



UNIVERSIDADE
Estadual de LONDRINA

MURILO CRIVELLARI CAMARGO

**FRAMEWORK PARA DESENVOLVIMENTO ÁGIL DE
SOFTWARE EDUCACIONAL VOLTADO AO PORTADOR DE
TRANSTORNO DO ESPECTRO AUTISTA**

LONDRINA
2019

MURILO CRIVELLARI CAMARGO

**FRAMEWORK PARA DESENVOLVIMENTO ÁGIL DE
SOFTWARE EDUCACIONAL VOLTADO AO PORTADOR DE
TRANSTORNO DO ESPECTRO AUTISTA**

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação do Departamento de Computação da Universidade Estadual de Londrina, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Rodolfo Miranda de Barros

LONDRINA
2019

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Camargo, Murilo Crivellari.

Framework para desenvolvimento ágil de software educacional voltado ao portador de transtorno do espectro autista / Murilo Crivellari Camargo. - Londrina, 2019.
152 f. : il.

Orientador: Rodolfo Miranda de Barros.

Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2019.

Inclui bibliografia.

1. Desenvolvimento ágil de software - Tese. 2. Transtorno do espectro autista - Tese. 3. Software educacional - Tese. 4. Projeto de acessibilidade - Tese. I. Barros, Rodolfo Miranda de. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

MURILO CRIVELLARI CAMARGO

**FRAMEWORK PARA DESENVOLVIMENTO ÁGIL DE
SOFTWARE EDUCACIONAL VOLTADO AO PORTADOR DE
TRANSTORNO DO ESPECTRO AUTISTA**

Dissertação apresentada ao Programa de Mestrado
em Ciência da Computação do Departamento de
Computação da Universidade Estadual de
Londrina, como requisito parcial para a obtenção
do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Prof. Dr. Rodolfo Miranda de Barros
Universidade Estadual de Londrina

Profa. Dra. Cristiane Affonso de Almeida Zerbetto
Universidade Estadual de Londrina

Prof. Dr. Vitor Valério de Souza Campos
Universidade Estadual de Londrina

Londrina-PR, ____ de _____ de 2019.

CAMARGO, M. C. **Framework para desenvolvimento ágil de software educacional voltado ao portador de Transtorno do Espectro Autista**. 152p. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina-PR, 2019.

RESUMO

O Transtorno do Espectro Autista (TEA) é um distúrbio de etiologia desconhecida, que envolve limitações de ordem social, comunicativa e comportamental. Estudos recentes sugerem um aumento no número de portadores do transtorno, especialmente ocasionado pela adoção de critérios de diagnóstico mais abrangentes e flexíveis. Com isso, nota-se também um aumento considerável de softwares educacionais voltados para este público, incluindo o uso de tecnologias como Realidade Virtual, Kinect, e até mesmo terapia musical. Apesar deste cenário, o desenvolvimento de software para indivíduos autistas é complexo. Por um lado, as diretrizes de acessibilidade digital não são específicas para o autismo e, por outro, existe uma carência de processos de desenvolvimento que contemplem a usabilidade do público TEA com plenitude. Desta forma, este trabalho propõe um *framework* para desenvolvimento ágil de softwares educacionais voltados ao portador de Transtorno do Espectro Autista. Inicialmente, realizou-se uma revisão sistemática de fontes primárias cujo objeto de estudo fosse o desenvolvimento de software educacional acessível ao público TEA. Em seguida, o *framework* foi modelado e descrito detalhadamente, incluindo a arquitetura do processo, artefatos gerados, agentes envolvidos, entradas e saídas e documentação. Por fim, apresenta-se uma investigação da integração do *framework* com o processo de desenvolvimento ágil adotado por um laboratório de Tecnologia da Informação e Comunicação (TIC), bem como a aplicação de uma avaliação qualitativa com especialistas. Os resultados obtidos sugerem que o *framework* pode ser integrado com sucesso aos modelos ágeis vigentes, auxiliando o desenvolvimento de softwares acessíveis que de fato contemplem as necessidades e preferências do indivíduo autista.

Palavras-chave: Desenvolvimento ágil. Software educacional. Acessibilidade. Transtorno do Espectro Autista.

CAMARGO, M. C. **Framework for agile development of educational software targeted to Autism Spectrum Disorder.** 152p. Dissertation (Master in Computer Science) – State University of Londrina, Londrina-PR, 2019.

ABSTRACT

Autism Spectrum Disorder (ASD) is a disturbance that involves social, behavioral, and communication deficits. Autism etiology is unclear, and recent studies suggest that the number of people diagnosed with ASD increased significantly, due to the adoption of broader and more flexible diagnosis criteria. Also, the number of autism-targeted educational software has escalated proportionally, using alternative approaches such as Virtual Reality, Kinect, and even music therapy. Despite that, the development of autism-specific software is complex. On the one hand, digital accessibility guidelines are not designed specifically for ASD and, on the other, there is a lack of development processes oriented towards autism usability. Thus, this study proposes a framework for agile development of educational software targeted to Autism Spectrum Disorder. Firstly, a systematic review was conducted with primary sources on the development of autism-specific educational software. Then, the framework was modeled and described thoroughly, including the process architecture, artifacts, inputs and outputs, roles and responsibilities, and documentation. Finally, an investigation is presented on the integration of the framework with the agile development process used by an Information and Communication Technology laboratory. In addition, a qualitative questionnaire was applied with experts from the fields of computer sciences and design. Results suggest that the framework can be successfully integrated to current agile methods, thus supporting the development of accessible software to meet the needs and preferences of autistic individuals.

Keywords: Agile development. Educational software. Accessibility. Autism Spectrum Disorder.

LISTA DE ILUSTRAÇÕES

Figura 1 – Etapas ágeis comuns	45
Figura 2 – Objetivos de aprendizagem extraídos dos artigos	59
Figura 3 – Públicos-alvo extraídos dos artigos.....	60
Figura 4 – Ciclo de vida de desenvolvimento do framework.....	70
Figura 5 – Processo de desenvolvimento de software do laboratório Gaia.....	97
Figura 6 – Etapa de análise e planejamento, PDS Gaia.....	99
Figura 7 – Etapa de execução e implementação, PDS Gaia.....	101
Figura 8 – Etapa de validação e testes, PDS Gaia	101
Figura 9 – Etapa de entrega, PDS Gaia	102

LISTA DE TABELAS

Tabela 1 – Objetivos de aprendizagem e PDS	57
Tabela 2 – Levantamento de requisitos e técnicas de aprendizagem	60
Tabela 3 – Design da Interface do Usuário, testes e resultados	63
Tabela 4 – Mecanismos e diretrizes do framework.....	85
Tabela 5 – Melhores práticas para acessibilidade TEA.....	87
Tabela 6 – Entradas e saídas do framework	89
Tabela 7 – Verificação do framework por especialistas.....	107

LISTA DE ABREVIATURAS E SIGLAS

ABC	Autism Behavior Checklist
ADDIE	Analysis, Design, Development, Implementation, Evaluation
ADDM	Autism and Developmental Disabilities Monitoring
ASDS	Asperger's Syndrome Diagnostic Scale
CARS	Childhood Autism Rating Scale
CAST	Childhood Autism Spectrum Test
CDC	Center for Disease Control and Prevention
CHAT	Checklist for Autism in Toddlers
CID	Classificação Internacional de Doenças
CMM	Capability Maturity Model
DCU	Design Centrado no Usuário
DSA	Desenvolvimento de Software Adaptativo
DSM	Diagnostic and Statistical Manual of Mental Disorders
IEC	International Electrotechnical Commission
ISO	International Organization of Standardization
IGU	Interface Gráfica do Usuário
IU	Interface do Usuário
LA	Lições Aprendidas
MDMD	Metodologia de Desenvolvimento de Modelos Dinâmicos
MM	Modelo de Maturidade
MMU	Modelo de Maturidade de Usabilidade
ONU	Organização das Nações Unidas
PECS	Picture Exchange Communication System
PDS	Processo de Desenvolvimento de Software
RPG	Role-Playing Game
RUP	Rational Unified Process
RV	Realidade Virtual
STAT	Screening Tool for Autism in Toddlers
TEA	Transtorno do Espectro Autista

TEACCH	Treatment and Education of Autistic and related Communication-handicapped Children
TGD	Transtorno Global do Desenvolvimento
TIC	Tecnologia da Informação e Comunicação
TID	Transtorno Invasivo do Desenvolvimento
TID-SOE	Transtorno Invasivo do Desenvolvimento Sem Outra Especificação
UML	Unified Modeling Language
UX	Experiência do Usuário (do inglês, <i>User eXperience</i>)
W3C	World Wide Web Consortium
WCAG	Web Content Accessibility Guidelines
XP	eXtreme Programming

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Problemática de Pesquisa	16
1.2	Escopo	17
1.3	Objetivo Geral	17
1.4	Objetivos Específicos	17
1.5	Justificativa	18
1.6	Organização do Trabalho	18
1.7	Metodologia de Pesquisa.....	19
2	TRANSTORNO DO ESPECTRO AUTISTA.....	21
2.1	Terminologia e Evolução do Olhar Clínico	21
2.2	Epidemiologia e Etiologia.....	23
2.3	Diagnóstico e Ferramentas de Triagem.....	25
2.4	Prognóstico e Tratamento.....	29
3	ACESSIBILIDADE E INCLUSÃO	32
3.1	Comportamento no Ambiente Virtual.....	32
3.2	Softwares Educacionais.....	34
3.3	Acessibilidade Digital	37
4	DESENVOLVIMENTO DE SOFTWARE ACESSÍVEL.....	42
4.1	Modelos de Desenvolvimento	42
4.2	Design Centrado no Usuário	48
4.3	Gestão da Qualidade.....	50
5	REVISÃO SISTEMÁTICA	53
5.1	Protocolo de Pesquisa	53
5.2	Condução da Busca	55
5.3	Resultados e Discussão	57

5.4	Considerações Parciais	65
6	MODELAGEM DO FRAMEWORK	66
6.1	Definição da Estratégia	66
6.2	Organização do Processo	67
6.3	Planejamento de Produção	68
6.4	Arquitetura do Processo	69
6.4.1	Levantamento de Requisitos	70
6.4.2	Planejamento	75
6.4.3	Arquitetura de Software	76
6.4.4	Desenvolvimento	78
6.4.5	Testes Unitários	81
6.4.6	Implantação	82
6.4.7	Validação do Sistema	82
6.4.8	Entrega	83
7	ESPECIFICAÇÕES PROCESSUAIS	84
7.1	Documentação	84
7.2	Configuração de Mecanismos e Diretrizes	85
7.3	Mapa de Artefatos	88
7.4	Entradas e Saídas	89
7.5	Meios de Comunicação	91
7.6	Identificação de Problemas	91
7.7	Melhorias	94
7.8	Gestão do Conhecimento	95
8	INTEGRAÇÃO DO FRAMEWORK	96
8.1	Apresentação da Empresa	96
8.2	Integração do Framework com o PDS Gaia	98
8.3	Verificação Qualitativa	103

9	CONSIDERAÇÕES FINAIS	110
9.1	Objetivos Alcançados.....	110
9.2	Limitações e Desafios	111
9.3	Perspectivas Futuras.....	112
	REFERÊNCIAS	113
	APÊNDICE A – RELATÓRIO GERAL DE INTERVENÇÃO	122
	APÊNDICE B – RELATÓRIO DE UX E IU	124
	APÊNDICE C – QUESTIONÁRIO PÓS-ENTREGA	127
	APÊNDICE D – REGISTRO DE LIÇÕES APRENDIDAS	129
	ANEXO A – CHILDHOOD AUTISM RATING SCALE (CARS)	131
	ANEXO B – LEARNING STYLE INVENTORY	139
	ANEXO C – HEURÍSTICAS DE ACESSIBILIDADE TEA	145
	ANEXO D – MODELO DE MATURIDADE AUCDI	147
	TRABALHOS PUBLICADOS PELO AUTOR	152

1 INTRODUÇÃO

Desde que foi descrito pela primeira vez pelo austríaco Leo Kanner, em 1943, o autismo vem ganhando cada vez mais atenção por parte da comunidade acadêmica. Segundo o Manual Diagnóstico e Estatístico de Transtornos Mentais, quinta edição (DSM-5), disponibilizado pela Associação Americana de Psiquiatria [5], o Transtorno do Espectro Autista (TEA) é caracterizado pela dificuldade de iniciar ou manter interações sociais, deficiências na comunicação verbal e não-verbal, e padrões comportamentais restritivos ou repetitivos.

O entendimento acerca do transtorno tornou-se, a partir do DSM-5, mais abrangente, o que flexibilizou o processo de diagnóstico de novos pacientes. De fato, indicadores recentes, como os apresentados pelo *Center for Disease Control and Prevention* (CDC), órgão de saúde pública dos Estados Unidos, apontam um crescimento exponencial de indivíduos diagnosticados no espectro. O relatório do *Autism and Developmental Disabilities Monitoring Network* (ADDM), conexo ao referido órgão, indica a incidência de TEA em 1 a cada 59 crianças estadunidenses na faixa dos 8 anos de idade [7].

Estes números enfatizam a importância e a urgência de se compreender com mais profundidade o transtorno e suas implicações clínicas, sintomas, graus de severidade e prognósticos. Ademais, explicitam a necessidade de se discutir a acessibilidade. Afinal, a despeito do desconhecimento científico que ainda permeia o transtorno, os indivíduos TEA encontram-se efetivamente inseridos na sociedade, e demandam soluções que estimulem sua autonomia. Esta premissa é especialmente relevante nos ambientes digitais, em função das preferências deste público.

Nos estudos conduzidos por Kuo et al [45] e Mazurek e Wenstrup [54], onde foram realizadas pesquisas com crianças e adolescentes TEA para analisar como se relacionavam com mídias digitais, os resultados apontaram prevalência na utilização de mídias com apelo visual, como televisões, vídeo games e computadores. No caso dos últimos, o tempo médio de uso chega a 4,9 horas por dia entre adolescentes TEA de 12 a 18 anos (meninos, predominantemente). Este fato acentua, por um lado, o interesse de que tais mídias possam ser instrumento de intervenção ao transtorno e, por outro, a necessidade de que sejam ambientes acessíveis para este público.

Esforços têm sido feitos no sentido de tornar a acessibilidade no universo digital uma realidade viável e efetiva. A *World Wide Web Consortium* (W3C), por exemplo, foi responsável pelo desenvolvimento de um dos mais importantes e conhecidos documentos relativos à acessibilidade web – o *Web Content Accessibility Guidelines* (WCAG) 2.0 [78]. Trata-se de um conjunto de diretrizes que, quando aplicadas às plataformas digitais (softwares, sites, aplicativos, entre outros), garantem a acessibilidade dos conteúdos disponíveis. Este documento foi incorporado pela *International Organization for Standardization*, e compilado na norma ISO/IEC 40500:2012.

Entretanto, as recomendações presentes na WCAG 2.0 não são específicas ao público autista. Ao contrário, estas diretrizes têm como objetivo abranger a acessibilidade de forma global, incluindo as deficiências auditivas, visuais e motoras, por exemplo. Isto reflete uma prática que tem sido estimulada por diversas organizações – como a própria W3C – chamada acessibilidade universal (ou design universal). Nesta abordagem, busca-se tornar os recursos e ambientes acessíveis a todas as pessoas, independentemente das limitações que possuam.

Embora essa abordagem seja inegavelmente importante para a conscientização e a viabilização da acessibilidade, ela pode não ser suficiente para contemplar os indivíduos TEA em sua plenitude. Isso porque este público apresenta uma complexidade clínica que muitas vezes não pode ser solucionada objetivamente. Uma solução bastante óbvia para tornar recursos digitais acessíveis a pessoas cegas, por exemplo, seria disponibilizar o mesmo conteúdo em formato de áudio. Similarmente, se o objetivo é superar as deficiências auditivas, deve-se incluir legendas em vídeos ou outros conteúdos sonoros.

O que fazer, contudo, para superar as dificuldades de um público com variações de sintomas e graus de severidade tão amplas? Como propor uma solução única a um grupo cujas capacidades individuais são tão específicas? Esta é, precisamente, a realidade do indivíduo TEA – trata-se de um público com necessidades muito particulares e heterogêneas. Não por outro motivo, recomendações de acessibilidade específicas ao autismo são escassas. Por certo, a complexidade de se trabalhar a acessibilidade para os indivíduos TEA é um obstáculo a ser transposto na busca por conferir-lhes mais independência e qualidade de vida.

Na verdade, o número de softwares educacionais desenvolvidos com foco no público TEA tem aumentado consideravelmente, como se pode comprovar nos estudos de Fletcher-Watson [23] e Ramdoss et al [62]. Os autores reuniram relatos presentes na literatura

acerca do desenvolvimento de software com finalidade pedagógica voltados aos indivíduos no espectro, revelando um crescente interesse da comunidade científica em propor ferramentas de apoio ao ensino, mesmo em face da escassez de recursos metodológicos.

O desenvolvimento de softwares acessíveis é uma tarefa complexa. Por um lado, requer conhecimento técnico que muitas vezes não está disponível às equipes de desenvolvimento. Por outro, a acessibilidade tende a ser encarada como um ônus desnecessário ao processo de produção, e não como um aspecto que agrega valor ao produto de software. No caso do autismo, este quadro agrava-se em função do próprio desconhecimento acerca do transtorno. A falta de conscientização, somada às limitações teóricas e tecnológicas, apresentam-se como empecilhos a serem superados rumo à acessibilidade para o público autista.

1.1 Problemática de Pesquisa

Como mencionado anteriormente, o desenvolvimento de softwares acessíveis ao público TEA tornou-se objeto de crescente interesse. Apesar disso, os mais importantes e conhecidos documentos sobre acessibilidade, como o WCAG 2.0, caminham na direção da acessibilidade universal, sem maior enfoque às deficiências cognitivas ou ao autismo.

Evidentemente, a acessibilidade universal é uma abordagem valiosa quando se pretende desenvolver recursos para todos os públicos. Entretanto, no caso dos softwares educacionais cuja finalidade principal seja intervir no processo de aprendizagem do sujeito autista, esta linha de raciocínio não parece ser a mais adequada, pois não considera suas necessidades individuais.

A complexidade clínica do transtorno e a escassez de recursos metodológicos específicos podem, portanto, fragilizar o processo de desenvolvimento. Por um lado, o quadro clínico do autismo apresenta variações que impossibilitam a adoção de uma estratégia universal. Por outro, a inexistência de uma padronização metodológica de desenvolvimento põe em cheque a qualidade do processo.

1.2 Escopo

Em face do contexto supracitado, esta pesquisa questiona como é possível facilitar o desenvolvimento de softwares educacionais para o indivíduo TEA. Trabalha-se com a hipótese de que, ao conferir maior rigor processual e maior grau de formalidade às metodologias empregadas, obter-se-á mais dinamismo no processo de desenvolvimento e maior qualidade no produto de software. A pesquisa está delineada com o seguinte escopo:

- Público-alvo: equipes de desenvolvimento lidando com indivíduos autistas;
- Produto: desenvolvimento de *softwares educacionais*, ou seja, softwares voltados a intervir no processo de ensino-aprendizagem do público TEA;
- Processo: sustentado pelos *métodos ágeis* de desenvolvimento; e
- Abordagem metodológica: design centrado no usuário.

1.3 Objetivo Geral

De acordo com a problemática e o escopo apresentados, esta pesquisa tem como objetivo geral a elaboração de um *framework* para desenvolvimento de softwares voltados ao público TEA. Este *framework* pretende ser um facilitador no processo de desenvolvimento, a fim de conferir maior rigor processual durante o percurso.

1.4 Objetivos Específicos

- Identificar eventuais fraquezas e possibilidades de melhoria na condução do processo de desenvolvimento de software (PDS) educacional ao público TEA;
- Estabelecer a modelagem do *framework*, tanto em nível macro (definição das etapas de desenvolvimento e fluxo de trabalho) quanto micro (artefatos, entradas e saídas, documentação, agentes envolvidos); e
- Realizar verificação preliminar da integração do *framework* com modelos ágeis existentes e coletar opiniões de especialistas.

1.5 Justificativa

O *framework* proposto neste trabalho pretende ser uma abordagem inovadora e padronizada ao desenvolvimento de softwares educacionais para portadores de TEA, sem desconsiderar as estratégias já empregadas e o conhecimento empírico proveniente da *práxis* de desenvolvimento. Neste sentido, espera-se contribuir para o aumento do rigor processual neste contexto de produção. Acredita-se que a formalização dos procedimentos metodológicos seja de grande ajuda para melhorar aspectos de acessibilidade e usabilidade e, conseqüentemente, aumentar a qualidade final do produto de software.

Ademais, espera-se contribuir com a propagação de recursos de acessibilidade para a comunidade científica e as equipes de desenvolvimento que desejem trabalhar com o público TEA. Os conteúdos aqui apresentados (práticas, métodos, ferramentas, etc.) podem servir tanto para fomentar a discussão acerca da acessibilidade de modo geral, quanto para guiar o processo de desenvolvimento focado no sujeito autista. A literatura acerca do tema, embora em inegável expansão, ainda carece de esforços que se orientem neste sentido.

Por fim, há que se considerar os impactos na qualidade de vida do público TEA. Se o *framework* proposto cumprir satisfatoriamente seu intuito de inspirar profissionais e sustentar o trabalho das equipes de desenvolvimento, haverá ganhos significativos em termos de pesquisa e nos produtos desenvolvidos, o que se reflete diretamente na percepção do usuário final. Estima-se que os indivíduos TEA possam se beneficiar de softwares educacionais cada vez mais adequados e eficientes, contribuindo para sua autonomia e conseqüente inclusão social.

1.6 Organização do Trabalho

Este trabalho organiza-se em mais oito capítulos. Os capítulos 2, 3 e 4 correspondem à revisão de literatura, nos quais encontra-se o referencial teórico que serve de arcabouço para o trabalho. No capítulo 2, descreve-se o Transtorno do Espectro Autista sob um viés clínico, com a finalidade de identificar o distúrbio e facilitar seu entendimento. São abordados aspectos referentes à sua evolução terminológica, epidemiologia e etiologia. Além disso, apresenta-se o complexo processo de diagnóstico – que deve ser suportado por técnicas de vigilância e ferramentas de triagem – e as perspectivas de tratamento dos pacientes.

Já no capítulo 3, trata-se da acessibilidade e inclusão do indivíduo autista, sobretudo no contexto digital. Abordam-se as preferências e hábitos deste público ao utilizar mídias eletrônicas, e quais as oportunidades e os entraves advindos destes comportamentos. Também se faz um breve apanhado dos softwares educacionais para o TEA, que têm despertado recente interesse na comunidade acadêmica e corporativa. Por fim, fala-se dos recursos de acessibilidade no ambiente digital, comparando-se as ferramentas de acessibilidade geral daquelas específicas ao autismo.

No capítulo 4, adentra-se no desenvolvimento de software, com foco na interação com o usuário. Primeiro, destacam-se os modelos de desenvolvimento de software existentes, abrangendo-se tanto os modelos tradicionais quanto os ágeis. Em seguida, fala-se do Design Centrado no Usuário, uma abordagem que, quando aliada ao desenvolvimento ágil, pode melhorar significativamente os aspectos de usabilidade. Finalmente, são elencadas as contribuições da gestão de qualidade, com a descrição de ferramentas como os Modelos de Maturidade e as Lições Aprendidas.

O capítulo 5 discorre sobre a revisão de literatura realizada, seu protocolo e condução, e discute os resultados obtidos. No capítulo 6, modelagem do *framework*, apresentam-se as etapas da arquitetura do processo, sendo que as especificações processuais referentes aos artefatos gerados, entradas e saídas, mecanismos e diretrizes e demais detalhamentos ocorrem no capítulo 7. O Capítulo 8, por sua vez, demonstra como ocorreria a integração do framework a um modelo ágil adotado por uma organização e o processo de verificação qualitativa com especialistas. Por fim, o capítulo 9 encerra este estudo relatando as conclusões extraídas com a sua realização e apontando perspectivas de trabalhos futuros.

1.7 Metodologia de Pesquisa

A metodologia de pesquisa aplicada neste trabalho tem natureza exploratória e descritiva, com relação aos seus objetivos. Por um lado, seu caráter exploratório visa a investigar com maior profundidade o objeto de estudo, a fim de esclarecer a problemática de pesquisa. Por outro, é também descritiva no sentido que busca explicitar as inter-relações inerentes ao fenômeno de pesquisa por meio da coleta padronizada de dados [27]. Com relação aos procedimentos técnicos empregados, adotam-se a pesquisa bibliográfica (em fontes primárias e

secundárias revisadas por pares), a revisão sistemática e a verificação do *framework*, com abordagem qualitativa.

Após a pesquisa bibliográfica, realiza-se uma revisão sistemática com base nos procedimentos indicados por Kitchenham [40]. A revisão sistemática inicia-se com o delineamento do protocolo de pesquisa, em que se encontram as questões de pesquisa, os critérios de inclusão e exclusão, e os termos de busca. Em seguida, passa-se à condução da busca, realizada em 6 bases de dados de fontes primárias. Os resultados da busca foram compilados em relatórios individuais de avaliação, sendo que, ao final do processo, 30 artigos foram considerados elegíveis. Por fim, ocorreu a extração e análise dos dados, permitindo uma comparação qualitativa dos estudos.

Na sequência, inicia-se o processo de modelagem do *framework* propriamente dito. Esta modelagem ocorre levando-se em consideração os dados coletados com a revisão sistemática, e é guiada pelo meta-processo de desenvolvimento proposto por Bolanos et al [11]. O meta-processo envolve 12 fases que devem ser seguidas a fim de que o processo seja completo, incluindo-se descrições detalhadas de suas atividades e tarefas, documentação, mecanismos e diretrizes, artefatos, agentes, entradas e saídas, entre outros. O *framework* também incorpora materiais adicionais para garantir alta performance em termos de planejamento, avaliação, e gestão da qualidade e do conhecimento.

Finalmente, apresenta-se a integração do *framework* com um modelo real de desenvolvimento. Esta investigação é conduzida relacionando-se o *framework* com o processo de desenvolvimento adotado pelo Laboratório Gaia, uma organização pública de Tecnologia da Informação e Comunicação vinculada à Universidade Estadual de Londrina. Por fim, realiza-se uma verificação preliminar do *framework*, com base nos procedimentos descritos por Horita [30] e Rautenberg [63]. Nesta etapa, foram envolvidos especialistas das áreas da ciência da computação, design e educação, e coletadas informações qualitativas acerca do funcionamento do processo, que são um indicativo de sua validade.

2 TRANSTORNO DO ESPECTRO AUTISTA

A compreensão acerca do Transtorno do Espectro Autista alterou-se significativamente desde que foi descoberto. O entendimento clínico evoluiu, modificando não apenas a compreensão de sua complexidade, como também flexibilizando seu diagnóstico. Ainda, pesquisadores e profissionais da área clínica têm se esforçado para fornecer esclarecimentos globais relativos aos sintomas, índices epidemiológicos e alternativas de tratamento. Nos próximos tópicos, realiza-se uma sucinta explicação desses aspectos, começando com a evolução terminológica, e passando-se para a epidemiologia e etiologia, diagnóstico e prognóstico.

2.1 Terminologia e Evolução do Olhar Clínico

O termo autismo, embora não tenha sido cunhado por ele, popularizou-se com os estudos do psiquiatra austríaco Leo Kanner em 1943, ocasião em que foram apresentados onze casos de “crianças com condições marcadamente diferentes de tudo relatado até então”. Os casos estudados por Kanner apresentavam sintomas particulares – tais quais “autismo extremo, obsessão, estereotipia, e ecolalia” – que, segundo o autor, eram substancialmente diferentes dos encontrados em outros distúrbios, caracterizando, portanto, uma condição única, desconhecida à época [38].

Nesse sentido, o discurso de Kanner foi importantíssimo para distanciar o autismo de outros distúrbios, como a esquizofrenia infantil. O autor aponta que, apesar das similaridades entre ambas as condições, o autismo possui características exclusivas, como, por exemplo, o relacionamento incomum das crianças autistas com o seu entorno: ao mesmo tempo em que são capazes de utilizar objetos (que não interfiram em sua introspecção) de maneira propositada e inteligente, mostram-se ansiosas e tensas quando precisam interagir com outras pessoas. Kanner [38] explica que o indivíduo esquizofrênico, por conta de sua condição, abandona a realidade da qual faz parte e imerge em seu próprio universo. O autista, por outro lado, não pode se conectar à realidade porque se sente um completo estranho nela desde o início, tendendo a isolar-se em si mesmo.

Além disso, Kanner destaca ainda a preferência das crianças avaliadas pela solidão e o apreço pela organização sequencial, bem como o desenvolvimento gradual de vínculos com pessoas mais próximas, e de comunicação – dando-se, contudo, de forma limitada e apenas com o propósito de satisfazer suas necessidades. Desta maneira, Kanner revolucionou a psiquiatria infantil ao propiciar uma observação clínica sobre um transtorno desconhecido até então, bem como ao diferenciá-lo da esquizofrenia infantil, já que muitas crianças com quadros autísticos eram diagnosticadas equivocadamente como esquizofrênicas.

A partir deste marco, o interesse pelo autismo cresceu e encontrou aporte em outros trabalhos. Em 1944, o pediatra vienense Hans Asperger publicou um estudo envolvendo crianças com características muito similares às aquelas apresentadas por Kanner. A pesquisa permaneceu difícil de acessar por aqueles não-falantes da língua alemã, até finalmente ser traduzida para o inglês por Wing, em 1981. Ao fazê-lo, Wing afirmou que o quadro descrito por Asperger não era uma síndrome diferente daquela analisada por Kanner, mas uma variação dentro do mesmo espectro do transtorno [74].

Outro avanço significativo nos estudos envolvendo o autismo parte de Rutter [64]. O autor avalia a pesquisa conduzida por Kanner e sugere que a falta de critérios específicos e pré-estabelecidos quanto aos limites da síndrome do autismo (termo ao qual, aliás, Rutter se opõe em seus estudos) era o que levava a comunidade médica ao diagnóstico equivocado – confundindo autismo com esquizofrenia e psicose infantil, por exemplo.

Por isso, Rutter propõe uma definição baseada em quatro critérios para o diagnóstico do autismo. Segundo o autor, o indivíduo deveria apresentar: (1) dificuldade de socialização, (2) problemas de comunicação (como inversão pronominal e ecolalia), (3) comportamentos estereotipados e/ou maneirismos, e (4) início antes dos 30 meses de idade [64]. Cabe ressaltar que, segundo a proposição de Rutter, tais problemas não se dão, necessariamente, em função de possível deficiência intelectual.

Com esta nova definição, e alavancado pelo crescente interesse científico sobre o tópico, o autismo foi incluído, em 1980, no DSM-III (*Diagnostic and Statistical Manual of Mental Disorders – Third Edition*), da Associação Americana de Psiquiatria, em uma nova classe de transtornos: os Transtornos Invasivos do Desenvolvimento (TIDs). Eles também foram incluídos na Classificação Estatística Internacional de Doenças e Problemas Relacionados à Saúde (CID), possibilitando um maior consenso por parte da comunidade médica ao lidar com o

autismo. Posteriormente, os estes transtornos foram redefinidos como TGDs (Transtornos Globais do Desenvolvimento), ocasionando certo conflito entre a utilização de ambas as terminologias.

Para dar conta dos diversos níveis de severidade da síndrome, até o DSM-IV, de 1994, os TIDs eram categorizados individualmente, incluindo: (1) Transtorno autista, (2) síndrome de Asperger, (3) síndrome de Rett, (4) transtorno desintegrativo da infância, e (5) transtorno invasivo do desenvolvimento sem outra especificação (TID-SOE) [4]. O mesmo ocorreu no CID-10, em que se pode encontrar oito entradas para descrever o autismo, incluídos sob o código F84 [57].

Tais instruções valeram até o lançamento do DSM-V, quando foram feitas mudanças significativas. Em termos de sintomas, caracteriza o autismo em uma tríade de deficiências: social, verbal e comportamental. Além disso, o termo Transtorno Invasivo do Desenvolvimento foi, afinal, substituído por Transtorno do Espectro Autista (TEA), o que representa “um modo novo, mais assertivo, e mais útil clinicamente e cientificamente de diagnosticar indivíduos com distúrbios relativos ao autismo” [5, tradução nossa]. Conforme o DSM-5, as mudanças visam a unificar o diagnóstico do TEA sem limitar a sensibilidade dos critérios adotados.

O transtorno passou a ser visto, portanto, como um “guarda-chuva” sob o qual estão os pacientes, independentemente das variações de sintomas e graus de severidade que possuam. A Organização Mundial da Saúde mantém suas recomendações com base no CID-10, mas já se fala em mudança: a décima primeira edição do documento, com lançamento previsto para 2022, tende a seguir a orientação presente no DSM-V, e condensar o entendimento do transtorno sob um código único (6A02).

2.2 Epidemiologia e Etiologia

A epidemiologia do Transtorno do Espectro Autista segue difícil de mapear com exatidão. Embora a literatura existente apresente pesquisas com índices numéricos, é difícil estimar um indicador definitivo, em função da variação de resultados. Assumpção e Pimentel [6], por exemplo, indicam uma incidência de 1 a 5 casos em cada 10.000 crianças, sendo 2 a 3 meninos para cada menina. Já Klin [41], aponta dados extraídos de estudos mais recentes que

sugerem maior incidência: 1 indivíduo para cada 1.000 com transtorno autista, e 4 indivíduos com diversas síndromes conexas, como Asperger e TID-SOE.

Outro relatório, de 2014, apresentado pelo *Center for Disease Control and Prevention (CDC)*, órgão de saúde dos Estados Unidos, aponta a incidência de TEA em 1 a cada 68 crianças estadunidenses na faixa dos 8 anos de idade. Isso significa 14,7 indivíduos com TEA a cada 1.000 nascimentos, apenas considerando-se crianças com 8 anos, uma taxa 120% maior do que a estimada entre os anos 2000 e 2002 [7]. Desta maneira, nota-se que o aumento dos índices ao longo do tempo acompanham o processo de flexibilização do diagnóstico apontado pelo DSM-V.

O relatório do CDC ainda revela outros dados interessantes: que a incidência de TEA em meninos é cinco vezes maior que em meninas (1 em 42 meninos e 1 em 189 meninas) e que quase metade das crianças diagnosticadas com TEA (46%) apresentam habilidades intelectuais médias ou acima da média (QI superior a 85). É importante ressaltar, contudo, que o estudo apresenta dados relativos apenas aos Estados Unidos, e que não reflete, necessariamente, a realidade de outros países. Ademais, diversas nações (como na América Latina, o Brasil) não possuem indicadores oficiais sobre o TEA, tornando o mapeamento epidemiológico ainda mais difícil.

