



PGDESIGN | Programa de Pós-Graduação
Mestrado | Doutorado



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
FACULDADE DE ARQUITETURA
PROGRAMA DE PÓS-GRADUAÇÃO EM DESIGN

Gissele Azevedo Cardozo

**HYPERCAL^{3D} MOBILE: APLICATIVO PARA AUXÍLIO NO ENSINO DE
GEOMETRIA DESCRITIVA**

Tese de Doutorado

Porto Alegre
2022

GISSELE AZEVEDO CARDOZO

HYPERCAL^{3D} MOBILE: APLICATIVO PARA AUXÍLIO NO ENSINO DE GEOMETRIA DESCRITIVA

Tese apresentada ao Programa de Pós-Graduação em Design da Universidade Federal do Rio Grande do Sul, como requisito parcial à obtenção do título de Doutor em Design.

Orientador: Prof. Dr. Régio Pierre da Silva

Orientador: Prof. Dr. Fábio G. Teixeira

Coorientador: Prof. Dr. Sérgio Leandro dos Santos

Porto Alegre
2022

CIP - Catalogação na Publicação

Cardozo, Gissele Azevedo
HYPERCAL3D MOBILE: APLICATIVO PARA AUXÍLIO NO
ENSINO DE GEOMETRIA DESCRITIVA / Gissele Azevedo
Cardozo. -- 2022.
212 f.
Orientadores: Régio Pierre da Silva, Fábio G.
Teixeira.

Coorientador: Sérgio Leandro dos Santos.

Tese (Doutorado) -- Universidade Federal do Rio
Grande do Sul, Escola de Engenharia, Programa de
Pós-Graduação em Design, Porto Alegre, BR-RS, 2022.

1. HyperCAL 3D. 2. Mobile learning. 3. Interfaces
gestuais. 4. Usabilidade. 5. Geometria Descritiva. I.
Silva, Régio Pierre da, orient. II. Teixeira, Fábio
G., orient. III. Santos, Sérgio Leandro dos,
coorient. IV. Título.

GISSELE AZEVEDO CARDOZO

HYPERCAL^{3D} MOBILE: APLICATIVO PARA AUXÍLIO NO ENSINO DE GEOMETRIA DESCRITIVA

Esta Tese foi julgada adequada para a obtenção do Título de Doutor em Design, e aprovada em sua forma final pelo Programa de Pós-Graduação em Design da UFRGS.

Porto Alegre, 19 de dezembro de 2022.

Prof. Dr. Fabio Pinto da Silva

Coordenador do Programa de Pós-Graduação em Design da UFRGS

Banca Examinadora:

Orientador: **Prof. Dr. Régio Pierre da Silva**

Programa de Pós-Graduação em Design da UFRGS (PGDesign-UFRGS)

Orientador: **Prof. Dr. Fábio G. Teixeira**

Programa de Pós-Graduação em Design da UFRGS (PGDesign-UFRGS)

Coorientador: **Prof. Dr. Sérgio Leandro dos Santos**

Departamento de Design e Expressão Gráfica

Prof. Dr. Pablo Ermida Corrêa

Catho online Ltda.– Examinador Externo

Prof. Dr. Fernando Batista Bruno

Departamento de Design e Expressão Gráfica – Examinador Externo

Prof. Dr. Vinícius Gadis Ribeiro

Programa de Pós-Graduação em Design da UFRGS (PGDesign-UFRGS)-
Examinador Interno

AGRADECIMENTOS

Aos professores Régio Pierre da Silva, Fábio G. Teixeira, Sérgio Leandro dos Santos, pela orientação, suporte e dedicação. Este trabalho é mais um fruto no longo e constante processo de melhorias no ensino que vocês desenvolvem há vários anos na Universidade Federal do Rio Grande do Sul. Agradeço a oportunidade de participar deste projeto que coloca o ensino de geometria descritiva da UFRGS no mais alto nível.

Aos membros da banca examinadora, professores Vinícius, Fernando e Pablo, suas contribuições e sugestões enriqueceram e qualificaram esta pesquisa.

A Universidade Federal do Rio Grande do Sul, com especial agradecimento ao programa de Pós-Graduação em Design e seus integrantes, pela oportunidade de aprendizagem e pelos conhecimentos adquiridos.

A Universidade Federal de Pelotas e aos professores dos cursos de Cinema e Design por permitir e apoiar esta jornada.

Aos colegas e amigos que contribuíram direta ou indiretamente para a conclusão deste trabalho.

Aos meus irmãos Francine e Vinícius que sempre estiveram comigo e me deram força para continuar quando eu não acreditava que seria capaz de seguir em frente.

Ao meu marido Daniel que me inspira e incentiva a buscar sempre o crescimento, ser resiliente e dar o melhor de si. Agradeço o apoio e compreensão.

A minha filha Natália que me ensinou a ver o mundo de outra forma, é a minha inspiração para continuar em frente e razão pela qual procuro o desenvolvimento. Te ver crescer e se tornar a pessoa maravilhosa que és, sempre será minha melhor conquista.

RESUMO

CARDOZO, G. A. **HyperCAL^{3D} mobile: Aplicativo para o auxílio no ensino de geometria descritiva** 2022. 204f Tese (Doutorado em Design) – Escola de Engenharia / Faculdade de Arquitetura, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2022

A tecnologia vem moldando o ensino há algum tempo, na Universidade Federal do Rio Grande do Sul o grupo VID, para aprimorar o ensino-aprendizagem de geometria descritiva, desenvolveu o *software* HyperCAL^{3D} que já teve a sua eficiência comprovada pelo tempo. No entanto, devido as novas tecnologias móveis como celulares e tablets, novos recursos podem ser utilizados para a ampliação do acesso do *software*. Assim, o presente trabalho tem por objetivo a adaptação dele ao *mobile learning*. A revisão sistemática de literatura revelou que os estudos sobre o desenvolvimento de *mobile learning*, tendem à utilização de modelos de aceitação da tecnologia que, em geral, não apresentam soluções práticas, como requisitos para a implementação. Por este motivo, foi realizado um cruzamento teórico entre as heurísticas de usabilidade móvel SMART com o modelo de aceitação de tecnologia para *m-learning* MLAM, na busca por requisitos para a análise de similares, prevista na metodologia *Design Science Research* (DSR) que orienta esta pesquisa. As análises de aplicativos destinados a modelagem tridimensional e desenho técnico, similares ao HyperCAL^{3D}, revelaram que a falta de periféricos que auxiliem a interação com objetos 3D e 2D, pode ser subvertida pela utilização de interfaces gestuais. O desenvolvimento do artefato foi realizado a partir de ciclos (*Sprints*) de implementação presentes na metodologia ágil SCRUM, apoiados pela metodologia XP. Foram realizados, ao todo, seis *sprints* para a adaptação do HyperCAL^{3D} ao *mobile learning*, quatro destes tratavam da adaptação da tecnologia, linguagem de programação e estratégias de implementação da versão *desktop* para a versão móvel. Um ciclo foi necessário para a correção do desempenho e no último *sprint* foram desenvolvidas as questões pertinentes aos modelos de aceitação da tecnologia, usabilidade e interfaces gestuais observadas neste estudo. Após a conclusão dos ciclos, o aplicativo HyperCAL^{3D} *mobile* foi submetido à avaliação de usabilidade MATCH - *Measuring Usability of Touch screen Phone Applications*, na qual atingiu a pontuação máxima. O conhecimento adquirido neste estudo pode ser utilizado como base para a desenvolvimento de novas pesquisas em *mobile learning*, assim como interfaces gestuais em aplicativos 3D e 2D.

