arcabouço para o IOM.	130
63	Diagrama de contêineres para o IOM.	132
64	Modelo de conceitual ampliado.	137

LISTA DE TABELAS

1	Comparação entre Arquitetura e AS.	35
2	Tipos de arquiteturas de software mais comuns.	36
3	Critérios de exclusão e inclusão utilizados no MSL.	60
4	Trabalhos caracterizados como arcabouços no MSL.	84
5	Trabalhos agrupados segundo classificação por <i>Funcionalidade</i> . . .	86
6	Característica dos protótipos estudados no MSL.	86
7	Trabalhos agrupados por tipo de representação.	89
8	Características básicas dos dispositivos estudados.	94
9	Funcionalidades encontradas nos arcabouços dos dispositivos. . .	94
10	Comparação entre as funcionalidades das aplicações, com os métodos encontrados nas soluções comerciais.	121
11	Aspectos funcionais percebidos nas prototipações desenvolvidas. .	126
12	Funcionalidades essenciais presentes no arcabouço.	127
13	Descrição das funcionalidades do Protocolo de comunicação. . . .	131

LISTA DE ABREVIATURAS E SIGLAS

ALL	<i>Ambient Assisted Living</i>
ACS	<i>AsTeRICS Configuration Suite</i>
ADA	<i>American Disabilities Act</i>
AID	<i>Adaptive Interface Design</i>
API	<i>Application Programming Interface</i>
ARE	<i>AsTeRICS Runtime Environment</i>
AS	Arquitetura de Software
AsTeRICS	<i>Assistive Technology Rapid Integration & Construction Set</i>
ATACP	<i>Assistive Technology Applications Certificate Program</i>
ATLab	<i>Assistive Technology Laboratory</i>
AU	Acessibilidade Universal
AVSR	<i>Audio-visual Russian Speech</i>
BCI	<i>Brain Computer Interface</i>
BNCI	<i>Brain Neuronal Computer Interfaces</i>
CBSE	<i>Component-Based Software Engineering</i>
CVDS	Ciclo de Vida de Desenvolvimento de Sistemas
dTDS	<i>dual-mode Tongue Drive System</i>
ED	Engenharia de Domínio
EA	Engenharia de Aplicação
EEG	Eletroencefalografia
ELA	Esclerose Lateral Amiotrófica
EMG	Eletromiografia
EOG	Eletro-oculografia
ERD	<i>Event Related Desynchronization</i>
ERS	<i>Event Related Synchronization</i>
ES	Engenharia de Software

ETM	<i>Eye Tracking Mouse</i>
EUA	Estados Unidos da América
EUSTAT	<i>Empowering Users Through Assistive Technology</i>
FHCI	<i>Facial Human-Computer Interface</i>
GUI	<i>Graphical User Interfaces</i>
hBCI	<i>Híbrid Brain Neuronal Computer Interfaces</i>
HEART	<i>Horizontal European Activities in Rehabilitation Technology</i>
HID	<i>Human Interface Devices</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
ICANDO	<i>Intellectual Computer AssistaNt for Disabled Operators</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IFSul	Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense
IGU	Interface Gráfica de Usuário
IHC	Interação Humano-Computador
IOM	Interface Óculos-Mouse
IoT	<i>Internet of Things</i>
ISO	<i>International Organization for Standardization</i>
iTV	<i>Interactive TV (TV Interativa)</i>
jSSC	<i>Java Simple Serial Connector</i>
JVM	<i>Java Virtual Machine</i>
LPS	Linha de Produtos de Software
M2M	<i>Machine To Machine</i>
MEMS	<i>Micro-Electro-Mechanical Systems</i>
MSL	Mapeamento Sistemático de Literatura
MVC	<i>Model-View-Controller</i>
MWiMT	<i>My-World-in-My-Tablet</i>
NUI	<i>Natural User Interfaces</i>
OSGi	<i>Open Services Gateway Initiative</i>
OSKA	<i>On-Screen Keyboard Application</i>
PS3	<i>PlayStation 3</i>
RDP	<i>Remote Desktop Protocol</i>
RPi	<i>Raspberry Pi</i>
SaaS	<i>Software as a Service</i>
SAM	<i>Self-Assessment Manikin</i>

SBC	<i>Single Board Computer</i>
SCCIOM	Software de Configuração e Calibração do IOM
SDK	<i>Software Development Kit</i>
SO	Sistema Operacional
SOA	<i>Service-Oriented Architecture</i>
SSD	Sistemas de Suporte a Decisões
SSVEP	<i>Steady-State Visual Evoked Potentials</i>
STB	<i>Set-Top Box</i>
STT	<i>Speech-to-Text</i>
STTK	<i>Speech-To-Text plus Kinect</i>
TA	Tecnologia Assistiva
TCC	Trabalho de Conclusão de Curso
TDS	<i>Tongue Drive System</i>
TI	Tecnologia da Informação
TIDE	<i>Technology Initiative for Disabled and Elderly People</i>
TTS	<i>Text-to-Speech</i>
TTY	<i>Teletypewriter</i>
TUI	<i>Tangible User Interfaces</i>
UA	Universidade de Aveiro
UFPeI	Universidade Federal de Pelotas
UDP	<i>User Datagram Protocol</i>
UML	<i>Unified Modeling Language</i>
VIM	<i>Virtual Infrastructure Manager</i>
VM	<i>Virtual Machine</i>
VOIP	<i>Voice over IP</i>
WHO	<i>World Health Organization</i>

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Motivação	20
1.2	Questão de Investigação e Hipóteses	22
1.3	Metodologia de Trabalho	23
1.3.1	Método de Pesquisa	23
1.3.2	Método de Desenvolvimento	25
1.3.3	Estudo de Caso	26
1.4	Escopo	26
1.5	Contribuições	27
1.6	Organização da Tese	32
2	FUNDAMENTAÇÃO CONCEITUAL	34
2.1	Aspectos Construtivos para Desenvolvimento de Software	34
2.1.1	Arquitetura de Software (AS)	35
2.1.2	Prototipação	40
2.1.3	Processo de Software	42
2.1.4	Técnicas de Reutilização de Software	43
2.1.5	Desenvolvimento de software para TA	48
2.2	Tecnologia Assistiva (TA)	49
2.2.1	Nível de Tecnologia	50
2.2.2	Norma Internacional ISO 9999	51
2.2.3	HEART (<i>Horizontal European Activities in Rehabilitation Technology</i>)	52
2.2.4	ATACP (<i>Assistive Technology Applications Certificate Program</i>)	53
2.3	Considerações Finais do Capítulo	54
3	ESTADO DA ARTE	57
3.1	Mapeamento Sistemático de Literatura (MSL)	57
3.1.1	Análise Exploratória Inicial	57
3.1.2	Preenchimento do Protocolo MSL	58
3.1.3	Execução do protocolo MSL	60
3.2	Trabalhos Recuperados no MSL	61
3.2.1	AsTeRICS (<i>Assistive Technology Rapid Integration & Construction Set</i>)	62
3.2.2	ATLab (<i>Assistive Technology Laboratory</i>)	64
3.2.3	GNomon	65
3.2.4	STTK (<i>Speech-To-Text plus Kinect</i>)	66
3.2.5	FHCI (<i>Facial Human-Computer Interface</i>)	67
3.2.6	ETM (<i>Eye Tracking Mouse</i>)	68

3.2.7	<i>Universal AT with Multimodal Input and Multimedia Output Interfaces</i>	69
3.2.8	<i>Cloud-based AT</i>	71
3.2.9	<i>My-World-in-My-Tablet (MWiMT)</i>	72
3.2.10	Protótipo multimodal	73
3.2.11	<i>dTDS (dual-mode Tongue Drive System)</i>	73
3.2.12	<i>DOSVOX Gyro & microFênix</i>	73
3.2.13	<i>UserTracking</i>	74
3.2.14	WPan	75
3.2.15	<i>SBC (Single Board Computer)</i>	75
3.2.16	<i>hBCI (Hybrid Brain Computer Interface)</i>	76
3.2.17	SMARTUNIVERS	77
3.2.18	<i>Fire Bird V</i>	78
3.2.19	<i>Human-Computer Interface Based On Iot Embedded Systems</i>	78
3.2.20	<i>AID (Adaptive Interface Design)</i>	79
3.2.21	<i>Gaze-Based Interaction System for People with Cerebral Palsy</i>	80
3.2.22	Beto na Floresta	81
3.2.23	SINASense	82
3.2.24	ActiveIris	83
3.2.25	Doce Labirinto	83
3.3	Taxonomia para Soluções voltadas a TA	84
3.3.1	Classificação por Tipo de Solução Desenvolvida	84
3.3.2	Classificação por Funcionalidades das Soluções	85
3.4	Discussão dos resultados do MSL	88
3.5	Análise das Questões de Pesquisa - MSL	91
3.6	Estudo de Soluções Comerciais	92
3.7	Considerações Finais do Capítulo	95
4	PROTOTIPAÇÃO DE APLICAÇÕES UTILIZANDO O IOM	97
4.1	Metodologia de Desenvolvimento	97
4.2	IOM (Interface Óculos Mouse)	99
4.2.1	Antecedentes	99
4.2.2	Especificações	100
4.3	Aplicações Desenvolvidas	102
4.3.1	SCCIOM	103
4.3.2	Doce Labirinto	105
4.3.3	IOM4Home	110
4.3.4	IOM4TV	115
4.4	Lições Aprendidas	120
5	DESENVOLVIMENTO DE UM ARCABOUÇO PARA O IOM	123
5.1	Requisitos	124
5.2	Processo de Modelagem	125
5.3	Arcaçoço para o dispositivo de interação IOM	126
5.3.1	Visão de Implementação	127
5.3.2	Diagramas de Alto Nível	130
5.3.3	Avaliação	133
5.4	Aspectos Construtivos para Desenvolvimento de TA	134
5.4.1	Aderência a padrões de projeto	135
5.4.2	Recomendações para aplicações de controle.	135

5.4.3	Modelo Ampliado para Construção de TA	137
6	CONCLUSÃO	140
6.1	Discussão das Hipóteses	142
6.2	Discussão: Questão de Investigação	144
6.3	Limitações	147
6.4	Contribuições	148
6.5	Trabalhos Futuros	150
	REFERÊNCIAS	152
	ANEXO A PROTOCOLO MSL	163
	ANEXO B CERTIFICADO DE ESTÁGIO DE PÓS-GRADUAÇÃO NA UNIVER- SIDADE DE AVEIRO, EM PORTUGAL	174

1 INTRODUÇÃO

Os avanços em diversas áreas de conhecimento tecnológicas permitem que seja experimentado um aumento no desenvolvimento de aplicações, serviços e dispositivos voltados aos mais diversos públicos e finalidades. Dentre os perfis de usuários beneficiados neste cenário, o formado por pessoas com limitações físicas compõe um universo que pode ser diretamente favorecido. De acordo com o relatório produzido pela *World Health Organization (WHO)*, mais de um bilhão de pessoas possui algum tipo de deficiência em todo o mundo (WHO, 2017).

Especificamente no Brasil, segundo dados do censo realizado em 2010 pelo Instituto Brasileiro de Geografia e Estatística (IBGE), 23,9% da população brasileira declarou ter ao menos um tipo de deficiência (IBGE, 2010). Deste total, 7% especificamente relatam alguma forma de incapacidade motora, variando desde restrições leves, como tremores ou falta de mobilidade de um membro do corpo, até problemas mais severos, tais como esclerose lateral amiotrófica (ELA) ou tetraplegia. A Figura 1 evidencia as porcentagens da população brasileira agrupada por tipos de deficiência.

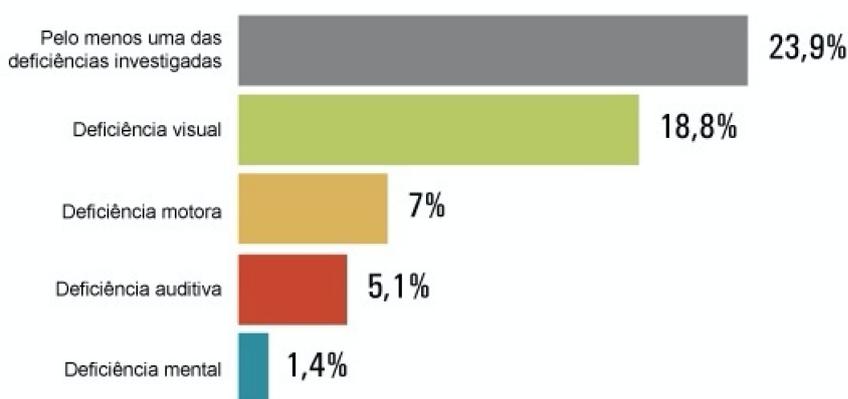


Figura 1 – População brasileira que declarou algum tipo de deficiência física.
Fonte: IBGE (2010).

Estes números revelam a existência de uma parcela considerável da população que fica muitas vezes impossibilitada de realizar tarefas cotidianas adequadamente, tais como se alimentar, comunicar, fazer higiene pessoal, trabalhar ou se locomover. Neste cenário, a Ciência da Computação pode colaborar para a inclusão de pessoas

com deficiência e idosos, por meio da oferta de soluções voltadas à interação deste público com seu meio social (COSTA, 2018).

Especificamente, a Tecnologia Assistiva (TA) é a área de pesquisa que procura propor, projetar, desenvolver e avaliar dispositivos e aplicações que melhorem a qualidade de vida das pessoas com limitações físicas ou cognitivas, provendo maior autonomia e independência a elas (PENNSYLVANIA, 2012). Uma definição que permite compreender a importância das soluções assistivas no cotidiano deste público-alvo é feita por RADABAUGH (1993):

“Para as pessoas sem deficiência a tecnologia torna as coisas mais fáceis. Para as pessoas com deficiência, a tecnologia torna as coisas possíveis”.

TA é uma expressão genérica que abrange produtos, tecnologias, serviços e aplicações utilizados por deficientes ou idosos, permitindo a realização de atividades que seriam difíceis, perigosas ou impossíveis de outra forma (HERSH, 2010). É uma área de natureza multidisciplinar, a qual trata da pesquisa e concepção de instrumentos que promovem a inclusão das pessoas com limitações físicas ou cognitivas na sociedade.

Pesquisas relacionadas ao desenvolvimento de TA vêm conquistando espaço, produzindo soluções que auxiliam ou simplificam a vida das pessoas. Próteses, cadeiras de rodas, óculos ópticos, legendas em televisores, aparelhos auxiliares para higiene, robôs assistentes, ferramentas para acesso ao computador, entre outros, são exemplos de produtos com natureza assistiva.

Esta ampla variedade de soluções produzidas motivou o surgimento de diversas classificações para categorizar os tipos de TA. De acordo com SEABRA; MENDES (2009 apud LAUAND, 2005), uma delas leva em consideração o custo e o funcionamento dos recursos adotados em seu desenvolvimento, dividindo-os em três níveis:

- TA de Baixo Nível - dispositivos simples, não elétricos e de baixo custo;
- TA de Médio Nível - equipamentos com algum tipo de fonte de alimentação, mas que não utilizam recursos computacionais; e
- TA de Alto Nível - soluções mais sofisticadas que utilizam software e/ou hardware específicos para sua operação.

TA de Baixo Nível são aparatos mais simples, podendo ser produzidos até artesanalmente, como instrumentos adaptados para higiene e alimentação. Também se encaixam nesta classe utensílios como óculos, lupas ou próteses. TA de Médio Nível, por sua vez, envolvem equipamentos que utilizam alguma fonte de alimentação em seu funcionamento. Nesta categoria se encontram aparatos como cadeiras de rodas ou utensílios assistivos pessoais alimentados por baterias, por exemplo.

Por fim, as TA de Alto Nível são soluções mais sofisticadas, caracterizadas pela utilização de recursos computacionais na sua concepção. Dentre os objetivos que podem ser alcançados com a criação desta classe de TA, estão soluções que investem, por exemplo, no desenvolvimento de jogos assistivos, seja para fins de entretenimento ou reabilitação de seus usuários (SCARDOVELLI; FRÈRE, 2015; LIN et al., 2017; CARDOSO et al., 2016). Outras aplicações, por sua vez, exploram questões relacionadas ao controle de ambiente, buscando fornecer maior autonomia a seus usuários em suas residências ou locais de trabalho (PAUL et al., 2013; TRIPATHY; RAHEJA, 2014).

Existem ainda soluções para fornecer acesso alternativo a computadores para pessoas com diferentes tipos de limitações físicas ou cognitivas (ANTUNES et al., 2016; LUPU; UNGUREANU; SIRITEANU, 2013; RANTANEN et al., 2011; CRUZ et al., 2015; GALANTE; MENEZES, 2012). Nesta última categoria, encontra-se em desenvolvimento o projeto Interface Óculos-Mouse (IOM). Concebido para ser uma solução de baixo custo, este trabalho visa a construção de um dispositivo para interação alternativa, voltado para pessoas com movimentos da cabeça preservados (MACHADO, 2010; RODRIGUES et al., 2016; MACHADO et al., 2019).

Pesquisas como as supracitadas, especificamente relacionadas ao desenvolvimento de TA de Alto Nível, vêm ampliando seu espaço, produzindo dispositivos, aplicações e serviços que auxiliam ou simplificam a vida das pessoas. Tornam-se muito relevantes especialmente em tempos nos quais a Internet, e suas aplicações, são fundamentais para o desenvolvimento das interações humanas em suas diversas extensões (MEDEIROS, 2013).

O processo de criação de uma TA¹ que utiliza recursos computacionais em sua concepção, usualmente reúne diversos campos de estudo. A mescla de conceitos provenientes de linhas de pesquisa como Acessibilidade Universal (AU), Sistemas de Suporte a Decisões (SSD), *Ambient Assisted Living* (AAL) e Interação Humano-Computador (IHC) potencializam o desenvolvimento de TA para melhorar a vida das pessoas com alguma limitação física/cognitiva (ALONSO, 2015). Em comum, estes campos de estudo exercitam um processo de desenvolvimento de software cujo produto principal é a própria aplicação desenvolvida.

Considerando este cenário, o presente trabalho inicialmente investiga as principais soluções baseadas em TA encontradas na literatura, com o intuito de identificar e analisar os aspectos construtivos de software utilizados durante os seus desenvolvimentos. Neste sentido, são analisadas questões relacionadas tanto às arquiteturas de software quanto aos processos de modelagem, desenvolvimento e avaliação adotadas nas soluções encontradas na literatura. A partir destas percepções, a a ideia é

¹A partir deste momento sempre que o acrônimo TA aparecer no texto, sem explicitamente definir o nível, o termo se refere a TA de Alto Nível, foco desta tese.

aplicar técnicas de desenvolvimento, sobre um domínio específico de conhecimento, de forma que seja possível propor uma estrutura reutilizável de software, aplicável a diferentes contextos de utilização de um dispositivo de interação.

Nesta perspectiva, o processo de construção de uma TA se torna um grande desafio, uma vez que os dispositivos de interação usualmente são adaptados às necessidades únicas dos usuários, focando no aprimoramento de suas habilidades preservadas. Este panorama favorece a elaboração de soluções pontuais, não preocupadas com reaproveitamento de conhecimento ou artefatos utilizados previamente no projeto e concepção de alguma TA desenvolvida.

1.1 Motivação

Segundo a IEEE (*Institute of Electrical and Electronic Engineers*) a Engenharia de Software (ES) pode ser definida como (RADATZ; GERACI; KATKI, 1990) apud (PRES-SMAN; MAXIM, 2016):

"A aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software, isto é, a aplicação de engenharia ao software."

A ES dispõe de diferentes técnicas que podem ser utilizadas para o desenvolvimento de soluções de software. Uma delas se concentra na reutilização de componentes previamente produzidos. Esta abordagem possui duas particularidades intimamente relacionadas: "construção para reutilização" e "construção com reutilização". A primeira é relacionada com a criação de componentes reutilizáveis, enquanto a segunda foca em utilizar tais componentes na construção de uma nova solução (BOURQUE; FAIRLEY et al., 2014). Assim, seu alicerce fundamental é o conceito de reutilização, processo pelo qual a criação de aplicações acontece a partir de um software pré-existente. A adoção desta prática para a concepção de produtos com necessidades semelhantes, além de simplificar o processo, gera um menor custo (MATEEN; KAUSAR; SATTAR, 2017). Essa característica acaba se tornando um elemento de vantagem competitiva dentro da perspectiva da ES.

Para propiciar a reutilização é necessário prover abstrações das principais propriedades e funcionalidades de um determinado domínio de conhecimento. Isto pode ser conseguido a partir de estruturas reutilizáveis, ou arcabouços de software. O projeto e desenvolvimento de um arcabouço de software envolve a aplicação de algumas abordagens de ES. Parte destes métodos estimula o desenvolvimento de uma série de aplicações, em um domínio particular, mas sob contextos de utilização diferentes (ROBERTS; JOHNSON, 1996; LOPEZ-HERREJON; LINSBAUER; EGYED, 2015). A finalidade é identificar propriedades e funcionalidades recorrentes que devem compor uma estrutura reutilizável mínima, fundamentada em uma família de aplicações.

Em contrapartida a estes preceitos, encontra-se o cenário de concepção de TA. O processo de modelagem, desenvolvimento e avaliação de uma TA é desafiador, uma vez que as soluções propostas frequentemente são adaptações às necessidades particulares de grupos de pessoas, ou ainda, de um único indivíduo. Estas aplicações de TA visam potencializar as capacidades preservadas, de forma a amenizar as dificuldades enfrentadas por estas pessoas na realização de atividades cotidianas (COOK; POLGAR, 2014). Neste contexto, diferentes frentes de atuação podem ser exploradas, tais como prototipações específicas, adaptações metodológicas, ou mesmo a criação de estruturas para desenvolvimento de soluções assistivas (PINHEIRO JR, 2013; OLIVEIRA et al., 2015; OKUMURA, 2017; SOUSA, 2017; ALVES, 2018).

Este trabalho aborda esta problemática, adotando estratégias de ES, em cenários de aplicação diversificados, no contexto de TA. O objetivo é propor um modelo de estrutura reutilizável de software, para um domínio particular. Um projeto que pode explorar um desenvolvimento desta natureza é o IOM. Conforme será evidenciado, o IOM é um dispositivo de interação alternativa cujo objetivo primordial é permitir o controle de apontadores, a partir de movimentos da cabeça.

Apesar de ter sido concebido inicialmente para controle de computador, a evolução de estudos sobre este dispositivo vislumbra a ampliação de sua variedade de aplicabilidade. No entanto, por se tratar de um protótipo, a criação de aplicações que o utilizem em outros contextos torna-se difícil, pois é necessário um conhecimento razoável de seu funcionamento para projetar soluções voltadas a ele. Desta forma, a adoção do IOM como estudo de caso é pertinente, pois permitirá atingir duas finalidades:

1. Simplificar a utilização do IOM em diferentes contextos, por meio do provimento de estruturas de software reutilizáveis próprias; e
2. Identificar subsídios e requisitos que auxiliem a responder à questão de investigação desta tese, confirmando ou não as hipóteses levantadas.

Contudo, a criação de aplicações de TA é especialmente complexa uma vez que deve considerar, além das questões técnicas, os aspectos psicológicos e de bem-estar dos usuários finais. Além disso, diferentemente de um processo de desenvolvimento convencional, o qual visa alcançar o maior número de usuários possível, uma solução de TA pode ser utilizada para ampliar as habilidades preservadas de um único indivíduo (COOK; POLGAR, 2014). Ainda, as soluções de mercado existentes usualmente possuem um alto custo associado, não permitindo que uma parcela considerável tenha acesso às aplicações ou equipamentos de que necessitam (MACHADO et al., 2019). Desta forma, este processo apresenta-se como um grande desafio, uma vez que lida com inúmeras questões, muitas das quais ultrapassam o âmbito técnico.

1.2 Questão de Investigação e Hipóteses

O desenvolvimento de estruturas de software reutilizáveis é derivado a partir do levantamento de elementos de diversas naturezas. Requisitos funcionais, não funcionais, arquiteturais e tecnológicos devem ser considerados, para originar um modelo conceitual adequado, que abrigue as principais funcionalidades e características que devem estar presentes em uma solução deste tipo. Em especial, a elicitação dos requisitos arquiteturais de uma solução orienta a proposição de um modelo conceitual reutilizável dentro de um domínio específico de aplicação. Neste contexto, esta tese de doutorado é norteada pela seguinte questão de investigação:

“Que requisitos arquiteturais devem ser considerados na concepção de estruturas de software reutilizáveis, que possam ser utilizados no desenvolvimento de soluções de TA, voltadas a pessoas com capacidade de movimentação da cabeça preservada, em diversos contextos?”

Esta pergunta foi definida com o amadurecimento do trabalho, considerando uma série de diretivas e princípios definidos por QUIVY; VAN CAMPENHOUDT (1998). Segundo é percebido na questão, o objetivo final é elencar uma série de elementos arquiteturais, ou seja, aspectos de construção de software, que devem ser considerados no desenvolvimento de soluções assistivas. Ainda, com o intuito de restringir o foco de atuação, o trabalho define como seu domínio específico de aplicação, soluções voltadas a usuários com movimentos preservados de cabeça.

Como é detalhado no Capítulo 3, a maior parte das aplicações de TA disponíveis são protótipos desenvolvidos para atender demandas específicas. Geralmente são adaptações de dispositivos consolidados no mercado para atingir uma finalidade particular. Este modo de desenvolvimento não propicia o reaproveitamento sistemático dos modelos ou estruturas de software utilizadas. Nesse contexto, um dos objetivos desta tese é discutir aspectos construtivos relacionados ao desenvolvimento de soluções reutilizáveis voltadas a TA. Diante do exposto, foram delineadas algumas hipóteses, a serem confirmadas ao final desta tese:

- *Hipótese 1* - Processos para desenvolvimento de software convencionais atendem as necessidades para a criação de soluções de TA.
- *Hipótese 2* - A concepção de soluções assistivas reutilizáveis é simplificada com a identificação de elementos recorrentes em aplicações dentro de domínio específico, mesmo em diferentes contextos de utilização.
- *Hipótese 3* - Obstáculos tecnológicos de desenvolvimento impactam diretamente no suporte aos diferentes perfis de usuários finais.

Com o intuito de verificar estas hipóteses e responder à questão de investigação primordial deste trabalho, foi organizado um conjunto de etapas metodológicas a serem cumpridas. Cada um destes passos é detalhado na próxima seção.

1.3 Metodologia de Trabalho

Conforme destacado anteriormente, este trabalho investiga as principais soluções baseadas em TA existentes, visando identificar e analisar as particularidades construtivas, em termos de software, utilizados em suas concepções. A partir desta compreensão geral planeja-se o desenvolvimento de soluções voltadas a TA, em cenários de utilização diversificados, que permitam alcançar o **objetivo principal** do trabalho, ou seja, a proposição de uma estrutura de software reutilizável para cobrir um domínio específico de aplicação.

Visando atingir estas metas, e consequentemente responder à questão de investigação e verificar as hipóteses levantadas, foi planejada a execução de uma série de etapas metodológicas envolvendo atividades teóricas e práticas. Neste sentido, o trabalho foi dividido em três etapas:

- Método de Pesquisa;
- Método de Desenvolvimento; e
- Estudo de Caso.

As subseções a seguir destacam estas etapas complementares, especificando os processos metodológicos e as atividades inerentes a cada uma delas.

1.3.1 Método de Pesquisa

A primeira parte da metodologia diz respeito aos estudos teóricos realizados ao longo de todo o trabalho. Foi estabelecida uma série de passos complementares, os quais são destacados pela Figura 2.

Conforme pode ser observado, inicialmente foi realizado um ciclo de estudos, o qual compreendeu uma pesquisa exploratória simples em engenhos de busca, utilizando palavras-chave essenciais no contexto desta tese. O objetivo era recuperar e analisar trabalhos que abordassem temas similares ao investigado na tese. Esta etapa resultou na definição de um artigo âncora, utilizado para auxiliar o desenvolvimento do ciclo seguinte, a revisão bibliográfica sobre o *estado da arte* da área de pesquisa abordada.

Nesta fase foi conduzida a execução do Mapeamento Sistemático da Literatura (MSL). Esta metodologia prevê a execução de diversos passos para a geração do



Figura 2 – Etapas previstas na metodologia de pesquisa.

Fonte: Elaborada pelo Autor (2019).

mapa de literatura da área, ou seja, um conjunto amplo de artigos que traz uma prospecção geral do tema abordado como objeto de pesquisa (PETERSEN et al., 2008). Uma característica importante desta técnica é que todos os procedimentos devem ser documentados, permitindo que o estudo realizado possa ser reproduzível por outros pesquisadores, de forma que estes consigam alcançar resultados similares em um dado período de tempo. Assim, o objetivo deste ciclo foi identificar os principais trabalhos desenvolvidos acerca de *Soluções voltadas ao desenvolvimento de TA*, no meio acadêmico. A partir da leitura e análise dos trabalhos resultantes do MSL, foi possível examinar e extrair dados relevantes, além de perceber relações, semelhanças e diferenças entre os diversos projetos existentes na área explorada. Esses dois primeiros ciclos são detalhados na Seção 3.1.

Posteriormente ao MSL, foi realizado um levantamento qualitativo sobre as características técnicas de dispositivos de interação alternativa consolidados no mercado (TAYLOR; BOGDAN; DEVAULT, 2015). Neste passo foram analisados os arcabouços de software utilizados por soluções comerciais, com o objetivo de identificar as principais funcionalidades expostas por elas, verificando a existência de algum tipo de correlação entre elas. O objetivo era aferir possíveis similaridades, em termos de funções e propriedades presentes nestes arcabouços, complementando os resultados do MSL com este detalhamento tecnológico. Os resultados desta etapa metodológica são apresentados e discutidos na Seção 3.6.

1.3.2 Método de Desenvolvimento

Paralelamente às investigações e levantamentos teóricos, foi prevista uma etapa prática do trabalho. Seu objetivo essencial era aplicar técnicas de modelagem, desenvolvimento e avaliação de software, para criação de um conjunto de aplicações voltadas a TA.

Conforme destacado anteriormente, esta tese foca em TA voltada a pessoas com capacidade de movimentação da cabeça preservada. Esta delimitação foi necessária para restringir o escopo do trabalho. A escolha por esta limitação em si, foi suscitada já que a perda da capacidade funcional, especificamente em termos de mobilidade, é um fator natural no decorrer da vida dos indivíduos, ocorrendo gradativamente a diminuição das habilidades para desempenhar atividades cotidianas (CRUZ; EMMEL, 2012; NUNES et al., 2010). Além desta motivação, o foco foi encorajado pois, oportunamente, encontra-se em curso o projeto de desenvolvimento do IOM, um dispositivo de interação alternativa, voltado para pessoas com limitações motoras nos membros superiores, com capacidade de movimentação da cabeça preservado.

Assim, foi planejada uma série de ciclos de desenvolvimento de TA utilizando o IOM como equipamento de interação, em diferentes cenários de aplicação. A ideia é desenvolver pelo menos três aplicações para viabilizar a criação de um arcabouço de software dentro do domínio específico de conhecimento explorado (ROBERTS; JOHNSON, 1996). A Figura 3 destaca esta metodologia de desenvolvimento.



Figura 3 – Metodologia de desenvolvimento.

Fonte: Elaborada pelo Autor (2019).

O objetivo de aplicar estes ciclos de desenvolvimento é utilizar as técnicas de ES, destacadas no Capítulo 2, em diferentes cenários de uso. A ideia é reunir as percepções advindas a partir de cada ciclo, de forma a viabilizar a proposição, e consequente criação, de uma estrutura de software reutilizável para o domínio explorado, a partir do desenvolvimento sistemático de software (BOSCH, 2000; LINDEN, 2002; BAYER et al., 1999). Neste sentido, em cada um dos ciclos, foram adotadas diferentes técnicas e processos para modelar, desenvolver e avaliar as soluções, visando:

- Perceber os requisitos dos usuários no contexto da aplicação;
- Propor/modelar soluções adequadas aos diferentes cenários de utilização;

- Conceber soluções na forma de protótipos;
- Avaliar os protótipos desenvolvidos; e
- Analisar os resultados dos testes sobre cada uma das aplicações desenvolvidas.

Além das análises pontuais relacionadas às aplicações, ao final de cada ciclo eram listados os principais requisitos percebidos durante o desenvolvimento. Como destacado anteriormente, para diversificar os tipos de interações realizadas, e assim ampliar e enriquecer os dados coletados, as soluções abordam diferentes cenários de uso. Para tanto, a escolha dos contextos das aplicações desenvolvidas foi orientada pela taxonomia proposta nesta tese, a qual será destacada na Seção 3.3. Os quatro ciclos que compõem a metodologia de desenvolvimento são especificados ao longo da Seção 4.1.

1.3.3 Estudo de Caso

Por fim, o último passo previsto na metodologia deste trabalho diz respeito ao cruzamento dos dados resultantes das etapas teóricas, com as características e diretrizes extraídas a partir das prototipações realizadas no método de desenvolvimento. Assim, a partir dos elementos elencados nas etapas metodológicas anteriores, esta fase busca projetar, desenvolver e avaliar um arcabouço de software para o dispositivo IOM. Neste contexto, o estudo de caso proposto utiliza uma abordagem de desenvolvimento *bottom-up*, ou seja, a concepção da solução parte da especificação dos elementos elementares que a constituem, indo em direção a proposição de sua arquitetura geral (MARTINEZ et al., 2015).

Desta forma, o estudo de caso reúne os resultados provenientes dos passos anteriores da metodologia. Ou seja, esta etapa do trabalho adota técnicas de modelagem e implementação de estruturas de software reutilizáveis, explorados no longo do Capítulo 2. Também são abordados os principais conceitos e percepções levantados a partir dos estudos realizados no Capítulo 3, relacionados ao estado da arte da área de pesquisa. Por fim, são utilizados os elementos extraídos como consequência das prototipações previstas nos ciclos do método de desenvolvimento, descritos no Capítulo 4. A análise desta etapa é inicialmente discutida na Seção 4.4, e ampliada ao longo do Capítulo 5, com a proposição, modelagem, desenvolvimento e avaliação do arcabouço para o dispositivo de interação alternativa IOM.

1.4 Escopo

Desde o início do trabalho, foram impostas algumas delimitações à esta pesquisa. Em especial, a escolha pelo dispositivo de interação IOM para desenvolvimento dos

estudos de caso, ao invés de uma solução de mercado. Por se tratar de um protótipo em desenvolvimento, esta opção trouxe benefícios e dificuldades à efetivação do trabalho. Por um lado esta escolha ofereceu maior flexibilidade para utilização do dispositivo, abrindo espaço para incorporação de funcionalidades e inovações no projeto. Em contrapartida, por se tratar de um protótipo acadêmico, não foi possível usufruir do suporte que os dispositivos de mercado proporcionariam.

Vale ressaltar ainda, que as aplicações desenvolvidas ficaram restritas às funcionalidades disponíveis na versão 1.0 do IOM. Apesar de existirem planos de incorporação tecnológicas ao dispositivo, elas não foram integradas em tempo, inviabilizando a implementação de certas funcionalidades. Contudo alguns dos desdobramentos deste trabalho beneficiam o projeto de ampliação do IOM.

1.5 Contribuições

O desenvolvimento deste trabalho prevê uma série de contribuições, ao longo da execução de todas as etapas metodológicas previstas, sejam no nível teórico ou prático. Dentre as contribuições gerais pretendidas com este estudo está a identificação de abordagens, tendências e aspectos tecnológicos relativos ao desenvolvimento de TA, por meio do MSL. Devido a evolução natural do trabalho e os diferentes pontos-de-vista abordados no tema, diversos trabalhos derivados no MSL foram publicados:

- **Cardoso, R. C.**; Costa, V. K.; Rodrigues, A. S.; Tavares, T. A. *Análise de Frameworks para o Desenvolvimento de Produtos voltados a Tecnologia Assistiva. XVII Encontro de Pós-Graduação da Universidade Federal de Pelotas*, 2016. (Sem qualis – Artigo Completo).
- Costa, V. K., Rodrigues, A. S., **Cardoso, R. C.**, Tavares, T. A. *Investigação preliminar sobre interfaces de usuário em produtos de Tecnologia Assistiva. In: Anais do ENPOS - XVII Encontro de Pós-Graduação da UFPEL*, Pelotas, 2016. (Sem Qualis - Artigo resumido).
- Costa, V. K., Rodrigues, A. S., **Cardoso, R. C.**, Tavares, T. A. (2017) *Best practices for Graphical User Interfaces design with interaction based on head movements. In: Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems*. ACM, 2017. p. 14. (Qualis B1 – Artigo Completo).
- Rodrigues, A. S., Costa, V. K., **Cardoso, R. C.**, Machado, M. B., Tavares, T. A. (2019). *Um Estudo de Mapeamento Sistemático sobre Metodologias de Avaliação em Interação Humano-Computador voltadas à Tecnologia Assistiva com foco em Pessoas com Deficiência Motora. iSys-Revista Brasileira de Sistemas de Informação*, 11(3), 90-126. (Qualis B2 – Periódico).

- Costa, V. K., Rodrigues, A. S., **Cardoso, R. C.**, Tavares, T. A. (2019). *Mapeamento Sistemático de Literatura sobre estudos de Interfaces de Usuário em Tecnologia Assistiva*. **Ergodesign & HCI**, v. 5, p. 31-40, 2017. (Sem qualis – Periódico).
- **Cardoso, R. C.** Tavares, T. A. *Uma Análise das Características relacionadas ao Desenvolvimento de Aplicações voltadas a Tecnologia Assistiva*. In: **XVI Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais**, 2017, Joinville. (IHC 2017), 2017. (Qualis B2 – Artigo resumido).

Adicionalmente, a condução do estudo qualitativo sobre dispositivos de interação disponíveis no mercado examina e compara as funcionalidades e propriedades essenciais expostas pelas estruturas reutilizáveis disponibilizadas. A seguinte publicação destaca os principais resultados deste levantamento:

- Ferreira, A. , XAVIER, K. F. , Peroba, J. A. , **Cardoso, R. C.**, Costa, V. K., Rodrigues, A., Machado, M. B. , Tavares, Tatiana Aires. *A comparative analysis about natural user interface technologies*. In: **XXIII Simpósio Brasileiro de Sistemas Multimídia e Web - WebMedia**, Gramado, SBC, 2017. (Qualis B2 - Artigo resumido).

Outra contribuição, no nível de análise conceitual, foi a proposição de uma taxonomia para classificação das soluções de TA recuperadas por meio do MSL. Este resultado é apresentado com detalhes na Seção 3.3, além de estar em processo de revisão para publicação em 2020 no seguinte periódico:

- **Cardoso, R. C.**, Rodrigues, A. S., Costa, V. K., Tavares, T. A. *Solutions focused on High-Level Assistive Technology: Perceptions and Trends observed from a Systematic Literature Mapping*. **SBC Journal on Interactive Systems**. Em processo de revisão, com a publicação prevista para 2020. (Qualis B3 – Periódico).

Além das contribuições resultantes a partir do método de pesquisa, o desenvolvimento e avaliações práticas previstas na metodologia, levou a uma série de publicações relevantes ao contexto deste trabalho. Especificamente tratando do IOM, uma série de experimentos, conduzidos pelo grupo de pesquisa *WeTech*, foi realizada com este dispositivo. Estas avaliações resultaram nos seguintes trabalhos:

- Rodrigues, A. S., Costa, V., Machado, **Cardoso, R. C.**, M. B., Rocha, A. L., de Oliveira, J. M., Tavares, T. A. (2016, July). *Evaluation of the use of eye and head movements for mouse-like functions by using iom device*. In **International Conference on Universal Access in Human-Computer Interaction** (pp. 81-91). Springer, Cham, 2016. (Qualis B1 – Artigo Completo).

- Rodrigues, A. S., Costa, V. K., **Cardoso, R. C.**, Tavares, T. A.. *Análise de Métricas para Avaliação da Interação baseada em Apontadores: Um estudo de caso para o dispositivo IOM. XVII Encontro de Pós-Graduação da UFPel*, 2016. (Sem qualis – Artigo Completo).
- Rodrigues, A. S., da Costa, V. K., **Cardoso, R. C.**, Machado, M. B., Machado, M. B., & Tavares, T. A. (2017, June). *Evaluation of a head-tracking pointing device for users with motor disabilities. In Proceedings of the 10th International Conference on Pervasive Technologies Related to Assistive Environments* (pp. 156-162). ACM, 2017. (Qualis B1 - Artigo Completo).

As próximas contribuições dizem respeito ao *design*, desenvolvimento e avaliação de soluções de TA produzidas neste trabalho. Elas são fruto dos ciclos de desenvolvimento previstos na metodologia, sendo discutidos em detalhes ao longo do Capítulo 4:

- Xavier, K. F., Peroba, J. A., Costa, V. K., **Cardoso, R. C.**, Peroba, J. A., Peglow, J., Machado, M. B., Rodrigues, A. S.. Desenvolvimento de software de configuração para dispositivo de Tecnologia Assistiva Interface Óculos Mouse (IOM). **In: XXV Congresso de Iniciação Científica da Universidade Federal de Pelotas. Anais do CIC UFPel**, Pelotas, 2016. (Sem Qualis - Artigo resumido).
- Peroba, J. A., Xavier, K. F., Rodrigues, A. S., **Cardoso, R. C.**, Costa, V. K., Machado, M. B.. Proposta de desenvolvimento de um firmware para Tecnologia Assistiva do dispositivo Interface Óculos Mouse (IOM). **In: XXV Congresso de Iniciação Científica da Universidade Federal de Pelotas - Anais do CiC UFPel 2016**, Pelotas, 2016. (Sem Qualis - Artigo resumido).
- Costa, V. K., Rodrigues, A. S., **Cardoso, R. C.**, Tavares, T. A., Quadros, C., Machado, M. B. *Interface Óculos Mouse (IOM): Inclusão através de uma Tecnologia Assistiva de baixo custo com foco em acessibilidade. In: I Congresso Luso Brasileiro sobre Transtorno do Espectro do Autismo e Educação Inclusiva*, 2016, Pelotas. (Sem Qualis - Artigo resumido).
- Costa, V. K., Rodrigues, A. S., **Cardoso, R. C.**, Tavares, T. A. *Uma Análise sobre Metodologias de Avaliação em Interação Humano-Computador voltadas a Tecnologia Assistiva. In: Anais do ENPOS - XVIII Encontro de Pós-Graduação da UFPEL*, Pelotas, 2017.(Sem Qualis - Artigo resumido).
- **Cardoso, R. C.**, Costa, V. K., Rodrigues, A. S., Tavares, T. A. *Uma Análise sobre o Desenvolvimento de Soluções voltada a Tecnologia Assistiva de Alto Nível. In: Anais do ENPOS - XVIII Encontro de Pós-Graduação da UFPEL*, Pelotas, 2017. (Sem Qualis - Artigo resumido).

- Costa, V. K., Rodrigues, A. S., **Cardoso, R. C.**, Tavares, T. A. *Sugestões para Projeto de Interfaces Gráficas de Usuário (IGU) com Interação Baseada em Movimentos de Cabeça*. In: **Anais do ENPOS - XVIII Encontro de Pós-Graduação da UFPEL**, Pelotas, 2017. (Sem Qualis - Artigo resumido).
- Peroba, J. A., da Costa, V. K., Rodrigues, A. S., Tavares, T. A., Machado, M. B., Xavier, K. F., **Cardoso, R. C.** (2017, October). *An IoT application for Home Control focused on Assistive Technology*. In **Anais Estendidos do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web** (pp. 119-122), Gramado SBC, 2017. (Qualis B2 - Artigo resumido).
- Xavier, K. F., da Costa, V. K., **Cardoso, R. C.**, Peroba, J. A., Ferreira, A. O. L., Machado, M. B., Rodrigues, A. S. (2017, October). *VisiUMouse: An Ubiquitous Computer Vision Technology for People with Motor Disabilities*. In **Anais Estendidos do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web** (pp. 115-118). Gramado, SBC, 2017. (Qualis B2 - Artigo resumido).
- **Cardoso, R.**; TAVARES, T.; OLIVEIRA, R.; SILVA, T. *IOM4TV: An AT-based solution for an interactive TV system targeted to people with motor disabilities*. In: **Ibero American Conference on Applications and Usability fo Interactive TV**, 2018. Anais.[S.l.: s.n.], 2018. p.17–31. (Capítulo de Livro).
- **Cardoso, R. C.**, Rodrigues, A., Coelho, M., Tavares, T., Oliveira, R., Silva, T. (2018, October). *IOM4TV: An AT-Based Solution for People with Motor Disabilities Supported in iTV*. In **Iberoamerican Conference on Applications and Usability of Interactive TV** (pp. 99-114). Springer, Cham. (Qualis B2 - artigo completo).
- Machado, M. B., Rodrigues, A. S., **Cardoso, R. C.**, Costa, V. K., Machado, M. B., Tavares, T. A. *An adaptive Hardware and Software Based Human Computer Interface for People with Motor Disabilities*. In **IEEE Latin America Transactions**, v. 13, n. 11, p. 3690-3696, 2019. (Qualis B2 - Periódico).
- **Cardoso, R. C.** Costa, V. K. ; Rodrigues, A. S.; Tavares, T. A. *Doce Labirinto: Experiência de jogo utilizando interação baseada em movimentos da cabeça e recursos tangíveis*. In **XV Simpósio Brasileiro de Jogos e Entretenimento Digital**, 2016. (Qualis B1 - Artigo Completo).
- **Cardoso, R. C.**, Rodrigues, A., Costa, V., Oliveira, R., Silva, T., Tavares, T. *Enhancing the Architectural Requirements of an AT Software Framework through iTV Ecosystem*. **Proceedings of the 6th Iberoamerican Conference on Applications and Usability of Interactive TV (JAUTI 2019)**, 2019. (Sem qualis - Artigo Completo).

O desenvolvimento das aplicações, previstas na Subseção 1.3.2, faz parte do processo metodológico adotado para a proposição de soluções de software reutilizáveis. Parte deste processo é descrita em CARDOSO et al. (2019), detalhando a coleta de requisitos no contexto de iTV (*interactive TV*, TV interativa em português). No entanto, a aplicação deste método, por si só, também pode ser vista como uma contribuição relevante desta tese. Além disso, a proposição do arcabouço referência para o IOM, a implementação de sua API (*Application Programming Interface*), e a proposição do modelo ampliado, são contribuições importantes no contexto do trabalho. Estes resultados finais estão sendo compilados em artigos, e serão submetidos a pelo menos dois dos periódicos relevantes nesta área de pesquisa ao longo de 2020.

Considera-se como outras contribuições relevantes a elaboração e implementação de projetos de pesquisa relacionados a este trabalho, como coordenador. Foram aprovados quatro trabalhos ligados a questões trabalhadas nesta tese. Cada um destes projetos resultou em orientações de trabalhos de Iniciação científica (IC) de alunos do Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense (IFSul):

- ArchTech - Desenvolvimento de software para uma plataforma voltada à Tecnologia Assistiva (2016-2017);
- Concepção de uma Plataforma de Desenvolvimento voltada à Tecnologia Assistiva de Alto Nível (2017-2018);
- M2AT – *Middleware* para a concepção de soluções voltadas à Tecnologia Assistiva (TA) (2018-2019);
- Lib4IOM – Uma biblioteca para desenvolvimento de software voltado ao dispositivo de interação alternativa IOM (2019-Atual).

Também são contribuições relevantes, a elaboração e participação nos seguintes projetos de pesquisa, como colaborador:

- MA-IHC4 - Metodologias de Avaliação em Interação Humano-Computador para Tecnologia Assistiva (2018-2019);
- Metodologias de Avaliação em Interação Humano-Computador voltadas à Tecnologia Assistiva (2017-2018);
- Projeto de Interfaces Gráficas do Usuário (IGU) com interação baseada em movimentos de cabeça (2017-2018);
- Metodologia de Avaliação para Interfaces Tangíveis Voltadas para Educação Inclusiva (2016-2017);

- Interfaces de Uso (IU) para Tecnologias Assistivas com foco em deficientes com restrição motora (2016-2017);
- Interface Óculos Mouse (IOM): Interface do computador para pessoas com deficiência físico-motora e monitoramento da atividade do corpo humano (2016-2018).

Por fim, outros resultados importantes consistiram em orientações de Trabalho de Conclusão de Curso (TCC) no IFSul, interligados às pesquisas desenvolvidas durante o doutorado.

1.6 Organização da Tese

Esta tese é constituída por seis capítulos. O presente capítulo inicialmente destaca o cenário da deficiência no mundo e no Brasil. Além disso, ele apresenta a perspectiva da ampliação de capacidades das pessoas que são acometidas por limitações motoras, refletindo sobre o papel que as soluções tecnológicas podem assumir em suas vidas. Apresentam-se ainda as motivações, os principais objetivos, a metodologia adotada e as contribuições realizadas durante o desenvolvimento deste trabalho. O restante desta tese está organizado da seguinte forma:

O Capítulo 2 apresenta a fundamentação conceitual sobre diversos conteúdos utilizados ao longo deste trabalho. São explorados conceitos relacionados aos aspectos construtivos de softwares reutilizáveis, além de um aprofundamento nos princípios essenciais de TA.

Por sua vez, o Capítulo 3 destaca os resultados alcançados com a revisão bibliográfica para mapear o estado-da-arte da área de pesquisa na qual o projeto está inserido. Esse capítulo esmiúça os passos realizados no MSL, além de apresentar a taxonomia proposta com base neste estudo, e descrever as principais percepções e tendências do campo de investigação coberto. Por fim, o capítulo detalha um levantamento qualitativo realizado sobre as estruturas reutilizáveis, expostas por dispositivos de interação alternativa consolidados no mercado.

O Capítulo 4 especifica a metodologia de desenvolvimento utilizada. Para tanto, inicialmente é apresentado o dispositivo IOM, utilizado nos ciclos experimentais realizados. Como sequência o capítulo destaca o conjunto de aplicações de TA desenvolvido. São detalhados o objetivo de cada aplicação, a arquitetura projetada e a avaliação realizada. Além disso, cada seção é finalizada destacando os requisitos percebidos a partir dos experimentos realizados.

O Capítulo 5, por sua vez, apresenta o projeto de arcabouço desenvolvido para o dispositivo de interação IOM. São apresentados os resultados em termos de modelagem, implementação e testes desenvolvidos. O capítulo discute ainda aspectos construtivos para software voltado a TA percebidos ao longo do percurso. Estes aspectos

compõem um modelo conceitual ampliado, derivado a partir deste desenvolvimento.

Por fim, os principais resultados e contribuições alcançados ao longo da tese são enfatizados no Capítulo 6, juntamente com sugestões de trabalhos futuros.