Klin [41] explica que o aumento dos números está relacionado ao estímulo do diagnóstico correto de indivíduos TEA, de modo que o que tem aumentado é o número de *pacientes diagnosticados*, e não o número de *acometimentos*. Alguns fatores que explicam isso podem ser a flexibilização do diagnóstico (que passou a considerar o autismo como um espectro), e a conscientização da comunidade médica; o estímulo à busca de apoio profissional, cuja intervenção precoce geralmente resulta em desfechos positivos; e a investigação com base populacional, que expandiu o diagnóstico às pessoas com menos acesso aos procedimentos clínicos.

É preciso ter em mente, também, que as pesquisas utilizam diferentes procedimentos metodológicos, critérios de inclusão e amostras populacionais, fato que torna tais pesquisas difíceis de serem comparadas [35]. Com base nos indicadores disponíveis, porém, parece plausível afirmar o consenso de que a incidência de TEA seja maior em homens que em mulheres, e que os índices epidemiológicos têm aumentado ao longo do tempo – o que não significa que a *incidência geral* de fato esteja aumentando.

No tocante às causas do TEA, ainda não há estudos definitivos que estabeleçam sua etiologia. Buscando referências históricas, pode-se evocar as considerações de Bettelheim [10]. O autor contrariou as afirmações de Kanner de que o autismo seria uma síndrome inata, alegando que haviam indícios de fatores ambientais que, se não exclusivamente, influenciariam no desenvolvimento do comportamento autístico.

Foi Bettelheim que, na ocasião, descreveu o que ficaria conhecido como “hipótese da mãe-geladeira”: para ele, o fator responsável pelo aparecimento da síndrome seria a falta de contato afetivo com a criança na primeira infância. Essa tese foi amplamente aceita até a década de 1970. Nos dias de hoje, novos estudos demonstram resultados diferentes.

Johnson e Myers indicam que “o TEA é um distúrbio hereditário complexo que envolve múltiplos genes e apresenta grande variação fenotípica” [35, p. 1186]. Os autores apontam que em alguns casos (menos de 10%) o TEA pode estar associado a condições clínicas ou síndromes conhecidas. Para Pereira [61], há estudos que indicam o envolvimento de diversas regiões cerebrais, mas as anormalidades celulares e metabólicas não foram devidamente identificadas.

Novamente, os avanços clínicos na área ainda não se mostram suficientes para estabelecer conclusões definitivas sobre os fatores neurológicos, metabólicos e genéticos – de que se tem conhecimento – associados à etiologia do Transtorno do Espectro Autista. Em linhas gerais, o que se tem aceitado entre a comunidade médica, cada vez mais, é que as causas da síndrome sejam multifatoriais, com protagonismo genético e, com menor frequência, fatores ambientais [35].

2.3 Diagnóstico e Ferramentas de Triagem

Com a publicação do DSM-V, o autismo passou a ser tratado como um espectro de condições variadas, aumentando, desta forma, a sensibilidade dos critérios de diagnóstico. Por esta mesma razão, é recomendado que o processo de avaliação dos pacientes seja cauteloso. Os critérios que estabelecem a sintomatologia do TEA não estão discriminados no DSM-V, sendo que o próprio manual instrui que tais referências sejam extraídas de sua versão anterior, o DSM-IV.

A quarta edição do manual, como já dito anteriormente, traz as características para diagnóstico de cinco TIDs: (1) transtorno autista, (2) síndrome de Asperger, (3) síndrome de Rett, (4) transtorno desintegrativo da infância, e (5) transtorno invasivo do desenvolvimento sem outra especificação. A característica chave que diferencia o TEA das demais síndromes é a dificuldade de interação social. Outros fatores comuns do diagnóstico incluem limitações de comunicação verbal e não-verbal, e padrões comportamentais estereotipados.

Ademais, o autismo configura-se como uma síndrome espectral, cujos sintomas variam em grau de severidade. Isso significa que o nível de acometimento para cada indivíduo é absolutamente particular e específico. Por exemplo, as limitações de comunicação podem variar desde a completa ausência de linguagem verbal, passando por níveis intermediários (nos quais o indivíduo conhece palavras ou consegue formar frases simples), até os casos mais leves, em que o sujeito se comunica articuladamente. Nestes casos, pode haver apenas um ou outro traço de inaptidão linguística (inversão pronominal ou ecolalia, por exemplo).

Para efeitos de caracterização, listam-se, abaixo, alguns sintomas extraídos do DSM-IV. Note-se que os sintomas não estão necessariamente presentes em todos os casos, bem como podem se apresentar em níveis de severidade variados. Ademais, existem fatores físicos e relacionados à faixa etária de aparecimento dos sintomas que também influenciam o diagnóstico, de modo que o processo exige conhecimento técnico específico do profissional de saúde e acompanhamento do paciente. Alguns dos sintomas presentes no DSM-IV [4] são:

- Prejuízo nas interações não-verbais (troca de olhares, expressões faciais ou comunicação gestual);
- Incapacidade de estabelecer relacionamentos sociais;
- Ausência de reciprocidade social ou emocional;
- Atraso no (ou mesmo a completa ausência de) desenvolvimento de comunicação oral;
- Incapacidade de iniciar ou manter uma conversa;
- Uso repetitivo ou estereotipado do idioma, ou linguagem idiossincrática;
- Ausência do brincar lúdico;
- Preocupação demasiada com padrões estereotipados de interesse;
- Aderência inflexível a rituais ou rotinas específicas e não-funcionais;

- Maneirismos motores estereotipados ou repetitivos; e
- Preocupação persistente com partes de objetos.

As variações entre os diversos graus de severidade do autismo e também a estreita relação com outros transtornos torna o processo de diagnóstico uma tarefa complexa, longa e que requer acompanhamento clínico sistemático. Como forma de dar suporte à tomada de decisão referente ao diagnóstico do autismo, e evitar equívocos que podem ser absolutamente prejudiciais aos pacientes, o profissional de saúde pode utilizar *ferramentas de triagem e técnicas de vigilância*.

Vigilância é um processo flexível e contínuo cujo objetivo é auxiliar profissionais da área da saúde a identificar crianças que possam ter problemas relativos ao desenvolvimento. Trata-se de acompanhar o paciente por períodos consideráveis de tempo, a fim de identificar fatores de risco com precisão, possibilitando a documentação de registros históricos e, ao mesmo tempo, sanando dúvidas provenientes dos próprios pacientes ou seus responsáveis.

As ferramentas de triagem, por sua vez, são recursos que possibilitam ao profissional da saúde abordar qualitativamente as capacidades do paciente e compará-las ao sintomas específicos do TEA. Geralmente, as ferramentas de triagem apresentam um sistema de pontuação para cada quesito avaliado, e a somatória permite ao profissional decidir se o processo de vigilância deve ser mantido, ou se devem ser feitos encaminhamentos a outros profissionais para reavaliação.

Muitas são as ferramentas de triagem disponíveis na literatura atualmente. Johnson e Myers [35] explicam que elas podem ser separadas em dois grandes grupos, de acordo com seu. O primeiro grupo é composto por ferramentas aplicáveis a todas as crianças, no contexto do cuidado médico primário, visando a diferenciar indivíduos potencialmente portadores de TEA dos demais.

Um exemplo clássico de sua aplicação seria quando pais chegam com seus filhos ao consultório médico alegando que a criança apresenta problemas de desenvolvimento. Algumas das ferramentas deste grupo são: *Checklist for Autism in Toddlers* (CHAT) – e suas variações, CHAT-23 e M-CHAT), *Childhood Autism Spectrum Test* (CAST), e *Pervasive Developmental Disorders Screening Test II – Primary Care Screener*.

No segundo grupo, temos as ferramentas utilizadas em programas de intervenção precoce, cuja intenção é diferenciar indivíduos TEA daqueles com outros distúrbios do desenvolvimento. Neste caso, já existe a suspeita de que o paciente tenha seu desenvolvimento cognitivo comprometido, mas se deseja avaliar se o quadro é autístico. Neste grupo, alguns exemplos são: *Asperger Syndrome Diagnostic Scale* (ASDS), *Autism Behavior Checklist* (ABC), *Childhood Autism Rating Scale* (CARS), e *Screening Tool for Autism in Two-year-olds* (STAT), entre outras.

Dentre estas, talvez uma das mais populares seja a *Childhood Autism Rating Scale*. A CARS é uma ferramenta que apresenta tópicos de interesse para avaliação divididos em 15 categorias: (1) relacionamento interpessoal, (2) imitação, (3) resposta emocional, (4) uso do corpo, (5) uso de objetos, (6) adaptação à mudanças, (7) resposta visual, (8) resposta auditiva, (9) resposta ao paladar, olfato e tato, (10) medo ou nervosismo, (11) comunicação verbal, (12) comunicação não-verbal, (13) nível de atividade, (14) nível de coerência da resposta intelectual, e (15) impressões gerais [61].

Cada um destes tópicos está relacionado a uma escala que vai de 1 a 4, aumentando em intervalos de 0,5 ponto. Cada pontuação é acompanhada de uma breve explicação do que a caracteriza, possibilitando ao profissional de saúde que posicione adequadamente o paciente na escala. Após a avaliação de todos os tópicos, o profissional realiza a somatória dos pontos, e o escore total dará o resultado da aplicação. A pontuação final pode variar de 1 a 100, em intervalos de 10 pontos, sendo de 1-10 o pior cenário e de 90-100 o melhor.

Pacientes que pontuam entre 1-10 são considerados os casos mais graves, incluindo probabilidade de ferir gravemente a si mesmo ou aos outros, inclusive levando à morte. Conforme a pontuação aumenta, mais leves tornam-se os sintomas ou limitações presentes. Indivíduos posicionados entre 90-100 pontos são aqueles considerados sem problemas de desenvolvimento, ou cujo funcionamento é superior na ampla maioria das atividades monitoradas pela ferramenta [61].

Embora as ferramentas de triagem compartilhem o objetivo primário de detectar possíveis sintomas de autismo nos pacientes, elas não são intercambiáveis. Johnson e Myers [35] explicam que cada uma delas avalia determinadas faixas etárias, com relação a tópicos de desenvolvimento específicos, em determinados contextos clínicos. Além disso, a sua utilização

exclusiva não é suficiente para realizar um diagnóstico definitivo, que deve ser amparado por outros tipos de avaliação, incluindo, preferencialmente, mais de um profissional.

A abordagem das ferramentas também varia, podendo se dar com a aplicação de questionário, entrevistas com os pais, lista de verificação, interação direta com o paciente, ou uma combinação de várias destas técnicas. Algumas ferramentas apresentam questionários que devem ser respondidos pelos pais e são disponibilizadas gratuitamente na internet. Contudo, é importante ressaltar que apenas profissionais familiarizados com o material detêm conhecimento suficiente para aplicação, pontuação e interpretação dos resultados [35].

2.4 Prognóstico e Tratamento

O TEA é uma síndrome que não tem cura, mas é administrável por meio da intervenção médica e/ou do uso de medicamentos. Segundo Johnson e Myers [35], a melhora no quadro geral dos pacientes é difícil de prever, pois está relacionada a fatores como o nível de habilidades motoras, funcionamento cognitivo/intelectual e a severidade dos sintomas autísticos. Estudos recentes mostram melhora significativa em indivíduos TEA com o avanço da idade, especialmente naqueles com TID-SOE, embora persistam sintomas clínicos residuais, como estranheza social, interesses restritivos e estereotipia.

De maneira geral, o nível de habilidades cognitivas/intelectuais está diretamente relacionado com o prognóstico dos pacientes. Indivíduos com funcionamento intelectual normal e sintomas autísticos leves apresentam os melhores resultados, seguidos por aqueles com funcionamento intelectual normal e sintomas autísticos severos. Paralelamente, indivíduos com prejuízos intelectuais graves e sintomas severos são os que apresentam piores resultados dentro do espectro [35].

Segundo Klin [41], a maioria dos indivíduos acometidos pelo TEA não se torna capaz de viver de forma completamente autônoma, necessitando do apoio familiar ou institucionalização – dois terços das crianças autistas têm um desfecho pobre, e apenas um terço (aproximadamente) acaba adquirindo algum grau de independência na vida adulta. Ainda conforme a autora, a maioria das crianças autistas apresenta alguma melhora com o passar do tempo, mas somente uma minoria (um décimo dos indivíduos TEA) adquire capacidade de exercer atividade profissional e viver de maneira independente.

Apesar dos indivíduos com Síndrome de Asperger apresentarem prejuízos mais leves na cognição e na comunicação, não se pode afirmar que sejam eles a apresentar os melhores resultados prognósticos. Hans Asperger previa desfechos positivos nestes casos porque esses indivíduos eram, com frequência, capazes de exercer atividade profissional e sustentar-se financeiramente.

De fato, muitas destas crianças podem frequentar escola regular – mesmo sendo alvo de chacotas por conta de seu comportamento excêntrico. Contudo, não há estudos sistemáticos a longo prazo que comprovem resultados prognósticos mais positivos entre portadores de Síndrome de Asperger que entre outros níveis de acometimento [41].

O manejo do TEA pode envolver diversos tipos de profissionais e diversas ações, como intervenção comportamental, terapia de linguagem e programas educacionais. Além disso, muitos medicamentos são utilizados para o controle dos sintomas (hiperatividade, comportamento obsessivo/agressivo ou autoagressão), mas os ensaios clínico-farmacológicos nessa área são raros e, como o transtorno é crônico, a intervenção tende a se estender por longos períodos [61].

No âmbito educacional, dois métodos de intervenção destacam-se por sua popularidade: o TEACCH (*Treatment and Education of Autistic and related Communication-handicapped Children*) e o PECS (*Picture Exchange Communication System*). Ambos foram desenvolvidos para serem aplicados com crianças autistas e/ou com outros tipos de distúrbio de comunicação e comportamento. Estas técnicas têm sido utilizadas tanto no ambiente clínico quanto em escolas de educação especial como ferramenta de apoio ao ensino do indivíduo TEA.

O método TEACCH, desenvolvido pela Universidade da Carolina do Norte, é um programa extensivo que inclui serviços clínicos, avaliações diagnósticas, treinamento e materiais para serem utilizados na sua condução. Trata-se de uma estratégia que consiste em organizar o ambiente de trabalho da criança com estímulos visuais de tal modo que ela seja orientada quanto ao teor das atividades que precisa cumprir, bem como o procedimento utilizado e a sequencialidade das tarefas.

Cada tarefa é segmentada em partes menores, facilitando a compreensão e a performance do educando. Também é levado em consideração o nível de autonomia que cada criança possui. Assim, as tarefas podem ser realizadas com ou sem a intervenção do educador. Por meio de estímulos visuais, corporais e sinestésicos, busca-se estabelecer uma comunicação

(que não necessariamente a oral) com a criança, a fim de eliminar ou amenizar comportamentos indesejáveis e substituí-los por outros que promovam sua independência.

Por sua vez, o método PECS é um sistema de comunicação alternativa, constituído por um conjunto de cartões que apresentam uma palavra e sua respectiva imagem (por exemplo, a palavra “chocolate” e a ilustração de um chocolate), por meio dos quais a criança poderá desenvolver suas habilidades de comunicação. Quando quiser comer chocolate, a criança deverá escolher o cartão correspondente e entregá-lo ao terapeuta, indicando assim seu desejo.

No início do método, a criança aprende a associar a representação simbólica do objeto com seu correspondente lexical, por meio de que é estimulada a realizar trocas e interagir com o terapeuta. Aos poucos, o nível de dificuldade aumenta. Na última fase de aplicação do método, o paciente é estimulado com perguntas e encorajado a responder com sentenças mais complexas. Mesmo para as pessoas que já falam, o PECS pode ser utilizado para incrementar as habilidades de aproximação e iniciação da comunicação interativa, ou então para expandir o vocabulário.

Com o apoio destas ou outras estratégias, fica a cargo da instituição de ensino e da equipe pedagógica decidir qual a melhor abordagem de intervenção ao indivíduo TEA. Segundo Orrú [58], os educadores devem ser capazes de enxergar a criança autista sem o estigma patológico que geralmente se associa a ela. Muitas vezes, conforme a autora, essas pessoas são estigmatizadas em função daquilo que “não podem” ou “não conseguem” fazer, levando educadores a negligenciar a busca por estratégias interventivas e modelos pedagógicos que as auxiliem a desenvolver-se.

3 ACESSIBILIDADE E INCLUSÃO

Ainda que muito sobre o Transtorno do Espectro Autista permaneça obscuro do ponto de vista clínico, é fato que esta parcela da população está inserida na sociedade, e demanda soluções que garantam sua inclusão social e promovam maior qualidade de vida. A complexidade do quadro do indivíduo TEA certamente impõe barreiras, mas estas precisam ser superadas pelos esforços científicos e corporativos na busca por uma sociedade mais igualitária. Neste capítulo, serão abordadas as características de comportamento do indivíduo autista no ambiente digital, suas preferências e hábitos. Além disso, observa-se a crescente expansão de softwares para este público, e quais são os avanços realizados em termos de acessibilidade e design.

3.1 Comportamento no Ambiente Virtual

Apesar da complexidade clínica que envolve o público autista e das dificuldades enfrentadas por estes no mundo real, o seu engajamento no ambiente digital trilha caminhos diferentes. Ao contrário do que se poderia imaginar, o indivíduo autista tem se mostrado cada vez mais ativo na utilização de recursos tecnológicos no ciberespaço. Isso se pode comprovar, por exemplo, pela ascensão do número de aplicativos mobile desenvolvidos para o indivíduo TEA e seu familiares, conforme demonstra Fletcher-Watson [23].

Televisão, vídeo games e computadores estão entre os preferidos do público autista, comportamento este que tem despertado o interesse de pesquisadores e profissionais. Kuo et al [45] realizaram um estudo com adolescentes TEA na faixa dos 12 a 18 anos de idade, em Baltimore, Estados Unidos, acerca da utilização destas mídias, e encontraram resultados surpreendentes. A média de utilização diária foi de 2,3 horas para televisão e 4,9 horas para o computador, das quais 2 horas eram gastas navegando-se em websites e outras 2 horas com jogos on-line. Resultados similares podem ser encontrados em Mazurek e Wenstrup [54].

Além da frequência de uso, Kuo et al [45] realizaram também um mapeamento das preferências de conteúdo em cada uma dessas mídias. No caso da televisão, os programas mais assistidos são desenhos animados, correspondendo a 37% da amostra. Em seguida, aparecem programas de comédia, filmes e noticiários. No caso dos vídeo games, a preferência

está em jogos de ação, atingindo a marca de 46% dos participantes. Outros gêneros bastante mencionados são RPGs (*Role-playing games*), simuladores, esportes e estratégia.

Com os computadores, a principal utilização é acessar websites em busca de informações relativas ao entretenimento, como sites sobre vídeo games, animes, música, ou filmes. Em segundo lugar está o próprio entretenimento, como jogos on-line ou *streamings* de filmes, músicas e programas de televisão. Uma pequena porcentagem da amostra também mencionou o uso do computador para realizar compras ou assistir a noticiários. Também foi averiguado o uso do computador para comunicar-se e relacionar-se com outros, como o envio de e-mails ou a utilização de programas de mensagens instantâneas, sendo que este aspecto não passou de 15% da amostra.

Este resultado contrasta significativamente com as descobertas feitas por Mazurek [51]. Neste caso, o autor conduziu uma pesquisa sobre o uso de mídias sociais entre adultos diagnosticados com Transtorno do Espectro Autista, na faixa dos 18 a 62 anos de idade. O autor reporta que quase 80% do número de participantes afirmou fazer uso de mídias sociais como *Facebook* e *MySpace*, com uma frequência média de 3,2 horas por dia.

Ademais, o autor indica que o uso de mídias sociais se dá por variados motivos, como contato com a família, trabalho, entretenimento, e a criação de conexões sociais com novos amigos, sendo este último o responsável por 79,2% da utilização. Este resultado é muito diferente dos estudos conduzidos com adolescentes. Segundo o autor, tal preferência está diretamente relacionada com o sentimento de solidão experimentado pelo adulto autista, que busca nas mídias sociais formas de compensar sua falta de habilidade com interações sociais no ambiente real.

Este fenômeno não é exclusividade do uso de computadores, contudo. O uso massivo de mídias eletrônicas por indivíduos TEA pode ser explicado por dois motivos fundamentais. O primeiro é exatamente a sensação de conforto que esses canais transmitem ao observador. Como assistir televisão ou jogar vídeo games são atividades que exigem pouco esforço social ou cognitivo, são meios que permitem ao indivíduo autista evitar o contato direto com outras pessoas [53].

Por outro lado, tais atividades vão de encontro com a afinidade cognitiva deste público. Estudos indicam que os indivíduos autistas são “*visual thinkers*”, ou seja, estão mais confortáveis quando são levados a utilizar representações mentais visuais [44, 37]. Embora não haja consenso sobre o motivo desta ocorrência, os estudos conduzidos com os indivíduos TEA

apontam com sugestiva veemência que seu processamento visual é aguçado. Desta forma, as mídias digitais também representam uma “zona de conforto” ao público autista.

Embora isso revele um potencial a ser explorado, certamente há entraves que merecem atenção. Em primeiro lugar, a utilização excessiva de tecnologias digitais pode estimular comportamentos repetitivos ou isolamento social, já que, ao engajar em atividades virtuais, o indivíduo pode negligenciar parcial ou completamente a busca por interação social no ambiente real [62, 45]. Não obstante, quadros de ansiedade ou depressão têm se tornado cada vez mais comuns, sobretudo em adultos autistas [51].

Por outro lado, a preferência praticamente exclusiva por atividades em locais fechados tem se mostrado problemática também para crianças e adolescentes autistas. Mazurek e Engelhardt [52] realizaram um estudo acerca do uso de vídeo games por meninos, comparando os portadores de autismo com aqueles com desenvolvimento típico. As métricas incluíam faixa etária (cujas médias foram de 12 anos), número de irmãos, etnia, renda familiar e estado civil dos pais. De modo geral, os autores descobriram que os adolescentes TEA passam muito mais tempo jogando vídeo game em comparação ao grupo com desenvolvimento típico. Este comportamento sedentário contribui para que eles desenvolvam distúrbios de sono e obesidade. Daí a importância de se pensar programas de intervenção que estimulem a prática de atividades físicas, o contato com ambientes naturais e a interação social não-mecânica [49].

Como se pode notar, o ambiente virtual não é estranho ao indivíduo TEA, que reivindica seu lugar no ciberespaço. Pelo contrário, o alto nível de percepção e processamento visual que estes indivíduos apresentam tornam as mídias digitais meios bastante adequados para a atuação clínica e o entretenimento. Contudo, o nível de sofisticação das intervenções digitais precisa acompanhar as especificidades clínicas do público, a fim de evitar que novas complicações clínicas se somem ao já complexo quadro do indivíduo autista.

3.2 Softwares Educacionais

Em resposta aos padrões de comportamento do público TEA em ambientes digitais, pesquisadores e equipes de desenvolvimento começaram a propor softwares educacionais de intervenção, ou seja, softwares cuja finalidade é facilitar o processo de ensino-aprendizagem destes indivíduos. Esta tendência pode ser observada nos trabalhos de Fletcher-

Watson [23], Ramdoss et al [62] e Wass e Porayska-Pomsta [73], os quais reúnem exemplos de softwares de intervenção TEA disponíveis na literatura.

Embora todos tenham o objetivo primário de servir como recurso educacional ao público autista, tais softwares são bastante variados entre si. Eles apresentam, por exemplo, inúmeros objetivos pedagógicos voltados a diferentes graus de severidade do transtorno. Também utilizam diversas abordagens e recursos tecnológicos para cumprir suas funções, como Realidade Virtual, simuladores, robótica, gamificação, e até mesmo Kinect™ e terapia musical. Além disso, existem variadas plataformas de acesso (desktop, mobile, aplicação on-line) e locais de utilização (casa, escola, ambiente clínico, etc.).

Hulusic e Pistoljevic [32], por exemplo, propõem uma série de jogos de computador para o ensino de habilidades básicas às crianças autistas. Os jogos, baseados em estímulos visuais e auditivos, têm a intenção de promover o desenvolvimento de associações entre elementos, refinamento de habilidades motoras, e a categorização de conceitos e formas. Este recurso, segundo os autores, é voltado a crianças autistas em fase pré-escolar, ou nos primeiros anos da educação básica, e deve ser aplicado como suporte no próprio ambiente de ensino.

Com o uso de Realidade Virtual (RV), podemos citar o trabalho de Winoto, Xu e Zhu [76]. Os autores desenvolveram um ambiente virtual, acessado com o uso de óculos especiais, em que objetos do dia-a-dia são apresentados às crianças. Ao olhar fixamente para um objeto por um determinado período de tempo, o sistema envia um estímulo auditivo com o nome do objeto. Os autores advogam que esta estratégia faz com que os indivíduos autistas relacionem a imagem do objeto com seu correspondente lexical. Esta intervenção pode ser utilizada tanto na escola quanto em casa.

Ainda sobre a comunicação como objetivo pedagógico, De Leo et al [19] propõem uma adaptação do sistema PECS para dispositivos mobile. Professores ou cuidadores acessam um banco de dados e selecionam as imagens adequadas ao nível cognitivo e às expectativas de aprendizagem para cada criança. Por sua vez, a criança autista utiliza as imagens e palavras disponíveis no sistema mobile para expressar seus desejos ou necessidades. O sistema possui níveis de dificuldade que vão tornando a representação simbólica mais complexa.

Outra abordagem possível é a utilização de tecnologias digitais no auxílio à organização e realização de tarefas cotidianas. Barros et al [8] apresentam um aplicativo mobile

que funciona como uma agenda para crianças TEA entre 8 e 12 anos, indicando as atividades a serem cumpridas e mostrando, por meio de estímulos visuais e auditivos, o passo a passo para sua realização. Professores, pais ou cuidadores monitoram e controlam o progresso da criança, recompensando-a quando sua performance for bem-sucedida.

No tocante à interação social, um estudo que merece destaque encontra-se em Washington et al [72]. O trabalho descreve o desenvolvimento de um sistema para reconhecimento automático de expressões faciais acoplado ao Google Glass. Reconhecer expressões faciais é uma tarefa difícil para os indivíduos TEA, o que pode prejudicar sua desenvoltura ao interagir com outros. Neste sistema, a criança autista recebe dicas visuais acerca da expressão facial das pessoas que encontra, o que as auxilia na dinâmica de relacionamentos sociais. Os autores encorajam que este recurso seja utilizado em qualquer ambiente, sempre com a supervisão dos pais.

O uso do Kinect™, por sua vez, é apresentado no desenvolvimento de um simulador descrito por Cai et al [16]. A simulação desenvolvida pelos autores parte da terapia com golfinhos para propor uma intervenção educacional. A tecnologia Kinect™ é utilizada para rastrear os movimentos da criança autista enquanto ela “treina” os golfinhos no simulador, encorajando-a a estabelecer meios não-verbais de comunicação. Esta intervenção demanda um ambiente controlado de aplicação, com a possibilidade de projetar a simulação, com a disponibilidade de equipamento de som e, obviamente, com um aparelho dotado de tecnologia Kinect™.

Por fim, Lima e Castro [48] propõem uma intervenção clínica aliada à terapia musical. Segundo os autores, esta abordagem possibilita às crianças autistas desenvolverem habilidades de comunicação, imaginação e interação social. O sistema desenvolvido utiliza recursos de Realidade Virtual para ensinar as crianças a tocar instrumentos, seguindo partituras virtuais. Este recurso foi desenvolvido como aplicativo para iOS, e explora a comunicação com grupos de amigos para incentivar o cumprimento das atividades propostas.

Os exemplos listados até aqui não têm a intenção de representar a totalidade das possibilidades de intervenção disponíveis atualmente, mas sim de demonstrar, ainda que sucintamente, a variedade de abordagens presentes na literatura. Claro está que esta área tem crescido exponencialmente e, conforme o corpo de estudo torna-se cada vez mais robusto, novas descobertas servem de base para que a pesquisa científica continue avançando. Por outro lado, as

implicações práticas também revelam desafios que precisam ser considerados ao longo do processo de desenvolvimento, conforme será descrito no próximo tópico.

3.3 Acessibilidade Digital

Segundo a Organização das Nações Unidas, pessoas com deficiências incluem “todos aqueles com limitações físicas, mentais, intelectuais ou sensoriais de longo prazo, cuja interação com variados obstáculos pode dificultar sua participação plena e efetiva na sociedade em comparação aos demais” [56, tradução nossa]. Frente às complexas condições clínicas que envolvem o Transtorno do Espectro Autista, está claro que estes indivíduos necessitam de soluções que se alinhem aos princípios da acessibilidade.

A acessibilidade é um termo abrangente que pode variar quanto ao seu entendimento específico dependendo da área em que é utilizada. Ainda segundo a ONU [56], é ela que viabiliza às pessoas com deficiência viver com independência e participar plenamente em todos os aspectos da vida. É responsabilidade do Estado garantir que esta premissa se faça cumprir, garantindo às pessoas com deficiência acesso “ao ambiente físico, ao transporte, à informação e comunicação, incluindo sistemas e tecnologias da informação e comunicação, à infraestrutura e aos serviços públicos ou disponibilizados ao público, tanto em áreas urbanas quanto rurais” [56, p. 8, tradução nossa].

O relatório apresentado pela ONU inclui, também, direcionamentos importantes sobre a implantação de conceitos, a disseminação de informação, aos direitos humanos das pessoas com deficiência e aos tipos de adaptações estruturais a serem realizadas pelos países-membros. O Brasil é signatário deste protocolo, e promulgou sua execução por meio do Decreto nº 6.949, de 25 de agosto de 2009 [12]. Apesar disso, a acessibilidade plena às pessoas com deficiência e, sobretudo, ao público autista, ainda parece ser uma realidade distante.

De fato, a acessibilidade parece bem mais tangível quando as limitações são físicas. Ao se falar em pessoas cadeirantes, imediatamente pensamos em corrimões ou rampas de acesso. Para pessoas cegas, temos soluções em braile ou com estímulos sonoros. Se a limitação é auditiva, fala-se em legendar conteúdos ou disponibilizar alternativas em linguagem de sinais. O que fazer, contudo, quando se trata de uma defasagem intelectual ou cognitiva?

Infelizmente, este questionamento ainda não encontra respostas satisfatórias nos avanços científicos sobre a acessibilidade. Tome-se como exemplo o *Web Content Accessibility Guidelines* (WCAG 2.0), disponibilizado pela *World Wide Web Consortium* (W3C). O WCAG 2.0 é um dos documentos mais populares e respeitados acerca da acessibilidade digital, servindo de inspiração para muitos pesquisadores e desenvolvedores no mundo todo. As diretrizes nele presentes versam sobre a acessibilidade em vários níveis, incluindo alternativas textuais e sonoras, soluções operacionais de hardware (relativas ao uso do teclado e mouse, por exemplo), e questões de compatibilidade [78].

No entanto, ele não trata da acessibilidade voltada especificamente às limitações cognitivas ou intelectuais. O mesmo ocorre na ISO/IEC 40500:2012, que é derivada do WCAG 2.0. Já na ISO/IEC 71:2014 (*Guide for addressing accessibility in standards*) há uma subseção a respeito de limitações cognitivas. Embora as informações contidas nesta norma sejam inegavelmente valiosas, não existe a indicação de quais diretrizes sejam mais adequadas ao público autista, nem uma explicação detalhada de como atingir os objetivos contidos nela. A mesma norma também destina seções às deficiências físicas, sensoriais e imunológicas [33].

Isso acontece porque tanto o WCAG 2.0 quanto as normas ISO mencionadas baseiam-se na abordagem da *acessibilidade universal*. Segundo Payá e Navarro [60], a acessibilidade universal (também chamada de *design universal*) é uma questão de caráter moral-político, na medida em que se preocupa com a proposta de soluções para todas as pessoas, em igual medida, sem a necessidade de adaptação ou design específico. Em outras palavras, a utilização de produtos ou serviços deve independe das limitações enfrentadas pelos sujeitos.

Evidentemente, esta é uma abordagem útil, em especial quando se tratam de produtos e/ou serviços voltados a atingir o maior número de pessoas. Por exemplo, é interessante que o site de uma universidade seja acessível a todos, pois qualquer pessoa tem direito à educação. Da mesma forma, uma empresa de dispositivos eletrônicos consegue atingir um público maior de clientes se utiliza os princípios da acessibilidade universal.

Esta afirmação também é verdadeira no caso dos indivíduos autistas, especialmente em ambientes digitais, pois como explicado anteriormente o público TEA é ativo e tem se engajado cada vez mais no ciberespaço. Entretanto, quando se fala em softwares educacionais, o escopo é bem mais específico. Estes recursos se prestam a atender um público

determinado (no caso, o público TEA) e, portanto, as soluções de acessibilidade são melhor aproveitadas quando o direcionamento é mais exclusivo.

Pesquisas a respeito da acessibilidade específica ao indivíduo autista existem, mas não são tão numerosas. Um ponto de consenso entre os pesquisadores, contudo, parece ser o do alto nível de processamento visual deste público, como já mencionado anteriormente. Por esse motivo, suportes visuais são uma estratégia amplamente adotada no desenvolvimento de softwares acessíveis ao indivíduo TEA, uma vez que ajudam a promover suas habilidades de interação com as tecnologias.

Fletcher-Watson [23] também menciona a importância de se fornecer feedback imediato ao usuário autista, como forma de motivá-lo e aumentar seu grau de concentração e confiança. Este feedback pode ocorrer de formas variadas, com o uso de texto, sons e imagens, por exemplo. A autora também indica a utilização de recompensas do mundo real. Quando o indivíduo autista completa satisfatoriamente uma atividade no ambiente digital, ele é recompensado com estímulos reais que lhe agradem. Isso serve para reforçar a ideia de ter atingido resultados positivos, encorajando-o a repetir tal comportamento.

Com relação ao design, uma série de recomendações voltadas especificamente à composição, uso de elementos gráficos e aspectos de usabilidade acessível podem ser encontradas em Britto [13]. A autora propõe um guia de diretrizes de acessibilidade web para o público TEA dividido em 10 categorias: (1) vocabulário visual e textual, (2) customização, (3) engajamento, (4) representações redundantes, (5) multimídia, (6) visibilidade do estado do sistema, (7) *affordance*, (8) navegabilidade, (9) resposta às ações, e (10) interação com tela sensível ao toque.

Cada uma destas categorias é subdividida em tópicos, que a autora explica em detalhes, incluindo a descrição do tópico, a justificativa, o direcionamento gráfico que deve ser adotado para a sua realização bem-sucedida, e exemplos de aplicação. As práticas descritas são o resultado de uma extensa pesquisa realizada tanto na literatura disponível quanto por meio de questionário. De acordo com Britto [13], e seguindo as categorias listadas acima, as principais recomendações para a acessibilidade digital aos portadores de TEA são:

(1) Vocabulário visual e textual: manter bom contraste de cores entre figura e fundo, e não utilizar cores como única forma de transmitir conteúdos e significados; utilizar linguagem visual e textual simples e familiar ao usuário, evitando o uso de jargões, metáforas ou

abreviações; evitar textos longos e utilizar marcações que facilitem a leitura (títulos e subtítulos); e utilizar iconografia que seja compatível com elementos do mundo real, facilitando seu reconhecimento e entendimento.