Palavras-chave: HyperCAL^{3D}, *Mobile learning*, Interfaces gestuais, Usabilidade, Geometria Descritiva, Computação gráfica.

ABSTRACT

CARDOZO, G. A. **HyperCAL3D mobile: Application for assistance in teaching descriptive geometry** 2022. 204f Thesis (PhD in Design) - School of Engineering / Faculty of Architecture, Federal University of Rio Grande do Sul, Porto Alegre, 2022

Technology has been shaping teaching for some time, at the Federal University of Rio Grande do Sul, the VID group, to improve teaching and learning of descriptive geometry, developed the HyperCAL 3D software, which has already had its efficiency proven by time. However, due to new mobile technologies such as cell phones and tablets, new resources can be used to expand software access. The present work aims to adapt it to mobile learning. The systematic literature review revealed that studies on the development of mobile learning tend to use models of acceptance of technology that generally do not present practical solutions, such as requirements for implementation. For this reason, a theoretical crossing was made between smart mobile usability heuristics with the technology acceptance model for MLAM m-learning, in the search for requirements for the analysis of similar, provided for in the Design Science Research (DSR) methodology that guides this research. The analysis of applications for three-dimensional modeling and technical design, like HyperCAL ^{3D}, revealed that the lack of peripherals that help with the interaction of 3D and 2D objects can be subverted by the use of gestural interfaces. The development of the artifact was carried out from implementation cycles (Sprints) present in the agile SCRUM methodology, supported by the XP methodology. A whole of six sprints were performed to adapt HyperCAL ^{3D} to mobile learning, four of which dealt with the adaptation of technology, programming language and implementation strategies, between the desktop version for the mobile version. A cycle was necessary for the correction of performance and in the last sprint were developed the issues pertinent to the models of acceptance of technology, usability and gestural interfaces observed in this study. After the cycles were completed, the HyperCAL ^{3D} mobile application was submitted to the MATCH - Measuring Usability of Touch screen Phone Applications usability assessment, in which it reached the maximum score. The knowledge acquired in this study can be used as a basis for the development of new research in mobile learning, as well as gestural interfaces in 3D and 2D applications.

Keywords: HyperCAL ^{3D}, Mobile learning, Gesture interfaces, Usability, Descriptive geometry, Computer Graphics

Lista de Figuras

Figura 1 - Modelo TCK	18
Figura 2- Modelo TPACK	18
Figura 3 - Modelo TAM.....	33
Figura 4 - Modelo UTAUT	34
Figura 5 - Modelo de adoção de aprendizagem móvel (MLAM) em três estágios de serviço.....	37
Figura 6 - <i>Feedback</i> formulário	42
Figura 7 - Ícones do WhatsApp	42
Figura 8 - Botão cancelar envio do Gmail	42
Figura 9 - Botões de navegação no top a esquerda do Google Chrome	43
Figura 10 - Mensagem de erro na senha Gmail	43
Figura 11 - Datas passadas desabilitadas no site Decolar.....	43
Figura 12 - Pix realizados anteriormente salvos no aplicativo do banco do Brasil	44
Figura 13 - Lista de atalhos do Blender.....	44
Figura 14 - Tela de login do Gmail	45
Figura 15 - Interrogações no formulário para passaporte da Polícia Federal.....	45
Figura 16 - Modelo de gerenciamento empírico	64
Figura 17 – Ciclo do <i>Framework</i> Scrum	65
Figura 18 - HyperCAL ^{3D} versão 1.0.....	71
Figura 19 - HyperCAL ^{3D} versão 2.0.....	71
Figura 20 - Mudanças no HyperCAL ^{3D} versão 3.0	72
Figura 21 - HyperCAL versão 4.0.....	72
Figura 22 - Folha padronizada da épura confeccionada no HyperCAL ^{3D}	73

Figura 23 - HyperCAL versão 5.0.....	73
Figura 24 - HyperCAL 3D versão 9.0	74
Figura 25 - Etapas da metodologia Design Science Research	77
Figura 26 - Botões de layout das viewport 3d e épura do HyperCAL ^{3D} web	81
Figura 27 - <i>Framework</i> Scrum e Metodologia XP.....	82
Figura 28 - Desenho da pesquisa	85
Figura 29 - Fases da Revisão Sistemática de Literatura	86
Figura 30 - Comparativo entre pilares do <i>mobile learning</i> e HyperCAL ^{3D}	101
Figura 31 - Fase do modelo MLAM e construtos utilizados.....	105
Figura 32 - Cruzamento entre o modelo MLAM e as Heurísticas SMART	107
Figura 33 – Ligações entre Compatibilidade Percebida e SMATS 7 e 8.....	108
Figura 34 - Ligações entre Consciência Percebida e SMARTS 1 e 2	110
Figura 35 - Ligações entre Benefício Funcional Percebido e SMARTS 5, 6, 9 e 12.....	112
Figura 36 - Ligações entre Capacidade de Uso Percebida e SMARTS 3 e 11	114
Figura 37 - Mensagem do aplicativo Tinkercard.....	115
Figura 38 - Tetra – Easy 3D Creation.....	116
Figura 39 - EDS 3D Modeling tool.....	116
Figura 40 – Tutorial 3D Modeling app	118
Figura 41 - menus escondidos 3D Modeling App.....	120
Figura 42 - Teclado Virtual 3D Modeling App.....	121
Figura 43 - Navegação no OnShape	121
Figura 44 - Dispositivo para navegação OnShape	123
Figura 45 - Entrada de dados no OnShape.....	124
Figura 46 - Telas iniciais do app Wuweido	125
Figura 47 - Qubism 3D Modeling.....	127

Figura 48 - SDF 3D	129
Figura 49 - Edição de desenhos AutoCAD	132
Figura 50- Botões para a navegação no AutoCAD	132
Figura 51 - Atalho para comandos do AutoCAD	133
Figura 52 – UVCAD.....	134
Figura 53 – GnaCAD.....	136
Figura 54 - Edição no GnaCAD	138
Figura 55- Zoom e centralização do desenho no GnaCAD	138
Figura 56 - Menus GnaCAD	139
Figura 57 - Editor Three.js.....	140
Figura 58 - Ferramenta de navegação ortográfica Three.js	142
Figura 59 - Legenda de cores	143
Figura 60 - Ferramentas de criação de sólidos no HyperCAL 3D web.....	156
Figura 61 - Modo Claro/ Escuro HyperCAL 3D web	156
Figura 62 - Alteração da visualização das <i>views</i> no HyperCAL 3D web	157
Figura 63 - Mudança de sistema de projeção	158
Figura 64 - Estrutura principal do HyperCAL 3D web.....	159
Figura 65 - Implementação da classe Tface	163
Figura 66 - Primeiro teste da classe Tface	164
Figura 67 - Nomes dos pontos	165
Figura 68 - Sprint classe Taresta	167
Figura 69 - Sprint da classe Tplano.....	168
Figura 70 - <i>Sprint</i> Épura	171
Figura 71 - <i>Spring</i> correção do desempenho	172
Figura 72 - <i>Sprint</i> Adaptação.....	174