Complementarmente, o Anexo A apresenta o protocolo utilizado no MSL desta tese. Já o Anexo B, destaca o certificado do estágio sanduíche, realizado durante o ano de 2018, na Universidade de Aveiro (UA) em Portugal.

2 FUNDAMENTAÇÃO CONCEITUAL

Este capítulo apresenta diversos conceitos relacionados com esta pesquisa, necessários para a compreensão do trabalho. Eles estão divididos em duas subáreas:

- Aspectos Construtivos para Desenvolvimento de Software; e
- Tecnologia Assistiva.

Desta forma, inicialmente são discutidos tópicos de Engenharia de Software. São explorados princípios e abordagens relacionados à arquitetura e ao processo de desenvolvimento de software. Em especial, questões relativas à construção de software reutilizável. A segunda parte do capítulo, por sua vez, engloba a Tecnologia Assistiva. São apresentados conceitos, características e classificações relacionadas a este campo de investigação.

2.1 Aspectos Construtivos para Desenvolvimento de Software

Atualmente, o uso de sistemas de software de diversas naturezas se incorporou à vida cotidiana da população, a qual é afetada, direta ou indiretamente, por algum tipo de aplicação computacional. Pessoas e organizações dependem cada vez mais de software, seja para realização de tarefas pessoais, ou para tomadas de decisões estratégicas institucionais.

Este cenário, no qual inúmeras necessidades e requisitos surgem dos mais diversos tipos de usuários, torna o desenvolvimento de software cada vez mais complexo. A Engenharia de Software (ES) é a área de conhecimento que foca nos aspectos da produção de software, desde os estágios iniciais da sua especificação, até a fase final de manutenção (SOMMERVILLE et al., 2011).

As próximas subseções abordam, no âmbito da ES, conceitos relacionados a princípios de Arquitetura de Software (AS) e suas principais formas de representação. Na sequência também são exploradas definições sobre técnicas e processos de desenvolvimento de software. Também são discutidos assuntos mais específicos, como

metodologias para projeto e concepção de software reutilizável, além das nuances inerentes ao processo de desenvolvimento de soluções de TA.

2.1.1 Arquitetura de Software (AS)

Um dos artefatos mais relevantes no processo de construção de software consiste na AS. A referência a este termo, imediatamente remete à analogia existente com a arquitetura tradicional, responsável pelo projeto e construção de edificações. No entanto, quando melhor analisadas, é possível perceber que estas duas áreas possuem princípios distintos. A Tabela 1 evidencia estas diferenças.

Tabela 1 – Comparação entre Arquitetura e AS.

Arquitetura Tradicional	Arquitetura de Software
Decisões são tomadas antes do início da construção.	As tomadas de decisões ocorrem em qualquer momento.
Resultado final obtido é sólido e estático (edificação).	O resultado final é maleável (software).
Projeto que não acolhe mudanças estruturais após finalizado.	O projeto acolhe mudanças, uma vez que é possível modificar o produto final, tanto por questões de performance quanto tecnológicas.

Fonte: Adaptado de HUNT; THOMAS (1999).

Sob esta perspectiva, HUNT; THOMAS (1999) relacionam o desenvolvimento de arquitetura à jardinagem. Ou seja, assim como as plantas em um jardim, o software (e sua arquitetura) está sempre em evolução, já que, além de ser difícil planejá-lo, ele é dependente de manutenção contínua. Deixando essas constatações à parte, existem várias conceituações utilizadas para definir o termo AS. FOWLER (2003) destaca uma ponderação feita por Ralph Johnson¹ na qual ele declara que na maioria dos projetos de software bem-sucedidos, os seus desenvolvedores têm uma compreensão compartilhada do *design* do sistema. Esta percepção é chamada de arquitetura. Este entendimento compartilhado inclui como o sistema é dividido em componentes e como estes interagem entre si.

Outra definição que corrobora e compartilha elementos deste conceito é apresentada por BASS; CLEMENTS; KAZMAN (2012):

"A arquitetura de software de um programa, ou sistema computacional, é a estrutura (ou estruturas) de um sistema, que inclui os elementos de software, as propriedades externamente visíveis destes elementos, e os relacionamentos entre eles".

¹Um dos autores de *Design Patterns: Elements of Reusable Object-Oriented Software* (GAMMA et al., 1995).

Uma AS se caracteriza por seguir um estilo ou tipo arquitetural, ou seja, uma coleção de princípios utilizados para atingir um objetivo. Este estilo descreve um conceito geral que busca prover uma estrutura abstrata de funcionamento para uma aplicação ou um conjunto delas (PATTERNS, 2009).

Cada tipo possui especificações que determinam componentes, relacionamentos, restrições e suposições de como estas partes são interconectadas. Assim, um tipo de arquitetura promove a reutilização do *design*, fornecendo soluções para problemas recorrentes. A Tabela 2 destaca os principais estilos arquiteturais de software.

Tabela 2 – Tipos de arquiteturas de software mais comuns.

Tipo	Descrição
Cliente-Servidor.	Sistema dividido em duas camadas, onde o cliente faz requisições ao servidor.
Baseada em Componentes.	Aplicação é decomposta em componentes, os quais expõe interfaces de comunicação bem definidas.
Em camadas.	Separa as responsabilidades da aplicação em camadas.
<i>N-tier</i> .	Também separa funcionalidades em camadas, no entanto elas estão fisicamente distribuídas.
<i>Service-Oriented Architecture (SOA)</i> .	Aplicações que expõem e consomem funcionalidades como um serviço.
Orientada a Objetos.	Divide uma aplicação em objetos individuais reutilizáveis, com dados e comportamentos próprios.

Fonte: Adaptado de PRESSMAN; MAXIM (2016).

Existe uma série de fatores que podem influenciar o tipo de arquitetura escolhido para um determinado projeto. Dentre eles podem ser destacados: a capacidade e experiência do time de desenvolvimento; os recursos tecnológicos disponíveis; e as restrições de ambiente e infraestrutura.

É possível combinar diversos estilos arquiteturais para compor uma solução mais completa e robusta. É factível, por exemplo, a estruturação de uma arquitetura em camadas, as quais possuem internamente diversos componentes, os quais por sua vez, são desenvolvidos como serviços ou de forma orientada a objetos.

Assim, dependendo do tamanho, quantidade de componentes e atores envolvidos em um projeto, uma AS pode ficar bastante complexa. Para facilitar sua compreensão, a AS pode ser representada em vários níveis, conforme a necessidade de apresentar (ou esconder) detalhes do projeto às pessoas que desempenham diferentes papéis ou que possuem expertises heterogêneas em uma organização. Pensando nisso, BROWN (2013) propõe a utilização do modelo C4 (Contexto, Contêineres, Componentes e Classes/Código), um conjunto de diagramas que possuem diferentes propósitos e níveis de abstração.

A Figura 4 destaca um exemplo de diagrama de Contexto. Como o seu próprio nome revela, ele apresenta a contextualização da solução, destacando a aplicação ao centro, cercada por possíveis usuários, módulos ou sistemas que interagem de alguma forma com o software proposto. É um diagrama de alto nível que pode ser utilizado como ponto de partida para discussões, permitindo identificar com quem é preciso conversar até a compreensão da solução.

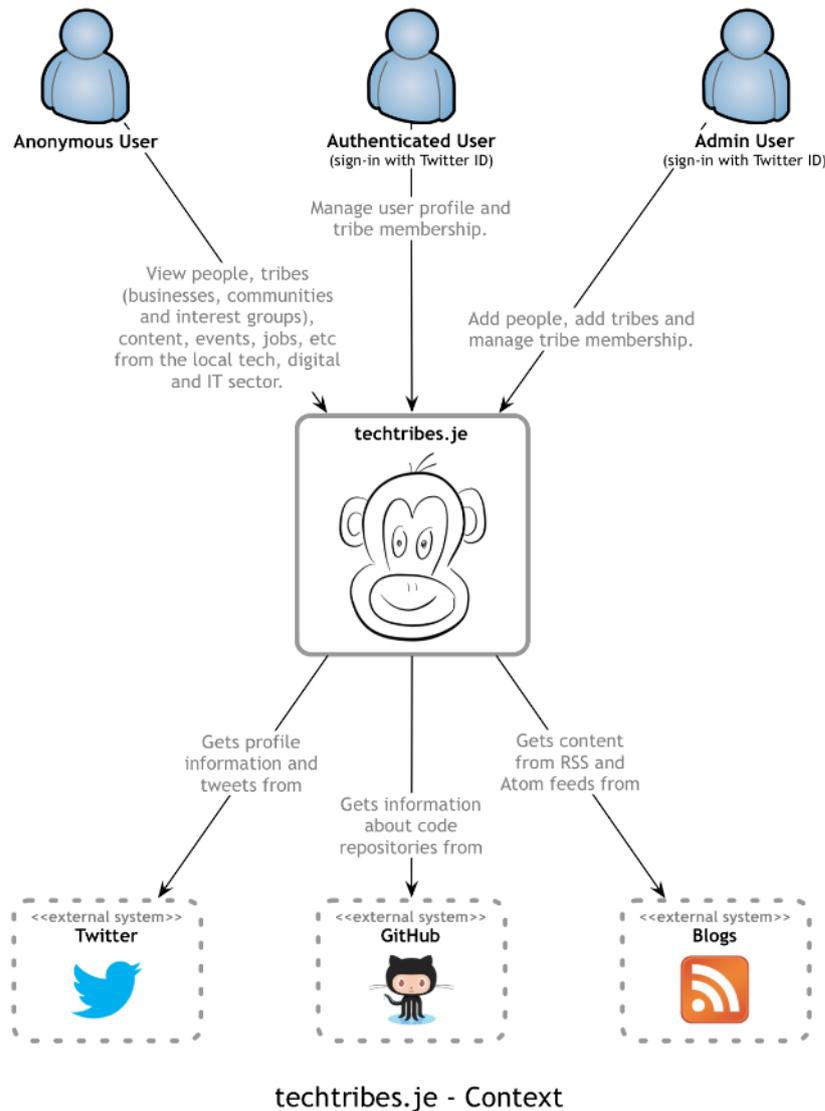
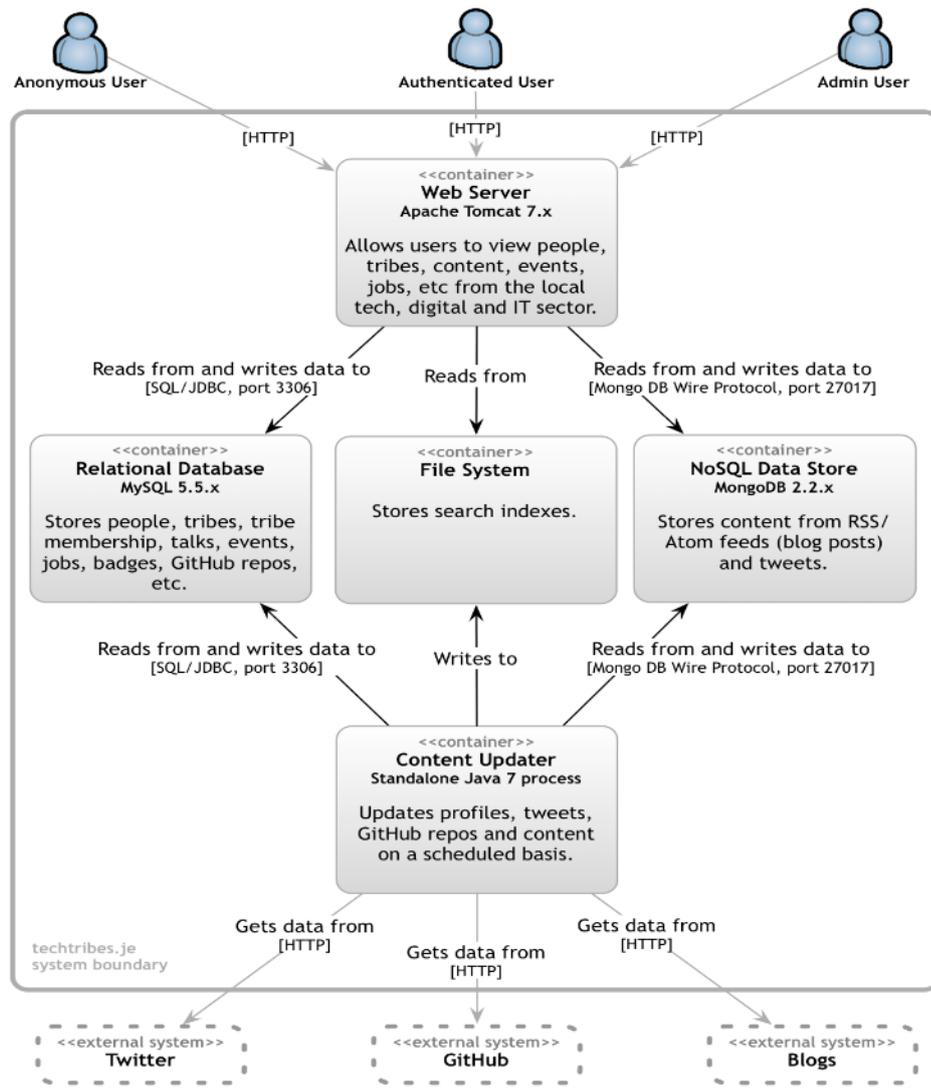


Figura 4 – Exemplo de diagrama de Contexto, segundo o modelo C4.
Fonte: BROWN (2013).

Já o diagrama de Contêineres, destaca as escolhas tecnológicas feitas no projeto, evidenciando onde cada componente da aplicação será executado. Utilizado como elo entre os diagramas de contexto e de componentes. A Figura 5 apresenta um exemplo deste tipo de diagrama.

As outras duas representações propostas pelo modelo C4 são os diagramas de Componentes e Classes. Estes diagramas são mais voltados às equipes de desenvol-



techtribes.je - Containers

Figura 5 – Exemplo de diagrama de Contêineres, segundo o modelo C4.
Fonte: BROWN (2013).

vimento, já que apresentam especificidades do projeto de software. A Figura 6 destaca um exemplo de diagrama de Componentes, representação pela qual uma aplicação é dividida em elementos com responsabilidades e dependências específicas.

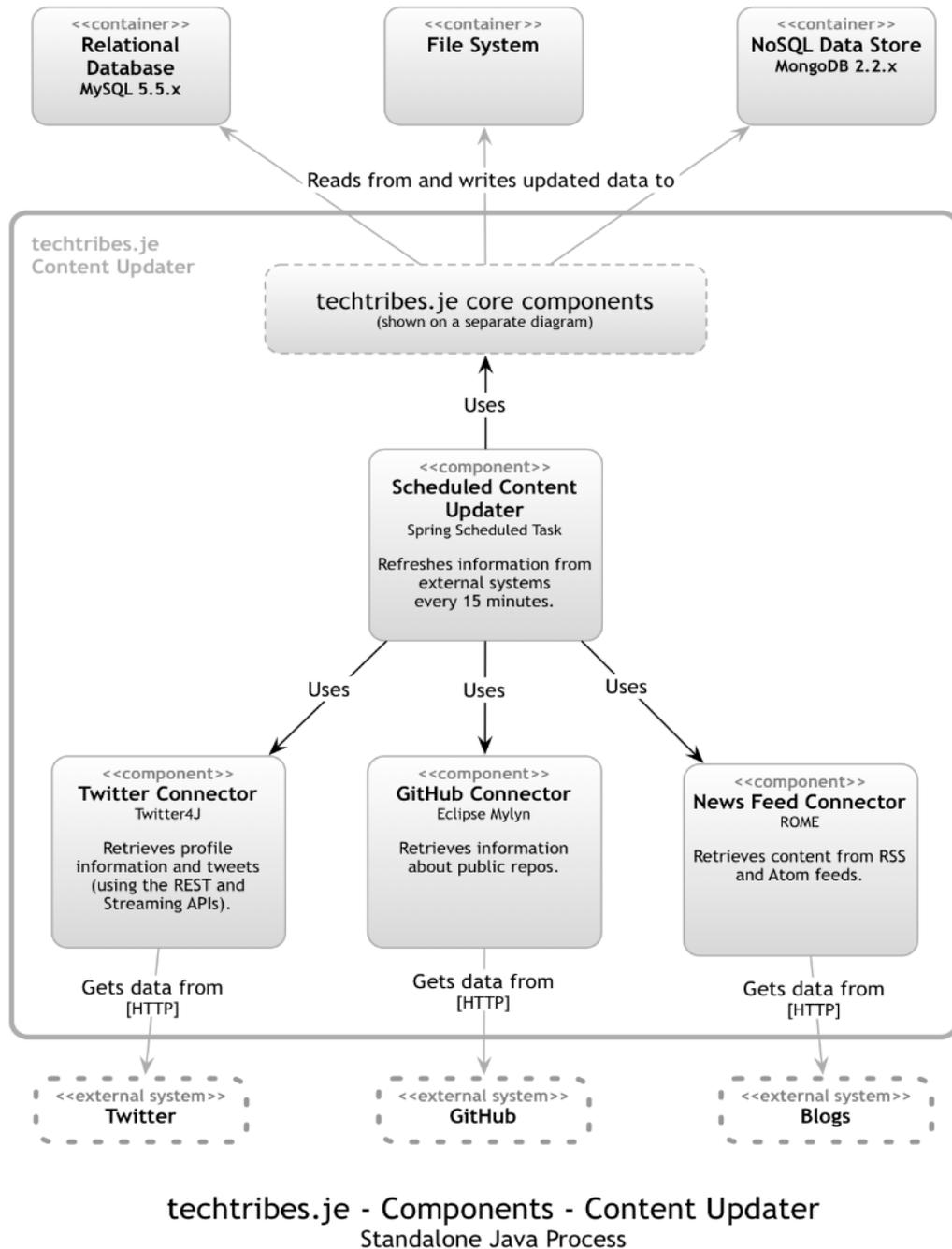


Figura 6 – Exemplo de diagrama de Componentes, segundo o modelo C4.
Fonte: BROWN (2013).

Por fim, o diagrama de Classes é o último proposto pelo modelo C4. Seu objetivo é apresentar detalhes de implementação, focando nas estruturas internas dos componentes de software. A Figura 7 apresenta um exemplo desta forma de representação.

A ideia do C4 é que a utilização de um número maior de diagramas simplificados facilite o entendimento das aplicações desenvolvidas e suas interações. Além disso,

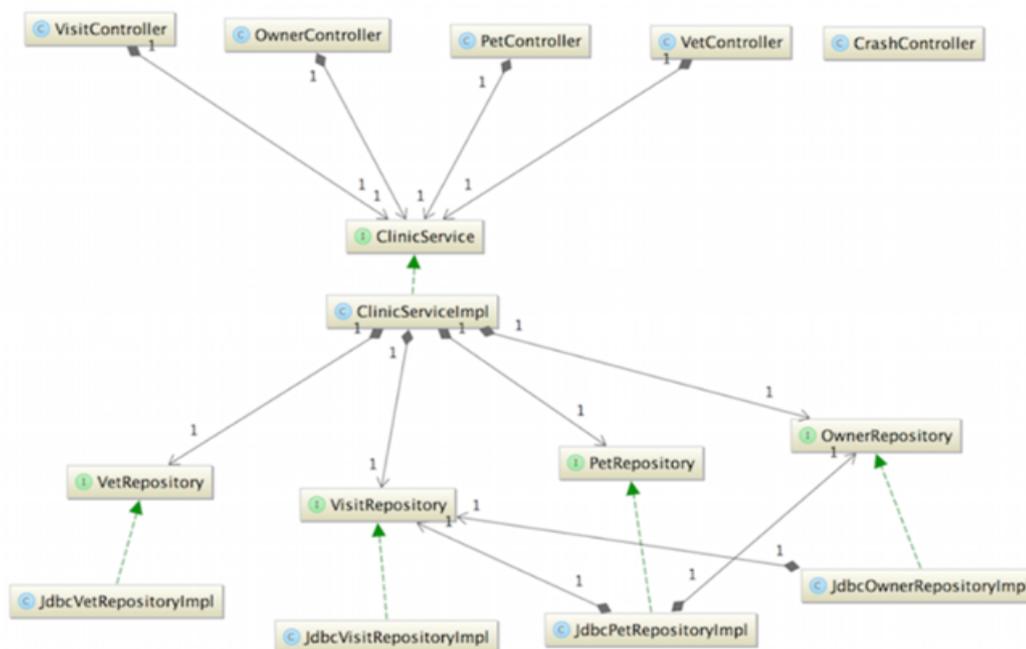


Figura 7 – Exemplo de diagrama de Classes, segundo o modelo C4.
Fonte: BROWN (2013).

permite que pessoas com diferentes expertises consigam compreender como a solução funciona, compartilhando uma visão geral da arquitetura proposta.

Para avaliar e compreender os requisitos e funcionalidades de soluções de software, ou validar uma AS, uma abordagem bastante utilizada é a prototipação.

2.1.2 Prototipação

Na ES, um protótipo é uma aplicação em estado inicial de desenvolvimento, útil para demonstrar conceitos ou experimentar opções disponíveis, com o intuito de vislumbrar diferentes formas de atingir objetivos desejados. Segundo NIELSEN (1994) os protótipos podem ser divididos em duas dimensões:

- *Protótipo horizontal* - provê uma visão de uma aplicação completa, com foco na interação do usuário. Útil para validar requisitos das interfaces de usuários e escopo da solução, demonstrar a versão atual do sistema, estimar tempos de desenvolvimento, custo e esforços necessários; e
- *Protótipo vertical* - foca na elaboração de uma parte particular da aplicação, sendo adequado para levantar os requisitos de uma determinada funcionalidade. Sua adoção resulta em benefícios como refinamento de funcionalidades específicas, mudanças de interface, identificação de problemas de desempenho ou elucidação de requisitos complexos.

Segundo VIANNA (2012), a prototipação permite ao desenvolvedor: (a) selecionar, refinar, materializar e avaliar interativamente ideias; (b) validar as soluções junto a uma

amostra do público; e (c) antecipar problemas com o intuito de reduzir riscos e otimizar gastos. Assim, a prototipação pode ser percebida como um processo evolucionário (ou incremental), já que ele se caracteriza por iniciar com uma aplicação simples, na qual os requisitos mais importantes são implantados de forma que a solução seja ampliada e modificada à medida que novos requisitos são descobertos (PRESSMAN; MAXIM, 2016).

Em termos de desenvolvimento científico, a prototipação pode ser relacionada com a experimentação. A investigação e o desenvolvimento de aplicações por experimentação não são recentes. No contexto da ES ela se concentra na coleta de evidências, por meio de medições e experimentos envolvendo sistemas de software. Segundo LIMA; NETO; EMER (2014), existem três principais formas de se conduzir uma investigação deste tipo:

- *Experimento* - se caracteriza pela manipulação de algumas variáveis e a observação de outras em um ambiente controlado. É um procedimento planejado, com base em uma hipótese;
- *Estudo de caso* - pesquisa aprofundada a respeito de um ou poucos objetos de investigação, a qual busca permitir amplo e detalhado conhecimento sobre o mesmo (GIL, 2009); e
- *Survey* - análise executada em retrospectiva, com o objetivo de determinar distribuição de atributos/características, explicar a adoção de técnicas/tecnologias, ou ser um estudo preliminar que antecede uma investigação mais profunda.

Os experimentos, especificamente, buscam permitir que seja possível avaliar aspectos de atividade humana de forma sistemática, disciplinada, computável e controlada. Segundo WOHLIN et al. (2012 apud TRAVASSOS; GUROV; AMARAL, 2002), existem quatro métodos para condução de experimentos na ES:

- *Método Científico* - abordagem para construção de modelos. Método que observa o mundo, sugere o modelo ou a teoria de comportamento, mede e analisa, verifica as hipóteses do modelo ou da teoria;
- *Método de Engenharia* - orientado à melhoria evolutiva, a qual assume a existência de algum modelo do processo ou software, e o modifica com o propósito de aprimorá-lo;
- *Método Experimental* - sugere o modelo, desenvolve o método quantitativo e qualitativo, aplica um experimento, mede e analisa, avalia e depois repete este processo;

- Método Analítico - é dedutivo, voltado à sugestão de teorias formais, na qual seu desenvolvimento deriva resultados que, se possível, são comparados a observações empíricas.

No contexto de ES, à primeira vista o método experimental parece mais adequado, uma vez que busca propor e avaliar modelos por meio de experimentos sucessivos (WOHLIN et al., 2012). No entanto, é viável a aplicação dos outros métodos para finalidades específicas. A abordagem científica, por exemplo, pode ser usada para verificar a forma como o software é desenvolvido. O método de engenharia, por sua vez, pode ajudar a demonstrar se uma determinada aplicação possui melhor desempenho que outra. Por fim, o analítico pode ser adotado em cenários nos quais seja preciso provar conceitos por meio de modelos matemáticos.

Assim, a prototipação é uma técnica muito importante já que tem como função primordial permitir avaliar as ideias geradas, validando os requisitos estabelecidos, durante a execução do processo de desenvolvimento de software.

2.1.3 Processo de Software

O processo de desenvolvimento de software, formalmente conhecido como Ciclo de Vida de Desenvolvimento de Sistemas (CVDS²), consiste em um conjunto de atividades que devem ser realizadas quando algum artefato será criado (PRESSMAN; MAXIM, 2016). O modelo parte da premissa que cada atividade possui um conjunto de ações de engenharia. Por sua vez, cada ação tem uma coleção de tarefas que:

1. Identificam o que precisa ser completado;
2. Definem os artefatos que devem ser produzidos;
3. Especificam os fatores de qualidade; e
4. Determinam os *checkpoints* para controle de progresso.

Esta organização pode ser observada na Figura 8, a qual representa um processo de software genérico.

O CVDS mais antigo é o Cascata, o qual organiza e efetua estas atividades de forma sequencial. Ele é caracterizado pela divisão em fases bem definidas, nas quais cada etapa gera um artefato que é utilizado como entrada na fase seguinte (ALSHAMRANI; BAHATTAB, 2015).

Outra abordagem de desenvolvimento é chamada de Incremental, método no qual as atividades são intercaladas de forma a produzir uma série de versões que originarão o software, e demais artefatos planejados, ao final do processo. Combinando

²Tradução do termo em inglês *Systems Development Life Cycle* (SDLC).

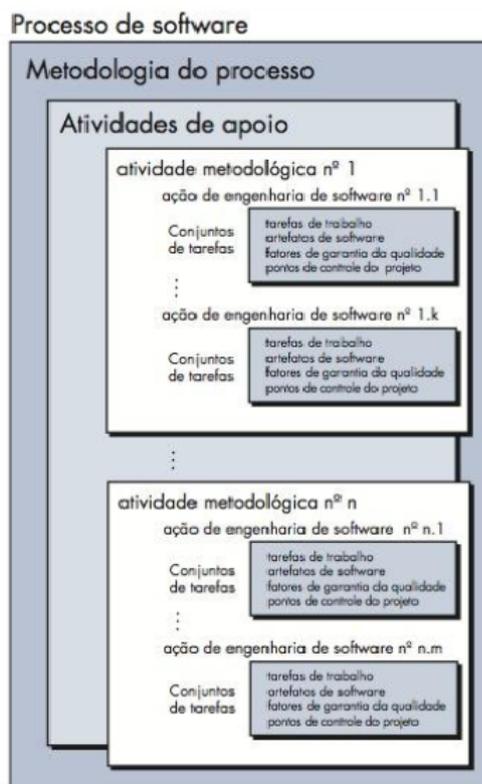


Figura 8 – Processo de software genérico.
Fonte: PRESSMAN; MAXIM (2016).

elementos do modelo em cascata com estágios e atributos da prototipação, a filosofia de desenvolvimento incremental é a base do que atualmente é conhecido como desenvolvimento ágil de software (STOICA et al., 2016).

A ideia de agilidade na concepção de aplicações se origina em fevereiro de 2001, quando representantes de 17 organizações se reuniram para discutir e identificar maneiras mais fáceis de desenvolver projetos de software. Este encontro originou o conhecido *Manifesto para o Desenvolvimento Ágil de Software*, e seus 12 princípios (BECK et al., 2001). Abordagens ágeis compartilham a ideia do processo centrado nas pessoas, com geração de artefatos a partir de iterações, impondo um caráter adaptativo durante os ciclos de desenvolvimento (TELES, 2017).

Outra vertente amplamente utilizada é a *Component-Based Software Engineering* (CBSE) a qual prima pelo desenvolvimento de aplicações de software por meio da construção baseada em componentes pré-existentes. Sua premissa mais marcante é a orientação ao reuso, característica que preza pela criação de elementos de software reutilizáveis (VALE et al., 2016).

2.1.4 Técnicas de Reutilização de Software

Além da CBSE, existe uma variedade de técnicas que procuram por similaridades em aplicações em um determinado domínio, para explorar seu potencial de reutiliza-

ção. A Figura 9 evidencia este cenário, destacando uma série de abordagens que adotam o reuso em diferentes níveis em um processo de desenvolvimento.

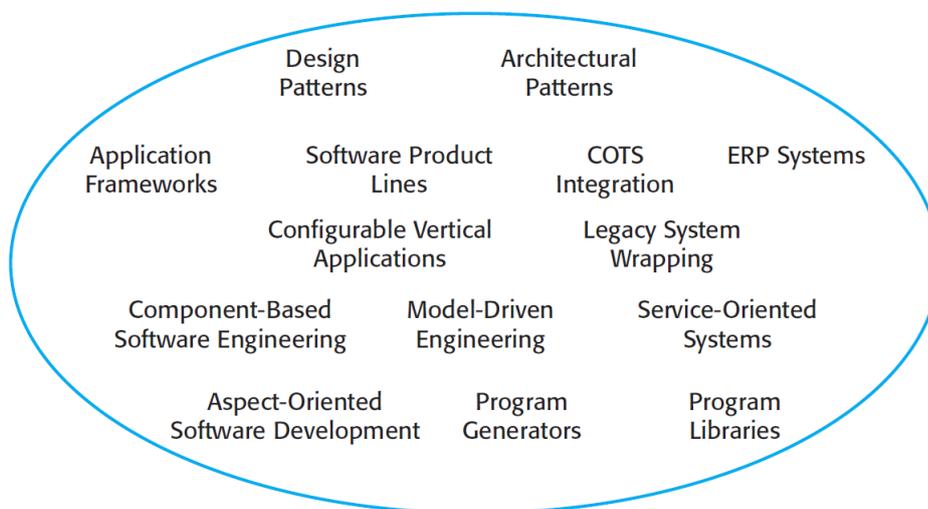


Figura 9 – Técnicas de reutilização.
Fonte: SOMMERVILLE (2011).

Como pode ser observado, o reuso de software pode ser aplicado em diversos níveis. Conceitualmente, por exemplo, os *Design Patterns* (ou Padrões de Projeto) são descrições de soluções recorrentes para um problema comum de *design* (GAMMA et al., 2004). Um nível acima, ainda podem ser vistos *Architectural Patterns*, um conceito semelhante, aplicável a um contexto mais amplo, o de AS.

Em um nível mais baixo, pode ser encontrado o conceito de arcabouço. Segundo MACEDO; BARANAUSKAS; BULÇÃO-NETO (2018), são considerados arcabouços de software, projetos reutilizáveis, de parte ou da totalidade, de um sistema descrito por um conjunto de códigos, bibliotecas, ferramentas ou API. Os arcabouços possuem algumas classificações utilizadas para categorizá-los. Uma delas considera seu objetivo, dividindo-os em conceituais ou lógicos.

- *Conceituais* - definem um esquema teórico de dados, o qual poderá posteriormente ser traduzido para um esquema de dados de um domínio específico; e
- *Lógicos* - estabelecem um conjunto de classes que serão implementadas em alguma linguagem de programação.

Já WANGLER; PAHEERATHAN (2000) organizam os arcabouços de software de acordo com a sua natureza, dividindo-os em:

- *Horizontais* - estruturas genéricas que podem ser utilizadas para desenvolver aplicações de diferentes finalidades. São utilizadas para solucionar questões

relacionadas a camadas específicas de um software como, por exemplo, *frameworks* voltados a camada de persistência de dados ou interfaces gráficas (SOBRAL, 2015); e

- *Verticais* - soluções mais específicas, com sua aplicação limitada a um domínio particular bem identificado. São construídas com base na experiência obtida num determinado contexto específico. Por exemplo, arcabouços para desenvolvimento de aplicações de aviação, comércio eletrônico, seguros, entre outros.

Em se tratando de arcabouços de software, diversos tipos de soluções podem ser construídos gradativamente, dependendo das necessidades ou objetivos a serem alcançados em determinado projeto. Uma **API**, por exemplo, é um conjunto de assinaturas que são exportadas e disponibilizadas aos usuários de uma biblioteca ou *framework*, para que estes escrevam suas aplicações (BOURQUE; FAIRLEY et al., 2014). É uma forma mais abstrata e conceitual já que descreve os serviços ou funcionalidades fornecidas por um software. Uma API é útil quando existe a intenção de que outros desenvolvedores criem produtos associados ao serviço que está sendo disponibilizado. Várias API disponibilizam seus códigos e instruções para serem usados em outros *sites* da maneira mais conveniente para seus usuários. Um exemplo bastante utilizado é a API do *Google Maps*³, a qual disponibiliza serviços para acesso e manipulação de dados geográficos.

Por sua vez, uma **Biblioteca** é definida por IEEE (1990) como uma coleção de software e documentos relacionados, projetada para auxiliar o desenvolvimento, uso ou manutenção de software. Ela ajuda a realizar funções específicas, podendo ser entendida como a implementação das regras de uma API, sendo, portanto, mais concreta. Trata-se de uma ferramenta a partir da qual o desenvolvedor utiliza os métodos ou funções dentro do código da aplicação (BOURQUE; FAIRLEY et al., 2014).

Um **Framework**, segundo GAMMA et al. (2004), é um conjunto de classes cooperantes que compõem um projeto reutilizável para uma classe específica de software. Permite que aplicações sejam criadas com base em um alicerce comum, o qual possui um nível superior de abstração para permitir a reutilização de componentes ou estruturas disponibilizadas por ele. Podem ser compreendidos como conjuntos de bibliotecas necessárias para executar uma operação maior. Ele encapsula comportamentos da API em implementações mais complexas, permitindo o seu uso de forma mais flexível, por meio de extensões, configurações e inversões de controle. Um framework caracteriza-se por chamar o código da aplicação desenvolvida, já que usualmente provê lacunas que devem ser preenchidas para executar conforme as necessidades de cada solução, funcionando como esqueleto de uma aplicação.

³<https://developers.google.com/maps/>

Por fim, um **SDK** (*Software Development Kit*) pode ser visto como um conjunto de bibliotecas que auxiliam o desenvolvimento de soluções voltadas a uma determinada plataforma ou Sistema Operacional (SO) (BOURQUE; FAIRLEY et al., 2014).

Devido ao seu grau de abstração mais amplo, existem abordagens que auxiliam o desenvolvimento de arcabouços de software. Nesta perspectiva, ROBERTS; JOHNSON (1996), por exemplo, descrevem uma série de padrões que auxiliam a evolução de componentes reutilizáveis. A ideia consiste em executar uma série de etapas, com o objetivo de identificar os elementos que podem compor o software reutilizável. A Figura 10 apresenta os padrões propostos.

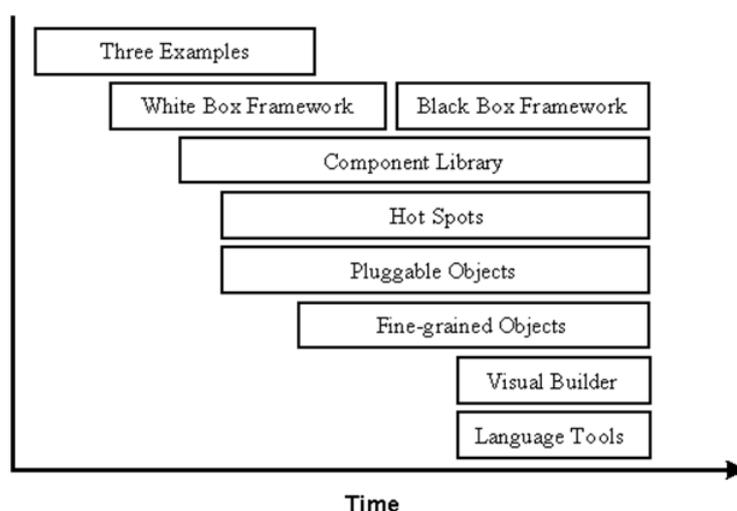


Figura 10 – Padrões para desenvolvimento de Arcabouços.
Fonte: ROBERTS; JOHNSON (1996).

O ponto de partida da abordagem prevê a criação de pelo menos três exemplos de aplicações dentro de um determinado domínio, em contextos diferentes de utilização. A partir do desenvolvimento destas soluções de software, os demais passos são derivados sucessivamente até alcançar o padrão de maturidade desejado para o projeto. O desenvolvimento destas três aplicações permite, em teoria, a criação de arcabouços *White-box* ou *Black-box*. O primeiro se caracteriza por possuir classes e métodos acessíveis e modificáveis. Já o *Black-box* é um arcabouço que apresenta um escopo de utilização mais fechado. Ou seja, seus elementos internos não são visíveis ou alteráveis. A evolução destes tipos de arcabouços converge para os demais padrões destacados.

Uma *Component Library* é um padrão que possui elementos bem definidos, os quais serão chamados e utilizados por terceiros. O passo seguinte consiste em identificar locais dentro do arcabouço que podem abrigar funcionalidades específicas da aplicação, os quais são denominados *Hot Spots*. O objetivo é definir tanto estes customizáveis, quanto as suas partes imutáveis (*frozen spots*).

Na Figura 10 ainda se podem observar dois padrões: *Pluggable* e *Fine-grained*

objects. Estes padrões sugerem projetar componentes adaptáveis que podem ser parametrizadas, em diferentes níveis de granularidade. Os passos se sucedem até chegar a um *Visual Builder* ou Ferramentas de Linguagem (*Language Tools*), em seus últimos níveis. Como os próprios nomes sugerem, soluções neste nível estão bem próximas de uma forma de programação que utiliza componentes visuais reutilizáveis no processo de implementação.

É interessante perceber que, a partir de um dado momento, a linha temporal fica sobreposta entre diversos padrões. Esta característica denota que o seguimento de todos os níveis não é mandatório, cabendo ao projeto desenvolvido definir o possível nível que deseja ou necessita alcançar.

Nesta mesma categoria de processos para desenvolvimento de soluções reutilizáveis, surge o conceito de Linha de Produtos de Software (LPS). Ela consiste em uma técnica que provê o reuso de forma sistemática para a construção de software dentro de um mesmo domínio de aplicação. A ideia é trabalhar sobre um grupo de aplicações que compartilham um domínio comum, gerenciado por funcionalidades. É uma abordagem para construção sistemática de software baseada em uma família de produtos. Um dos principais pontos é a identificação das características comuns e variáveis entre os produtos, a qual pode ser utilizada para modelar as variabilidades e similaridades das aplicações em um determinado domínio.

Assim o reuso é um dos principais motivadores da aplicação da LPS, sendo muito presente em aplicações devido ao reaproveitamento de software, ou partes deles, no processo de criação de novas aplicações (BOSCH, 2000; LINDEN, 2002; BAYER et al., 1999). O processo de desenvolvimento, seguindo os preceitos de uma LPS, acontece geralmente em duas fases distintas:

1. Engenharia de Domínio (ED) - nesta etapa é estabelecida a plataforma de reutilização, definindo o que é comum e o que é variável da linha de produtos; e
2. Engenharia de Aplicação (EA) - nesta fase são derivadas aplicações concretas a partir da plataforma estabelecida na ED.

Durante a ED, é feita uma análise dos potenciais artefatos para reuso, de acordo com as etapas de processo de desenvolvimento. Ou seja, análise de domínio nos requisitos elicitados; arquitetura de domínio sobre as arquiteturas de software projetadas; implementação de domínio na codificação produzida; e testes de domínio nos documentos/códigos de testes reutilizáveis.

Um novo produto pode ser construído a partir do reuso de artefatos de um ou vários produtos. Por fim, é feito um *feedback* da evolução de aplicações no processo de desenvolvimento, analisando o novo produto e verificando se houve evolução de software.

2.1.5 Desenvolvimento de software para TA

Uma das hipóteses, levantadas por esta tese, reflete sobre a questão de utilizar os preceitos, adotados no desenvolvimento de software convencional, dentro do cenário de TA. Soluções voltadas à TA compartilham o fato de usarem ou adaptarem tecnologias considerando um grupo que se deseja abranger, visando o provimento de modos de interação alternativos para acesso a recursos computacionais.

Todavia, em termos de processo de desenvolvimento de TA especificamente, além das questões convencionais já abordadas, aspectos multidisciplinares relacionados com a prescrição, avaliação e aplicação destas soluções devem ser considerados. OLIVEIRA et al. (2015) propõe um fluxo sequencial de etapas que devem ser seguidos para o desenvolvimento da solução, destacado na Figura 11.



Figura 11 – Etapas do processo de desenvolvimento de um produto de TA.
Fonte: OLIVEIRA et al. (2015).

Percebe-se pelo fluxograma que pontos foram adicionados antes e depois do *Desenvolvimento do projeto* propriamente dito. As etapas que antecedem o desenvolvimento abrigam atividades que visam avaliar o estado no qual se encontra o usuário, de forma que seja possível mapear quais os dispositivos mais adequados para a concepção da TA, dadas as necessidades individuais que devem ser satisfeitas. As etapas posteriores ao desenvolvimento estão relacionadas com a familiarização do usuário à ferramenta produzida (treinamento), e à avaliação, tanto do processo de interação do indivíduo com a TA, quanto do levantamento de novas necessidades que devam ser adicionadas à solução criada.

Estes passos adicionais são importantes para garantir a adequação da TA às limitações de cada pessoa, sendo um processo de concepção totalmente voltado ao

usuário. Nesse contexto, as necessidades inerentes a cada usuário devem ser endereçadas na adaptação da TA visando a potencialização das habilidades preservadas de cada indivíduo. Para identificar quais as técnicas de ES que são melhor aplicáveis ao cenário de desenvolvimento de TA, é preciso conhecer e compreender terminologias e conceitos desta área específica. A próxima seção introduz este tópico, apresentando os fundamentos e classificações essenciais a respeito do tema.

2.2 Tecnologia Assistiva (TA)

A expressão Tecnologia Assistiva, ou simplesmente TA, é relativamente nova, sendo uma definição ainda em processo de construção e amadurecimento. No entanto, o uso de recursos de TA não é recente, já que remete aos primórdios da história da humanidade. Àquela época, a adoção de mecanismos para auxiliar a vida cotidiana das pessoas já podia ser observada na utilização de artefatos primitivos, como um pedaço de pau, empregado para funcionar como uma bengala, por exemplo. Todavia, a preocupação emergente com questões relacionadas a TA, em parte, deve ser creditada a pensamentos mais contemporâneos, os quais primam pela inclusão de todos os indivíduos de uma sociedade, onde cada um tem o direito de participar ativamente da mesma.

Formalmente, o termo TA tem origem em 1988, sendo definido nos Estados Unidos da América (EUA), como elemento jurídico dentro da legislação do país, conhecida como *Public Law 100-407*, a qual compõe o *American Disabilities Act (ADA)*. Estas leis regulam os direitos dos cidadãos estadunidenses com deficiência, além de prover a base legal dos fundos públicos para compra dos recursos que estas pessoas necessitam (LAW, 1988). Fundamentando-se nos critérios estabelecidos pelo ADA, HUSSEY; COOK (2002) definem TA como:

Uma ampla gama de equipamentos, serviços, estratégias e práticas concebidas e aplicadas para minorar os problemas funcionais encontrados pelos indivíduos com deficiência.

Esta compreensão de TA leva à percepção que a área vai além da concepção de meros dispositivos, equipamentos ou ferramentas, englobando, no cerne de seu conceito, também processos, estratégias e metodologias relacionados a eles. Seguindo esta linha de pensamento, PENNSYLVANIA (2012) estabelece que o termo TA pode ser utilizado para identificar dispositivos tecnológicos, ferramentas ou serviços que buscam melhorar a qualidade de vida e aumentar a autonomia de pessoas que sofrem com limitações físicas decorrentes de problemas de nascença, doenças degenerativas ou acidentes ocorridos ao longo da vida.

No âmbito europeu, a expressão Tecnologia Assistiva frequentemente é substituída, ou utilizada como sinônimo, para *Ajudas Técnicas* ou *Tecnologias de Apoio*.

GALVÃO FILHO (2009) destaca especificamente o projeto EUSTAT⁴ - *Empowering Users Through Assistive Technology*, no qual o termo Tecnologias de Apoio possui a seguinte conceituação:

“O termo Tecnologias de Apoio engloba todos os produtos e serviços capazes de compensar limitações funcionais, facilitando a independência e aumentando a qualidade de vida das pessoas com deficiência e pessoas idosas”.

No Brasil, o processo de apropriação e sistematização do conceito de TA é ainda mais recente. Dentro do contexto brasileiro, BERSCH (2013) evidencia que TA pode ser vista como uma área de pesquisa que visa proporcionar maior independência, qualidade de vida e inclusão social aos deficientes físicos, ampliando as possibilidades de comunicação, mobilidade, interação com ambiente, aprendizado e integração com a família, amigos e sociedade.

Contudo, independentemente de localização geográfica, é possível perceber que todas as definições congruem para um objetivo essencial: a melhoria da qualidade de vida por meio do investimento em processos, dispositivos ou serviços que favoreçam, potencializem ou auxiliem habilidades ou funções pessoais comprometidas por algum tipo de deficiência ou pelo envelhecimento.

Nesta conjuntura, pesquisas relacionadas ao desenvolvimento de TA vêm conquistando espaço, produzindo soluções que auxiliam ou simplificam a vida das pessoas. Próteses, cadeiras de rodas, óculos ópticos, legendas em televisores, aparelhos auxiliares para higiene, robôs assistentes, ferramentas para acesso ao computador, entre outros, são exemplos de soluções com natureza assistiva. A diversidade de propostas, visando atacar uma variedade heterogênea de problemas advindos de diversas deficiências existentes, motiva a categorização das TA. As próximas subseções apresentam algumas destas classificações.

2.2.1 Nível de Tecnologia

Conforme destacado no Capítulo 1, uma classificação utilizada para agrupar as TA leva em consideração o custo e o funcionamento dos recursos adotados em sua concepção. De acordo com SEABRA; MENDES (2009 apud LAUAND, 2005), as TA estão divididas em três níveis principais:

- TA de Baixo Nível;
- TA de Médio Nível; e
- TA de Alto Nível.

⁴Detalhes do projeto disponível em <http://www.siva.it/research/eustat/index.html>

Soluções de baixo nível consistem em dispositivos mais simples, os quais não utilizam fontes de alimentação, tendo geralmente um baixo custo para construção. Muitos deles podem ser concebidos de maneira artesanal. Exemplos de TA deste nível incluem instrumentos adaptados para higiene ou alimentação, lupas, próteses e bengalas.

Já as soluções de TA nível médio se caracterizam pela possibilidade de utilização de algum equipamento de produção de energia, como por exemplo, baterias. No entanto estas soluções não dispõem de recursos computacionais avançados, tais como hardware ou software específicos em seu funcionamento. Nesta categoria se encontram aparatos como cadeiras de rodas elétricas ou utensílios assistivos pessoais que dispõem de alguma fonte de alimentação.

Por fim, as TA de Alto Nível tratam de soluções mais sofisticadas, contando com o auxílio de software e/ou hardware específicos em seu funcionamento. São utilizadas nos mais variados cenários de aplicação, fomentando questões relacionadas a acesso a dispositivos computacionais, controle de ambientes, adaptação de jogos, entre outros. Exemplos deste tipo de TA serão amplamente explorados e debatidos ao longo do Capítulo 3.

2.2.2 Norma Internacional ISO 9999

A *International Organization for Standardization* (ISO) é uma federação mundial de organismos nacionais de normalização. A norma internacional ISO 9999 estabelece uma classificação de produtos de apoio, desenvolvidos para pessoas com deficiência (BOUGIE et al., 2008). De acordo com o GARCIA; GALVÃO FILHO (2012), a norma ISO 9999 define *Ajudas Técnicas* como:

Qualquer produto, instrumento, equipamento ou sistema tecnológico, de produção especializada ou comumente à venda, utilizado por pessoa com deficiência para prevenir, compensar, atenuar ou eliminar uma deficiência, incapacidade ou desvantagem.

Essa definição reforça a concepção de TA entendida apenas como produtos e ferramentas. A ISO 9999 adota um método de classificação em três níveis para estruturar a diversidade de TA entre grupos de usuários e domínios de aplicação para uma orientação melhor e simplificada (KOUROUPETROGLOU, 2013). O primeiro nível especificado pela ISO 9999 é formado pelas seguintes classes:

- *Classe 04* - Produtos de apoio para medir, apoiar, treinar ou substituir funções do corpo;
- *Classe 05* - Soluções voltadas a educação e treinos de habilidades;

- *Classe 06* - Produtos assistivos ligados ao corpo para apoiar funções neuromusculoesqueléticas ou relacionadas ao movimento (órteses) e substituição de estruturas anatômicas (próteses);
- *Classe 09* - Soluções focadas em cuidados pessoais de proteção;
- *Classe 12* - Produtos de apoio para atividades e participação relacionadas a mobilidade pessoal e transporte;
- *Classe 15* - Produtos assistivos para atividades domésticas;
- *Classe 18* - Mobiliário e adaptações para habitação e outros ambientes;
- *Classe 22* - Soluções de apoio para gestão de comunicação e informação;
- *Classe 24* - Produtos assistivos para o controle, transporte, movimentação e manuseio de objetos e dispositivos;
- *Classe 27* - Soluções focadas no controle, adaptação ou medição de elementos de ambientes físicos;
- *Classe 28* - Produtos para apoiar atividades de trabalho e participação no emprego; e
- *Classe 30* - Produtos de apoio para fins recreacionais e lazer.

A partir dessas classes principais, existem outros dois níveis de especialização previstos na ISO 9999: Subclasses e Divisões. A identificação dentro desta hierarquia é feita por meio de atribuição de códigos de dois dígitos em cada um dos três níveis propostos. Embora amplamente aceita e utilizada, a norma ISO 9999, por vezes, não atende as necessidades de algumas soluções propostas, uma vez que caracteriza apenas produtos como sendo formas de apoio, desconsiderando outros mecanismos que também compartilham natureza assistiva.