(2) Customização: permitir a customização de elementos da página, como cores, fontes e tamanho de texto; oferecer opções de customização de texto, sons e imagens, adaptando-se as preferências do usuário; permitir a customização da quantidade e arranjo dos elementos; e incluir modo de leitura ou impressão para atividades que exigem concentração.

(3) Engajamento: evitar a utilização de elementos que distraiam a atenção (quando essenciais, deve haver a possibilidade de suprimi-los); projetar interfaces simples, que contenham apenas os elementos necessários à realização da tarefa atual; utilizar espaços em branco na interface para separar conteúdos ou criar áreas de atenção; e fornecer instruções e orientações claras para a realização de tarefas.

(4) Representações redundantes: oferecer múltiplas formas de representação de conteúdo: textual, imagética, sonora, etc.; incluir descrição textual de símbolos, pictogramas e ícones; e fornecer legendas em áudio para texto.

(5) Multimídia: fornecer informações em diferentes mídias; permitir que imagens sejam ampliadas para melhor visualização; e evitar o uso de sons que possam ser perturbadores.

(6) Visibilidade do estado do sistema: fornecer mensagens para facilitar a interação, e também sobre erros e como solucioná-los; permitir que ações críticas possam ser canceladas, desfeitas ou revertidas; e permitir que o usuário realize até 5 tentativas em atividades antes de mostrar a resposta correta.

(7) Affordance: facilitar a previsibilidade do sistema, ou seja, elementos e interações similares devem produzir os mesmos resultados; usar botões, ícones e controles com área ampla, que facilitem o clique ou o toque; e fornecer feedback imediato como resposta à alguma ação importante do sistema.

(8) Navegabilidade: permitir navegação simples entre as páginas, com opções de avanço e retorno claras e de fácil acesso; e evitar o redirecionamento automático de páginas ou a perda de informações em campos de entrada.

(9) Resposta às ações: fornecer feedback multimídia sobre a confirmação de ações e possibilidade de erro.

(10) Interação com tela sensível ao toque: fornecer soluções que evitem a seleção indesejada ou toque acidental em dispositivos mobile.

Já com relação à avaliação de interfaces gráficas para indivíduos TEA, um bom exemplo encontra-se nas heurísticas para verificação de sistemas interativos proposta por Khowaja e Salim [39]. Os autores sugerem um grupo de 15 heurísticas baseadas naquelas propostas originalmente por Jakob Nielsen. As 5 primeiras são as mesmas do grupo original de Nielsen, as 5 seguintes são versões atualizadas das originais, e as 5 últimas são heurísticas completamente novas.

Desta maneira, as 15 heurísticas propostas pelos autores são: (1) visibilidade do estado do sistema; (2) equivalência com o mundo real; (3) consistência e padronização; (4) reconhecimento ao invés de memorização; (5) estética e design minimalista; (6) controle e liberdade do usuário (atualizada); (7) prevenção de erros (atualizada); (8) flexibilidade e eficiência de uso (atualizada); (9) reconhecimento, diagnóstico e recuperação de erros (atualizada); (10) ajuda e documentação (atualizada); (11) personalização de elementos; (12) sutileza na progressão entre telas; (13) responsividade do sistema; (14) histórico e monitoramento de atividades; e (15) uso de comunicação multimídia.

Os estudos mencionados acima são inegáveis contribuições rumo à acessibilidade aos indivíduos autistas. Entretanto, pesquisas do gênero estão ainda ganhando corpo e não são encontradas em números expressivos. Ao passo que os esforços referentes à acessibilidade universal são importantes na inserção social dos indivíduos com deficiência, se faz necessária a busca por soluções que contemplem especificamente o público TEA, propondo meios de atender as necessidades desta parcela da população que normalmente é negligenciada.

4 DESENVOLVIMENTO DE SOFTWARE ACESSÍVEL

O interesse pelo público autista parece ter se difundido entre a comunidade acadêmica e corporativa nos últimos anos. Por um lado, houve um aumento significativo no número de softwares educacionais para estes indivíduos. Por outro, a escassez de recursos de acessibilidade voltadas ao TEA ainda impõem barreiras que precisam ser superadas no universo digital. Neste sentido, é preciso atentar para as práticas atuais de desenvolvimento de software, que certamente impactam na maneira como este público recebe e lida com os artefatos digitais. Este capítulo tratará dos modelos de desenvolvimento, tanto tradicionais como ágeis, e também da abordagem do Design Centrado no Usuário, que busca solucionar problemas de usabilidade. Por fim, fala-se sobre a gestão da qualidade, incluindo os Modelos de Maturidade e as Lições Aprendidas.

4.1 Modelos de Desenvolvimento

O desenvolvimento de software é uma área da Engenharia de Software que se preocupa com as metodologias e os procedimentos inerentes à sua obtenção. Para tanto, emprega diversos processos, ou seja, grupos de atividades que, quando ordenados em uma sequência lógica, fornecerão os subsídios necessários à produção do software. Historicamente, os Processos de Desenvolvimento de Software (ou, simplesmente, PDSs) evoluíram no sentido de se adaptar às mudanças impostas pela sociedade moderna.

O primeiro modelo de desenvolvimento de que se tem notícia (ou, ao menos, o mais famoso), é o modelo Cascata (*Waterfall*, em inglês). Este modelo foi proposto na década de 1970, e começa com uma extensa e longa fase de pesquisa em que a equipe de desenvolvimento estabelece um grupo definitivo de requisitos de projeto. Em seguida, a equipe de desenvolvimento utiliza os requisitos para construir a arquitetura do software. Então, os programadores codificam e implementam o projeto, antes do software ser testado e, finalmente, entregue [18].

O modelo Cascata possui um fluxo de desenvolvimento bastante linear, o que em teoria ajudaria a minimizar riscos durante o processo. Entretanto, a prática das equipes de desenvolvimento se mostra diferente. Os clientes, muitas vezes, mudavam de ideia com o tempo,

e os requisitos que tinham sido levantados ao longo de meses se alteravam, transformando o projeto completamente. Na tentativa de solucionar esse problema, emergiram as técnicas incrementais e iterativas, evoluções do modelo Cascata.

De acordo com Cohen, Lindvall e Costa [18], ambas as técnicas buscam dividir o ciclo de vida de desenvolvimento em partes menores, possibilitando uma abordagem multitarefa que agiliza o processo. Como no modelo Cascata, o modelo Incremental também se inicia com o levantamento de requisitos. No entanto, a entrega do software é feita em módulos (ou “incrementos”) que são individualmente funcionais. Cada novo módulo é agregado ao software à medida que fica pronto, resultando em um produto de software completo ao final do processo.

O processo Iterativo, por sua vez, divide o ciclo de desenvolvimento em partes chamadas “iterações”. Cada iteração compreende um fluxo cíclico com as mesmas etapas presentes no modelo Cascata (levantamento de requisitos, análise, design, implementação e teste). Cada iteração gera, também, um protótipo funcional do software, ao qual se vão agregando novas partes conforme as iterações avançam. Uma diferença marcante com relação ao modelo Cascata é que, neste caso, os requisitos não contemplam o projeto todo desde o início: a iteração começa assim que os requisitos necessários àquela etapa estejam definidos.

Os processos Incrementais e Iterativos representam uma evolução do modelo Cascata pela agilidade conferida ao possibilitar etapas concomitantes de trabalho. Ademais, essas técnicas lidam melhor com mudanças de requisitos de projeto e gestão de riscos. Ainda, assim, eles nem sempre se comportavam como o esperado. A resposta às mudanças drásticas de requisitos nem sempre eram suficientemente rápidas, e a documentação excessiva gerada por esses processos tornava o ciclo moroso e nem sempre satisfatório.

Ao longo da década de 1990, os métodos tradicionais de desenvolvimento foram ficando obsoletos, ao passo que se reconhecia, cada vez mais, a necessidade de um processo flexível, centrado no cliente e seus usuários finais, que oferecesse um fluxo adaptável às mudanças de um ambiente dinâmico e em constante transformação [18]. Em resposta a essa demanda, e em busca de soluções inovadoras para a indústria do desenvolvimento de software, um grupo de pesquisadores e profissionais se reuniu em 2001. Deste encontro surgiu, então, o Manifesto Ágil.

Este Manifesto trouxe consigo uma nova maneira de se produzir softwares: as *metodologias ágeis de desenvolvimento*. Na verdade, esse pensamento já estava em processo de amadurecimento durante a década de 1990, sendo que o Manifesto Ágil tem mais importância como um marco simbólico do que como a criação de novas metodologias propriamente ditas.

Quando se fala em método ágil não se está descrevendo um método único, mas sim um conjunto de metodologias que seguem a mesma linha de raciocínio. De fato, existem muitas metodologias ágeis, tais como: *Extreme Programming* (XP), Scrum, Metodologia de Desenvolvimento de Modelos Dinâmicos (MDMD), Desenvolvimento de Software Adaptativo (DSA), Desenvolvimento Guiado por Funcionalidades (do inglês, *Feature-Driven Development*), metodologias *Crystal*, entre outras.

Cada uma dessas metodologias tem suas particularidades em termos de composição da equipe, papéis e responsabilidades, artefatos e documentação, fluxo de trabalho, organização e entregas funcionais. Desta maneira, é uma tarefa muito difícil descrever cada uma delas em profundidade. No entanto, todas elas compartilham princípios comuns, que as caracterizam como metodologias ágeis. O Manifesto Ágil [9] traz quatro princípios básicos. Os métodos deviam valorizar:

- **Indivíduos e interações mais que processos e ferramentas:** o foco está na funcionalidade do produto em si, mesmo que o processo de desenvolvimento não seja tão rigoroso;
- **Software funcional mais que documentação abrangente:** a entrega de software funcional se sobrepõe à geração de documentação excessiva que, em muitos casos, acaba não sendo consultada posteriormente;
- **Colaboração com o cliente mais que negociação de contratos:** entender as necessidades do cliente deve ser o foco principal das equipes de desenvolvimento; e
- **Responder a mudanças mais que seguir um plano:** agilidade em atender mudanças de requisito muitas vezes exige alterar drasticamente o planejamento.

Por meio destes valores demonstra-se, claramente, a mudança de paradigma ocasionada em relação às metodologias tradicionais de desenvolvimento. Garg [26] indica que as metodologias ágeis visam à satisfação do cliente, que mede o progresso por meio das entregas parciais de software funcional. Mudanças de requisitos são bem-vindas, mesmo em fases avançadas do projeto, e são resolvidas pelas próprias equipes, que se organizam e se auto-gerenciam. Ademais, ainda de acordo com o autor, os projetos ágeis preveem estreita colaboração entre desenvolvedores e *stakeholders*, encorajando a comunicação direta e cara a cara, sempre que possível.

Para facilitar o entendimento do conjunto de abordagens ágeis, Fagundes [21], apresenta uma adaptação que analisa diversas metodologias a partir das atividades propostas no processo Incremental. A autora sugere que as metodologias ágeis apresentam etapas comuns de desenvolvimento, que podem ser resumidas em 8 fases: (1) definir esboço dos requisitos, (2) atribuir requisitos às iterações, (3) projetar arquitetura do sistema, (4) desenvolver incremento do sistema, (5) validar incremento, (6) integrar incremento, (7) validar sistema, e (8) entrega final. A representação deste esquema pode ser observada na Figura 1.

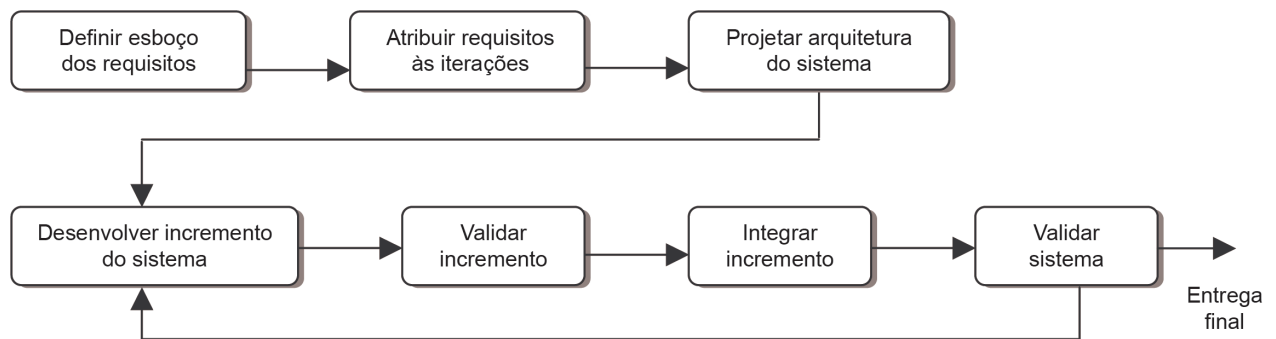


Figura 1. Etapas ágeis comuns [21].

A **definição do esboço dos requisitos** é o momento em que serão definidos os requisitos funcionais e não-funcionais do software. Esta etapa não é diferente daquela presente nos modelos tradicionais de desenvolvimento, mas é preciso lembrar que, na abordagem ágil, não há a necessidade de um grupo definitivo de requisitos para começar o processo. No Scrum, por

exemplo, esse processo é chamado de *product backlog*, enquanto que no XP, a atividade é direcionada aos clientes, que escrevem as *histórias de usuário*.

Em seguida, ocorre a etapa de **atribuição dos requisitos às iterações**, ou seja, os requisitos levantados previamente serão alocados à equipe e aos prazos em uma sequência lógica. É aqui que haverá uma priorização dos componentes a serem desenvolvidos e uma divisão do software em partes menores, independentes, para que ao final de cada iteração resulte um componente de software funcional. No Scrum, ocorre a definição da *sprint backlog*, em que cada *sprint* corresponde a um ciclo de iteração. No XP, a equipe seleciona quais histórias de usuário serão desenvolvidas na próxima iteração. Em ambos os casos, as iterações não duram mais do que 30 dias.

Embora apareça como a terceira atividade no fluxo proposto por Fagundes [21], o **projeto da arquitetura do sistema**, na realidade, é uma etapa que acontece paralelamente ao levantamento de requisitos. Conforme os requisitos vão sendo definidos, os engenheiros de software e analistas constroem as camadas necessárias ao funcionamento do produto. Nem o Scrum nem o XP indicam ferramentas ou artefatos para esta atividade, mas ambas as metodologias assumem que a arquitetura vai tornando-se mais robusta ao longo do projeto, em função das mudanças naturais de requisitos.

Continuando o ciclo de vida, chega-se à etapa de **desenvolvimento do incremento do sistema**. Neste momento, ocorre a implementação dos requisitos programados para a iteração em curso (o que, na prática, contempla tanto a etapa de design de interface quanto a de codificação). Algumas técnicas sugeridas pelo XP incluem a programação em pares e a refatoração de código. O Scrum não menciona técnicas de implementação, mas recomenda a realização de reuniões diárias com a equipe para verificação do desenvolvimento.

Uma vez que o incremento é desenvolvido, ocorre a etapa de **validação do incremento**, ou seja, os testes para verificar que aquele incremento está de acordo com os requisitos. No XP, a equipe de desenvolvimento realiza os testes unitários, enquanto que os clientes realizam testes de aceitação. A metodologia XP indica que os testes devem ser planejados antes da implementação do incremento. O Scrum não indica nenhuma técnica para realização dos testes, mas eles acontecem sempre ao final do ciclo corrente de iteração, coincidindo com o final da *sprint*.

Após a validação bem-sucedida do incremento, é hora de integrá-lo ao restante do sistema. Na etapa de **integração do incremento**, as partes do software que foram sendo desenvolvidas nas iterações se unem e ganham corpo, passando a funcionar em unidade. O XP, por exemplo, advoga que esta integração deve ser contínua, ou seja, deve acontecer conforme o projeto vai se desenvolvendo. A prática da integração ao longo do processo permite que erros de compatibilidade sejam diagnosticados e corrigidos (ou mesmo evitados) em etapas precoces do ciclo de vida.

É importante lembrar que essas etapas se repetem quantas vezes forem necessárias, até que todos os incrementos tenham sido desenvolvidos, validados e integrados, conforme o planejamento das iterações. Ao final da última iteração, ocorre a **validação do sistema**. Nesta etapa, o software completo é disponibilizado ao cliente para que este avalie o sistema como um todo. Caso surjam problemas ou alterações, a equipe de desenvolvimento deve realizar os refinamentos finais no produto.

Por fim, ocorre a etapa da **entrega final**, que representa mais um marco físico de encerramento do projeto do que um grupo de atividades a serem desenvolvidas. Na verdade, ocorre exatamente o oposto: o produto de software está pronto para a entrega quando *não há mais atividades a serem executadas*. O cliente deve estar plenamente satisfeito com o produto que lhe foi entregue, e não deve ter mais solicitações a fazer.

As metodologias ágeis representaram uma revolução tão grande na maneira de se desenvolver softwares que, nos dias de hoje, os modelos tradicionais praticamente se extinguíram na prática corporativa. Tome-se como exemplo o estudo realizado por Dingsøyr et al [20]. Os autores realizaram uma revisão sistemática sobre estudos contemplando metodologias ágeis entre os anos de 2001 e 2010, e descobriram um aumento de quase oito vezes no número de publicações.

Foram analisados 63 países dos cinco continentes, em bancos de dados das dez mais importantes conferências do mundo sobre o tema, revelando um interesse cada vez maior na metodologia XP, e um declínio no número de estudos sobre o Scrum. Segundo os autores, embora o estudo refira-se à comunidade acadêmica, ele certamente reflete as práticas emergentes na indústria. Por esse motivo, trabalhos com foco no desenvolvimento de software devem adotar, preferencialmente, uma abordagem ágil, se desejam ser relevantes e abrangentes na prática corporativa atual.

4.2 Design Centrado no Usuário

Conforme se pode concluir a partir da leitura da seção anterior, as metodologias ágeis representam, hoje, a maior parte da realidade das empresas de desenvolvimento de software. Ademais, os próprios métodos ágeis já preveem uma série de melhorias quando comparados aos modelos tradicionais, como a divisão do fluxo de trabalho, entregas parciais e funcionais, e maior comunicação com os clientes e *stakeholders*, no sentido de melhor suprir suas necessidades e expectativas. Este último aspecto, em especial, ajuda a gerar produtos de software que sejam mais satisfatórios aos seus usuários finais.

No entanto, retornando ao público-alvo deste trabalho, não se deve esquecer que os indivíduos autistas têm necessidades e preferências muito específicas. Além disso, os recursos disponíveis acerca da acessibilidade TEA são atualmente bastante escassos. Neste sentido, as metodologias ágeis podem não ser suficientes para contemplar toda a complexidade de se projetar softwares com foco no autismo. Como alternativa para solucionar essa demanda, tem crescido o interesse em utilizar abordagens de design como aliadas ao PDS ágil.

Uma das abordagens a que se tem dado destaque nos últimos tempos é o *Design Centrado no Usuário* (DCU). Esta abordagem (do inglês, *User-Centered Design*) inclui uma variedade de métodos e ferramentas que têm como finalidade colocar o usuário no centro do projeto de design. Existem níveis diversos de interação, mas a ideia geral é que o usuário seja efetivamente envolvido no processo em algum momento [1].

Conforme indicam as autoras, o nível de envolvimento pode se dar em diferentes intensidades. Os usuários podem ser envolvidos apenas em etapas cruciais do projeto, como o levantamento de requisitos e/ou os testes de usabilidade. Ou então podem estar presentes durante todo o processo, desde a idealização do projeto até a entrega final. Ainda, existem linhas de pensamento nas quais os usuários são agentes participativos na proposta de soluções e na tomada de decisões.

Embora o DCU possa ser aplicado a qualquer projeto que envolva o design, esta abordagem tem sido aplicada, com resultados bastante positivos, no desenvolvimento de software, inclusive como aliado às metodologias ágeis. Segundo Göransson, Gulliksen e Boivie [28], o problema da Engenharia de Usabilidade é que, tradicionalmente, esta prática pauta sua atuação na análise e avaliação da usabilidade por meio de métricas rígidas, por vezes

negligenciando o processo de design. Neste sentido, o Design Centrado no Usuário representa uma revolução de pensamento, já que, neste caso, a usabilidade é fruto da interação e do envolvimento com os usuários, tornando o processo de design mais dinâmico e fluído.

Para compreender melhor como se dá a integração do DCU nas metodologias ágeis, podemos utilizar o estudo conduzido por Caballero, Moreno e Seffah [14]. Os autores mencionam que estas abordagens são propícias a integrar-se, já que ambas têm como objetivo melhorar a usabilidade e qualidade do produto pela ênfase no usuário e pelo envolvimento do mesmo durante o processo. Além disso, ressalta-se a importância da organização da equipe, do feedback constante e da coparticipação na proposta de soluções.

Estes autores realizaram uma revisão sistemática e descobriram que o DCU é mais comumente aplicado em conjunto com o Scrum e o XP, e que as técnicas mais utilizadas são o desenvolvimento de protótipos, histórias de usuário, testes de usabilidade, criação de cenários, e geração de personas. Outras técnicas mencionadas, em menor número, incluem o uso de *index cards*, *storyboards*, testes de aceitação e avaliações heurísticas. Novamente, a maioria destas técnicas são usadas no Scrum e no XP.

Com relação à abordagem de desenvolvimento, constatou-se que o método mais comum é o *minimal up-front design*. Nesta técnica, o produto de software vai sendo construído com o mínimo possível de design na interface, possibilitando aos usuários e *stakeholders* efetuarem a validação dos componentes antes que haja um extenso trabalho de implementação. A segunda técnica mais utilizada é a realização das etapas de maneira concorrente, ou seja, há a sobreposição do design com a codificação.

Já em relação à composição da equipe, destaca-se a organização de times independentes, mas que mantêm um esquema de colaboração bastante fortalecido com os usuários e *stakeholders*. Em seguida, aparecem os times multidisciplinares, compostos por profissionais de diversas áreas, sobretudo aquelas que tenham relação direta com o escopo do projeto. Por fim, em menor número, estão as equipes independentes e que não possuem vínculo de colaboração com os usuários. Esta prática é desencorajada, porque descaracteriza os princípios fundamentais tanto do DCU quanto do desenvolvimento ágil.

Desta forma, claro está que o desenvolvimento ágil tem grandes chances de se beneficiar do Design Centrado no Usuário. Estas abordagens não são apenas similares quanto ao enfoque e seus objetivos. Mais que isso, elas são complementares, à medida que as técnicas

presentes auxiliam um processo mais dinâmico, promovendo a organização da equipe e a comunicação com os clientes e usuários. Similarmente, o alto nível de usabilidade e qualidade conferido pelo DCU-ágil pode ser aplicado com êxito em projetos concernindo o desenvolvimento de software ao público autista.

4.3 Gestão da Qualidade

A partir das explicações fornecidas nas seções anteriores, verifica-se a importância da seleção adequada de modelos e abordagens e em como eles impactam na qualidade do produto de software. Entretanto, o processo de desenvolvimento dificilmente será capaz de se auto-gerenciar a ponto de se tornar autossustentável. Diversas áreas do conhecimento relacionadas ao desenvolvimento de projetos (como a Administração, o Marketing, e a própria Engenharia de Software) nos informam sobre a importância de se realizar a gestão da qualidade em projetos de qualquer natureza.

Com os projetos de desenvolvimento de software não é diferente. Nesse sentido, necessitam-se de estratégias e ferramentas que propiciem um monitoramento e forneçam controle sobre as ações executadas. Na gestão de projetos de software, duas das ferramentas mais populares e eficazes, utilizadas por profissionais do mundo todo, são os Modelos de Maturidade (MMs) e as Lições Aprendidas (LAs).

É mais fácil explicar o funcionamento dos modelos de maturidade com um exemplo em mãos e, por isso, tomaremos o CMM (*Capability Maturity Model*), desenvolvido pelo *Software Engineering Institute*. Um modelo de maturidade é uma ferramenta gerencial que analisa determinadas esferas organizacionais de acordo com níveis de maturidade. No caso do CMM, são cinco níveis: (1) inicial, (2) repetitivo, (3) definido, (4) gerenciado quantitativamente, e (5) em otimização.

Este modelo define 18 processos fundamentais e 52 objetivos a serem cumpridos em termos de qualidade processual. Toda organização que deseje implementar o CMM precisa fazer uma auto-avaliação para ver em qual nível se encontra atualmente. Conforme os objetivos vão sendo cumpridos, a organização passa para níveis mais altos [59].

Segundo Paulk et al [59], os principais objetivos a serem alcançados com o uso do CMM é atingir níveis elevados de sofisticação processual em relação à consistência,

previsibilidade e confiabilidade. Um dos aspectos mais interessantes do CMM, e que também está presente em muitos outros modelos de maturidade, é a consciência de que, ao chegar ao nível mais alto, a organização não pode abandonar o gerenciamento da qualidade de seus processos. De fato, no quinto nível do CMM, existe a preocupação de que haja constante monitoramento e melhoria contínua.

Os modelos de maturidade tornaram-se ferramentas importantíssimas no gerenciamento de projetos de software e, com o tempo, foram se aperfeiçoando e diversificando sua área de atuação. Atualmente, um dos grandes interesses científicos tem sido o desenvolvimento de Modelos de Maturidade de Usabilidade (MMUs). Nos estudos de Jokela, et al [36] e Lacerda e Wangenheim [47], por exemplo, é possível encontrar uma grande variedade de MMUs disponíveis na literatura. Ambos os trabalhos mencionados realizaram revisões sistemáticas, desvendando um número considerável de modelos.

Além dos Modelos de Maturidade, outro recurso valioso na gestão da qualidade são as Lições Aprendidas (LAs). Conforme explicam Al-Mudimigh, Ullah e Alsubaie [2], as Lições Aprendidas são, por definição, recursos que têm a intenção de descrever um tópico de conhecimento específico, ou qualquer tipo de conhecimento adquirido por meio da experiência. A equipe deve ser responsável por documentar Lições Aprendidas sobre o andamento do projeto e os resultados obtidos com as atividades executadas. Vale ressaltar que as LAs cabem tanto para resultados positivos quanto para problemas, obstáculos ou erros identificados durante a fase de operação.

De forma geral, as LAs devem dar apoio à tomada de decisões por meio da indicação dos resultados esperados em contraste com o que de fato ocorreu, bem como problemas enfrentados durante o processo. Nesse sentido, as Lições Aprendidas precisam ser não apenas devidamente documentadas, mas também compartilhadas com todos, a fim de que se tornem referencial de consulta. Quando a disseminação das LAs é efetiva, elas têm o poder de influenciar positivamente o processo, evitando erros e prevenindo riscos.

A utilização desta estratégia, no entanto, enfrenta alguns obstáculos. Para Al-Mudimigh, Ullah e Alsubaie [2], existem cinco pontos que merecem atenção na sua implantação: (1) conscientizar sobre a importância das LAs, (2) reunir e documentar as experiências, (3) analisar as ações que resultaram em sucesso ou fracasso, (4) compartilhar o conhecimento adquirido, e (5) manter os registros atualizados. Quando esses pontos funcionam em conjunto, as

Lições Aprendidas têm a capacidade de criar um ambiente permanente de transferência de conhecimento.

Evidentemente, o gestor tem papel fundamental para que as LAs sejam implementadas e bem-sucedidas, pois é ele que suscitará, na equipe, a necessidade e o desejo de que isso aconteça. Por outro lado, deve haver um processo bem definido de implementação das Lições, integrado ao fluxo de trabalho, para que o hábito do registro ocorra de forma natural. As Lições Aprendidas podem ser uma ferramenta valiosa no processo de desenvolvimento de software, sobretudo de software acessível, visto que os requisitos tendem a ser mais complexos e mutáveis.

5 REVISÃO SISTEMÁTICA

Uma vez abordados os tópicos de interesse para esta pesquisa, passa-se agora para a etapa de desenvolvimento do *framework* propriamente dito. Este capítulo apresenta uma revisão sistemática realizada acerca de softwares educacionais para o público autista. Revisões sistemáticas a respeito do tema existem na literatura, como, por exemplo, os estudos de Fletcher-Watson [23], Ramdoss et al [62], e Wass e Porayska-Pomsta [73]. Tais revisões, no entanto, não contemplam o ciclo de desenvolvimento do começo ao fim e, por este motivo, julgou-se necessária a realização de um estudo mais abrangente e mais atualizado, com a intenção de se identificar fraquezas e possibilidades de melhoria no processo.

A revisão sistemática apresentada neste capítulo tem como objetivo principal a busca, identificação, análise e discussão acerca de softwares educacionais voltados ao público TEA, com especial ênfase no ciclo de desenvolvimento de software. Nesta revisão, além das etapas principais de desenvolvimento (levantamento de requisitos, arquitetura do sistema, design da Interface do Usuário – IU, testes e resultados), incluem-se também os objetivos de aprendizagem do software, o público-alvo, e informações sobre o PDS e a composição da equipe, sempre que estiverem disponíveis.

Para a realização da revisão sistemática, utilizou-se a metodologia descrita por Kitchenham [40]. Em primeiro lugar, desenvolve-se o protocolo de pesquisa, no qual constam as questões de pesquisa, os critérios de inclusão e exclusão, os termos de busca empregados e as diretrizes gerais de busca. Em seguida, ocorre a condução da busca por estudos primários e a extração de dados das fontes. Finalmente, analisam-se os resultados obtidos e discorre-se sobre as reflexões que são fruto desta revisão.

5.1 Protocolo de Pesquisa

Inicialmente, realizaram-se pesquisas preliminares que ajudaram a delimitar o escopo da revisão. O protocolo de pesquisa foi desenvolvido para fornecer diretrizes ao processo de revisão, a fim de possibilitar a identificação de fontes primárias relevantes de conteúdo e favorecer o processo de extração de dados ao mesmo tempo. As questões de pesquisa são:

- Quais são os objetivos de aprendizagem?
- Quem é o público-alvo?
- Qual é a metodologia / abordagem / processo de desenvolvimento empregado?
- Como se deu o levantamento de requisitos?
- Quais técnicas de aprendizagem foram utilizadas?
- Que métodos de representação foram aplicado à IU?
- Como o software foi testado / validado?
- Os resultados encontrados são positivos, negativos ou mistos?

A seguir, uma lista de palavras-chave foi definida a partir das questões de pesquisa. As palavras-chave deveriam contemplar três tópicos de importância para a busca: (1) o público-alvo, ou seja, indivíduos autistas; (2) o objeto de estudo – desenvolvimento de software acessível; e (3) o objetivo do software – sistemas de intervenção educacional. As palavras-chave foram organizadas usando-se *Booleans* ANDs e ORs. Os termos foram definidos em inglês, por propiciarem uma busca mais abrangente neste idioma. A *search string* resultante foi:

(“*Autism spectrum disorder*” OR “*autistic individuals*” OR “*autism*” OR “*high-functioning autism*”) AND (“*computer-assisted learning*” OR “*computer-assisted intervention*” OR “*computer-enhanced learning*” OR “*therapy software*”) AND (“*software development*” OR “*agile development*” OR “*development process*”). Em sequência, foi estabelecido um conjunto de critérios de inclusão e exclusão. Para ser incluído na revisão, o artigo avaliado deveria contemplar tanto os critérios de inclusão quanto os de exclusão:

- Critérios de inclusão:
 - Descrever uma contribuição inovadora acerca de softwares educacionais para indivíduos TEA;
 - Fornecer informações relevantes sobre os métodos, técnicas ou procedimentos utilizados no processo de desenvolvimento;
 - Ter sido testado ou avaliado por usuários finais ou *stakeholders*; e
 - Apresentar, ao menos, resultados preliminares.

- Critérios de exclusão:
 - Trabalhos secundários;
 - Artigos cujo objetivo principal não seja o desenvolvimento de software educacional para indivíduos autistas (relatórios clínicos ou artigos da área da saúde);
 - Artigos sobre robótica, intervenções de hardware, ou dispositivos sem interface gráfica;
 - Artigos sobre um mesmo sistema (apenas a publicação mais recente foi considerada); e
 - Artigos publicados antes de 2008 (amostragem de 10 anos).

5.2 Condução da Busca

Para a condução da busca, foram selecionadas seis fontes principais: (1) IEEE *Xplore Digital Library*, (2) *ACM Digital Library*, (3) *SpringerLink*, (4) SCOPUS, (5) *Science Direct*, e (6) *Google Scholar*. Tais bases de dados foram escolhidas por publicarem trabalhos relevantes na área da Engenharia de Software. Ademais, periódicos importantes referentes ao autismo podem ser acessados por meio delas, como o *Journal of Autism and Developmental Disorders*, da Springer, e o *Autism*, da SCOPUS.

A busca foi realizada utilizando-se a *search string* descrita previamente. Múltiplas tentativas foram feitas, variando-se os termos de busca, a fim de identificar o maior número possível de fontes primárias. Os artigos encontrados foram baixados, e excluíram-se arquivos duplicados. Após isso, observou-se os títulos e os resumos dos trabalhos para verificar sua relevância em relação ao tema de estudo. Caso restassem dúvidas após este filtro, eram analisadas também as seções de introdução e conclusão. Este processo resultou em 127 artigos considerados importantes para uma avaliação mais profunda.

Os artigos remanescentes foram, então, submetidos aos critérios de inclusão e exclusão. Aqueles que não atendessem satisfatoriamente os critérios eram retirados da lista. Após este procedimento, restaram 30 artigos elegíveis para inclusão na revisão. Cada um dos 30 artigos selecionados foi cuidadosamente analisado, em busca de informações que respondessem às questões de pesquisa.

Posteriormente, as informações extraídas foram sintetizadas em relatórios de inspeção de qualidade, contendo a identificação do estudo (nomes dos autores, título do trabalho, base de dados em que foi encontrado, data da busca e data da inspeção) e as informações qualitativas extraídas dele. Os tópicos para extração das informações, que compunham os relatórios, são os seguintes:

1. **Ano de publicação;**
2. **Nome do software;**
3. **Objetivos de aprendizagem:** quais os resultados esperados com o uso do software;
4. **Público-alvo:** qual o grupo TEA específico a que se destina o software;
5. **Processo de desenvolvimento de software:** informações sobre a metodologia utilizada (métodos ágeis ou tradicionais, por exemplo);
6. **Organização de equipe:** a composição do time de desenvolvimento, e se houve o envolvimento de usuário finais ou *stakeholders*, bem como qual o nível de integração;
7. **Levantamento de requisitos:** métodos e técnicas empregados para a etapa de levantamento de requisitos;
8. **Técnicas de aprendizagem:** tecnologias ou estratégias usadas para aumentar a acessibilidade e melhorar a experiência de aprendizagem do indivíduo TEA;
9. **Design da Interface do Usuário:** elementos visuais ou métodos de representação aplicados na IU;
10. **Testes:** procedimentos de validação e amostragem;
11. **Resultados:** descrição dos resultados obtidos nos trabalhos, sejam positivos, negativos ou mistos; e
12. **Comentários gerais.**

As informações extraídas foram tabuladas e transformadas em gráficos para melhor visualização dos dados. Um vez terminada a extração, passou-se para a fase de análise das informações. Alguns trabalhos possuíam um detalhamento maior dos procedimentos utilizados, enquanto que em outros o nível de especificação era menor. Em alguns casos, não foi possível identificar certos aspectos do desenvolvimento do trabalho, e por isso alguns campos ficaram em branco.