Figura 73 - Fluxograma de identificação de toques HyperCAL ^{3D} <i>mobile</i>	176
Figura 74 - HyperCAL ^{3D} <i>mobile</i> no navegador Chrome vertical	185
Figura 75 - HyperCAL ^{3D} <i>mobile</i> no navegador Chrome horizontal	185

Lista de quadros

Quadro 1 - Definição conceitual dos construtos e hipóteses do MLAM	39
Quadro 2 - Resultados obtidos na busca pelas <i>Strings</i>	91
Quadro 3 Avaliação de trabalhos por critérios de qualidade	92
Quadro 4 – Pontuação final dos Critérios de qualidade	93
Quadro 5 – Divisão dos artigos entre análises e desenvolvimento	95
Quadro 6 – Modelos utilizados em ordem temporal	98
Quadro 7 - Construtos da fase de transação destacados devido ao grau de adoção da aceitação	104
Quadro 8 - Navegação, seleção, desseleção e edição 3D Modeling App.....	119
Quadro 9 - Navegação, seleção, desseleção e edição OnShape	122
Quadro 10 -Navegação, seleção, desseleção e edição Wuweido.....	125
Quadro 11 - Navegação, seleção, desseleção e edição Qubism 3D Modeling.....	127
Quadro 12 - Navegação, seleção, desseleção e edição SDF 3D.....	129
Quadro 13 - Navegação, seleção, desseleção e edição AutoCAD	131
Quadro 14 - Navegação, seleção, desseleção e edição UVCAD.....	135
Quadro 15 - Navegação, Seleção e Edição GnaCAD	137
Quadro 16 - Navegação, seleção, desseleção e edição do Editor Three.js	141
Quadro 17 - Gestos de navegação dos aplicativos 3D analisados	144
Quadro 18 - Gestos de navegação dos aplicativos 2D analisados	145
Quadro 19 -Gestos de seleção.....	146
Quadro 20 – Gestos para desselecionar.....	147
Quadro 21 - Gestos para editar	148
Quadro 22 – Mecanismos de <i>feedback</i>	149

Quadro 23 - Metodologia XP no Sprint Tface.....	166
Quadro 24 – Metodologia XP no Sprint Taresta.....	167
Quadro 25 - Metodologia XP <i>Sprint</i> Tplano.....	170
Quadro 26 - Metodologia XP na implementação da Épura	172
Quadro 27 - Metodologia XP na correção do desempenho	173
Quadro 28 -Gestos da navegação HyperCAL ^{3D} <i>mobile</i>	177
Quadro 29 - Seleção HyperCAL ^{3D} <i>mobile</i>	178
Quadro 30 -Metodologia XP na implementação da Seleção.....	179
Quadro 31 - Deseleção HyperCAL ^{3D} <i>mobile</i>	180
Quadro 32 -Metodologia XP na implementação da Deseleção	181
Quadro 33 - Atalhos HyperCAL ^{3D} <i>mobile</i>	183
Quadro 34 Pontuação obtida na escala MATch	195

SUMÁRIO

1	INTRODUÇÃO	17
1.1.	CONTEXTUALIZAÇÃO DO TEMA	17
1.2.	DELIMITAÇÃO DO TEMA	24
1.3.	PROBLEMA DE PESQUISA	25
1.4.	HIPÓTESE	25
1.5.	OBJETIVOS	25
1.5.1.	Objetivo Geral	25
1.5.2.	Objetivos Específicos	25
1.6.	JUSTIFICATIVA DA PESQUISA	26
2.	FUNDAMENTAÇÃO TEÓRICA	29
2.1.	MOBILE LEARNING	29
2.1.1.	Modelo TAM	33
2.1.2.	Modelo UTAUT	33
2.1.3.	Modelo MLAM	36
2.2.	USABILIDADE	40
2.2.1.	Avaliação Heurística	41
2.2.1.	Usabilidade Móvel	46
2.2.2.	ESCALA MATCH	50
2.3.	INTERFACES NATURAIS	55
2.4.	PROCESSO DE DESENVOLVIMENTO DE SOFTWARES	60
2.4.1.	Framework SCRUM	62
2.4.2.	Metodologia Extreme Programming (XP)	65
2.4.3.	Testes	68
2.5.	HYPERCAL ^{3D}	70
3.	METODOLOGIA DA PESQUISA	75
3.1.	IDENTIFICAÇÃO DO PROBLEMA	77

3.2.	CONSCIENTIZAÇÃO DO PROBLEMA	78
3.3.	REVISÃO SISTEMÁTICA DA LITERATURA	79
3.4.	IDENTIFICAÇÃO DOS ARTEFATOS E CONFIGURAÇÃO DAS CLASSES DE PROBLEMAS	79
3.5.	PROJETO E DESENVOLVIMENTO DO ARTEFATO.....	81
3.6.	AVALIAÇÃO DO ARTEFATO	83
3.7.	EXPLICITAÇÃO DAS APRENDIZAGENS	83
3.8.	CONCLUSÃO	83
3.9.	GENERALIZAÇÃO PARA UMA CLASSE DE PROBLEMAS.....	83
3.10.	COMUNICAÇÃO DOS RESULTADOS	83
4.	DESENVOLVIMENTO DA PESQUISA	86
4.1.	REVISÃO SISTEMÁTICA DE LITERATURA.....	86
4.2.	CRUZAMENTO TEÓRICO	101
4.2.1.	Mobile learning e HyperCAL ^{3D}	101
4.2.2.	Modelo MLAM e Usabilidade SMART	106
4.3.	IDENTIFICAÇÃO DOS ARTEFATOS E CONFIGURAÇÃO DAS CLASSES DE PROBLEMAS	115
4.3.1.	Análise do aplicativo 3D Modeling App	118
4.3.2.	Análise do aplicativo OnShape 3D CAD	121
4.3.3.	Análise do aplicativo Wuweido.....	125
4.3.4.	Análise do aplicativo Qubism 3D Modeling.....	127
4.3.5.	Análise do aplicativo SDF 3D.....	128
4.3.6.	Análise do aplicativo AutoCAD	131
4.3.7.	Análise do aplicativo UVCAD.....	134
4.3.8.	Análise do aplicativo GnaCAD.....	136
4.3.9.	Análise do site Three.js Editor.....	140
4.3.10.	Conclusões das análises	143
4.3.11.	Configuração das classes de problemas	151
4.4.	PROPOSIÇÃO DE ARTEFATOS PARA RESOLUÇÃO DO PROBLEMA	152