2.2.3 HEART (*Horizontal European Activities in Rehabilitation Technology*)

Orientada a conhecimentos, esta classificação se organiza em componentes técnicos, humanos e sociais. Este modelo surgiu a partir do programa europeu TIDE (*Technology Initiative for Disabled and Elderly People*), que propõe uma caracterização das Tecnologias de Apoio, baseada nos conhecimentos envolvidos na sua utilização (GALVÃO FILHO, 2009). Esta classificação considera três áreas de formação da TA:

1. Componentes Técnicos - consideram os recursos técnicos para o exercício de diferentes atividades, sendo subdivididos em *Comunicação, Mobilidade, Manipulação e Orientação*;

2. Fatores Humanos - pondera os impactos causados pela deficiência às pessoas, agrupando-os em: *Tópicos sobre a deficiência; Aceitação da TA; Seleção da TA; Aconselhamento sobre a TA; e Assistência Pessoal; e*
3. Elementos Socioeconômicos - considera as relações, interações e impactos que podem ser estabelecidos entre o usuário final da TA e realidades do seu contexto. Estes componentes podem ser caracterizados como: *Noções básicas de TA; Noções básicas do Design Universal; Emprego; Prestação de Serviços; Normalização/Qualidade; Legislação/Economia; e Recursos de Informação.*

Essa classificação proporciona uma concepção que vai além da ideia de produtos e dispositivos. Ela vai ao encontro de uma visão de deficiência referenciada em um contexto inclusivo, analisando as limitações encontradas pelo indivíduo, interpostas pelo ambiente, por realidades pessoais e condições socioeconômicas.

Assim, no contexto da classificação HEART, a pesquisa e o desenvolvimento de TA considera todo este cenário para propor aplicações, dispositivos ou metodologias, visando amenizar não apenas as limitações do indivíduo, mas também do ambiente físico e social no qual ele se encontra.

2.2.4 ATACP (*Assistive Technology Applications Certificate Program*)

Outra categorização de TA, proposta por BERSCH (2013), foi projetada com base na análise de classificações pré-existentes, e especialmente a partir da formação dos autores no (ATACP) da *California State University Northridge*. Na categorização proposta, uma TA pode ser qualificada como:

- *Auxílios para a vida diária* - Soluções que favorecem desempenho autônomo e independente em tarefas rotineiras, tais como se alimentar, cozinhar, vestir-se, tomar banho e realizar necessidades pessoais;
- *Comunicação Aumentativa e Alternativa* - Visa atender pessoas sem fala ou escrita funcional ou em descompasso entre sua necessidade comunicativa e sua habilidade em falar, escrever ou compreender;
- *Recursos de acessibilidade ao computador* - Soluções de hardware/software projetadas para tornar o computador acessível a pessoas com privações visuais, auditivas, intelectuais ou motoras;
- *Controle de ambiente* - Aplicações que permitem que pessoas com limitações motoras controlem aparelhos eletroeletrônicos, sistemas de segurança, entre outros, localizados em seu ambiente residencial ou de trabalho, por exemplo;

- *Projetos arquitetônicos para acessibilidade* - Adaptações estruturais e projetos de edificação e urbanismo que garantem acesso, funcionalidade e mobilidade a todas as pessoas, independentemente de sua condição física e sensorial;
- *Órteses e próteses* - Peças artificiais que substituem partes ausentes do corpo;
- *Adequação Postural* - Seleção de recursos para garantir posturas alinhadas, estáveis, confortáveis e com boa distribuição do peso corporal;
- *Auxílios de mobilidade* - Cadeiras de rodas manuais e motorizadas, bases móveis, andadores, *scooters* de 3 rodas e qualquer outro veículo, equipamento ou estratégia que melhore a mobilidade pessoal;
- *Auxílios para cegos ou com visão subnormal* - Dispositivos para auxílio óptico, lentes, lupas manuais e lupas eletrônicas; e software ampliador de tela. Material gráfico com texturas e relevos, mapas e gráficos táteis, por exemplo;
- *Auxílios para surdos ou com déficit auditivo* - Equipamentos infravermelho, aparelhos para surdez, telefones com teclado — teletipo (TTY), sistemas com alerta tátil-visual, entre outros;
- *Adaptações em veículos* - Acessórios que possibilitem a condução de veículos, elevadores para cadeiras de rodas, camionetes modificadas e outros veículos para transporte pessoal; e
- *Esporte e Lazer* - Recursos que favorecem a prática de esportes e participação em atividades de lazer.

No Brasil, esta categorização já foi adotada pelo Ministério da Fazenda, pelo Ministério da Ciência, Tecnologia e Inovação e também pela Secretaria Nacional de Direitos Humanos da Presidência da República, na publicação da Portaria Interministerial Nº 362, de 24 de Outubro de 2012, a qual trata sobre crédito subsidiado para compra de bens e contratação de serviços de TA para pessoas com deficiência.

2.3 Considerações Finais do Capítulo

Este capítulo buscou introduzir conceitos relativos a duas áreas de pesquisa relevantes dentro do contexto deste trabalho: Engenharia de Software e Tecnologia Assistiva.

Em termos de ES, foram destacados aspectos relacionados à construção de software. Inicialmente, foram tratadas questões de arquitetura de software, evidenciando seus principais estilos e formas de representação. Esses conceitos são essenciais,

uma vez que tanto os trabalhos relacionados, explorados no Capítulo 3, quanto os resultados da aplicação da metodologia de desenvolvimento, destacados ao longo dos capítulos 4 e 5, utilizam estas técnicas para representar e descrever as soluções tecnológicas desenvolvidas.

Complementarmente também foram abordados conceitos relacionados a prototipação, ciclo de vida do desenvolvimento de aplicações e técnicas voltadas à reutilização de software. A abordagem de tais conteúdos é vital, uma vez que parte desses métodos e nomenclaturas é utilizada em capítulos posteriores desta tese. Em especial, o processo de desenvolvimento de estruturas de software reutilizáveis é vastamente explorado nos capítulos 4 e 5, visando a concepção do arcabouço de referência como estudo de caso. Como destacado previamente, a ideia é identificar as principais características e funcionalidades que devem compor este arcabouço, por meio da criação de uma série de aplicações em um domínio de utilização, mas em diferentes contextos de uso. Esta solução de software genérica e reutilizável deve possibilitar acesso simplificado ao conjunto de propriedades e funções essenciais encontradas nos ciclos de concepção dessas aplicações.

Como destacado, estas soluções abordam conceitos de Tecnologia Assitiva, voltadas a pessoas com movimentos de cabeça preservados utilizando o dispositivo de interação alternativa IOM. Neste contexto, ao longo do presente capítulo também foram discutidos conceitos, características e taxonomias relacionadas à TA. Esta fundamentação teórica apresentou noções básicas deste campo de estudo, buscando familiarizar o leitor ao tema. Em especial, mapeando o trabalho corrente dentro das classificações discutidas, é possível categorizá-lo como:

- **ISO 9999** - A solução desta pesquisa se encaixa inicialmente na Classe 04 (produto que substitui funções do corpo) ou na classe 06 (produto anexado ao corpo para apoiar funções relacionadas ao movimento⁵. Além desta caracterização primária, dependendo do tipo de aplicação desenvolvida para utilização em conjunto com o dispositivo proposto, a solução ainda poderá ser enquadrada em pelo menos em mais duas classes: 24 (produtos para controle/manuseio de objetos ou dispositivos) e 30 (soluções para recreação ou lazer);
- **HEART** - No contexto desta classificação, a solução proposta inicialmente se atém dentro dos componentes técnicos definidos, sendo qualificada no contexto de *Manipulação e Comunicação*; e
- **ATACP** - A proposta se encaixa nos seguintes segmentos: *Recursos de acessibilidade ao computador*; *Controle de Ambiente*; *Esportes e lazer*.

⁵Não foi possível finalizar esta definição, dado o acesso restrito a norma ISO 9999:2016 completa. O mesmo vale para a caracterização em nível de subclasses e divisões.

Além destas classificações, outra categorização relevante, já destacada na seção introdutória deste trabalho, considera os recursos adotados na concepção de uma TA, dividindo-os em três níveis: Alta; Média; e Baixa. Conforme já destacado, o foco de estudo deste projeto é em características inerentes ao processo de concepção de TA de alto nível, ou seja, desenvolvimento envolvendo a utilização de recursos de hardware e software.

Como sequência deste trabalho, o próximo capítulo se aprofunda no tema específico desta tese, apresentando o estado da arte da área de pesquisa. Para tanto, é detalhado o planejamento e a condução de um Mapeamento Sistemático de Literatura (MSL) acerca do tema de pesquisa. Complementarmente, o capítulo ainda discorre sobre um levantamento qualitativo realizado sobre dispositivos de interação alternativas disponíveis comercialmente.

3 ESTADO DA ARTE

O processo de desenvolvimento de dispositivos e aplicações de TA envolve e impulsiona pesquisas científicas em diversos campos de conhecimentos. Esta pluralidade de áreas se justifica, uma vez que as possibilidades de interação proporcionadas por estas ferramentas assistivas podem utilizar os mais variados métodos para atingir as finalidades desejadas. Isso ocorre na medida que o desenvolvimento de uma TA abrange tanto aspectos tecnológicos, buscando utilizar dispositivos que ampliem a capacidade de interação preservada dos usuários com limitações físicas, quanto aos aspectos humanos ligados a saúde mental e fisiológica destes indivíduos.

Assim, com o intuito de mapear o estado-da-arte da área de pesquisa foi realizada uma ampla revisão bibliográfica do tema. O objetivo é observar o que existe em desenvolvimento nesta área, além de elencar características relevantes no processo de desenvolvimento de soluções voltadas a TA no meio acadêmico. As próximas seções detalham a metodologia utilizada, descrevendo todos os procedimentos executados e os resultados alcançados.

3.1 Mapeamento Sistemático de Literatura (MSL)

Para realizar a revisão bibliográfica proposta foi adotado o MSL, uma metodologia de pesquisa que prevê a execução de uma série de passos para gerar o mapa de literatura da área, ou seja, um conjunto de artigos que traz uma prospecção ampla do tema de pesquisa abordado (PETERSEN et al., 2008). As subseções a seguir detalham cada uma das etapas executadas neste MSL, seguindo o fluxo apresentado na Figura 12.

3.1.1 Análise Exploratória Inicial

Antes de preencher o protocolo do MSL, é recomendável executar a busca por um artigo de referência para a revisão bibliográfica. O objetivo é encontrar, por meio de pesquisa exploratória simples, um trabalho que se aproxime do tema de pesquisa que se deseja explorar. Esse artigo será utilizado como uma espécie de *calibrador*

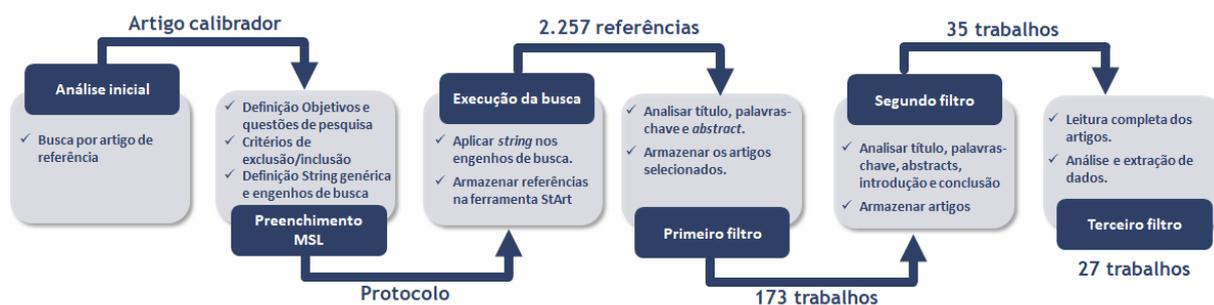


Figura 12 – Fluxo de execução do MSL.

Fonte: Elaborada pelo Autor (2019).

durante o preenchimento do restante do protocolo MSL. Como resultado desta etapa foram examinados quatro trabalhos:

- OSSMANN et al. (2014);
- SALNIKOV et al. (2014);
- MIESENBERGER et al. (2014); e
- NETO et al. (2012).

A análise comparativa entre estes trabalhos resultou na escolha do artigo de OSSMANN et al. (2014) como artigo referência para este MSL. Esse trabalho apresenta o AsTeRICS (*Assistive Technology Rapid Integration & Construction Set*), uma plataforma que permite a criação de soluções assistivas específicas por meio de componentes de software reutilizáveis. Os demais artigos encontrados nesta etapa, descrevem soluções focadas em problemas particulares, não preocupadas com aspectos de reutilização das estruturas desenvolvidas (CARDOSO et al., 2016).

3.1.2 Preenchimento do Protocolo MSL

Antes de prosseguir com a execução do protocolo MSL, é preciso especificar alguns elementos essenciais da pesquisa a ser efetuada. O primeiro item a ser estabelecido é o objetivo geral da revisão bibliográfica que será conduzida. Para este mapeamento ele foi definido como:

“Identificar e caracterizar o estado da arte sobre TA com foco em acessibilidade ao computador para pessoas com deficiência motora.”

Neste contexto o MSL, que teve a duração de seis meses, abrangeu três diferentes eixos de pesquisa:

1. Soluções voltadas ao desenvolvimento de TA;
2. Interfaces de interação baseadas em TA; e

3. Metodologias de avaliação de TA.

Este trabalho, especificamente, concentra suas atenções no primeiro tópico. Para esta temática, foram elaboradas três questões de pesquisa que deveriam ser respondidas ao final do MSL:

1. *Como as pesquisas mostram o processo de desenvolvimento e avaliação de interfaces e software voltados a TA que possibilitam as pessoas com deficiência física no uso do computador?*
2. *Existem arcabouços para desenvolvimento de software, especificamente voltadas a TA? Quais? e*
3. *O que caracteriza uma solução como sendo voltado a TA?*

Considerando estas questões e utilizando o artigo referência como calibrador, o restante do protocolo MSL¹ foi preenchido. A partir destas definições, foi elaborada uma *string* de busca, a qual visa satisfazer os três eixos de pesquisa recém destacados. Desta forma, a seguinte *string* genérica foi desenvolvida:

```
("human-computer interaction" OR HCI) AND
("assistive technology" OR accessibility) AND
(physical OR impairment OR disability) AND
(framework OR platform OR API OR evaluation OR test OR guidelines OR protocols)
```

Esta *string* de busca foi adaptada à sintaxe específica de cada um dos indexadores científicos utilizados para a coleta inicial dos artigos. Neste MSL foram adotados os seguintes engenhos de busca:

- *IEEE Xplore Digital Library*²;
- *ACM Digital Library*³;
- *Springer*⁴; e
- *Science Direct*⁵.

Estes engenhos de busca foram selecionados por agregarem uma quantidade considerável de trabalhos dentro da área de pesquisa abordada. Por fim, foi estabelecido um conjunto de critérios de inclusão e exclusão de artigos para filtragem dos trabalhos. Estes critérios foram aplicados durante as etapas de filtragem executadas no MSL. A Tabela 3 destaca os critérios utilizados.

¹Protocolo completo do MSL, disponível em <https://goo.gl/AG1cLy>.

²Disponível em <http://ieeexplore.ieee.org>

³Disponível em <http://dl.acm.org>

⁴Disponível em <http://link.springer.com/>

⁵Disponível em <http://www.sciencedirect.com>

Tabela 3 – Critérios de exclusão e inclusão utilizados no MSL.

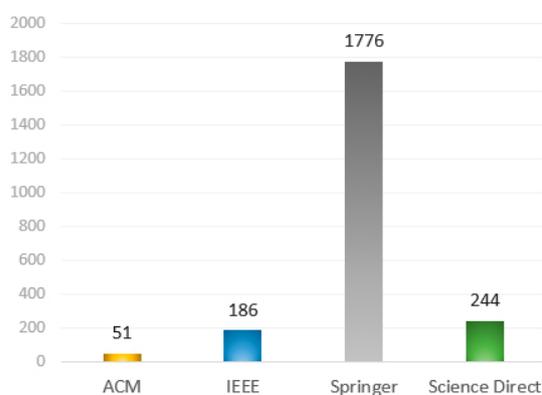
Tipo	Critério
Exclusão	Artigos publicados em evento exclusivo de área médica.
Exclusão	Artigos publicados anteriormente a 2012
Exclusão	Resumos, pôsteres ou artigos com até duas páginas
Exclusão	Artigos duplicados
Exclusão	Artigos relacionados apenas a outras deficiências
Exclusão	Livros completos
Exclusão	Artigos relacionados exclusivamente a idosos
Inclusão	Artigos envolvendo TA para acesso ao computador
Inclusão	Artigos relevantes publicados antes de 2012

Fonte: Elaborada pelo Autor (2019).

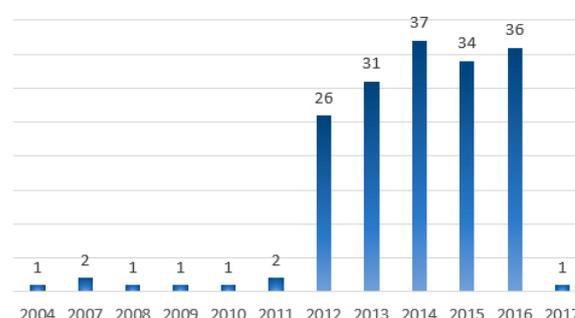
A ideia era tanto permitir o descarte de trabalhos pouco relevantes dentro da pesquisa proposta, quanto para incluir algum artigo pertinente que por ventura não tenha sido recuperado.

3.1.3 Execução do protocolo MSL

Com o protocolo devidamente preenchido, o MSL teve prosseguimento segundo o fluxo destacado anteriormente na Figura 12. A etapa de execução da busca foi realizada sobre os engenhos de pesquisa e, como resultado, 2.257 referências bibliográficas foram recuperadas e armazenadas na ferramenta de planejamento e execução de revisões sistemáticas StArt⁶. O conjunto de artigos resultante desta fase, classificados de acordo com o engenho de busca utilizado, é apresentado na Figura 13 (a).



(a) Artigos distribuídos por engenho de busca.



(b) Artigos agrupados por ano, após primeiro filtro.

Figura 13 – Gráficos sobre os artigos recuperados no MSL.

Fonte: Elaborados pelo Autor (2019), via ferramenta StArt.

⁶Ferramenta de apoio a revisões sistemáticas. Disponível em: <http://lapes.dc.ufscar.br/tools/>

Esta coleção de artigos foi então submetida ao primeiro filtro previsto, no qual um grupo de sete pesquisadores realizou uma triagem analisando título, palavras-chave e resumo dos trabalhos, utilizando os critérios de exclusão e inclusão de artigos, estabelecidos no protocolo MSL. A filtragem sobre o conjunto inicial de trabalhos resultou em um subconjunto de 173 artigos. A Figura 13 (b) destaca a distribuição dos artigos selecionados, pós aplicação do primeiro filtro, agrupados por ano de publicação.

O segundo ciclo de filtragem foi então iniciado, no qual três pesquisadores analisaram título, resumo, palavras-chave, introdução e conclusão dos 173 trabalhos. A intenção desta etapa foi caracterizar estes artigos dentro dos três eixos de pesquisa previamente estabelecidos. Os resultados individuais destas análises foram cruzados, em discussões mediadas pelo orientador da pesquisa, resultando no diagrama de Venn apresentado na Figura 14.

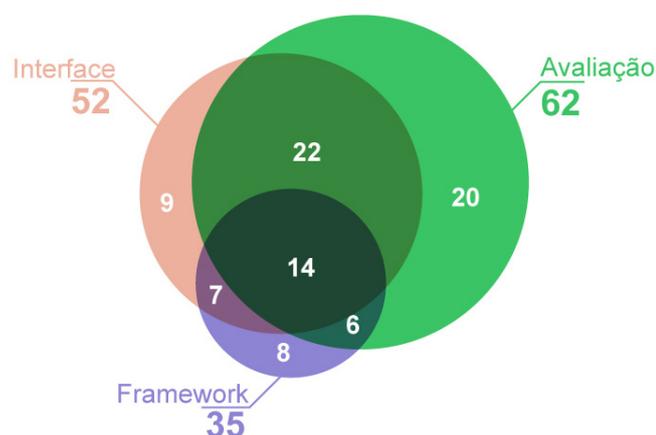


Figura 14 – Artigos caracterizados de acordo com eixo de pesquisa.
Fonte: Elaborados pelo Autor (2019), via ferramenta StArt.

Especificamente a respeito do foco deste trabalho, foram selecionados 35 trabalhos relacionados ao tema de pesquisa, identificados como *Framework* na Figura 14. Este conjunto enxuto de trabalhos foi submetido ao terceiro filtro, o qual consistia na leitura e análise completa desses artigos. Esta última filtragem resultou em um conjunto de 27 trabalhos, cujo estudo e análise originaram os resultados detalhados a seguir.

3.2 Trabalhos Recuperados no MSL

Esta seção é destinada a descrever brevemente os trabalhos que foram selecionados pelo MSL. São descritas as funcionalidades e características mais relevantes dentro do contexto deste trabalho.

3.2.1 AsTeRICS (*Assistive Technology Rapid Integration & Construction Set*)

Neste contexto, os trabalhos de OSSMANN et al. (2014) e VEIGL et al. (2013) descrevem o AsTeRICS, uma solução robusta e flexível que provê um conjunto de ferramentas para o desenvolvimento de funcionalidades baseadas em conceitos de TA, adaptáveis às necessidades de cada usuário. Sua base conceitual é formada por uma plataforma de computação embarcada executando o AsTeRICS *Runtime Environment* (ARE) com *plugins* OSGi (*Open Services Gateway Initiative*) provendo as funções de TA. Sensores e atuadores que podem ser conectados à plataforma permitem a solução desenvolvida interagir com o ambiente externo. O esquema conceitual do AsTeRICS pode ser observado na Figura 15.

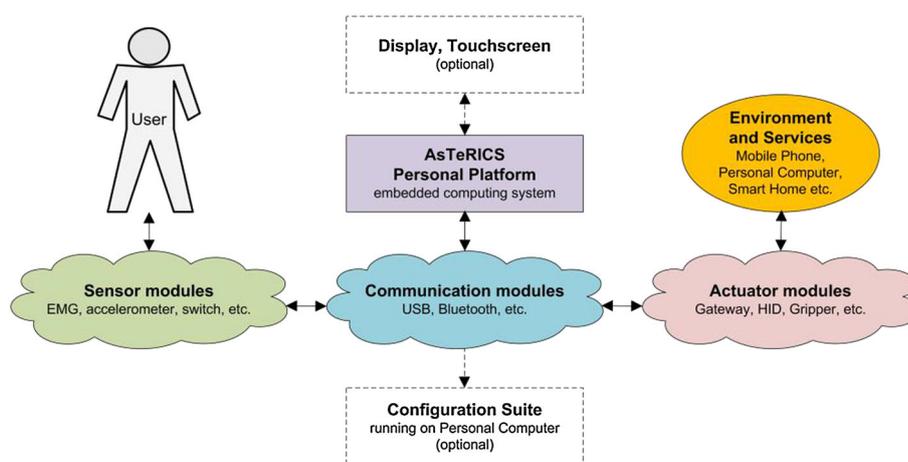


Figura 15 – Esquema conceitual do AsTeRICS.

Fonte: OSSMANN et al. (2014)

Dentre os principais elementos que compõem o AsTeRICS, podem ser destacados:

- ACS (*AsTeRICS Configuration Suite*) - uma GUI (*Graphical User Interface*) que auxilia a criação dos modelos usados na concepção de aplicações;
- ARE - um *middleware* baseado em Java/OSGi utilizado para processar sinais, gerenciar componentes e executar os modelos de TA; e
- *Plugins* - elementos que definem as possibilidades do sistema em termos de sensoriamento, processamento e atuação.

A plataforma foi desenvolvida em Java (vide a arquitetura do AsTeRICS na Figura 16), podendo ser executada sobre qualquer sistema operacional que processe esta tecnologia.

A criação de soluções utilizando o AsTeRICS é feita por meio da ACS, interface que permite conectar os componentes (dispositivos, sensores e atuadores) para produzir aplicações específicas. Os sensores disponíveis variam bastante, podendo ser

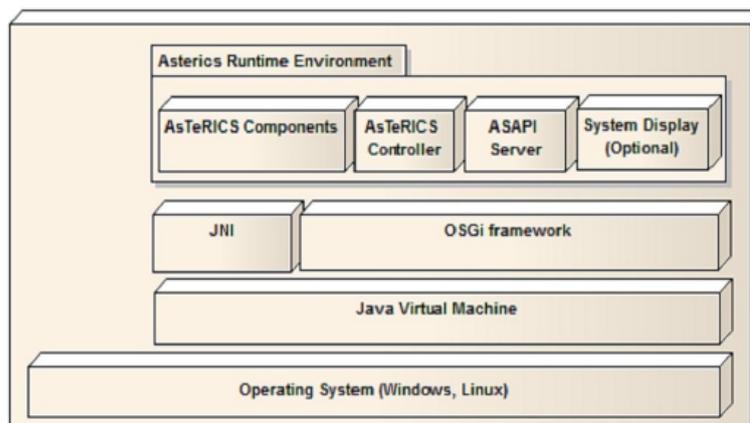


Figura 16 – Arquitetura do AsTeRICS.

Fonte: VEIGL et al. (2013).

comutadores, sensores *flex*, de sopro e inalação, sensores para rastreamento de movimentos de cabeça, reconhecimento de voz, consoles de jogos, interfaces como *Microsoft Kinect*⁷, tecnologias BNCI (*Brain/Neuronal Computer Interfaces*), entre outros. Complementarmente, vários atuadores têm suporte no AsTeRICS. Intermediando sensores e atuadores, estão os módulos de processamento, os quais adaptam os dados provenientes dos sensores para atender os usuários.

VEIGL et al. (2013) mostra o uso do AsTeRICS por meio do desenvolvimento de protótipos baseados em modelos. A partir da análise das capacidades e das necessidades específicas dos usuários, diversos modelos são desenvolvidos visando suprir três cenários: (a) Substituição do *mouse*; (b) Substituição do teclado QWERTY; e (c) Controle de ambientes. Neste cenário, o trabalho apresenta aplicações criadas por meio do AsTeRICS, tais como:

- Substituição do *mouse* baseadas em comutadores;
- Controle de *mouse* baseado em *Webcam*, utilizando *plugins* de rastreamento de face e clique por tempo;
- Simulação de *mouse* por rastreamento de movimentos da cabeça ou olhos, utilizando um sensor de inalação/exalação para simular cliques;
- Leitor de música SSVEP (*Steady-State Visual Evoked Potentials*⁸); e
- Controle de ambiente usando Enobio⁹, detecção de piscar de olhos para cliques, teclado OSKA (*On-Screen Keyboard Application*) adaptado e um transmissor infravermelho que emula controles de aparelhos de som e TV.

⁷Sensor de movimentos para os consoles *Xbox 360* e *Xbox One* da Microsoft.

⁸Sinais cerebrais que ocorrem em resposta a estímulos visuais.

⁹Sensor de eletrofisiologia vestível e *wireless* para captar sinais eletroencefalográficos (EEG).

Por sua vez, OSSMANN et al. (2014) amplia o universo de aplicações modeladas, desenvolvidas e testadas com o AsTeRICS, apresentando outros componentes disponíveis na plataforma. Ambos trabalhos descrevem os testes executados por pessoas com deficiências físicas específicas.

Já OSSMANN et al. (2012) apresenta uma aplicação desenvolvida com AsTeRICS, por meio da criação de um jogo que adapta um *PlayStation 3* (PS3) para esta finalidade. O elemento chave utilizado nesse trabalho é o módulo atuador *Human Interface Devices* (HID) Universal do ASTeRICS, um *dongle*¹⁰ USB que funciona como *mouse*, teclado ou *joystick* em um sistema computacional. As informações de controle são enviadas do ARE para o *dongle* do atuador HID através de conexão *Bluetooth* ou cabo USB. O ARE transforma as informações provenientes de outros dispositivos de entrada para as ações de controle do jogo. Este esquema de funcionamento é representado na Figura 17 (a).



Figura 17 – PS3 adaptado.
Fonte: OSSMANN et al. (2012).

O artigo descreve um modelo que permite uma nova forma de controlar jogos do PS3 via teclado OSKA. A Figura 17 (b) apresenta o layout do teclado OSKA adaptado para o *joystick* do PS3. Como estudo de caso, o artigo ajusta dois jogos para utilizar o HID Universal como controle adaptado: *Marvel Pinball* e *ModNation Racers*. Devido às características de cada jogo são utilizados diferentes métodos de entrada para simular os botões de controle do PS3.

3.2.2 ATLab (*Assistive Technology Laboratory*)

O ATLab, proposto por MIESENBERGER et al. (2014), é uma solução que visa ser simples e flexível para desenvolvimento de aplicações para as plataformas móveis Android, iOS e Windows. Seus principais objetivos são: (a) incluir dispositivos apontadores na interação com *tablets* usados por pessoas com deficiência motora e/ou cognitiva; e (b) especificar um *framework* para a concepção de aplicações específicas para este público.

A solução busca ampliar a forma natural de interação, permitindo a construção de aplicações que ofereçam aos usuários vários métodos de entrada de dados (Gestos,

¹⁰Dispositivo físico, conectado a uma porta do computador.

*Switch Scanning*¹¹ e Escaneamento de toque), os quais permitem ao usuário adotar a forma de interação mais adequada à sua necessidade. A solução possui um componente denominado *ScreenNavigator*, responsável por construir/desconstruir as telas e gerenciar o fluxo da aplicação.

O ATLab provê também um editor para a criação de aplicações com interface amigável, permitindo que pessoas sem experiência em programação consigam produzir jogos. Como estudo de caso, o trabalho descreve o desenvolvimento do *Switch Trainer*, um jogo para ensinar os usuários a manusear o dispositivo *Switch Scanning*. A Figura 18 apresenta algumas telas desse jogo.



Figura 18 – *Switch Trainer*, jogo criado a partir do ATLab.

Fonte: MIESENBERGER et al. (2014).

O *Switch Trainer* foi testado por um grupo de usuários com diferentes deficiências físicas, para verificar a utilidade dos métodos de entrada fornecidos pelo ATLab.

3.2.3 GNomon

Um projeto que propõe uma solução também voltada ao desenvolvimento de jogos assistivos é descrita por ACED LOPEZ; CORNO; DE RUSSIS (2015). Segundo os seus autores, o GNomon é um framework que permite a criação de jogos utilizando como forma de interação o dispositivo de um botão (*one-switch device*). Aplicações desenvolvidas por meio dessa ferramenta são destinadas a crianças com deficiências motoras severas que não conseguem desfrutar dos jogos disponíveis no mercado, uma vez que estes exigem maior destreza para realizar comandos complexos, seja por meio de *joysticks* dos consoles, ou por meio de teclados de computador.

O GNomon, é baseado no modo de interação de um botão NOMON ((BRODERICK; MACKAY, 2009) *apud* (ACED LOPEZ; CORNO; DE RUSSIS, 2015)), o qual permite selecionar um ou mais elementos da tela sem qualquer outro tipo especial de hardware ou configurações adicionais. Esse método associa um relógio a cada

¹¹Dispositivos de entrada, compostos por um ou dois botões conectados via Bluetooth ou USB.

elemento selecionável na tela. Todos os relógios possuem duas marcações: uma vermelha, fixada ao meio-dia; e outra preta girando à mesma velocidade em todos os elementos (em pontos diferentes). Para selecionar um elemento, o usuário deve pressionar a chave do *one-switch device* quando as marcações se cruzarem. Como estudo de caso, o artigo descreve o desenvolvimento de dois jogos: *Ladybugs* e *Invaders*, cujas telas são destacadas na Figura 19.



Figura 19 – Jogos *Ladybugs* e *Invaders*, criados a partir do GNomon.
Fonte: ACED LOPEZ; CORNO; DE RUSSIS (2015).

Os jogos foram projetados com a colaboração de um conjunto de especialistas, responsáveis também pela sua validação inicial.

3.2.4 STTK (*Speech-To-Text plus Kinect*)

O trabalho de SALNIKOV et al. (2014) trata de uma *Natural User Interface* (NUI) multimodal, o STTK. Tem como objetivo melhorar a experiência de acesso ao computador, por usuários com deficiência motora nos membros superiores. Segundo o autor, os fatores que transformam uma GUI em uma NUI são as formas de interação mais naturais pelas quais um usuário interage com essa interface. A solução combina dois tipos de entrada de dados: gestos e voz.

O STTK utiliza tecnologias como *Speech-To-Text* (STT), *Text-To-Speech* (TTS), e reconhecimento de gestos, sendo baseado na sinergia de dois protótipos desenvolvidos previamente, o *MailMe* e o *ChatOn*. O primeiro é uma aplicação que utiliza um padrão para acessibilidade de movimentos de mãos por meio do *Microsoft Kinect*, enquanto que o segundo é a implementação *front-end* de um módulo para STT. A arquitetura do STTK, apresentada na Figura 20, utiliza o conceito de componentes organizados em camadas, isolando as responsabilidades dos diversos elementos que processam os diferentes tipos de sinais e a apresentação da aplicação.

Ela possibilita a execução concorrente dos módulos de reconhecimento de voz, gestos e geração da GUI. Em termos de avaliações, a solução foi submetida a dois ciclos de testes por um usuário que sofre de distúrbios nos membros superiores. Esta avaliação permitiu a evolução da solução por meio de melhorias efetuadas após o *feedback* da primeira etapa de testes. A partir do retorno recebido, o STTK foi submetido a um conjunto de modificações que objetivaram aprimorá-lo.

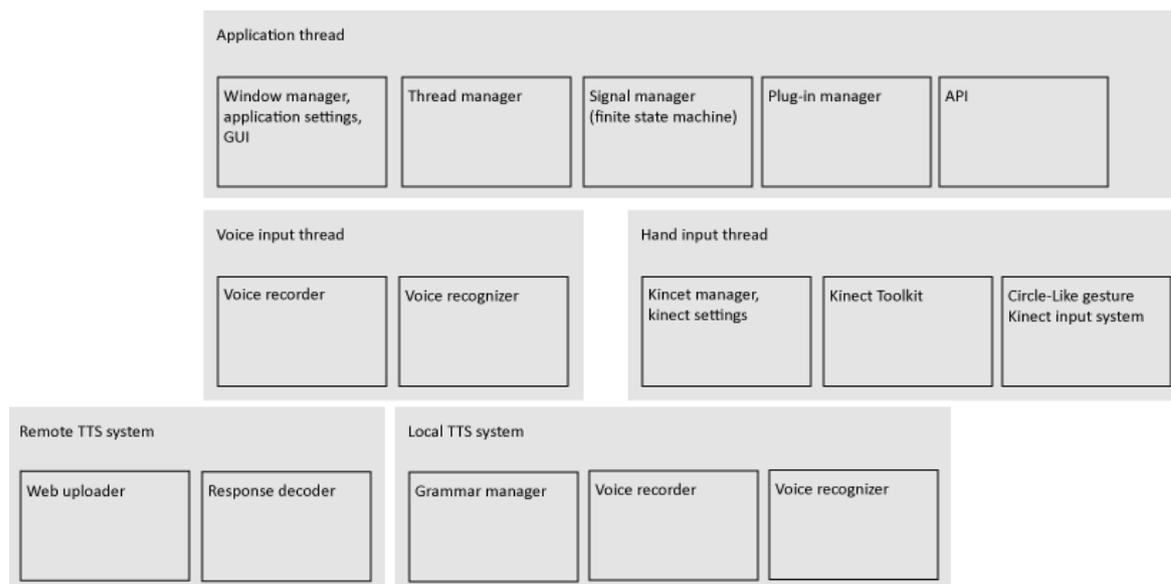


Figura 20 – Arquitetura em camadas do STTK.

Fonte: SALNIKOV et al. (2014).

3.2.5 FHCI (*Facial Human-Computer Interface*)

ANTUNES et al. (2016) apresentam uma arquitetura projetada para auxiliar os usuários a manusear o computador sem ajuda das mãos. O *Facial Human-Computer Interface* (FHCI) utiliza um esquema de controle *online* combinando a posição do cursor com a intenção visual do usuário. O FHCI é composto por uma interface embarcada de tempo real, sem fio, que substitui os HID clássicos, como *mouse* ou teclado. A interface também registra dados sobre eventos como posição corrente do cursor, valores dos botões e coordenadas de intenção visual do usuário, além de dar suporte à adaptação de outros sensores para simular ações dos botões. A arquitetura de contexto do FHCI é destacada na Figura 21.

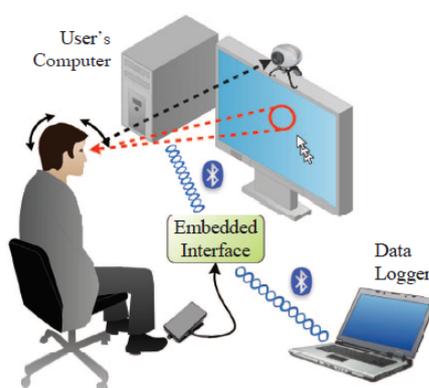


Figura 21 – Arquitetura de contexto do FHCI.

Fonte: ANTUNES et al. (2016).

O protótipo é composto por uma aplicação de rastreamento de face que combina dados captados a partir de uma *webcam* conectada ao computador do usuário para

mover o cursor na tela. Um segundo computador também pode se conectar à interface embarcada para registrar dados. A arquitetura de componentes do FHCI pode ser vista na Figura 22.

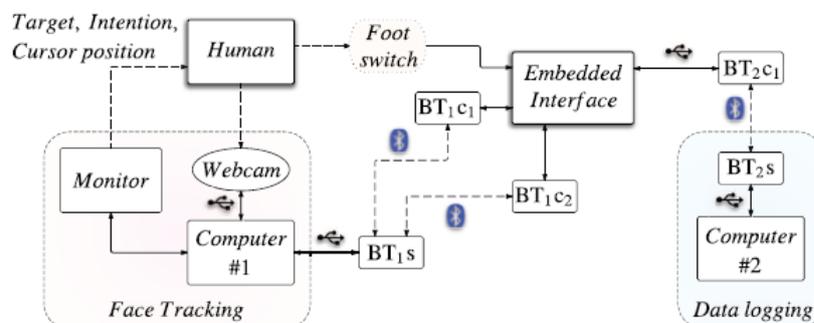


Figura 22 – Arquitetura do FHCI.
Fonte: ANTUNES et al. (2016).

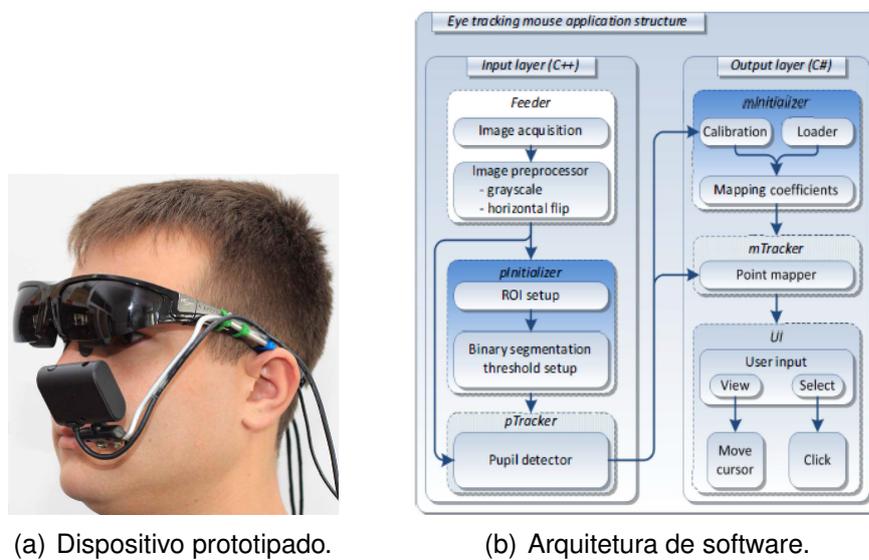
O FHCI foi testado por 10 participantes sem nenhuma deficiência, os quais executaram tanto tarefas de movimentação de cursor, quanto de digitação no teclado virtual visando avaliar a sua usabilidade.

3.2.6 ETM (*Eye Tracking Mouse*)

De acordo com LUPU; UNGUREANU; BOZOMITU (2012) *apud* LUPU; UNGUREANU; SIRITEANU (2013), o ASISTSYS é uma aplicação que simplifica o acesso ao computador, implementada em um sistema embarcado, o qual utiliza um dispositivo móvel para executar um algoritmo para rastreamento de olhar por vídeo.

LUPU; UNGUREANU; SIRITEANU (2013) propõem o ETM, um protótipo que estende o ASISTSYS, considerando *feedbacks* recebidos sobre o mesmo. O ETM é uma aplicação que utiliza um óculos de vídeo, uma *webcam* e um software específico executando um algoritmo de acompanhamento de olhar. O dispositivo utilizado no protótipo do ETM pode ser observado na Figura 23 (a). A inovação, frente ao ASISTSYS, está no uso dos óculos de vídeo em substituição ao monitor de computador. O uso dos óculos faz com que a posição da cabeça não afete o funcionamento do algoritmo de rastreamento ocular após a calibração. O artigo foca no método de rastreamento de olhar, permitindo que o usuário use o ETM para navegar por uma GUI própria utilizando apenas seus olhos.

A arquitetura do ETM é composta por duas camadas principais (*input* e *output*), conforme pode ser visto na Figura 23 (b). A camada de *input* é formada por três módulos: *Feeder*, *pInitializer* e *pTracker*, responsáveis por capturar os dados necessários para detectar os movimentos de pupilas dos usuários, os quais determinam as coordenadas do *mouse*. A camada de *output* determina como a informação que chega da camada de *input* é processada. O módulo *Point mapper* calcula as novas coordenadas do cursor, baseado nos dados de posição das pupilas provenientes da webcam,



(a) Dispositivo prototipado.

(b) Arquitetura de software.

Figura 23 – ETM.

Fonte: LUPU; UNGUREANU; SIRITEANU (2013).

realizando a movimentação desejada. O ETM foi submetido a testes de usabilidade com 20 usuários.

3.2.7 Universal AT with Multimodal Input and Multimedia Output Interfaces

KARPOV; RONZHIN (2014) apresentam um modelo conceitual de uma arquitetura de hardware e software para uma TA Universal. A denominação universal se dá pelo fato de permitir que usuários com diferentes deficiências possam utilizar a proposta por meio de alguma de suas formas de interação. O modelo, apresentado na Figura 24, prevê um módulo central que se relaciona com uma camada composta por métodos capazes de realizar processamento automático de áudio, vídeo, reconhecimento de texto e síntese de voz.

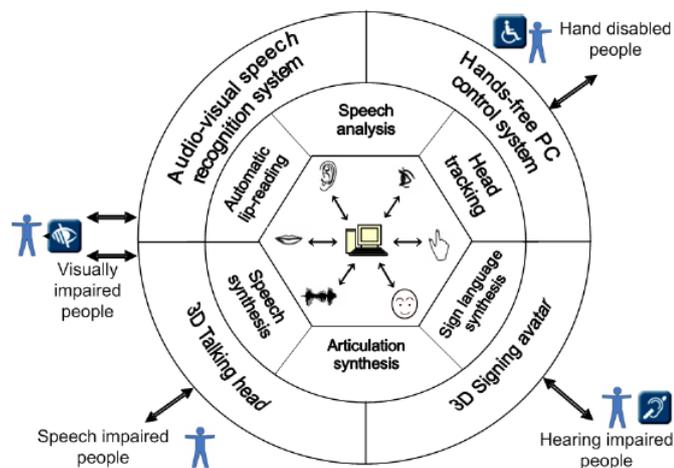


Figura 24 – Modelo conceitual da TA Universal.

Fonte: KARPOV; RONZHIN (2014).

Já a camada mais externa é constituída por interfaces de usuários que provém funções para: reconhecimento automático de áudio, vídeo e fala; síntese de fala (*Text-To-AudioVisual*); síntese visual de linguagem de sinais; e controle de computador *hands free*. A arquitetura do projeto foi concebida em cinco níveis, conforme destacado na Figura 25.

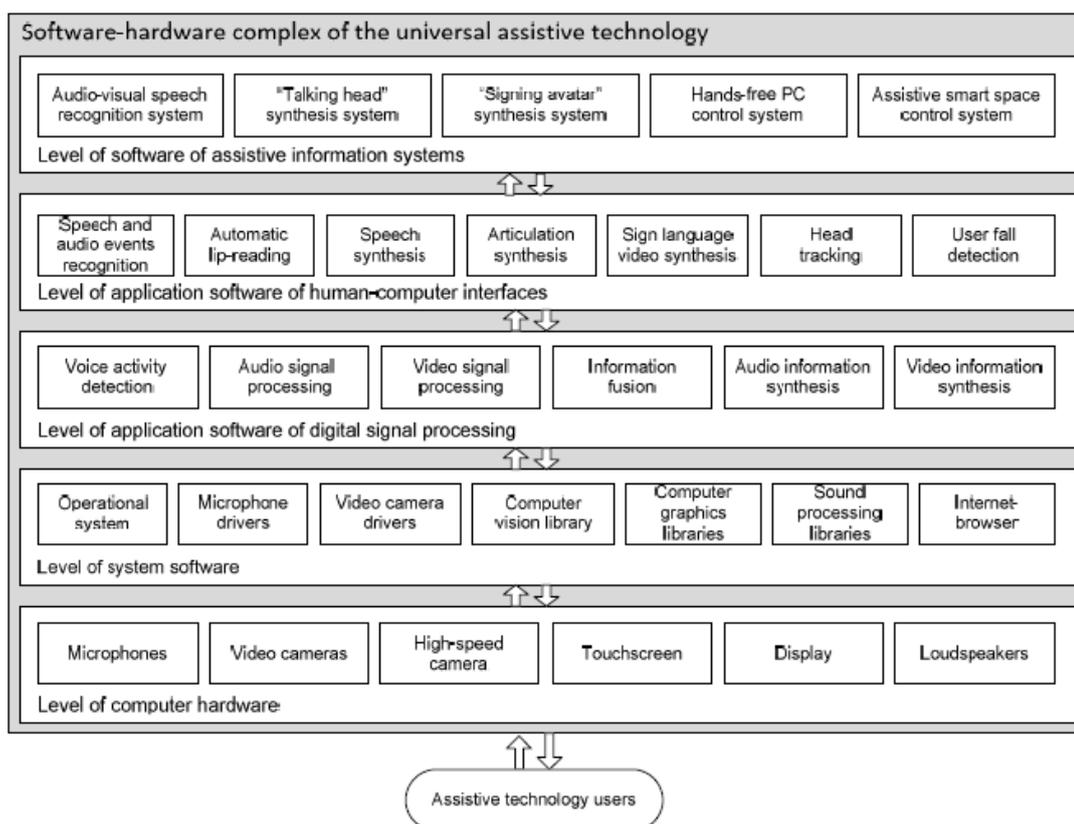


Figura 25 – Arquitetura em camadas da TA universal.
Fonte: KARPOV; RONZHIN (2014).

1. *Nível de hardware:* sensores de entrada e dispositivos de saída conectados a um servidor, tais como microfones, câmeras de vídeo, alto-falantes e telas *touchscreen*;
2. *Nível de software de sistema:* sistema operacional (*Windows*), *drivers* dos dispositivos, navegador web e bibliotecas de processamento específicas;
3. *Nível de software de aplicação de processamento de sinais:* módulos de detecção de voz, processamento de vídeo e áudio, fusão de informação e síntese de informação visual;
4. *Nível de software de aplicação de IHC:* sistemas de reconhecimento de fala e áudio, leitura labial, TTS, síntese visual de mímica e articulação, síntese visual de linguagem de sinais, *head tracking* baseado em vídeo e detecção de queda de usuário; e

5. *Nível de sistema de informação assistivo*: sistema de reconhecimento de fala AVSR (*Audio-visual Russian Speech*), de síntese "Talking Head", "Signing avatar", controle de computador *hands-free* ICANDO (*Intellectual Computer Assistant for Disabled Operators*) e controle de espaço assistivo.

A solução vislumbra a possibilidade de integração de módulos que usem outros dispositivos de TA, tanto para entrada, quanto para a saída de dados.

3.2.8 Cloud-based AT

MULFARI; CELESTI; VILLARI (2015) apresenta o projeto, implementação e avaliação de uma arquitetura que utiliza computação em nuvem para acessar software baseado em TA. O trabalho discute o impacto de utilizar SaaS (*Software as a Service*) com aplicações de TA acessíveis via web, especificando uma arquitetura composta pelos seguintes elementos, destacados na Figura 26:

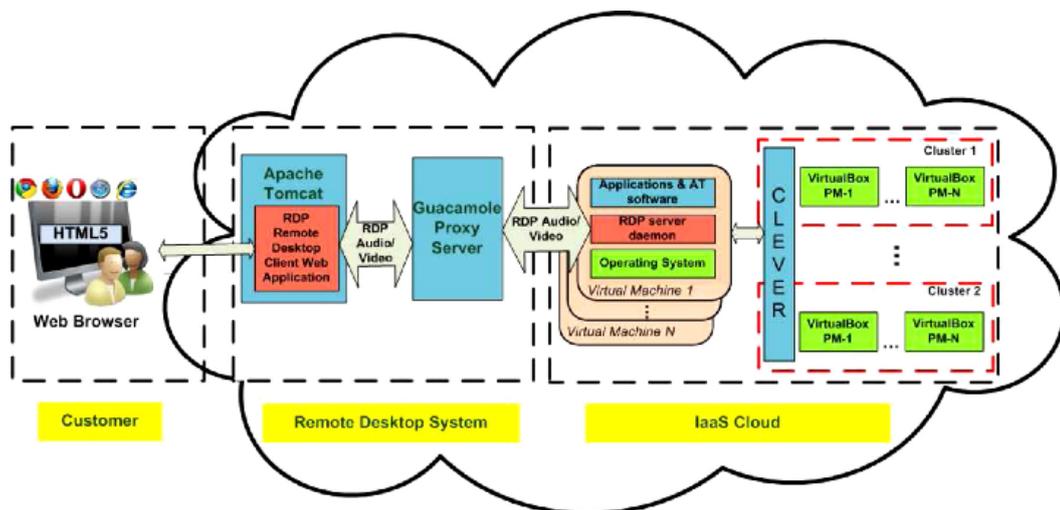


Figura 26 – Componentes da arquitetura proposta.

Fonte: MULFARI; CELESTI; VILLARI (2015).

- Cliente web para acessar a VM (*Virtual Machine*) onde as aplicações voltadas a TA estão instaladas;
- Aplicação web RDP (*Remote Desktop Protocol*), a qual permite ao usuário acessar VM via navegador;
- Servidor de *proxy* Guacamole que age como uma interface entre a aplicação cliente e o servidor RDP executando qualquer VM particular;
- VM executando em diferentes servidores físicos remotos por meio de uma *VirtualBox*;

- VIM (*Virtual Infrastructure Manager*) organizado com o *middleware* CLEVER para gerir a infraestrutura virtual.