5.3 Resultados e Discussão

Em primeiro lugar, é importante contextualizar acerca do ambiente de desenvolvimento que os artigos descrevem, bem como os objetivos de aprendizagem propostos por eles. A Tabela 1, abaixo, apresenta a referência do artigo (para facilitar o acompanhamento), seguido da autoria e ano de publicação. Em seguida, aparece(m) o(s) objetivo(s) de aprendizagem e as informações sobre o PDS empregado ou a composição da equipe de desenvolvimento.

Tabela 1. Objetivos de aprendizagem e PDS.

ID	Autor(es) - ano	Objetivo(s) de aprendizagem	Informações do PDS
[3]	Alvarado et al (2017)	Comunicação	DCU, equipe multidisciplinar
[8]	Barros et al (2014)	Atividades diárias	Equipe multidisciplinar, ágil
[15]	Cabiellles-Hernández et al (2016)	Comunicação	Iterativo
[16]	Cai et al (2013)	Múltiplos	Equipe multidisciplinar
[17]	Chatzara et al (2014)	Interação social	Iterativo, ágil
[19]	De Leo et al (2011)	Comunicação	DCU, equipe multidisciplinar
[22]	Finkelstein et al (2010)	Habilidades motoras	
[24]	Fotjik (2010)	Comunicação	<i>Extreme Programming</i>
[29]	Hayes et al (2010)	Comunicação, atividades diárias	Design participativo, equipe multidisciplinar
[31]	Hourcade, Bullock-Rest e Hansen (2012)	Múltiplos	Design participativo, equipe multidisciplinar
[32]	Hulusic e Pistoljevic (2014)	Educação básica	Equipe participativa
[34]	Iradah e Rabiah (2011)	Educação básica	Equipe multidisciplinar
[37]	Kamaruzaman et al (2016)	Educação básica	Design participativo, equipe multidisciplinar
[43]	Konstantinidis et al (2009)	Múltiplos	Iterativo
[46]	Kurniawan et al (2018)	Atividades diárias	Abordagem ADDIE, equipe multidisciplinar
[50]	Malinverni et al (2017)	Múltiplos	Design participativo, equipe multidisciplinar
[65]	Shminan et al (2017)	Comunicação	Iterativo, ágil
[66]	Signore, Balasi e Yuan (2014)	Comunicação	DCU, equipe multidisciplinar
[67]	Silva, Fernandes e Grohmann (2014)	Comunicação	<i>Applied Behavior Analysis</i> , equipe participativa
[68]	Tan, Harrold e Rosser (2013)	Reconhecimento facial	Iterativo

[69]	Tsiopela e Jimoyiannis (2016)	Interação social	
[70]	Venkatesh et al (2013)	Múltiplos	Equipe participativa
[71]	Wadhwa e Jianxiong (2013)	Comunicação	Design participativo, equipe multidisciplinar
[72]	Washington et al (2017)	Reconhecimento facial	Iterativo, equipe participativa
[75]	Winoto, Cao e Tang (2017)	Comunicação	DCU, equipe multidisciplinar
[76]	Winoto, Xu e Zhu (2016)	Comunicação	Equipe multidisciplinar
[77]	Wojciechowski e Al-Musawi (2017)	Comunicação, interação social	Equipe participativa
[79]	Yan (2011)	Múltiplos	Equipe participativa
[80]	Zaffke et al (2014)	Comunicação	Iterativo
[81]	Zhang et al (2018)	Comunicação, interação social	Iterativo

Conforme apresentado na Tabela 1 (acima) e na Figura 2 (abaixo), os artigos analisados demonstram diversos objetivos de aprendizagem. Quase metade deles (46,6%) focam na melhoria de habilidades de comunicação, quase sempre por meio do ensino de novo vocabulário ou da expansão dos conhecimentos já existentes. Outros objetivos encontrados incluem a interação social (13,3%), educação básica (10%) e atividades diárias (10%). Além disso, dois estudos [68, 72] focaram no reconhecimento facial e apenas um [22] no refinamento de habilidades motoras. Ainda, seis artigos (20%) apresentaram múltiplos objetivos, ou seja, uma combinação de dois ou mais dos objetivos listados.

Com relação ao processo de desenvolvimento de software, nem sempre as informações estavam suficientemente detalhadas, mas na maioria dos casos foi possível extrair dados relevantes. Abordagens citadas com frequência são o Design Participativo e o Design Centrado no Usuário, técnicas que envolvem usuários finais e/ou *stakeholders* no processo de desenvolvimento. Um dos estudos citou a abordagem ADDIE (do inglês, *Analysis, Design, Development, Implementation, Evaluation*) e outro mencionou o *Extreme Programming*. Diversos estudos apontaram o uso de uma abordagem “iterativa” ou “ágil”, embora não tenham oferecido explicações detalhadas.

A organização das equipes de desenvolvimento foi participativa ou multidisciplinar, na maioria dos casos. Equipes multidisciplinares são aquelas compostas pela equipe de desenvolvimento e por *stakeholders*, que são envolvidos durante todo o ciclo de vida do processo. Em contraste, equipes participativas envolvem os *stakeholders* apenas em etapas

específicas do desenvolvimento, como o levantamento de requisitos ou a fase de testes. Em ambos os casos, fica claro o esforço em se entender e atender às preferências e necessidades do público-alvo.

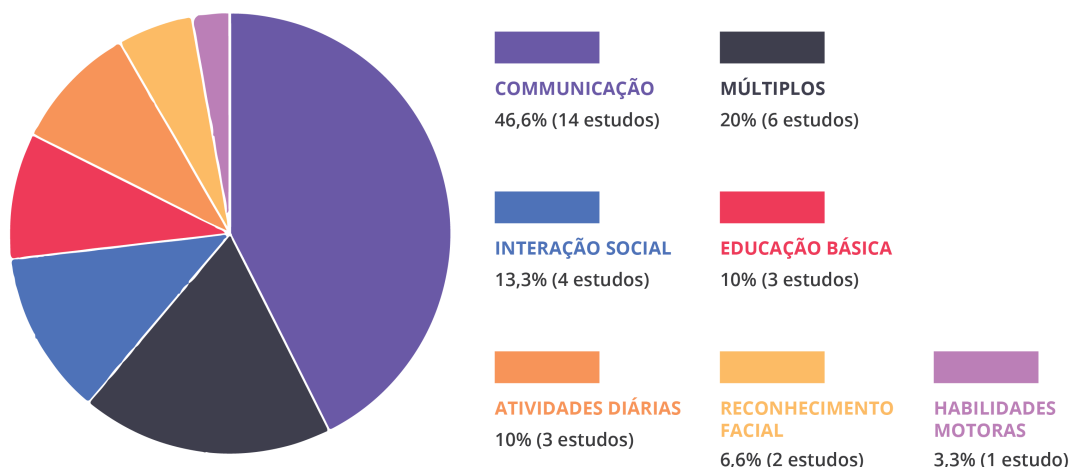


Figura 2. Objetivos de aprendizagem extraídos dos artigos. Fonte: elaborado pelo autor.

Com relação ao público-alvo, a Figura 3 resume as descobertas feitas. A grande maioria dos estudos analisados (76,6%) refere-se ao seu público-alvo simplesmente como “crianças autistas” ou alguma variação do termo. Apenas quatro estudos indicam um público específico (autismo de alto-funcionamento, baixo-funcionamento, ou ambos). Dois estudos referem-se ao público como “indivíduos autistas”, levando à compreensão de que o software proposto pode ser utilizado por diversas faixas etárias. Apenas um estudo aplica-se a “distúrbios específicos” – autismo, síndrome de Down, e deficiência intelectual, por exemplo [24].

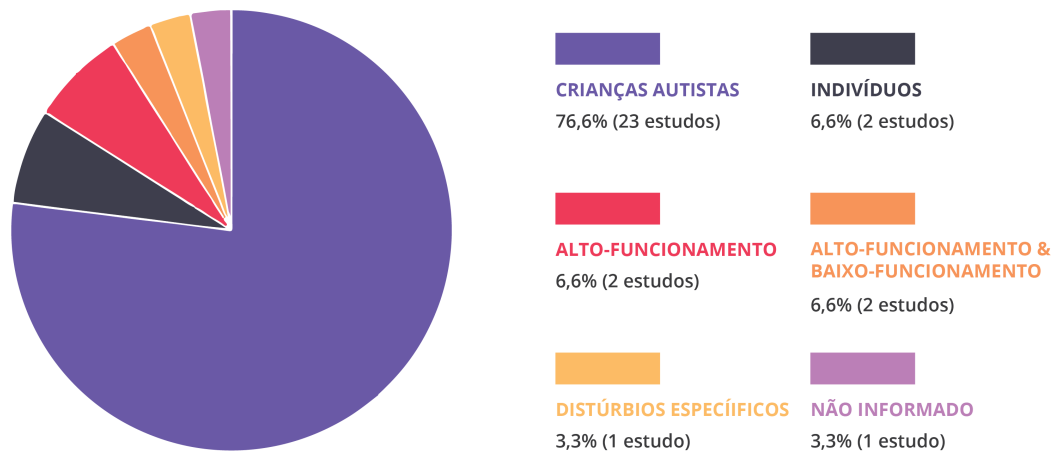


Figura 3. Públicos-alvo extraídos dos artigos. Fonte: elaborado pelo autor.

Em seguida, foram analisados os aspectos relativos ao ciclo de vida de desenvolvimento propriamente dito. A Tabela 2 reúne informações acerca dos métodos utilizados para a realização do levantamento de requisitos, e também as técnicas de aprendizagem. As técnicas de aprendizagem compreendem o uso de tecnologias e suportes empregados para possibilitar o cumprimento dos objetivos de aprendizagem especificados anteriormente.

Tabela 2. Levantamento de requisitos e técnicas de aprendizagem

ID	Levantamento de requisitos							Técnicas de aprendizagem									
	Entrevistas	Questionários	Observações em campo	<i>Demo Sessions</i>	Grupos focais	<i>Proxy users</i>	Outros	Feedback	Recompensas	Realidade Virtual	Gamificação	Personalização	Multimídia	<i>Storytelling</i>	Monitoramento	Customização de conteúdos	Outros
[3]			✓	✓				✓			✓						
[8]				✓	✓			✓	✓			✓			✓	✓	
[15]								✓			✓				✓	✓	
[16]			✓						✓	✓	✓		✓				
[17]		✓						✓				✓		✓	✓	✓	✓
[19]				✓							✓	✓			✓	✓	
[22]				✓				✓	✓	✓	✓						

[24]				✓		✓	✓			✓	✓
[29]	✓		✓		✓			✓		✓	✓
[31]			✓			✓			✓		
[32]	✓				✓			✓			
[34]				✓			✓	✓		✓	
[37]				✓					✓	✓	✓
[43]	✓								✓	✓	✓
[46]			✓				✓	✓		✓	
[50]	✓			✓	✓			✓			
[65]	✓									✓	✓
[66]	✓				✓			✓	✓		
[67]		✓			✓			✓		✓	✓
[68]			✓			✓			✓		
[69]							✓		✓		✓
[70]			✓				✓			✓	✓
[71]	✓	✓	✓					✓	✓	✓	
[72]	✓						✓	✓	✓		✓
[75]				✓		✓		✓		✓	✓
[76]		✓					✓	✓		✓	
[77]								✓	✓		✓
[79]	✓		✓				✓	✓			
[80]					✓			✓		✓	✓
[81]					✓		✓		✓		✓

Para a etapa de levantamento de requisitos, técnicas tradicionais bastante utilizadas são as entrevistas (30%) e os questionários (13%), além das observações de campo (26,6%). As entrevistas e questionários foram aplicados, em sua maioria, com pais ou professores, enquanto que as observações de campo tinham a intenção de analisar o comportamento das próprias crianças autistas. Um dos estudos também informou sobre a utilização de *proxy users* [75]. Neste caso, os professores foram responsáveis por selecionar os objetivos de aprendizagem, enquanto que os pais ajudaram a desenvolver o conteúdo.

Também foram utilizadas *demo sessions* e grupos focais, com 23,3% de predominância cada. *Demo sessions* são sessões de demonstração em que se define o funcionamento geral do sistema. Grupo focais, por sua vez, contam com a participação ativa dos *stakeholders* para levantar requisitos e desenvolver soluções de uso do software. Outras técnicas mencionadas, com menor frequência, foram o uso de diários em vídeo, design instrucional, observação participante e sessões de *brainstorming*.

Já no tocante às técnicas de aprendizagem, a gama de tecnologias e recursos é ainda mais ampla. Um grande número de artigos apresentou a utilização de feedback (50%) ou sistemas de recompensa (33%). Estas técnicas são empregadas para encorajar comportamentos positivos nas crianças autistas. No caso do feedback, podem ser estímulos visuais ou sonoros emitidos pela interface. As recompensas, ao contrário, geralmente são estímulos reais: uma fruta, um jogo, ou qualquer coisa que a criança associe a ter concluído uma tarefa com sucesso.

A gamificação também é uma técnica popular entre os estudos analisados (60%). Ela tem como finalidade incluir elementos de *games* nas interfaces, como personagens virtuais, seleção do nível de dificuldade, *multiplayer*, ou outros recursos que darão a sensação de que o usuário está jogando, ao invés de colocá-lo para repetir atividades tediosas. Pode ser aplicada em conjunto com sistemas de Realidade Virtual e uma abordagem multimídia, pois a natureza lúdica da gamificação é receptiva à utilização de recursos variados.

Além disso, foi identificada uma grande predominância no emprego de técnicas de personalização e/ou controle parental. No caso da personalização, trata-se da seleção dos componentes gráficos, como ilustrações, cores, fontes, formas, etc. Também é possível customizar o conteúdo a ser trabalhado, ajustando os níveis de dificuldade ao ritmo de cada aluno. Estas técnicas são empregadas pensando-se na variedade de condições clínicas do público autista, a fim de se respeitar as habilidades de cada um. Adicionalmente, vários softwares estudados aplicaram um sistema de monitoramento, por meio da sincronização dos dispositivos da criança com seus pais, professores ou cuidadores.

Outra técnica mencionada é o *storytelling*, que encoraja a criação de narrativas próprias por meio de suportes de mídia variados (texto, imagens, sons, animação e vídeo). Desta maneira, pode-se promover a autonomia das crianças, além de incentivar seu desenvolvimento simbólico de comunicação. Na coluna “outros”, temos o uso de familiarização (uso de imagens ou sons do cotidiano da criança) [72, 69], compartilhamento de informação [81], avatares virtuais [43], experimentação ativa [17], e controle remoto [80].

Tabela 3. Design da Interface do Usuário, testes e resultados

ID	Design da IU							Testes							Resultados
	Ilustração	Fotografias	Palavras / Texto	Seleção de cores	Sons	Voz	Vídeo / RV	Prototipação	Teste de usabilidade	Avaliação heurística	Estudo de caso	Questionários	Demo Sessions	Outros	
[3]	✓		✓	✓	✓				✓						Positivos
[8]	✓	✓	✓	✓	✓							✓	✓		Mistos
[15]	✓	✓	✓	✓	✓		✓		✓			✓			Positivos
[16]					✓		✓	✓	✓			✓			Mistos
[17]	✓	✓	✓		✓				✓			✓			Positivos
[19]	✓	✓	✓	✓							✓				Mistos
[22]			✓	✓	✓		✓		✓			✓			Positivos
[24]	✓		✓	✓					✓				✓	✓	Mistos
[29]	✓	✓	✓			✓			✓				✓		Positivos
[31]	✓		✓	✓	✓						✓		✓		Positivos
[32]	✓			✓	✓				✓						Positivos
[34]	✓		✓	✓				✓	✓						Mistos
[37]	✓		✓	✓				✓	✓						Mistos
[43]	✓		✓	✓	✓				✓			✓	✓		Positivos
[46]	✓		✓	✓			✓		✓				✓		Positivos
[50]			✓	✓	✓		✓	✓			✓				Positivos
[65]	✓	✓	✓					✓	✓			✓			Positivos
[66]	✓		✓	✓				✓	✓			✓			Positivos
[67]	✓		✓		✓		✓		✓						Mistos
[68]		✓		✓	✓	✓	✓		✓			✓			Positivos
[69]	✓			✓	✓		✓		✓						Positivos
[70]	✓		✓				✓		✓						Mistos
[71]		✓	✓	✓					✓	✓					Mistos
[72]	✓		✓	✓	✓		✓		✓					✓	Mistos
[75]		✓	✓			✓			✓				✓		Mistos
[76]				✓	✓		✓		✓						Positivos
[77]	✓	✓	✓	✓					✓			✓			Mistos
[79]	✓		✓	✓	✓				✓				✓		Mistos
[80]	✓		✓		✓							✓		✓	Positivos
[81]	✓			✓								✓		✓	Positivos

Prosseguindo com a análise dos resultados, a Tabela 3 demonstra os métodos de representação gráfica utilizados no design da IU, os procedimentos de teste e validação empregados e os resultados atingidos com o uso do software. Nota-se o uso quase unânime de ilustrações e textos / palavras para transmitir significados e informações. O estilo de ilustrações é bastante variado, desde formas geométricas básicas até ilustrações mais estilizadas. O uso de ilustrações realistas não foi identificado nos estudos. Nos casos em que os pesquisadores desejavam uma abordagem mais realística, optou-se por usar fotografias, pois elas poderiam ser mais facilmente reconhecidas pelas crianças autistas.

A seleção da paleta de cores também aparece como uma grande preocupação no momento das decisões de design. Novamente, as abordagens são mistas: alguns trabalhos indicam o uso de tons pastéis, enquanto outros aplicam uma paleta de cores vibrantes à interface. Não há, contudo, indicativo de uma reavaliação das cores baseado no feedback dos usuários. Muitos estudos (56,6%) também utilizaram sons na construção do software. “Sons”, neste caso, refere-se à trilha sonora ou efeitos próprios da interface, e não o uso de gravações de voz, que é bem mais escasso. Por fim, vários estudos fizeram, ainda, uso de vídeo, em forma de pré-gravações ou animações, e mesmo associados a um sistema de Realidade Virtual.

Os testes mais populares são os de usabilidade (83,3%). Segundo as fontes, tais testes foram aplicados com perfis variados de *stakeholders*, como professores, terapeutas, pais, cuidadores, ou mesmo os próprios indivíduos autistas, dependendo da disponibilidade de recursos humanos. Também se identificou a utilização de questionários, estudos de caso, avaliações heurísticas e, novamente, *demo sessions* (que, desta vez, tinham a intenção de validar aspectos do sistema). Menciona-se, ainda, a técnica da prototipação, que ocorre ao final de cada ciclo de iteração e antes do início do ciclo seguinte, e que permite aos *stakeholders* validarem partes do software antes que sejam implementadas.

Outras técnicas encontradas foram: avaliações de longa duração, conduzidas no ambiente familiar dos pacientes [72]; uso de um questionário virtual embutido no próprio sistema [24]; e testes automáticos [81], embora a aplicação desta abordagem pareça ser bem rara. Mais da metade dos estudos analisados (56,6%) indicou resultados positivos após os procedimentos de teste com os respectivos softwares. A outra parcela (43,3%), apresentou resultados mistos, ou seja, em parte positivos e em parte negativos. Resultados exclusivamente negativos não foram reportados por nenhum dos artigos analisados.

5.4 Considerações Parciais

A partir da revisão realizada e da análise dos dados extraídos, é possível elencar algumas considerações importantes referentes ao processo de desenvolvimento de software acessível voltado ao público TEA. Tais reflexões, apresentadas abaixo, apontam práticas recorrentes no processo e também revelam algumas fragilidades:

1. Os objetivos de aprendizagem relacionam-se diretamente com os principais déficits enfrentados pelo público autista;
2. Não há consenso sobre a utilização de um PDS específico, embora as abordagens iterativas ou ágeis apareçam com mais frequência;
3. O envolvimento dos *stakeholders* é importante, e aparece em diversos momentos do processo, com variados graus de intensidade;
4. A definição do público-alvo geralmente é abrangente e genérica, sem um recorte clínico específico;
5. Todas as etapas de desenvolvimento apresentam ampla variedade de técnicas empregadas (muitas vezes a combinação de duas ou mais), de acordo com a metodologia utilizada e a disponibilidade de recursos humanos;
6. As técnicas de aprendizagem mais utilizadas incluem feedback (virtual ou real), aspectos de gamificação, personalização e monitoramento;
7. As representações gráficas na interface geralmente envolvem mais de um método (imagens, texto, cores, sons), a fim de contemplar os vários graus de severidade do transtorno;
8. Os procedimentos de teste em muitos casos são conduzidos com uma amostra pequena de usuários, o que diminui seu rigor técnico;
9. Os testes normalmente são aplicados sem distinção entre grupos específicos de indivíduos TEA (alto-funcionamento e baixo-funcionamento, por exemplo), e sem o uso de grupos de controle;
10. Os resultados obtidos, em sua maioria, são apresentados de forma sucinta e pouco detalhada, sem apontar direcionamentos futuros. Também não há acompanhamento do uso do software pós-entrega.

6 MODELAGEM DO FRAMEWORK

Com base nas reflexões resultantes da revisão sistemática, obtém-se os direcionamentos que servirão de sustentáculo à modelagem do *framework*. Neste capítulo, portanto, ocorre a descrição detalhada das etapas e atividades do processo, a fim de demonstrar seu funcionamento de forma clara e fornecer o aporte necessário para tal. Para a modelagem, utiliza-se o meta-processo de desenvolvimento proposto por Bolanos et al [11], que contempla 12 (doze) passos. A seguir, apresentam-se os quatro primeiros: definição da estratégia, organização do processo, planejamento de produção, e arquitetura do processo.

6.1 Definição da Estratégia

O *framework* proposto neste trabalho tem a intenção de ser uma estrutura de apoio ao processo de desenvolvimento ágil, pensado para que as organizações usem como base o processo e os conhecimentos que já possuem. Já existem modelos de desenvolvimento absolutamente populares e comprovadamente eficazes, como o Scrum, o *Extreme Programming* (XP), o *Rational Unified Process* (RUP), entre outros. Todavia, nenhum deles tem foco específico em acessibilidade TEA, pois foram projetados para o sustentar um desenvolvimento genérico.

Ademais, o desenvolvimento de software educacional para o público autista é uma demanda esporádica das equipes de desenvolvimento. Desta forma, o *framework* pretende ser um processo que se adapta ao modelo de desenvolvimento corrente de cada organização, que pode ser integrado para resolver um problema pontual, e descontinuado quando não for mais necessário. Para que este sistema funcione conforme o esperado, existem algumas premissas:

- O *framework* foi modelado para integrar-se a processos de desenvolvimento ágeis, visando contemplar a realidade da maior parte das empresas de desenvolvimento, já que estes modelos são os mais populares;
- Ele destina-se ao desenvolvimento de softwares educacionais, pois estes produtos são voltados especificamente para indivíduos TEA. Seu uso não faz

sentido em contextos nos quais se busca a acessibilidade universal, já que não se encarrega de outros tipos de deficiência;

- Para a implantação do *framework*, espera-se que a equipe já esteja familiarizada com seu próprio processo de desenvolvimento. O *framework* não explica *como adotar* um modelo, mas sim *como integrar* as etapas que o constituem no modelo já existente na organização;
- Recomenda-se que a equipe de desenvolvimento seja multidisciplinar ou, caso não seja possível, que possa comunicar-se com profissionais e especialistas de outras áreas para cumprir as tarefas necessárias.

6.2 Organização do Processo

Como o *framework* foi modelado para adaptar-se aos mais variados contextos de desenvolvimento, não existem exigências específicas quanto à formatação da equipe. O Scrum, por exemplo, emprega nomenclaturas que não estão presentes em outros modelos (*Scrum Master*, *Product Owner*, etc.). Desta forma, ater-se a esta ou àquela metodologia impediria a universalização do *framework*. A ideia é que as equipes de desenvolvimento possam adaptar os processos do *framework* a sua realidade, dependendo dos recursos humanos disponíveis e da organização que já possuem. No entanto, ele opera com a premissa da existência de alguns agentes:

- **Cliente(s)**, que contrata o serviço da equipe de desenvolvimento, perante remuneração ou não, e que não necessariamente possui conhecimento clínico acerca do autismo. Os clientes também devem ser envolvidos em testes de aceitação, e idealmente acompanham o processo de desenvolvimento de perto;
- **Gerente de projeto**, responsável pelo planejamento, execução e monitoramento das iterações, bem como a mediação de relacionamentos interpessoais. O gerente de projeto normalmente é o elo de comunicação entre os agentes envolvidos no processo;
- **Equipe de desenvolvimento**, podendo incluir arquitetos ou engenheiros de software, analistas, programadores, designers, etc. São os responsáveis por operacionalizar as atividades e tarefas do processo, ou seja, produzir o software;

- **Profissionais e especialistas**, que detêm conhecimento clínico acerca do autismo e fornecem *insights* sobre o desenvolvimento. Podem incluir professores, médicos, terapeutas, psicólogos e mesmo pessoas de convívio próximo ao indivíduo TEA, como pais e cuidadores. Embora sejam absolutamente indispensáveis na etapa de testes, também recomenda-se que sejam envolvidos em outros momentos, como o levantamento de requisitos;
- **Usuários finais**, que são os próprios indivíduos autistas. Também podem (e devem) ser incluídos durante todo o processo de desenvolvimento, especialmente nas etapas de testes.

Claro está que, embora a formação da equipe dependa do processo de desenvolvimento adotado por cada organização, recomenda-se fortemente que a equipe seja multidisciplinar. Mesmo quando isso não é possível, em função de limitações de recursos humanos, é necessário que haja comunicação constante com agentes externos, que contribuirão com o processo de desenvolvimento e podem garantir mais qualidade no produto final.

6.3 Planejamento de Produção

O planejamento de produção do *framework* está dividido em 8 fases que são comuns aos modelos de desenvolvimento ágil, conforme descrito por Fagundes [21]. Desta forma, o ele pode integrar-se ao modelo adotado, independentemente de qual seja. As nomenclaturas podem variar de acordo com a metodologia empregada, mas as principais etapas (e que também constituem as etapas do *framework*) são as seguintes:

1. **Levantamento de requisitos:** contempla a identificação do público, o levantamento e a análise de requisitos funcionais, não-funcionais e atributos. Normalmente inclui prototipação de baixa fidelidade;
2. **Planejamento:** momento em que será definido o cronograma de trabalho, a atribuição de papéis e responsabilidades, e a alocação de recursos;
3. **Arquitetura de software:** define-se a Experiência do Usuário (UX), as estratégias e técnicas de aprendizagem, e modela-se a arquitetura do sistema;

4. **Desenvolvimento:** em linhas gerais, contempla o design da Interface Gráfica do Usuário (IGU) e a codificação;
5. **Testes unitários:** inclui a execução de testes unitários, relativos à iteração corrente, e o planejamento das alterações necessárias. Deste ponto, outra iteração se inicia, enquanto for necessário;
6. **Implantação:** ao final da última iteração, ocorre a integração do sistema e a transferência do mesmo ao ambiente real de uso;
7. **Validação do sistema:** realizam-se os testes de aceitação pelo cliente, e as adequações finais, caso existam;
8. **Entrega:** ocorre a entrega do produto de software, com o processo de melhoria contínua e atualizações, sempre que for o caso.

Além das etapas principais que compõem o processo, o *framework* utiliza duas abordagens para garantir que a produção ocorra com qualidade e eficiência. Para auxiliar a equipe de produção, utiliza-se o Design Centrado no Usuário (DCU), com a aplicação de diretrizes, métodos e ferramentas que garantam um produto em conformidade com as especificidades necessárias ao trabalho com o público TEA.

Para a gestão da qualidade, emprega-se o Modelo de Maturidade de Usabilidade descrito por Mostafa [55] e as Lições Aprendidas. Mesmo que a organização já possua um plano de gestão da qualidade, é recomendado que se realize uma nova avaliação baseada no que o *framework* propõe, pois este Modelo de Maturidade é especificamente voltado ao DCU-ágil. Estas etapas e operações serão descritas em maiores detalhes na próxima seção.

6.4 Arquitetura do Processo

O processo do *framework* está baseado em 8 etapas (E), já mencionadas anteriormente, que consistem de diversas **atividades** (A). Estas, por sua vez, são divididas em *tarefas* (T). Conforme se pode observar na Figura 4, as etapas que vão desde o levantamento de requisitos, passando pelo planejamento, arquitetura de software e desenvolvimento, até chegar aos testes unitários compreendem o ciclo de iteração. Este ciclo pode ser repetido indefinidamente até que todos os requisitos do software tenham sido atendidos. Em seguida,

ocorrem as etapas de implantação, validação do sistema e entrega. O ciclo de iteração é governado pelo Design Centrado no Usuário, e a gestão de qualidade deve estar presente durante todo o processo, do início ao fim.

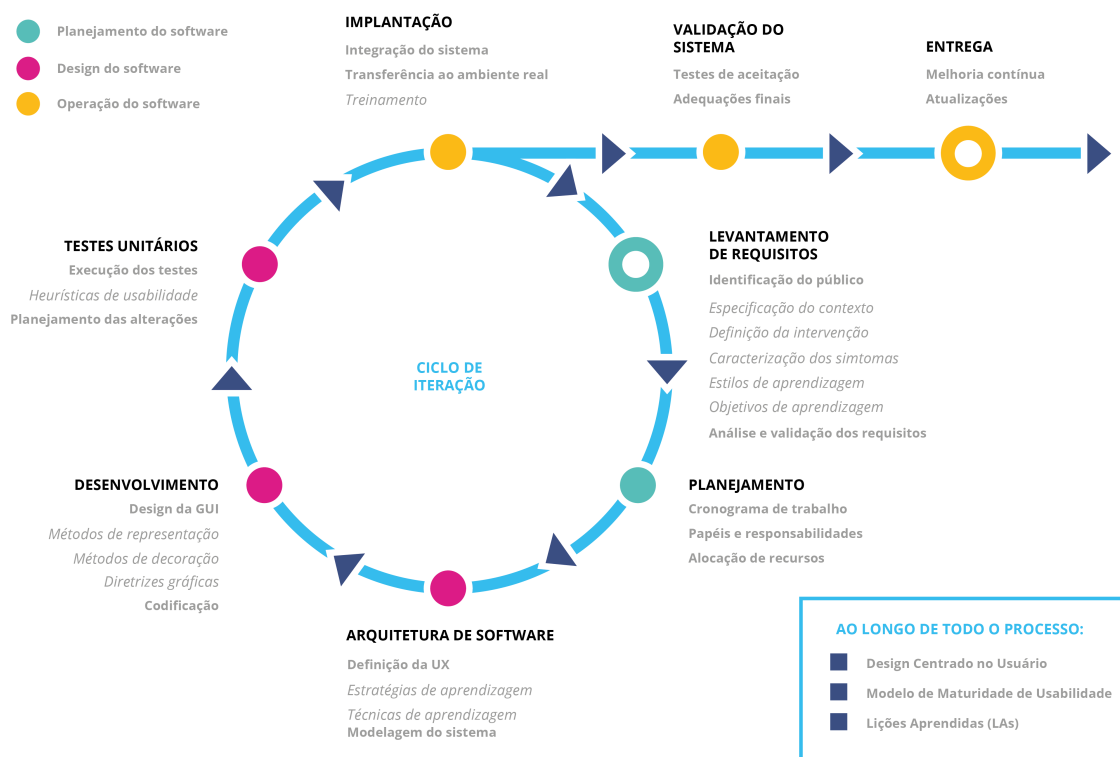


Figura 4. Ciclo de vida de desenvolvimento do framework. Fonte: elaborado pelo autor.

6.4.1 Levantamento de Requisitos

O levantamento de requisitos (E1) é a primeira etapa do processo e ocorre após o fechamento do contrato com o cliente, que solicita um produto de software à equipe de desenvolvimento. Nesta etapa, ocorre a especificação do que o software deve fazer e quais as condições mínimas de funcionamento, incluindo requisitos funcionais e não-funcionais e atributos de qualidade. A organização desta etapa é a seguinte:

(E1) Levantamento de requisitos

(A1) Identificação do público

(T1) Especificação do contexto

(T2) Definição da qualidade da intervenção

(T3) Caracterização dos sintomas

(T4) Seleção dos estilos de aprendizagem

(T5) Definição dos objetivos de aprendizagem

(A2) Análise de requisitos

(A3) Validação de requisitos

A primeira atividade desta etapa (A1) é a **identificação do público**. Esta atividade tem por objetivo levantar informações sobre o público-alvo a que se destina o software. Embora isso seja uma prática comum aos modelos ágeis, neste caso vale ressaltar que o público TEA é bastante diversificado em termos de diagnóstico, sintomas e graus de severidade, portanto se faz primordial um entendimento abrangente e aprofundado acerca do usuário final.

A primeira tarefa envolvida nesta atividade é a *especificação do contexto* (T1). A equipe de desenvolvimento deve realizar pesquisas preliminares acerca do contexto geral de desenvolvimento do software. Não se tratam, ainda, de especificações clínicas ou de intervenção terapêutica, mas sim relativas ao macro-ambiente em que se insere o público TEA. Alguns questionamentos a serem respondidos são:

- Existem legislações de acessibilidade específicas que afetam o desenvolvimento do projeto?
- Quais recomendações e/ou normatizações nacionais e internacionais devem ser consideradas durante o desenvolvimento?
- Quais aspectos culturais devem ser respeitados? Existem restrições?
- Quais considerações morais e éticas devem ser respeitadas ao se trabalhar com o indivíduo TEA?

A segunda tarefa é a *definição da qualidade da intervenção* (T2). Neste momento, descrevem-se as características da intervenção por meio do software, ou seja, a natureza da sua aplicação. “Qualidade”, neste caso, não se refere ao sentido tradicional da qualidade que é atribuída ao software, mas sim *como* ou *o que* o software deve fazer. Questiona-se:

- Qual será o ambiente de uso do software (casa, escola, clínica, etc.)?
- Quais serão os agentes envolvidos na manipulação do software (indivíduos TEA, pais, professores, terapeutas, etc.)?
- Quais serão os agentes mediadores do software (responsáveis pelo seu controle e/ou monitoramento, se for o caso)?
- Qual a duração e a frequência de uso esperada para o software?