4.5.	PROJETO E DESENVOLVIMENTO DO ARTEFATO.....	153
4.5.1.	Funcionalidades da versão web.....	155
4.5.2.	Arquitetura da versão web HyperCAL 3D.....	158
4.5.3.	Viewport 3D e Épura.....	161
4.5.4.	Adaptação do HyperCAL ^{3D} ao <i>mobile</i>	162
4.6.	AVALIAÇÃO DO ARTEFATO.....	186
5.	EXPLICITAÇÃO DAS APRENDIZAGENS.....	196
6.	CONSIDERAÇÕES FINAIS.....	198
6.1.	CONCLUSÕES.....	198
6.2.	GENERALIZAÇÃO DA CLASSE DE PROBLEMAS.....	200
6.3.	SUGESTÃO PARA TRABALHOS FUTUROS.....	201
	REFERÊNCIAS.....	203
	APÊNDICE A.....	211

1 INTRODUÇÃO

A utilização de tecnologias não é mais uma novidade. Comunicação, lazer, trabalho, gerenciamento financeiro e pessoal podem ser acessados a qualquer momento em dispositivos menores que uma agenda, que vibram e anunciam o evento mais próximo. Atualmente, indivíduos sem dispositivos eletrônicos são exceção e não mais a regra. Um estudo da Fundação Getúlio Vargas (MEIRELLES, 2022) apontou que até maio de 2022, existiam cerca de 1,6 dispositivos, entre *smartphones*, *notebooks* e tablets, por habitante no Brasil.

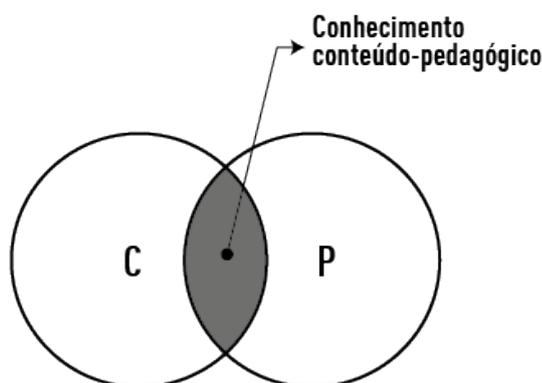
1.1. CONTEXTUALIZAÇÃO DO TEMA

É inegável que a tecnologia possui um potencial inestimável nas diferentes áreas da vida cotidiana. A facilidade de acesso, o envio de informações, o consumo de bens e serviços, todos facilitados pela tecnologia e em especial pela internet. Na educação não é diferente: e-mails, vídeos, formulários on-line, entre outros, foram formas encontradas por educadores para facilitar a transmissão dos conteúdos ministrados e/ou realizar pesquisas.

O dueto tecnologia e educação não é uma novidade, um modelo chamado *Technological Pedagogical And Content Knowledge* – TPACK (Conhecimento Pedagógico, Tecnológico e de Conteúdo), proposto por Mishra e Koehler (2006), sugere a integração eficaz entre tecnologias digitais e o processo de ensino-aprendizagem. A base para a criação do modelo surgiu dos estudos de Shulman (1986) que propôs que a interseção entre conteúdo (C) e pedagogia (P) surge o conhecimento conteúdo-pedagógico (*Pedagogical Content Knowledge* - PCK). Isso significa que o conteúdo, aliado as melhores estratégias de ensino, têm a capacidade de melhorar a compreensão por parte dos alunos.

Ao longo dos anos, o PCK ganhou notoriedade por compreender que os tópicos ministrados podem ser representados pelas melhores analogias, ilustrações, exemplos e demonstrações. Isso significa que a maneira de apresentação (pedagogia) do conteúdo que o torna acessível aos alunos. De modo simplificado, a união entre conteúdo e pedagogia gera conhecimento. A Figura 1, idealizada por Mishra e Koehler (2006), procura sintetizar o modelo PCK.

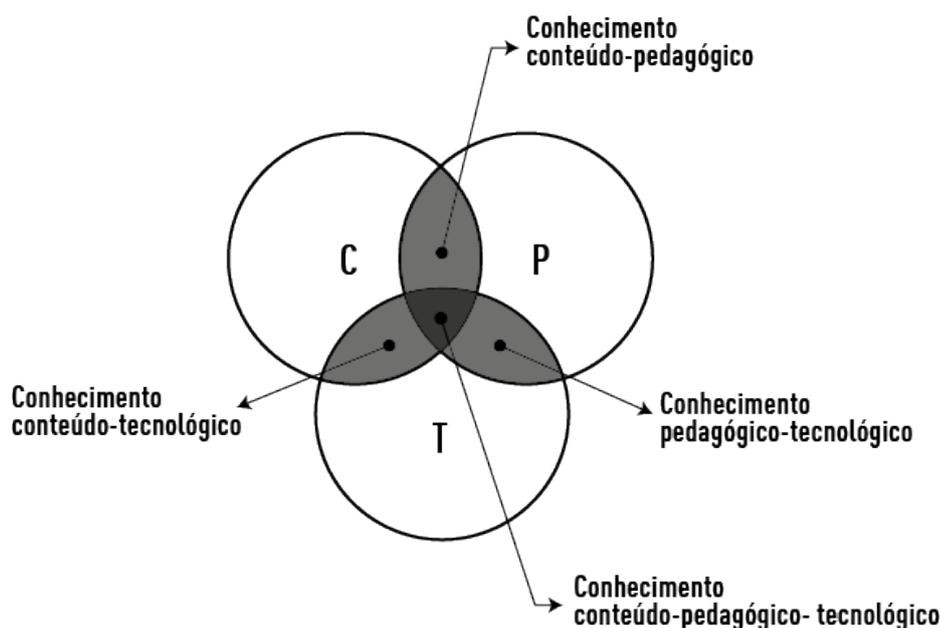
Figura 1 - Modelo TCK



Fonte: Traduzido de Mishra e Koehler (2006)

Como já mencionado, o modelo TPACK utiliza a mesma base do PCK, conteúdo e pedagogia, porém insere a tecnologia na educação. Para tanto, não bastam conhecimentos e habilidades tecnológicas, é importante que os professores façam o melhor uso destas no ensino. Em outras palavras, a tecnologia precisa ser ressignificada para que atenda os fins pedagógicos do conteúdo ministrado.

Figura 2- Modelo TPACK



Fonte: Traduzido de Mishra e Koehler (2006)

Portanto, TPACK foi proposto a partir de três componentes centrais: o conhecimento tecnológico (T), o conhecimento do conteúdo da área de atuação (C), e o conhecimento pedagógico (P). Mishra e Koehler (2006) partem da interseção destes elementos, assim como no PCK, conteúdo e pedagogia geram conhecimento conteúdo-pedagógico, no modelo TPACK mais três fontes de conhecimento são geradas: conhecimento conteúdo-tecnológico, conhecimento pedagógico-tecnológico

e conhecimento conteúdo-pedagógico-tecnológico (Figura 2). Assim, os professores podem pensar o conteúdo ministrado apoiado pelas melhores práticas de ensino e com o auxílio das ferramentas mais relevantes.