Por meio desta infraestrutura, a proposta permite que o usuário acesse suas ferramentas de TA remotamente, evitando a necessidade de instalação de software especial na máquina utilizada. A solução foi submetida a testes de usabilidade considerando o tempo entre a entrada de dados (clique do *mouse*, por exemplo) e a saída de dados provida por alguma ferramenta assistiva acessível pela nuvem.

3.2.9 *My-World-in-My-Tablet (MWiMT)*

Em CARUSO et al. (2013) é apresentado o MWiMT, uma plataforma modular centrada no usuário, na qual um *tablet* é utilizado como contêiner de integração de um conjunto de tecnologias. O MWiMT dá suporte a dois tipos de entradas de dados: *input* de dispositivos de hardware (controles adaptados, por exemplo) e dados provenientes de bio-sinais. A arquitetura do MWiMT, apresentada na Figura 27, destaca os módulos responsáveis por tratar estas entradas de dados em destaque.

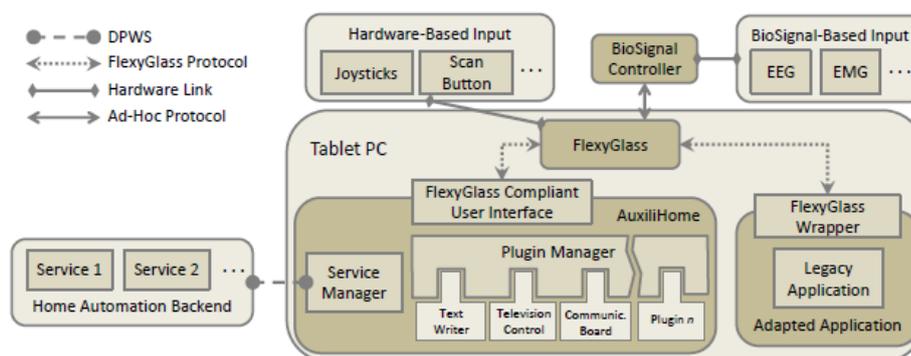


Figura 27 – Arquitetura do MWiMT.

Fonte: CARUSO et al. (2013).

O MWiMT usa o *tablet* como centralizador, no qual são executados os seus componentes de software principais:

- *AuxiliHome* - Utilizado para controle residencial por meio de aplicações que fornecem funcionalidades para usuários com deficiências, acessíveis em uma GUI própria; e
- *FlexyGlass* e adaptadores - módulo de software que torna o tipo de *input* transparente para a aplicação controlada.

O trabalho em si foca no uso de BCI (*Brain Computer Interface*) como dados de entrada, utilizando o BCI2000 para traduzir bio-sinais em comandos para a aplicação. Foram executados testes com três usuários finais (de diferentes níveis de deficiência física) seguindo um protocolo que incluía tarefas de controle de ambiente residencial visando medir o grau de satisfação no uso do MWiMT.

3.2.10 Protótipo multimodal

Por sua vez, RANTANEN et al. (2011) propõe uma solução para acesso a computador, voltada a pessoas com mobilidade restrita nas mãos. Utiliza um protótipo que combina o rastreamento de olhar, baseado em vídeo, com detecção de movimentos faciais (focado nas sobrancelhas). Esse dispositivo realiza o rastreamento de olhar utilizando uma fonte de comando de seleção integrado.

O protótipo foi submetido a testes de carga, visando comprovar que esta combinação de entradas de dados é um caminho promissor como forma alternativa de uso do computador. No entanto, avaliações somente com indivíduos sem deficiências e a constatação de problemas de desempenho do algoritmo de detecção de pupilas evidenciaram que o protótipo necessita evoluir para se tornar uma solução viável.

3.2.11 dTDS (*dual-mode Tongue Drive System*)

O TDS consiste em uma interface vestível e sem fio, desenvolvida para permitir que indivíduos controlem ambientes, usem computadores e até mesmo conduzam cadeiras de rodas motorizadas usando os movimentos da língua. No entanto, o uso exclusivo da língua para controle da solução eventualmente causava desconforto, fadiga ou mesmo espasmos aos seus usuários. Buscando amenizar estes problemas, o trabalho de HUO et al. (2013) propõe o dTDS (*dual-mode Tongue Drive System*) com o objetivo de expandir o TDS permitindo o uso de comandos por voz.

Assim, além de detectar o movimento da língua do usuário por meio de rastreador e um conjunto de sensores magnéticos embarcados em um *headset*, o dTDS também captura comandos de voz do usuário utilizando um microfone embutido no mesmo *headset*. O dTDS foi testado com 14 usuários sem limitações e 3 usuários com deficiência física. Os resultados mostraram que o dTDS melhora significativamente o tempo de execução de tarefas no computador. Os autores apontam, como próximos passos, a adição de novas formas de entradas de dados como, por exemplo, captação de movimentos da cabeça para execução de comandos de direção.

3.2.12 DOSVOX Gyro & microFênix

CRUZ et al. (2015) propõem duas extensões de software, utilizando o dispositivo Emotiv EPOC¹². A primeira solução, denominada DOSVOX Gyro, consiste em uma extensão do DOSVOX¹³ que permite a navegação pelos menus do sistema por meio dos movimentos da cabeça que indicam as direções capturadas. Esta funcionalidade adapta DOSVOX, inicialmente destinado a deficientes visuais, para uso de pessoas com deficiência motora.

¹²Headset composto por quatorze sensores, que capturam sinais EEG, e um sensor giroscópio MEMS (*Micro-Electro-Mechanical Systems*) que capta as direções de movimentos.

¹³Sistema baseado no uso de síntese de voz para facilitar o acesso a computadores.

A segunda solução descrita no trabalho consiste no aprimoramento do microFênix. A aplicação simula o *mouse* e o teclado, permitindo o controle de computadores (ambiente *Windows*), a partir de sons, capturados por microfones, ou sopros, captados por sensores específicos. A Figura 28 destaca o funcionamento do microFênix utilizando o EPOC. Esta evolução atualiza a interface visual do microFênix, fornecendo entrada de dados por meio do EPOC para a execução de tarefas. O artigo menciona que foram realizados testes com as duas ferramentas propostas. No entanto, não informa como os testes foram feitos, tampouco o público que efetuou as avaliações.

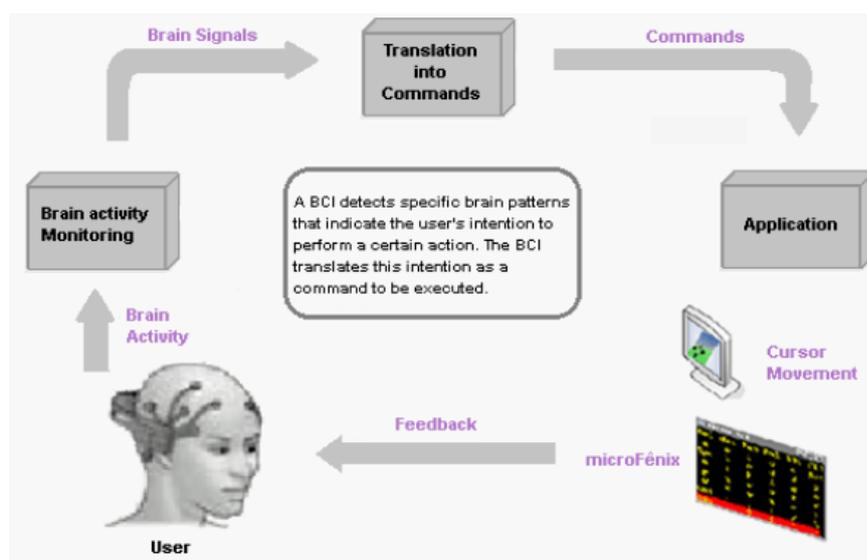


Figura 28 – Operação BCI com o microFênix.
Fonte: CRUZ et al. (2015).

3.2.13 UserTracking

O artigo MARTINS; RODRIGUES; MARTINS (2015) propõe uma interface multimodal que utiliza o *MS Kinect* para controlar o computador por meio de movimentos da cabeça, sobrancelhas (cima/baixo) ou boca (abrir/fechar) e comandos por voz.

A aplicação, denominada *UserTracking*, foi desenvolvida utilizando o *Xbox 360* e sensores *Windows*. A aquisição de dados de áudio, cores e profundidade é realizada por meio do *Kinect*, colocado sobre a tela do computador. A posição do cursor do *mouse* é controlada pelo movimento da face por meio do algoritmo *FaceTracking*. Os botões do *mouse* podem ser simulados usando tanto os movimentos da boca quanto das sobrancelhas dos usuários. Além deste monitoramento facial, as ações de *mouse* também podem ser disparadas por meio de comandos de voz. A aplicação foi testada por 8 usuários, buscando avaliar quais movimentos os usuários tinham maior dificuldade de execução, além do nível de conforto da solução.

3.2.14 WPan

O artigo de PAUL et al. (2013) propõe o projeto e a prototipação de uma interface para controle de ambientes, utilizando tecnologias sem fio como *Zigbee* ou *Bluetooth* e que pode ser comandada a partir de diferentes métodos de interação, como por exemplo, toques ou movimentos da língua. A interface descobre automaticamente utensílios domésticos que entendem protocolos de tecnologias sem fio presentes no ambiente, apresentando os serviços com suporte nesses dispositivos em uma GUI própria. A Figura 29 ilustra a proposta, destacando o foco do artigo na caixa tracejada.

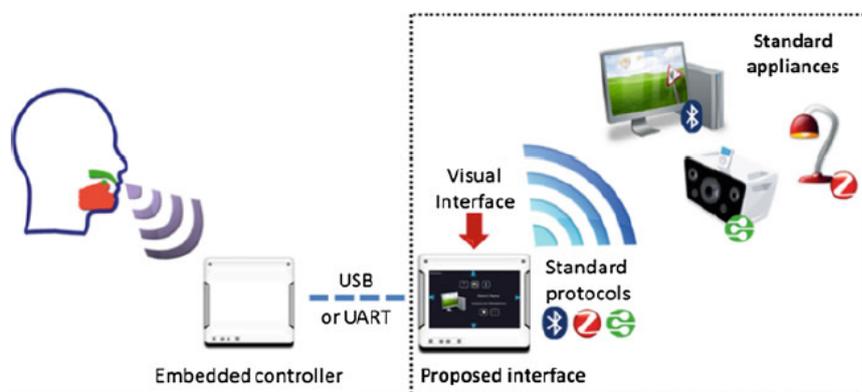


Figura 29 – Interface para controle de dispositivos em um ambiente assistivo *wireless*.
Fonte: PAUL et al. (2013).

Com relação à arquitetura, a aplicação foi organizada em três pacotes (*Core*, *GUI* e *Communication*), provendo separação entre os diferentes componentes que formam a solução. Além disso, foram criados *plugins* para a descoberta e a comunicação com os dispositivos via protocolos *Zigbee* e *Bluetooth*. A arquitetura SOA (vide Tabela 2) foi adotada para simplificar a descoberta dos dispositivos e o acesso aos serviços disponíveis (DHARA; DHARMALA; SHARMA, 2015).

A solução foi avaliada por usuários sem deficiência, os quais realizaram testes funcionais sobre a interface. A aplicação obteve êxito em descobrir dispositivos disponíveis, permitir troca entre dispositivos e acessar aplicações por meio de diferentes protocolos. Apesar de possuir um viés assistivo, permitindo controle de ambientes por meio de formas de interação não convencionais, o artigo se concentra mais na questão de descoberta e disponibilidade de dispositivos que serão apresentados no ambiente assistivo.

3.2.15 SBC (*Single Board Computer*)

O trabalho de MULFARI et al. (2014) parte das premissas que o uso de TA está diretamente ligado ao computador no qual ela está instalada, e que a configuração adequada de um ambiente para usuários com deficiência é uma tarefa difícil de ser realizada. Assim, o artigo propõe uma ferramenta que funciona como um adaptador

entre equipamento assistivo e os computadores utilizados pelos usuários.

A solução foi projetada para funcionar em qualquer dispositivo computacional que dê suporte a *mouses* ou teclados compatíveis com o protocolo HID. O sistema explora uma conexão USB para fazer a interface com um dispositivo final (computador, por exemplo), enquanto usa o protocolo HID para simular ações do cursor e movimentos no computador. A Figura 30 destaca os componentes da solução descrita no trabalho.

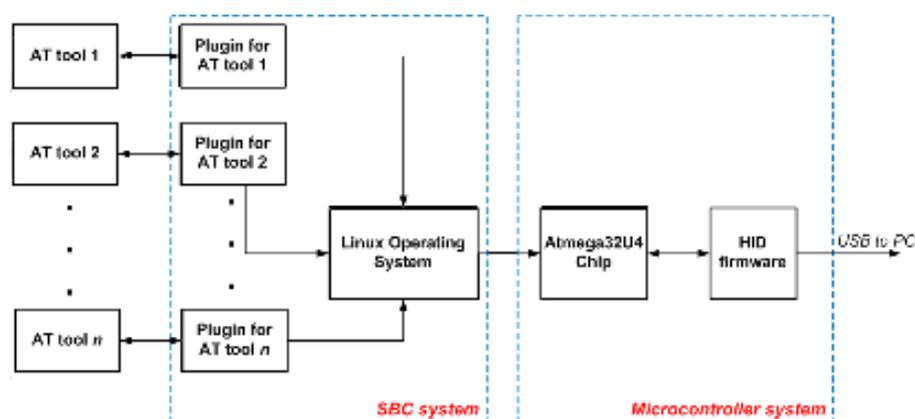


Figura 30 – Diagrama de componentes do SBC.

Fonte: MULFARI et al. (2014).

O objetivo é avaliar como sistemas embarcados, baseados em plataformas SBC, podem interagir com soluções de software de TA, executadas em um dispositivo pessoal. Um protótipo composto por uma *Raspberry Pi* (RPi) e uma placa controladora foi desenvolvido. O RPi interage com os equipamentos assistivos e processa os dados provenientes deles. Por sua vez, a placa controladora gerencia a comunicação com o dispositivo assistivo via USB.

A interação entre a RPi e equipamentos de TA requer o desenvolvimento de *plugins* específicos para processar dados a partir de cada ferramenta. Como estudo de caso, foi desenvolvido um *plugin* que interage com um aplicativo Android que explora o serviço de reconhecimento de fala *online* do Google. O aplicativo recebe a palavra reconhecida pelo serviço e envia um comando para o RPi que traduz a entrada em movimentos do cursor do computador. O artigo, no entanto, não descreve testes sobre o protótipo desenvolvido.

3.2.16 hBCI (*Hybrid Brain Computer Interface*)

O principal objetivo de MCCULLAGH; GALWAY; LIGHTBODY (2013) é propor um sistema multimodal para controle de ambiente virtual. A solução, denominada *Hybrid BCI* (hBCI), combina o envio de sinais EEG com interação baseada no rastreamento de olhar. A ideia é estender o projeto BRAIN, o qual funciona a partir de envio de sinais EEG usando interfaces SSVEP ou ERD/ERS (*Event Related Desynchronization/Event Related Synchronization*). A arquitetura modular do BRAIN,

permitiu a sua evolução para um híbrido, combinando sinais EEG com dados proveniente do *eye-tracking*, para atuar sobre uma interface gráfica de controle de ambiente. A arquitetura do hBCI é destacada na Figura 31.

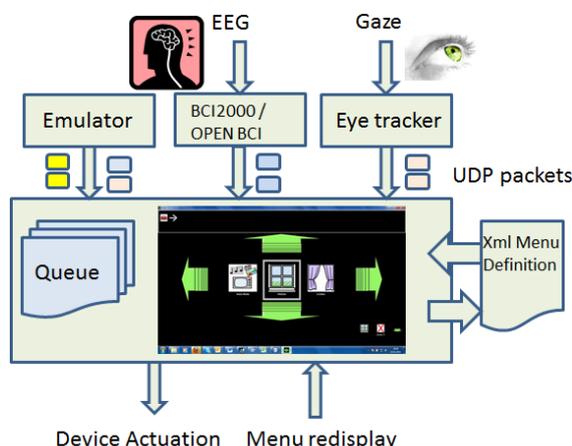


Figura 31 – Arquitetura do hBCI.

Fonte: MCCULLAGH; GALWAY; LIGHTBODY (2013).

A GUI, desenvolvida em Java para ser independente de sistema operacional, comunica-se com os módulos de sinais de EEG e *eye tracking*, os quais por sua vez enviam os dados por meio do protocolo de comunicação UDP (*User Datagram Protocol*). O sistema conta com ferramentas de emulação que permitem testar a aplicação por meio da geração de sequências de sinais ERS/ERD, SSVEP e de *eye-tracking*.

O sistema utiliza operação SSVEP com 4 LED montadas ao redor do computador para permitir a sua navegação. Isso ocasiona o problema de alguns usuários não conseguirem ter a visão completa da tela já que precisam focar nas LED ao redor do computador, tornando a navegação confusa. No momento de escrita, a integração do sistema BCI com o de rastreamento de olhar encontrava-se em andamento, não havendo testes executados sobre a solução proposta. Mesmo assim, foram percebidos dois problemas na solução: o uso da ferramenta Tobii¹⁴ encarece o preço da aplicação, dado seu custo. Além disso, o fato da sua API não ser *open-source*, reduz as possibilidades de ampliação da solução.

3.2.17 SMARTUNIVERS

O trabalho de ALY et al. (2015) apresenta o SMARTUNIVERS, uma interface para pessoas com diferentes deficiências físicas, projetada para integrá-las ao ambiente de trabalho. A solução é composta por dois módulos principais:

- *SMARTHELP* - provê ajuda a respeito de localização e fornece serviços de comunicação por VOIP (*Voice over IP*); e

¹⁴<https://www.tobii.com/>

- *SMARTEDIT* - interface de edição multimodal.

Os módulos propostos usam tecnologias para reconhecimento de fala, TTS, teclado/*mouse* virtuais e teclados braille para uso no ambiente de trabalho. O artigo destaca onze combinações de deficiências que são contempladas pela aplicação. Para cada grupo, diretrizes específicas são usadas para adaptar a interface dinamicamente.

3.2.18 *Fire Bird V*

Em TRIPATHY; RAHEJA (2014) é descrito o desenvolvimento de uma aplicação para controlar o robô *Fire Bird V*, para que ele seja utilizado como assistente para pessoas com deficiência. O robô recebe comandos a partir de sinais EEG enviados por meio do dispositivo EPOC por meio de uma GUI específica. Esta GUI é composta por botões de direção que são acionados pelos sinais cerebrais. Quando clicados, os botões disparam instruções ao robô que executa os movimentos desejados. A Figura 32 destaca o diagrama de implementação deste protótipo.

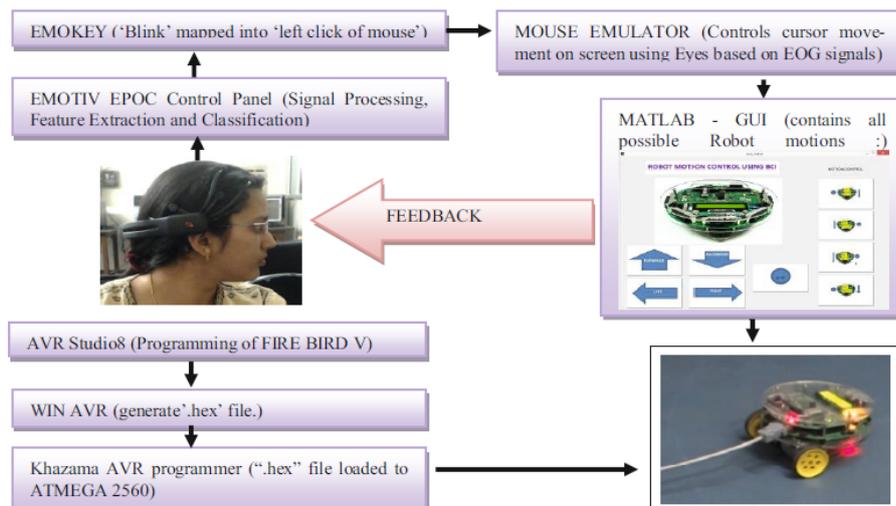


Figura 32 – Diagrama de implementação para controle do robô via sinais BCI.

Fonte: TRIPATHY; RAHEJA (2014).

O artigo menciona que o protótipo foi avaliado, no entanto não especifica o protocolo de testes, nem os usuários envolvidos.

3.2.19 *Human-Computer Interface Based On Iot Embedded Systems*

O artigo de MULFARI et al. (2015) explora conceitos da *Internet of Things* (IoT), para utilizar sistemas embarcados de baixo custo como forma de apoio à interação entre usuários com deficiência e computadores. O objetivo é criar soluções voltadas a TA que possam ser instaladas em um dispositivo embarcado que, quando conectado a um computador via USB, se comporte como um periférico HID permitindo que o usuário utilize a TA sem necessidade de configurações extras. Uma vez que pessoas com

deficiências físicas geralmente interagem com computadores por meio de dispositivos específicos que requerem ajustes, a ideia é que esta configuração possa ser obtida a partir deste equipamento IoT.

A arquitetura proposta utiliza dois componentes-chave: um microcontrolador Atmega32u4, usado para funcionar como um *mouse* nativo ao ser conectado a um computador via USB; e um processador usado para inicializar o Linux embarcado com *runtime* do *Python*, o que permite o uso desta linguagem para capturar e processar dados de dispositivos externos e sensores, além de possibilitar conexão com a Internet. O diagrama de contexto da solução é apresentado na Figura 33.

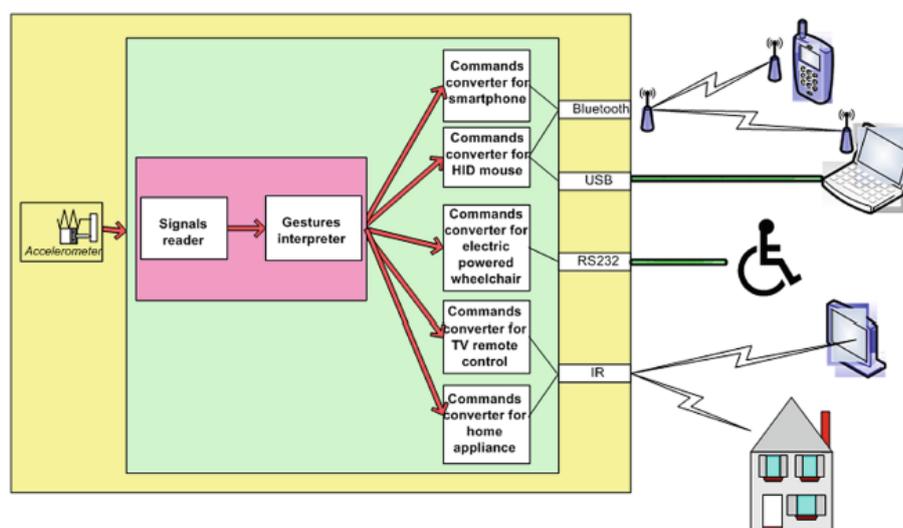


Figura 33 – Diagrama de contexto do sistema.

Fonte: MULFARI et al. (2015).

Como estudo de caso, o projeto detalha o desenvolvimento de um dispositivo de entrada alternativo baseado no acelerômetro MEMS e no microcontrolador Atmega32u4. O protótipo consiste em um óculos adaptado para indivíduos com deficiência motora para controlar o computador sem o uso das mãos. O trabalho de MULFARI; CELESTI; VILLARI (2015), aparentemente faz uso do sistema em nuvem proposto anteriormente, dando a ideia que os dois protótipos compõem um projeto maior, uma vez que parecem funcionar de forma integrada. O artigo não descreve testes com o protótipo criado.

3.2.20 AID (Adaptive Interface Design)

Em LIN et al. (2017) é discutida a AID, uma interface multimodal que provê assistência para uso do computador por pessoas com artrite. A Figura 34 apresenta uma visão geral do funcionamento da solução.

A AID é composta por três componentes principais:

1. Tobbi - sistema de *eye tracking* que substitui o *mouse* na tarefa de apontar o

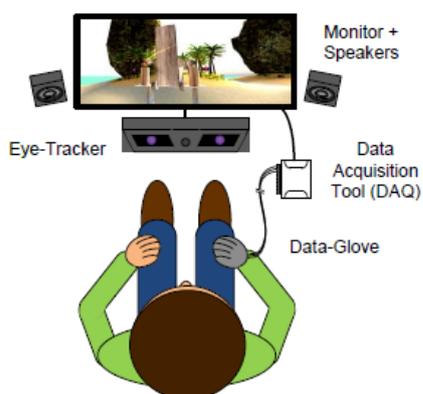


Figura 34 – Visão geral do AID.
Fonte: LIN et al. (2017).

cursor na tela;

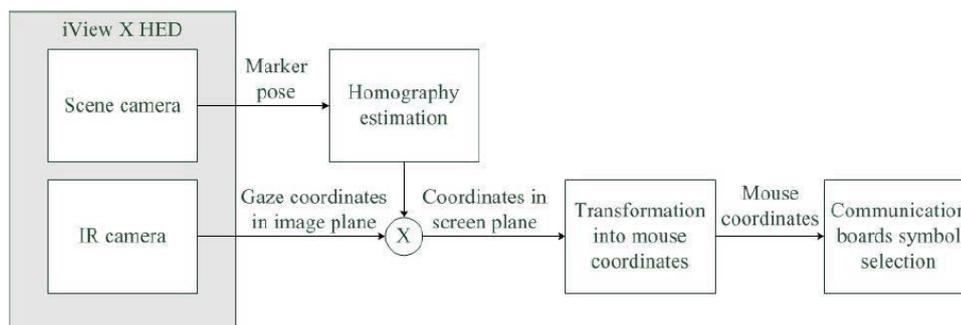
2. *Data glove* - luva especial desenvolvida para medir movimentos de dedos e força de preensão, que substitui o teclado para navegação e interação; e
3. O jogo em si (desenvolvido em *Unity*) - no qual o jogador vaga por uma ilha tropical sob a perspectiva de primeira pessoa.

O AID foi testado durante seus ciclos de desenvolvimento por usuários sem limitações físicas ou cognitivas. Na versão estável, 7 participantes com diferentes níveis de artrites nas mãos testaram a solução. Os testes visavam qualificar aspectos relacionados à experiência e à usabilidade do jogo em termos de efetividade, eficiência e satisfação.

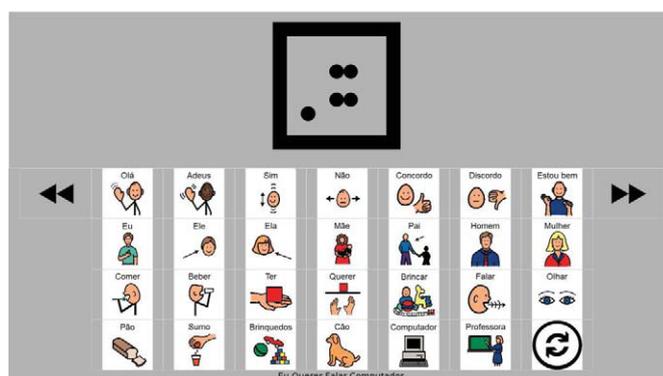
3.2.21 *Gaze-Based Interaction System for People with Cerebral Palsy*

O artigo de GALANTE; MENEZES (2012) propõe uma solução que utiliza técnicas de rastreamento de olhar. Esta técnica permite que usuários com paralisia cerebral consigam se comunicar por meio da escolha de símbolos existentes em uma placa de comunicação virtual. A aplicação utiliza um dispositivo colocado na cabeça (*SMI iView X HED*), composto por uma câmera que rastreia os olhos do usuário, e uma segunda câmera que identifica o símbolo mirado pelo usuário. O rastreador relaciona a direção do olhar com a imagem da cena vista, mas não com os objetos observados.

Para identificar qual símbolo que o usuário está olhando, a aplicação mapeia as coordenadas fornecidas pelo SMI para as coordenadas da placa de comunicação. Para realizar a seleção de símbolos escolhidos, o método inicial de entrada de dados foi o uso de um interruptor adaptado. A Figura 35 (a) mostra o diagrama de blocos da solução, destacando as etapas envolvidas no processo de captação de olhar, cálculo de coordenadas, transformação de dados e seleção de símbolos.



(a) Diagrama esquemático de blocos da aplicação.



(b) GUI da aplicação.

Figura 35 – Placa de comunicação.
 Fonte: GALANTE; MENEZES (2012).

Em termos de avaliações, o protótipo desenvolvido foi submetido a testes funcionais de usabilidade com 9 usuários sem deficiências. A GUI desenvolvida para simular a placa de comunicação é apresentada na Figura 35 (b), composta por 27 símbolos representando palavras e 3 símbolos de ação. As frases geradas aparecem na parte baixa da interface.

3.2.22 Beto na Floresta

SCARDOVELLI; FRÈRE (2015) apresentam um jogo voltado a TA como tema. O trabalho utiliza um dispositivo de acesso periférico e um jogo computadorizado 3D que não requerem ações de arrastar, clicar ou ativar vários botões ao mesmo tempo. O dispositivo de acesso periférico consiste em uma *webcam* e um sistema de supervisão que processa as imagens provenientes dela.

O jogo, denominado "Beto na Floresta", é executado na maior parte da tela do computador, enquanto que o sistema de supervisão é processado em segundo plano em uma janela de controle ao lado. Esta janela possui três regiões que podem ser ativadas pelo jogador, sendo as únicas partes analisadas pelo sistema de supervisão. Quando algum movimento é capturado nestas áreas, este dado é detectado, e um comando é enviado ao jogo 3D. Com base nesta informação, é definida a ação que

o usuário deseja tomar para avançar nas fases. O jogo adota o estilo de aventura no qual o jogador resolve problemas para superar obstáculos, tomando decisões e observando as consequências causadas.

O sistema de supervisão, descrito ao longo do trabalho, foi desenvolvido utilizando o padrão MVC (*Model-View-Controller*), realizando a separação entre as camadas de software. A Figura 36 apresenta o diagrama de componentes utilizado, destacando os componentes para captura e processamento de imagens, e os módulos de comunicação entre as aplicações.

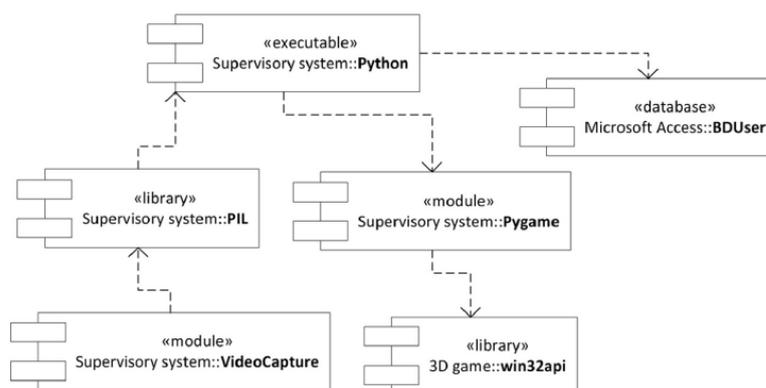


Figura 36 – Diagrama de componentes do jogo.

Fonte: SCARDOVELLI; FRÈRE (2015).

Para avaliação do sistema, foram utilizados teclado convencional e dispositivo periférico adaptado, por 20 voluntários sem deficiências e por 13 usuários com deficiências motoras nos membros superiores.

3.2.23 SINASense

O SINASense, proposto em (MANRESA YEE; MUNTANER; ARELLANO, 2013), é um suporte tecnológico e educacional que provê a seus usuários maior autonomia na interação com o ambiente no qual estão inseridos. O SINASense é um sistema que oferece estímulo sensorial baseado nos movimentos do usuário, detectando e rastreando a movimentação por meio de uma *webcam* e uma fita colorida colocada na mão do usuário.

O público alvo são crianças com deficiências de desenvolvimento, as quais dependem de outras pessoas para interagir com o ambiente. A aplicação foi submetida a testes com 7 usuários com deficiências físicas severas (alguns com paralisia cerebral) durante três meses. Como resultado, foram percebidos aumentos consideráveis de movimentos intencionais de todos os usuários, especialmente relacionados ao tempo de duração de movimentos.

3.2.24 Activelris

Já LEVY et al. (2013) descrevem o Activelris, um conjunto de ferramentas para acessibilidade que utiliza uma *webcam* para captar o movimento dos olhos dos seus usuários. Por meio de uma interface própria, o Activelris provê acesso ao computador, facilitando o acesso a redes sociais, *smartphone* e navegação na Internet, além de fornecer um certo nível de controle de ambiente residencial a seus usuários. Dentre suas principais funcionalidades, destacam-se:

- Interação com computador via controle de cursor;
- Teclado virtual customizado para facilitar a digitação;
- Integração com a web por meio de navegador, ferramenta de busca, visualização de *feeds* RSS e previsão de tempo;
- Interação com redes sociais (*Twitter* e *Facebook*);
- Controle de ambiente integrado, combinando *ZigBee* com transmissão infravermelha; e
- Aplicação android para integração com smartphones.

O Activelris foi testado por 8 voluntários (apenas um usuário com deficiência), divididos em três níveis. Os testadores realizaram uma série de tarefas definidas em um protocolo de testes, de forma a verificar o grau de usabilidade e efetividade das funções disponíveis na aplicação.

3.2.25 Doce Labirinto

Por fim, o artigo de CARDOSO et al. (2016) apresenta o Doce Labirinto, um jogo que utiliza dispositivos de interação natural e TUI (*Tangible User Interfaces*). A ideia é atravessar um labirinto físico com o Sphero¹⁵ o qual é controlado por meio de movimentos da cabeça captados pelo IOM. O jogo possui uma aplicação de controle que estabelece a comunicação entre o IOM e o Sphero. A aplicação se conecta ao IOM por meio de uma conexão serial, e ao *Sphero*, por meio de *Bluetooth*. Uma vez conectados, os movimentos realizados com o IOM são traduzidos em comandos para o *Sphero*.

Com relação a avaliações, o jogo foi submetido a uma carga de testes considerável durante a Fenadoce 2016¹⁶, onde os visitantes da feira puderam utilizá-lo. Vale ressaltar que este é um dos trabalhos criados ao longo do processo de concepção do arcabouço proposto por esta tese, o qual será detalhado no Capítulo 4.

¹⁵Robô em formato esférico. Disponível em <http://www.sphero.com/>

¹⁶Evento cultural e gastronômico realizado anualmente em Pelotas-RS.

3.3 Taxonomia para Soluções voltadas a TA

O primeiro resultado alcançado com a análise dos trabalhos científicos provenientes do MSL foi a proposição de uma taxonomia para as soluções desenvolvidas. A partir das observações realizadas, foram delineadas duas classificações: Tipo de Solução Desenvolvida e Funcionalidades das Soluções.

3.3.1 Classificação por Tipo de Solução Desenvolvida

Esta classificação agrupa os trabalhos de acordo com o artefato proposto ou desenvolvido. Nesta perspectiva, os estudos recuperados foram qualificados como:

- **Arcabouços de software** - propostas de componentes reutilizáveis, utilizados para simplificar o desenvolvimento de aplicações; e
- **Protótipos ou ferramentas específicas** - soluções para objetivos específicos, geralmente utilizadas para validação ou estudo de caso individual.

Dentro do universo analisado, quatro trabalhos recuperados se enquadraram como Arcabouços de Software, e são listados na Tabela 4

Tabela 4 – Trabalhos caracterizados como arcabouços no MSL.

Autores	Arcabouço
OSSMANN et al. (2014)	AsTeRICS
VEIGL et al. (2013)	
MIESENBERGER et al. (2014)	ATLab
ACED LOPEZ; CORNO; DE RUSSIS (2015)	GNomon

Fonte: Elaborado pelo Autor (2019).

Dentre estes, o AsTeRICS consiste em um arcabouço de software baseado em componentes, o qual utiliza conceitos de prototipação por meio de modelagem visual. Esta estrutura viabiliza a construção de soluções de TA sem a necessidade de conhecimento profundo de linguagens ou técnicas de programação. Considerando o processo evolutivo de construção de arcabouços de software (vide Figura 10), o AsTeRICS se caracteriza como uma ferramenta visual, uma vez que permite a geração de aplicações por meio da conexão de componentes, os quais produzem modelos responsáveis por originar as soluções desejadas.

Por sua vez, o ATLab e o GNomon são soluções destinadas à criação de aplicações dentro de um domínio mais específico, ou seja, jogos voltados a TA. Ambas as soluções têm características de *frameworks white box*, uma vez que para se desenvolver aplicações por meio delas, é necessário possuir um conhecimento razoável sobre o funcionamento e estrutura de seu código fonte.

Todos os demais trabalhos encontrados no MSL desenvolveram protótipos, utilizados em contextos particulares. Destinavam-se a resolver problemas específicos, não sendo desenvolvidos com o intuito de serem reutilizáveis ou customizáveis para diferentes cenários de aplicação.

3.3.2 Classificação por Funcionalidades das Soluções

Esta categorização qualifica os estudos de acordo com as funções primordiais que as soluções atacam. Sob esta perspectiva foram encontrados trabalhos que buscavam prover:

- *Acesso ao computador* - simplificam ou possibilitam a este dispositivo;
- *Controle de ambiente* - permitem a gestão do espaço no qual o usuário está inserido;
- *Jogos voltados a TA* - disponibilizam jogos adaptados à necessidades especiais; e
- *Soluções mistas* - aplicações que podem ser utilizadas para duas ou mais finalidades.

A Figura 37 destaca a distribuição dos trabalhos de acordo com a classificação por funcionalidades.

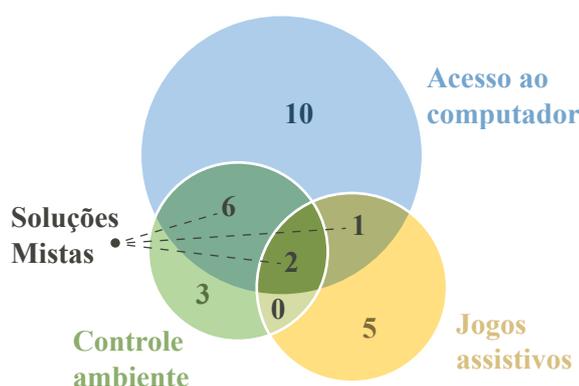


Figura 37 – Trabalhos do MSL, distribuídos por funcionalidades.

Fonte: Elaborados pelo Autor (2019), via ferramenta StArt.

A maior parte dos trabalhos encontrados busca viabilizar ou simplificar o Acesso ao Computador, adaptando dispositivos e desenvolvendo aplicações que privilegiam as habilidades preservadas dos usuários. Por sua vez, três trabalhos são projetados para trabalhar exclusivamente com Controle de Ambiente. O número de projetos nestas categorias é ampliado consideravelmente, quando consideradas as Soluções Mistas.

Por fim, cinco trabalhos desenvolvem jogos, tanto para entretenimento, quanto para reabilitação. Este número razoável de trabalhos nesta vertente vai ao encontro da quantidade de dispositivos de interação alternativa focados nesta fatia de mercado. O fato deles possuírem arcabouços específicos para incentivar desenvolvimento de jogos, simplifica a sua adaptação para outros horizontes que não o seu original, incluindo a sua utilização no contexto de TA. A Tabela 5 relaciona os trabalhos encontrados segundo a classificação por funcionalidade.

Tabela 5 – Trabalhos agrupados segundo classificação por *Funcionalidade*.

Funcionalidades	Trabalhos
Acesso ao computador	SALNIKOV et al. (2014); ANTUNES et al. (2016); LUPU; UNGUREANU; SIRITIANU (2013); KARPOV; RONZHIN (2014); MULFARI et al. (2015); RANTANEN et al. (2011); CRUZ et al. (2015); MARTINS; RODRIGUES; MARTINS (2015); MCCULLAGH; GALWAY; LIGHTBODY (2013); GALANTE; MENEZES (2012).
Controle de ambiente	PAUL et al. (2013); TRIPATHY; RAHEJA (2014); MANRESA YEE; MUNTANER; ARELLANO (2013).
Jogos assistivos	OSSMANN et al. (2012); SCARDOVELLI; FRÈRE (2015); CARDOSO et al. (2016); MI-ESENBERGER et al. (2014); ACED LOPEZ; CORNO; DE RUSSIS (2015).
Soluções mistas	CARUSO et al. (2013); HUO et al. (2013); MULFARI et al. (2014); MULFARI; CELESTI; VILLARI (2015); TRIPATHY; RAHEJA (2014); LEVY et al. (2013); LIN et al. (2017); VEIGL et al. (2013); OSSMANN et al. (2014).

Fonte: Elaborado pelo Autor (2019).

Durante as análises, foi perceptível que, apesar da classificação por funcionalidade, todos as soluções lidam, em certo momento, com alguma interface intermediária que mediava a interação entre usuário e aplicação. Esta interface tem dois propósitos principais:

- Efetuar configurações que permitam sua utilização; e
- Realizar/auxiliar as funções próprias da aplicação, para atingir seus os objetivos.

Assim, é possível afirmar que todas as propostas utilizam formas de interação com algum dispositivo computacional. A Tabela 6 apresenta um resumo sobre as características dos protótipos discutidos. São destacadas as tecnologias de implementação, os dispositivos e a forma de interação de cada solução.

Tabela 6 – Característica dos protótipos estudados no MSL.

Trabalho	Tecnologias	Dispositivos	Interação
SALNIKOV et al. (2014)	C, Google <i>Speech</i> API.	MS Kinect.	Voz. Gestos.

ANTUNES et al. (2016)	MIT <i>Processing</i> , OpenCV, Teclado virtual.	Interface embarcada de tempo real <i>wireless</i> .	Rastreamento de face. Acionamento mecânico. Sensores.
LUPU; UNGUREANU; SIRITEANU (2013)	OpenCV, C++/C#	Webcam, <i>video glasses</i> (óculos adaptado).	Rastreamento de olhar.
KARPOV; RONZHIN (2014)	C/C++, OpenCV, OpenGL, DirectX, HTK, Julius, MFC.	Câmeras. Microfone. <i>Headset</i> adaptado.	Reconhecimento de áudio e vídeo. Síntese de voz. Síntese visual. Movimentos da cabeça.
MULFARI; CELESTI; VILLARI (2015)	HTML5, CLEVER (<i>middleware</i>), API Guacamole, common-js.	TA específica do usuário.	Dependente de TA.
CARUSO et al. (2013)	Não informado.	Tablet, BCI2000.	Controles adaptados. Biosinais
RANTANEN et al. (2011)	MS C++, MS <i>Foundation Classes</i> (MFC), OpenCV, OpenEyes e bibliotecas Boost.	Óculos adaptados. Câmeras CMOS.	Rastreamento de olhar. Movimentos faciais.
CRUZ et al. (2015)	Não informado.	EPOC.	Voz. Biosinais.
HUO et al. (2013)	LabView, C.	<i>Headset wireless</i> . Sensores magnéticos. Microfones.	Comandos de língua. Voz.
MARTINS; RODRIGUES; MARTINS (2015)	MS Kinect SDK, C#, MS <i>Speech</i> API.	Microsoft Kinect, sensores.	Movimentos (cabeça, sobrancelhas, boca). Voz.
PAUL et al. (2013)	Kit Atmel's. Texas Instrument SDK, Qt Linux embarcado.	Microcontrolador. Módulo de LCD.	Movimentos de língua. Toque.
MULFARI et al. (2014)	Arduino bootloader. Bibliotecas <i>mouse</i> /teclado, reconhecimento de voz.	Microcontrolador Atmel Atmega32U4. Raspberry Pi.	Dependente de dispositivo.
MCCULLAGH; GALWAY; LIGHTBODY (2013)	Java. Python. OpenBCI. Tobii.	BCI2000. 4 LEDs para EEG (SSVEP). Webcam.	Sinais EEG. Rastreamento de olhar.
ALY et al. (2015)	<i>Engines</i> reconhecimento voz. TTS. Teclado/ <i>mouse</i> virtuais. Teclados braile.	Não informado.	Voz. Toque.

TRIPATHY; RAHEJA (2014)	C, ATMEL AVR Studio, Khazama AVR programmer, Emokey, MATLAB R2012b.	EPOC, FIRE BIRD V.	Sinais EEG.
MULFARI et al. (2015)	Python.	Microcontrolador, Óculos adaptado. Acelerômetro, Arduino.	Movimentos da cabeça.
OSSMANN et al. (2012)	AsTeRICS, teclado OSKA.	PS3. HID AsTeRICS. Dispositivo I/O de propósito geral. Wii mote, MS Kinect. Webcam.	Rastreamento (cabeça, boca e olhos).
SCARDOVELLI; FRÈRE (2015)	Python. VideoCapture. MS Access (PyByODBC). Python Imaging Library. PyGame e Blender.	Webcam. Dispositivo periférico adaptado.	Gestos.
LIN et al. (2017)	Tobii. Unity. LabVIEW.	Luva adaptada com sensores, Sistema NI-DAQ.	Movimento dos dedos. Rastreamento de olhar.
GALANTE; MENEZES (2012)	Não informado.	SMI iView X HED. Dispositivo montado na cabeça. Botão adaptado.	Rastreamento de olhar. Dispositivos de acionamento mecânico (pedais, botões).
MANRESA YEE; MUNTANER; ARELLANO (2013)	OpenCV. Biblioteca IrrKlang.	MS Kinect, Webcam. Fita colorida.	Movimentos (mãos ou braços).
(LEVY et al., 2013)	Java. C++. OpenCV. Android. IR. Zigbee.	Smartphone, Webcam.	Rastreamento de olhar.
(CARDOSO et al., 2016)	Java. Sphero-Desktop-API. jSSC. Bluecove	IOM, Sphero.	Movimentos da cabeça.

Fonte: Elaborado pelo Autor (2019).

3.4 Discussão dos resultados do MSL

Além de propor uma classificação e categorizar as soluções recuperadas pelo MSL, a análise desses trabalhos permitiu extrair dados relevantes sobre os mesmos. Esta observação permitiu traçar um panorama geral da área, com a identificação de técnicas e tendências adotadas.

Por exemplo, com relação aos tipos de arquiteturas, a maior parte dos trabalhos analisados apresenta traços arquiteturais baseados no modelo Cliente-Servidor, dividindo as estruturas em pelo menos duas camadas de software. As exceções são os projetos que adotam conceitos de SOA na sua concepção. Os artigos apresentam

diferentes formas de representação de suas arquiteturas. A opção pela utilização de diagramas de contexto para TA é natural, uma vez que os trabalhos utilizam estratégias de *design* e desenvolvimento centradas no usuário da aplicação. Nesta perspectiva, prover uma visão global da aplicação sob o ponto de vista do usuário, faz sentido para contextualizar e evidenciar os atores presentes na interação. As figuras 15, 21, 24, 28, 29 e 34 apresentam exemplos de arquiteturas representadas desta forma.

Por sua vez os trabalhos que representam suas estruturas por meio de diagramas de componentes organizados por camadas utilizam níveis de abstração e detalhamento que variam de acordo com o grau de especificidade desejado. Esta diagramação é bastante utilizada, facilitando aspectos que envolvem a compreensão do funcionamento e da comunicação entre os componentes das arquiteturas. As figuras 16, 20, 23 (b) e 25 destacam algumas dessas representações arquiteturais. Por fim, além destas formas de representação, uma terceira possibilidade é explorada por CARUSO et al. (2013). Nesse trabalho, uma visão por componentes agrupados de acordo com os contêineres nos quais serão executados, é adotada. A Tabela 7 agrupa os trabalhos segundo a representação utilizada.

Tabela 7 – Trabalhos agrupados por tipo de representação.

Diagrama	Trabalhos
Contexto	ANTUNES et al. (2016); CRUZ et al. (2015); PAUL et al. (2013); MULFARI et al. (2014); MCCULLAGH; GALWAY; LIGHTBODY (2013); MULFARI et al. (2015); LIN et al. (2017); CARDOSO et al. (2016); KARPOV; RONZHIN (2014); OSSMANN et al. (2012); LEVY et al. (2013).
Componentes por camadas	SALNIKOV et al. (2014); ANTUNES et al. (2016); LUPU; UNGUREANU; SIRITEANU (2013); KARPOV; RONZHIN (2014); MCCULLAGH; GALWAY; LIGHTBODY (2013); MULFARI et al. (2014); MULFARI; CELESTI; VILLARI (2015); TRIPATHY; RAHEJA (2014); SCARDOVELLI; FRÈRE (2015); GALANTE; MENEZES (2012).
Componentes por contêiner	CARUSO et al. (2013).

Fonte: Elaborado pelo Autor (2019).

Também é perceptível uma tendência pela adoção de soluções multimodais, já que 56% dos trabalhos analisados utilizam pelo menos dois tipos de entrada de dados nas aplicações. Neste sentido, existem soluções que possibilitam tanto que o usuário da solução escolha um dos tipos de entrada existentes, ou utilize concorrentemente mais de um tipo de entrada de dados. As principais motivações para isso são:

1. Aperfeiçoamento - o segundo *input* de dados é utilizado para refinar a entrada principal da aplicação, visando aprimorar a qualidade da interação; e
2. Fadiga - a entrada de dados alternativa é utilizada para diminuir o cansaço causado pelo excesso de comandos a partir de uma única forma de interação.

Um fator recorrente nas soluções multimodais é a utilização de comandos de voz como segunda forma de interação nas aplicações. Foram encontradas também soluções que não focam na interação propriamente dita, mas como facilitadores para o uso de alguma TA específica dos usuários (MULFARI; CELESTI; VILLARI, 2015; PAUL et al., 2013; MULFARI et al., 2014). A ideia destes projetos é fornecer uma infraestrutura que permita que uma TA específica seja utilizada por seus usuários fora de um ambiente previamente configurado. Estratégias como acesso às configurações para uma TA a partir da nuvem, por exemplo, são adotadas para desonerar o utilizador do processo de configuração da TA em um novo ambiente. Já quanto à saída de dados e ao *feedback* de funcionamento, praticamente todas as soluções provêm GUI para interação com os usuários. Apesar de existirem trabalhos que utilizam saída de dados alternativas, eles ainda se restringem a um universo pequeno (OSSMANN et al., 2014; VEIGL et al., 2013; CARDOSO et al., 2016).