Seguindo em frente, a terceira tarefa refere-se à *caracterização dos sintomas* (T3). Aprofundando-se nos aspectos clínicos do público, deve-se especificar quais os sintomas gerais dos indivíduos que se beneficiarão com o uso do software, lembrando sempre da heterogeneidade do público TEA. Para esta tarefa, recomenda-se a recuperação do histórico clínico dos indivíduos, ou o contato direto com o médico/terapeuta responsável. Ainda, sugere-se a aplicação de alguma escala diagnóstica. A escala sugerida pelo *framework* é a CARS (*Childhood Autism Rating Scale*), por ser uma ferramenta popular e comprovadamente eficaz, traduzida para o português por Pereira [61] (Anexo A). No entanto, ressalta-se que sua aplicação deve ser feita por profissionais capacitados e treinados para tal. Algumas questões que ajudam a refletir sobre esta tarefa são:

- Quais as dificuldades de comunicação verbal e não-verbal apresentadas pelos indivíduos?
- Quais as dificuldades de interação social e/ou adaptação aos contextos sociais?
- Quais estereótipos comportamentais esses indivíduos possuem? Existem outras limitações importantes a serem consideradas?
- Qual o grau de acometimento (severidade dos sintomas) encontrados no público?
- Qual a idade média dos indivíduos? Este é um aspecto que afeta o desenvolvimento do projeto?
- Quais restrições são impostas pelos sintomas encontrados?

Em seguida, ocorre a *seleção dos estilos de aprendizagem* (T4). Baseada nas informações levantadas até então, a equipe de desenvolvimento seleciona um ou mais estilos de

aprendizagem que nortearão o desenvolvimento do software. Os estilos de aprendizagem dizem respeito à forma com que os indivíduos aprendem, com base em suas preferências. Recomenda-se, para tanto, que a ferramenta *Learning Style Inventory 4.0* [42] seja aplicada com uma amostra da população de usuários finais.

As instruções para aplicação desta ferramenta e enquadramento dos estilos de aprendizagem estão contidas no Anexo B. Os autores indicam a existência de 9 estilos: (1) iniciativo, (2) experimental, (3) imaginativo, (4) ativo, (5) reflexivo, (6) decisivo, (7) pensativo, (8) analítico, e (9) equilibrado. A seleção de estilos de aprendizagem é importante para se mapear as preferências de aprendizagem dos indivíduos, ou seja, como eles aprendem melhor. Vejamos algumas das principais características de cada perfil:

1. **Iniciativo:** este perfil possui iniciativa para lidar com experiências e situações. Caracterizado pela habilidade de inicializar tarefas e procedimentos, envolvendo a experimentação ativa e a experiência concreta.
2. **Experimental:** este perfil encontra sentido e se envolve profundamente por meio da experiência. Busca experiências significativas enquanto participa ativamente e pratica a observação reflexiva.
3. **Imaginativo:** imagina possibilidades por meio da observação e da reflexão. Combina as habilidades da experimentação concreta e da observação reflexiva.
4. **Ativo:** possui forte motivação a cumprir objetivos e integrar pessoas e tarefas. Prefere a experimentação concreta e ativa.
5. **Reflexivo:** conecta experiências e ideias por meio da análise reflexiva. Equilibra as competências da experimentação concreta e da conceptualização abstrata.
6. **Decisivo:** usa teorias e modelos para tomar decisões e planejar ações. Combina a conceptualização abstrata e a experimentação ativa.
7. **Pensativo:** envolve-se com muita disciplina no raciocínio abstrato e lógico. Atinge a conceptualização abstrata por meio da experimentação ativa e da observação.

8. **Analítico:** integra ideias, modelos e sistemas pela reflexão. Combina a observação reflexiva e a conceptualização abstrata.
9. **Equilibrado:** adapta-se ao ambiente pesando os prós e os contras de cada ação, bem como as consequências do pensar e agir. É um perfil que equilibra habilidades de experimentação concreta e ativa, abstração e reflexão.

Algumas questões importantes são para nortear a execução desta tarefa, após a seleção dos estilos de aprendizagem, são:

- Quais estilos de aprendizagem devemos incorporar em nosso software? Quais estarão no foco de atenção durante o desenvolvimento?
- O software proposto será mais abrangente ou mais específico, em relação aos estilos de aprendizagem e aos sintomas detectados?
- Qual o impacto dos estilos de aprendizagem na nossa estratégia de desenvolvimento?
- Que cuidados devemos ter ao adotar determinados estilos de aprendizagem, levando em consideração a heterogeneidade do público?

Por fim, a última tarefa desta atividade é a *definição dos objetivos de aprendizagem* (T5). Aqui, a equipe de desenvolvimento reflete sobre aquilo que deseja atingir, em termos pedagógicos, com o desenvolvimento e aplicação do software. Objetivos comuns aos softwares educacionais para o público TEA geralmente envolvem o desenvolvimento de comunicação verbal, estímulo às interações sociais, organização de tarefas diárias e mesmo suporte à educação básica. Podem haver múltiplos objetivos, mas é importante ressaltar que isso aumenta a complexidade do processo. A equipe deve se perguntar:

- Quais nossos objetivos com o desenvolvimento do software? Quais benefícios queremos que os indivíduos alcancem com sua utilização?
- Quais sintomas serão o foco primário de atenção? Quais podem ser incorporados como objetivos secundários?

- Temos múltiplos objetivos pedagógicos? Possuímos capacidade técnica e conhecimento clínico suficiente para que todos sejam alcançados?

A realização da quinta tarefa encerra a atividade (A1) **identificação do público**.

A segunda atividade desta etapa é a **análise de requisitos** (A2). Esta atividade caracteriza-se pela depuração das informações levantadas acerca do usuário e em como tais informações serão transpostas em termos de requisitos de software. Podem-se utilizar ferramentas como criação de personas, cenários, histórias do usuário ou casos de uso, dependendo da abordagem adotada. O analista, então, documenta e especifica os requisitos funcionais, não-funcionais, e atributos de qualidade do software.

Por fim, ocorre a atividade (A3) **validação de requisitos**, que serve para verificar se os requisitos estabelecidos previamente correspondem às necessidades e expectativas dos *stakeholders*. A validação pode ocorrer de diversas formas, como em uma reunião geral da equipe. No entanto, segundo as práticas do design centrado no usuário, é recomendado que a validação aconteça por meio de protótipos, que viabilizam uma inspeção mais realista dos requisitos levantados. Ao final deste processo, gera-se um *relatório geral de intervenção*, contendo todas as informações levantadas nesta etapa (modelo disponível no Apêndice A).

6.4.2 Planejamento

Como o próprio nome sugere, nesta etapa ocorrerá o planejamento do desenvolvimento do software (E2), incluindo a elaboração do cronograma de trabalho, o fluxo de atividades, a distribuição de responsabilidades, a alocação ou disponibilização de recursos, entre outros. Nos modelos ágeis, esta etapa representa o planejamento dos ciclos de iteração, ou a entrega de incrementos, no caso do desenvolvimento incremental. No Scrum, por exemplo, ocorre a definição das *sprints* de desenvolvimento.

Este *framework* não prevê nenhuma atividade adicional além das que já são realizadas normalmente nos processos de desenvolvimento, mas recomenda-se que algumas instruções relativas ao Design Centrado no Usuário sejam observadas. É neste momento que se planejam os testes que serão realizados ao longo do percurso (unitários, de integração, de

aceitação, etc.), quais os agentes externos envolvidos no processo (educadores, terapeutas, pais, usuários finais, etc.), e como e em quais momentos se dará o seu envolvimento.

6.4.3 Arquitetura de Software

Na terceira etapa, comum aos métodos ágeis, ocorre a definição da arquitetura de software (E3). O software deverá ser modelado de acordo com suas funcionalidades e em consonância com os requisitos levantados previamente. O arquiteto de software estabelece os níveis e as camadas do software, define as relações entre os componentes e assegura-se de que a infraestrutura tecnológica será capaz de atender às demandas de desenvolvimento. As atividades e tarefas envolvidas são as seguintes:

(E3) Arquitetura de software

(A1) Definição da Experiência do Usuário

(T1) Seleção das estratégias de aprendizagem

(T2) Seleção das técnicas de aprendizagem

(A2) Modelagem do sistema

Inicialmente, ocorre a **definição da Experiência do Usuário (A1)**. É muito provável que, a esta altura, a equipe já tenha um esboço geral de como será o software. Entretanto, a arquitetura do software depende do entendimento da aplicação das funcionalidades que serão incorporadas a ele. Nesta atividade, definem-se as características da experiência do usuário, que pode ser definida como as percepções que o usuário terá ao manipular o software, como se espera que esse relacionamento ocorra, e quais as consequências cognitivas que derivam dele.

Para garantir que o usuário tenha uma manipulação adequada e satisfatória do software, a equipe deve *definir as estratégias de aprendizagem (T1)*. Tais estratégias representam os meios pelos quais os usuários alcançarão determinados resultados. Portanto, elas estão diretamente relacionadas aos objetivos de aprendizagem detectados no levantamento de requisitos. Como exemplo, tomemos um contexto em que se identificou a dificuldade do público em iniciar e manter interações sociais. Neste contexto, a estratégia de aprendizagem será a resposta para a pergunta: o que fazer para melhorar a qualidade das interações sociais do indivíduo TEA com outras pessoas?

Esta pergunta pode ser respondida de diversas formas. Pode-se propor o desenvolvimento de uma rede social que conecte indivíduos TEA com interesses em comum entre si. Ou então um jogo educacional para ser aplicado em contexto de sala de aula, no qual os jogadores tenham que jogar cooperativamente para vencer. Ou, ainda, uma plataforma digital com conteúdo de entretenimento, que estimule pais e filhos TEA a passar mais tempo juntos. Estas são três soluções que, potencialmente, poderiam ajudar a estimular maior interação social no público.

Observa-se que, apesar das três opções apresentadas terem por finalidade cumprir o mesmo objetivo de aprendizagem (estimular a interação social), cada uma delas tem uma natureza distinta, podendo envolver diferentes agentes, mídias, ambientes e processos de interação. A seleção da estratégia de aprendizagem é uma materialização genérica *do que* o software deve fazer, e não de *como* ele deve fazer.

Uma vez definida a estratégia de aprendizagem, passa-se à *seleção das técnicas de aprendizagem* (T2). As técnicas de aprendizagem, por sua vez, estabelecem as abordagens utilizadas para a realização da estratégia, adotada anteriormente. Por meio da revisão sistemática, pode-se elencar algumas técnicas comumente empregadas no desenvolvimento de software educacional para o público TEA, tais como: feedback, sistema de recompensas, realidade virtual gamificação, personalização da interface, abordagem multimídia, *storytelling*, monitoramento, customização da aprendizagem, e *multiplayer* (para jogos).

É importante ressaltar que a escolha das técnicas de aprendizagem não é arbitrária. Elas devem, necessariamente, ser escolhidas em função dos estilos de aprendizagem identificados no levantamento de requisitos, e tendo em mente a estratégia adotada. É possível – e recomendado – que várias técnicas sejam utilizadas em conjunto, a fim de obter melhores resultados. Recuperando o exemplo anterior, pensemos no desenvolvimento de um jogo educacional cooperativo para estimular a interação social em indivíduos TEA.

Pode ser que o estilo de aprendizagem do público seja majoritariamente Experimental. Neste caso, deve-se adotar técnicas que facilitem a aprendizagem por imersão e pela experimentação ativa, como a Realidade Virtual e a personalização da interface. Contrariamente, um público cujo estilo de aprendizagem seja Imaginativo requer a adoção de técnicas de criação de possibilidades e pela observação reflexiva, como o *storytelling* (geração de narrativas próprias) e um sistema de recompensas, por exemplo. Destaca-se, neste momento do

desenvolvimento, *como* os objetivos serão alcançados, uma vez que *o que* será feito já foi estabelecido anteriormente.

Encerrando-se a atividade (A1), passa-se à atividade (A2) **modelagem do sistema**. Um vez definidas a estratégia de aprendizagem e as respectivas técnicas que serão adotadas, é possível ao arquiteto de software estabelecer a modelagem do sistema. Esta etapa corre naturalmente no processo de desenvolvimento, sendo executada de acordo com a metodologia empregada por cada organização, conforme o contexto e a realidade de cada projeto.

6.4.4 Desenvolvimento

A quarta etapa corresponde ao desenvolvimento (E4). É neste momento que ocorrerá o design da interface gráfica e os processos de codificação. Esta fase inicia-se com a transposição de requisitos em forma de componentes visuais que serão codificados, e resulta em um protótipo funcional da parte do software prevista no ciclo de iteração. As atividades e tarefas pertencentes a esta etapa são:

(E4) Desenvolvimento

(A1) Design da Interface Gráfica do Usuário

(T1) Métodos de representação

(T2) Métodos de decoração

(T3) Seleção de diretrizes gráficas

(A2) Codificação

A Interface Gráfica do Usuário (IGU) é a parte da interface que é visível aos usuários e com a qual eles podem interagir, a fim de executar determinadas tarefas. O **Design da IGU** (A1) nada mais é do que a transposição de componentes em elementos visuais. Esta atividade é particularmente importante porque representa o contato direto do usuário com os mecanismos do software. Para que isso aconteça satisfatoriamente, em primeiro lugar é preciso *selecionar os métodos de representação* (T1).

Os métodos de representação caracterizam o nível mais específico dos elementos que compõem a interface. São os componentes com os quais o usuário pode interagir e executar tarefas. Prosseguindo com o exemplo citado anteriormente, detalha-se como as técnicas de aprendizagem serão executadas de maneira compreensível para o usuário. O contexto

mencionado apresenta uma situação em que se deseja desenvolver um software para estimular interações sociais em indivíduos TEA.

A estratégia de aprendizagem selecionada é o desenvolvimento de um jogo educacional no qual alunos TEA cooperam entre si para cumprir objetivos. Supondo que o estilo de aprendizagem do público seja majoritariamente Experimental, decide-se adotar a Realidade Virtual e a personalização como técnicas de aprendizagem. Os métodos de representação são, portanto, os canais de entrada e saída de estímulos que permitem a aplicação destas técnicas. O uso da Realidade Virtual, por exemplo, demanda estímulos visuais em formato de vídeo e/ou animação, e também auditivos para a trilha sonora do jogo. Para a personalização do ambiente, podemos utilizar ícones representando as opções que o usuário pode escolher, e fotografias dos próprios jogadores, que eles possam carregar no sistema e usar como seu avatar.

Caso o público tivesse estilo de aprendizagem Imaginativo, o que nos levasse a escolher as técnicas de *storytelling* e o sistema de recompensas, os estímulos possivelmente seriam outros: para o *storytelling*, podemos disponibilizar ilustrações vetoriais dispostas aleatoriamente, para que o jogador escolhesse sua própria sequência de contar a história do jogo. Além disso, textos curtos poderiam acompanhar as imagens, auxiliando a construção da narrativa. As recompensas, por sua vez, podem estimular o jogador de duas maneiras: visualmente, caso sejam recompensas virtuais, ou sensorialmente, caso sejam recompensas no mundo real.

A importância de escolher corretamente os métodos de representação está em conectar as funções disponíveis no ambiente virtual às reais capacidades e habilidades do público TEA. Por exemplo, estímulos textuais não funcionam com indivíduos TEA que não sabem ler. Estimular a leitura não é um objetivo de aprendizagem no exemplo aqui apresentado, portanto incluir textos, nesse caso, deve ser uma decisão cautelosa. Uma vez escolhidos os métodos de representação, a próxima tarefa é a *seleção de métodos de decoração* (T2).

Os métodos de decoração são similares aos métodos de representação, pois também se referem aos estímulos de entrada e saída que compõem a interface do usuário. A diferença está em que os métodos de decoração são aqueles que não são essenciais ao funcionamento da interface e que não estão diretamente atrelados à execução de tarefas que levem ao cumprimento dos objetivos de aprendizagem. Por exemplo, digamos que ao executar corretamente uma atividade no software, um feedback visual (uma mensagem de comemoração) aparece na tela para indicar o sucesso (método de representação).

Simultaneamente, pode haver um efeito sonoro breve, ou a troca de cor de algum elemento, que reforce a ideia da atividade ter sido bem-sucedida. Estes não são métodos imprescindíveis para a compreensão do que está acontecendo, mas ajudam a reforçar a ideia geral e facilitam a assimilação do que se apresenta. Este é um exemplo de *método de decoração*. Pode-se decidir pela não-utilização de métodos de decoração, se a equipe julgar que eles não trarão benefícios concretos à manipulação do software ou ao cumprimento dos objetivos de aprendizagem estabelecidos.

Finalmente, após a definição dos métodos de decoração (se houverem), passa-se à última tarefa desta atividade, que é a *seleção das diretrizes gráficas* (T3). Diretrizes gráficas são recomendações que visam a viabilizar a execução dos métodos de representação e decoração. Elas se relacionam diretamente com o trabalho do designer gráfico, já que apresentam soluções para arranjo espacial, alinhamentos, tamanhos dos elementos, composição, combinações cromáticas, etc.

Este *framework* usa como referência e recomenda as diretrizes gráficas para acessibilidade TEA propostas por Britto [13]. No entanto, a equipe de desenvolvimento pode selecionar as diretrizes que melhor se encaixem ao escopo do projeto que estiver desenvolvendo. Neste momento, gera-se um relatório de UX e IU (modelo disponível no Apêndice B), contendo todas as informações referentes às estratégias e técnicas de aprendizagem, à seleção dos métodos de representação e decoração, e as diretrizes gráficas. Este relatório tem a intenção de guiar designers e programadores na atividade seguinte, além de ser um documento de referência mesmo para projetos futuros.

Após todo este processo, a interface gráfica estática é convertida em componentes interativos por meio da **codificação** (A2). Esta atividade corresponde à escrita do código-fonte do software. É neste momento que o design estático das telas se traduz em linguagem de programação que poderá ser processada pelos dispositivos de acesso. Esta atividade deve se dar de acordo com a metodologia ágil utilizada pela equipe, seguindo as técnicas comuns a cada método, que normalmente incluem *pair programming*, refatoração de código, etc. O processo de codificação culmina a materialização de um protótipo funcional que será usado para a realização de testes.

6.4.5 Testes Unitários

Nesta etapa, que acontece logo após o processo de codificação da parte do software prevista para o ciclo de iteração atual, a intenção é verificar possíveis falhas ou deficiências no software, possibilitando sua prevenção ou correção ao longo dos ciclos de desenvolvimento. Esta etapa normalmente inclui a execução dos testes propriamente ditos (de acordo com o estabelecido na etapa de planejamento), a análise dos relatórios gerados, e o planejamento das alterações a serem realizadas (que serão incluídas em ciclos de iteração futuros).

Este *framework* não propõe nenhuma atividade ou tarefa adicional ao que já é comumente empregado nesta etapa, dependendo do método ágil adotado. No entanto, ressalta-se que, de acordo com a abordagem do design centrado no usuário, a fase de testes deve, necessariamente, incluir um ou mais grupos de *stakeholders*, pois somente o contato com usuários ou especialistas poderá garantir maior nível de qualidade no produto final. Alguns métodos possível para a realização de testes, extraídos com a revisão sistemática, são: prototipação, inspeções de usabilidade, avaliações heurísticas, estudos de caso, questionários e/ou entrevistas, e grupos focais.

Além disso, este *framework* também sugere a utilização de um conjunto de heurísticas de usabilidade voltados especificamente para o público TEA, descrito por Khowaja e Salim [39]. As heurísticas de usabilidade são recomendações específicas para avaliar a conformidade dos produtos de software em termos de qualidade de uso. Os autores propõem um conjunto de 15 heurísticas, baseadas nas heurísticas originais de Nielsen, mas voltadas especialmente para o público autista.

Elas são: (1) visibilidade do estado do sistema; (2) equivalência com o mundo real; (3) consistência e padronização; (4) reconhecimento ao invés de memorização; (5) estética e design minimalista; (6) controle e liberdade do usuário (atualizada); (7) prevenção de erros (atualizada); (8) flexibilidade e eficiência de uso (atualizada); (9) reconhecimento, diagnóstico e recuperação de erros (atualizada); (10) ajuda e documentação (atualizada); (11) personalização de elementos; (12) sutileza na progressão entre telas; (13) responsividade do sistema; (14) histórico e monitoramento de atividades; e (15) uso de comunicação multimídia. Estas heurísticas podem ser encontradas, na íntegra, ao final deste trabalho (Anexo C).

6.4.6 Implantação

A implantação é a etapa em que ocorre a transferência do software do ambiente de produção para o ambiente real em que ele será utilizado. É o momento em que o software será disponibilizado em sua versão final para o cliente. Isso indica, portanto, que todos os ciclos de iteração devem estar concluídos, ou seja, o software deve ser entregue em sua integralidade, com todos os componentes e funcionalidades, conforme as especificações de requisitos e atributos do software.

Nesta etapa, recomenda-se fortemente que haja um processo de treinamento quanto à manipulação do software, pensando que isso pode ajudar a prevenir que sua má utilização resulte em erros inesperados e que todo o seu potencial de aplicação seja aproveitado. A equipe pode, por exemplo, realizar sessões de treinamento com os profissionais que utilizarão o software. Ou então disponibilizar tutoriais em formato de áudio ou vídeo. Ou, ainda, produzir um manual técnico de utilização, contendo as diretrizes gerais de funcionamento.

6.4.7 Validação do Sistema

Nesta etapa, ocorrem duas atividades de validação do sistema já transferido para o ambiente real de utilização. Em um primeiro momento, acontecem os testes de aceitação, que são testes conduzidos pelo cliente acerca do produto de software. A equipe de desenvolvimento normalmente não tem controle sobre a condução destes testes, pois este é o momento em que os clientes e demais *stakeholders* verificarão a conformidade total do software que foi entregue.

Em segundo lugar, o cliente fornece feedback indicando erros ou inadequações que por ventura passaram despercebidos aos olhos da equipe. Se for o caso, a equipe de desenvolvimento deve realizar as modificações solicitadas. No entanto, estima-se que os erros detectados pelo cliente, caso existam, serão mínimos, já que segundo as práticas do DCU, o contato com *stakeholders* foi constante durante o desenvolvimento, permitindo que correções fossem sendo feitas ao longo do percurso.

6.4.8 Entrega

O produto de software finalizado é entregue ao cliente, coincidindo com o final do processo de desenvolvimento. Caso esteja especificado nos acordos contratuais, a equipe de desenvolvimento poderá implantar um programa de melhoria contínua, disponibilizando atualizações do software com base no feedback dos usuários ao longo do tempo. Mesmo que este não seja o caso, recomenda-se que a equipe de desenvolvimento mantenha contato com seu cliente, acompanhando e pedindo feedback da utilização do software. Por meio desta prática é possível reunir conhecimento sobre as experiências prévias de desenvolvimento, melhorando a performance do time em projetos futuros. Um modelo de questionário de acompanhamento pós-entrega é apresentado no Apêndice C.

7 ESPECIFICAÇÕES PROCESSUAIS

Este capítulo é uma continuação da descrição do *framework*, ainda segundo o meta-processo proposto por Bolanos et al [11]. A seguir, apresentam-se explicações detalhadas acerca das especificações processuais, referentes às 8 (oito) etapas restantes do meta-processo, que são as seguintes: documentação, configuração de mecanismos e diretrizes, mapa de artefatos, entradas e saídas, meios de comunicação, identificação de problemas, melhorias e gestão do conhecimento. O detalhamento destes aspectos ajuda a compreender as regras particulares de funcionamento do *framework*, ao mesmo tempo em que viabilizam sua implantação.

7.1 Documentação

Dado que o *framework* foi pensado para integrar-se aos modelos ágeis, a documentação não deve ser excessiva, e sim suficiente para dar suporte ao processo. Evidentemente, a equipe de desenvolvimento pode (e deve) manter os padrões de documentação de que já faça uso, por exemplo, documentos UML (*Unified Modeling Language*), histórias de usuário, casos de uso, etc. Em complementação a esta documentação básica, gerada de acordo com cada modelo adotado, este *framework* prevê a geração de três documentos adicionais, descritos a seguir:

(1) Relatório geral da intervenção: este relatório é gerado ao término da etapa de levantamento de requisitos, e deve conter todas as especificações a respeito da intervenção (software). Não apenas do ponto de vista de requisitos e atributos, mas também uma explicação detalhada do público-alvo, envolvendo o macro e microambiente de aplicação, o perfil dos indivíduos, os sintomas presentes neles, seus estilos de aprendizagem, e os objetivos de aprendizagem esperados. O modelo de relatório de intervenção pode ser encontrado ao final deste trabalho (Apêndice A).

(2) Relatório de UX e IU: após as atividades de modelagem da arquitetura do software e design da interface gráfica, e antes do início da codificação, produz-se o relatório de Experiência do Usuário e Interface Gráfica (Apêndice B). Neste relatório estão contidas todas as informações da arquitetura do sistema, incluindo estratégias e técnicas de aprendizagem

empregadas, e as diretrizes gráficas, bem como os métodos de representação e decoração. A intenção deste relatório é fornecer subsídios para que os programadores consigam conduzir a codificação com o maior rigor possível ante aos requisitos do projeto e àquilo que foi pensado pelo designer gráfico.

(3) Questionário pós-entrega: Este questionário deve ser entregue aos clientes após a entrega do software e o encerramento do projeto. A intenção é obter feedback sobre o funcionamento do software, tanto com relação ao aparecimento de erros e *bugs*, mas especialmente quanto à concretização dos objetivos pedagógicos propostos. Idealmente, este relatório é encaminhado ao cliente alguns meses após a entrega, para que se tenha um tempo de utilização do software antes de buscar *insights*. O modelo deste documento está disponível no Apêndice C.

7.2 Configuração de Mecanismos e Diretrizes

Prosseguindo com descrição do *framework*, a configuração de mecanismos e diretrizes traz uma visão panorâmica das etapas, atividades e tarefas que o constituem, denominados *mecanismos do processo*. Por sua vez, as *diretrizes* são condições essenciais ou instruções para a sua implantação. Todas estas informações estão condensadas na Tabela 4, que se apresenta abaixo.

Tabela 4. Mecanismos e diretrizes do framework

Nome	Descrição	Diretrizes
Levantamento de requisitos	Etapas	Contato direto com especialistas e <i>stakeholders</i>
Identificação do público	Atividade	
Especificação do contexto	Tarefa	Análise do macro-ambiente
Definição da qualidade de intervenção	Tarefa	Análise do micro-ambiente de aplicação
Caracterização dos sintomas	Tarefa	Utilização de escala de diagnóstico (CARS)
Estilos de aprendizagem	Tarefa	Estilos de aprendizagem de [42]
Objetivos de aprendizagem	Tarefa	
Análise de requisitos	Atividade	
Validação de requisitos	Atividade	Prototipação de baixa fidelidade

Planejamento	Etapa	
Cronograma de trabalho	Atividade	Planejamento de testes
Atribuição de responsabilidades	Atividade	Listagem de <i>stakeholders</i> envolvidos
Alocação de recursos	Atividade	
Arquitetura de software	Etapa	
Definição da UX	Atividade	
Estratégias de aprendizagem	Tarefa	Baseadas nos sintomas detectados
Técnicas de aprendizagem	Tarefa	Baseadas nos estilos de aprendizagem
Modelagem do sistema	Atividade	
Desenvolvimento	Etapa	Entrega de software funcional a cada iteração
Design da IGU	Atividade	
Métodos de representação	Tarefa	Baseados na Experiência do Usuário
Métodos de decoração	Tarefa	Baseados na Experiência do Usuário
Diretrizes gráficas	Tarefa	Diretrizes de acessibilidade [13]
Codificação	Atividade	
Testes unitários	Etapa	
Execução dos testes	Atividade	Heurísticas de acessibilidade TEA [39]
Planejamento de alterações	Atividade	
Implantação	Etapa	
Integração do sistema	Atividade	Manual de utilização do sistema
Transferência ao ambiente real	Atividade	Treinamentos / tutoriais
Validação do sistema	Etapa	
Testes de aceitação	Atividade	
Adequações finais	Atividade	
Entrega	Etapa	
Melhoria contínua	Atividade	
Acompanhamento e feedback	Atividade	Produzir relatório de acompanhamento

Ademais, é possível elencar algumas técnicas e métodos que se adequam com êxito às diferentes atividades e tarefas do *framework*. A Tabela 5 apresenta as melhores práticas para a condução e realização de determinadas ações, além de sucinta explicação sobre seu uso. Estas práticas foram retiradas dos artigos analisados na revisão sistemática. Embora sejam apenas sugestões, o seu uso recorrente na literatura indica uma clara preferência por estas ferramentas e sugere que sua aplicação seja bem-sucedida.

Tabela 5. Melhores práticas para acessibilidade TEA

Etapas, atividade ou tarefa	Técnica ou método	Descrição
Levantamento de requisitos	Entrevistas	Entrevistas estruturadas ou semi-estruturadas conduzidas com <i>stakeholders</i> , mas não com usuários finais
	Observações de campo	Observações controladas do usuário final em ambiente real (casa, clínica, escola, etc.)
	<i>Demo sessions</i>	Sessões de demonstração envolvendo os <i>stakeholders</i> , para demonstrar o pressuposto funcionamento do software. Normalmente realizada para validar requisitos
	Grupos focais	<i>Stakeholders</i> e usuários finais são envolvidos em sessões que objetivam extrair requisitos diretamente dos participantes
Técnicas de aprendizagem	Feedback	Dar um retorno (seja ele virtual ou real) ao usuário acerca das atividades que realiza, sejam os resultados positivos ou negativos
	Sistema de recompensas	Recompensar os usuários por atividades realizadas corretamente. A recompensa pode ser virtual ou real, mas deve ser dada logo após a realização da atividade
	Gamificação	Aspectos de jogos que são incluídos no software, como uso de avatares, <i>multiplayer</i> , <i>puzzles</i> , sistema de fases, etc.
	Personalização	A interface permite que o usuário personalize aspectos gráficos (cores, fontes, formas, layout, etc.) de acordo com suas preferências ou necessidades
	Monitoramento	Permite que pais, professores ou terapeutas acompanhem o progresso do usuário autista. Normalmente com a integração de um módulo de gerenciamento
	Customização de conteúdo	Permite que professores e terapeutas selecionem os conteúdos adequados à aprendizagem, bem como o nível de dificuldade que se deseja
	Multimídia	Abordagem que combina duas ou mais técnicas e métodos, a fim de que a redundância de representação atinja um público mais numeroso
Métodos de representação e/ou decoração	Ilustração	Uso de ilustrações vetoriais ou feitas à mão, com estilos diversos: <i>cartoon</i> , <i>outline</i> , geométrico, etc.
	Fotografias	Retiradas de bancos de imagens ou das próprias galerias pessoais dos usuários. Utilizadas quando se deseja maior nível de realismo no software

	Palavras/texto	Uso de palavras ou textos muito curtos para dar instruções. Não devem ser usados textos longos para públicos com dificuldade de comunicação
	Cores	As preferências de cores podem variar de pessoa para pessoa. Um recurso útil pode ser fornecer paletas de cores pré-definidas, que o usuário escolha para atender às suas próprias necessidades
	Sons	Sons inerentes à interface, como música de fundo ou feedback de ações. Em alguns casos, também são usadas narrações e gravações pessoais
	Vídeo	Uso de vídeo ou animações para aumentar o dinamismo e a interação. Podem ser vídeos de pessoas familiares ao indivíduo autista
Testes	Testes de usabilidade	Realizados com pais, professores, terapeutas, ou usuários finais, dependendo da disponibilidade de recursos humanos. Sugere-se que os testes sejam aplicados com uma amostra estatisticamente adequada da população
	Questionários	Geralmente aplicados com pais, professores e/ou terapeutas quando não é possível envolver usuários finais, ou quando se deseja obter mais de uma fonte de feedback
	<i>Demo sessions</i>	Sessões de demonstrações do produto final, em que também ocorre a interação dos <i>stakeholders</i> , mas sem um procedimento rígido de testes

7.3 Mapa de Artefatos

Os artefatos produzidos com o uso deste *framework* se dividem em duas categorias: os artefatos de desenvolvimento e os artefatos de suporte. Os *artefatos de desenvolvimento* são aqueles oriundos da própria produção do software, e que têm como objetivo progredir no ciclo de desenvolvimento. Os artefatos de desenvolvimento iniciam com a produção dos protótipos na fase de levantamento de requisitos, incluindo também os componentes de software funcional que são entregues ao final de cada ciclo de iteração.

Já os *artefatos de suporte* são aqueles que visam a capacitação de pessoal para a utilização do produto de software, e são gerados na fase de implantação. Como já mencionado anteriormente, recomenda-se a produção de um manual de funcionamento do software, que é entregue ao cliente antes dos testes de aceitação. Este é um exemplo de possível artefato de suporte. Além disso, a equipe de desenvolvimento pode fornecer aos clientes sessões de

treinamento e tutoriais, com o intuito de garantir a correta utilização do sistema. Podem ser gerados, ainda, outros artefatos que sejam previstos nos respectivos modelos de desenvolvimento adotados por cada equipe.

7.4 Entradas e Saídas

As entradas e saídas do framework estão sintetizadas na Tabela 6. Note que, como as etapas são divididas em atividades e subdivididas em tarefas, em alguns casos as entradas e saídas são idênticas. Ademais, as etapas de levantamento de requisitos, planejamento, arquitetura de software, desenvolvimento e testes unitários constituem o *ciclo de iteração* e, portanto, podem ser repetidas quantas vezes forem necessárias.

Tabela 6. Entradas e saídas do framework

Entrada	Mecanismo	Saída
Solicitação do cliente	Levantamento de requisitos	Relatório de intervenção
Solicitação do cliente	Identificação do público	Análise do macro e microambiente
Solicitação do cliente	Especificação do contexto	Análise do macro-ambiente
Análise do macro-ambiente	Definição da qualidade de intervenção	Análise do microambiente
Análise do microambiente	Caracterização dos sintomas	Resultado da escala diagnóstica
Resultado da escala diagnóstica	Estilos de aprendizagem	Análise dos estilos de aprendizagem
Análise do macro e microambiente, resultado da escala diagnóstica, análise dos estilos de aprendizagem	Objetivos de aprendizagem	Planilha de objetivos de aprendizagem
Planilha de objetivos de aprendizagem	Análise de requisitos	Requisitos funcionais, não-funcionais, atributos de qualidade
Requisitos funcionais, não-funcionais, atributos de qualidade	Validação de requisitos	Relatório de intervenção
Relatório de intervenção	Planejamento	Cronograma de iterações
Relatório de intervenção	Cronograma de trabalho	Documento do cronograma

Documento do cronograma	Atribuição de responsabilidades	Especificação das tarefas
Especificação das tarefas	Alocação de recursos	Cronograma de iterações
Cronograma de iterações, relatório de intervenção	Arquitetura de software	Plano de arquitetura do software
Cronograma de iterações, relatório de intervenção	Definição da UX	Plano de Experiência do Usuário
Planilha de objetivos de aprendizagem	Estratégias de aprendizagem	Documento de estratégias de aprendizagem
Análise dos estilos de aprendizagem	Técnicas de aprendizagem	Documento de técnicas de aprendizagem
Documento de estratégias e técnicas de aprendizagem	Modelagem do sistema	Plano de arquitetura do software
Plano de arquitetura do software, relatório de intervenção	Desenvolvimento	Protótipo funcional
Plano de arquitetura do software, relatório de intervenção	Design da IGU	Relatório de UX e UI
Plano de arquitetura do software, relatório de intervenção	Métodos de representação	Planilha de métodos de representação
Planilha de métodos de representação	Métodos de decoração	Planilha de métodos de decoração (se houver)
Planilha de métodos de representação e decoração	Diretrizes gráficas	Relatório de UX e UI
Plano de arquitetura do software, relatório de UX e UI	Codificação	Protótipo funcional
Protótipo funcional	Testes unitários	Novo ciclo de iteração ou software pronto para integração
Protótipo funcional	Execução dos testes	Relatório de testes
Relatório de testes	Planejamento de alterações	Novo ciclo de iteração ou software pronto para integração
Software pronto para integração	Implantação	Software em ambiente real de uso
Software pronto para integração	Integração do sistema	Software integrado
Software integrado	Transferência ao ambiente real	Software em ambiente real de uso
Software em ambiente real de uso	Validação do sistema	Software pronto para entrega

Software em ambiente real de uso	Testes de aceitação	Software aceito ou relatório de adequações (se houver)
Relatório de adequações	Adequações finais	Software pronto para entrega
Software pronto para entrega	Entrega	
Software pronto para entrega	Melhoria contínua	Plano de melhoria contínua
Software pronto para entrega	Acompanhamento e feedback	Questionário pós entrega e plano de melhoria contínua

7.5 Meios de Comunicação

Este *framework* não especifica meios de comunicação obrigatórios entre os agentes e membros da equipe. Espera-se que as equipes utilizem os meios de comunicação a que já estão acostumados em seu dia-a-dia, para que a integração do *framework* seja a mais sutil possível. No entanto, cabe lembrar da importância de que a equipe tenha uma comunicação direta e eficiente com os *stakeholders*, incluindo clientes, especialistas, pais, cuidadores, e o próprio público TEA.