A partir do alinhamento entre tecnologia e ensino, outras modalidades de aprendizagem surgiram. As tecnologias digitais, aliadas à internet, permitiram a criação da aprendizagem eletrônica ou *e-learning*. Nesta modalidade, são comumente utilizados os ambientes virtuais de aprendizagem (AVAs), que podem ser acessados de qualquer navegador, sem a necessidade de instalação de programas ou pacotes de dados. Com o advento do *e-learning* estilos de ensino como: Síncrono, Assíncrono e Híbrido passaram a fazer parte do conjunto de estratégias disponíveis para a aprendizagem.

Como os nomes sugerem, ensino Síncrono é realizado em tempo real, geralmente mediante agendamento, no qual instrutores e alunos se encontram em salas virtuais de aprendizagem, ou vídeo conferências (SHAHABADIA e UPLANE, 2015). No ensino Assíncrono, o conteúdo é disponibilizado em meio eletrônico como os AVAs de ensino e os alunos podem acessá-lo em qualquer local, em horários flexíveis. O ensino Híbrido, considera a aprendizagem em diferentes meios: presencial sem o auxílio de tecnologia, por meio eletrônico em sala de aula, ou fora dela, em tempo real ou com horários flexíveis.

Com exceção do ensino Síncrono, que simula o ambiente presencial em um meio virtual, estas novas formas de ensino, passaram a utilizar uma abordagem centrada no aluno e não mais com o professor como figura centralizadora (SHAHABADIA e UPLANE, 2015).

Além destas possibilidades, outro fator determinante na utilização da tecnologia no ensino é o dispositivo. Relativamente novos quando comparados a computadores pessoais, os *smartphones* e *tablets*, com suas *touch screens* ganharam uma quantidade considerável de adeptos ao longo dos anos. A mesma pesquisa da Fundação Getúlio Vargas (MEIRELLES, 2022) evidencia a existência de 1,6 dispositivos eletrônicos por habitante do Brasil e afirma que até maio de 2022, havia cerca de 242 milhões de *smartphones* no país, que abrigava, no mesmo período 212

milhões de pessoas. A utilização de dispositivos móveis¹ no ensino abriu uma nova forma de interação com conteúdo e pedagogias, pois apesar de apresentarem estruturas semelhantes ao *e-learning*, o tamanho das telas, a conectividade, o processamento e as diferentes tecnologias envolvidas, apresentam desafios peculiares a esta nova forma de ensino-aprendizagem.

O ensino a partir de dispositivos móveis vem sendo debatido há alguns anos, porém, a definição de *mobile learning (m-learning)* ainda está emergindo a partir da união do conceito de mobilidade com a aprendizagem eletrônica. Para Churchill, Fox e King (2016), o uso destes recursos permite aos alunos troca de informações, compartilhamento de arquivos, tutoria inteligente, disseminação de informações, coleta de dados durante viagens a campo, entre outros, é uma forma útil para os professores engajarem a tecnologia em suas práticas.

“A tecnologia móvel tem potencial para apoiar a reflexão dos alunos, levando a uma melhor realização de aprendizagem quando há uma combinação apropriada entre o estilo de ensino de um professor e o estilo de aprendizagem dos alunos”² (CHURCHILL, FOX e KING, 2016).

Contudo, pensar em *m-learning* está longe de ser uma simples utilização de recursos preexistentes. A adaptação dos conteúdos e materiais didáticos às tecnologias digitais, vai além da simples utilização de leitores de pdf, vídeos, troca de mensagens e aulas virtuais. Não é errado pensar que estas ferramentas auxiliam nas aulas, porém, “as tecnologias raramente são desenvolvidas dentro do campo da educação ou destinadas especificamente a ele” (LAURILLARD, 2012, p. 2).

Para o desenvolvimento da aprendizagem móvel, são necessários pelo menos quatro construtos fundamentais: pedagogia, dispositivos tecnológicos, contexto e interações sociais (CROMPOTON, 2013a). É necessário o conhecimento pedagógico do que será ensinado, o conhecimento tecnológico para viabilizar a criação dos aplicativos³, compreender se o conteúdo será utilizado em um contexto formal

¹ Este estudo considera como dispositivos móveis apenas *smartphones* e *tablets*, devido as características destes dispositivos como tamanho de tela, processamento e interfaces gestuais e a não necessidade de utilização de periféricos como mouses e teclados.

² Tradução livre do texto: “*mobile technology has potential to support students’ reflection leading to improved learning achievement when there is an appropriate match between a teacher’s teaching style and students’ learning style.*” (CHURCHILL, FOX e KING, 2016)

³ Para fins de estudo, são considerados aplicativos os recursos que podem ser acessados via

(acadêmico) ou informal (autogerenciado) e o acesso dos alunos com seus professores ou tutores para esclarecimentos.

Cada construto apresenta diferentes desafios e, portanto, se tornam claros os desafios enfrentados pelo desenvolvimento do *m-learning*. Por outro lado, assim como citado anteriormente, alguns autores como Traxler (2007), buscam uma aproximação do campo de estudo do *mobile learning* com o *electronic learning*. Mesmo que não soe absurda esta aproximação, questões como mobilidade e usabilidade de dispositivos moveis e computadores podem sugerir o contrário.

A usabilidade é um tema amplamente debatido, já há algum tempo. Desde sua criação até os dias atuais, diferentes heurísticas, como as de Nilsen e Molich (1990) e regras de ouro criadas por Schneiderman (2009), surgiram como forma de tentar avaliar e compreender o que leva um usuário a preferir determinado produto ou software em detrimento de outro.

Um dos autores mais influentes da área, Jakob Nielsen vem estudando, desde 1990, a forma como os usuários interagem com os dispositivos, sites, sistemas entre outros. É dele juntamente com Molich (1990) a autoria das heurísticas para design e análise de interfaces mais utilizadas até os dias de hoje (NIELSEN e MOLICH, 1990).

Porém, os dispositivos moveis apresentam desafios únicos provenientes de suas características peculiares como, tamanho, performance e mobilidade. Esta mudança de cenário, apresenta desafios particulares. Mesmo que as tecnologias tenham evoluído, fatores como a mobilidade representam adversidades nas escolhas do designer que deve compreender em que contexto seus aplicativos são utilizados para assim poder fazer melhores escolhas.

Apesar da tecnologia não ser uma novidade no ensino, os estudos que visavam sua aproximação ganharam foco inesperado no ano de 2020. Quando um novo vírus, chamado Corona Vírus, surgido no final de 2019 que tomou proporções globais no ano de 2020, ameaçou o sistema imunológico humano e desafiou os sistemas de saúde mundiais. Nos primeiros meses do ano de 2020, devido à falta de conhecimento

smartphones ou *tablets*, sendo eles aplicativos web móvel (acessíveis via navegador), nativos (programados para determinada plataforma) e híbridos (com navegador incorporado). Estes são detalhados no item 2.5.2 deste relatório.

do “*modus operandi*” do vírus, a Organização Mundial da Saúde (OMS) aconselhou os governantes dos países o isolamento social como forma de impedir a proliferação do patógeno. Repentinamente o cenário se modificou, novas ou reformuladas formas de interação precisaram ser utilizadas para tentar subverter a distância e emergencialmente atender as demandas preexistentes. Vídeos substituíram reuniões e aulas online na rede privada de ensino quase que instantaneamente ao encerramento das aulas presenciais, fato que não ocorreu na rede pública de ensino.