Sobre as tecnologias empregadas, percebe-se que as escolhas são dependentes da expertise da equipe de trabalho, ou estão diretamente ligadas às características das aplicações pretendidas e dos dispositivos adaptados. Para diminuição de custos, a maior parte dos trabalhos prima por utilizar ferramentas livres ou adaptar equipamentos. No entanto, alguns trabalhos empregam soluções proprietárias, prática que, além de aumentar o custo final da aplicação, também limita as possibilidades de inovação, uma vez que não é possível efetuar muitas customizações ou extensões nestas ferramentas. Quanto ao uso de bibliotecas, algumas convergências podem ser notadas, como, por exemplo, a utilização do OpenCV para captação e processamento de imagens. Em termos de dispositivos de interação, o mesmo princípio se aplica ao EPOC para soluções que utilizam BCI.

Por fim, os trabalhos estudados não descrevem ou citam técnicas utilizadas ao longo do processo de desenvolvimento. O uso de métodos, semelhantes aos praticados na indústria, seria coerente uma vez que a concepção de aplicações de natureza assistiva compartilha características com a construção de software convencional. Com respeito a testes, grande parte dos estudos realiza avaliações funcionais sobre as soluções, utilizando técnicas que analisam os resultados de uso das mesmas para indicar possíveis caminhos para melhorias das propostas. Entretanto, alguns trabalhos não efetuam quaisquer tipos de testes sobre as soluções criadas, justificando isso pelas propostas ainda estarem em etapas embrionárias. Nenhum trabalho relata o uso de ferramentas voltadas a garantir qualidade de software, as quais poderiam ser relevantes especialmente aos trabalhos que ainda não estão finalizados. Finalmente, outro ponto importante é a dificuldade em testar as propostas junto ao público alvo final. Grande parte dos protótipos desenvolvidos é testada por usuários sem qualquer tipo de limitação.

3.5 Análise das Questões de Pesquisa - MSL

Por meio da execução do MSL foi possível abranger conjunto amplo e relevante de trabalhos que exploram o tema pesquisado. O resultado desta análise buscou responder três questões de pesquisa, apresentadas se Seção 3.1.2, sendo as respostas evidenciadas a seguir.

QP1: *Como as pesquisas mostram o processo de desenvolvimento e avaliação de interfaces e software para TA que ajudem as pessoas com deficiência física no uso do computador?*

A partir dos estudos sobre os trabalhos recuperados, não foi possível perceber padronização no processo para desenvolvimento de aplicações ou dispositivos de TA. Existem, no entanto, soluções que apresentam indícios de adoção de métodos de desenvolvimento, como, por exemplo, em OSSMANN et al. (2014). Os trabalhos analisados, no entanto, não explicitam o uso de técnicas de desenvolvimento da indústria na produção das soluções propostas. Especificamente tratando de avaliações, os trabalhos usualmente realizam testes funcionais sobre as soluções desenvolvidas. Não é descrito o uso de testes estruturais, ou seja, técnicas que garantam a qualidade ou cobertura de testes sobre os módulos, classes ou mesmo partes do software desenvolvido. Ainda existem trabalhos que não realizaram qualquer tipo de avaliação, justificando a isso por se tratar de modelos conceituais ou por estarem em estágio inicial de desenvolvimento.

QP2: *Existem frameworks ou plataformas de desenvolvimento de software, voltados a TA? Quais são eles?*

Conforme apresentado na Tabela 4, foram encontradas três soluções que podem ser caracterizadas como *frameworks* ou plataformas de software para TA (OSSMANN et al., 2014; VEIGL et al., 2013; MIESENBERGER et al., 2014; ACED LOPEZ; CORNO; DE RUSSIS, 2015). Dentre estes, o projeto AsTeRICS apresenta visões tanto sobre o contexto de uso da solução proposta quanto a respeito do modelo arquitetural de componentes de alto nível em camadas. Este framework possui características de ferramenta para edição visual, ou seja, ele permite gerar aplicações a partir da composição de elementos por meio de ferramenta gráfica, possibilitando que pessoas com pouco conhecimento técnico criem soluções assistivas. Os outros trabalhos caracterizados como *frameworks* são restritos a um nicho específico de aplicação (jogos em ambos os casos). Os trabalhos que geraram protótipos focam no desenvolvimento de uma solução para atender determinada finalidade, ou auxiliar a execução de alguma atividade de um público alvo específico. Não são caracterizados como frameworks de software pois não são projetados para serem reutilizáveis.

QP3: *O que caracteriza uma plataforma/framework como sendo voltado a TA?*

Não foram encontrados indícios, com relação à aplicação de técnicas de desenvolvimento ou a adoção de padrões arquiteturais, permitindo afirmar que um dispositivo ou aplicação é assistivo. O que as propostas têm em comum é o fato de usarem ou adaptarem tecnologias considerando um grupo que se deseja abranger. Além disso, a busca pelo provimento de modos de interação alternativos para acesso a recursos computacionais é o que mais se aproximaria de uma resposta para esta questão. Apesar de existir a necessidade de dispositivos específicos para TA, a aplicação de princípios de *Design Universal* na construção de produtos *mainstream* pode vir a eliminar a necessidade de alguns deles. No entanto, para alcançar este patamar é preciso considerar tanto características semelhantes quanto diferentes em termos de *design* e desenvolvimento dos produtos voltados a qualquer pessoa, independente de possuir ou não qualquer limitação física.

3.6 Estudo de Soluções Comerciais

A execução do MSL possibilitou uma visão macro do estado da arte da área de pesquisa. Sua aplicação foi relevante para contextualizar a investigação em curso dentro do campo de estudo abordado, permitindo responder as perguntas definidas no protocolo, além de delinear uma classificação dos tipos de projetos desenvolvidos no domínio de pesquisa. Embora amplo e relevante, o resultado alcançado com a execução do MSL deixou algumas lacunas a serem preenchidas. Isso em parte pode ser explicado pela *string* de busca utilizada, já que para satisfazer os três eixos de pesquisas abordados e respeitar o número máximo de caracteres aceitos pelos engines de busca, algumas palavras-chave importantes, sob o ponto de vista deste trabalho acabaram sendo preteridas. Assim, estudos potencialmente relevantes podem não ter sido recuperados pela busca efetuada.

Este resultado, contudo, suscitou a necessidade de ampliação dos estudos de forma que o mesmo abrangesse também soluções já consolidadas no mercado. Isso foi necessário para preencher as lacunas que não foram respondidas pelo MSL. Para complementar os resultados do MSL, foi conduzido um estudo avaliativo sobre os dispositivos de interação que possuem algum tipo de arcabouço disponível para desenvolvimento de aplicações (FERREIRA et al., 2017). Os aparatos escolhidos para esta análise são formas de NUI que possuem características que os habilitam para serem adaptados às necessidades de aplicações de TA. As NUI podem ser vistas como um paradigma no qual os dados de entrada são gerados a partir de ações naturais das pessoas (LIDDY et al., 2000).

Estas interfaces buscam aproveitar as habilidades dos usuários, como movimentos e comunicação verbal, para interagir com dispositivos computacionais. As NUI mais intuitivas se baseiam em movimentos, captando e rastreando atividades das mãos,

cabeça ou olhos, por exemplo (KOO; KIM; CHO, 2014; CANTONI et al., 2016). Originalmente, grande parte dessas soluções que propõe o uso de NUI está ligado ao entretenimento digital, focando especialmente no suporte para jogos eletrônicos. Contudo, outra área na qual as NUI's mostram grande potencial de aplicabilidade é a TA, já que muitos dos trabalhos encontrados ao longo do MSL utilizam estes dispositivos de forma adaptada.

Assim, foi conduzida uma análise qualitativa, sendo examinadas prioritariamente as características gerais dos arcaouços expostos pelos dispositivos, e não dados estatísticos e quantitativos. Os dispositivos analisados em algum momento apareceram nos trabalhos recuperados no MSL, possuindo características intrínsecas que permitem sua adaptação. Foram observados os seguintes dispositivos:

- *Emotiv EPOC*¹⁷ - Dispositivo que utiliza EEG para captação de sinais cerebrais;
- *Microsoft Kinect*¹⁸ - Dispositivo equipado com câmera e sensores infravermelhos que captam os movimentos do corpo, além de microfones para interação por voz (aplicações TTS);
- *Leap Motion*¹⁹ - Equipamento que realiza o rastreamento das mãos do usuário utilizando câmera e sensores infravermelhos. Focado em ambientes computacionais, principalmente em realidade virtual;
- *MYO*²⁰ - Pulseira que permite controlar equipamentos via gestos, os quais são captados por sensores de movimentos e também por eletromiografia (EMG); e
- *TOBII*²¹ - Dispositivo voltado ao rastreamento ocular por meio de eletro-oculografia (EOG).

A partir deste levantamento, foram enumeradas as características básicas destes dispositivos, as quais são resumidas na Tabela 8.

Ao analisar as formas de entrada de dados com suporte em cada dispositivo, foi constatado que *Leap Motion* e o *MYO* possuem características semelhantes, já que ambos funcionam por meio de gestos de mãos e/ou braços. Entretanto, eles adotam estratégias de captação distintas, o primeiro por meio de câmeras e infravermelhos, e o segundo por meio de EMG. O *MS Kinect*, por sua vez, mapeia os movimentos de todo o corpo, além de proporcionar comandos de voz, o que multiplica suas possibilidades de uso. Já o *TOBII* utiliza a técnica de *eye tracking* como abordagem de

¹⁷<http://www.emotiv.com/epoc/>

¹⁸<http://developer.microsoft.com/pt-br/windows/kinect/hardware>

¹⁹<http://www.leapmotion.com>

²⁰<http://www.myo.com>

²¹<http://www.tobii.com>

Tabela 8 – Características básicas dos dispositivos estudados.

Dispositivo	Input	Tecnologias
EPOC	EEG	C#
MS Kinect	Gestos, Voz	Java, Visual Studio
Leap Motion	Gestos Mão	JS, Java, Unity, Unreal Engine, C, C++, C#, Python
MYO	Gestos, EMG	Java, C, C++, C#, Unity
TOBII	EOG	C, C++, C#, Unity

Fonte: Elaborado pelo Autor (2019).

rastreamento. O EPOC, por se tratar de uma BCI, adota como entrada sinais cerebrais captados por meio de EEG.

Em termos de tecnologias utilizadas para desenvolvimento, percebe-se que a maior parte dos dispositivos dá suporte a pelo menos duas linguagens de programação, sendo a única exceção o EPOC, o qual utiliza apenas C#. Considerando aspectos de configuração dos dispositivos, todos disponibilizam software específico próprios para realizar ajustes e personalizações de uso. Ademais, todos contam com alguma estrutura de software para auxiliar a criação de aplicações por terceiros, sejam bibliotecas, API ou mesmo SDK. As principais funcionalidades que aparecem nestas estruturas, voltados à simplificação do desenvolvimento de aplicações para estes dispositivos, são destacadas na Tabela 9.

Tabela 9 – Funcionalidades encontradas nos arcabouços dos dispositivos.

Funcionalidade	Leap Motion	Myo	MS Kinect	TOBII
OnConnect	X	X	X	-
OnDisconnect	X	X	X	-
OnFailure	X	X	X	-
OnPose/Gesture	X	X	X	X
Vector3	X	X	X	X
setPosition	X	X	X	X
getPosition	X	X	X	X
OnAccelerometerdata	X	-	-	-
OnGyroscopeData	X	-	-	-
RequestBatterylevel	X	-	-	-
Config	X	-	-	-

Fonte: Elaborado pelo Autor (2019).

Os métodos `onConnect` e `onDisconnect` são relacionados a questões de conectividade dos dispositivos, de forma que alguma resposta fosse disparada ao se conec-

tar ou desconectar com a aplicação. Ainda relativo ao tema de conexões, o método `onFailure` é utilizado para reagir aos eventuais erros de funcionamento do dispositivo. Por sua vez, o método `onPose/Gesture` dispara alguma ação em resposta à detecção de determinado gestos. Já `vector3` é relacionado ao posicionamento do dispositivo para utilização em ambientes tridimensionais.

Os métodos `setPosition` e `getPosition` são utilizados para recuperar ou configurar as posições do cursor. Já as funcionalidades `onAccelerometerdata` e `onGyroscopeData` trabalham com acesso direto aos dados dos respectivos sensores. O método `requestBatteryLevel` fornece dados sobre a utilização da bateria. Por fim, a função `config` está relacionada a questões de configurações.

3.7 Considerações Finais do Capítulo

Este capítulo apresentou uma visão do estado da arte da área de pesquisa abordada, destacando os resultados alcançados com a execução do MSL. De forma complementar, foi conduzido um estudo qualitativo sobre dispositivos de interação alternativa disponíveis no mercado.

A visão da literatura proporcionada pelo MSL, além de responder as questões de pesquisa destacadas na Seção 3.5, permitiu perceber uma predisposição de desenvolvimento focado na resolução de problemas individuais de pessoas (ou de um pequeno grupo de pessoas) acometidas por uma determinada deficiência. Isso é uma tendência natural, já que o grau de uma limitação de um indivíduo é diretamente relacionado tanto à sua genética, quanto as suas aptidões e habilidades particulares. Assim, é legítimo que os trabalhos acadêmicos sejam guiados por estudos de casos específicos, buscando potencializar as habilidades individuais dos usuários.

A cabeça é a última parte do corpo que perde sua mobilidade, ou seja, tem seus movimentos preservados (MULFARI et al., 2015). Assim, o desenvolvimento de soluções assistivas que utilizem os movimentos desse membro do corpo é bastante relevante. Foi constatado pelo MSL que os trabalhos que utilizam movimentação da cabeça, como forma primordial de interação, consistem em protótipos específicos para atingir finalidades particulares. Ou seja, não permitem reutilizar os experimentos desenvolvidos em outros cenários de uso, diferentemente das soluções categorizadas como arcabouços de software, destacadas na Subseção 3.3.1, e enumeradas pela Tabela 4.

Sob a perspectiva das soluções comerciais, exploradas na Seção 3.6, percebe-se uma busca por generalidade, isto é, que os dispositivos sejam desenvolvidos tentando abranger o maior número de usuários possível. Portanto, enquanto os produtos comerciais buscam fornecer soluções genéricas, cobrindo um número mais significativo de usuários, os projetos científicos preocupam-se com soluções que atendam às indi-

vidualidades dos usuários.

Foi observado que a existência de estruturas de software reutilizáveis dos dispositivos comerciais, motivam a sua adoção no desenvolvimento de soluções adaptadas para atender às necessidades específicas dos usuários, em soluções assistivas. Assim, embora exista a necessidade de dispositivos específicos de TA, a aplicação de princípios como o *Design Universal*, por exemplo, para criar produtos convencionais pode eliminar gradualmente a necessidade de alguns deles. No entanto, para atingir esse estágio de maturidade, é necessário considerar semelhanças e diferenças em termos de projeto e desenvolvimento de produtos, visando atingir um nível de acessibilidade que inclua todos os usuários.

Por sua vez, em termos do processo de desenvolvimento, a principal diferença verificada na criação de soluções assistiva é a necessidade de considerar os aspectos humanos envolvidos. Preferencialmente, este processo deve contar com o acompanhamento de profissionais de saúde e pelo próprio usuário final, como forma de garantir que o produto final seja desenvolvido adequadamente.

Combinando os resultados do MSL com a análise do levantamento sobre soluções de mercado, foi possível estabelecer uma base de conhecimento para a continuidade do projeto. Assim, a sequência do trabalho destaca o processo de modelagem do arcabouço de referência voltado ao IOM. O próximo capítulo descreve a metodologia planejada, destacando os ciclos de desenvolvimento práticos realizados.

4 PROTOTIPAÇÃO DE APLICAÇÕES UTILIZANDO O IOM

Com base na conjunção das percepções e tendências científicas obtidas a partir do MSL, e da análise sobre as principais funcionalidades encontradas em dispositivos de interação comerciais, foi delineado um conjunto de ações práticas para o projeto de um arcabouço voltado a pessoas com capacidade de movimentação da cabeça preservada.

Neste sentido, este capítulo evidencia as principais etapas previstas pela metodologia de desenvolvimento para a concepção deste arcabouço. Também é apresentado o dispositivo de interação IOM, utilizado como estudo de caso neste trabalho. São então detalhadas as aplicações desenvolvidas a partir do método proposto. Por fim, são destacadas as principais lições aprendidas ao longo da execução deste processo.

4.1 Metodologia de Desenvolvimento

Paralelamente às investigações e aos levantamentos teóricos realizados, foi planejada uma etapa mais prática de desenvolvimento do projeto. A ideia é definir um processo que permita levantar os principais requisitos e funcionalidades, para a proposição de um arcabouço de software voltado à criação de soluções assistivas para pessoas que sofram com limitações motoras.

Para tanto, foi planejada a concepção de uma série de protótipos, utilizando um dispositivo de interação alternativo, em diferentes cenários de utilização. O objetivo a ser alcançado, com o desenvolvimento de tais protótipos, é seguir os padrões estabelecidos pelas premissas para criação de estruturas de software reutilizáveis, destacadas na Seção 2.1.4. Segundo estas técnicas, a criação de um conjunto de aplicações, dentro de um domínio específico, possibilita identificar características e funcionalidades que devem compor esta solução reutilizável.

Para orientar o desenvolvimento destas soluções foi adotada, como norteadora, a taxonomia proposta nesta tese após a execução do MSL (Seção 3.3). A ideia era conceber aplicações, de forma que fossem abrangidos os escopos de uso destacados pela taxonomia. Assim, considerando a classificação por funcionalidades, o objetivo

era realizar experimentos que proporcionassem:

- Acesso ao computador;
- Jogos assistivos; e
- Controle de ambiente.

Complementarmente, a criação de aplicações dentro dessas categorias permitiria realizar, por meio do desenvolvimento de **protótipos**, a proposta de um **arcabouço**. Ou seja, os tipos delimitados pela classificação de tipo de solução da taxonomia (Subseção 3.3.2) também serão abordados. Para tal, foram planejados quatro ciclos de desenvolvimento de TA utilizando o dispositivo de interação alternativa em diferentes cenários de aplicação, processo destacado pela Figura 38.

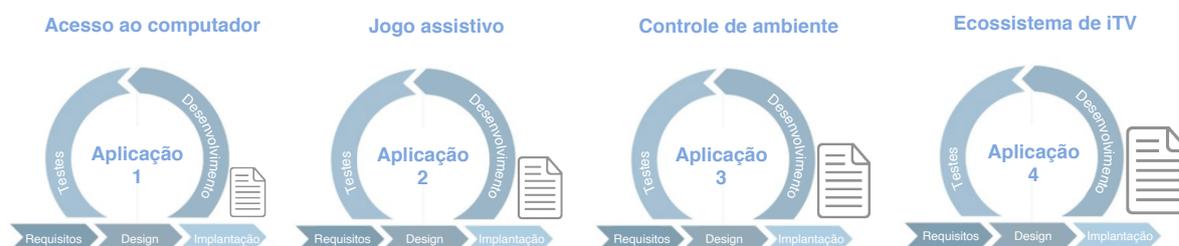


Figura 38 – Ciclos de desenvolvimento previstos.

Fonte: Elaborada pelo Autor (2019).

Além de criar aplicações segundo os três grupos principais expostos pela classificação por funcionalidades, um quarto ciclo foi adicionado prevendo a exploração do cenário de iTV. Como será destacado na Seção 4.3.4, o desenvolvimento deste ciclo foi motivado pela parceria estabelecida com o grupo de pesquisa DigiMedia, da UA, o qual possui um de seus focos voltado ao desenvolvimento de aplicações de iTV.

Assim, a intenção era aplicar, em cada um dos ciclos, técnicas de modelagem, desenvolvimento e avaliação de software, para criação deste conjunto de aplicações voltadas a pessoas com capacidade de movimentação da cabeça preservada. Este processo tem como objetivo final utilizar técnicas de desenvolvimento para criação de software reutilizável, destacadas no Capítulo 2. Assim, cada ciclo de desenvolvimento, em contextos diferentes, visa levantar e confirmar os requisitos e funcionalidades para proposição de um modelo de arcabouço para um domínio específico.

Conforme amplamente destacado, o cerne desta tese está relacionado a TA voltada a pessoas com movimentos de cabeça preservado, em especial, com limitações motoras permanentes ou transitórias. Esta perda de capacidade funcional, especificamente em termos de mobilidade, é um fator natural, já que ocorre a diminuição gradativa das habilidades e conseqüentemente das capacidades para desempenhar atividades cotidianas no decorrer da vida. Além desta motivação, a escolha pelo tema

foi instigada pois, oportunamente, encontra-se em desenvolvimento no IFSul, o IOM, um dispositivo de interação alternativa, inicialmente voltado para pessoas com limitações motoras nos membros superiores.

4.2 IOM (Interface Óculos Mouse)

O dispositivo IOM é o resultado de um projeto que visa o desenvolvimento de uma tecnologia para controle do computador que seja eficiente, de baixo custo e configurável. Seu objetivo é prover acessibilidade às pessoas com deficiência motora nos membros superiores, permitindo maior autonomia e independência na realização de tarefas por meio do computador. Ele consiste em uma armação de óculos adaptada, utilizando tecnologia embarcada. O IOM é dotado de sensores que implementam uma interface para simular as ações do *mouse* a partir da movimentação da cabeça. A Figura 39 destaca o IOM versão 1.0, e seus elementos essenciais.

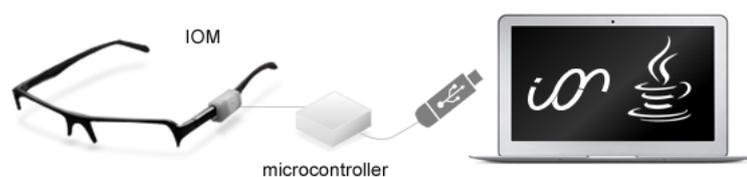


Figura 39 – Diagrama de contexto do IOM.
Fonte: CARDOSO et al. (2016).

Em sua versão atual o IOM utiliza uma armação de óculos com um acelerômetro embutido. Este sensor captura os movimentos de cabeça do usuário e transmite as coordenadas captadas por meio de uma conexão serial para o computador. Estas coordenadas são então convertidas em movimentos do cursor do *mouse*. Para habilitar os cliques, esta versão adota a técnica *Dwell Time*, estratégia pela qual eles são disparados quando o dispositivo permanece parado por um tempo pré-determinado.

4.2.1 Antecedentes

A primeira versão do protótipo, originalmente denominado "Óculos-mouse", surgiu em 2009 durante um projeto multidisciplinar desenvolvido por alunos e professores do curso técnico de mecatrônica no campus Charqueadas do IFSul, motivados pela ideia de fomentar a inclusão social por meio do provimento de acesso ao computador. Nesta etapa o projeto concentrava esforços no desenvolvimento de hardware, não prevendo o desenvolvimento de componentes de software para o dispositivo. Com relação ao processo de criação do hardware, nenhuma metodologia formal foi adotada, de forma que o dispositivo foi desenvolvido por meio de prototipações baseadas em observações e testes.

Apesar de sua evolução, que naquela época levou ao registro de patente do dispositivo (MACHADO, 2010), uma série de fatores culminou na estagnação do projeto.

Somente em 2015 a iniciativa foi retomada com o objetivo de aprimorar o protótipo inicial e torná-lo de fato uma inovação tecnológica. Rebatizado de IOM, o projeto teve continuidade no campus Pelotas do IFSul, tendo o suporte de uma equipe multidisciplinar composta por professores e alunos ligados as áreas de Engenharia, Computação, *Design* e Pedagogia. Nesta fase do projeto, foram adotados princípios de *Design Thinking* buscando aumentar as possibilidades de utilização do dispositivo (BROWN et al., 2010). A ampliação do projeto levou à adoção de técnicas para sua gerência e controle, fazendo com que ferramentas como o Canvas e técnicas do Scrum fossem incorporadas ao projeto. O foco do projeto foi aumentado, considerando a necessidade do desenvolvimento de componentes de software específicos para o IOM, além de aplicar melhorias de hardware ao projeto original do dispositivo.

4.2.2 Especificações

O dispositivo IOM é formado por dois componentes de software principais: um sistema embarcado, e uma aplicação de configuração. O sistema embarcado é formado pelo *firmware* específico do dispositivo, o qual é carregado pelo *bootloader* do microcontrolador. Este componente arquitetural executa duas funções primordiais:

1. Captar o movimento e inclinação da cabeça; e
2. Analisar estas ações enviando comandos de movimentação ao computador.

Para tanto, atualmente o IOM possui um acelerômetro que captura informações de posicionamento e inclinação provenientes da movimentação do dispositivo e os envia como sinais analógicos para o microcontrolador. A partir daí o sinal é convertido para o formato digital e então o *firmware* processa o sinal digitalizado, a fim de detectar um movimento do IOM. O *firmware* contém as instruções que deverão ser seguidas pelo dispositivo embarcado quando o mesmo estiver em execução. As rotinas executadas pelo software embarcado do IOM foram escritas por meio da linguagem *Wire C*. O fluxograma de funcionamento do software embarcado é mostrado na Figura 40.

Este sistema apresenta duas funcionalidades principais. A primeira é a função de *setup*, pela qual são realizadas as configurações iniciais da aplicação. A partir daí entra em execução o laço de instruções que fica executando continuamente. As instruções executadas no laço realizam uma série de validações, tomando decisões baseadas nos dados que recebem continuamente dos sensores de movimentação (acelerômetro) e tempo. O tempo inicialmente predefinido para que um clique seja acionado é de 800ms. No entanto, ajustes no tempo de clique podem ser feitos, de acordo com a necessidade do usuário.

O segundo componente do IOM é denominado Software de Configuração e Calibração do IOM (SCCIOM). Ele é o responsável por estabelecer uma interface com

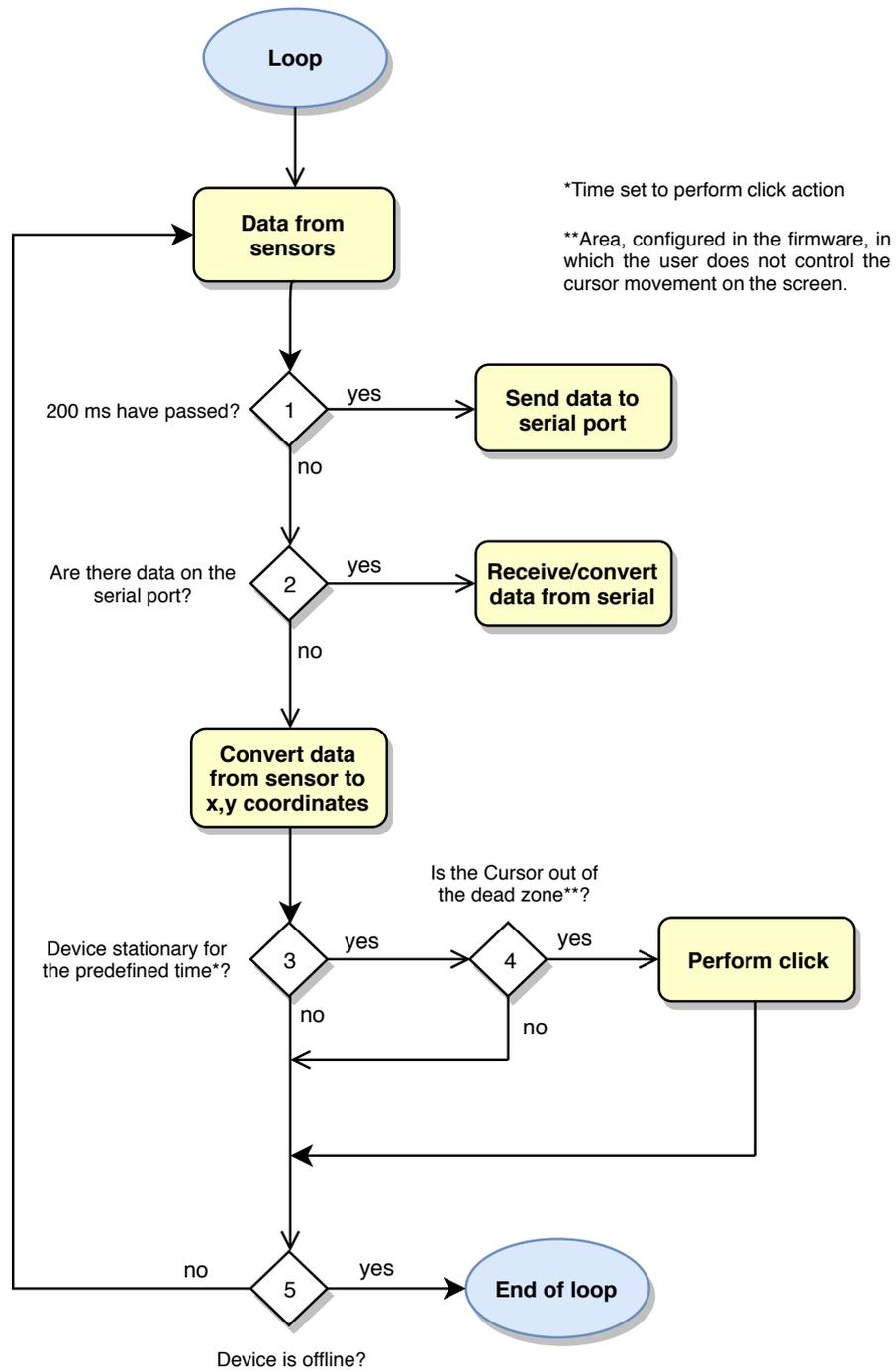


Figura 40 – Fluxograma de funcionamento do *firmware* do IOM.
 Fonte: Memorial descritivo do Projeto IOM (2017).

o usuário permitindo assim comunicação entre o dispositivo vestível e o computador. Além disso, provê orientação de uso, calibração do dispositivo e adaptações de aplicações computacionais de acordo com as necessidades do usuário. As Figuras 41 (a) e (b) apresentam duas das telas de interface providas pelo SCCIOM.

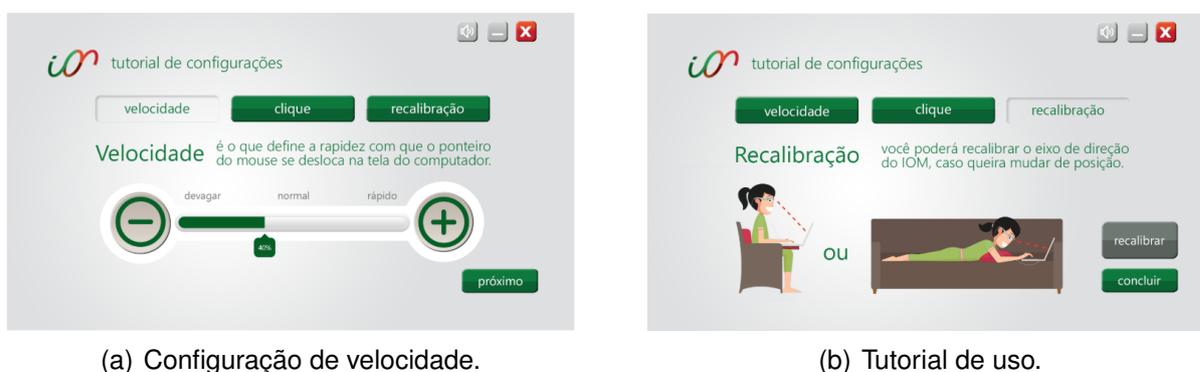


Figura 41 – GUI do SCCIOM.
Fonte: Memorial descritivo do Projeto IOM (2017).

A troca de dados entre o dispositivo IOM e o sistema de usuário é feita por meio do estabelecimento de uma comunicação serial de dados usando uma porta USB. A biblioteca utilizada para estabelecer a comunicação e efetuar a troca de dados é a jSSC¹ (*Java Simple Serial Connector*). Existe ainda um protocolo interno de comunicação, de forma que os dados trocados entre o dispositivo IOM e a Interface de usuário sejam compreendidos adequadamente em ambos os lados desta comunicação.

4.3 Aplicações Desenvolvidas

Esta seção destaca as soluções de TA previstas no processo prático deste trabalho. O dispositivo IOM é utilizado dentro dos cenários vislumbrados pela taxonomia proposta nesta tese. Para tanto, cada uma das aplicações é delineada considerando:

- A descrição da solução desenvolvida;
- As técnicas de ES utilizadas;
- A arquitetura proposta dentro do cenário de uso;
- A avaliação da aplicação; e
- Os principais requisitos elicitados.

A apresentação de cada uma das aplicações segue a ordem cronológica na qual seus processos de modelagem e desenvolvimento foram iniciados.

¹Disponível em <https://github.com/scream3r/java-simple-serial-connector>.

4.3.1 SCCIOM

O primeiro software desenvolvido teve como objetivo estabelecer uma interface focada nos aspectos de uso e configuração do dispositivo IOM. Conforme destacado na Seção 4.2.2, o SCCIOM permite configurar o dispositivo para atender às necessidades do usuário, além de fornecer diretrizes básicas de uso.

Para o SCCIOM, especificamente, foi realizado um processo de Engenharia Reversa (ER) sobre a aplicação desenvolvida, buscando compreender como a mesma funcionava por meio da análise de sua estrutura e de seu comportamento. Por meio do emprego desta técnica é possível analisar e extrair conceitos utilizados em sua construção, os quais podem ser padrões arquiteturais, diagramas de classes, arquitetura do sistema, ou qualquer informação que contribua para a compreensão da solução. Neste ciclo o objetivo era aplicar a técnica de ER, buscando se familiarizar ao dispositivo, analisando as principais funcionalidades e propriedades disponíveis no mesmo. Além disso, durante esta fase foi possível efetuar refatorações pontuais no código fonte da aplicação.

4.3.1.1 Projeto e Arquitetura

Como resultado da aplicação da ER sobre o SCCIOM, foi possível derivar uma arquitetura de contexto da solução, a qual é destacada pela Figura 42.

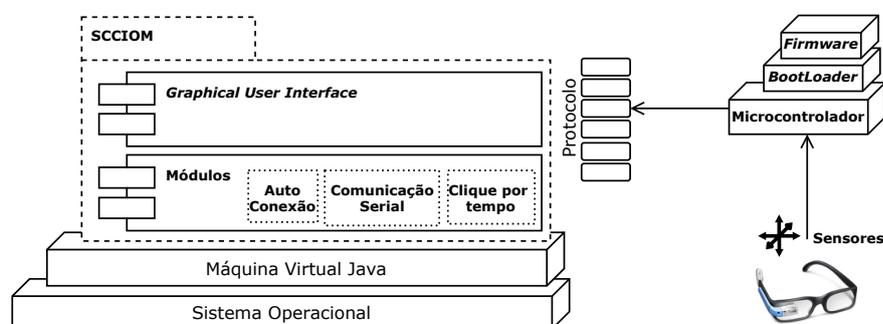


Figura 42 – Arquitetura do SCCIOM.

Fonte: MACHADO et al. (2019).

Por meio da elaboração deste diagrama foi possível perceber os principais elementos envolvidos na utilização do IOM. Além do SCCIOM, o protocolo de comunicação utilizado, e sistema embarcado destacados na Seção 4.2.2 estão presentes. Especificamente sobre o SCCIOM, ele está organizado em uma arquitetura com duas camadas. A primeira funciona como uma espécie de camada lógica, a qual executa diversos processos transparentes aos usuários finais. Ela é composta por três módulos:

1. Autoconexão - encarregado de identificar e estabelecer a conexão com o dispositivo;

2. Comunicação Serial - encarregado de se comunicar com o IOM, utilizando um protocolo interno próprio, o qual determina os tipos, intervalos e ordem dos dados trocados entre a aplicação e o dispositivo; e
3. Cliques - responsável por configurar o tempo para disparar os cliques, o tipo de clique e a velocidade do ponteiro na tela. Implementa a funcionalidade que verifica se o cursor está em movimento, permitindo a aplicação do *Dwell time*.

A segunda camada é responsável por gerenciar as GUI da aplicação. Suas telas se comunicam com os módulos da camada lógica, permitindo realizar configurações no dispositivo. A Figura 43 apresenta uma das telas do SCCIOM.

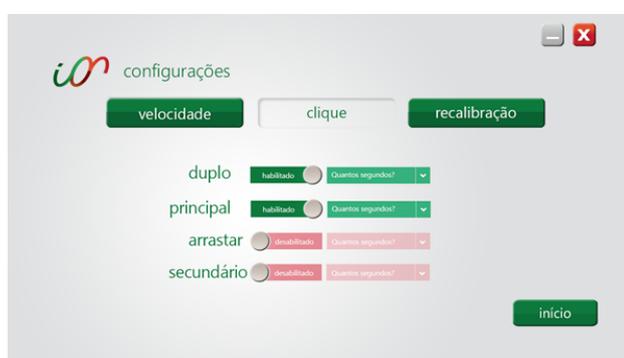


Figura 43 – Interface de configuração do IOM.
Fonte: Memorial descritivo do Projeto IOM (2017).

Esta interface, particularmente, permite a realização de configurações básicas sobre o dispositivo, tais como, a velocidade do cursor na tela do computador, o tipo de clique e a calibração inercial do dispositivo. Por fim, vale ressaltar que o SCCIOM foi desenvolvido em Java, possibilitando a sua execução a partir de mais de uma plataforma computacional.

4.3.1.2 Avaliação

O SCCIOM, especificamente, não foi avaliado segundo métodos ou instrumentos particulares. A aplicação foi testada indiretamente, em conjunto com o dispositivo IOM, por um usuário que possui deficiência física motora nos membros superiores. A sua utilização foi realizada durante sessões de avaliação mediadas pela equipe multidisciplinar do projeto de pesquisa. Durante essas sessões, tanto profissionais de saúde quanto especialistas em avaliação de IHC fizeram observações sobre os aspectos técnicos e ergonômicos do dispositivo IOM, e conseqüentemente do SCCIOM.

Detalhes sobre o processo de concepção, utilização e avaliação do SCCIOM, são destacados nos trabalhos de XAVIER et al. (2016), RODRIGUES et al. (2016) e MACHADO et al. (2019).

4.3.1.3 Requisitos Elencados

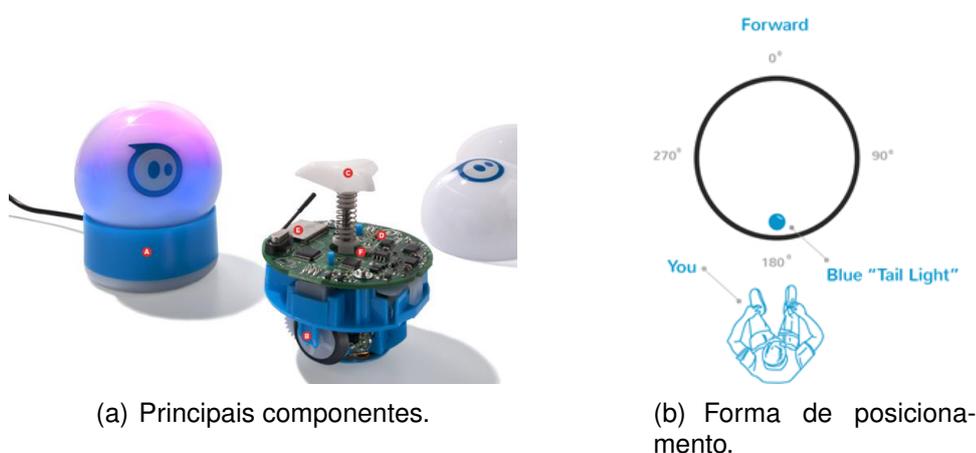
O estudo e refatoração sobre o SCCIOM foi essencial tanto para familiarização inicial com o dispositivo, quanto verificar os aspectos de interação com o dispositivo:

- Identificação e conexão com o dispositivo;
- Comunicação com o dispositivo, utilização de um protocolo interno;
- Configuração de tipo de clique;
- Configuração de velocidade de movimentação do cursor; e
- Configurações do *Dwell Time*.

Estas funcionalidades, consideradas essenciais, devem estar disponíveis em todos os outros cenários de uso.

4.3.2 Doce Labirinto

A segunda aplicação apresenta a concepção de um jogo lúdico e interativo, envolvendo dois dispositivos não convencionais que fazem uso de interação natural e recursos tangíveis: o IOM e o *Sphero* (CARDOSO et al., 2016). O *Sphero* é um robô, em formato de esfera, que pode ser utilizado para diversas finalidades, em especial para entretenimento. Ele foi concebido para ser utilizado, sobretudo, a partir de aplicativos para as plataformas móveis Android, IOS e Windows Phone. O *Sphero* é controlado via *Bluetooth*, e é dotado de três sensores (acelerômetro, magnetômetro e giroscópio), os quais lhe permitem realizar tarefas de posicionamento. Possui motores que possibilitam que ele role, além de um sistema de iluminação próprio. A Figura 44 (a) apresenta o *Sphero* 1.0 utilizado neste trabalho.



(a) Principais componentes.

(b) Forma de posicionamento.

Figura 44 – Dispositivo tangível *Sphero*, versão 1.0.
Fonte: CARDOSO et al. (2016).

Uma das características deste dispositivo é o fato do seu posicionamento ser relativo ao usuário que está no seu comando. Para tanto, o *Sphero* é equipado de uma pequena luz azul, denominada *Blue Tail Light*, a qual deve ser posicionada de forma a apontar para o usuário no comando, conforme ilustrado pela Figura 44 (b).

O *Doce Labirinto*, propriamente dito, consiste em um percurso físico que desafia o jogador, por meio de movimentos de sua cabeça captados pelo IOM, a levar o *Sphero* até a formiga (mascote da Fenadoce²) no final do labirinto. A Figura 45 (a) apresenta o cenário de uso do jogo. Já a Figura 45 (b) destaca o *Story Board* Doce Labirinto.

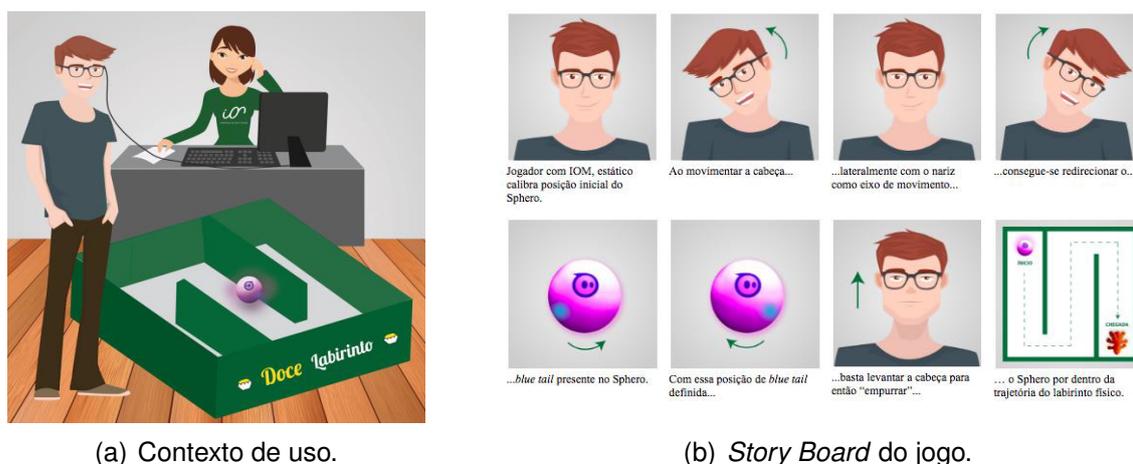


Figura 45 – Doce Labirinto.
Fonte: CARDOSO et al. (2016).

A ideia é que o jogador controle o *Sphero*, visualizando o interior do labirinto e gerando comandos a partir de movimentos do IOM. Para tanto, foi delimitado um conjunto de movimentos reconhecidos pelo jogo. Essencialmente, nesta aplicação, três movimentos podem ser efetuados. Para posicionar o *Blue Tail*, quando o *Sphero* está parado, movimentos para esquerda ou direita permitem que o jogador gire a esfera robótica até posicioná-la no ângulo desejado. O terceiro movimento, aceito no jogo, faz com que o *Sphero* role na direção em que está apontando. Este comando é disparado quando o usuário movimenta a cabeça para cima.

4.3.2.1 Projeto e Arquitetura

Para permitir o funcionamento do Doce Labirinto, foi desenvolvida uma aplicação, denominada de Painel de Controle, para conectar e configurar os dois dispositivos envolvidos na interação. Uma vez estabelecida esta comunicação, o IOM passa a enviar os sinais de movimentos para a aplicação de controle, a qual traduz e encaminha estes comandos, via *Bluetooth*, para o *Sphero*. A arquitetura do jogo é apresentada na Figura 46.

²Conforme destacado na Subseção 3.2.25, este jogo foi desenvolvido com foco nesta feira.

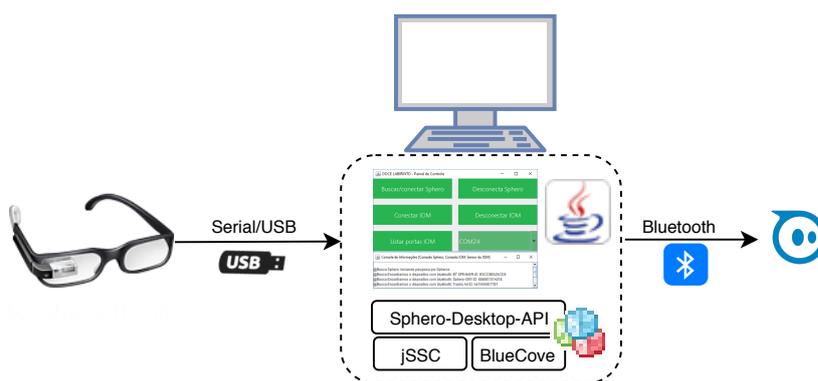


Figura 46 – Arquitetura do Doce Labirinto.
Fonte: CARDOSO et al. (2016).

A arquitetura apresenta como item central um computador, responsável por executar a aplicação de controle. Este computador deve estar apto a estabelecer tanto conexão serial, via USB, quanto por *Bluetooth*. Isso é necessário para que seja possível vinculá-lo ao IOM e ao Sphero, respectivamente. Além disso, este computador deve reconhecer Java, já que a solução foi desenvolvida nesta linguagem de programação. Entre as funcionalidades básicas a serem efetuadas pela aplicação de controle do Doce Labirinto, foram definidas as seguintes operações:

- Buscar dispositivos disponíveis para conexão Bluetooth;
- Conectar/desconectar com o Sphero via Bluetooth;
- Enviar comandos de ações ao Sphero;
- Listar portas seriais disponíveis; e
- Conectar/desconectar com o IOM via serial.

Especificamente tratando do *Sphero*, existem SDK que expõem métodos para sua manipulação específicos de cada plataforma. Além das API oficiais, existem diversas outras, não oficiais, as quais estendem alguma funcionalidade das SDK originais. Neste trabalho foi utilizada a Sphero-Desktop-API³, uma adaptação para funcionar em um ambiente de desenvolvimento *desktop*. Ela permite que uma aplicação se conecte a múltiplos dispositivos *Sphero*, possibilitando o envio de comandos individuais para cada um, por meio de identificadores únicos associados aos robôs.

Para estabelecer conexões *Bluetooth*, a Sphero-Desktop-API utiliza de forma integrada, o *BlueCove*⁴, uma biblioteca que possibilita descobrir e conectar aplicações e dispositivos por meio deste protocolo de comunicação. Assim, para este protótipo,

³Disponível em: <https://github.com/nicklasgav/Sphero-Desktop-API>.

⁴Disponível em <http://www.bluecove.org/>.

foram criadas funcionalidades que ao serem chamadas, executam os métodos responsáveis por estas tarefas. A Figura 47 destaca estas operações dentro do Painel de Controle do Doce Labirinto.



Figura 47 – Painel de gerenciamento do Doce Labirinto.
Fonte: CARDOSO et al. (2016).

Para transmitir os dados gerados pelos movimentos da cabeça do usuário via IOM por meio de comunicação serial foi utilizada a biblioteca jSSC. A partir do momento que o IOM se conecta ao sistema, ele fica apto a transmitir dados. Os dados enviados pelo IOM ao painel de controle consistem em conjunto de identificadores de movimentos, na forma de *strings*. Nesta aplicação o IOM envia quatro diferentes códigos que indicam que o usuário movimentou a cabeça para esquerda, direita, cima ou baixo. No entanto, apenas os três primeiros movimentos disparam comandos.

Ao receber estes dados, o painel de controle os traduz e realiza chamadas de métodos, via *Bluetooth*, para o robô esférico. Com relação ao *Sphero*, foram analisados os comandos que este dispositivo é capaz de realizar. Entre suas principais funcionalidades estão: rolar, brilhar e girar. Assim, no Doce Labirinto, três comandos disparam ações no *Sphero*. Movimentos para esquerda e direita rotacionam o dispositivo para uma destas direções enquanto ele está parado, utilizando o *Blue Tail Light* como referencial. Por fim, o movimento para cima dispara a ação de rolar do *Sphero*.

4.3.2.2 Avaliação

Um dos desafios da avaliação de um jogo é qual a perspectiva que deve ser adotada. Produtos lúdicos promovem a curiosidade, a exploração e não necessariamente têm ênfase na ordenação de tarefas. Comumente, estratégias que consideram a usabilidade perfazem uma alternativa de avaliação. Neste contexto, o Doce Labirinto foi avaliado por meio do instrumento AttrakDiff (HASSENZAHN, 2008).

A avaliação do jogo por meio dessa ferramenta envolveu 20 voluntários, sendo nove participantes na faixa etária de 7-12 anos, seis entre 13-20 anos e cinco variando de 21 a 45 anos. Os voluntários jogaram o Doce Labirinto durante a 30ª Fenadoce,

um evento cultural da cidade de Pelotas-RS. Os voluntários foram recepcionados no estande e convidados a interagir com o jogo. Depois de receber as instruções de funcionamento do jogo, os usuários tinham tempo livre para jogar e, após, eram solicitados a responder o questionário da ferramenta AttrakDiff.

A Figura 48 (a) apresenta os resultados na forma de portfólio gerada pelo AttrakDiff. Nesta representação o eixo vertical representa a qualidade hedônica (HQ), enquanto o eixo horizontal se refere a qualidade pragmática (PQ) da solução avaliada. Observa-se que a relação entre as dimensões PQ e HQ ficou localizada no quadrante “*self-oriented*”, demonstrando que aplicação é auto-orientada às suas funcionalidades.