Esta comunicação deve ser constante para que o *framework* funcione como o esperado e gere resultados positivos. Embora os meios de comunicação dependam da organização da equipe e da hierarquia presente, sugere-se que os desenvolvedores possam contatar diretamente os responsáveis pelas aprovações, evitando ruídos indesejados em comunicações intermediárias.

7.6 Identificação de Problemas

A identificação de problemas no processo ocorre por meio das ferramentas de gestão da qualidade. Este *framework* utiliza o Modelo de Maturidade de Usabilidade descrito por Mostafa [55]. O modelo apresentado pelo autor é particularmente interessante porque trata da maturidade em processos DCU-ágeis, e portanto encaixa-se com perfeição ao escopo deste trabalho. O autor inclui explicações extensivas a respeito dos níveis de maturidade propostos, as áreas de avaliação que o modelo contempla, a escala de avaliação utilizada e, ainda, os procedimentos de avaliação. Os níveis propostos pelo autor são seis:

- **Nível 0 – impossível:** A integração do DCU ágil é desencorajada. Não há motivação para realizar a integração e as equipes não compreendem ou não valorizam o DCU.
- **Nível 1 – possível:** A integração do DCU ágil não é desencorajada. Há motivação para realizar a integração e alguns membros da equipe compreendem e valorizam o DCU.
- **Nível 2 – encorajado:** A cultura organizacional encoraja a integração do DCU ágil. Os valores, benefícios e significados do DCU são reconhecidos e compreendidos, e seus resultados são alcançado em alguns projetos.
- **Nível 3 – viabilizado/praticado:** A integração do DCU ágil é praticada. Métodos e ferramentas são utilizados para viabilizar a integração, os postos de trabalho são organizados e existe uma equipe que capacita as atividades de integração. Os gestores apoiam e promovem a integração do DCU ágil.
- **Nível 4 – gerenciado:** A integração do DCU ágil não só é praticada como também esperada das equipes de desenvolvimento. As atividades fazem parte do ciclo de desenvolvimento. Métodos, ferramentas, postos de trabalho e equipes direcionam-se no sentido da integração. Há treinamento disponível. Todos os membros das equipes são comprometidos com o DCU, desde os gestores, aos desenvolvedores e demais *stakeholders*.
- **Nível 5 – melhoria contínua:** Os processos DCU são revisados e avaliados para que se possam planejar melhorias. Os métodos, ferramentas, postos de trabalho e equipes são regularmente monitorados e continuamente melhorados.

Em primeiro lugar, é preciso realizar uma avaliação diagnóstica da organização, que ocorre por meio de listas de verificação sobre as áreas analisadas (Anexo D). Com os resultados da análise diagnóstica em mãos, a gerência da equipe pode definir quais ações serão tomadas para que a usabilidade geral do processo DCU-ágil seja maximizada. Para localizar a organização dentro dos níveis, o autor propõe uma avaliação dos processos em cinco grupos distintos: (a) infraestrutura do DCU, (b) processo do DCU, (c) equipe de desenvolvimento, (d) clientes e usuários, e (e) melhoria contínua do DCU. Cada grupo é composto por outros elementos de verificação:

(a) infraestrutura do DCU: avalia se há fundos de investimento para a realização do DCU, e também a disponibilidade de equipe qualificada, ferramentas, métodos e treinamento. Estes devem seguir padrões, diretrizes e guias de estilo.

(b) processo do DCU: avalia se o processo DCU é planejado. Deste planejamento deve emergir a identificação do usuário e a análise de suas tarefas, resultando em requisitos do usuário a serem incorporados na interface. Além disso, avalia a existência de documentação leve e se as tarefas de design (design gráfico e codificação) acontecem paralelamente.

(c) equipe de desenvolvimento: avalia a existência e as relações entre a equipe de Experiência do Usuário (do inglês, UX) e a equipe de implementação, em termos de competência e comunicação. Deve haver uma cultura organizacional que motive a interação entre as equipes, aumentando suas habilidades de comunicação, cooperação, coordenação e colaboração.

(d) clientes e usuários: avalia se a integração com estes *stakeholders* ocorre de maneira efetiva e eficiente. Deve haver a identificação de clientes e usuários, envolvimento precoce no desenvolvimento, que também deve ocorrer durante todo o processo. Além disso, clientes e usuários devem ser conscientizados sobre a importância do DCU.

(e) melhoria contínua do DCU: avalia a existência e o gerenciamento de processos de monitoramento, bem como a sistematização de melhorias. As decisões de design devem partir das necessidades e expectativas do usuário, promovendo a geração de valor.

Em seguida, temos a escala de avaliação, que serve para localizar a organização nos níveis de maturidade, e que varia em um espectro de seis pontos. Ademais, é possível avaliar determinado aspecto com “Não sei”, indicando que o respondente não tem segurança para responder ao tópico, ou que aquele aspecto específico não se aplica ao contexto. A escala divide-se em: Muito alto > 4,5; Alto, de 3,5 a 4,5; Médio, de 2,5 a 3,5; Baixo, de 1,5 a 2,5; Muito baixo > 0 e < 1,5; ou Nenhum = 0.

Por fim, para efetivamente avaliar os aspectos mencionados e possibilitar inserir a organização dentro dos níveis de maturidade, procede-se com a aplicação dos seguintes passos: (a) diretrizes da avaliação, (b) contextos típicos, (c) avaliação da maturidade, (d) pontuação dos níveis de maturidade, e (e) relatório da avaliação.

(a) diretrizes da avaliação: são uma série de recomendações que devem ser seguidas antes, durante e após o processo de avaliação. Ditam também o comportamento do avaliador, o contexto e os recursos de avaliação.

(b) contextos típicos: uma série de possíveis respostas que o avaliador pode obter durante a avaliação. Isso ajuda o avaliador a posicionar determinado aspecto dentro dos níveis de maturidade em situações nas quais o respondente pode fornecer informações dúbias ou incompletas.

(c) avaliação da maturidade: são disponibilizadas folhas de referência com todos os aspectos que devem ser avaliados. Devem ser utilizadas como um padrão para assegurar que todas as competências do modelo de maturidade serão verificadas.

(d) pontuação dos níveis de maturidade: seguindo as diretrizes de avaliação, e com base nos contextos típicos e nas folhas de referência, o avaliador posiciona cada aspecto avaliado dentro da pontuação proposta na escala de avaliação.

(e) relatório da avaliação: a partir das informações colhidas com a avaliação, gera-se o relatório da avaliação, no qual apresenta-se o nível de maturidade de cada item avaliado. Além disso, deve-se indicar o estado atual de cada item, se são necessárias melhorias ou, ainda, se o processo precisa ser implantado completamente.

Em consonância com o modelo de maturidade, a equipe deve ser encorajada a documentar lições aprendidas (modelo disponível no Apêndice D). As lições aprendidas são ferramentas poderosas que tem a intenção de registrar e divulgar as experiências da equipe, a fim de direcionar processos futuros de desenvolvimento e dar suporte à tomada de decisões. A documentação de lições aprendidas deve ser motivada durante todo o processo de desenvolvimento, sempre que sua aplicação for pertinente.

7.7 Melhorias

A partir dos resultados colhidos com a avaliação diagnóstica e com as lições aprendidas documentadas, a equipe pode sugerir melhorias no processo. As melhorias devem ser categorizadas em curto, médio e longo prazo, dependendo da dificuldade de implantação e dos recursos disponíveis no momento. Recomenda-se que a equipe de desenvolvimento realize reuniões esporádicas com essa finalidade.

É interessante que uma reunião deste tipo ocorra logo após a aplicação da avaliação diagnóstica, para verificar quais pontos já podem ser melhorados antes do início do processo de desenvolvimento. Além disso, uma reunião final, após a aplicação do questionário

pós-entrega, pode ajudar a vislumbrar o projeto recém-terminado de maneira mais profunda, e propiciar o aparecimento de *insights*. Finalmente, reuniões para sugestão de melhorias podem ser conduzidas a qualquer momento durante o processo, sempre que verificada sua necessidade.

7.8 Gestão do Conhecimento

De posse das informações e documentos gerados ao longo do processo, a gestão do conhecimento deve ser facilitada. Obviamente, a avaliação diagnóstica e as lições aprendidas são documentos que devem ser divulgados. Mas outros documentos gerados pelo *framework*, como o relatório geral de intervenção, o relatório de UX e IU, e o questionário pós-entrega também são fontes importantes de conhecimento. Contudo, para que a gestão do conhecimento realmente funcione, ela precisa se atentar às recomendações tecidas por Al-Mudimigh, Ullah e Alsubaie [2].

Em primeiro lugar, é preciso conscientizar a equipe da importância da gestão da qualidade e do conhecimento, para que as ferramentas disponíveis de fato sejam utilizadas. Depois, é necessário reunir e documentar as experiências, e analisar se os resultados foram bem-sucedidos ou não. Em seguida, este conhecimento precisa ser divulgado e compartilhado com a equipe. Novos membros podem chegar a qualquer momento, e se não forem instruídos sobre os documentos existentes, poderão cometer erros já solucionados. Por fim, os registros precisam se manter atualizados, já que novos projetos certamente resultarão em novas experiências de fracasso ou sucesso.

8 INTEGRAÇÃO DO FRAMEWORK

Neste capítulo, apresenta-se uma investigação conduzida no laboratório Gaia – Soluções em Tecnologia da Informação e Comunicação (TIC). O laboratório é uma iniciativa pública, localizado nas dependências do Departamento de Computação da Universidade Estadual de Londrina, Centro de Ciências Exatas. Primeiro, apresenta-se a empresa na qual o estudo foi conduzido, incluindo importantes aspectos de contextualização como atuação corporativa, aspectos organizacionais, equipe, projetos e o modelo de desenvolvimento de software adotado.

Em seguida, realiza-se uma investigação acerca da integração do *framework* com o modelo de desenvolvimento já adotado, com a finalidade de verificar como se dá essa integração e como operam os mecanismos do *framework* em um modelo real. Por fim, apresentam-se os resultados obtidos com a verificação de especialistas. Embora de caráter preliminar, esta verificação apresenta um indicativo de que o *framework* desenvolvido pode ser útil ao processo de desenvolvimento de software, além de apontar possibilidades de futuras melhorias.

8.1 Apresentação da Empresa

O laboratório Gaia – Soluções em TIC, é uma organização de iniciativa pública vinculado ao Departamento de Computação da Universidade Estadual de Londrina. O laboratório desenvolve trabalhos de suporte às organizações quanto ao uso de Tecnologias da Informação, em termos de infraestrutura, recursos humanos, processos e sistemas [25]. De fato, o laboratório tem um catálogo de serviços bastante extenso, incluindo aplicativos, softwares, *frameworks*, objetos de aprendizagem e tutoriais. Ademais, por seu vínculo a uma universidade pública, há preocupação com a produção científica: todos os projetos desenvolvidos no laboratório são compilados em trabalhos e divulgados para a comunidade acadêmica.

O laboratório Gaia possui sua estrutura no próprio Departamento de Computação, dispondo de sala própria, computadores e um servidor para atender às demandas internas. A equipe do laboratório é composta por colaboradores de diversas áreas, como programadores, analistas, designers, professores e especialistas. Também envolve a comunidade

acadêmica, tanto em nível de graduação quanto pós-graduação. As produções acadêmicas desenvolvidas pela equipe incluem artigos, publicados no Brasil e no exterior, dissertações, teses, e registros de software. O Processo de Desenvolvimento de Software adotado pelo laboratório pode ser observado na Figura 5, que foi extraído do site oficial do laboratório (gaia.uel.br).

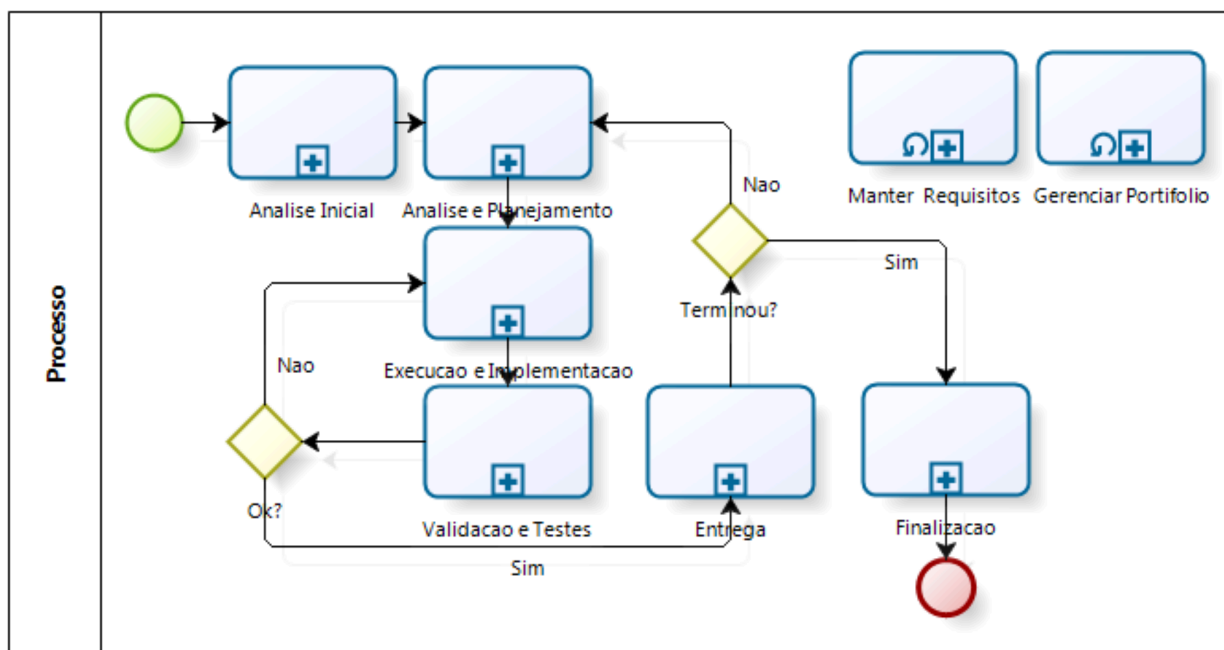


Figura 5. Processo de desenvolvimento de software do laboratório Gaia [25].

O PDS do Gaia inicia-se com uma Análise Inicial para verificar a viabilidade do projeto, que culmina com a aceitação do projeto e a escritura de um contrato. Em seguida, passa para etapa de Análise e Planejamento, que inclui o levantamento de requisitos e sua validação. Nesta etapa, também ocorre o planejamento das entregas, análise de riscos e estimativa de prazos e custos. Após esta fase, progride-se para a Execução e Implementação, que compreende as etapas de design da interface, programação, e realização de demais atividades. Segue-se, então, para a etapa de Validação e Testes, e por fim a Entrega e a Finalização do projeto. Paralelamente a isso, existe a preocupação de se manter os requisitos e gerenciar o portfólio. O PDS será explicado com maiores detalhes na próxima seção.

Como se pode notar, o laboratório Gaia é uma organização que já tem bases muito bem definidas quanto ao seu processo de desenvolvimento de software. Ademais, a empresa conta com uma equipe diversificada e atende demandas igualmente plurais de pessoas físicas e jurídicas, inclusive apoiando o empreendedorismo. Como seu catálogo de produtos e

serviços é bastante amplo, há a possibilidade de que demandas de acessibilidade (inclusive para portadores de TEA) surjam. Neste sentido, seria interessante ter um *framework* disponível para a condução de projetos específicos ao público autista, que pudesse ser acoplado ao PDS para suprir demandas pontuais e desacoplado quando seu uso não for mais necessário.

8.2 Integração do Framework com o PDS Gaia

Embora não haja identificação do modelo de desenvolvimento utilizado pelo laboratório, o fluxo do seu processo (Figura 5) deixa claro que todas as principais etapas dos métodos ágeis estão presentes, e que a produção é controlada por iterações. Além disso, conforme mencionado, a equipe do Gaia engloba profissionais de diversas áreas, com uma abordagem aparentemente multidisciplinar. Neste sentido, há grandes chances de que o *framework* proposto consiga se integrar satisfatoriamente ao PDS da organização. Acompanhe, a seguir, uma explicação mais detalhada deste processo e como se daria a integração do *framework* a ele.

Em primeiro lugar, o PDS Gaia parte de uma etapa de **Análise Inicial**. Nesta etapa, reuniões são realizadas para se definir o escopo do projeto, bem como realizar estimativas e preparar o ambiente. Por meio desta análise, verifica-se a viabilidade do projeto. Caso a proposta seja aceita, formaliza-se um contrato e concretiza-se o lançamento do projeto. Esta etapa não é contemplada pelo *framework* e não será discutida nesta investigação, já que suas atividades não dizem respeito ao desenvolvimento em si. Passa-se, portanto, à segunda etapa do PDS, a **Análise e Planejamento** (Figura 6).

No PDS Gaia, a etapa de análise e planejamento começa com o levantamento de requisitos, sendo que não há um claro indicativo de como isso deve ser feito. Recomenda-se, apenas, que os requisitos sejam levantados em conjunto com os *stakeholders*. Após esta atividade, segue-se para a revisão e validação de requisitos. Esta primeira parte do processo corresponde exatamente a primeira etapa do *framework*, portanto sua integração tende a ser bastante simples e sutil. Basta que, no momento do levantamento de requisitos, a equipe realize as atividades adicionais propostas: a especificação do contexto, a definição da intervenção, a caracterização de sintomas, e a definição dos estilos e objetivos de aprendizagem.

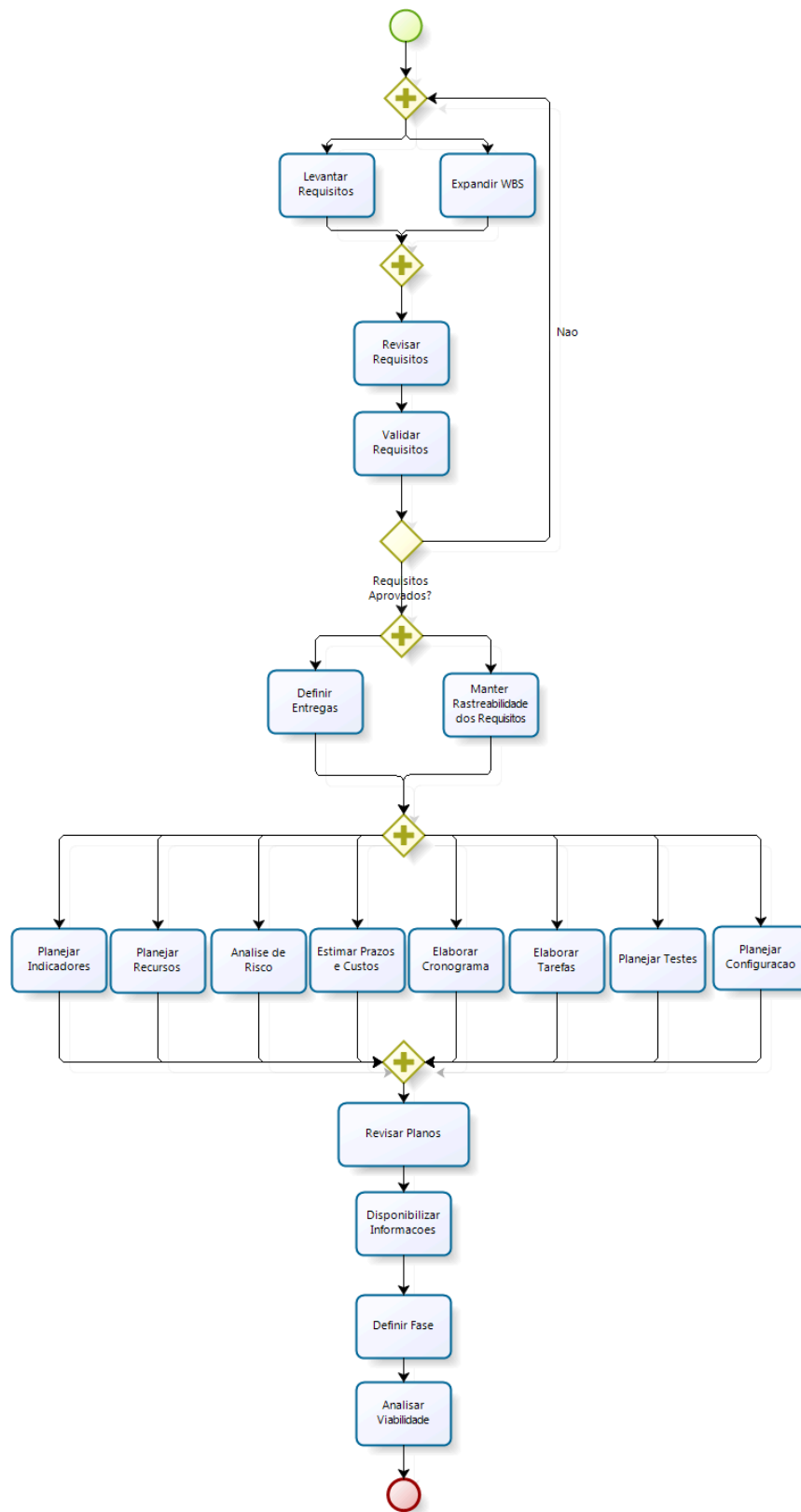


Figura 6. Etapa de análise e planejamento, PDS Gaia [25]

O passo final é a geração do relatório geral de intervenção. Uma vez produzido este documento, segue-se para a revisão e validação dos requisitos. Em seguida, o PDS Gaia segue para a parte de planejamento, que também é uma etapa coincidente com o processo do *framework*. Segundo o PDS Gaia, neste momento ocorre a definição de entregas (cronograma de iterações) e a gestão de rastreabilidade dos requisitos. Ademais, outras sub-tarefas são executadas: planejar indicadores, planejar recursos, analisar riscos, estimar prazos e custos, elaborar cronograma, elaborar tarefas, planejar testes e configuração. Conforme mencionado anteriormente, não existem especificações formais no *framework* para a condução desta etapa, portanto espera-se que a organização siga o processo a que já está acostumada.

A próxima fase do PDS Gaia é a etapa de **Execução e Implementação**. Antes de iniciar essa fase, é necessário integrar novamente o processo do *framework* e definir a Experiência do Usuário. Conforme descrito, isso requer a definição das estratégias e técnicas de aprendizagem, sempre envolvendo especialistas e usuários para dar suporte à tomada de decisões. Uma vez completada esta atividade, progride-se para as tarefas de Execução e Implementação previstas pelo PDS Gaia (Figura 7).

No momento de execução das tarefas de design, novamente voltamos a obter ajuda do *framework*. A fase que compreende o design da Interface Gráfica deve ser precedida, conforme especificado, pela definição dos métodos de representação e decoração e pela seleção de diretrizes gráficas. Ao final, o relatório de UX e IU é gerado, contendo as informações da Experiência do Usuário e os métodos e diretrizes selecionados para compor a interface gráfica. Em seguida, a implementação ocorre conforme a metodologia do próprio PDS. Além disso, o PDS Gaia inclui a realização de subatividades, que são: gerenciar riscos, garantir qualidade, gerenciar comunicação, e gerenciar configuração.

Em seguida, temos a etapa de **Validação e Testes** (Figura 8), que se organiza de forma muito similar à proposta pelo *framework*. Resumidamente, os testes programados no momento do planejamento são executados, partindo-se depois para a análise dos resultados e realização das correções, se houverem. O que cabe ressaltar aqui é que, a fim de se obter resultados consistentes e cientificamente respaldados, recomenda-se que os testes envolvam usuários finais e outros *stakeholders*, e que a população de teste seja abrangente e significativa. Pode ser usado, por exemplo, um grupo de controle para comparação de resultados com e sem o uso do software.

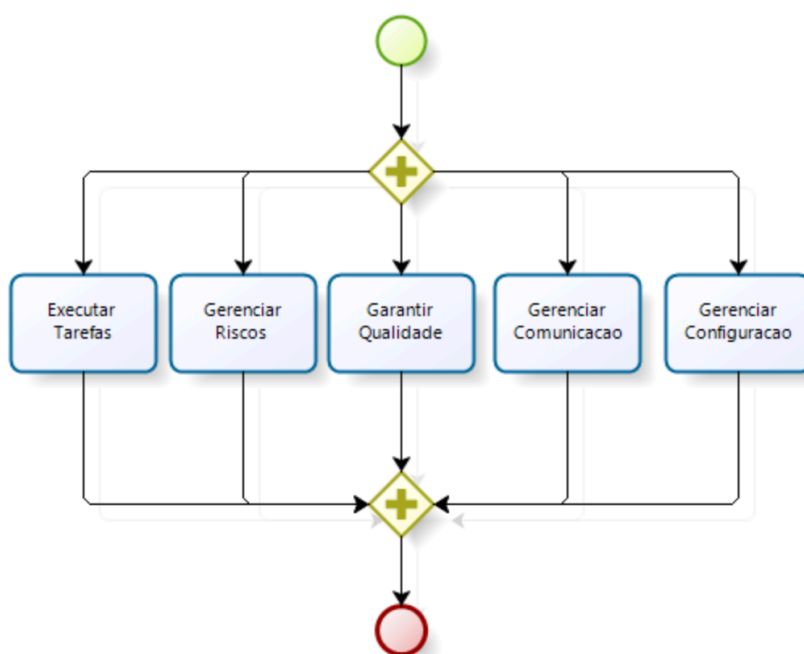


Figura 7. Etapa de execução e implementação, PDS Gaia [25].

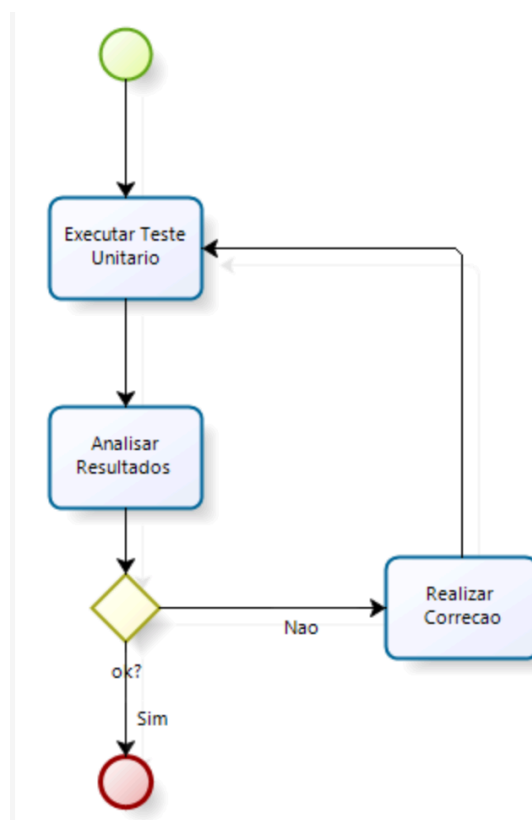


Figura 8. Etapa de validação e testes, PDS Gaia [25].

Ademais, o *framework* sugere testes de usabilidade com as heurísticas de acessibilidade TEA propostas por Khowaja e Salim [39]. Continuando o processo, o PDS Gaia avança para a fase de **Entrega**. Apesar do nome, na prática esta fase corresponde aos processos de implantação e entrega (ver Figura 9). Esta fase se inicia com os testes de integração (bem como análise dos resultados e correções, se for o caso) e segue para a integração dos componentes propriamente dita. Isso corresponde de forma muito similar ao processo do *framework*, que não propõe atividades complementares.

Em seguida, o PDS Gaia enfatiza uma atividade que corresponde a “realizar reunião de feedback (equipe e cliente)”. Não há muitas informações sobre essa atividade, de modo que não é possível saber com exatidão a que ela se refere. De qualquer modo, é necessário realizar testes de aceitação com o cliente antes da entrega efetiva, para que ele indique possíveis alterações ou ajustes finais.

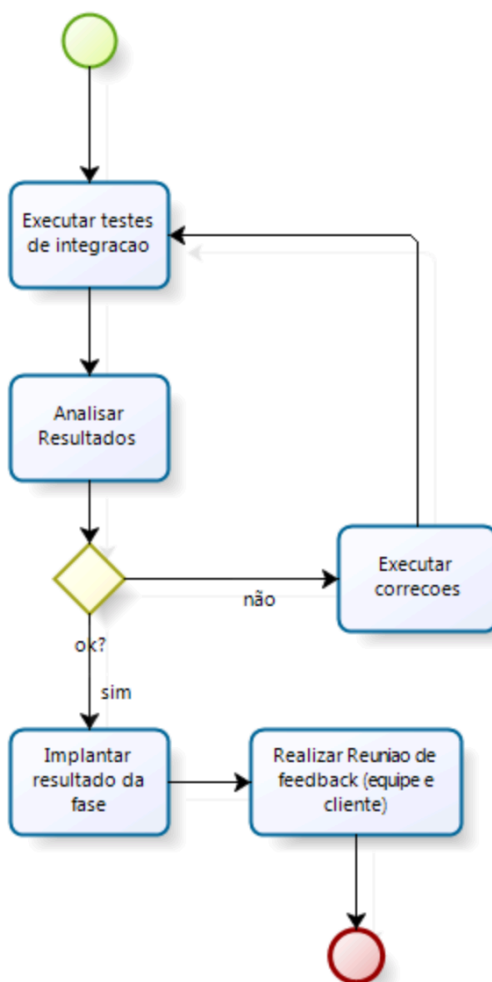


Figura 9. Etapa de entrega, PDS Gaia [25].

A última etapa prevista pelo PDS Gaia é a **Finalização**, que corresponde a reunião final com os *stakeholders* para entrega do produto de software e finalização do contrato de projeto. Após esta etapa, entra em cena a última fase proposta pelo *framework*, que inclui a melhoria contínua do software, com a disponibilização de atualizações (caso esteja previsto em contrato). Mesmo que não haja um plano de melhoria contínua para o produto, o *framework* sugere que seja feito um acompanhamento de uso do software, com a aplicação do questionário pós-entrega. Isso porque o feedback coletado com esse instrumento pode servir de base para projetos futuros e revelar indicadores de performance da equipe.

Como se pode notar, o PDS Gaia é absolutamente possível de ser integrado com o processo proposto pelo *framework*, pois a similaridade de etapas presentes em ambos (que são, não por acaso, etapas comuns aos modelos ágeis) permite isso. Com relação à abordagem do Design Centrado no Usuário, acredita-se que a integração também não seria difícil, pois o laboratório alega que já envolve uma equipe multidisciplinar em seus projetos. Evidentemente, espera-se que ao menos alguns dos profissionais envolvidos tenham conhecimento técnico e clínico específico sobre o Transtorno do Espectro Autista.

Para a gestão de qualidade da usabilidade, o *framework* sugere o uso do Modelo de Maturidade de Usabilidade DCU descrito por Mostafa [55]. A aplicação deste modelo é facilitada porque a avaliação diagnóstica pode ser realizada a qualquer momento, ao final da qual a organização terá um claro entendimento do nível de maturidade de cada um de seus processos. Ademais, não existem exigências específicas: as melhorias podem ser implantadas conforme a necessidade e a disponibilidade de recursos. Por fim, o uso das Lições Aprendidas certamente é um processo que demanda esforço gerencial, mas que não precisa de um ambiente muito específico para ser implantado.

8.3 Verificação Qualitativa

Uma vez investigada a possibilidade de integração do *framework* com o processo de desenvolvimento adotado pelo laboratório Gaia, recorre-se a um último instrumento de verificação. A verificação qualitativa utilizada neste estudo baseia-se nos procedimentos descritos por Horita [30] e Rautenberg [63]. Esta técnica consiste em coletar opiniões de especialistas acerca do objeto de estudo, que servirão de indicativo de sua eficácia. Especialistas,

neste caso, são profissionais com conhecimento e formação específicos na área ou que tenham considerável tempo de atuação na indústria. Recomenda-se que os especialistas sejam selecionados também em função da variedade de áreas do conhecimento, ou seja, que incluam profissionais de diversos domínios diferentes. Ao todo, foram contatados 17 especialistas relacionados à área da ciência da computação, design e educação. Abaixo são apresentados, de forma sucinta, o perfil destes especialistas:

Especialista 1: Da área da ciência da computação, com 5 anos de experiência, o Especialista 1 trabalha majoritariamente com Engenharia de Software, incluindo processos e gestão da qualidade.

Especialista 2: Também da área da ciência da computação, atuando especialmente com projetos de desenvolvimento de software, enfatizando a gestão dos processos e planejamentos. Este especialista possui 33 anos de experiência na área.

Especialista 3: Atuando na área da ciência da computação, com domínio nos temas de Governança de Tecnologia da Informação, gerenciamento de serviços, definição de requisitos e arquitetura de software, este especialista apresenta 11 anos de experiência.

Especialista 4: Profissional da ciência da computação, atuando na área a 6 anos. O especialista possui maior conhecimento com desenvolvimento e engenharia de software, atuando como gerente de projetos.

Especialista 5: Advindo da área da ciência da computação, este profissional com 5 anos de experiência atua no desenvolvimento de software, sendo que sua especialidade é a gestão da qualidade.

Especialista 6: Outro profissional da ciência da computação, com 2 anos de experiência prática no mercado de trabalho. O foco de atuação deste especialista é em desenvolvimento web e mobile.