As aulas da rede pública precisaram ser adaptadas à realidade da população. A falta de internet ou meios de acesso deixaram expostas as diferenças socioeconômicas dos estudantes ao redor do globo. Governos de diferentes países precisaram adaptar o acesso ao material didático de acordo com a necessidade da população.

Apesar de a pandemia de corona vírus, a cada dia se distanciar no passado, alguns fatos importantes foram observados durante o período de isolamento, que podem ser utilizados, independente da distância social. Este é o caso dos dispositivos utilizados no período para o acesso à internet.

De maneira a compreender os recursos disponíveis aos alunos, a Faculdade de Arquitetura da UFRGS realizou uma consulta, na forma de um questionário online, entre os alunos dos cursos de Arquitetura e Urbanismo, Design Visual e Design de Produto (FACULDADE DE ARQUITETURA, 2020). Foram obtidas setecentos e oitenta e seis respostas de um total de novecentos e sessenta e um alunos, o que representa 81,8% dos acadêmicos dos três cursos. Na questão referente ao acesso à internet, 87,5% dos estudantes responderam que possuem acesso à internet através dispositivo móvel e computador, 4,8% apenas em dispositivos móveis, 4,5% através de um computador e 2,9% têm dificuldade de se conectar à internet. No entanto a pesquisa não deixa claro se a dificuldade vem da indisponibilidade de internet ou de um aparelho para a conexão (FACULDADE DE ARQUITETURA, 2020).

A partir desta pesquisa é possível notar que 92,6% dos entrevistados possuem acesso a algum dispositivo móvel com acesso à internet. Este fato, demonstra uma quantidade significativa de estudantes que poderiam utilizar seus dispositivos móveis como ferramentas de aprendizagem. Se por um lado, abundam recursos para

aprendizagem, por outro, a adaptação dos conteúdos aos dispositivos existentes não é uma tarefa simples, este é o caso da Geometria Descritiva (GD).

A GD é um ramo da geometria que busca traduzir objetos tridimensionais no plano bidimensional, foi sistematizada pelo Matemático Francês Gaspard Monge (1746-1818) e representa a base conceitual do desenho técnico moderno. Em teorias mais tradicionais de ensino-aprendizagem, o Método Mongeano, como também é conhecida a GD, o ensino usualmente parte de uma abordagem do abstrato para o concreto (PEREIRA, 2021). São trabalhados os pontos, em sequência as retas e seu relacionamento com o espaço. Posteriormente, são apresentados os planos, soltos no espaço e suas relações (SANTOS, 2016). Esta é “uma abordagem extremamente abstrata, com pouca relação com o mundo concreto e totalmente dissociada das experiências prévias dos estudantes” (TEIXEIRA, *et al.*, 2007).

Na Universidade Federal do Rio Grande do Sul, mais precisamente nos diversos cursos de Engenharia, Arquitetura e Urbanismo, Design de Produto e Design Visual, essa abordagem, desde 2006, vem sendo substituída por uma mais próxima do mundo concreto, a metodologia de ensino-aprendizagem Baseada em Projetos. Esta metodologia utiliza projetos que buscam a solução de problemas, tomada de decisão e atividades investigativas e de pesquisa, que resultam produtos ou apresentações. Desta forma, tende a organizar o processo de aprendizagem o que facilita a compreensão dos discentes (TEIXEIRA, *et al.*, 2007).

Com o uso de objetos sólidos, os alunos podem desenvolver projetos complexos a partir dos conceitos de Geometria Descritiva. Os alunos são desafiados a construir sólidos com propriedades específicas e a GD é utilizada como ferramenta no processo de projeto (TEIXEIRA, *et al.*, 2007).

A utilização de objetos sólidos em vez dos habituais ponto, reta e plano, tende a ser mais palpável aos estudantes. Neste caso, há a redução da abstração necessária para a compreensão do conteúdo devido à inversão na ordem de ensino, pois os sólidos apresentam em sua composição vértices (pontos), arestas (retas) e faces (planos) os mesmos elementos trabalhados individualmente na abordagem tradicional (TEIXEIRA, *et al.*, 2007).

Além da utilização da metodologia baseada em projetos, o grupo de pesquisa Virtual Design (ViD) da Universidade Federal do Rio Grande do Sul (UFRGS), pioneiro

“no desenvolvimento de ferramentas e metodologias com o objetivo de melhorar a qualidade do ensino de GD, buscando a melhoria do processo de aprendizagem dos alunos” (TEIXEIRA, *et al.*, 2007), desenvolveu um aplicativo gráfico, o HyperCAL ^{3D} “para permitir a visualização e interação com objetos estudados, garantindo que os objetivos desta abordagem de ensino sejam alcançados” (SANTOS, 2016).

O HyperCAL ^{3D} vem sendo aprimorado desde sua criação em 2006 e sua utilização já se provou no tempo como significativa ao ensino de GD, sua versão atual foi desenvolvida para uso em computadores. A pandemia de corona vírus de 2020, acendeu a necessidade de utilização de todos os recursos disponíveis para ao ensino. Dispositivos móveis como *smartphones* são abundantes entre os estudantes, no entanto *softwares* como o HyperCAL ^{3D} não são simples de serem adaptados às tecnologias móveis. Contudo, se por um lado as dificuldades de adaptação são flagrantes, por outro seu aproveitamento pode significar ampliar o acesso os estudantes.

1.2. DELIMITAÇÃO DO TEMA

A utilização do HyperCAL ^{3D} como ferramenta de ensino de geometria descritiva na Universidade Federal do Rio Grande do Sul, já foi validada pelo uso durante os mais de 10 anos de seu desenvolvimento. Por este motivo, justificar a sua relevância nos cursos de Engenharia, Arquitetura e Urbanismo, Design de Produto e Design Visual da UFRGS não é o foco desta pesquisa.

No entanto, a adaptação do software para dispositivos móveis não passa apenas pela alteração da linguagem de programação. Questões como tamanho de tela, processamento, ausência de periféricos, interfaces gestuais e entrada de dados, são extremamente importantes em dispositivos móveis.

Para tanto, são debatidos conceitos pertinentes ao m-learning e seu desenvolvimento, assim como princípios de usabilidade e interfaces gestuais. Estes temas aliados a metodologias ágeis de desenvolvimento de aplicativos, servem de base para o projeto e execução da adaptação do software de ensino de geometria descritiva HyperCAL ^{3D}, utilizado nos cursos de Arquitetura, Design e Engenharia da UFRGS, ao mobile learning.

1.3. PROBLEMA DE PESQUISA

Como promover a ampliação ao acesso do *software* de ensino de geometria descritiva HyperCAL ^{3D}, a partir da utilização dos dispositivos móveis pertencentes aos estudantes em uma perspectiva do *mobile learning*?

1.4. HIPÓTESE

A adaptação do HyperCAL ^{3D} à tecnologia móvel pode promover a ampliação do acesso a ferramenta de ensino de geometria descritiva, auxiliando a visualização e o entendimento das operações gráficas principais, podendo potencializar os estudos no campo de *mobile learning*, a reflexão dos alunos sobre a disciplina.