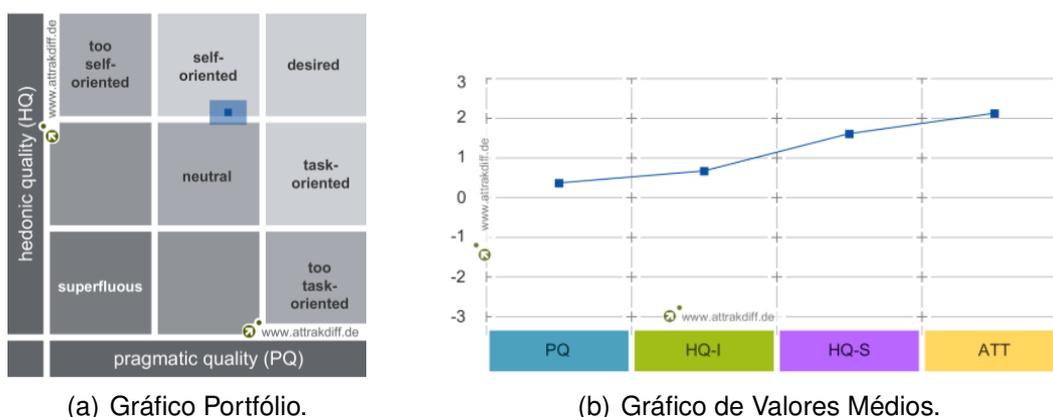


Figura 48 – Resultado da avaliação do Doce Labirinto.
Fonte: CARDOSO et al. (2016).

Já o gráfico apresentado na Figura 48 (b), mostra que o valor médio da PQ ficou próxima do eixo neutro (0), indicando que é possível evoluir as funcionalidades da aplicação. Por sua vez, as dimensões de HQ para identidade (HQ-I) e estímulo (HQ-S), atingiram pontuação de 0,69, 1,64, respectivamente. Esses valores denotam interesse no jogo por parte dos usuários, uma vez que mesmo apresentando dificuldades uso, os jogadores avaliaram como positiva e estimulante a experiência de uso do Doce Labirinto. Por fim a medida de atratividade (ATT) da solução ficou em 2,14, demonstrando que a avaliação geral do jogo foi positiva. Os resultados completos dos testes realizados sobre o Doce Labirinto, bem como os detalhes sobre o seu projeto e implementação são descritos detalhadamente em CARDOSO et al. (2016).

4.3.2.3 Requisitos Elencados

O Doce Labirinto permitiu corroborar algumas das funcionalidades básicas, já percebidas no processo de análise do SCCIOM, além de agregar outros aspectos ainda não explorados. Desta forma, as seguintes operações foram levantadas:

- Listar dispositivos disponíveis para conexão Bluetooth;
- Conectar/desconectar via *Bluetooth*;

- Enviar comandos via *Bluetooth*;
- Listar portas seriais disponíveis;
- Conectar/desconectar com o IOM via serial; e
- Utilização de comandos específicos, ao invés de movimentação contínua.

Apesar da versão do IOM utilizada não dar suporte à conexão *Bluetooth*, esta é uma funcionalidade prevista no projeto da próxima versão do dispositivo, sendo interessante já prever estas funcionalidades.

4.3.3 IOM4Home

A terceira aplicação proposta foi desenvolvida dentro do contexto de controle de ambiente. O *IOM4Home* é uma solução que tem como objetivo principal gerenciar um ambiente doméstico usando o IOM. A ideia desta aplicação era desenvolver uma solução assistiva, a qual utilizasse conceitos provenientes da IoT. O intuito era simplificar a vida de pessoas com deficiência físico motora, permitindo uma maior autonomia na execução de tarefas domésticas. A Figura 49 destaca o cenário de utilização da solução, no qual o usuário gerencia domóticos e sensores, disponíveis em um ambiente, por meio de uma interface projetada para o IOM.



Figura 49 – Contexto de uso do IOM4Home.
Fonte: PEROBA et al. (2017).

A solução organiza os domóticos disponíveis nos ambientes de uma residência, de forma que eles possam ser controlados pelo usuário em uma interface preparada para interação com o IOM (PEROBA et al., 2017; PEROBA; CARDOSO; COSTA, 2018).

4.3.3.1 Projeto e Arquitetura

O *IOM4Home* foi desenvolvido em duas etapas. A primeira é relacionada diretamente com a interface assistiva para controle dos domóticos. A segunda é relativa à infraestrutura necessária, envolvendo tecnologias relacionadas a IoT, com a utilização

de sensores e protocolos de comunicação específicos da área. O processo de modelagem e desenvolvimento do *IOM4Home* utilizou técnicas ágeis de desenvolvimento, em conjunto com o SCRUM. A Figura 50 destaca os *sprints* planejados, de acordo com esta metodologia de desenvolvimento.

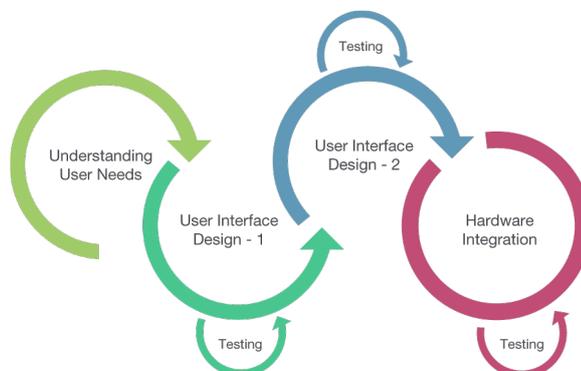


Figura 50 – Sprints de desenvolvimento do IOM4Home.
Fonte: PEROPA et al. (2017).

Foi previsto um ciclo inicial, para levantamento de requisitos que deveriam ser atendidos pela solução. Este *sprint* foi seguido por dois ciclos de desenvolvimento focados na criação da interface assistiva, e por um último voltado à integração com as tecnologias de IoT. Durante os *sprints* realizados no processo adotado foi concebida a arquitetura geral do IOM4Home, a qual é destacada na Figura 51.

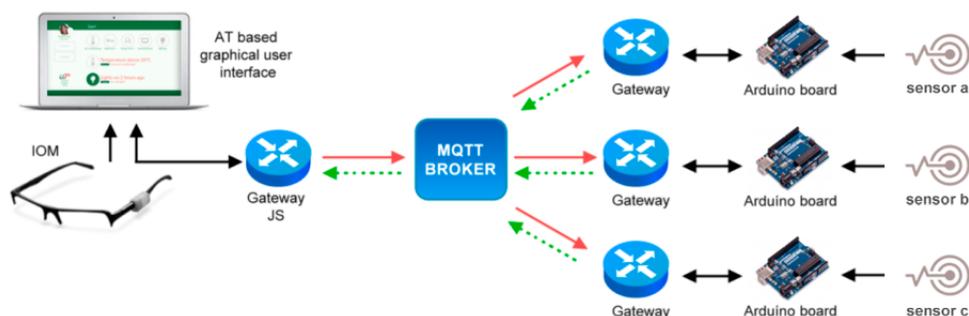


Figura 51 – Arquitetura do IOM4Home.
Fonte: PEROPA et al. (2017).

Como pode ser observado, do lado da infraestrutura existe o *Broker*, um componente central responsável por gerenciar a comunicação entre a interface e os dispositivos disponíveis no ambiente. Para comunicação entre o *Broker* e os demais agentes o *IOM4Home* foi utilizado o MQTT⁵, um protocolo IoT, baseado no paradigma de troca de mensagens *publisher/subscriber*, para a comunicação M2M (*Machine To Machine*). Para possibilitar a troca de mensagens com o MQTT *Broker*, foram desenvolvidos dois tipos de componentes de software:

⁵<http://mqtt.org/> disponível em março de 2018

- *Gateways Arduino* - capturam dados dos domóticos e enviam ao *broker*; e
- *Gateway JS (JavaScript)* - gerencia a troca de dados entre a interface e o *broker*.

Especificamente tratando da interface da aplicação, ela agrega uma série de informações e dados a respeito de diferentes tipos de aparelhos interconectados. Os dados captados pelos sensores e aparelhos domésticos monitorados são condensados em uma interface desenvolvida para o IOM. Tecnicamente, esta interface foi desenvolvida como uma aplicação responsiva Web, por meio do *framework Bootstrap*⁶. Esta forma de implementação permitiu incorporar às interfaces o *GateWay JS*, responsável por se comunicar com o *broker*. Essencialmente, esse componente atua como *listener*, o qual aguarda mensagens provenientes do *broker* a respeito dos dispositivos presentes no ambiente controlado.

4.3.3.2 Avaliação

Durante a construção do *IOM4Home*, vários protótipos foram criados usando uma abordagem iterativa de desenvolvimento e avaliação. Como pode ser visto na Figura 50, a aplicação foi desenvolvida em três *sprints*. Cada um deles possuía uma etapa de avaliação específica, visando melhorar o *IOM4Home* incrementalmente. A avaliação no primeiro *sprint* foi realizada por cinco usuários sem limitações físicas, em uma abordagem orientada a tarefas focada na usabilidade da interface da aplicação. Duas das interfaces avaliadas são apresentadas nas Figuras 52 (a) e (b).



Figura 52 – Interface do *IOM4Home*, primeira versão.
Fonte: PEROPA et al. (2017).

Neste *sprint*, foi utilizado o protocolo *Think Aloud*, técnica pela qual os usuários são orientados a verbalizar seus sentimentos ao realizar as tarefas propostas, enquanto a equipe de avaliação observa o processo, registrando os comentários (JÄÄSKELÄINEN, 2010). Os resultados destes testes levaram a alterações da aplicação, especialmente em aspectos de *design* e disposição de elementos na interface. O segundo *sprint* resultou na remodelação das interfaces, destacadas nas Figuras 53 (a) e (b).

⁶<http://getbootstrap.com/> April, 2018

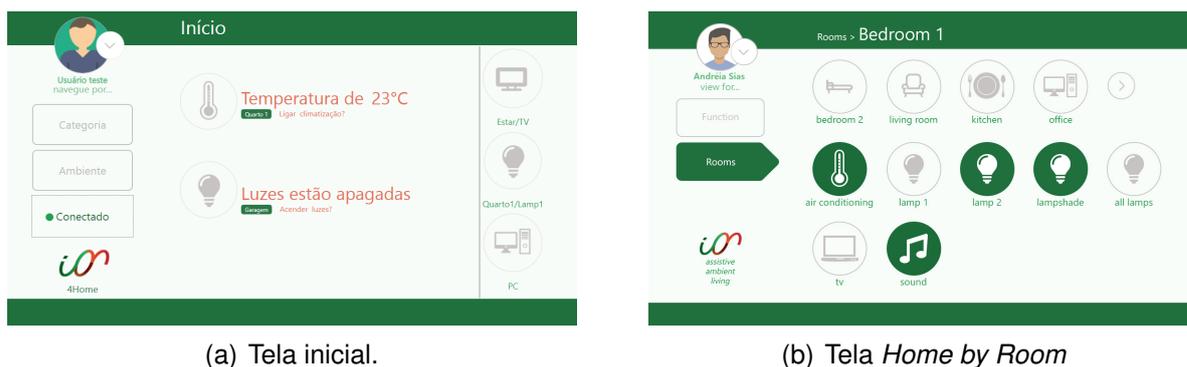


Figura 53 – Interface do *IOM4Home*, segunda versão.
Fonte: PEROPA et al. (2017).

Os testes do segundo *sprint* sobre o *IOM4Home* utilizou a ferramenta AttrakDiff. Esta rodada envolveu 20 estudantes, com idade entre 18 e 25 anos, sem nenhuma limitação. Assim como no primeiro *sprint*, cada um dos participantes executou uma lista de tarefas pré-determinadas. Estas atividades consistiam em realizar ações como acender lâmpadas ou desligar o ar condicionado, por exemplo. Após a realização dos testes, os usuários responderam ao questionário AttrakDiff.

Como pode ser percebido no gráfico portfólio, apresentado na Figura 54 (a), as dimensões PQ e HQ alcançaram um grau de confiabilidade de 0,33 e 0,17, respectivamente. Observando a relação entre estas dimensões, é perceptível que o *IOM4Home* proporcionou satisfação aos seus usuários, sendo considerado um produto com características desejáveis, não limitado por suas funcionalidades.

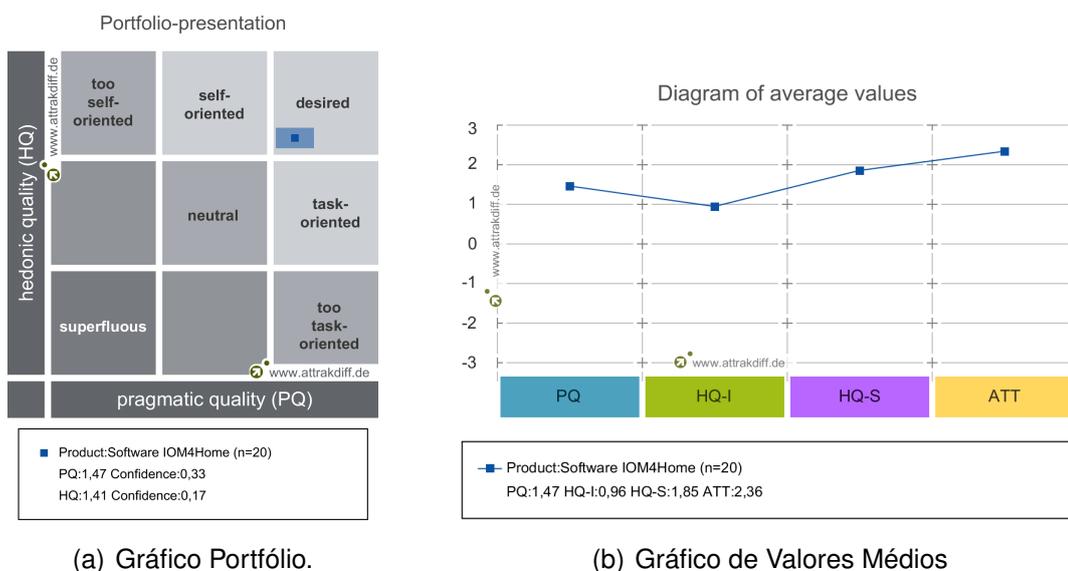


Figura 54 – Resultado da avaliação do *IOM4Home*.
Fonte: PEROPA et al. (2017).

Isso também é percebido pelo retângulo azul posicionado no quadrante "*desired*", mostrando uma avaliação positiva da solução. No entanto, sua localização denota que

há espaço para melhorias, tanto em aspectos atratividade (HQ) quanto de usabilidade (PQ). Os resultados do gráfico de valores médios, destacado pela Figura 54 (b), corroboram isso. Em especial, o HQ-I registrou o menor valor das dimensões avaliadas (0,96), sugerindo que melhorias podem ser feitas no produto, de forma que o seus usuários se identifiquem mais com o *IOM4Home*.

Por fim, o terceiro *sprint* focou na execução de testes funcionais sobre a solução completa. Para tanto, um ambiente controlado foi organizado para realizar a avaliação, contando com três sensores conectados à interface: Sensor de temperatura DHT11; Sensor para controle de trava, *Micro Slave 9g SG90*; e sensor de presença ambiente, HC-SR04. Cinco usuários sem limitações executaram uma série de tarefas usando o *IOM4Home* para acessar os dados dos sensores. Assim como no primeiro *sprint*, esta avaliação utilizou o protocolo *Think Aloud* na coleta dos resultados. Todos os usuários completaram satisfatoriamente as atividades, avaliando positivamente o uso da ferramenta durante esse teste funcional.

Mais detalhes tanto sobre o processo iterativo de desenvolvimento do *IOM4Home*, quanto dos resultados alcançados ao longo de suas etapas de avaliação são amplamente explorados em PEROBA et al. (2017) e PEROBA; CARDOSO; COSTA (2018).

4.3.3.3 *Requisitos Elencados*

Em relação aos requisitos para modelagem de estrutura reutilizável, os seguintes elementos foram percebidos no desenvolvimento do *IOM4Home*:

- Utilização de *feedback* visual com cores contrastantes;
- Adoção de botões com tamanho razoável, utilizando espaçamento adequado na interface, de modo a permitir facilitar o número de cliques corretos na GUI;
- Ainda em relação aos cliques, foi possível extrair outro requisito, a incorporação da ação *gravitacional* (uma espécie de seleção forçada), funcionalidade pela qual o cursor do *mouse* deveria ser identificado ao se aproximar de uma área clicável da GUI; e
- Finalmente, outra percepção foi a solicitação pela utilização de uma segunda entrada de dados, como reconhecimento de voz, por exemplo. De acordo com as percepções dos usuários esse recurso seria útil, especialmente, na simplificação do disparo de cliques.

Como pode ser percebido, o desenvolvimento do *IOM4Hme* levantou questões relacionadas à interação da GUI com o IOM, dada a natureza da solução construída. Desta forma, os *sprints* evidenciaram questões relacionadas a funcionalidades que devem ser tratadas em termos de interface da aplicação, quando se utilizando o modo de funcionamento padrão do IOM.

4.3.4 IOM4TV

A iTV é um conceito que pavimenta o caminho para o desenvolvimento de novos tipos de experiências que utilizam a televisão para diversos fins, além de sua aplicação original (SILVA et al., 2017). A ideia da iTV é constituída por dois conceitos fundamentais: a televisão propriamente dita e a interatividade que ela proporciona (SILVA, 2014).

Nesta perspectiva, a plataforma +TV4E, desenvolvida na Universidade de Aveiro, em Portugal, visa utilizar a TV como canal de difusão de informações sobre serviços públicos e sociais customizados para pessoas idosas, intercalando esses conteúdos com a programação normal de TV (SILVA, 2017). A plataforma foi concebida na perspectiva de tornar seu público-alvo em usuários, e não apenas espectadores passivos (SILVA et al., 2016). Unir a TV com algum tipo de interação permite que conteúdos sejam distribuídos de acordo com os hábitos e gostos específicos de cada usuário, proporcionando uma experiência de uso mais rica e adequada (CARAVAU; SILVA, 2016). A Figura 55 destaca os principais componentes do +TV4E.

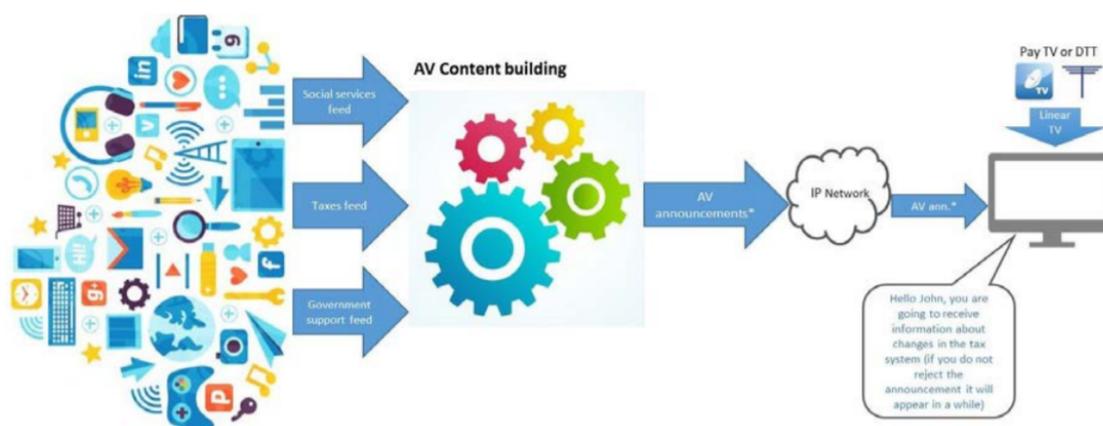


Figura 55 – Arquitetura da plataforma +TV4E.

Fonte: SILVA et al. (2016).

O foco da solução é apresentar conteúdos informativos que facilitem o acesso a informações relevantes no contexto dos usuários, enquanto estes assistem TV. Para que nenhum programa em andamento seja perdido, o +TV4E permite pausar o programa que está sendo transmitido, possibilitando que os usuários assistam um vídeo informativo e, em seguida, retornem a transmissão a partir do ponto em que ela foi parada. A plataforma possui uma biblioteca que agrega os vídeos enviados aos usuários, além de utilizar um sistema de recomendação próprio para selecionar conteúdo de acordo com perfis. O IOM4TV foi desenvolvido com o intuito de integrar os projetos +TV4E e o IOM. O objetivo é utilizar o IOM para interagir com o +TV4E, permitindo ao usuário realizar as principais funções disponibilizadas pela aplicação de iTV.

4.3.4.1 Projeto e Arquitetura da Aplicação

Para atingir seus objetivos e realizar o processo de integração destas tecnologias, a solução proposta utiliza os seguintes componentes:

- **TV** - Meio de interação principal, o qual exibe o conteúdo da plataforma +TV4E;
- **Set-Top Box (STB)** - Equipamento conectado à TV com uma fonte de sinal externa que transforma o conteúdo em um formato que pode ser exibido na TV;
- **IOM** - Dispositivo de interação baseado em movimentos da cabeça; e
- **Contêiner de Execução** - Computador que se conecta ao IOM e converte os movimentos da cabeça em movimentos de cursor na tela.

Tendo estes componentes em vista, foi elaborada uma arquitetura de alto nível que permite a integração e o funcionamento destes elementos. Esta arquitetura é destacada na Figura 56.

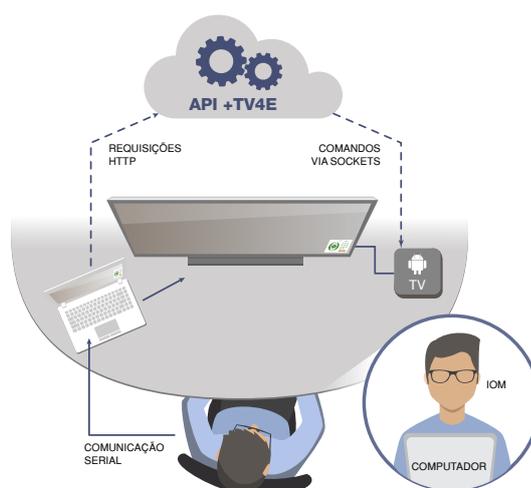


Figura 56 – Arquitetura do IOM4TV.
Fonte: CARDOSO et al. (2019).

Como pode ser observado, existem dois componentes de software que fazem a conexão de todos os elementos supracitados, permitindo sua comunicação: API +TV4E e a interface de controle virtual. O primeiro atua como uma fachada de software que recebe dados (via solicitações HTTP POST) da interface de controle virtual e os converte em comandos executados no STB.

Já o segundo componente, ou seja, a interface de controle adaptado para o IOM, teve duas versões desenvolvidas. Ambas realizam as tarefas básicas de utilização da plataforma +TV4E: troca de canais, acesso e navegação dos vídeos na biblioteca da plataforma. A primeira versão desenvolvida é baseada no comportamento padrão do

IOM, ou seja, confia nos movimentos contínuos do dispositivo para mover o cursor e dispara os cliques utilizando a abordagem *Dwell Time*. Por sua vez, a versão alternativa permite a navegação por eventos, ou seja, ela reage a movimentos pré-definidos para executar alguma ação. Assim, foram estabelecidos quatro movimentos básicos reconhecidos pela aplicação: para cima, baixo, esquerda e direita.

4.3.4.2 Avaliação da aplicação

Para avaliar o IOM4TV, em uma abordagem qualitativa, foram utilizados dois instrumentos: AttrakDiff e SAM (*Self-Assessment Manikin*). A ideia é perceber as emoções sentidas pelos usuários (via SAM) e avaliar a experiência do usuário (por meio do AttrakDiff), durante a interação com o protótipo. Foram realizadas duas rodadas de avaliação, envolvendo cinco participantes (com mais de 55 anos) sem limitações, sendo três mulheres e dois homens, em ambas as séries. As Figuras 57 (a) e (b) destacam a evolução entre as interfaces testadas nestes dois experimentos.

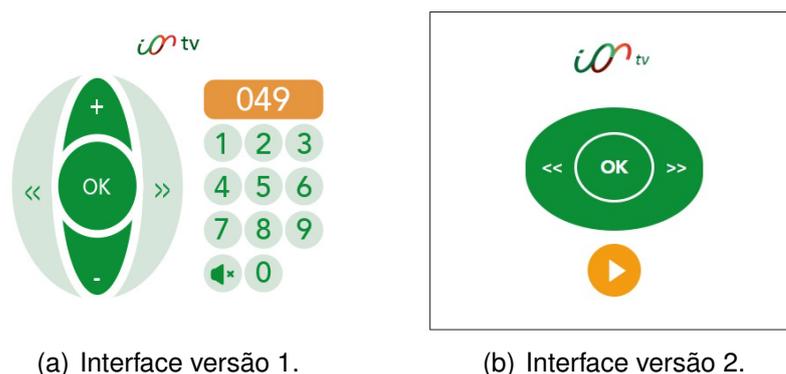


Figura 57 – Protótipos do IOM4TV.

Fonte: CARDOSO et al. (2018).

Cada usuário realizou um treinamento com o IOM, para se familiarizar com este dispositivo de interação antes de proceder com o teste do *IOM4TV*. A ferramenta *FittsTask*⁷ foi usada para gerar um conjunto de exercícios para este treinamento. A primeira rodada de testes ocorreu em um laboratório de informática. Ou seja, não havia preocupação em tornar o ambiente confortável. Por sua vez, a segunda sessão foi preparada com a orientação de um terapeuta ocupacional. Desta forma, um ambiente mais acolhedor foi montado, semelhante a sala de estar de uma casa.

Como o SAM é um instrumento de avaliação de aspectos emocionais, vários fatores podem influenciar os usuários ao preencher sua escala. Assim, é necessário que os especialistas analisem nuances que ocorrem ao longo de cada avaliação. Em uma observação inicial, percebe-se que os participantes consideraram a solução muito satisfatória, como pode ser destacado na Figura 58.

⁷<http://www.yorku.ca/mack/FittsLawSoftware/>



Figura 58 – Resultados do SAM. Em laranja, as respostas do primeiro experimento. Em verde, as respostas referentes ao segundo experimento.
Fonte: CARDOSO et al. (2018).

Em relação ao aspecto **Satisfação**, em ambas as rodadas de teste, todos os usuários selecionaram o nível máximo neste item de avaliação. A avaliação muito positiva nessa dimensão pode estar ligada ao senso de utilidade e inserção que os idosos sentiram ao participar dos testes. Por outro lado, o aspecto **Excitação** obteve resultados muito diferentes. Os dados indicam que os usuários estavam bastante ansiosos e agitados durante o primeiro experimento. Esse aspecto teve resultados opostos durante a segunda bateria de testes, demonstrando maior familiaridade, relaxamento e confiança ao utilizar a aplicação. Por fim, na dimensão **Controle** percebeu-se que os usuários não se sentiam no comando da aplicação na primeira rodada de testes. Já na segunda rodada, os resultados positivos demonstram que os participantes já estavam mais seguros ao usar o *IOM4TV*, refletindo a avaliação positiva.

Complementarmente a ferramenta AttrakDiff, por sua vez, resultou nos gráficos destacados pelas Figura 59 (a) e (b).

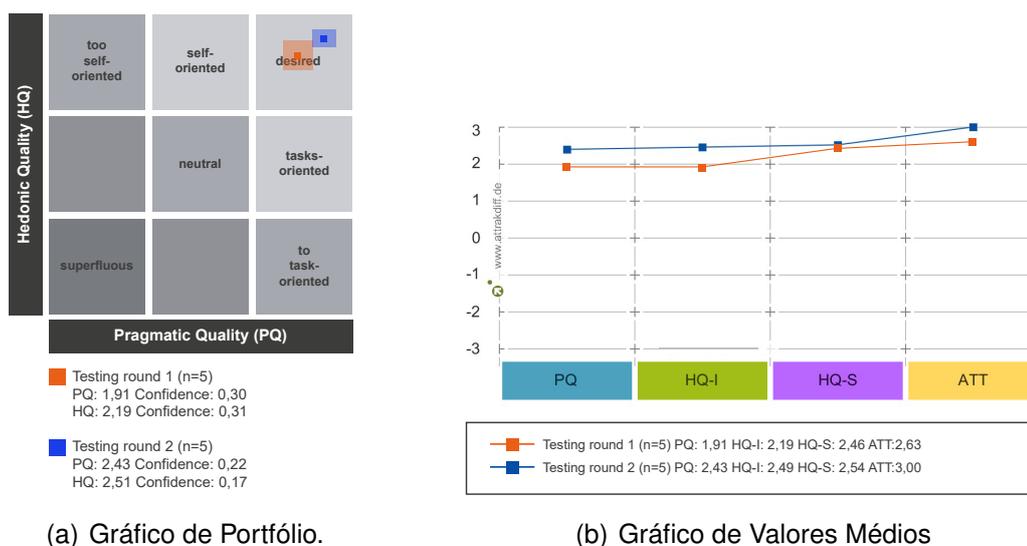


Figura 59 – Resultado das avaliações do IOM4TV.
Fonte: CARDOSO et al. (2018).

Os resultados da primeira rodada de testes nesta ferramenta resultou em uma boa

avaliação, tanto em aspectos pragmáticos (PQ=1,91) quanto hedônicos (HQ=2,19). A segunda avaliação obteve um resultado ainda melhor (PQ=2,43 e HQ=2,51). Com relação ao retângulo de confiança, ambos ficaram localizados no quadrante "*desired*", com um tamanho razoavelmente pequeno, atestando a confiabilidade dos resultados.

Já a Figura 59 (b) apresenta o diagrama de valores médios das avaliações. Os valores de PQ sugerem que houve uma ligeira melhora na avaliação da usabilidade, possivelmente relacionado ao fato dos usuários já terem contato com a forma de interação previamente. O valor HQ-I segue esta tendência, apresentando melhora nesta dimensão. Além da familiaridade com a aplicação, o ambiente acolhedor também pode impactar nesse resultado. O valor do HQ-S, por sua vez se manteve estável, indicando que o produto teve o mesmo nível de estímulo para o usuário. Por fim, a ATT ressalta que o produto gera uma alta atratividade, uma vez que atingiu níveis altos em ambas rodadas de avaliação.

As observações, realizadas durante estas avaliações, levaram à concepção da segunda versão do protótipo, a qual funciona com base na detecção de eventos específicos. O objetivo desta versão é diminuir a quantidade de movimentos que devem ser realizados para utilizar o IOM4TV. Esta interface, ainda em desenvolvimento, é apresentada na Figura 60.



Figura 60 – Interface do protótipo baseado em eventos
Fonte: CARDOSO et al. (2019).

Este protótipo funciona com um *listener*, o qual fica esperando envio de algum dos movimentos a partir do IOM, permitindo a navegação na interface por meio dos quatro movimentos básicos existentes. Esta aplicação, desenvolvida em Java, deve evoluir de forma que a interface da versão 2, seja incorporada a ela. Vale ressaltar que este protótipo foi testado apenas em ambiente de desenvolvimento.

Maiores detalhes sobre estes experimentos estão em CARDOSO et al. (2018) e CARDOSO et al. (2019). Além disso, encontra-se em avaliação uma ampliação da discussão sobre os resultados obtidos, para publicação em um periódico da área.

4.3.4.3 *Requisitos Elencados*

As observações durante as rodadas de avaliação dos protótipos do IOM4TV corroboram as mesmas tendências observadas no *IOM4Home*, em termos de interfaces com o usuário. Ou seja:

- *Feedback* visual contrastantes;
- Botões de tamanho razoável, e com espaçamento entre os mesmos;
- Função gravitacional - atrair o cursor ao se aproximar de uma área clicável; e
- Comandos a partir de uma segunda entrada de dados, como reconhecimento de voz ou acionamento mecânico alternativo, por exemplo.

Além disso, algumas constatações aferidas durante as avaliações, em especial em relação ao esforço e à fadiga dos usuários, levaram ao desenvolvimento da versão que responde a eventos específicos. Este funcionamento é similar ao implementado no Doce Labirinto, já que a interação com a aplicação se dá por meio da definição de um conjunto básico de movimentos (cima, baixo, direita, esquerda), os quais dispararam a execução das operações propriamente ditas. Assim, este modo de operação é um requisito relevante que deve ser considerado no projeto da estrutura de software reutilizável.

4.4 Lições Aprendidas

A criação das soluções descritas ao longo do capítulo se baseou em uma abordagem empírica, com o objetivo de utilizar o dispositivo IOM em diferentes cenários de uso. Ao longo destas etapas foi possível abordar e utilizar diversos conceitos relacionados a ES, como por exemplo, prototipação, engenharia reversa, desenvolvimento de software reutilizável para um domínio específico e técnicas de desenvolvimento ágil, entre outros. Também foi possível explorar diferentes métodos de avaliação nos quatro ciclos executados. Estes testes foram valiosos, tanto para o levantamento de requisitos, quanto para a percepção das principais dificuldades encontradas pelos usuários nos processos de interação com as soluções desenvolvidas.

Além disso, a investigação realizada durante o desenvolvimento dos protótipos confirmou que as funcionalidades e propriedades essenciais aparecem de forma recorrente, em diferentes nichos de uso. Parte significativa destas funcionalidades está relacionada com os métodos encontrados durante o levantamento apresentado na Seção 3.6. A Tabela 10 relaciona estas funções com os principais aspectos encontrados nas soluções desenvolvidas durante o desenvolvimento das aplicações.

Neste contexto, os métodos `onConnect` e `onDisconnect` estão relacionados com questões de conectividade dos dispositivos. Ou seja, tratam de ações referentes a

Tabela 10 – Comparação entre as funcionalidades das aplicações, com os métodos encontrados nas soluções comerciais.

Funcionalidades	SCCIOM	DL	IOM4Home	IOM4TV
onConnect	X	X	-	X
onDisconnect	X	X	-	-
onFailure	X	-	-	-
onPose/Gesture	X	X	X	X
vector3	-	-	-	-
setPosition	X	X	X	X
getPosition	X	X	X	X
onAccelerometerdata	X	X	-	X
onGyroscopeData	-	-	-	-
requestBatteryLevel	-	-	-	-
config	X	-	-	-

Fonte: Elaborado pelo Autor (2019).

conexão ou desconexão dos dispositivos com a aplicação que os utilizam. As quatro aplicações, destacadas neste capítulo, lidaram em algum ponto com esta questão de conexão. O *IOM4Home* realizou este processo via SCCIOM, assim não lidou diretamente com esta tarefa no seu desenvolvimento. Eventos de desconexão foram abordados especificamente pelo SCCIOM e Doce Labirinto. Ainda nesta questão de conexões, o método `onFailure` é utilizado nas soluções para reagir a erros de funcionamento do dispositivo que causam interrupção mais abruptas.

Por sua vez, o método `onPose/Gesture` trata de ações que respondem à execução de determinados gestos ou movimentos. Nas aplicações criadas, esta funcionalidade possui correspondência em todas elas, por meio de diferentes formas de utilização. O Doce Labirinto e o IOM4TV, por exemplo, puderam ser explorados por meio da captação de movimentos por eventos. Os métodos `setPosition` e `getPosition`, também relacionados às questões de movimentações, aparecem de alguma forma em todas as aplicações, seja recuperar a posição atual, ou configurar a posição do cursor, quando utilizando o modo de operação padrão do IOM. Nas aplicações por eventos, no entanto, acabam não sendo necessárias, uma vez que o cursor acaba não sendo utilizado na interação.

A função `vector3` trata da posição dos dispositivos de interação para utilização em ambientes tridimensionais. Esta característica, entretanto, não foi explorada em nenhum dos desenvolvimentos realizados, uma vez que o IOM ainda não oferece suporte a este tipo de funcionalidade. Por sua vez os métodos `onAccelerometerdata` e `onGyroscopeData` trabalham com acesso direto aos dados destes sensores, Nas aplicações desenvolvidas, este aspecto não foi diretamente explorado. No entanto, esta

questão pode ser abordada futuramente, visando utilizar melhor os dados proveniente do acelerômetro. Outra funcionalidade não utilizada é tratada por métodos como `requestBatteryLevel`. Dado que o IOM não possui bateria acoplada em sua versão corrente, esta função não é abordada.

A funcionalidade `config` está relacionada com as funções de configurações do dispositivo. No caso do IOM seriam os métodos relacionados a tarefas de configuração e calibração são usados nas aplicações desenvolvidas. Além disso, outras percepções estão relacionadas ao próprio dispositivo, como o funcionamento alternativo por eventos, ou a funcionalidade de inferir objetos selecionáveis (ação gravitacional), para simplificar o foco em objetos passíveis de clique. Por fim, o tópico relacionado à utilização de interação multimodal foi bastante mencionado durante as avaliações desenvolvidas.

Diante do exposto, o Capítulo 5 trata da proposta de um modelo de arcabouço aplicável para o domínio específico de atuação do IOM. São apresentados e discutidos os aspectos construtivos, voltados ao desenvolvimento de TA, levantados a partir das prototipações discutidas ao longo do presente capítulo.

5 DESENVOLVIMENTO DE UM ARCABOUÇO PARA O IOM

Conforme destacado ao longo dos capítulos deste trabalho, o desenvolvimento de soluções assistivas, e consequente disseminação de ferramentas de TA é uma estratégia promissora para a inclusão das pessoas com deficiências na sociedade. Os resultados alcançados pelos estudos conduzidos, em especial na Seção 3.2, mostram que a utilização de meios tecnológicos para amenizar as barreiras impostas aos indivíduos com limitações físicas e cognitivas é uma abordagem que serve para diminuir obstáculos que eles enfrentam cotidianamente. No entanto, devido ao seu caráter multidisciplinar, o desenvolvimento de tais soluções não é uma tarefa trivial. Além das questões tecnológicas, devem ser considerados aspectos sociais, financeiros e psicológicos durante o processo de concepção das TA.

Neste contexto, a utilização de técnicas de desenvolvimento de software para a construção de instrumentos que permitam simplificar este processo, fornecendo não só ferramentas, mas formas de reutilização, é bastante desafiador (HEINEMANN, 2012; OLIVEIRA, 2015). No entanto, aplicações desta natureza apresentam uma complexidade maior, uma vez que trabalham com um nível mais elevado de abstração, sendo necessário utilizar métodos específicos para sua construção.

Visando trabalhar a questão de investigação desta tese, e verificar as hipóteses levantadas, este capítulo destaca o modelo de arcabouço de referência proposto. Ele foi construído a partir da conjunção dos estudos apresentados no Capítulo 3, e das percepções práticas verificadas por meio dos experimentos, descritos no Capítulo 4. A estrutura reutilizável proposta, na forma de arcabouço de software, visa ser um meio que simplifica o processo de desenvolvimento de TA, focada em usuários com movimento de cabeça preservados. Para tanto, as próximas seções compilam as informações levantadas nos capítulos anteriores, apresentando a modelagem do arcabouço para o IOM.

5.1 Requisitos

Para propor um arcabouço voltado ao dispositivo de interação IOM, foi necessário elencar as condições básicas que deveriam ser observadas no processo. Assim, foram enumerados os requisitos funcionais e não funcionais, essenciais para a concepção da solução. Em termos de requisitos funcionais, ou seja, as funcionalidades que devem ser satisfeitas, foram previstos:

- Capturar movimentos de pequena amplitude por meio de sensores externos. Relacionado aos dados de movimentação captados pelos componentes de *hardware* disponíveis no dispositivo e o seu tratamento inicial;
- Converter dados captados a partir dos sensores, em movimentos do cursor na tela de diferentes dispositivos. Requisito complementar ao anterior, relacionado ao uso dos dados captados, no nível de aplicação;
- Fornecer meio alternativo para cliques e seleções na tela, simulando o comportamento do *mouse*. Identificar a viabilidade de realizar este tratamento tanto no nível de aplicação quanto de *firmware*;
- Permitir a calibração do dispositivo, de acordo com as configurações disponíveis. Analisar a possibilidade de ajustes, de forma que seja possível realizar, no nível de aplicação, adaptações de acordo com as necessidades específicas dos usuários; e
- Estabelecer um protocolo de comunicação entre o dispositivo de interação e a estrutura reutilizável de software, para viabilizar o funcionamento adequado da solução desenvolvida.

Os requisitos não funcionais, por sua vez, estão relacionados a necessidades que não podem ser atendidas por meio de funções específicas. Com relação aos requisitos desta natureza, a solução deve:

- Ser multiplataforma, ou seja, deve ser independente de sistema operacional;
- Possuir baixo custo, para que possa ser adotada pela população com faixa de renda menos privilegiada;
- Ser facilmente instalável e ser simples de ser configurada;
- Ser confortável, minimizando a fadiga que o seu uso contínuo pode causar;
- Apresentar bom desempenho de execução; e

- Ser adaptável para diferentes cenários de utilização e necessidades específicas dos usuários.

Estes requisitos são importantes no contexto do trabalho, uma vez que a estrutura reutilizável de software deve ser uma camada de abstração que contempla a efetivação deles de alguma forma. Apresentadas estas especificações iniciais, a próxima seção destaca o processo de proposição e modelagem do arcabouço de referência para o IOM.

5.2 Processo de Modelagem

Arquitetura de software, de um domínio específico, consiste em uma identificação dos elementos estruturais que definem e/ou devem estar presentes em uma classe ou tipo de software. Projetar uma AS desta natureza é importante para desenvolver uma classe de sistemas voltada a um nicho de aplicação, considerando suas características e especificidades. Se projetada adequadamente, a AS serve como um guia para a orientação no desenvolvimento de soluções dentro de seu domínio (MARTINS, 2011).

Conforme detalhado no Capítulo 4, o projeto e a criação de soluções em diferentes contextos de aplicação do IOM foram realizados empiricamente, pela prototipação das mesmas (CARDOSO et al., 2016; XAVIER et al., 2016; PEROBA et al., 2017; CARDOSO et al., 2018). A investigação, realizada durante as prototipações, possibilitou elencar um conjunto mínimo de funcionalidades para compor uma arquitetura do domínio de uso do dispositivo IOM. Neste cenário, foram levantadas tanto funcionalidades recorrentes, quanto propriedades específicas relacionadas a determinadas situações, mas também relevantes para o funcionamento da solução.

A aplicação de diferentes instrumentos de avaliação sobre as soluções desenvolvidas, também contribuiu substancialmente para este processo, uma vez que certos aspectos de utilização só foram percebidos por meio das observações realizadas, ou por meio de comentários e críticas dos testadores das aplicações. A Tabela 11 resume os principais aspectos funcionais percebidos durante os quatro ciclos experimentais de desenvolvimento de aplicações utilizando o dispositivo IOM.

O processo de concepção destas aplicações utilizou abordagens de desenvolvimento *top-down*. Ou seja, as aplicações partiam de uma arquitetura conceitual geral, elaborada considerando o contexto de utilização específico. A partir desta ideia abrangente, as soluções projetadas eram decompostas em componentes, com responsabilidades mais bem definidas. Por sua vez, estes componentes eram divididos em unidades de software (classes, por exemplo), com suas propriedades e métodos particulares. Os resultados obtidos com as prototipações realizadas, segundo esta estratégia, auxiliaram a delinear um conjunto mínimo de funcionalidades gerais que deveriam compor a estrutura inicial de arcabouço de referência.

Tabela 11 – Aspectos funcionais percebidos nas prototipações desenvolvidas.

Aplicação	Funcionalidades relevantes
SCCIOM	Conexão, Comunicação, Configuração, Posicionamento.
Doce Labirinto	Conexão, Comunicação, Configuração, Posicionamento, Eventos.
IOM4Home	Conexão, Comunicação, Configuração, Posicionamento, Seleção de objetos.
IOM4TV	Conexão, Comunicação, Configuração, Posicionamento, Eventos.

Fonte: Elaborado pelo Autor (2019).

Em contrapartida, a modelagem do arcabouço propriamente dito utilizou uma abordagem de construção inversa, ou seja, foi adotada uma metodologia *bottom-up*. Neste caso, o desenvolvimento inicia com elementos elementares que constituem as soluções, indo em direção a arquitetura geral. Assim, foram utilizados, como ponto de partida, os componentes essenciais desenvolvidos anteriormente. A ideia é que estes elementos sejam organizados de forma interativa, originando componentes mais amplos, gradativamente.

Assim, a próxima seção apresenta este processo, destacando as diferentes visões do arcabouço, partindo dos detalhes de implementação em direção ao projeto arquitetural de alto nível.

5.3 Arcabouço para o dispositivo de interação IOM

Para evidenciar as funcionalidades em nível de implementação, foi adotado como forma de representação o diagrama de classes UML (*Unified Modeling Language*) (MEDVIDOVIC et al., 2002). Além de ser um dos diagramas mais disseminados, tanto na academia quanto na indústria de Tecnologia da Informação (TI), ele também é o diagrama em nível de funcionalidade utilizado no modelo C4 (apresentado na Seção 2.1.1). O modelo apresentado nesta seção considera restrições tecnológicas e necessidades exclusivas do IOM, dispositivo utilizado como estudo de caso da tese. As funcionalidades básicas elicitadas foram organizadas em classes que separam as responsabilidades, conforme destacado na Tabela 12.

O modelo proposto relaciona estas funcionalidades com os aspectos básicos encontrados nas soluções desenvolvidas durante as prototipações (vide Seção 4.4) que serviram de base para o processo de modelagem do arcabouço de referência do IOM. Com base nas funcionalidades descritas, foi traçada a modelagem inicial da solução. A proposta busca atender as necessidades básicas do projeto, mapeando as classes essenciais que devem ser contempladas na versão corrente do dispositivo IOM. A próxima seção apresenta este modelo de classes, destacando as funcionalidades essenciais presentes na sua implementação.

Tabela 12 – Funcionalidades essenciais presentes no arcabouço.

Funcionalidade geral	Classe	Método
onConnect	IOMConnection	connect()
onDisconnect	IOMConnection	disconnect()
onFailure	IOM	failure()
setPosition	IOMRobot	setPosition()
getPosition	IOMRobot	getPosition()
onAccelerometerdata	ICommunication	handleAcelData()
config	IOMConfig	config()

Fonte: Elaborado pelo Autor (2019).

5.3.1 Visão de Implementação

Esta seção apresenta o modelo de classes desenvolvido para o arcabouço de referência do IOM. As classes e métodos são descritos de forma geral, com o intuito de permitir a compreensão dos principais elementos que compõem a solução. Por motivos didáticos, foi adotada a linguagem de programação Java para a primeira implementação das classes do modelo¹. Esta escolha se deu pelo formalismo da linguagem, a qual tem sintaxe bem definida e semântica precisa. No entanto, a ideia é que este modelo seja genérico suficiente de forma que possa ser posteriormente implementado em outras linguagens de programação. A Figura 61 destaca o modelo inicial deste arcabouço.

A classe principal, denominada IOM, representa uma instância do dispositivo IOM. Ela é composta por três outros objetos que dão acesso às funcionalidades essenciais do IOM. Os métodos expostos por esta classe, basicamente, fornecem acesso às classes que mantêm as responsabilidades específicas do dispositivo. Por sua vez, os objetos instanciados a partir da classe IOMConnection são responsáveis pelo estabelecimento de conexão entre a aplicação criada por meio do arcabouço e o IOM. Considerando as especificidades do dispositivo IOM, o modelo projetado prevê, inicialmente, dois tipos básicos de conexão:

1. *Serial* - tipo de conexão utilizada na versão 1.0 do dispositivo IOM; e
2. *Bluetooth* - conexão que deve ser implementada a partir da versão 2.0 do dispositivo IOM.

Para permitir o funcionamento destas formas de conexão, independentemente da tecnologia, a IOMConnection trabalha uma interface denominada ICommunication.

¹O Javadoc da API proposta está disponível em http://bit.ly/IOM_API.

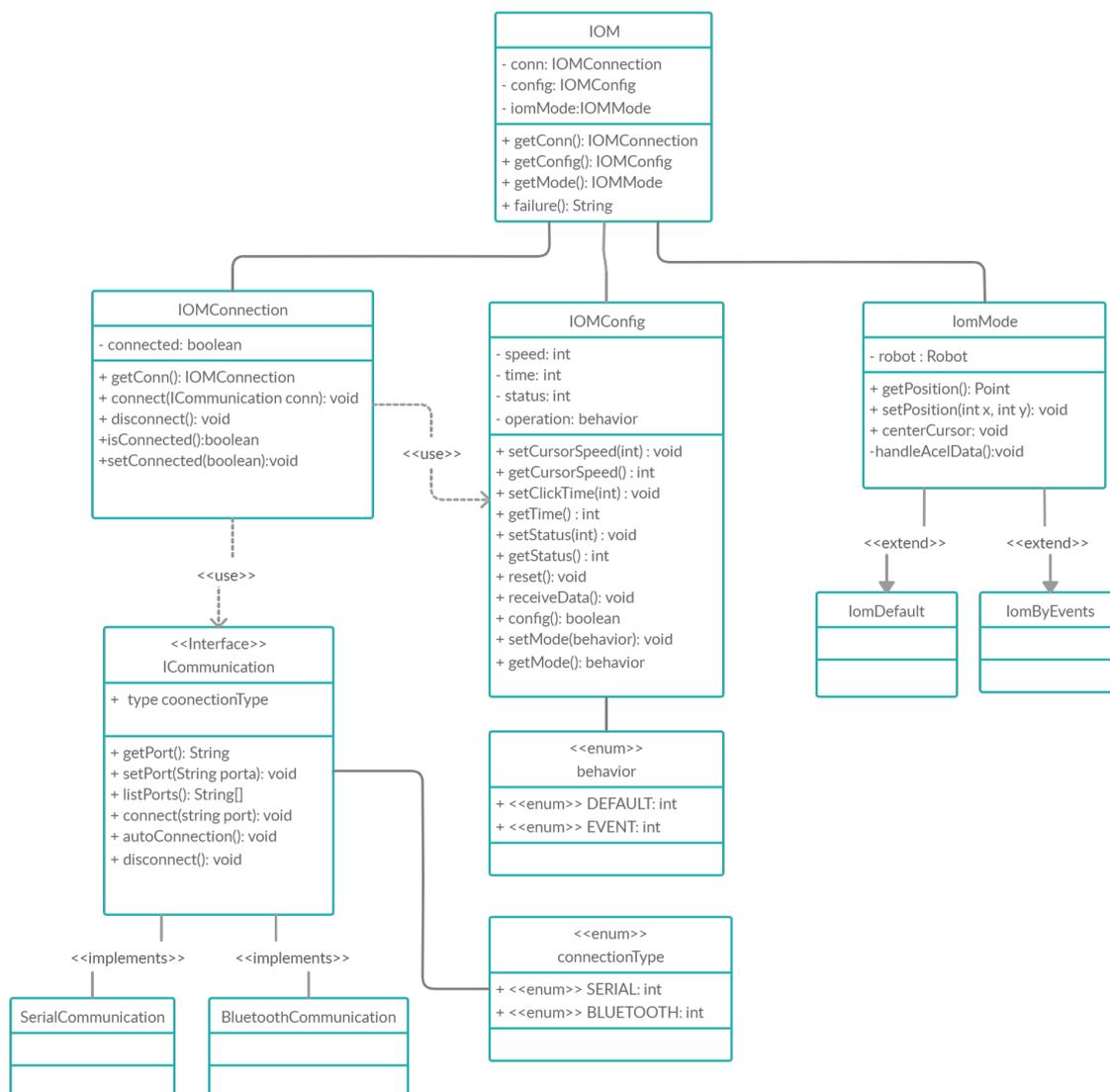


Figura 61 – Modelagem corrente do arcabouço para o IOM.
Fonte: Elaborada pelo Autor (2019).