Especialista 7: Este especialista está situado na área da ciência da computação, sendo que a ênfase de sua atuação é o desenvolvimento de software. Possui 30 anos de experiência no mercado.

Especialista 8: Profissional da ciência da computação que possui 5 anos de experiência na área. Atua principalmente com o desenvolvimento de softwares para web.

Especialista 9: Este especialista também é da área da ciência da computação, atuando em diversas frentes, como análise de requisitos, testes, usabilidade, gestão de projetos e processo de software. Acumula 10 anos de experiência na indústria.

Especialista 10: Da área do design, com especial ênfase em Experiência do Usuário e design da Interface Gráfica, este especialista tem 4 anos de experiência com design de soluções gráficas para mobile e desktop.

Especialista 11: Este especialista também vem da área do design, atuando especialmente com a Experiência do Usuário e enfoque em produtos inclusivos e acessibilidade universal. Possui 5 anos de experiência na área.

Especialista 12: Designer de interface com 6 anos de experiência na indústria. Este profissional foca sua atuação no design de interface para mobile e desktop, experiência do usuário e softwares educacionais.

Especialista 13: Este profissional vem da área do design e possui 3 anos de experiência no mercado de trabalho. Atua principalmente com a pesquisa de público que sustenta o levantamento de requisitos.

Especialista 14: Designer gráfico que atua com maior ênfase no design de Interface e Experiência do Usuário, tanto para mobile quanto para web. Possui 4 anos de atuação.

Especialista 15: Este profissional atua em duas grandes frentes, sendo a ciência da computação e a educação, acumulando 4 anos e 6 meses de experiência. Na ciência da computação, sua experiência é com desenvolvimento de software para web. No campo da educação, leciona para ensino médio e técnico.

Especialista 16: Professor de ciência da computação que atua a 28 anos na área de sistemas da informação. Este profissional também desenvolve e aplica métodos para projetos de Banco de Dados em múltiplas plataformas.

Especialista 17: Vinculado à área da ciência da computação, este profissional leciona no âmbito do ensino técnico e superior. Entre os diversos cursos e programas de ensino, acumula 20 anos de experiência em sala de aula.

Como se pode notar, os especialistas possuem variados perfis e experiências prévias. O tempo de atuação no mercado de trabalho, por exemplo, varia entre 2 e 33 anos, o que indica uma considerável variação de idade entre eles. Segmentando-se por área do

conhecimento, temos 9 especialistas da ciência da computação, 5 do design de interface, e 3 da educação. Ademais, sua ênfase prática também é bastante variável, incluindo temas como análise de requisitos, gestão de projetos, gestão da qualidade, pesquisa, desenvolvimento web e mobile, design de UX e IU, entre outras. Considera-se que esta variedade de perfis seja benéfica para a verificação do *framework*, pois poderá revelar informações advindas de pontos de vista muito diferentes.

Para a condução do procedimento de verificação, primeiro se fez necessário contextualizar e explicar o funcionamento do *framework* aos especialistas. Portanto, inicialmente, se fez uma apresentação dos temas a cada um deles, começando pelo Transtorno do Espectro Autista, tópico sobre o qual muitos especialistas não tinham conhecimento profundo. Em seguida, explicou-se a problemática do trabalho, seus objetivos, justificativa e como fora conduzido. Por fim, apresentou-se o *framework* em si, incluindo sua modelagem e especificando-se os mecanismos de seu processo. Um ponto importante foi, também, a explicação de como o *framework* busca solucionar os problemas e oportunidades detectados na revisão sistemática, conduzida anteriormente.

Uma vez que os especialistas eram contextualizados e ganhavam familiaridade com o processo do *framework*, eles eram convidados a responder um breve questionário online. O questionário era composto de cinco afirmações que, em linhas gerais, verificavam preliminarmente a eficácia do processo proposto. Foram elencados apenas cinco tópicos para que os participantes não fossem sobrecarregados com uma atividade muito extensa. O participante deveria ler a afirmação, refletir sobre ela, e então dizer se concordava ou não. Para isto, foi utilizada a Escala de Likert, com uma graduação de cinco pontos, sendo que 1 representa “discordo totalmente” e 5 representa “concordo totalmente”. As afirmações eram:

- (1) Na minha opinião, o *framework* apresentado pode se integrar com êxito aos modelos de desenvolvimento atuais;
- (2) Acredito que os processos do *framework* ajudarão no desenvolvimento de softwares para autistas;
- (3) Os processos, técnicas e ferramentas do *framework* possibilitam uma definição clara do público-alvo, suas preferências e necessidades;

- (4) A utilização do *framework* potencialmente ajuda a cumprir os objetivos de aprendizagem propostos por um software;
- (5) O *framework* fornece ferramentas úteis para melhorar a qualidade dos testes e o acompanhamento pós-entrega.

Após refletir sobre as afirmações e tentar relacionar o uso do *framework* com sua própria prática em ambiente corporativo, os especialistas deveriam atribuir uma nota na Escala de Likert. As respostas obtidas foram compiladas e então analisadas. Importante ressaltar que, como se trata de um processo preliminar de verificação, os resultados sugerem apenas um *indicativo* da eficácia do *framework*. Sendo assim, o nível de rigorosidade esperado com as respostas era de ao menos 4,5 para cada uma das afirmações, aproximando-se o máximo possível do ideal “concordo totalmente”. Os dados obtidos com a verificação estão compilados na Tabela 7 a seguir.

Tabela 7. Verificação do framework por especialistas.

Especialista	Afirmação					Média por especialista
	n1	n2	n3	n4	n5	
E1	3	4	4	3	4	3,6
E2	5	5	5	5	4	4,8
E3	5	5	5	5	5	5
E4	5	5	5	4	4	4,6
E5	5	5	5	5	5	5
E6	4	5	5	5	4	4,6
E7	5	5	5	5	5	5
E8	4	4	4	4	4	4
E9	5	5	5	5	5	5
E10	5	5	5	5	5	5
E11	5	5	5	5	5	5
E12	5	5	5	4	4	4,6
E13	5	5	5	5	5	5
E14	5	5	5	5	5	5
E15	5	5	4	5	5	4,8
E16	4	4	4	3	4	3,8
E17	4	4	4	4	4	4
Média total	4,6	4,7	4,7	4,5	4,5	4,6

Os tons de cinza da Tabela 7 separam os grupos de especialistas por área do conhecimento: em cinza claro, os profissionais da ciência da computação; em cinza médio, os especialistas de design; e em cinza escuro, os professores. Na primeira coluna, apresentam-se os especialistas propriamente ditos, denominados simplesmente por *En*. O mesmo ocorre nas colunas centrais, que são correspondentes a cada uma das cinco afirmações. Na última coluna, observa-se a média do escore por especialista. A média total da verificação chegou a 4,6, superando em 0,1 ponto o nível de rigorosidade. Mas a tabela também nos fornece informações relevantes acerca dos grupos de especialistas e dos eixos de avaliação.

Separando-se a avaliação por eixo, verifica-se que as afirmações com menor nota foram a 4 e a 5. Embora não sejam escores baixos (de fato, atingem o nível mínimo de rigorosidade), revela-se que os pontos de maior fragilidade no *framework* são relacionados ao cumprimento dos objetivos de aprendizagem e aos testes e acompanhamento pós-entrega. Provavelmente, estes aspectos precisam ser estudados com maior rigor e cautela, já que as avaliações mostram uma performance mais baixa indicada pelos especialistas (sobretudo a afirmação 4, que recebeu dois escores 3). Por outro lado, a maior pontuação foi atribuída às afirmações *n2* (a respeito da confiança de que o *framework* é útil ao desenvolvimento de software TEA) e *n3* (a respeito dos processos, técnicas e ferramentas disponibilizados por ele), sugerindo melhor desempenho neste aspecto.

Ao analisar a avaliação por grupos de especialistas, novos dados são encontrados. A maior média atribuída nos grupos de especialistas foi relativa aos designers, chegando à marca de 4,9 (ou 0,4 maior que o nível de rigorosidade). Em seguida, aparecem os profissionais relacionados à ciência da computação, cuja média geral foi de 4,6 (ou 0,1 maior que o nível de rigorosidade). Por fim, aparecem os profissionais da educação, em que a média foi de 4,2, estabelecendo-se em 0,3 ponto *abaixo* do nível de rigorosidade. Este foi o grupo que contou com menos participantes, portanto é difícil concluir que a avaliação dos aspectos pedagógicos do *framework* de fato seja insuficiente. Porém, o escore abaixo do nível previsto indica, ao menos, que estes profissionais podem ser consultados para a realização de possíveis melhorias.

Por fim, o índice de aprovação do *framework* foi de 76%, correspondente a 13 especialistas cujas médias ficaram acima do nível de rigorosidade (4,5). Destes, 9 avaliadores (53% dos participantes) dos grupos de especialistas da ciência da computação e do design atribuíram nota máxima em todos os critérios avaliados, revelando um bom indicativo dos

processos e mecanismos do *framework*. Por outro lado, houveram 4 participantes cujas médias finais ficaram abaixo do nível de rigorosidade: E1 (escore 3,6); E8 (escore 4); E16 (escore 3,8); e E17 (escore 4). Novamente, um feedback mais detalhado fornecido por estes profissionais poderia sustentar futuras melhorias no *framework*. De modo geral, todavia, pode-se considerar que, em caráter preliminar, a verificação qualitativa do *framework* indica pontos positivos e potencial eficácia de aplicação.

9 CONSIDERAÇÕES FINAIS

No último capítulo deste trabalho, fazemos uma retrospectiva do que foi realizado com esta pesquisa, apresentando os objetivos alcançados, as limitações e desafios, e as perspectivas futuras. Ao longo de suas páginas, este estudo buscou apresentar, de maneira clara mas também explicativa, conceitos importantes para a sua realização. Falou-se do Transtorno do Espectro Autista, sua evolução ao longo do tempo, suas condições clínicas, diagnóstico e tratamento, para contextualizar o leitor acerca deste tema tão complexo. Discorreu-se, também, sobre aspectos de acessibilidade digital, o comportamento deste público no cenário virtual, diretrizes e práticas correntes de acessibilidade e softwares educacionais. Por fim, forneceram-se informações sobre os modelos ágeis de desenvolvimento, o design centrado no usuário e a gestão da qualidade, por meio dos Modelos de Maturidade e das Lições Aprendidas.

9.1 Objetivos Alcançados

Acredita-se que este trabalho tenha alcançado seus objetivos principais com êxito. Em primeiro lugar, a revisão sistemática, que foi conduzida com bastante rigor processual, identificou fraquezas e possibilidades de melhoria para sustentar a modelagem do *framework*. Para além na aplicação neste trabalho, os dados reunidos na revisão certamente poderão ser analisados mais a fundo e explorados por outros pesquisadores e profissionais interessados na área. A quantidade de informações presentes e o tratamento conferido aos dados possibilita, agora, um estudo muito mais rápido e direcionado das contribuições presentes na recente literatura envolvendo a temática.

A modelagem do *framework*, propriamente dita, também é um avanço a ser considerado. Apesar do recente aumento literário sobre o autismo, tanto em quantidade como em qualidade, todos os esforços em se propiciar maior autonomia e independência aos indivíduos TEA são importantes. Conforme mencionado ao longo deste trabalho, o indivíduo TEA ainda é frequentemente negligenciado, portanto a busca por caminhos e soluções metodológicas que visem à inclusão social deste público é absolutamente necessária e desejada. Esta deve ser a preocupação da ciência, das corporações, e também da sociedade civil.

Por fim, a verificação do *framework*, conduzida com especialistas das áreas da ciência da computação, design e educação, revelam uma positiva aceitação do *framework* entre profissionais com variados perfis e atuações. Os resultados obtidos indicam, preliminarmente, um desejo e uma expectativa de que o processo de fato possa ser útil em contextos reais de desenvolvimento. Portanto, esta verificação conclui os objetivos especificados para este trabalho, deixando como legado um *framework* que potencialmente servirá para sustentar a prática da acessibilidade TEA e, também, para abrir futuras discussões sobre o tema.

9.2 Limitações e Desafios

Apesar dos objetivos terem sido cumpridos e bem-sucedidos, limitações e desafios também se impuseram ao longo da pesquisa. Por um lado, ressalta-se a própria escassez de recursos e estudos disponíveis na literatura. Muito do material que se encontra refere-se ao autismo sob a perspectiva da área médica, e mesmo quando tais fontes se prestam ao desenvolvimento de software, poucas informações metodológicas são encontradas. Evidentemente, tais estudos são valiosos e têm sua contribuição, mas o público autista ainda carece de pesquisadores e profissionais interessados em solucionar demandas do processo, que se refletem em termos de qualidade, usabilidade e acessibilidade.

Ademais, o quadro clínico heterogêneo do autismo adiciona elevado grau de complexidade em se trabalhar com este público. Além das preocupações comuns referentes ao perfil do público, como idade, gênero, contexto histórico, social e cultural, o autismo exige atenção redobrada para que se contemplem seus variados sintomas, graus de severidade, preferências e necessidades específicas. Adicione-se a isto o fato de que muito sobre o autismo ainda permanece desconhecido mesmo do ponto de vista clínico, e temos um contexto de desenvolvimento absolutamente dinâmico e de difícil adaptação.

Por fim, a verificação do *framework* com especialistas também se mostrou desafiadora. Apesar da preocupação com a acessibilidade ser crescente no ambiente corporativo, é difícil encontrar profissionais que sejam conhecedores do assunto para compartilhar suas experiências e dar feedback. Neste caso, vale a pena replicar o procedimento adotado neste trabalho, que forneceu explicações clínicas sobre o autismo aos especialistas antes de iniciar a sessão de verificação. Buscar alternativas para solucionar estes obstáculos não é tarefa fácil, mas

deve ser o comprometimento da comunidade acadêmica. Afinal, novos avanços só se consolidarão com o esforço conjunto em se enfrentar estes desafios e desvendar novas possibilidades rumo à dignidade e à inclusão do público autista.

9.3 Perspectivas Futuras

Esta pesquisa não termina aqui. De fato, muitas possibilidades se abrem com os resultados obtidos por este trabalho. Por um lado, o *framework* certamente deve e há de ser melhorado, com a sua aplicação em um ambiente real de desenvolvimento, aplicado a produtos para o público autista. Isso deve levar à compreensão mais profunda de sua integração com os modelos ágeis, bem como do funcionamento de seus mecanismos frente à situações complexas e à tomada de decisões. Ademais, muitas novas descobertas e *insights* devem emergir quando o *framework* for efetivamente colocado à prova, já que neste momento haverá inegável envolvimento humano que possibilita verificações mais concretas.

Além disso, embora este *framework* tenha seu escopo especificamente voltado ao público autista, acredita-se que possa ser expandido e aplicado a outras situações e contextos. Sua abordagem de natureza humanizada e exploratória confere ferramentas e técnicas desejáveis a outras áreas envolvendo a acessibilidade e a usabilidade. Desta forma, deve ser possível adaptá-lo para se encaixar em outros tipos de processo e ambientes de desenvolvimento. Fica aqui registrado o interesse em se continuar buscando caminhos rumo à inclusão e ao desenvolvimento social. Pesquisadores e profissionais devem se unir em prol da busca por soluções que incluam o indivíduo autista, contribuindo, desta forma, não apenas com o avanço da ciência, mas de toda a humanidade.

REFERÊNCIAS

- [1] ABRAS, C.; MALONEY-KRICHMAR, D.; PREECE, J. User-centered design. In: BAINBRIDGE, W. **Encyclopedia of Human-Computer Interaction**. Thousand Oaks: Sage Publications, 2004.
- [2] AL-MUDIMIGH, A. S.; ULLAH, Z.; ALSUBAIE, T. A. A framework for portal implementation: a case for Saudi organizations. **International Journal of Information Management**, v. 31, n. 1, p. 38-43, 2011.
- [3] ALVARADO, C.; et al. Valpodijo: Developing a software that supports the teaching of Chilean idioms to children with autism spectrum disorders. **Anais da 12th Latin American Conference on Learning Technologies (LACLO) 2017**, [S.I.], 2017.
- [4] AMERICAN PSYCHIATRIC ASSOCIATION. DSM-IV. **Manual diagnóstico e estatístico de transtornos mentais**. 4. ed. Porto Alegre: ARTMED, 2002.
- [5] AMERICAN PSYCHIATRIC ASSOCIATION. DSM-V. **Manual diagnóstico e estatístico de transtornos mentais**. 5. ed. Porto Alegre: ARTMED, 2013.
- [6] ASSUMPÇÃO JR., F. B.; PIMENTEL, A. C. M. Autismo infantil. **Revista Brasileira de Psiquiatria**. [S.I.], v. 22, n. 1, p. 37-39, jan./fev./mar. 2000.
- [7] BAIO, J.; et al. Prevalence of autism spectrum disorder among children aged 8 years, autism and developmental disabilities monitoring network, 11 sites, Unites States, 2014. **Morbidity and Mortality Weekly Report (MMWR)**, v. 67, n. 6, p. 1-23, 2014.
- [8] BARROS, V. T. O.; et al. Interactive and customizable use of mobile technology in the cognitive development process of children with autistic spectrum disorder. **Anais da 8th International Conference on Universal Access in Human-Computer Interaction**, Crete, Greece, v. 1, p. 443-453, 2014.
- [9] BECK, K; et al. **Agile Manifesto**. 2001. Disponível em: <<https://agilemanifesto.org>>. Acesso em: 19 nov. 2018.
- [10] BETTELHEIM, B. **The empty fortress: Infantile autism and the birth of the self**. New York: The Free Press, 1967.
- [11] BOLANOS, S.; et al. Softwareland chronicles: a software development meta-process proposal. **Applied Computer Systems**, v. 19, n. 1, p. 5-14, 2016.

- [12] BRASIL. Presidência da República. Casa Civil. Subchefia para Assuntos Jurídicos. Decreto nº 6.949, de 25 de agosto de 2009. **Promulga a Convenção Internacional sobre os Direitos das Pessoas com Deficiência e seu Protocolo Facultativo, assinados em Nova York, em 30 de março de 2007.** Brasília, DF, 2009.
- [13] BRITTO, T. C. P. **GAIA: Uma proposta de guia de recomendações de acessibilidade web com foco em aspectos do autismo.** Dissertação (Mestrado), Universidade Federal de São Carlos, 257 p., 2016.
- [14] CABALLERO, L.; MORENO, A. M.; SEFFAH, A. How agile developers integrate user-centered design into their processes: a literature review. **International Journal of Software Engineering and Knowledge Engineering**, v. 26, n. 8, p. 1175-1201, 2016.
- [15] CABIELLES-HERNÁNDEZ, D.; et al. Specialized intervention using tablet devices for communication deficits in children with autism spectrum disorders. **Transactions on Learning Technologies**, v. 10, n. 2, p. 182-193, 2016.
- [16] CAI, Y.; et al. Design and development of a virtual dolphinarium for children with autism. **Transactions on neural systems and rehabilitation engineering**, v. 21, n. 2, p. 208-217, 2013.
- [17] CHATZARA, K.; et al. Digital storytelling for children with autism: Software development and pilot application. **Research on E-learning and ICT in Education: Technological, Pedagogical and Instructional Perspectives**, [S.I.], p. 287-300, 2014.
- [18] COHEN, D.; LINDVALL, M; COSTA, P. An introduction to agile methods. **Advances in Computers**, v. 62, n. 1, p. 1-66, 2004.
- [19] DE LEO, G.; et al. A smart-phone application and a companion website for the improvement of the communication skills of children with autism: clinical rationale, technical development and preliminary results. **Journal of Medical Systems**, v. 35, p. 703-711, 2011.
- [20] DINGSØYR, T; et al. A decade of agile methodologies: towards explaining agile software development. **Journal of Systems and Software**, v. 85, n. 6, p. 1213-1221, 2012.
- [21] FAGUNDES, P. B. **Framework para comparação e análise de métodos ágeis.** 2005. Dissertação (Mestrado em Ciência da Computação) – Departamento e Computação, Universidade Federal de Santa Catarina, Florianópolis.

- [22] FINKELSTEIN, S.; et al. Astrojumper: motivating children with autism to exercise using a VR game. **Anais da 28th ACM Conference on Human Factors in Computing Systems**, p. 4189-4194, 2010.
- [23] FLETCHER-WATSON, S. *A targeted review of computer-assisted learning for people with autism spectrum disorder: Towards a consistent methodology*. **Journal of Autism and Developmental Disorders**, [S.I.], n. 1, p. 87-100, 2014.
- [24] FOTJIK, R. Agile methodology and development of software for users with specific disorders. **International Multiconference on Computer Science and Information Technology**, [S.I.], p. 687-691, 2010.
- [25] GAIA SOLUÇÕES EM TIC. **Fábrica de projetos em tecnologia da informação e comunicação**. Disponível em: <<http://gaia.uel.br/sobre>>. Acesso em 03 de fev. 2019.
- [26] GARG, A. Agile software development. DRDO Science Spectrum, **DESIDOC Journal of Library and Information Technology**, p. 55-59, mar. 2009.
- [27] GIL, A. C. **Como elaborar projetos de pesquisa**. 4a ed. São Paulo: Atlas, 2002.
- [28] GÖRANSSON, B.; GULLIKSEN, J.; BOIVIE, I. The usability design process: integrating user-centered systems design in the software development process. **Software Process Improvement and Practice**, v. 8, n. 1, p. 111-131, 2003.
- [29] HAYES, G. R.; et al. Interactive visual supports for children with autism. **Personal and Ubiquitous Computing**, v. 14, n. 1, p. 663-680, 2010.
- [30] HORITA, F. E. A.; BARROS, R. M. *GAIA Human Resources – An approach to integrate ITIL and Maturity Levels focused on improving the Human Resources Management in Software Development*. In: **25th International Conference on Computer Applications in Industry and Engineering (CAINE 2012)**, New Orleans, Louisiana USA, v. 1, p. 51-56, 2012.
- [31] HOURCADE, J. P.; BULLOCK-REST, N. E.; HANSEN, T. E. Multitouch tablet applications and activities to enhance the social skills of children with autism spectrum disorders. **Personal and Ubiquitous Computing**, v. 16, n. 1, p. 157-168, 2012.
- [32] HULUSIC, V.; PISTOLJEVIC, N. “LeFCA”: *Learning framework for children with autism*. **Procedia Computer Science**, [S.I.], n. 15, p. 4-16, 2012.
- [33] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). **ISO/IEC 71:2014 Guide for addressing accessibility in standards**, Suíça: ISO/IEC, 2014.

- [34] IRADAH, I. S.; RABIAH, A. K. EduTism: An assistive educational system for the treatment of autism children with intelligent approach. **Anais da International Visual Informatics Conference**, [S.I.], p. 193-204, 2011.
- [35] JOHNSON, C. P.; MYERS, S. M. Identification and evaluation of children with Autism Spectrum Disorders. **Pediatrics**. [S.I.], v. 120, n. 5, p. 1183-1215, nov. 2007.
- [36] JOKELA, T.; et al. A survey of usability capability maturity models: implications for practice and research. **Behavior and Information Technology**, v. 25, n. 3, p. 263-282, 2006.
- [37] KAMARUZAMAN, M. F.; et al. *Developing user interface design application for children with autism*. **Procedia Social and Behavioral Sciences**, [S.I.], n. 217, p. 887-894, 2016.
- [38] KANNER, L. Autistic disturbances of affective contact. **Nervous Child**. [S.I.], v. 2, n. 25, p. 217-250, 1943.
- [39] KHOWAJA, K.; SALIM, S. S. Heuristics to evaluate interactive systems for children with Autistic Spectrum Disorder (ASD). **PLoS ONE**, v. 10, n. 7, p. 1-27, 2015.
- [40] KITCHENHAM, B. **Procedures for performing systematic reviews**. Kelee, Reino Unido, Kelee University, v. 33, p. 1-26, 2004.
- [41] KLIN, A. Autismo e síndrome de Asperger. **Revista Brasileira de Psiquiatria**. São Paulo, v. 28, n. 1, p. 3-11, maio 2006.
- [42] KOLB, A. Y.; KOLB, D. A. **The Kolb learning style inventory 4.0: a comprehensive guide to the theory, psychometrics, research on validity and educational applications**. Experience Based Learning Systems, Estados Unidos, 2013.
- [43] KONSTANTINIDIS, E. I.; et al. Using effective avatars and rich multimedia content for education of children with autism. **Pervasive Technologies Related to Assistive Environments (PETRA) 2009**, [S.I.], 2009.
- [44] KUNDA, M.; GOEL, A. K. *Thinking in pictures as a cognitive account of autism*. **Journal of Autism and Developmental Disorders**, [S.I.], v. 41, p. 1157-1177, 2011.
- [45] KUO, M. H.; et al. *Media use among adolescents with autism spectrum disorder*. **Autism, Connecticut, United States**, v. 18, n. 8, p. 914-923, 2014.
- [46] KURNIAWAN, R.; et al. Development of game for self-help toilet learning for children with autism. **CommIT (Communication and Information Technology) Journal**, v. 12, n. 1, p. 1-12, 2018.

- [47] LACERDA, T. C.; WANGENHEIM, C. G. Systematic literature review of usability capability / maturity models. **Computer Standards and Interfaces**, v. 55, n. 1, p. 95-105, 2018.
- [48] LIMA, D.; CASTRO, T. *Music spectrum: A music immersion virtual environment for children with autism*. **Procedia Computer Science**, [S.I.], n. 14, p. 111-118, 2012.
- [49] MACDONALD, M.; ESPOSITO, P.; ULRICH, D. *The physical activity patterns of children with autism*. **BMC Research Notes**, [S.I.], v. 4, n. 442, p. 1-5, 2011.
- [50] MALINVERNI, L.; et al. An inclusive design approach for developing video games for children with autism spectrum disorder. **Computers in Human Behavior**, v. 71, n. 1, p. 535-549, 2017.
- [51] MAZUREK, M. O. *Social media use among adults with autism spectrum disorder*. **Computers in Human Behavior**, [S.I.], v. 29, p. 1709-1714, 2013.
- [52] MAZUREK, M. O.; ENGELHARDT, C. R. *Video game use in boys with autism spectrum disorder, ADHD, or typical development*. **Pediatrics**, [S.I.], v. 132, p. 260-266, jul. 2013.
- [53] MAZUREK, M. O.; et al. *Prevalence and correlates of screen-based media use among youths with autism spectrum disorder*. **Journal of Autism and Developmental Disorders**, [S.I.], v. 42, p. 1757-1767, 2012.
- [54] MAZUREK, M. O.; WENSTRUP, C. *Television, video game and social media use among children with ASD and typically developing siblings*. **Journal of Autism and Developmental Disorders**, [S.I.], v. 43, p. 1258-1271, 2013.
- [55] MOSTAFA, D. S. D. N. **Maturity models in the context of integrating agile development processes and user centred design**. 2013. Tese (Doctor in Philosophy in Computer Science) – University of York, York, Reino Unido.
- [56] ORGANIZAÇÃO DAS NAÇÕES UNIDAS. *Convention on the Rights of Persons with Disabilities [A/61/611]. Annex 1, Final Report on the Ad Hoc Committee on a Comprehensive and Integral International Convention on the Protection and Promotion of the Rights and Dignity of Persons with Disabilities*, dez. 2006.
- [57] ORGANIZAÇÃO MUNDIAL DA SAÚDE. **CID-10 Classificação Estatística Internacional de Doenças e Problemas Relacionados à Saúde**. 10a rev. São Paulo: Universidade de São Paulo; v. 1, 1997.

- [58] ORRÚ, S. E. A formação de professores e a educação de autistas. **Revista Iberoamericana de Educación (Online)**. Espanha, v. 31, n. 1, p. 1-15, 2003.
- [59] PAULK, M.; et al. Capability maturity model: version 1.1. **Journal IEEE Software**, v. 10, n. 4, p. 18-27, 1993.
- [60] PAYÁ, M. A.; NAVARRO, E. M. Accesibilidad universal: sentido normativo e implicaciones en la educación y la práctica profesional. **Revista Española de Discapacidad**, v. 5, n. 1, p. 25-41.
- [61] PEREIRA, A. M. **Autismo infantil**: Tradução e validação da CARS (Childhood Autism Rating Scale) para uso no Brasil. Porto Alegre: UFRGS, 2007. 99 p. Dissertação (Mestrado) – Programa de Pós-graduação em Ciências Médicas: Pediatria, Faculdade de Medicina, Universidade Federal do Rio Grande do Sul, 2007.
- [62] RAMDOSS, S.; et al. *Use of computer-based interventions to teach communication skills to children with autism spectrum disorders: A systematic review*. **Journal of Behavioral Education**, [S.I.], v. 20, p. 55-76, 2011.
- [63] RAUTENBERG, S.; STEIL, A. V.; TODESCO, J. L. Modelo de conhecimento para mapeamento de instrumentos da gestão do conhecimento e de agentes computacionais da engenharia do conhecimento. **Perspectivas em Ciência da Informação**, v. 16, n. 3, p. 26-46, 2011.
- [64] RUTTER, M. Diagnosis and definitions of childhood autism. **Journal of Autism and Childhood Schizophrenia**. [S.I.], v. 8, n. 2, p. 139-161, jun. 1978.
- [65] SHMINAN, A. S.; et al. AntiPECS: Mobile based learning of picture exchange communication intervention for caregivers of autistic children. **International Conference on Computer and Drone Applications**, [S.I.], p. 49-54, 2017.
- [66] SIGNORE, A.; BALASI, P.; YUAN, T. You Talk! – you vs. autism. **Anais da 16th International Conference on Computers Helping People with Special Needs ICCHP 2014**, [S.I.], p. 506-512, 2014.
- [67] SILVA, C. A.; FERNANDES, A. R.; GROHMANN, A. P. STAR: Speech therapy with augmented reality for children with autism spectrum disorders. **Anais da 16th International Conference on Enterprise Information Systems**, p. 379-396, 2014.

- [68] TAN, C. T.; HARROLD, N.; ROSSER, D. Can you CopyMe?: an expression mimicking serious game. **Anais da 40th International Conference on Computer Graphics and Interactive Techniques SIGGRAPH 2013**, [S.I.], 2013.
- [69] TSIOPELA, D.; JIMOYIANNIS, A. Pre-vocational skills laboratory: designing interventions to improve employment skills for students with autism spectrum disorders. **Universal Accessibility in the Information Society**, v. 16, n. 1, p. 609-627, 2016.
- [70] VENKATESH, S.; et al. TOBY: Early intervention in autism through technology. **Conference on Human Factors and Computing Systems – CHI 2013**, [S.I.], p. 3187-3196, 2013.
- [71] WADHWA, B.; JIANXIONG, C. Collaborative tablet applications to enhance language skills of children with autism spectrum disorder. **Anais da 11th Asia Pacific Conference on Computer Human-Interaction – APCHI 2013**, [S.I.], p. 39-44, 2013.
- [72] WASHINGTON, P.; et al. SuperpowerGlass: a wearable aid for the at-home therapy of children with autism. **Anais da ACM on Interactive, Mobile, Wearable and Ubiquitous Technology**, v. 1, n. 3, 2017.
- [73] WASS, S. V.; PORAYSKA-POMSTA, K. The uses of cognitive training technologies in the treatment of autism spectrum disorders. **Autism**, v. 18, n. 8, p. 851-871, 2014.
- [74] WING, L. Asperger's syndrome: a clinical account. **Psychological medicine**. Grã-bretanha, v. 11, n. 1, p. 115-129, fev. 1981.
- [75] WINOTO, P.; CAO, V. L.; TANG, E. M. A highly customizable parent-child word-learning mobile game for Chinese children with autism. **Anais da 11th International Conference on Universal Access in Human-Computer Interaction**, [S.I.], p. 545-554, 2017.
- [76] WINOTO, P.; XU, C. N.; ZHU, A. A. "Look to remove": a virtual reality application on word learning for Chinese children with autism. **Anais da 10th International Conference on Universal Access in Human-Computer Interaction**, Toronto, Canada, p. 257-264, 2016.
- [77] WOJCIECHOWSKI, A.; AL-MUSAWI, R. Assistive technology application for enhancing social and language skills of young children with autism. **Multimedia Tools and Applications**, v. 76, n. 1, p. 5419-5439, 2017.
- [78] WORLD WIDE WEB CONSORTIUM (W3C). **Web content accessibility guidelines (WCAG) 2.0**. CALDWELL, B.; et al. (Eds.). W3C Recommendation, 2008.

- [79] YAN, F. A sunny day: Ann and Ron's world an iPad application for children with autism. **International Conference on Serious Games Development and Applications**, [S.I.], p. 129-138, 2011.
- [80] ZAFFKE, A.; et al. iCanLearn: a mobile application for creating flashcards and social stories for children with autism. **Anais da International Conference on Smart Homes and Health Telematics**, [S.I.], p. 225-230, 2014.
- [81] ZHANG, L.; et al. Design and evaluation of a collaborative virtual environment (CoMove) for autism spectrum disorder intervention. **Transactions on Accessible Computing**, v. 11, n. 2, 2018.