1.5. OBJETIVOS

1.5.1. Objetivo Geral

Adaptar o *software* de ensino de geometria descritiva HyperCAL ^{3D}, utilizado nos cursos de Arquitetura e Urbanismo, Design de Produto e Design Visual e diversos cursos de Engenharia da UFRGS, ao *mobile learning*, tendo como base os conceitos de desenvolvimento de *m-learning* apoiados na usabilidade de dispositivos móveis e interfaces gestuais.

1.5.2. Objetivos Específicos

Compreender as premissas e conceitos que envolvem o desenvolvimento de *mobile learning*, desde a concepção da ideia até a produção efetiva do aplicativo.

Estudar a usabilidade e interfaces gestuais de dispositivos móveis a fim de identificar requisitos para análise de aplicativos similares ao HyperCAL ^{3D}.

Analisar as funcionalidades e características de ferramentas similares ao HyperCAL ^{3D} a fim de buscar soluções para o desenvolvimento da adaptação para dispositivos móveis do programa.

Compreender a organização e estrutura construtiva, assim como o sistema de classes do HyperCAL ^{3D} e as adaptações previamente produzidas, para juntamente com os requisitos encontrados nos similares propor uma adaptação ao mobile.

Implementar as adaptações dos requisitos encontrados para a adaptação do HyperCAL^{3D} para sua versão móvel.

Submeter a adaptação do HyperCAL^{3D} a avaliação de usabilidade.

1.6. JUSTIFICATIVA DA PESQUISA

Uma das disciplinas básicas mais importantes nos cursos de Engenharia, Arquitetura e Design, a Geometria Descritiva auxilia no aprimoramento do raciocínio lógico e da visão espacial dos acadêmicos, desenvolvendo habilidades primordiais à criação de projetos (TEIXEIRA, *et al.*, 2007).

A geometria descritiva tem por objetivo a consciência espacial dos objetos tridimensionais a partir de suas projeções, desta forma é de suma importância para as áreas da Engenharia, Arquitetura e Urbanismo e Design. No entanto, não foram encontrados aplicativos para dispositivos móveis que se proponham ao ensino de geometria descritiva, assim como estudos que buscam o *mobile learning* de GD.

O ensino de GD, em especial na Universidade Federal do Rio Grande do Sul, graças ao Grupo de Pesquisa ViD, vem evoluindo com a utilização da metodologia de ensino-aprendizagem Baseada em Projetos. Esta evolução, resultou em diferentes estudos e em ferramentas que possibilitam a melhor compreensão do conteúdo, como é o caso do HyperCAL^{3D}.

A criação do software gráfico HyperCAL^{3D}, permitiu a aproximação do ensino à realidade vivenciada pelos acadêmicos que, em sua maioria, cresceram rodeados de tecnologias como computadores, celulares, câmeras digitais, vídeo games etc. Porém, mesmo com o pioneirismo do Grupo ViD, a pandemia de Corona Vírus deixou exposta uma necessidade na evolução do ensino de GD e do HyperCAL^{3D}, a necessidade de adaptação a tecnologia móvel.

Apesar do que possa parecer, esta não é uma tarefa simples e necessita da compreensão de diferentes áreas de conhecimento, desde o entendimento da aprendizagem móvel ou *mobile learning*, passando pela aceitação dos estudantes a tecnologia e utilização de seus dispositivos pessoais, até efetivo o desenvolvimento da aplicação para adaptação em diferentes dispositivos móveis.

O *Mobile learning* ou *m-learning* é um campo de estudo que objetiva a aprendizagem a partir de dispositivos móveis. Mesmo não possuindo uma definição universalmente aceita, apresenta uma crescente mobilização em seu estudo, deixando expostas algumas lacunas em suas abordagens.

Por se tratar de um campo multidisciplinar, um dos principais *gaps* está no desenvolvimento de aplicações para o suporte ao *m-learning*. Alguns estudos representam esforços no sentido de compreender a aceitação da tecnologia, no entanto utilizam tecnologias que inicialmente não foram pensadas para o ensino móvel, como WhatsApp. Desta forma, é difícil precisar se a aceitação ocorre pois o aplicativo já é utilizado no cotidiano dos alunos ou se é porque efetivamente contribui para o *mobile learning*.

A utilização de modelos de aceitação pode ser bastante interessante para a análise de uma tecnologia, mesmo em momentos iniciais da idealização do aplicativo, tem a capacidade de indicar um caminho a seguir na busca por soluções efetivas para o *m-learning*. Porém, seus percursos são genéricos, subjetivos e não demonstram diretrizes claras a serem seguidas na efetivação do projeto. Por outro lado, a usabilidade apresenta um viés prático para a criação de ferramentas digitais, no entanto não deixa claro panoramas mais amplos de aceitação.

Aproveitar os recursos existentes aos alunos, foi uma necessidade surgida da adversidade da pandemia. Ampliar o alcance e a adaptabilidade do software HyperCAL^{3D} é uma necessidade e o próximo passo em uma história de estudos do grupo ViD iniciada em 2006. Estes mais de 15 anos de estudo e evolução no ensino de Geometria Descritiva, representam o pioneirismo e a resistência para manter a qualidade do ensino diante das adversidades.

Para a autora, em seus estudos em Computação Gráfica (CG), esta é a primeira oportunidade de estudar a CG a partir do código. Esta experiência apresenta uma oportunidade de aprendizagem sem precedentes. Vale ressaltar, que os desafios são proporcionais ao conhecimento adquirido e que a experiência prévia com a CG permitiu o avanço do trabalho.

Pelo exposto, esta pesquisa se justifica por pelo menos dois motivos principais, o avanço nos estudos de desenvolvimento de *mobile learning*, em especial ao *m-learning* de GD e a ampliação da contribuição do grupo ViD ao ensino de Geometria Descritiva, a partir da ampliação do acesso ao software HyperCAL ^{3D}.

2. FUNDAMENTAÇÃO TEÓRICA

O presente capítulo apresenta o referencial teórico utilizado nesta pesquisa. Os itens a seguir discorrem sobre temas pertinentes à busca de requisitos para o projeto do artefato digital, produto deste trabalho. Inicialmente, é necessário o entendimento dos conceitos, características e parâmetros necessários ao desenvolvimento de *mobile learning*. Considerando que o tema envolve a criação de ferramentas digitais, são analisadas também questões pertinentes à usabilidade com especial atenção aos dispositivos móveis. Além destes temas, é apresentado um breve histórico do HyperCAL^{3D} e suas ferramentas, assim como métodos utilizados para o desenvolvimento de aplicações móveis.

2.1. MOBILE LEARNING

O ensino a partir de dispositivos eletrônicos, ou *e-learning*, vem sendo utilizado há alguns anos como forma complementar ao ensino presencial em universidades ao redor do mundo (SITAR-TAUT e MICAN, 2021). Aproximadamente no mesmo período, os computadores pessoais e o acesso à internet se tornaram portáteis. Aparelhos celulares ganharam novas formas de interação, deixaram de ser apenas acessórios utilizados para a comunicação para se tornar controles de outros dispositivos, centralizadores de documentos e cartões de banco, além de outras funcionalidades.