Esta interface abriga os métodos que devem compor quaisquer classes que pretendam incorporar este comportamento. O método `+connect(Communication conn)`, por exemplo, é responsável por estabelecer a conexão com o dispositivo IOM. Ele recebe qualquer objeto que implemente os comportamentos definidos pelo parâmetro `Communication conn`. Isto garante a passagem polimórfica de objetos dentro do contexto da conexão, diminuindo o nível de acoplamento da solução. Os demais métodos da classe `IOMConnection` provém tarefas gerais de utilização.

No modelo apresentado, são listadas duas classes que implementam esta interface, a `SerialCommunication` e a `BluetoothCommunication`, cada qual sendo responsável por prover suas formas específicas para comunicação. O tipo de conexão estabelecido é armazenado por `ConnectionType`, um `enum` que especifica as possibilidades de conexões aceitas. No nível de implementação, os protótipos exploraram as bibliotecas `jSSC` e `Bluecove`, para comunicações serial e `Bluetooth`, respectivamente. Estas bibliotecas são integradas no desenvolvimento do arcabouço, buscando reaproveitar seus comportamentos.

A classe `IOMConfig`, por sua vez, encapsula as tarefas de ajustes do dispositivo IOM, tais como configurações de tempo de clique, acionar/desligar clique, velocidade do ponteiro e `reset`. Desta forma, ela é dependente de `IOMConnection`, uma vez que só poderá realizar as tarefas estando conectada ao dispositivo. Esta classe possui um `enum` que determina qual o comportamento que será utilizado pelo dispositivo para interação com a aplicação, podendo assumir dois valores inicialmente: movimentação padrão ou baseada em eventos.

Os modos de funcionamento são tratados pela classe `IOMMode`. Ela é responsável por questões relacionadas ao posicionamento e movimentação pela aplicação que interage com o dispositivo. Para tanto, existem duas subclasses que podem ser utilizadas: `IOMDefault` e `IOMEvents`. Elas mantêm métodos de tratamento para os dois modos de operação suportáveis pelo dispositivo. Importante salientar que cada um dos modos de interação, no momento das prototipações, utiliza *firmware* diferentes para execução. As aplicações `SCCIOM`, `IOM4Home` e `IOM4TV (default)`, por exemplo, utilizaram o *firmware* padrão. Já o Doce Labirinto e o `IOM4TV` usaram *firmware* adaptados, para funcionarem baseado em eventos. Ao invés de tratar dentro da aplicação, o *firmware* foi adaptado para enviar apenas quatro movimentos para aplicação de controle.

O atributo `robot` desta classe permite realizar tarefas sobre o cursor, nas aplicações que utilizam o comportamento padrão do dispositivo, em termos de implementação no lado das aplicações.

Com relação à adaptabilidade, inicialmente as classes relacionadas ao modo de interação são as que fornecem possibilidades de adequação mais direta às necessidades específicas dos usuários, ou aos cenários de utilização que surgem. O modelo

criado também comporta a incorporação de novas formas de interação e comunicação. No entanto, estas estão ligadas diretamente à integração de novas tecnologias ao protótipo IOM, tais como adição de novos sensores ou equipamentos que provenham suporte para diferentes formas de conexão, comunicação e percepção do ambiente. É válido ressaltar que este arcabouço de referência faz parte de um projeto de pesquisa em curso no grupo *WeTech*². Assim, é possível que sejam incorporadas novas funcionalidades ao mesmo, à medida que ele evolui.

Seguindo a abordagem *bottom-up* adotada nesta etapa, a próxima seção apresenta os demais diagramas do modelo C4.

5.3.2 Diagramas de Alto Nível

Seguindo a abordagem de modelagem *bottom-up*, utilizando o modelo C4 como referência, foram elaborados os demais diagramas previstos. A Figura 62 destaca o primeiro deles, ou seja, o diagrama de componentes da solução proposta.

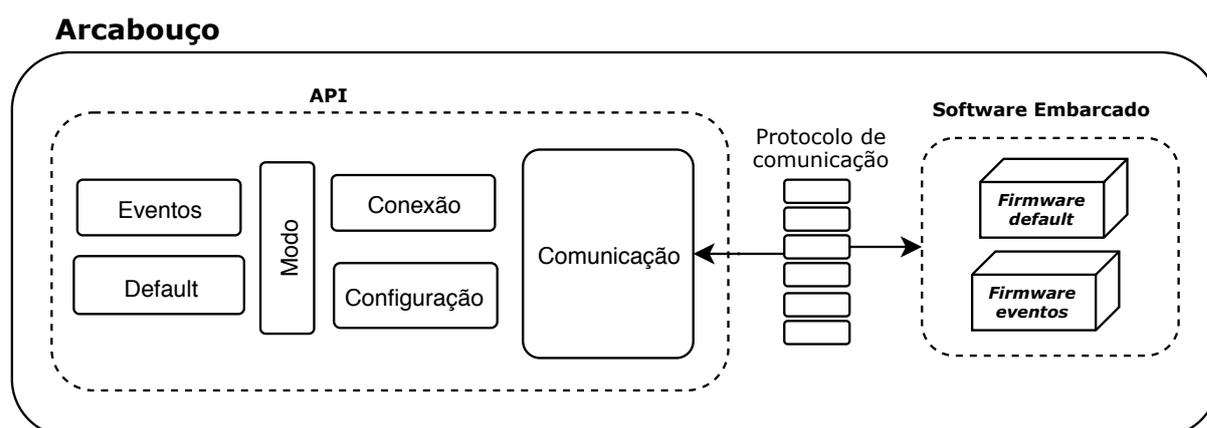


Figura 62 – Diagrama de componentes do arcabouço para o IOM.
Fonte: Elaborado pelo Autor (2019).

O diagrama de componentes, delineado para a solução, destaca os três elementos principais que compõem o arcabouço:

- Software embarcado - *Firmwares* em execução no dispositivo IOM;
- API - Componentes que encapsulam os comportamentos apresentados no diagrama de classes; e
- Protocolo de comunicação - um conjunto de regras de comunicação entre os dois elementos supracitados.

O *Software Embarcado*, como destacado anteriormente na Seção 4.2.2, consiste na *firmware* executada no dispositivo IOM. Ele é responsável por tratar os dados cap-

²*Lib4IOM* – Uma biblioteca para desenvolvimento de software voltado ao dispositivo IOM

turados a partir do acelerômetro e enviá-los de forma que sejam manipulados adequadamente no nível de aplicação. Este elemento também armazena as configurações básicas utilizadas durante os processos de calibração do. Devido às diferenças entre os modos de operação, e também a pequena capacidade de processamento e memória disponível, são utilizadas duas versões deste software:

- *Default* - Executa o comportamento padrão do IOM; e
- *Eventos* - Versão adaptada que utiliza captação de eventos.

Como não foram integradas em um único software, é necessário que a versão seja instalada no dispositivo, sempre que ocorrer uma troca no tipo de interação desejada.

O segundo componente tratado é o *Protocolo de Comunicação*. Ele consiste em uma série de valores que são utilizados para permitir a correta troca de dados entre o software embarcado e a API. A Tabela 13 destaca os valores utilizados atualmente para essa finalidade.

Tabela 13 – Descrição das funcionalidades do Protocolo de comunicação.

Funcionalidade	Valores	Descrição
Velocidade cursor	101 a 114	Determinada a velocidade de movimentação do cursor. Valor 101 é o mais rápido; 114 o mais lento.
Perfis de usuário	700 a 750	Reservado a definição de diferentes perfis de usuário. Não implementado.
Tempo clique nativo	500 a 599	Tempo parado para disparar o clique.
Reset da placa	9493	Valor de status que informa que o <i>firmware</i> será resetado para sua configuração padrão.
Recalibrar	8051	Valor que informa que o IOM está sendo recalibrado.
Calibrado	7400	Valor que informa que o IOM está calibrado.
Calibrando	7408	Status que informa que o IOM está em processo de calibração.
Clique por tempo	8477 - 7077	8477 ativa e 7077 desativa esta propriedade.

Fonte: Elaborado pelo Autor (2019).

Estes valores utilizados para a configuração do dispositivo obedecem a uma lógica utilizada para o cálculo de posicionamento. Em termos de movimentação ao utilizar o comportamento padrão, o acelerômetro capta a movimentação, e envia dados que indicam movimento nos eixos X e Y. Os valores enviados são baseados utilizando como referência a posição inicial configurada pelo usuário. Já quando executando o *firmware* de eventos, ele verifica estes dados de movimentação, mas ao invés de enviar coordenadas, ele envia um valor indicando um dos quatro possíveis movimentos básicos (cima, baixo, esquerda ou direita).

Por fim, o último elemento que compõe o arcabouço consiste na API, ou seja, as classes utilizadas para abstrair as funcionalidades elementares do dispositivo IOM. As classes previamente apresentadas, foram organizadas em componentes que provêm as funcionalidades apresentadas na seção anterior.

Na sequência da modelagem, seguindo a abordagem C4, o diagrama de contêineres apresenta uma visão macro, destacando os ambientes nos quais as partes da solução são executadas. A Figura 63 apresenta este diagrama no contexto do IOM.

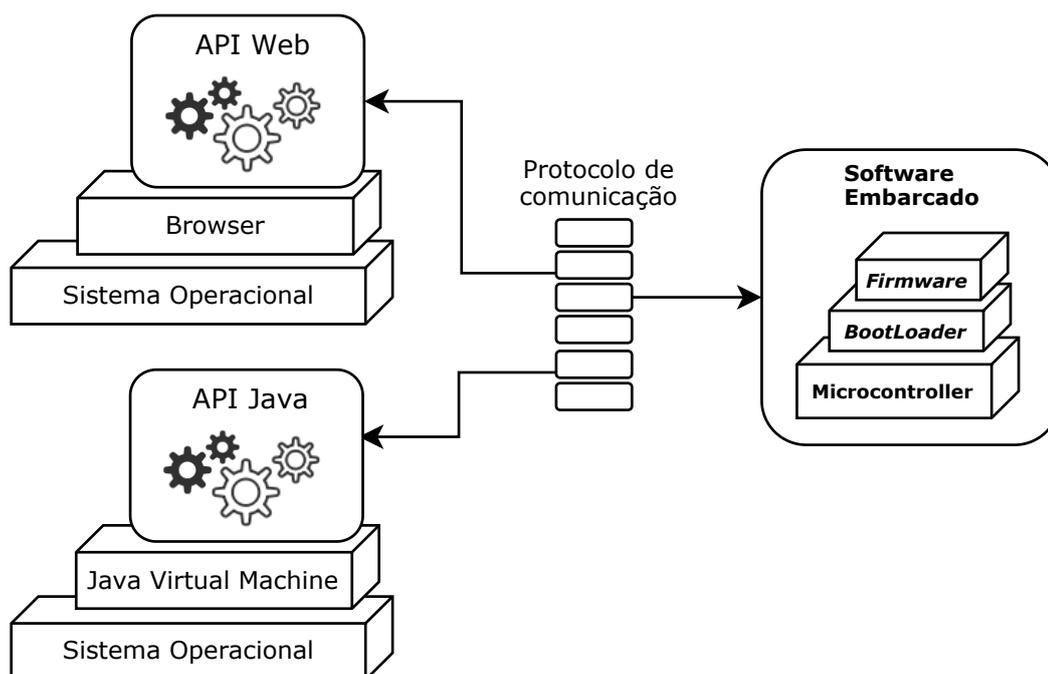


Figura 63 – Diagrama de contêineres para o IOM.

Fonte: Elaborado pelo Autor (2019).

O elemento responsável por executar o *software embarcado* é o microcontrolador presente no dispositivo. Por sua vez, a API prevista é dependente do tipo de tecnologia utilizada em sua implementação. No diagrama, para fins didáticos, são apresentados dois contêineres viáveis, baseados nas prototipações desenvolvidas:

- No caso de uma API criada com tecnologias *web*, o contêiner de execução seriam os navegadores com suporte a ela; e
- Em uma implementação utilizando Java, por exemplo, o contêiner seria um ambiente que suporte a JVM (*Java Virtual Machine*).

A ideia é que a arquitetura conceitual seja concebida, de forma que seja possível a implementação independente de tecnologia.

Por fim, o diagrama de contexto da solução, ou seja, aquele de mais alto nível, já foi destacado anteriormente, sendo observado na Figura 39. Como o próprio nome do diagrama já evidencia, seu principal objetivo é contextualizar o uso da aplicação, em especial destacando o papel do usuário na interação.

5.3.3 Avaliação

Para fins de verificação da solução desenvolvida foram adotadas duas estratégias. Primeiramente buscou-se observar se os módulos do arcabouço produzido (vide Figura 62) estavam em conformidade com os requisitos funcionais levantados na Seção 5.1. A captura de movimentos é realizada pelo acelerômetro, e no nível de aplicação é tratada pelo módulo de comunicação.

Já o tratamento dos dados e as conversões em ações na tela são endereçadas pelos módulos *Evento* e *Default*, dependendo do tipo de interação realizada. As tarefas de calibração e configuração são realizadas pelo módulo de mesmo nome. Por fim, o protocolo de comunicação estabelecido garante a correta troca de dados entre dispositivo e API, certificando o funcionamento adequado da solução desenvolvida.

A segunda verificação foi feita no nível de implementação. Para tanto, as classes previstas no modelo destacado na Figura 61 foram implementadas utilizando a linguagem de programação Java. Foram desenvolvidos testes unitários sobre as classes previstas no modelo inicial. Como o nome sugere, este tipo de procedimento avalia as unidades de trabalho isoladamente, isto é, em termos de codificação, são testes aplicados sobre os métodos ou as funções que compõem um sistema, dependendo do paradigma utilizado. Existe uma variedade de vantagens na utilização dessa técnica, dentre as quais:

- Garantem a cobertura de teste, ou seja, permitem testar mais partes do código, do que testes manuais;
- Previnem regressões na codificação, já que garantem que novas codificações não quebrem outras funcionalidades;
- Favorecem à refatoração; e
- Servem como documentação uma vez que, quando escritos de forma adequada, provém uma descrição de pontos essenciais da aplicação, em termos de funcionalidades.

Alguns fatores motivaram a verificação das classes que compõem o arcabouço por meio desta estratégia. Entre eles, está a disponibilidade de poucas unidades funcionais do dispositivo de interação IOM, para desenvolvimento de testes.

Devido a restrições pontuais, o projeto atualmente encontra-se sem a possibilidade de produção de novos dispositivos, ou mesmo de efetuar reparos a eventuais problemas de *hardware* que possam acometer as unidades em funcionamento. Assim, além de ser uma forma de garantir a qualidade de software, o uso de testes unitários dispensa a necessidade de contar diretamente com o manuseio do dispositivo (TIAN, 2005).

Ao considerar os trabalhos acadêmicos discutidos na Seção 3.2, apesar da utilização desta técnica não ser citada explicitamente, a sua aplicação seria plausível. Os testes desta natureza podem ser aplicados visando garantir as respostas esperadas das funcionalidades, em termos de respostas do software. Já as soluções consolidadas, exploradas na Seção 3.6, por se tratarem de aplicações comerciais, devem lançar uso de técnicas como estas para validação das soluções, ainda não avaliáveis em outro nível de testes, como funcionais, por exemplo.

Como destacado, o teste de unidade avalia o menor dos componentes de uma aplicação de maneira isolada. Neste sentido, os testes foram criados por meio do JUnit³. Em termos de ambiente de desenvolvimento, os testes foram produzidos a partir da IDE *Netbeans* 8.2, utilizando a versão 4.1 do JUnit. Para tanto, inicialmente foi criada uma classe JUnit para cada uma das classes previstas no arcabouço. Elas disparam um série de chamadas aos métodos existentes nas classes do arcabouço, testando as respostas recebidas em reação a cada um dos estímulos.

Assim, considerando especificamente a versão do arcabouço de software em desenvolvimento no momento de escrita desta tese, foram implementados testes unitários para todas as classes concretas destacadas na Figura 61. Cada um dos métodos disponíveis nessas classes foi submetido a pelo menos uma validação via testes de unidade. Além disso, esta avaliação possibilitou a prevenção de problemas com a incorporação de tratamento de erros e exceções específicos do dispositivo IOM. Vale ressaltar que este projeto segue em desenvolvimento. Portanto, tanto o modelo, quanto a implementação desta solução está em constante evolução, a medida que novas funcionalidades e refatorações devem ser efetuadas ao longo do tempo.

O processo de desenvolvimento e testes da aplicação no nível de implementação, em conjunto com os resultados dos diversos ciclos avaliativos das soluções desenvolvidas, levou a uma série de percepções relacionadas tanto a aspectos estruturais quanto de interação com interfaces de controle. A seção a seguir discute estas perspectivas, apresentando ainda um modelo ampliado que busca contemplar questões não abordadas no arcabouço de referência, devido a restrições tecnológicas.

5.4 Aspectos Construtivos para Desenvolvimento de TA

As etapas de análise, modelagem, implementação e avaliação realizadas ao longo dos ciclos de desenvolvimento deste trabalho, permitiram a percepção de diversas nuances no contexto de processo de criação de TA. Os *feedbacks* recebidos das avaliações foram importantes para detectar requisitos e necessidades dos usuários que utilizaram as aplicações criadas, em termos de funcionalidades e interação. Além disso, as análises resultantes do processo de implementação também permitiram perceber

³Framework open-source para geração de testes unitários em Java.

aspectos relacionados à estrutura do software propriamente dito. Estes aspectos são explorados ao longo das próximas subseções.

5.4.1 Aderência a padrões de projeto

Os padrões de projetos simplificam a reutilização de soluções e arquiteturas para construir software. Um projeto de software orientado a objetos adequado possibilita o reuso de componentes preexistentes em sistemas futuros. Neste sentido, foram observados alguns padrões que podem ser incorporados ao arcabouço de referência. Um primeiro padrão percebido que é aplicável no modelo proposto é o `singleton`. O objetivo deste padrão é garantir que um objeto tenha apenas uma instância na aplicação. Esse padrão é útil na classe `IOM` uma vez que garante a geração de apenas um objeto de seu tipo, o qual estará disponível unicamente no escopo de uma aplicação. Ele faz sentido, no contexto atual de utilização do `IOM`, no qual apenas um dispositivo utiliza a aplicação por vez.

Com relação ao modo de funcionamento baseado em eventos, por exemplo, a aplicação necessita, obrigatoriamente, implementar alguma forma de `Observer`, de modo que esteja preparada a trabalhar com os dados enviados adequadamente. O conceito deste padrão de projeto, resumidamente, declara que um objeto tem vários dependentes que são anexados a ele, e quando este objeto muda de estado todos os seus dependentes são avisados. Na prática, um `Listener` é uma forma de implementação deste padrão, já que a aplicação que o implementa fica “ouvindo” (ou observando) alterações que ocorram em algum objeto monitorado.

Ainda com relação a padrões de criação, há espaço de utilização do `Factory Method`. Este padrão define uma interface para criar objetos, mas deixa a responsabilidade de saber o que deve ser instanciado para as classes que o implementam. A interface `ICommunication` foi projetada com esta finalidade. Por fim, o uso do padrão `Adapter` ou `Wrapper` pode ser aplicado às classes que conversam com o dispositivo.

5.4.2 Recomendações para aplicações de controle.

Existem aspectos alusivos ao desenvolvimento das aplicações que interagem com dispositivos de interação alternativa, como o `IOM`, que devem ser considerados. Esta subseção trata de questões relativas a estas interfaces de controle.

Considerando o comportamento padrão, o desenvolvimento do `IOM4Home`, em especial, permitiu identificar funcionalidades relevantes em termos de criação de GUI apropriadas para o tipo de interação desejado. Por exemplo, a necessidade de simplificar a identificação de elementos de tela selecionáveis, por meio do fator gravitacional. Esta *feature* demanda que os componentes passíveis de cliques percebam a aproximação do cursor e o atraiam. A ideia desta funcionalidade é diminuir a fadiga causada

nos usuários durante o processo de seleção de elementos na tela.

Além desta funcionalidade, outros aspectos relacionados a GUI, evidenciados nas prototipações, foram:

- *Feedback* visual - Necessidade de prover clareza nas ações que estão sendo executadas durante a interação com a aplicação; e
- Componentes visuais - Elementos de tela com tamanho razoável e com espaçamento mínimo entre eles.

Nas aplicações de interação dos protótipos *IOM4Home* e *IOM4TV*, estes dois itens supracitados foram reivindicações recorrentes dos usuários, os quais declaravam ter dificuldades em acessar os botões desejados para disparo de alguma ação. Ainda considerando o comportamento padrão do dispositivo IOM, uma questão evidenciada durante as avaliações, em especial sobre o *IOM4TV*, é relacionada à área de navegação do cursor.

Este problema foi realçado na utilização desta aplicação, já que sua interface era consideravelmente pequena quando comparada às demais. O tamanho reduzido desta GUI (vide Figura 57) tornou evidente que a restrição do modo de navegação apenas à área específica ocupada por ela, pode reduzir problemas provenientes da movimentação do cursor para áreas indesejadas. O tratamento deste problema coibiria, por exemplo, o disparo acidental de cliques sobre outras aplicações de forma a alterar a aplicação em execução em primeiro plano.

Complementarmente, a incorporação de um segundo modo de entrada de dados ampliaria as possibilidades de adaptação do uso da solução a outros contextos de usuários. Especificamente para o IOM, a utilização de uma solução multimodal para permitir, por exemplo, alternativas para efetuar o disparo dos cliques. Entre as opções mais viáveis estariam a utilização de um módulo de captação de voz, ou ainda a incorporação de alguma forma de acionamento mecânico, tais como, botões físicos ou pedais adaptados.

Parte destes problemas percebidos durante a verificação dos requisitos levantados pôde ser amenizada com o desenvolvimento da versão baseada em captação de eventos. Adotando esta forma de utilização, a navegação pela interface de controle e o disparo de cliques ficam simplificados, uma vez que o seu funcionamento é restrito a um conjunto mínimo de comandos. Isto foi percebido durante o ciclo do *IOM4TV*. Sua utilização simplificou tarefas relacionadas a:

- Navegação - diminuição da necessidade de uma movimentação contínua para atravessar elementos da aplicação. O funcionamento padrão pode gerar cansaço, devido ao esforço necessário para navegar pela interface;

- Cliques - controle mais adequado sobre o disparo de cliques, o qual funciona a partir de um movimento específico; e
- Interação específica com a interface - restringe a interação da aplicação apenas às suas GUI;

Vale ressaltar que esta solução ainda precisa de validação utilizando testes com usuários finais, com o intuito de verificar seu funcionamento junto a público alvo. Por outro lado, a navegação via eventos não é aplicável ao contexto de utilização direta do computador, por exemplo. Neste cenário é interessante que o usuário tenha controle total do computador que está sendo manuseado. Neste sentido, o tipo de navegação utilizada acaba atrelado às necessidades específicas e às habilidades preservadas dos usuários.

Diante das análises apresentadas até aqui, a próxima subseção destaca um modelo conceitual ampliado. Este modelo busca acomodar questões não implementadas no arcabouço de referência. Ele visa ser uma abstração ampliada, contemplando questões ainda não tratadas no arcabouço.

5.4.3 Modelo Ampliado para Construção de TA

Conforme destacado anteriormente, a definição de uma arquitetura genérica para um determinado domínio de aplicação, permite que a mesma sirva como um guia para o desenvolvimento dentro do contexto abrangido. Esta seção apresenta o modelo ampliado, adicionando requisitos não contemplados pelo arcabouço específico do IOM. Estas questões, que não foram exploradas devido a restrições de diversas naturezas, estão mapeadas no modelo ampliado, destacado pela Figura 64.

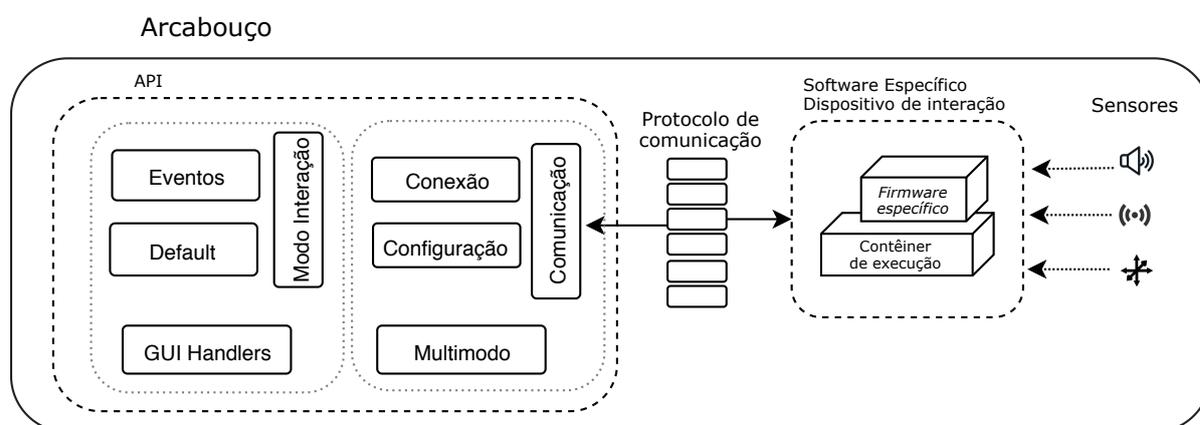


Figura 64 – Modelo de conceitual ampliado.

Fonte: Elaborado pelo Autor (2019).

Como pode ser observado, o modelo proposto é uma expansão do projetado para o IOM. Assim, os três elementos que alicerçam a solução genérica foram mantidos.

O software específico executado no dispositivo de interação utilizado, mostrado na Figura 64, pode ser mapeado no caso do modelo do IOM, para o seu *firmware*. Ou seja, este elemento consiste em algum equipamento, dotado de poder de processamento, o qual provê uma forma de interação alternativa. Este componente usualmente deve ser alimentado por dados capturados a partir de sensores externos, os quais enviam sinais que indicam, por exemplo, a movimentação de alguma parte do corpo preservada. Pelo menos um sensor deve enviar os estímulos capturados, os quais são os dados de entrada que serão tratados pelo dispositivo de interação, e posteriormente pela aplicação.

O segundo elemento do modelo é o protocolo de comunicação. Este componente, também presente na solução específica do IOM, é essencial para uma interpretação correta de interação pelos dois lados. Deve funcionar como uma espécie de contrato, estabelecendo um conjunto de especificações para interlocução adequada. Estes dois componentes são primordiais para o tipo de solução pretendida. Os módulos de tratamento dos dados estarão intrinsecamente conectados a estes elementos, guiando as especificidades que devem ser endereçadas pela API.

Este terceiro elemento provê uma camada de abstração responsável por simplificar a utilização da solução assistiva. Provê acesso a configurações básicas, além de ser capaz de dar suporte a níveis de adaptação que se adéquam à criação de soluções variadas. Neste sentido, os módulos de Comunicação, Configuração e Conexão devem prover acesso direto aos principais ajustes disponíveis no dispositivo utilizado pelo usuário final como forma de interação. Eles fornecem as funcionalidades essenciais para utilização das formas de entrada de dados adotadas pelo usuário, garantindo comunicação adequada e acesso às configurações dos equipamentos utilizados.

Caso a solução seja multimodal, ou seja, dê suporte ao uso de mais de um canal de interação, é preciso tratar este estímulo adequadamente. O modelo apresenta essa opção por meio do componente Multimodo. Este elemento adicional precisa ser consonante com a entrada de dados principal.

O componente Modo de Interação organiza a forma como o usuário utiliza uma aplicação. Neste caso, foram previstas inicialmente duas possibilidades de interação. A primeira (*default*) trata o dispositivo de entrada principal como apontador, semelhante ao uso de um *mouse*. O segundo comportamento adota a abordagem de captação de eventos específicos. Adicionalmente a estes componentes, foi delineado um elemento conceitual extra, referido no modelo como *GUI Handlers*. Este componente seria responsável por tratar questões relacionadas à interface, tais como definir comportamentos para determinar objetos "selecionáveis" dentro do contexto da aplicação desenvolvida (ação gravitacional).

Assim, o modelo ampliado apresentado nesta seção pode ser visto como uma espécie de *guideline*, por destacar diretrizes arquiteturais gerais, as quais podem guiar o

projeto de desenvolvimento em seu domínio específico. Além disso, provê orientações essenciais sobre as formas de navegação, além de padrões visuais recorrentes neste tipo de aplicação.

6 CONCLUSÃO

Esta tese investigou uma série de questões relacionadas ao desenvolvimento de soluções de Tecnologia Assistiva. O objetivo primordial foi identificar e analisar particularidades construtivas, especialmente em termos de software, utilizadas na concepção deste tipo de aplicações. A finalidade foi desenvolver um modelo arquitetural voltado a TA, dentro de um domínio específico, contemplando as funcionalidades e características essenciais elicitadas ao longo deste trabalho.

O Capítulo 1 contextualiza o trabalho, apresentando dados referentes ao cenário da deficiência física/cognitiva no Brasil e no mundo. Devido à ampla quantidade de tipos de deficiências existentes, esta tese restringiu este universo às limitações motoras de membros superiores. Este recorte se fez necessário pois, como justificado, o processo de produção de uma TA está intrinsecamente ligado à potencialização das habilidades preservadas pelos indivíduos. Assim, o fornecimento de uma solução adequada a um cenário geral fica dificultado, uma vez que o universo de variáveis a serem consideradas seria muito amplo.

A Introdução também apresenta a questão de investigação que norteia a tese, além de três hipóteses a serem confirmadas no decorrer do trabalho. Visando alcançar estes objetivos, foi delineada uma metodologia composta por três etapas complementares:

- Método de Pesquisa - definição de uma série de passos para estabelecer uma fundamentação conceitual sólida acerca da área de pesquisa estudada nesta tese;
- Método de Desenvolvimento - estabelecimento de um conjunto de ações práticas voltadas ao desenvolvimento de TA; e
- Estudo de caso - desenvolvimento de um estudo de caso, utilizando os dados levantados a partir das etapas prévias da metodologia.

Todavia, antes de aprofundar nestes passos metodológicos, o Capítulo 2 examinou uma série de conceitos relacionados a dois campos de estudo envolvidos nesta

tese: Engenharia de Software e Tecnologia Assistiva. A finalidade foi fundamentar adequadamente o trabalho, explorando conceitos utilizados durante as demais etapas da metodologia proposta. Em termos de ES, buscou-se apresentar sucintamente um conjunto de métodos e técnicas utilizadas no processo de desenvolvimento de software, em especial, relacionados a soluções reutilizáveis. Em relação a TA, foram observados conceitos e classificações que permitiram a compreensão das nuances desta área, e sua importância na vida das pessoas com limitações das mais diversas naturezas.

O método de pesquisa teve seus resultados destacados no Capítulo 3. Primeiramente foi apresentada a execução do Mapeamento Sistemático de Literatura, e conseqüentemente, o estado-da-arte da área de pesquisa abordada. Os trabalhos recuperados por meio desta técnica foram analisados e discutidos, respondendo às perguntas de pesquisa estabelecidas no protocolo. Este estudo permitiu a elaboração de uma taxonomia para a classificação dos trabalhos remanescentes. Contudo, apesar de proporcionar um aprofundamento sobre o tema, o resultado alcançado, suscitou a necessidade de um estudo qualitativo complementar. Assim, adicionalmente foi conduzida uma análise sobre os arcabouços de software utilizados por soluções comerciais. O objetivo deste levantamento foi complementar à investigação propiciada pelo MSL, especificando as funcionalidades elementares presentes nestas soluções.

Como continuidade desta tese, o Capítulo 4 abordou quatro ciclos de experimentos, seguindo a metodologia de desenvolvimento estabelecida. Neste contexto, o capítulo preliminarmente apresenta o dispositivo de interação alternativa IOM, utilizado como objeto de estudo de caso. O IOM foi empregado em quatro diferentes cenários de aplicação, buscando extrair funcionalidades recorrentes nos ciclos de desenvolvimento realizados, além de eventuais características específicas, relativas aos contextos escolhidos.

No campo de desenvolvimento, a utilização de técnicas exploradas na Seção 2.1, como prototipação e engenharia reversa, por exemplo, comprovam a sua utilidade também no viés assistivo. Outros aspectos relevantes neste processo foram as etapas de avaliação das aplicações. As observações realizadas durante as sessões de testes auxiliaram o levantamento de requisitos e de funcionalidades das diferentes soluções criadas, fornecendo características relevantes no contexto da arquitetura projetada.

A partir da conjunção dos resultados teóricos atingidos pela metodologia de pesquisa e das percepções advindas dos ciclos de desenvolvimento executados, o Capítulo 5 apresenta o projeto e prototipação de um arcabouço para o dispositivo de interação alternativa IOM. O arcabouço desenvolvida, levando em consideração os conceitos apresentados na Seção 2.1.4, derivou uma estrutura de software reutilizável que pode ser caracterizada como um *White-box framework*, já que, para permitir o reuso, é necessário conhecer minimamente a sua organização interna para o seu uso

adequado.

A solução, avaliada por meio de verificação de requisitos e testes unitários, derivou um modelo ampliado de arquitetura em nível conceitual. Este modelo genérico complementa o projeto do arcabouço do IOM, adicionando características necessárias para o funcionamento pleno de uma solução voltada a pessoas com limitações motoras. O objetivo é que esta arquitetura mais ampla funcione como um guia, o qual oriente o desenvolvimento de aplicações de TA neste domínio específico de atuação.

Além dos resultados supracitados, outro objetivo desta tese era tanto refletir acerca das hipóteses levantadas, quanto responder à questão de investigação que norteou o trabalho. Diante do exposto, as próximas seções discorrem sobre ambos os tópicos.

6.1 Discussão das Hipóteses

Conforme destacado na Seção 1.2, três hipóteses foram levantadas neste trabalho. Cada uma delas é analisada individualmente a seguir.

Hipótese 1:

Processos para desenvolvimento de software convencionais atendem as necessidades para a criação de soluções de TA.

Apesar dos trabalhos recuperados pelo MSL não revelarem, explicitamente, a utilização das técnicas de desenvolvimento adotadas na concepção das soluções propostas, diversos indícios presentes nos trabalhos apontam para a aplicação de técnicas convencionais, e com algum nível de adaptação no processo. Isso pode ser percebido inicialmente em termos relacionados a projeto e modelagem das soluções desenvolvidas. As formas de representação utilizadas em diversas das soluções corroboram a isto. A utilização de diagramas semelhantes aos defendidos pelo modelo C4, apresentados na Seção 2.1.1, evidenciam esta utilização no nível de projeto.

Na outra ponta do processo, encontra-se a aplicação de métodos de avaliação sobre as soluções desenvolvidas. As TA são submetidas a testes funcionais para validar se a aplicação executa os requisitos básicos estabelecidos. Neste ponto, pode-se considerar os impactos alcançados quando os usuários finais (ou profissionais de saúde) estão envolvidos no processo. Os resultados destas avaliações são mais ricos quando estes aspectos humanos são contemplados neste processo. Por outro lado, o fato dos trabalhos acadêmicos (listados na Tabela 6 da subseção 3.3.2) usualmente gerarem protótipos, desencorajam a incorporação dos usuários finais no processo construtivo, pois não garantem o suporte adequado após a criação das soluções assistivas. Neste contexto, os aspectos psicológicos e emocionais dos usuários devem ser considerados, de modo a não gerar expectativa, e conseqüentemente frustrações aos usuários. De qualquer forma, a inclusão de usuários ao longo do processo construtivo também é defendida em processos de desenvolvimento convencionais, em especial, os ágeis.

Assim, apesar de não ser explicitamente apresentado nos trabalhos, a adoção de metodologias e técnicas de desenvolvimento de software entre o projeto e a avaliação das aplicações de TA é factível. Neste cenário, a utilização de prototipação, examinada na Seção 2.1.2, foi uma técnica recorrente nos trabalhos recuperados pelo MSL. Em termos práticos, a prototipação se mostrou especialmente eficaz no levantamento de requisitos de um domínio de aplicação.

Diante do exposto, esta hipótese se mostrou verdadeira, ou seja, os processos de desenvolvimento de software convencionais são aplicáveis à construção de TA. No entanto, vale ressaltar que tais processos podem ser adaptados para incluir alguma fase específica de projeto e avaliação. Estas etapas adicionais dizem respeito a especificação ou prescrição de elementos relacionados à limitação que está sendo avaliada. Preferencialmente elas devem envolver os usuários finais ou profissionais de saúde habilitados, com o intuito de perceber as formas mais adequadas para potencializar as habilidades preservadas dos indivíduos beneficiados pela solução de TA.

Hipótese 2:

A concepção de soluções assistivas reutilizáveis é simplificada com a identificação de elementos recorrentes em aplicações dentro de domínio específico, mesmo em diferentes contextos de utilização.

Esta hipótese se mostrou verdadeira, uma vez que elementos essenciais apareceram recorrentemente, em diferentes cenários de uso. Os trabalhos recuperados pelo MSL no Capítulo 3, relacionados ao AsTeRICS, são exemplos nos quais isso ocorre na prática (OSSMANN et al., 2012, 2014; VEIGL et al., 2013). Uma série de aplicações é abordada por estes trabalhos, explorando a estrutura reutilizável proposta para desenvolver soluções particularmente dentro de um âmbito assistivo, mas dentro de contextos de utilização variados. Outro trabalho encontrado na literatura que amplia esta percepção é o GNomon (ACED LOPEZ; CORNO; DE RUSSIS, 2015). Este trabalho destaca o desenvolvimento dos jogos *Ladybugs* e *Invaders*, os quais são concebidos a partir de um *framework* aplicável ao domínio específico de jogos assistivos, utilizando o modo de interação NOMON (MIESENBERGER et al., 2014).

De forma complementar, o emprego desta técnica de desenvolvimento se mostrou bastante eficaz na elicitação de requisitos, visando a proposição de arcabouço de domínio específico. Conforme salientado no Capítulo 4, os ciclos de desenvolvimento sobre o SCCIOM (XAVIER et al., 2016), Doce Labirinto (CARDOSO et al., 2016), IOM4Home (PEROBA et al., 2017) e IOM4TV (CARDOSO et al., 2018) permitiram a detecção de funcionalidades e propriedades que foram aplicadas no arcabouço, destacado no Capítulo 5. Assim, estes ciclos de desenvolvimento executados em cenários de utilização particulares se mostraram relevantes e eficazes tanto para

identificar as necessidades comuns compartilhadas, quanto para constatar aspectos inerentes apenas à conjectura peculiar de cada experimento.

Hipótese 3:

Obstáculos tecnológicos de desenvolvimento impactam diretamente no suporte aos diferentes perfis de usuários finais.

Assim como as hipóteses anteriores, esta suposição também foi confirmada. Dado que as soluções de TA se diferem das convencionais pelo fato de, em um dado momento, precisarem ser adaptadas em algum nível, considerando especificidades do usuários. Isso é necessário pelo fato das habilidades funcionais preservadas geralmente serem aptidões individuais das pessoas que irão usufruir da TA. Assim, as soluções criadas investem na potencialização das capacidades únicas conservadas pelas pessoas beneficiadas por uma TA específica, conforme pôde ser percebido nos trabalhos discutidos na Seção 3.2.

Desta forma, as aplicações e os dispositivos desenvolvidos acabam sendo orientados pelo tipo de deficiência física ou cognitiva que se busca atenuar. No entanto, a criação de soluções genéricas de software acaba sendo valiosa, já que provém um conjunto básico de funcionalidades que simplificam o processo de desenvolvimento inicial. Esta característica pode ser observada, por exemplo, nos arcabouços de software analisados no MSL, e listados na Tabela 4 da Seção 3.3.1. Estas soluções, por sua vez, devem ser adaptáveis e flexíveis de forma que, posteriormente, possam ser especializadas buscando abordar as necessidades individuais dos usuários.

Em termos práticos, isso pôde ser observado durante as avaliações sobre as aplicações que utilizavam o IOM como objeto de estudo (vide Capítulo 4), em especial nos testes efetuados sobre o Doce Labirinto e o IOM4TV. No caso do primeiro, o público infantil apresentou mais destreza e facilidade de utilização do jogo, quando comparado ao público adulto ou sênior. Este fator pode estar relacionado ao fato das crianças possuírem menos experiências vividas, o que faz o processo de aprendizado sobre o funcionamento da solução ser simplificado. Já no IOM4TV, foi percebido que os obstáculos tecnológicos tendem a ser superados à medida que os usuários utilizam as soluções assistivas repetitivamente.

6.2 Discussão: Questão de Investigação

A questão de investigação desta tese que motivou tanto as hipóteses do trabalho, quanto as perguntas de pesquisa abordadas no MSL, indaga:

“Que requisitos arquiteturais devem ser considerados na concepção de soluções de TA voltadas a pessoas com deficiências físicas motoras

nos membros superiores, por forma a que possam ser utilizados em diversos contextos?

Estes requisitos arquiteturais, ou seja, os aspectos construtivos que devem ser considerados de forma a prover uma solução reutilizável para TA, foram elencados durante as diferentes etapas de trabalho. Em termos conceituais, as análises realizadas no Capítulo 3 permitiram vislumbrar os elementos básicos que deveriam constituir esses itens arquiteturais. Tanto por meio dos trabalhos recuperados pelo MSL, quanto da pesquisa qualitativa sobre soluções consolidadas no mercado, foi possível identificar os possíveis requisitos iniciais que deveriam compor a solução proposta.

Associado aos resultados oriundos a partir da metodologia de investigação, a aplicação do processo de desenvolvimento descrito no Capítulo 4, não somente corroborou os elementos levantados com os estudos, como também permitiu ampliar os requisitos que deveriam ser considerados neste trabalho. Estas condições arquiteturais são destacadas no Capítulo 5, com a proposição de um arcabouço para o dispositivo IOM, e suas funcionalidades essenciais. Por fim, este capítulo traz a evolução do modelo implementado para um arcabouço de referência. Este modelo ampliado mantém os componentes utilizados na estrutura desenvolvida para IOM, e o amplia com elementos que não puderam ser utilizados, devido às limitações tecnológicas inerentes ao dispositivo utilizado como objeto de estudo. Assim, os principais requisitos arquiteturais encontrados são relacionados aos aspectos de:

- **Comunicação** - responsável por estabelecer um canal de comunicação eficaz entre aplicação desenvolvida e o dispositivo de interação. Ao longo dos ciclos de desenvolvimento destacados no Capítulo 4, este aspecto apareceu recorrentemente nas prototipações. Em especial, esse item foi bastante trabalhado no SCCIOM (Seção 4.3.1), onde foi delineada a versão inicial do protocolo de comunicação seguido pelo IOM. Posteriormente, a adoção de funcionamento baseado em eventos, observada tanto no desenvolvimento do Doce Labirinto (Seção 4.3.2) quanto do IOM4TV (Seção 4.3.4), ampliou e diversificou os dados trocados entre a aplicação e o IOM (e sua interpretação), dependendo do tipo de interação escolhida. No modelo ampliado este aspecto é destacado nos três elementos que o compõem: (a) componente Comunicação na API; (b) Protocolo de regras de comunicação; e (c) *Firmware* do dispositivo de interação específico;
- **Configuração** - engloba tarefas essenciais de ajustes para uso da solução. Este aspecto é intimamente ligado a Comunicação, já que se preocupa com a identificação do dispositivo que será utilizado, a forma de conexão que será estabelecida e as funcionalidades disponíveis para ajustes. Assim como na Comunicação, esta questão foi abordada em todas as prototipações realizadas (vide Capítulo 4), uma vez que endereça questões essenciais relativas a utilização do

dispositivo de interação IOM. Considerando o estudo de caso, apenas a conexão serial foi implementada. No entanto, o arcabouço desenvolvido já prevê suporte *Bluetooth*, o qual deverá ser incorporado ao dispositivo em sua próxima versão. No modelo ampliado este item é representado na API pelos componentes Conexão e Configuração, destacados na Figura 64;

- **Forma primordial de interação** - define qual a forma preponderante de interação entre usuário e aplicação. No estudo de caso do trabalho, este aspecto está relacionado com a escolha do tipo de interação desejada, ou seja, movimentação contínua ou baseada em eventos. Conforme foi percebido ao longo dos ciclos de desenvolvimento, esta escolha está intrinsecamente ligada ao objetivo da solução proposta. A utilização da navegação por eventos, simplifica a utilização de aplicações que possuem GUI próprias, como no IOM4TV, por exemplo. Além disso, é uma forma de amenizar a fadiga sentida por seus usuários no uso constante da solução. Contudo, a utilização de movimentação contínua é necessária, em especial, para realizar atividades relacionadas ao acesso ao computador e suas aplicações. No modelo ampliado, este elemento é apresentado no componente Modo de Interação, o qual se divide em dois subcomponentes Eventos e *Default*, responsáveis por lidar as especificidades de cada forma de interação. Vale ressaltar que este fator é ligado com os aspectos de Configuração, bem como com os elementos *GUI Handlers* e Multimodo presentes no modelo;
- **Interface** - aspecto relativo ao endereçamento de elementos relacionados com a GUI que está sendo utilizada para interação. Dentre estas atribuições tratadas neste aspecto podem ser destacadas: tornar o objeto selecionável; restringir a navegação à interface; e preparar a interface para resposta a eventos. Durante as prototipações estes elementos foram bastante explorados, em especial no desenvolvimento dos protótipos do IOM4Home e do IOM4TV. Considerando o do modelo da Figura 64, este aspecto é refletido pelo componente *GUI Handlers*;
- **Dados** - refere-se ao tratamento dos dados provenientes dos sensores disponíveis. Durante as prototipações, foram utilizados dados provenientes do acelerômetro acoplado ao IOM, além do tempo, utilizado para disparo de cliques via *Dwell time*. O tratamento deste aspecto é realizado em diversos elementos existentes no modelo. Inicialmente os dados oriundos do sensor são tratados no *firmware* do próprio dispositivo de interação. Já no nível de aplicação, o manuseio destes dados é realizado de acordo com o tipo de interação que está sendo utilizado (*default* ou eventos); e
- **Multimodo** - relacionada à possibilidade de utilização de dois ou mais canais de interação pelos usuários. Como destacado no Capítulo 3, boa parte dos traba-

Esses estudos utilizam pelo menos dois canais de interação para os usuários interagirem. Apesar de não implementada nos protótipos, devido a limitações da versão corrente do dispositivo IOM, esse aspecto se apresenta como um requisito bastante relevante para aprimorar e simplificar o processo de interação. A utilização de entrada de voz, por exemplo, pode ser adotada como com a possibilidade para efetuar os cliques do mouse e outras ações futuramente. Desta forma, esse componente foi previsto no modelo ampliado, através do elemento Multimodo.

Estes foram os aspectos essenciais que devem ser contemplados no planejamento de uma arquitetura de TA. No entanto, o modelo deve ser suscetível ao acolhimento de mudanças, especialmente por meio de especialização das aplicações. Assim, apesar destes aspectos proverem uma base sólida para a estruturação das soluções, uma arquitetura assistiva deve ser capaz de comportar adaptações e especializações. Esta particularidade é importante já que, em um processo de concepção de uma TA, as necessidades e requisitos individuais devem ser avaliados de forma que seja possível introduzir componentes específicos a uma solução que atenda as demandas particulares de um grupo de pessoas ou mesmo de um único indivíduo.

6.3 Limitações

Conforme destacado na Seção 1.4, o trabalho desenvolvido teve algumas delimitações, que influenciaram diretamente os resultados alcançados.

Primeiramente, a utilização do dispositivo de interação IOM como objeto do estudo de caso do trabalho, por exemplo, impactou todas as etapas da tese. Por se tratar de um protótipo ainda em nível acadêmico, a solução não possuía o suporte que dispositivos consolidados no mercado fornecem. Além disso, a quantidade de dispositivos IOM em funcionamento adequado, também foi um obstáculo, já que a produção deles era realizada por meio de integrantes do grupo de pesquisa. Assim, esta dependência gerou certa instabilidade no processo de desenvolvimento. Vale ressaltar ainda, que as aplicações desenvolvidas ficaram limitadas às funcionalidades da versão 1.0 do IOM.

A segunda limitação do trabalho está relacionada com os testes das aplicações desenvolvidas ao longo do Capítulo 4. Apesar da tese focar em pessoas com capacidade de movimentos da cabeça preservados, seria importante realizar os testes das aplicações com pessoas com limitações motoras nos membros superiores. O resultado de testes especificamente com essa população certamente traria um *feedback* mais rico, útil para melhoria dos resultados obtidos. No entanto, a dificuldade em encontrar pessoas deste público-alvo específico, durante os tempos de implementação das soluções desenvolvidas, acabou inviabilizando isso.

6.4 Contribuições

Dentre as contribuições propiciadas por este trabalho pode ser destacada, no nível conceitual, a identificação de abordagens, tendências e aspectos tecnológicos relativos ao desenvolvimento de TA, por meio do MSL. Neste contexto, uma série de trabalhos relacionados a esta revisão bibliográfica foi produzida.

CARDOSO et al. (2016) discutem a etapa de busca exploratória por um artigo âncora para o protocolo MSL. Já em CARDOSO; TAVARES (2017) e CARDOSO et al. (2017) são relatados os resultados iniciais do mapeamento, apresentando tendências e percepções observadas no MSL, além da proposição inicial de uma taxonomia de classificação das TA encontradas nesta revisão bibliográfica. O resultado ampliado destas discussões encontra-se em revisão para publicação 2020 no *SBC Journal on Interactive Systems*, sob o título *Solutions focused on High-Level Assitive Technology: Perceptions and Trends observed from a Systematic Literature Mapping*. Por fim, todos os resultados advindos da execução do MSL estão condensados ao longo do Capítulo 3 e do Anexo A desta tese.

Outra contribuição, também no nível conceitual, foi a condução de uma análise qualitativa a respeito dos dispositivos de interação alternativa disponíveis no mercado. Neste contexto, o trabalho de FERREIRA et al. (2017) levanta e discute as principais funcionalidades e propriedades expostas pelos arcabouços específicos destes dispositivos comerciais, os quais são passíveis de serem adaptados em contextos diversificados de TA. Conforme destacado ao longo do trabalho, esses aspectos auxiliaram na delimitação das funcionalidades essenciais que compuseram o arcabouço de referência para o IOM. Os principais elementos resultantes do estudo supracitado são detalhados na Seção 3.6 deste trabalho.