Apêndices

4. Estilos de aprendizagem	
Estilo	Descrição

5. Objetivos de aprendizagem	
Objetivo	Descrição

Informações adicionais
Observações

APÊNDICE B – Relatório de UX E IU

Relatório de Experiência do Usuário e Interface do Usuário (Modelo)		
Empresa		Data
Responsável		Versão
1. Objetivos de aprendizagem		
Breve descrição		
2. Estratégias de Aprendizagem		
Estratégia	Descrição	Objetivo de aprendizagem relacionado

3. Técnicas de aprendizagem		
Técnica	Descrição	Estilo de aprendizagem relacionado
4. Métodos de representação		
Método	Descrição	

5. Métodos de decoração		
Método	Descrição	
6. Diretrizes gráficas		
Descrição		
Informações adicionais		
Observações		

APÊNDICE C – Questionário Pós-entrega

Questionário Pós-entrega (modelo)		
Projeto	Responsável	
Respondente	Data	
1. Durante o tempo em que o software está em uso, ele apresentou algum problema de funcionamento? Se sim, quais?		
Resposta		
2. Os gerenciadores do software (pais, professores, terapeutas) tiveram dificuldades em entender como o software funcionava? Quais dificuldades?		
Resposta		
3. Os usuários do software (indivíduos autistas) tiveram dificuldades em entender como o software funcionava? Quais dificuldades?		
Resposta		
4. Houve interesse por parte do público em utilizar o software? Descreva.		
Resposta		
5. Houve interesse em continuar utilizando o software após um período de aplicação?		
Resposta		
6. Houve resistência quanto ao uso do software por parte de profissionais, pais ou autistas? De que tipo?		
Resposta		
7. Você sente que o software respeita o nível de aprendizagem dos indivíduos autistas?		
Resposta		
8. Você acha que os objetivos de aprendizagem estão sendo cumpridos? Descreva.		
Resposta		
9. Houve feedback (positivos ou negativos) por parte dos usuários?		
Resposta		

10. Você acha que o software respeita as dificuldades e sintomas dos indivíduos autistas? Caso não, o que poderia ser mudado?
Resposta
11. Você acredita que o software têm tido resultados positivos? Descreva.
Resposta
12. Você sente que houve uma aproximação maior entre você / sua equipe e seu público após o início da aplicação do software?
Resposta
13. Você acha que as tecnologias usadas no software ajudam a aumentar a motivação e independência dos indivíduos autistas durante seu uso?
Resposta
14. Você está satisfeito com o software no geral? Caso não, descreva.
Resposta
15. Você tem alguma sugestão, reclamação ou elogio a fazer quanto ao tempo em que a equipe produziu o software? (em termos de cronograma, prazos, orçamento, comunicação, etc.)
Resposta

APÊNDICE D – Registro de Lições Aprendidas

Registro de Lições aprendidas (modelo)		
Projeto	Autor	
Departamento	Data	
1. Contexto (breve descrição do projeto)		
Descrição		
2. Procedimento (o que foi feito?)		
Descrição		
3. Lições aprendidas (resultados obtidos, sejam positivos ou negativos)		
4. Sugestões de melhoria (o que pode ser feito melhor da próxima vez?)		
Observações gerais		

Anexos

ANEXO A – Childhood Autism Rating Scale (CARS)

CARS-Childhood Autism Rating Scale VERSÃO EM PORTUGUÊS

I. RELAÇÕES PESSOAIS	
1	Nenhuma evidência de dificuldade ou anormalidade nas relações pessoais: O comportamento da criança é adequado à sua idade. Alguma timidez, nervosismo ou aborrecimento podem ser observados quando é dito à criança o que fazer, mas não em grau atípico.
1.5	
2	Relações levemente anormais: A criança pode evitar olhar o adulto nos olhos, evitar o adulto ou ter uma reação exagerada se a interação é forçada, ser excessivamente tímido, não responder ao adulto como esperado ou agarrar-se ao pai um pouco mais que a maioria das crianças da mesma idade
2.5	
3	Relações moderadamente anormais: Às vezes, a criança demonstra indiferença (parece ignorar o adulto). Outras vezes, tentativas persistentes e vigorosas são necessárias para se conseguir a atenção da criança. O contato iniciado pela criança é mínimo.
3.5	
4	Relações gravemente anormais: A criança está constantemente indiferente ou inconsciente ao que o adulto está fazendo. Ela quase nunca responde ou inicia contato com o adulto. Somente a tentativa mais persistente para atrair a atenção tem algum efeito.
	Observações:
II. IMITAÇÃO	
1	Imitação adequada: A criança pode imitar sons, palavras e movimentos, os quais são adequados para o seu nível de habilidade.
1.5	
2	Imitação levemente anormal: Na maior parte do tempo, a criança imita comportamentos simples como bater palmas ou sons verbais isolados; ocasionalmente imita somente após estimulação ou com atraso.
2.5	
3	Imitação moderadamente anormal: A criança imita apenas parte do tempo e requer uma grande dose de persistência ou ajuda do adulto; freqüentemente imita apenas após um tempo (com atraso).
3.5	
4	Imitação gravemente anormal: A criança raramente ou nunca imita sons, palavras ou movimentos mesmo com estímulo e assistência.
	Observações:

III. RESPOSTA EMOCIONAL	
1	Resposta emocional adequada à situação e à idade: A criança demonstra tipo e grau adequados de resposta emocional, indicada por uma mudança na expressão facial, postura e conduta.
1.5	
2	Resposta emocional levemente anormal: A criança ocasionalmente apresenta um tipo ou grau inadequados de resposta emocional. As vezes, suas reações não estão relacionadas a objetos ou a eventos ao seu redor.
2.5	
3	Resposta emocional moderadamente anormal: A criança demonstra sinais claros de resposta emocional inadequada (tipo ou grau). As reações podem ser bastante inibidas ou excessivas e sem relação com a situação; pode fazer caretas, rir ou tornar-se rígida até mesmo quando não estejam presentes objetos ou eventos produtores de emoção.
3.5	
4	Resposta emocional gravemente anormal: As respostas são raramente adequadas a situação. Uma vez que a criança atinja um determinado humor, é muito difícil alterá-lo. Por outro lado, a criança pode demonstrar emoções diferentes quando nada mudou.
	Observações:
IV. USO CORPORAL	
1	Uso corporal adequado à idade: A criança move-se com a mesma facilidade, agilidade e coordenação de uma criança normal da mesma idade.
1.5	
2	Uso corporal levemente anormal: Algumas peculiaridades podem estar presentes, tais como falta de jeito, movimentos repetitivos, pouca coordenação ou a presença rara de movimentos incomuns
2.5	
3	Uso corporal moderadamente anormal: Comportamentos que são claramente estranhos ou incomuns para uma criança desta idade podem incluir movimentos estranhos com os dedos, postura peculiar dos dedos ou corpo, olhar fixo, beliscar o corpo, auto-agressão, balanceio, girar ou caminhar nas pontas dos pés.
3.5	
4	Uso corporal gravemente anormal: Movimentos intensos ou freqüentes do tipo listado acima são sinais de uso corporal gravemente anormal. Estes comportamentos podem persistir apesar das tentativas de desencorajar as crianças a fazê-los ou de envolver a criança em outras atividades.
	Observações:
V. USO DE OBJETOS	
1	Uso e interesse adequados por brinquedos e outros objetos: A criança demonstra interesse normal por brinquedos e outros objetos adequados para o seu nível de habilidade e os utiliza de maneira adequada.
1.5	
2	Uso e interesse levemente inadequados por brinquedos e outros objetos: A criança pode demonstrar um interesse atípico por um brinquedo ou brincar com ele de forma inadequada, de um modo pueril (exemplo: batendo ou sugando o brinquedo)
2.5	
3	Uso e interesse moderadamente inadequados por brinquedos e outros objetos: A criança pode demonstrar pouco interesse por brinquedos ou outros objetos, ou pode estar preocupada em usá-los de maneira estranha. Ela pode concentrar-se em alguma parte insignificante do

3.5	brinquedo, tornar-se fascinada com a luz que reflete do mesmo, repetitivamente mover alguma parte do objeto ou exclusivamente brincar com ele.
4	Uso e interesse gravemente inadequados por brinquedos e outros objetos: A criança pode engajar-se nos mesmos comportamentos citados acima, porém com maior frequência e intensidade. É difícil distrair a criança quando ela está engajada nestas atividades inadequadas.
	Observações:

VI. RESPOSTA A MUDANÇAS

1	Respostas à mudança adequadas a idade: Embora a criança possa perceber ou comentar as mudanças na rotina, ela é capaz de aceitar estas mudanças sem angústia excessiva.
1.5	
2	Respostas à mudança adequadas à idade levemente anormal: Quando um adulto tenta mudar tarefas, a criança pode continuar na mesma atividade ou usar os mesmos materiais.
2.5	
3	Respostas à mudança adequadas à idade moderadamente anormal: A criança resiste ativamente a mudanças na rotina, tenta continuar sua antiga atividade e é difícil de distraí-la. Ela pode tornar-se infeliz e zangada quando uma rotina estabelecida é alterada.
3.5	
4	Respostas à mudança adequadas à idade gravemente anormal: A criança demonstra reações graves às mudanças. Se uma mudança é forçada, ela pode tornar-se extremamente zangada ou não disposta a ajudar e responder com acessos de raiva.
	Observações:

VII. RESPOSTA VISUAL

1	Resposta visual adequada: O comportamento visual da criança é normal e adequado para sua idade. A visão é utilizada em conjunto com outros sentidos como forma de explorar um objeto novo.
1.5	
2	Resposta visual levemente anormal: A criança precisa, ocasionalmente, ser lembrada de olhar para os objetos. A criança pode estar mais interessada em olhar espelhos ou luzes do que o fazem seus pares, pode ocasionalmente olhar fixamente para o espaço, ou pode evitar olhar as pessoas nos olhos.
2.5	
3	Resposta visual moderadamente anormal: A criança deve ser lembrada frequentemente de olhar para o que está fazendo, ela pode olhar fixamente para o espaço, evitar olhar as pessoas nos olhos, olhar objetos de um ângulo incomum ou segurar os objetos muito próximos aos olhos.
3.5	
4	Resposta visual gravemente anormal: A criança evita constantemente olhar para as pessoas ou para certos objetos e pode demonstrar formas extremas de outras peculiaridades visuais descritas acima.
	Observações:

VIII. RESPOSTA AUDITIVA

1	Respostas auditivas adequadas para a idade: O comportamento auditivo da criança é normal e adequado para idade. A audição é utilizada junto com outros sentidos.
1.5	
2	Respostas auditivas levemente anormal: Pode haver ausência de resposta ou uma resposta levemente exagerada a certos sons. Respostas a sons podem ser atrasadas e os sons podem necessitar de repetição para prender a atenção da criança. A criança pode ser distraída por sons externos.
2.5	
3	Respostas auditivas moderadamente anormal: As repostas da criança aos sons variam. Frequentemente ignora o som nas primeiras vezes em que é feito. Pode assustar-se ou cobrir as orelhas ao ouvir alguns sons do cotidiano.
3.5	
4	Respostas auditivas gravemente anormal: A criança reage exageradamente e/ou ou despreza sons num grau extremamente significativo, independente do tipo de som.
	Observações:

IX. RESPOSTA E USO DO PALADAR, OLFATO E TATO

1	Uso e reposta normais do paladar, olfato e tato: A criança explora novos objetos de um modo adequado a sua idade, geralmente sentindo ou olhando. Paladar ou olfato podem ser usados quando adequados. Ao reagir a pequenas dores do dia-a-dia, a criança expressa desconforto mas não reage exageradamente.
1.5	
2	Uso e reposta levemente anormais do paladar, olfato e tato: A criança pode persistir em colocar objetos na boca; pode cheirar ou provar/experimentar objetos não comestíveis. Pode ignorar ou ter reação levemente exagerada à uma dor mínima, para a qual uma criança normal expressaria somente desconforto.
2.5	
3	Uso e resposta moderadamente anormais do paladar, olfato e tato: A criança pode estar moderadamente preocupada em tocar, cheirar ou provar objetos ou pessoas. A criança pode reagir demais ou muito pouco.
3.5	
4	Uso e resposta gravemente anormais do paladar, olfato e tato: A criança está preocupada em cheirar, provar e sentir objetos, mais pela sensação do que pela exploração ou uso normal dos objetos. A criança pode ignorar completamente a dor ou reagir muito fortemente a desconfortos leves.
	Observações:

X. MEDO OU NERVOSISMO

1	Medo ou nervosismo normais: O comportamento da criança é adequado tanto à situação quanto à idade
1.5	
2	Medo ou nervosismo levemente anormais: A criança ocasionalmente demonstra muito ou pouco medo ou nervosismo quando comparada às reações de uma criança normal da mesma idade e em situação semelhante.
2.5	

3	Medo ou nervosismo moderadamente anormais: A criança demonstra bastante mais ou bastante menos medo do que seria típico para uma criança mais nova ou mais velha em uma situação similar.
3.5	
4	Medo ou nervosismo gravemente anormais: Medos persistem mesmo após experiências repetidas com eventos ou objetos inofensivos. É extremamente difícil acalmar ou confortar a criança. A criança pode, por outro lado, falhar em demonstrar consideração adequada aos riscos que outras crianças da mesma idade evitam.
	Observações:
	XI. COMUNICAÇÃO VERBAL
1	Comunicação verbal normal, adequada a idade e à situação.
1.5	
2	Comunicação verbal levemente anormal: A fala demonstra um atraso global. A maior parte do discurso tem significado; porém, alguma ecolalia ou inversão pronominal podem ocorrer. Algumas palavras peculiares ou jargões podem ser usados ocasionalmente.
2.5	
3	Comunicação verbal moderadamente anormal: A fala pode estar ausente. Quando presente, a comunicação verbal pode ser uma mistura de alguma fala significativa e alguma linguagem peculiar, tais como jargão, ecolalia ou inversão pronominal. As peculiaridades na fala significativa podem incluir questionamentos excessivos ou preocupação com algum tópico em particular.
3.5	
4	Comunicação verbal gravemente anormal: Fala significativa não é utilizada. A criança pode emitir gritos estridentes e infantis, sons animais ou bizarros, barulhos complexos semelhantes à fala, ou pode apresentar o uso bizarro e persistente de algumas palavras reconhecíveis ou frases.
	Observações:
	XII. COMUNICAÇÃO NÃO-VERBAL
1	Uso normal da comunicação não-verbal adequado à idade e situação
1.5	
2	Uso da comunicação não-verbal levemente anormal: Uso imaturo da comunicação não-verbal; a criança pode somente apontar vagamente ou esticar-se para alcançar o que quer, nas mesmas situações nas quais uma criança da mesma idade pode apontar ou gesticular mais especificamente para indicar o que deseja.
2.5	
3	Uso da comunicação não-verbal moderadamente anormal: A criança geralmente é incapaz de expressar suas necessidades ou desejos de forma não verbal, e não consegue compreender a comunicação não-verbal dos outros.
3.5	
4	Uso da comunicação não-verbal gravemente anormal: A criança utiliza somente gestos bizarros ou peculiares, sem significado aparente, e não demonstra nenhum conhecimento do significados associados aos gestos ou expressões faciais dos outros.
	Observações:

XIII. NÍVEL DE ATIVIDADE

- | | |
|---|---|
| 1
1.5
2
2.5
3
3.5
4 | <p>Nível de atividade normal para idade e circunstâncias: A criança não é nem mais nem menos ativa que uma criança normal da mesma idade em uma situação semelhante.</p> <p>Nível de atividade levemente anormal: A criança pode tanto ser um pouco irrequieta quanto um pouco “preguiçosa”, apresentando, algumas vezes, movimentos lentos. O nível de atividade da criança interfere apenas levemente no seu desempenho.</p> <p>Nível de atividade moderadamente anormal: A criança pode ser bastante ativa e difícil de conter. Ela pode ter uma energia ilimitada ou pode não ir prontamente para a cama à noite. Por outro lado, a criança pode ser bastante letárgica e necessitar de um grande estímulo para mover-se.</p> <p>Nível de atividade gravemente anormal: A criança exibe extremos de atividade ou inatividade e pode até mesmo mudar de um extremo ao outro.</p> |
|---|---|

Observações:

XIV. NÍVEL E CONSISTÊNCIA DA RESPOSTA INTELECTUAL

- | | |
|---|--|
| 1
1.5
2
2.5
3
3.5
4 | <p>A inteligência é normal e razoavelmente consistente em várias áreas: A criança é tão inteligente quanto crianças típicas da mesma idade e não tem qualquer habilidade intelectual ou problemas incomuns.</p> <p>Funcionamento intelectual levemente anormal: A criança não é tão inteligente quanto crianças típicas da mesma idade; as habilidades apresentam-se razoavelmente regulares através de todas as áreas.</p> <p>Funcionamento intelectual moderadamente anormal: Em geral, a criança não é tão inteligente quanto uma típica criança da mesma idade, porém, a criança pode funcionar próximo do normal em uma ou mais áreas intelectuais.</p> <p>Funcionamento intelectual gravemente anormal: Embora a criança geralmente não seja tão inteligente quanto uma criança típica da mesma idade, ela pode funcionar até mesmo melhor que uma criança normal da mesma idade em uma ou mais áreas.</p> |
|---|--|

Observações:

XV. IMPRESSÕES GERAIS	
1	Sem autismo: a criança não apresenta nenhum dos sintomas característicos do autismo.
1.5	
2	Autismo leve: A criança apresenta somente um pequeno número de sintomas ou somente um grau leve de autismo.
2.5	
3	Autismo moderado: A criança apresenta muitos sintomas ou um grau moderado de autismo.
3.5	
4	Autismo grave: a criança apresenta inúmeros sintomas ou um grau extremo de autismo
Observações:	

Escore por categoria

I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV	XV	Total

Resultado:

15-30: sem autismo

30-36: autismo leve-moderado

36-60: autismo grave

ESCALA DE AVALIAÇÃO GLOBAL DO FUNCIONAMENTO

Código	Descrição do Funcionamento
91 – 100	A pessoa não tem problemas OU tem funcionamento superior em uma ampla faixa de atividades OU é admirado e procurado por outros em vista de suas qualidades positivas
81 – 90	Sintomas ausentes ou mínimos (p.ex., leve ansiedade antes de um exame). Bom funcionamento em diversas áreas. Nada além de problemas ou preocupações passageiras.
71 – 80	Sintomas se estão presentes, eles são temporários e consistem de reações previsíveis a estressores psicossociais (por ex., dificuldade para concentrar-se após discussão em família); não mais do que leve prejuízo no funcionamento social, ocupacional ou escolar (por ex., apresenta declínio temporário na escola)
61 – 70	Alguns sintomas leves (por ex., humor depressivo e insônia leve) OU alguma dificuldade no funcionamento social, ocupacional ou escolar (por ex., faltas injustificadas à escola ocasionalmente, ou furto dentro de casa), mas geralmente funcionando muito bem; possui alguns relacionamentos interpessoais significativos
51 – 60	Sintomas moderados (por ex., afeto embotado e fala circunstancial, ataques de pânico ocasionais) OU dificuldade moderada no funcionamento social, ocupacional ou escolar (por ex., poucos amigos, conflitos com companheiros ou colegas de trabalho).
41 – 50	Sintomas sérios (por ex., ideação suicida, rituais obsessivos graves, freqüentes furtos em lojas) OU qualquer prejuízo sério no funcionamento social, ocupacional ou escolar (por ex., nenhum amigo, incapaz de manter um emprego)
31 – 40	Algun prejuízo no teste da realidade ou comunicação (por ex., fala às vezes ilógica, obscura ou irrelevante) OU prejuízo importante em diversas áreas, tais como emprego ou escola, relações familiares, julgamento, pensamento ou humor (por ex., homem deprimido evita amigos, negligencia a família e é incapaz de trabalhar; criança freqüentemente bate em crianças mais jovens, é desafiadora em casa e está indo mal na escola).
21 – 30	Comportamento é consideravelmente influenciado por delírios ou alucinações OU sério prejuízo na comunicação ou julgamento (por ex., ocasionalmente incoerente, age de forma grosseiramente inapropriada, preocupação suicida) OU incapacidade para funcionar na maioria das áreas (por ex., permanece na cama o dia inteiro; sem emprego, casa ou amigos).
11 – 20	Algun perigo de ferir a si mesmo ou a outros (por ex., tentativas de suicídio sem clara expectativa de morte; freqüentemente violento; excitação maníaca) OU ocasionalmente falha ao manter a higiene pessoal mínima (por ex., suja-se de fezes) OU prejuízo grosseiro na comunicação (por ex., amplamente incoerente ou mudo).
1 – 10	Perigo persistente de ferir gravemente a si mesmo ou a outros (por ex., violência recorrente) OU incapacidade persistente para manter uma higiene pessoal mínima OU sério ato suicida com clara expectativa de morte.

ANEXO B – Learning Style Inventory

This questionnaire will probably take you about 10 minutes to complete. The accuracy of your results depends on how honest you are. There are no right or wrong answers. **If you agree more than you disagree with a statement, place a tick (✓) in the box to the left of the question.** If you disagree more than you agree, leave the box blank. If you find yourself wondering which situation to think of when answering a question, just think about how you are when you are working with people. Go with your first gut reaction instead of over-thinking your response.

QUESTIONS

- ☐ 1. I have strong beliefs about what is right and wrong, good and bad.
- ☐ 2. I often act without considering the possible consequences.
- ☐ 3. I tend to solve problems using a step-by-step approach.
- ☐ 4. I believe that formal procedures and policies restrict people.
- ☐ 5. I have a reputation for saying what I think, simply and directly.
- ☐ 6. I often find that actions based on feelings are as sound as those based on careful thought and analysis.
- ☐ 7. I like the sort of work where I have time for thorough preparation and implementation.
- ☐ 8. I regularly question people about their basic assumptions.
- ☐ 9. What matters most is whether something works in practice.
- ☐ 10. I actively seek out new experiences.
- ☐ 11. When I hear about a new idea or approach, I immediately start working out how to apply it in practice.
- ☐ 12. I am keen on self discipline such as watching my diet, taking regular exercise, sticking to a fixed routine, etc.
- ☐ 13. I take pride in doing a thorough job.
- ☐ 14. I get on best with logical, analytical people and less well with spontaneous, 'irrational' people.

- ☐ 15. I take care over how I interpret data and avoid jumping to conclusions.
- ☐ 16. I like to reach a decision carefully after weighing up many alternatives.
- ☐ 17. I am attracted more to novel, unusual ideas than to practical ones.
- ☐ 18. I don't like disorganised things and prefer to fit things into a coherent pattern.
- ☐ 19. I accept and stick to laid down procedures and policies so long as I regard them as an efficient way of getting the job done.
- ☐ 20. I like to relate my actions to a general principle, standard or belief.
- ☐ 21. In discussions, I like to get straight to the point.
- ☐ 22. I tend to have distant, rather formal relationships with people at work.
- ☐ 23. I thrive on the challenge of tackling something new and different.
- ☐ 24. I enjoy fun-loving spontaneous people.
- ☐ 25. I pay careful attention to detail before coming to a conclusion.
- ☐ 26. I find it difficult to produce ideas on impulse.
- ☐ 27. I believe in coming to the point immediately.
- ☐ 28. I am careful not to jump to conclusions too quickly.
- ☐ 29. I prefer to have as many sources of information as possible – the more information to think over the better.
- ☐ 30. Flippant, superficial people who don't take things seriously enough usually irritate me.
- ☐ 31. I listen to other people's points of view before putting my own view forward.
- ☐ 32. I tend to be open about how I'm feeling.
- ☐ 33. In discussions, I enjoy watching the plotting and scheming of the other participants.
- ☐ 34. I prefer to respond to events in a spontaneous, flexible way rather than plan things out in advance.
- ☐ 35. I tend to be attracted to techniques such as flow charts, contingency plans etc.
- ☐ 36. It worries me if I have to rush work to meet a tight deadline.

- ☐ 37. I tend to judge people's ideas on their practical merits.
- ☐ 38. Quiet, thoughtful people tend to make me feel uneasy.
- ☐ 39. I often get irritated by people who want to rush things.
- ☐ 40. It is more important to enjoy the present moment than to think about the past or future.
- ☐ 41. I think that decisions based on a careful analysis of all the information are better than those based on intuition.
- ☐ 42. I tend to be a perfectionist.
- ☐ 43. In discussions, I usually produce lots of spontaneous ideas.
- ☐ 44. In meetings, I put forward practical, realistic ideas.
- ☐ 45. More often than not, rules are there to be broken.
- ☐ 46. I prefer to stand back from a situation and consider all the perspectives.
- ☐ 47. I can often see inconsistencies and weaknesses in other people's arguments.
- ☐ 48. On balance I talk more than I listen.
- ☐ 49. I can often see better, more practical ways to get things done.
- ☐ 50. I think written reports should be short and to the point.
- ☐ 51. I believe that rational, logical thinking should win the day.
- ☐ 52. I tend to discuss specific things with people rather than engaging in social discussion.
- ☐ 53. I like people who approach things realistically rather than theoretically.
- ☐ 54. In discussions, I get impatient with irrelevant issues and digressions.
- ☐ 55. If I have a report to write, I tend to produce lots of drafts before settling on the final version.
- ☐ 56. I am keen to try things out to see if they work in practice.

- ☐ 57. I am keen to reach answers via a logical approach.
- ☐ 58. I enjoy being the one that talks a lot.
- ☐ 59. In discussions, I often find I am a realist, keeping people to the point and avoiding wild speculations.
- ☐ 60. I like to ponder many alternatives before making up my mind.
- ☐ 61. In discussions with people I often find I am the most dispassionate and objective.
- ☐ 62. In discussions I'm more likely to adopt a 'low profile' than to take the lead and do most of the talking.
- ☐ 63. I like to be able to relate current actions to the longer-term bigger picture.
- ☐ 64. When things go wrong, I am happy to shrug it off and 'put it down to experience'.
- ☐ 65. I tend to reject wild, spontaneous ideas as being impractical.
- ☐ 66. It's best to think carefully before taking action.
- ☐ 67. On balance, I do the listening rather than the talking.
- ☐ 68. I tend to be tough on people who find it difficult to adopt a logical approach.
- ☐ 69. Most times I believe the end justifies the means.
- ☐ 70. I don't mind hurting people's feelings so long as the job gets done.
- ☐ 71. I find the formality of having specific objectives and plans stifling.
- ☐ 72. I'm usually one of the people who puts life into a party.
- ☐ 73. I do whatever is practical to get the job done.
- ☐ 74. I quickly get bored with methodical, detailed work.
- ☐ 75. I am keen on exploring the basic assumptions, principles and theories underpinning things and events.
- ☐ 76. I'm always interested to find out what people think.

- ☐ 77. I like meetings to be run on methodical lines, sticking to laid down agenda.
- ☐ 78. I steer clear of subjective (biased) or ambiguous (unclear) topics.
- ☐ 79. I enjoy the drama and excitement of a crisis situation.
- ☐ 80. People often find me insensitive to their feelings.

Scoring

You score one point for each item you **ticked**. There are no points for items you crossed. Go back over your responses and simply circle the question number in the table below for each question you **ticked**. Then add up the number of circled responses in the Totals row.

	QUESTION NUMBER			
	2	7	1	5
	4	13	3	9
	6	15	8	11
	10	16	12	19
	17	25	14	21
	23	28	18	27
	24	29	20	35
	32	31	22	37
	34	33	26	44
	38	36	30	49
	40	39	42	50
	43	41	47	53
	45	46	51	54
	48	52	57	56
	58	55	61	59
	64	60	63	65
	71	62	68	69
	72	66	75	70
	74	67	77	73
	79	76	78	80
Totals:				
	Activist	Reflector	Theorist	Pragmatist

Your preferred learning styles

Now circle your **total** scores for each learning style on the table below to determine the strength of your preference.

ACTIVIST	REFLECTOR	THEORIST	PRAGMATIST	
20	20	20	20	Very strong preference
19	19	19	19	
18	18	18	18	
17		17	17	
16		16		
15				
14				
13				
12	17	15	16	Strong preference
11	16	14	15	
	15			
10	14	13	14	Moderate preference
9	13	12	13	
8	12	11	12	
7				
6	11	10	11	Low preference
5	10	9	10	
4	9	8	9	
3	8	7	8	Very low preference
2	7	6	7	
1	6	5	6	
0	5	4	5	
	4	3	4	
	3	2	3	
	2	1	2	
	1	0	1	
	0		0	

ANEXO C – Heurísticas de Acessibilidade TEA

No.	Heuristics
1	[Visibility] of system status The system should always keep users informed about the activities, for instance, what is going to happen, how long it will take, what is going on, through appropriate feedback within reasonable time.
2	[Match] between system and the real world The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3	[Consistency] and standards The system should use clear and consistent language so that users do not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions in the design for consistency.
4	[Recognition] rather than recall Minimise the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the screen to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
5	Aesthetic and [minimalist] design—minimise distraction and keep design simple The design of user interface screens should not contain information which is irrelevant or rarely needed as it may distract these children's attention. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
6	User [control] and freedom (improved) Users often choose system functions by mistake and will need a clearly marked 'emergency exit' to leave the unwanted state without having to go through an extended dialogue. Support undo and redo. <i>The system should allow users to move from one part to another and provide the facility to repetitively perform activities.</i>
7	[Error] prevention (improved) Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action. <i>If the users select wrong options, the system should provide alternative options for them to choose from.</i>
8	[Flexibility] and efficiency of use (improved) Accelerators—unseen by the novice user—may often speed up interactions for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. <i>The system should carry out initial evaluation of performance to identify the level of the user and suggest an appropriate activity to start with.</i>
9	Help users recognise, diagnose, and [recover] from errors (improved) Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest how to avoid this error. <i>The system should provide multimedia demonstration to provide suggestions to the user when an error occurs.</i>
10	Help and [documentation] (improved) Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easily available to the user, focus on the user's task, <i>provide multimedia demonstration of tasks to be carried out, and not be too large.</i>
11	[Personalisation] of screen items The system should allow personalisation of screen items based on needs, abilities and preferences of an individual child. Screen items should be large enough for children to read and interact with. It should also allow them to change various settings of system background, font, colour, screen size and others.
12	User interface [screens] of the system The change on the screens of user interface of the system should take place step-by-step as children with ASD will not be able to cope with sudden or drastic changes made.
13	[Responsiveness] of the system Each action performed by children with autism (for instance click and select) should have no latency as children with autism have shorter attention span, typically forget quickly and can easily get frustrated.

No. Heuristics	
14	[Track] user activities monitor performance and repeat activity The system should keep a history of all the activities performed by the user, time spent, responses provided, results and others. They should be allowed to view their performance over a period of time and can return to any of the past activities to repeat it.
15	Use of [multi-modalities] for communication Users should be given the option to use different devices to provide input to the system. The communication between users and the system should take place using multimedia (text, digitised audio, images, animation, video and others).

ANEXO D – Modelo de Maturidade AUCDI

Item	Practices	Rating
Funds	Presence/utilization of dedicated funding for early user focus, field studies, usability and / or user experience activities, etc.	
Staff	Presence of qualified UCD team led by UCD manager Presence of qualified UCD practitioner per project/in this project	
Tools	Availability and utilization of prototyping tools Availability and utilization of usability labs	
Methods	Utilization of appropriate UCD methods	
Management UCD Awareness	Awareness of the importance of the UCD practitioner role Management understand that user centered design must be part of the business strategy Management support the inclusion and prioritization of UCD activities in the agile development process	
Training	UCD Awareness Training to Developers UCD Awareness Training to Agile coaches /scrum master and product owner UCD Awareness Training to Customers UCD Awareness Training to Business Managers UCD Practitioner Role Awareness Training to Developers UCD Practitioner Role Awareness Training to Agile coaches /scrum master and product owner UCD Practitioner Role Awareness Training to Customers UCD Practitioner Role Awareness Training to Business Managers Agile Awareness Training to UCD Practitioners Agile Awareness Training to Customers Agile Awareness Training to Business Managers Agile Developer Role Awareness Training to UCD Practitioners	
Standards, Patterns, Style Guides	Utilization of UCD standards Utilization of user interface patterns Utilization of UCD style guides	
Workspace	Colocation of developers and UCD practitioners	

Process	Practice	Rating
Plan the UCD Process	Identify and plan customer involvement Identify and plan user involvement Select user centered design methods Identify the relevant UCD specialist skills required and plan to provide them	
User Groups	Identification and understanding of user groups	
Context Of Use	Identification and understanding of user characteristics Identification and understanding of user tasks Identification and understanding of user goals Identification and understanding of the technical environment of old product Identification and understanding of the organizational environment of old product Identification and understanding of the physical environment of old product Identification and understanding of the technical environment of new product Identification and understanding of the organizational environment of new product Identification and understanding of the physical environment of new product	
Task Analysis	Identification and understanding of the non-functional tasks' attributes Identification and understanding of problems that users meet when performing tasks Identification and understanding of risks that users meet when performing a task	
User Requirements	Identification and understanding of usability and/or user experience requirements	
UI Design Requirements	Identification and understanding of UI design requirements	
Lightweight Documentation	Maintaining lightweight documentation for results of UCD process planning, user analysis and task analysis, user and UI design requirements	

Process	Practice	Rating
Synchronization Efforts	Developers share product vision with UCD team	
	UCD practitioner share user information with developers	
	UCD practitioner share task analysis information with developers	
	UCD practitioner share UCD vision with developers	
—	Design Chunking	
—	Inclusion of UCD activities (usability/user experience features) in product backlog or user stories	
—	Prioritization of UCD activities/features in cycles (sprints/iterations)	
—	Developers implementing back-end functionality	
—	UCD practitioners performing interaction and user task design for cycle N+1	
—	UCD practitioners maintaining lightweight documentation of design and design rationale for cycle N+1	
—	UCD practitioners gathering user and UI design requirements for cycle N+2	

Process	Practice	Rating
—	Developers implementing functionality and coding for interaction and user task designs	
—	Synchronization efforts - Developers understand design - UCD practitioner continually clearing up any design questions posed by developers	
—	UCD practitioners performing usability evaluation and analysis of usability evaluation results for cycle n code	
—	Synchronization efforts- UCD practitioners sharing usability evaluation results for cycle n code with the development team	
—	UCD practitioners performing interaction and user task design for cycle N+2	
—	UCD practitioners maintaining lightweight documentation of design and design rationale for cycle N+2	
—	UCD practitioners gathering user and UI design requirements for cycle N+3	

Process	Practice	Rating
—	Developers implementing functionality and coding for interaction and user task designs	
—	Synchronization efforts - Developers understand design - UCD practitioner continually clearing up any design questions posed by developers	
—	UCD practitioners performing usability evaluation and analysis of usability evaluation results for cycle N+1 code	
—	Synchronization efforts - UCD practitioners sharing usability evaluation results for cycle N code with the development team	
—	UCD practitioners performing interaction and user task design for cycle N+3	
—	UCD practitioners iteratively redoing interaction and user task design for cycle n code according to usability evaluation results acquired in cycle N+1	
—	UCD practitioners maintaining lightweight documentation of design and design rationale for cycle N+3	
—	UCD practitioners gathering user and UI design requirements for cycle N+4	

Person	Practices	Rating
Customers	Identification and selection of customers Awareness of Agile Awareness of User Centered Design Early Involvement Continual Involvement	
Users	Identification and selection of users Early Involvement Continual Involvement	
UCD Practitioner	Competence Awareness- Understanding of Agile values, principles and practices Awareness- Understanding of Agile developer role Attitude- Continuous communication Attitude- Respect and trust for developers Attitude- Maintaining visibility of UCD team work throughout the agile development process	
Developers	Communication skills Awareness- Understanding of UCD Awareness- Understanding of UCD practitioner role Attitude- Respect and trust for UCD practitioner Attitude- Trust customer	
Coach/Scrum Master and Product Owner	Communication skills Awareness- Understanding of UCD Awareness- Understanding of UCD practitioner role Attitude- Respect and trust for UCD practitioner Attitude- Trust customer	

Practice	Rating
Presence of a UCD monitoring process across projects	
Presence of a UCD systematic improvement process for UCD activities, tools, methods, workspace, skills and awareness	
Benchmarking product's usability and/or user experience against competitive products' usability and/or user experience	
Product decisions emerge from end user and customer studies and are targeted to meet users' needs and expectations	

TRABALHOS PUBLICADOS PELO AUTOR

1. CAMARGO, M. C.; BARROS, R. M. BARROS, V. T. O. **Visual design checklist for Graphical User Interface (GUI) evaluation.** In: 33rd Annual ACM Symposium, 2018, Pau. Anais da 33rd Annual ACM Symposium on Applied Computing – SAC’ 18, 2018, p. 670-672 (Qualis CC 2016, A1).
2. CAMARGO, M. C.; CARVALHO, T. C. P.; BARROS, R. M.; BARROS, V. T. O.; SANTANA, M. **Improving usability of a mobile application for children with Autism Spectrum Disorder using heuristic evaluation.** In: 21st International Conference on Human-Computer Interaction, Orlando, Estados Unidos (Qualis CC 2016, B2).