Especificamente, tratando de trabalhos ligados aos aspectos tecnológicos e de avaliação do IOM, foram realizadas diversas contribuições relacionadas a evolução deste dispositivo, utilizado como estudo de caso na tese. O artigo de MACHADO et al. (2019), por exemplo, relata a concepção do dispositivo IOM, destacando tanto os detalhes arquiteturais quanto as avaliações iniciais realizadas sobre o dispositivo. Já o trabalho de RODRIGUES et al. (2016) apresenta experimentos realizados com o IOM utilizando métodos de avaliação focados em usabilidade, métricas de apontadores e experiência de usuário. Por fim, RODRIGUES et al. (2017) investigam a viabilidade de uso do IOM como dispositivo de apontamento para computador, comparando os resultados alcançados em sua utilização com o desempenho do *HMagic* em KURAUCHI et al. (2015).

Contribuições em um nível mais prático estão ligadas ao projeto, desenvolvimento e a avaliação de soluções de TA utilizando o dispositivo de interação alternativa IOM. Além de serem exemplos de utilização e integração de diversas tecnologias, o desen-

volvimento de tais aplicações pode servir como referência para futuros pesquisadores, já que aborda aspectos relativos a modelagem, implementação e avaliação destas soluções. Neste cenário, é possível destacar os seguintes trabalhos:

- XAVIER et al. (2016) apresentam o **SCCIOM**, software utilizado para configuração e calibração do IOM. Neste contexto, o artigo explora as operações essenciais que permitem a preparação e a configuração do dispositivo para uso;
- CARDOSO et al. (2016) destacam o processo de desenvolvimento e avaliação do **Doce Labirinto**, um jogo interativo que desafia os seus usuários a para atravessar um labirinto físico com o *Sphero*, utilizando o dispositivo IOM para controlar este robô;
- O desenvolvimento do **IOM4Home** (PEROBA et al., 2017), por sua vez, foi aplicada no contexto de AAL, O trabalho destaca a integração de tecnologias de IoT na concepção de uma solução de software que utiliza preceitos de TA para possibilitar o gerenciamento simplificado de ambientes residenciais; e
- Por sua vez, CARDOSO et al. (2018) e CARDOSO et al. (2019) tratam da concepção do **IOM4TV**, uma solução concebida no contexto de utilização de iTV. Esta aplicação proporcionou a integração de diversos componentes de software, permitindo o desenvolvimento de um controle alternativo para uso do sistema +TV4E, por meio do IOM. Uma ampliação dos resultados alcançados com o IOM4TV encontra-se em desenvolvimento para publicação em 2020.

O desenvolvimento destas aplicações faz parte da metodologia de desenvolvimento adotada para a proposição de soluções de software reutilizáveis. Parte deste processo é descrito em CARDOSO et al. (2019), detalhando a etapa de coleta de requisitos no contexto de iTV. Além disso, a própria experiência de aplicação e avaliação do método desenvolvimento, relatada ao longo dos Capítulos 4 e 5, pode ser vista como uma contribuição relevante no contexto da ES, uma vez que destaca o uso de diversas técnicas para produção de uma solução reutilizável, aplicável ao domínio de TA, para pessoas com movimentos de cabeça preservados.

A modelagem do arcabouço, concebido a partir destes ciclos de desenvolvimento, além da implementação da API específica para o IOM, também são contribuições relevantes desta tese. Além disso, a proposição de um modelo de referência é um fruto importante deste trabalho, podendo ser utilizado como base para desenvolvimentos similares. Vale ressaltar ainda que os resultados finais desta tese estão sendo compilados, para serem submetidos em revistas científicas especializadas.

Ademais, ao longo do período de desenvolvimento deste trabalho, foram realizadas duas orientações de TCC com temas interligados às pesquisas do doutorado.

Também foram aprovados diversos projetos de pesquisa, os quais contribuíram pontualmente em questões abordadas por esta tese. Cada um destes projetos resultou em orientações de trabalhos no nível de Iniciação Científica. Por fim, vale recordar que todas estas contribuições, além de outras não revisitadas aqui, estão detalhadas na Seção 1.5 deste trabalho.

6.5 Trabalhos Futuros

Este trabalho discutiu amplamente o desenvolvimento de soluções no contexto de TA de Alto Nível, aplicando técnicas de Engenharia de Software para desenvolver uma série de aplicações assistivas que buscavam auxiliar pessoas com limitações motoras nos membros superiores. A análise e condensação de características e funcionalidades, presentes nestas soluções, permitiu a implementação de um arcabouço reutilizável com funções essenciais disponíveis no dispositivo IOM.

O arcabouço reutilizável, específico do IOM, derivou um modelo ampliado, complementado por um conjunto de funcionalidades, as quais não foram implementadas devido a restrições tecnológicas. Assim, apesar de ter resultado em uma série de contribuições, tanto em termos conceituais quanto práticos, a tese deixa uma variedade de possibilidades que podem ser exploradas em pesquisas futuras.

Com relação ao arcabouço de software do IOM, é interessante refatorar as aplicações desenvolvidas. O objetivo é utilizar o arcabouço, verificando as melhorias reais de sua aplicação. Apesar da verificação feita por análise de cumprimento de requisitos e testes unitários, a sua utilização sobre uma destas aplicações é uma forma de refinar o arcabouço de referência. Esta atividade está prevista como atividade no projeto de pesquisa atualmente em execução no IFSul.

Outra perspectiva pertencente ao escopo do arcabouço, é a incorporação das propriedades previstas no modelo ampliado. Parte desta integração, no entanto, está intimamente relacionada com a evolução do dispositivo, em termos de hardware. Encontra-se em estudo, por exemplo, formas de permitir o uso de mais de uma segunda fonte de entrada de dados no processo de interação. Neste sentido, o provimento a comandos por voz, por exemplo, é um caminho factível, mas que ainda está em estágio inicial de análise de viabilidade. Uma outra oportunidade de atuação está no suporte às questões relacionadas à criação das interface de controle, discutidas na Seção 5.4.2.

Em termos de evolução das classes presentes no arcabouço é interessante implementar os padrões de projeto listados na Seção 5.4.1. A utilização deles é um fator de garantia de qualidade da solução desenvolvida. Posteriormente a isso, um cenário de contribuição possível é a evolução do arcabouço para uma série de componentes individuais, sob a forma de serviços. A ideia seria tratar estes elementos como unida-

des atômicas que se comunicam por meio de um protocolo bem definido. O objetivo seria permitir que estes serviços sejam usados de acordo com as necessidades dos projetos.

Com relação às aplicações apresentadas no Capítulo 4, o IOM4TV será submetido a uma terceira rodada de avaliações, buscando comparar os modos de operação padrão e evento, correntemente implementados. Neste sentido, outro possível desdobramento diz respeito à reestruturação do *IOM4Home*, de forma que ele dê suporte ao funcionamento no modo de interação por eventos.

Em termos do dispositivo IOM propriamente dito, uma série de avanços pode ser explorada futuramente. A implementação de melhorias no dispositivo, no entanto, passa por um processo de incorporação de tecnologias ao projeto do IOM. Entre estas oportunidades de investigação e desenvolvimento é possível destacar:

- Protocolo de comunicação - refatoração deste artefato, buscando simplificá-lo e reorganizá-lo;
- *Bluetooth* - incorporação de comunicação sem fio, prevista para a próxima versão do IOM;
- *Driver*: desenvolvimento de um *driver* específico para o IOM; e
- *Firmware*: refatoração para integração dos modos de funcionamento *default* e por eventos.

A medida que as novas funcionalidades forem incorporadas ao dispositivo, devem ser refletidas no arcabouço, e conseqüentemente nas aplicações, de forma que evoluam conjuntamente.

REFERÊNCIAS

ACED LOPEZ, S.; CORNO, F.; DE RUSSIS, L. Gnomon: Enabling dynamic one-switch games for children with severe motor disabilities., **Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems**, p.995–1000, 2015.

ALONSO, E. C. **Some Contributions to Smart Assistive Technologies**. 2015. Tese (Doutorado em Ciência da Computação) — The University of Basque Country.

ALSHAMRANI, A.; BAHATTAB, A. A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model., **International Journal of Computer Science Issues (IJCSI)**, v.12, n.1, p.106, 2015.

ALVES, A. K. S. **TeamBridge**: Middleware para adaptação de games e controles de reabilitação motora. 2018. Dissertação (Mestrado em Ciência da Computação) — Brasil.

ALY, S.; KBAR, G.; ABDULLAH, M.; AL-SHARAWY, I. Modeling the Interaction and Control of Smart Universal Interface for Persons with Disabilities., **International Conference on Human Aspects of IT for the Aged Population**, p.377–388, 2015.

ANTUNES, R. A. et al. Intelligent human-computer interface for improving pointing device usability and performance., **12th IEEE International Conference on Control and Automation (ICCA)**, p.714–719, 2016.

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software Architecture in Practice**. 3.ed. [S.I.]: Addison-Wesley Professional, 2012.

BAYER, J. et al. PuLSE: A Methodology to Develop Software Product Lines., **SSR**, v.99, p.122–131, 1999.

BECK, K. et al. Manifesto for agile software development., **Software Development**, 2001.

BERSCH, R. **Introdução a Tecnologia Assistiva**. Disponível em: <http://www.assistiva.com.br/Introducao_Tecnologia_Assistiva.pdf>.

BOSCH, J. **Design and use of software architectures: adopting and evolving a product-line approach**. [S.l.]: Pearson Education, 2000.

BOUGIE, I. et al. ISO 9999 Assistive Products for Persons with Disability: Classification and Terminology., **The Engineering Handbook of Smart Technology for Aging, Disability, and Independence**, p.117–126, 2008.

BOURQUE, P.; FAIRLEY, R. E. et al. **Guide to the software engineering body of knowledge (SWEBOK (R))**: Version 3.0. [S.l.]: IEEE Computer Society Press, 2014.

BRODERICK, T.; MACKAY, D. J. Fast and flexible selection with a single switch., **PloS one**, v.4, n.10, p.e7481, 2009.

BROWN, S. Software Architecture for Developers., **Coding the Architecture**, 2013.

BROWN, T. et al. Design Thinking: uma metodologia poderosa para decretar o fim das velhas ideias., **Rio de Janeiro: Elsevier**, 2010.

CANTONI, V.; MERLANO, L.; NUGRAHANINGSIH, N.; PORTA, M. Eye tracking for cultural heritage: a gaze-controlled system for handless interaction with artworks. In: INTERNATIONAL CONFERENCE ON COMPUTER SYSTEMS AND TECHNOLOGIES 2016, 17., 2016. **Proceedings...** [S.l.: s.n.], 2016. p.307–314.

CARAVAU, H.; SILVA, T. +TV4E: An Interactive Television Platform as a Support to Broadcast Information About Social Services., **Libro de Aplicaciones y Usabilidad de la Televisión Digital Interactiva**, v.11, p.17, 2016.

CARDOSO, R. C.; COSTA, V. K.; RODRIGUES, A. S.; TAVARES, T. A. Análise de Frameworks para o Desenvolvimento de Produtos voltados a Tecnologia Assistiva., **XVII Encontro de Pós Graduação da Universidade Federal de Pelotas**, 2016.

CARDOSO, R. C.; COSTA, V. K.; RODRIGUES, A. S.; TAVARES, T. A. Uma Análise sobre o desenvolvimento de Soluções voltadas a Tecnologia Assistiva de Alto Nível., **XIX Encontro de Pós Graduação da Universidade Federal de Pelotas**, 2017.

CARDOSO, R. C. et al. Doce Labirinto: Experiência de jogo utilizando interação baseada em movimentos da cabeça e recursos tangíveis., **XV Simpósio Brasileiro de Jogos e Entretenimento Digital**, 2016.

CARDOSO, R. C.; TAVARES, T. A. Uma Análise das Características relacionadas ao Desenvolvimento de Aplicações voltadas a Tecnologia Assistiva., **Workshop de Teses**

e Dissertações do XVI Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC), 2017.

CARDOSO, R. et al. IOM4TV: An AT-Based Solution for People with Motor Disabilities Supported in iTV. In: IBEROAMERICAN CONFERENCE ON APPLICATIONS AND USABILITY OF INTERACTIVE TV, 2018. **Anais...** [S.l.: s.n.], 2018. p.99–114.

CARDOSO, R. et al. Enhancing the Architectural Requirements of an AT Software Framework through iTV Ecosystem., **Proceedings of the 6th Iberoamerican Conference on Applications and Usability of Interactive TV (JAUTI 2019)**, 2019.

CARUSO, M. et al. My-world-in-my-tablet: an architecture for people with physical impairment., **International Conference on Human-Computer Interaction**, p.637–647, 2013.

COOK, A. M.; POLGAR, J. M. **Assistive Technologies-E-Book**:Principles and Practice. [S.l.]: Elsevier Health Sciences, 2014.

COSTA, D. L. V. **Método para desenvolvimento de aplicativos com abordagem de design participativo assistivo-DADPA**. 2018. Tese (Doutorado em Ciência da Computação) — .

CRUZ, A. J. O. et al. New Solutions for Old Problems: Use of Interfaces Human/Computer to Assist People with Visual and/or Motor Impairment in the Use of DOSVOX and microFênix. In: . [S.l.]: Springer, 2015. p.1073–1079.

CRUZ, D. M. C. da; EMMEL, M. L. G. Papéis ocupacionais de pessoas com deficiências físicas: diferenças de gênero e ciclos de desenvolvimento., **Revista Baiana de Terapia Ocupacional**, v.1, n.1, 2012.

DHARA, K. M.; DHARMALA, M.; SHARMA, C. K. A Survey Paper on Service Oriented Architecture Approach and Modern Web Services., , 2015.

FERREIRA, A. O. L. et al. A comparative analysis about natural user interface technologies. In: ESTENDIDOS DO XXIII SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB, 2017. **Anais...** [S.l.: s.n.], 2017. p.145–148.

FOWLER, M. Design - Who needs an architect?, **IEEE Software**, v.20, n.5, p.11–13, Sept 2003.

GALANTE, A.; MENEZES, P. A gaze-based interaction system for people with cerebral palsy., **Procedia Technology**, v.5, p.895–902, 2012.

GALVÃO FILHO, T. A. A Tecnologia Assistiva: de que se trata?, **Conexões: educação, comunicação, inclusão e interculturalidade**. Porto Alegre: Redes Editora, v.252, p.207–235, 2009.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. Design Patterns: Elements of Reusable Object-Oriented Software Addison-Wesley., **Reading, MA**, p.1995, 1995.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Padrões de Projeto–Soluções Reutilizáveis de Software Orientado a Objetos**. [S.l.]: Bookman–Porto Alegre, 2004.

GARCIA, J. C. D.; GALVÃO FILHO, T. A. Pesquisa nacional de tecnologia assistiva., **São Paulo: ITS Brasil/MCTI-Secis**, v.22, 2012.

GIL, A. C. Como elaborar projetos de pesquisa. São Paulo: Atlas, 2002., **Métodos e técnicas de pesquisa social**, v.6, p.22–23, 2009.

HASSENZAHN, M. User experience(UX): towards an experiential perspective on product quality., **Proceedings of the 20th Conference on I Interaction Homme-Machine**, p.11–15, 2008.

HEINEMANN, L. **Effective and efficient reuse with software libraries**. 2012. Tese (Doutorado em Ciência da Computação) — Technische Universität München.

HERSH, M. A. The design and evaluation of assistive technology products and devices part 1: Design., **International Encyclopedia of rehabilitation**, 2010.

HUNT, A.; THOMAS, D. **The Pragmatic Programmer: From Journeyman to Master**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

HUO, X.; PARK, H.; KIM, J.; GHOVANLOO, M. A dual-mode human computer interface combining speech and tongue motion for people with severe disabilities., **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, v.21, n.6, p.979–991, 2013.

HUSSEY, A. C. S.; COOK, S. Assistive Technologies: Principles and Practice., **Baltimore: Mosby**, 2002.

IBGE. **Instituto Brasileiro de Geografia e Estatística - Censo Demográfico 2010**. <http://www.ibge.gov.br/home/estatistica/populacao/censo2010/>. Disponível em: <<http://www.ibge.gov.br/home/estatistica/populacao/censo2010/>>.

IEEE. IEEE Standard Glossary of Software Engineering Terminology., **IEEE Std 610.12-1990**, p.1–84, Dec 1990.

JÄÄSKELÄINEN, R. Think-aloud protocol., **Handbook of translation studies**, v.1, p.371–374, 2010.

KARPOV, A.; RONZHIN, A. A universal assistive technology with multimodal input and multimedia output interfaces., **International Conference on Universal Access in Human-Computer Interaction**, p.369–378, 2014.

KOO, B.; KIM, J.; CHO, J. Leap motion gesture based interface for learning environment by using leap motion. In: HCI KOREA, 2014. **Proceedings...** [S.l.: s.n.], 2014. p.209–214.

KOUROUPETROGLOU, G. **Assistive technologies and computer access for motor disabilities**. [S.l.]: IGI Global, 2013.

KURAUCHI, A.; FENG, W.; MORIMOTO, C.; BETKE, M. HMAGIC: head movement and gaze input cascaded pointing., **Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments**, p.47, 2015.

LAUAND, G. **Fontes de informação sobre tecnologia assistiva para favorecer a inclusão escolar de alunos com deficiências físicas e múltiplas. 2005. 224 f.** 2005. Tese (Doutorado em Ciência da Computação) — Tese (Doutorado em Educação Especial)—Programa de Pós-Graduação em Educação Especial, Universidade Federal de São Carlos, São Carlos.

LAW, P. Law 100-407., **The Technology-Related Assistance for Individuals with Disabilities Act of**, 1988.

LEVY, P. C. et al. Activelris: uma solução para comunicação alternativa e autonomia de pessoas com deficiência motora severa., **Proceedings of the 12th Brazilian Symposium on Human Factors in Computing Systems**, p.42–51, 2013.

LIDDY, E. D. et al. **User interface and other enhancements for natural language information retrieval system and method**. [S.l.]: Google Patents, 2000. US Patent 6,026,388.

LIMA, V. C. M.; NETO, A. G. S. S.; EMER, M. C. F. P. Investigação Experimental e Práticas Ágeis: Ameaças à Validade de Experimentos envolvendo a Prática Ágil Programação em Par/Experimental. Investigation and Agile Practices: Threats to the Validity of Experiments involving the pair programming agile practice., **Revista Electronica de Sistemas de Informação**, v.13, n.1, p.1, 2014.

LIN, Y.; BREUGELMANS, J.; IVERSEN, M.; SCHMIDT, D. An Adaptive Interface Design (AID) for enhanced computer accessibility and rehabilitation., **International Journal of Human-Computer Studies**, v.98, p.14–23, 2017.

LINDEN, F. van der. Engineering Software Architectures, Processes and Platforms for System Families-ESAPS Overview. In: INTERNATIONAL CONFERENCE ON SOFTWARE PRODUCT LINES, 2002. **Anais...** [S.l.: s.n.], 2002. p.383–397.

LOPEZ-HERREJON, R. E.; LINSBAUER, L.; EGYED, A. A systematic mapping study of search-based software engineering for software product lines., **Information and software technology**, v.61, p.33–51, 2015.

LUPU, R. G.; UNGUREANU, F.; BOZOMITU, R. G. Mobile embedded system for human computer communication in assistive technology. In: INTELLIGENT COMPUTER COMMUNICATION AND PROCESSING (ICCP), 2012 IEEE INTERNATIONAL CONFERENCE ON, 2012. **Anais...** [S.l.: s.n.], 2012. p.209–212.

LUPU, R. G.; UNGUREANU, F.; SIRITEANU, V. Eye tracking mouse for human computer interaction., **E-Health and Bioengineering Conference (EHB)**, 2013, p.1–4, 2013.

MACEDO, A. A.; BARANAUSKAS, J. A.; BULCÃO-NETO, R. d. F. The Evolution of a Healthcare Software Framework: Reuse, Evaluation and Lessons Learned. In: FEDERATED CONFERENCE ON COMPUTER SCIENCE AND INFORMATION SYSTEMS (FEDCSIS), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.1–9.

MACHADO, M. e. a. **Óculos Mouse**: Mouse Controlado pelos movimentos da cabeça do usuário. Brazilian Patent INPI n. PI10038213.

MACHADO, M. et al. An adaptive Hardware and Software Based Human Computer Interface for People with Motor Disabilities., **IEEE Latin America Transactions**, 2019.

MANRESA YEE, C.; MUNTANER, J. J.; ARELLANO, D. A motion-based interface to control environmental stimulation for children with severe to profound disabilities., **CHI'13 Extended Abstracts on Human Factors in Computing Systems**, p.7–12, 2013.

MARTINEZ, J. et al. Bottom-up adoption of software product lines: a generic and extensible approach. In: INTERNATIONAL CONFERENCE ON SOFTWARE PRODUCT LINE, 19., 2015. **Proceedings...** [S.l.: s.n.], 2015. p.101–110.

MARTINS, G. D. **Arquitetura de software de domínio específico para apoio ao gerenciamento ágil de projetos distribuídos**. 2011. Tese (Doutorado em Ciência da Computação) — Universidade de São Paulo.

MARTINS, J. M.; RODRIGUES, J. M.; MARTINS, J. A. Low-cost natural interface based on head movements., **Procedia Computer Science**, v.67, p.312–321, 2015.

MATEEN, A.; KAUSAR, S.; SATTAR, A. R. A software reuse approach and its effect on software quality, An empirical study for the software industry., **arXiv preprint arXiv:1702.00125**, 2017.

MCCULLAGH, P.; GALWAY, L.; LIGHTBODY, G. Investigation into a mixed hybrid using SSVEP and eye gaze for optimising user interaction within a virtual environment., **International Conference on Universal Access in Human-Computer Interaction**, p.530–539, 2013.

MEDEIROS, F. P. A. d. **Uma abordagem de monitoramento abrangente das interações sociais em ambientes colaborativos virtuais de aprendizagem como suporte a presença docente**. 2013. Tese (Doutorado em Ciência da Computação) — .

MEDVIDOVIC, N.; ROSENBLUM, D. S.; REDMILES, D. F.; ROBBINS, J. E. Modeling software architectures in the Unified Modeling Language., **ACM Transactions on Software Engineering and Methodology (TOSEM)**, v.11, n.1, p.2–57, 2002.

MIESENBERGER, K. et al. ATLab: An App-Framework for Physical Disabilities., **Journal on Technology & Persons with Disabilities**, v.1, p.46–56, 2014.

MULFARI, D.; CELESTI, A.; FAZIO, M.; VILLARI, M. Human-computer Interface Based On Iot Embedded Systems For Users With Disabilities. In: . [S.I.]: Springer, 2015. p.376–383.

MULFARI, D.; CELESTI, A.; VILLARI, M. A computer system architecture providing a user-friendly man machine interface for accessing assistive technology in cloud computing., **Journal of Systems and Software**, v.100, p.129–138, 2015.

MULFARI, D. et al. Using embedded systems to spread assistive technology on multiple devices in smart environments., **Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on**, p.5–11, 2014.

NETO, N. et al. **VOICECONET: A Collaborative Framework for Speech-Based Computer Accessibility with a Case Study for Brazilian Portuguese**. [S.I.]: INTECH Open Access Publisher, 2012.

NIELSEN, J. **Usability engineering**. [S.I.]: Elsevier, 1994.

NUNES, D. P. et al. Capacidade funcional, condições socioeconômicas e de saúde de idosos atendidos por equipes de Saúde da Família de Goiânia (GO, Brasil)., **Ciência & Saúde Coletiva**, v.15, p.2887–2898, 2010.

OKUMURA, M. L. M. **Modelo Conceitual de Projeto Orientado para Tecnologia Assistiva - MPOTA**. 2017. Tese (Doutorado em Ciência da Computação) — Pontifícia Universidade Católica do Paraná.

OLIVEIRA, E. R. d. et al. **O uso de dispositivo multi-sensoriais para promoção do brincar em crianças com paralisia cerebral**. 2015. Dissertação (Mestrado em Ciência da Computação) — .

OLIVEIRA, M. S. de. **On the Use of Visualization for Supporting Software Reuse**. 2015. Tese (Doutorado em Ciência da Computação) — Universidade Federal do Rio de Janeiro.

OSSMANN, R. et al. Making the playstation 3 accessible with AsTeRICS., **International Conference on Computers for Handicapped Persons**, p.443–450, 2012.

OSSMANN, R. et al. AsTeRICS, a flexible AT construction set., **International Journal of Adaptive Control and Signal Processing**, v.28, n.12, p.1475–1503, 2014.

PATTERNS, M. **Microsoft Application Architecture Guide**. 2nd.ed. Redmond, WA, USA: Microsoft Press, 2009.

PAUL, B. et al. A context-aware user interface for wireless personal-area network assistive environments., **Wireless personal communications**, v.69, n.1, p.427–447, 2013.

PENNSYLVANIA, D. R. N. of. **Assistive Technology for Persons with Disabilities: An Overview**. [Online; accessed 13-Mar-2017], <http://disabilityrightspa.org/File/publications/assistive-technology-for-persons-with-disabilities---an-overview.pdf>.

PEROBA, J. A. et al. An IoT application for Home Control focused on Assistive Technology. In: ESTENDIDOS DO XXIII SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB, 2017. **Anais...** [S.l.: s.n.], 2017. p.119–122.

PEROBA, J.; CARDOSO, R.; COSTA, V. da. **IOM4HOME: Uma Solução de Ambiente Inteligente para Pessoas com Deficiência Físico Motora**. 2018. B.S. Thesis — .

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic Mapping Studies in Software Engineering., **EASE**, v.8, p.68–77, 2008.

PINHEIRO JR, C. G. **Assistive technology for the severely motor impaired by using online processing of motor unit action potentials of facial muscles**. 2013. Tese (Doutorado em Ciência da Computação) — Federal University of Uberlândia, Brazil.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software-8ª Edição**. [S.I.]: McGraw Hill Brasil, 2016.

QUIVY, R.; VAN CAMPENHOUDT, L. **Manual de investigação em ciências sociais**. [S.I.]: Gradiva Lisboa, 1998.

RADABAUGH, M. P. **Study on the financing of assistive technology devices of services for individuals with disabilities—a report to the President and the Congress of the United States, National Council on Disability**. Disponível em: <[http://http://www.ncd.gov/publications/1993/Mar41993/](http://www.ncd.gov/publications/1993/Mar41993/)>.

RADATZ, J.; GERACI, A.; KATKI, F. IEEE standard glossary of software engineering terminology., **IEEE Std**, v.610121990, n.121990, p.3, 1990.

RANTANEN, V. et al. A Wearable, Wireless Gaze Tracker with Integrated Selection Command Source for Human-Computer Interaction., **IEEE Transactions on Information Technology in BioMedicine**, v.15, n.5, p.795–801, 2011.

ROBERTS, D.; JOHNSON, R. Evolving frameworks., **Pattern languages of program design**, v.3, 1996.

RODRIGUES, A. et al. Evaluation of a Head-Tracking Pointing Device for Users with Motor Disabilities., **10th International Conference on Pervasive Technologies Related to Assistive Environments**, 2017.

RODRIGUES, A. S. et al. Evaluation of the Use of Eye and Head Movements for Mouse-like Functions by Using IOM Device., **International Conference on Universal Access in Human-Computer Interaction**, p.81–91, 2016.

SALNIKOV, A. et al. Natural interface to improve human-computer interaction for people with upper limb disabilities: exploring the potentials of voice input and hand gestures in application development to improve the communication possibilities of people with motor disorders., , 2014.

SCARDOVELLI, T. A.; FRÈRE, A. F. The design and evaluation of a peripheral device for use with a computer game intended for children with motor disabilities., **Computer methods and programs in biomedicine**, v.118, n.1, p.44–58, 2015.

SEABRA, M.; MENDES, E. G. Escolha dos Recursos de Alta Tecnologia Assistiva para a Inclusão Escolar de Crianças com Paralisia Cerebral., **V Congresso Brasileiro Multidisciplinar de Educação Especial**, 2009.

SILVA, C. **Geração automática de conteúdo audiovisual informativo para seniores**. 2017. Dissertação (Mestrado em Ciência da Computação) — Universidade de Aveiro, Aveiro, Portugal.

- SILVA, T. **Identificação de utilizadores seniores em televisão interativa**. 2014. Tese (Doutorado em Ciência da Computação) — Universidade de Aveiro, Aveiro, Portugal.
- SILVA, T. et al. +TV4E: Interactive Television as a support to push information about social services to the elderly., **Procedia Computer Science**, v.100, p.580–585, 2016.
- SILVA, T.; REIS, L.; HERNÁNDEZ, C.; CARAVAU, H. Building informative audio-visual content automatically: a process to define the key aspects. In: IBEROAMERICAN CONFERENCE ON APPLICATIONS AND USABILITY OF INTERACTIVE TV (JAUTI 2017), 6., 2017. **Proceedings...** [S.l.: s.n.], 2017. p.132–143.
- SOBRAL, A. J. F. d. C. **Frameworks Verticas vs Horizontais Gestão de Recursos Humanos com o OFBiz**. 2015. Tese (Doutorado em Ciência da Computação) — .
- SOMMERVILLE, I. Software engineering 9th Edition., **ISBN-10137035152**, 2011.
- SOMMERVILLE, I. et al. **Engenharia de software**. [S.l.]: Pearson Prentice Hall São Paulo, 2011. v.9.
- SOUSA, L. R. N. d. **Framework para criação de interfaces gestuais para aprendizagem de caligrafia de crianças com transtorno do desenvolvimento da coordenação**. 2017. Tese (Doutorado em Ciência da Computação) — . Faculdade de Computação e Informática (FCI).
- STOICA, M.; GHILIC-MICU, B.; MIRCEA, M.; USCATU, C. Analyzing Agile Development-from Waterfall Style to Scrumban., **Informatica Economica**, v.20, n.4, 2016.
- TAYLOR, S. J.; BOGDAN, R.; DEVAULT, M. **Introduction to qualitative research methods: A guidebook and resource**. [S.l.]: John Wiley & Sons, 2015.
- TELES, V. M. **Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade**. [S.l.]: Novatec Editora, 2017.
- TIAN, J. **Software quality engineering: testing, quality assurance, and quantifiable improvement**. [S.l.]: John Wiley & Sons, 2005.
- TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. **Introdução à engenharia de software experimental**. [S.l.]: UFRJ, 2002.
- TRIPATHY, D.; RAHEJA, J. L. Design and Implementation of Brain Computer Interface Based Robot Motion Control., **FICTA (2)**, p.289–296, 2014.
- VALE, T. et al. Twenty-eight years of component-based software engineering., **Journal of Systems and Software**, v.111, p.128–148, 2016.

VEIGL, C. et al. Model-based design of novel human-computer interfaces—The Assistive Technology Rapid Integration & Construction Set (AsTeRICS)., **Biosignals and Biorobotics Conference (BRC), 2013 ISSNIP**, p.1–7, 2013.

VIANNA, M. **Design thinking**: inovação em negócios. [S.l.]: Design Thinking, 2012.

WANGLER, B.; PAHEERATHAN, S. Horizontal and vertical integration of publisheral IT systems., **Information Systems Engineering**, 2000.

WHO. **Disability and rehabilitation**: World report on disability. Disponível em: <<http://bit.ly/2PW7ky9>>.

WOHLIN, C. et al. **Experimentation in software engineering**. [S.l.]: Springer Science & Business Media, 2012.

XAVIER, K. F. et al. Desenvolvimento de Software de Configuração para Dispositivo de Tecnologia Assistiva Interface Óculos Mouse (IOM)., **XXV Congresso de Iniciação Científica da Universidade Federal de Pelotas**, 2016.

ANEXO A PROTOCOLO MSL

Rafael Cunha Cardoso

**Protocolo utilizado no Mapeamento Sistemático de Literatura do tema:
*Frameworks e plataformas focados no desenvolvimento de produtos voltados a
Tecnologia Assistiva***

Mapeamento Sistemático de Literatura apresentado ao Programa de Pós-Graduação em Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Doutor em Ciência da Computação

Orientador: Prof^a. Dr^a. Tatiana Aires Tavares

Pelotas, 2017

SUMÁRIO

1	METODOLOGIA DE PESQUISA	2
1.1	Busca pelo artigo de referência	3
1.2	Planejamento do MSL	4
1.3	Execução do MSL	5
	REFERÊNCIAS	9

1 METODOLOGIA DE PESQUISA

Com o intuito de realizar a revisão bibliográfica proposta, foi adotado o Mapeamento Sistemático de Literatura ou MSL como metodologia de pesquisa bibliográfica (PETERSEN et al., 2008). Uma característica importante da MSL é que todos os procedimentos executados ao longo de sua duração devem ser documentados. Assim, o estudo realizado deve ser reproduzível por outros pesquisadores, de forma que estes consigam alcançar os mesmos resultados encontrados por seus antecessores.

A aplicação do MSL busca eliminar ou minimizar a ausência de trabalhos que poderiam ser considerados importantes dentro levantamento bibliográfico feito. Mesmo que esta situação ocorra, com a formalização de todos os passos realizados, é possível ao menos justificar a ausência de um determinado artigo através da apresentação das etapas seguidas durante o processo.

A utilização desta técnica gera o chamado mapa sistemático de literatura, um conjunto amplo de artigos que traz uma prospecção geral da área abordada pelo projeto de pesquisa, objeto do presente trabalho. As principais etapas previstas por esta metodologia de pesquisa são:

- Definição de questões de pesquisa;
- Elaboração da string de busca;
- Pesquisa por artigos relevantes;
- Seleção de artigos;
- Classificação dos artigos;
- Extração e mapeamento de dados.

No entanto, antes de se aplicar o MSL é necessário a execução de Uma etapa prévia, responsável por auxiliar a geração de todo protocolo previsto na metodologia. A etapa que precede a aplicação do MSL consiste na busca por um artigo que possua proximidade ao projeto que se pretenda desenvolver.

Esta busca é feita por meio de pesquisa exploratória simples em engenhos de busca científicos. Cada trabalho encontrado neste período deve ser analisado de forma criteriosa, para verificar se o mesmo se assemelha ao tema de pesquisa escolhido. O trabalho escolhido nesta etapa é denominado artigo de referência, e funcionará como uma espécie de norteador para aplicação das demais etapas do MSL. Esta etapa é descrita com detalhes na próxima seção.

1.1 Busca pelo artigo de referência

A realização de pesquisas exploratórias em engenhos de busca científicos resultou em um conjunto de artigos que abordam o tema pesquisado. Assim, durante esta etapa buscou-se por trabalhos que tinham como foco principal criar meios que permitissem ou facilitassem o desenvolvimento de aplicações para pessoas com deficiências motoras. Esta pesquisa resultou na seleção de quatro trabalhos, os quais foram analisados de forma mais profunda, e que têm suas principais características resumidas na Tabela 1.

NETO et al. (2012) propõe o VoiceCONET, um framework colaborativo para construção de aplicações especificamente utilizando tecnologias para reconhecimento e síntese de fala. SALNIKOV et al. (2014) também aposta em tecnologias de síntese de fala, descrevendo uma interface natural que suporta entrada de dados baseada em gestos combinada a reconhecimento de voz. Já MIESENBERGER et al. (2014) apresenta o ATLab, um sistema que visa ser simples e flexível suficiente para a produção de aplicações para dispositivos móveis que facilitem o uso de computadores por pessoas com limitações motoras.

Por fim, OSSMANN et al. (2014) descreve o framework AsTeRICS (*Assistive Technology Rapid Integration & Construction Set*), o qual além de englobar as tecnologias e dispositivos utilizados pelos outros projetos apresentados, amplia consideravelmente este universo de possibilidades. Através de uma interface de software, o AsTeRICS ainda permite interligar de forma simples estes dispositivos para criar aplicações que facilitam a execução de afazeres cotidianos (como ler ou comer, por exemplo), automatizam ambientes (permitindo controle luzes, televisores, aparelhos condicionadores de ar, entre outros) e possibilitam a inclusão das pessoas na sociedade por meio do uso de computadores e *smartphones*.

Devido a suas características, este trabalho foi definido como o artigo de referência do projeto de pesquisa que está sendo desenvolvido. Ele será utilizado para completar o protocolo formal do MSL e na definição de palavras-chave e critérios de inclusão/exclusão que serão empregados nas etapas da metodologia. Depois de encontrado o artigo de referência, foram aplicadas as demais etapas previstas no mapeamento, as quais são detalhadas a seguir.

Tabela 1 – Trabalhos encontrados na busca de artigo de referência

Trabalho	Framework	Dispositivos	Tecnologias utilizadas
NETO et al. (2012)	VoiceCONET	Computador pessoal	Reconhecimento de voz (TTS e STT).
(MIESENBERGER et al., 2014)	ATLab	Tablets e smartphones.	Integração com plataformas móveis: Android iOS e Windows.
(SALNIKOV et al., 2014)	STTK	Kinect, mouse	Google Speech API e .Net SpeechRecognizer para Reconhecimento de gestos e voz.
(OSSMANN et al., 2014)	AsTeRICS	Diversos sensores e atuadores disponíveis.	Inúmeras.

1.2 Planejamento do MSL

Esta etapa consiste no preenchimento adequado do protocolo de pesquisa que será seguido no MSL. O primeiro passo neste sentido, é a definição do objetivo central da revisão bibliográfica. Neste trabalho o seguinte objetivo foi adotado:

“Identificar e caracterizar o estado da arte sobre tecnologia assistiva com foco em acessibilidade ao computador para pessoas com deficiência motora.”

A partir da definição de objetivo principal da pesquisa, devem ser desmembradas algumas questões de pesquisa relacionadas ao tema central. O resultado do MSL necessariamente buscará responder algumas questões de pesquisa relacionadas ao projeto. Especificamente este trabalho busca fornecer respostas às seguintes perguntas:

- *Como as pesquisas mostram o desenvolvimento e avaliação de interfaces e softwares para a tecnologia assistiva que ajudem as pessoas com deficiência física no uso do computador?*
- *Existem frameworks ou plataformas de desenvolvimento de software, voltados a tecnologia assistiva? Quais são eles?*
- *O que caracteriza uma plataforma/framework como sendo voltada à tecnologia assistiva?*

Após a definição do tema e das questões de pesquisa o protocolo do MSL prevê a especificação de um conjunto de palavras-chave, devidamente calibrado pelo artigo de referência, para gerar uma string de busca a ser aplicada a indexadores científicos, e assim recuperar uma série de artigos relacionados ao objetivo da pesquisa, que possam responder as questões elaboradas. Abaixo a string de busca gerada baseadas no conjunto de palavras-chave escolhido:

(("human-computer interaction"OR HCI) AND ("assistive technology"OR accessibility) AND (motor OR physical OR impairment OR disability) AND (framework OR platform OR API OR evaluation OR test OR guidelines OR protocols))

Esta string de busca deve ser aplicada sobre indexadores científicos que serão utilizados para coletas dos artigos. Neste mapeamento especificamente, foram adotados os seguintes engenhos de busca científicos:

- *IEEE Xplore Digital Library;*
- *ACM Digital Library;*
- *Springer;*
- *Science Direct.*

Estes engenhos de busca foram selecionados por agregarem uma quantidade considerável de trabalhos dentro da área de pesquisa considerada. Com o intuito de restringir a quantidade de trabalhos recuperados nesta etapa de seleção, alguns critérios de seleção de fonte foram utilizados:

- São aceitos apenas artigos completos. Ou seja, resumos, artigos de até duas páginas ou pôsteres, são excluídos do resultado da busca;
- São considerados apenas artigos publicados a partir de 2012. No entanto, trabalhos anteriores a este ano podem ser incluídos manualmente, se forem considerados relevantes;
- O foco dos trabalhos deve em apresentação de TA voltado em acessibilidade de computador para deficiência motora. Assim, artigos voltados unicamente a acessibilidade de cegos ou surdos, por exemplos, não são classificáveis para a pesquisa.

Estas definições na etapa de planejamento permitiram iniciar a próxima etapa prevista no MSL, a Execução.

1.3 Execução do MSL

Esta etapa do MSL visa identificar um conjunto de trabalhos relevantes que auxiliem a responder as questões de pesquisa definidas previamente. Assim, foram criadas string de busca específicas para cada um dos engenhos de busca escolhidos, uma vez que ele adotam sintaxes diferentes. A aplicação das strings resultou no gráfico em destaque na Figura 1.

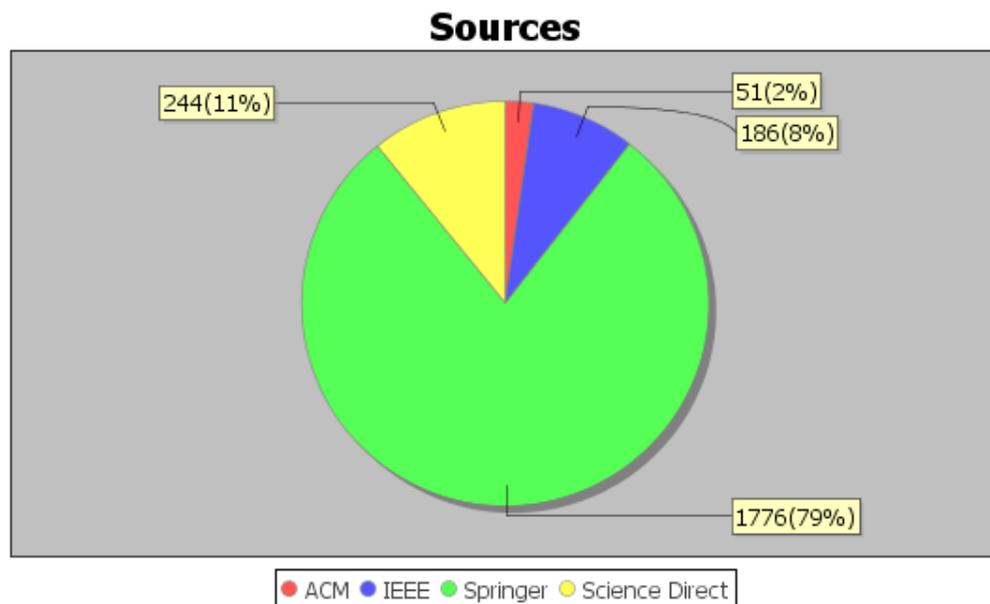


Figura 1 – Total de fontes recuperadas a partir da busca nos indexadores.

A etapa de identificação de fontes a partir de busca por palavras-chave, foi sucedida pela a etapa de pré-seleção de artigos. Nesta fase, foram definidos uma série de critérios de inclusão e exclusão de artigos que foram aplicados em cada uma destes artigos. Para exclusão de artigos foram adotados os seguintes critérios:

1. Artigo publicado em evento exclusivo de área médica;
2. Artigo publicado anteriormente a 2012 (apesar deste critério já ter sido adotado durante a etapa de busca automática, foi constatado que alguns trabalhos anteriores a este período ainda eram recuperados);
3. Resumos, pôsteres ou artigo com duas ou menos páginas (não classificável ou com dados insuficientes);
4. Artigo duplicado;
5. Artigos relacionados exclusivamente a apenas outras deficiências, que não fossem motoras (deficiência visual ou cognitiva, por exemplo);
6. Artigos duplicados (presentes em mais de um buscador);
7. Livros completos;
8. Artigos relacionados exclusivamente a idosos.

Em contrapartida, alguns critérios de inclusão foram criados dois critérios de inclusão para classificar trabalhos, se necessário. Foram eles:

1. Artigo envolvendo TA para acessibilidade ao computador;
2. Artigo publicado anteriormente a 2012, mas considerado relevante.

Devido o número relativamente grande de artigos identificados, este conjunto foi subdividido entre oito pesquisadores vinculados ao grupo de pesquisa WeTech. Baseado nos critérios definidos, cada pesquisador analisou título, palavras-chave, e resumo dos trabalho para realizar a triagem. Como resultado, obteve-se a seleção de 173 artigos enviados a fase seguinte de seleção. A Figura 2 apresenta os percentuais de artigos aceitos, recusados e duplicados durante esta etapa de avaliação.

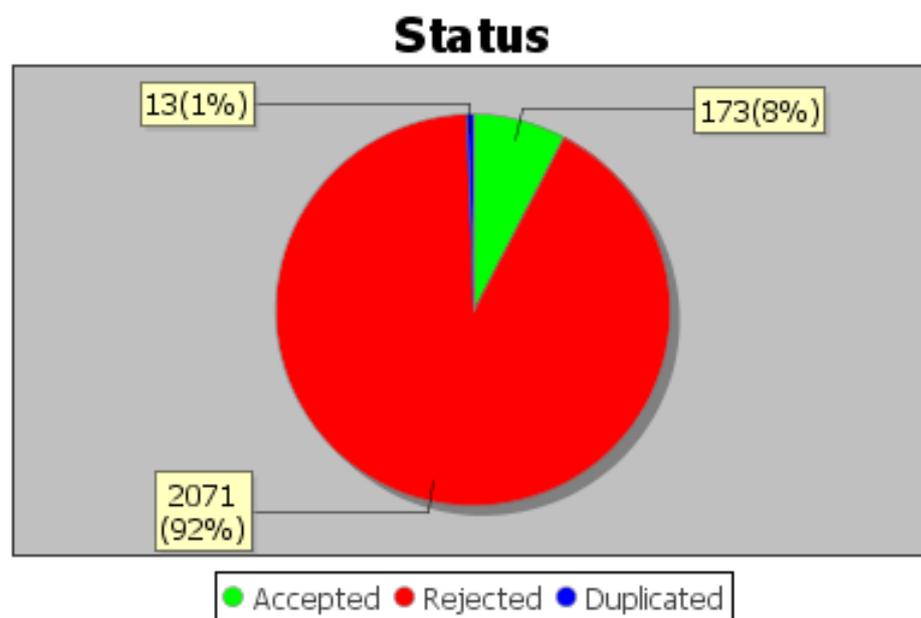


Figura 2 – *Status* das fontes após a primeira rodada de análise.

A partir daí, uma nova rodada de leitura e análise se iniciou, composta por três pesquisadores. Cada trabalho restante foi analisado individualmente por estes três pesquisadores que buscavam caracterizar os trabalhos em alguma das seguintes linhas de pesquisa:

- Artigo relacionado a **interface** assistiva;
- Artigo relacionado a metodologias de **avaliação** sobre sistema, interface ou dispositivo assistivo;
- Artigo relacionado a **frameworks**, APIs ou plataformas voltadas à criação de soluções focadas em TA;
- Artigo não aplicável a nenhum dos casos. Factível de exclusão.

Após a análise individual destes artigos, através da análise de Título, Resumo, Palavras-chave e Conclusão, os resultados individuais foram cruzados durante algumas reuniões mediadas pelo orientador dos trabalhos. Dos 173 trabalhos estudados, 87 foram excluídos nesta fase do protocolo por não revelarem indícios de caracterização direta a nenhum dos focos de pesquisa procurados. Os demais artigos puderam classificados em pelo menos um dos pontos de análise, sendo esta distribuição apresentada na Figura 3.

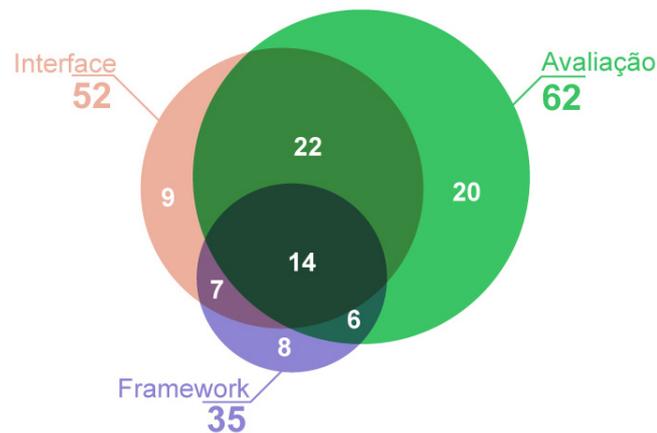


Figura 3 – Distribuição dos artigos nos eixos específicos de pesquisas do MSL desenvolvido.

Dos trabalhos analisados nesta etapa da metodologia, que não foram excluídos, restaram:

- 52 artigos inicialmente relacionados a interface assistiva;
- 62 artigos inicialmente relacionados a metodologias de avaliação de sistema, interface ou dispositivo assistivo;
- **35 artigos inicialmente relacionados a frameworks, APIs ou plataformas de desenvolvimento voltadas à criação de soluções focadas em TA;**

Este trabalho especificamente busca responder questões relativas a frameworks e plataformas destinadas ao desenvolvimento de sistema assistivo. No entanto, só a passagem por este filtro anterior, não necessariamente garante que o artigo deva estar presente na revisão bibliográfica, uma vez que depois de analisado integralmente, ele ainda pode se ser descartado, se for o caso. Os artigos analisados na íntegra, que não foram excluídos por outros critérios são abordados no próximo capítulo.

REFERÊNCIAS

MIESENBERGER, K. et al. ATLab: An App-Framework for Physical Disabilities. , [S.l.], 2014.

NETO, N. et al. **VOICECONET**: A Collaborative Framework for Speech-Based Computer Accessibility with a Case Study for Brazilian Portuguese. [S.l.]: INTECH Open Access Publisher, 2012.

OSSMANN, R. et al. AsTeRICS, a flexible AT construction set. **International Journal of Adaptive Control and Signal Processing**, [S.l.], v.28, n.12, p.1475–1503, 2014.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic Mapping Studies in Software Engineering. In: EASE, 2008. **Anais...** [S.l.: s.n.], 2008. v.8, p.68–77.

SALNIKOV, A. et al. Natural interface to improve human-computer interaction for people with upper limb disabilities: exploring the potentials of voice input and hand gestures in application development to improve the communication possibilities of people with motor disorders. , [S.l.], 2014.

ANEXO B CERTIFICADO DE ESTÁGIO DE PÓS-GRADUAÇÃO NA UNIVERSIDADE DE AVEIRO, EM PORTUGAL



CERTIDÃO

Mário Luís Dias Forte Pelaio, Diretor dos Serviços de Gestão Académica da Universidade de Aveiro certifica que o Doutor **Rafael Cunha Cardoso**, filho de Vanderlei Leite Cardoso e de Tânia Maria da Cunha Cardoso, natural do Brasil, efetuou um **Estágio de Pós-Graduação** em Informação e Comunicação em Plataformas Digitais, no Departamento de Comunicação e Arte, sob o tema "*Utilização do dispositivo de interação IOM no contexto de aplicações de TV interativa*", sob a orientação do Professor Doutor Telmo Eduardo Miranda Castelão da Silva, Professor Auxiliar da Universidade de Aveiro, pelo período de 5 meses e 15 dias, entre 2 de abril e 16 de setembro de 2018.

Universidade de Aveiro, em 04 de outubro de 2018

O Director dos Serviços de Gestão Académica,

(Mário Luís Pelaio)

CONFERIDO

Emol. 5,00€

Imp. €

Total 5,00€

Assinado