Estas mudanças, alteraram de forma significativa vários aspectos da vida cotidiana, o que inclui o ensino. No entanto, antes de elencar as características e conceitos que envolvem a aprendizagem por dispositivos móveis, ou *mobile learning*, vale ressaltar que apenas *smartphones* e *tablets* são considerados para esta pesquisa dentro desta classificação. Isso se deve, a sua natureza móvel, multiplicidade de funções, a conectividade e interação exclusivamente via *touch screen*. Além desta definição, é importante colocar que as compreensões de *mobile learning* e seu desenvolvimento são produtos da revisão sistemática de literatura descrita no capítulo 4 deste relatório.

Mobile learning ou *m-learning* é um campo de estudo que visa a aprendizagem via dispositivos móveis, sua definição ainda não é totalmente consensual, portanto, a seguir são levantados alguns pontos referentes às tentativas de sua conceituação.

Definição

A aprendizagem móvel ou *mobile learning* é um campo de pesquisa que vem sendo debatido há algum tempo em conferências, seminários e workshops. Segundo Traxler (2007) a primeira aparição do termo *m-learning* aconteceu em 2002 na cidade de Birmingham. Mesmo após quase 20 anos, ainda não há uma definição universalmente aceita e sim diferentes abordagens para as tentativas de definir ou conceituar o *mobile learning*.

Ainda de acordo com o autor, algumas definições conceituam o *m-learning* em termos de suas tecnologias e hardware, em outras palavras, que é entregue ou suportado por tecnologias portáteis e móveis. Outras definições se concentram na mobilidade que pode ser dividida em três áreas: mobilidade da tecnologia, mobilidade do aluno e mobilidade da aprendizagem (EL-HUSSEIN e CRONJE, 2013).

No entanto, Traxler (2007) analisa que o *m-learning* deveria ser observado como um fenômeno cultural, pois os dispositivos móveis fazem parte de diversos aspectos da vida cotidiana atual.

Não há dúvidas que os dispositivos móveis transformaram a forma como as pessoas se comunicam, interagem com conteúdo e se distraem. Portanto, a questão do momento procura compreender como a educação pode utilizar este dispositivo para facilitar o ensino e aprendizagem.

Zhang (2015) avalia que apesar da tecnologia móvel fazer parte do dia a dia das pessoas, o ensino e aprendizagem via *mobile* ainda se encontra em estágio embrionário. Segundo a autora, isso se deve a diferentes fatores:

Como alterar o currículo tradicional de ensino e materiais em conteúdos digitais, como projetar um bom conteúdo que se adapte a dispositivos móveis e métodos móveis de ensino, como melhorar as funções interativas e de comunicação em programas móveis, como educar educadores seniores a usar tecnologias móveis em seus ensinamentos, como proteger ip e informações confidenciais on-line, como manter redes e sinais estáveis durante o uso e como envolver os alunos em função de aprendizagem em vez de jogar jogos (ZHANG, Y.A., 2015).

Por outro lado, Crompton (2013a) identifica a definição de *mobile learning* centrada em quatro construtos: pedagogia, dispositivos tecnológicos, contexto e

interações sociais. Portanto, define *m-learning* como "aprender em vários contextos, através de interações sociais e de conteúdo, usando dispositivos eletrônicos pessoais" (CROMPOTON, 2013a).

Os construtos pedagogia, dispositivos tecnológicos e interações sociais, elencados por Compton (2013a), são autoexplicativos. No entanto, o contexto pode ser dividido em aprendizagem formal e informal. Na aprendizagem formal, os objetivos e o processo de aprendizagem são explicitamente definidos por um professor ou por um estabelecimento de ensino, portanto a aprendizagem móvel é utilizada como parte integrante de um contexto mais amplo e não o substitui completamente. Na aprendizagem informal, o aluno normalmente define os objetivos e o processo de aprendizagem (CROMPOTON, 2013a); (RIKALA, 2015).

Para Fuller e Joynes (2015) deveria haver menor ênfase em se a aprendizagem móvel deve ser implementada e mais no desenvolvimento de aprendizagem móvel que seja abrangente, sustentável, significativa e obrigatória.

Com diferentes modelos, sistemas e tecnologias, os dispositivos móveis individuais variam, entre outros motivos, de acordo com o orçamento, gosto pessoal e necessidade do usuário. A complexidade envolvida na adaptação do conteúdo a diferentes tipos de dispositivos, sistemas operacionais distintos, multiplicidade de linguagens de desenvolvimento, diferentes níveis de habilidade dos designers e educadores (ZHANG, Y.A., 2015), além das rápidas mudanças na tecnologia, se provaram um grande desafio as pesquisas de *m-learning* (ALRASHEEDI, CAPRETZ e RAZA, 2015).

Portanto, na tentativa de compreender o cenário do desenvolvimento de *mobile learning*, no próximo item são debatidas questões de desenvolvimento de ferramentas *m-learning*.

Desenvolvimento de *mobile learning*

O arcabouço teórico que embasa o desenvolvimento de *mobile learning*, na construção desta pesquisa, foi obtido a partir da revisão sistemática de literatura que pode ser encontrada em sua íntegra no capítulo 4 deste relatório.

Foram realizadas buscas nas bases SCOPUS (<http://www.scopus.com>) e IEEE Xplore Digital Library (<http://ieeexplore.ieee.org>) pela string "Mobile Learning development" or "mobile learning design" or "Mobile Learning model" or "Mobile Learning framework" or "m-learning development" or "m-learning design" or "m-learning model" or "m-learning framework" and "higher education", que retornaram 536 artigos.

Os resultados obtidos foram enviados para na ferramenta Rayyan (<https://rayyan.ai>) (OUZZANI, *et al.*, 2016) para a aplicação dos demais critérios de inclusão e exclusão e retornaram quatorze artigos para a análise e síntese.

Dos artigos analisados, dez tratam de análises, dos quais sete representavam análises de aceitação, na maior parte pelos estudantes, mas também foram encontrados artigos que lidavam com a aceitação por parte dos professores. Isso sugere que, "o desejo dos alunos de aprender em seus dispositivos móveis pessoais não deve ser uma suposição automática, pois alguns deles sentem fortemente em manter a vida pessoal separada da acadêmica"⁴ (TABOR, 2016). Outros fatores apontados pelos alunos para a rejeição da aprendizagem móvel recaem sobre a dificuldade de aprendizagem da tecnologia, o tamanho das telas e a performance dos equipamentos (TABOR, 2016).

Estas análises, em sua maioria, se baseiam em modelos de aceitação de tecnologia que representa uma área de teoria dos sistemas de informação que estuda fatores que levam os usuários a aceitarem ou não uma tecnologia. A seguir são apresentados alguns modelos de análise encontrados. A ordem de apresentação foi escolhida, devido ao fato de os modelos representarem uma linha evolutiva.

Modelos de análise de aceitação do *Mobile Learning*

Vários modelos para a análise de aceitação de tecnologia foram utilizados nos artigos investigados. Porém, a maiores ocorrências foram dos modelos TAM (*Technology Acceptance Model*), e o UTAUT (*Unified theory of acceptance and use of technology*). Estes modelos foram aplicados tanto de forma individual, como

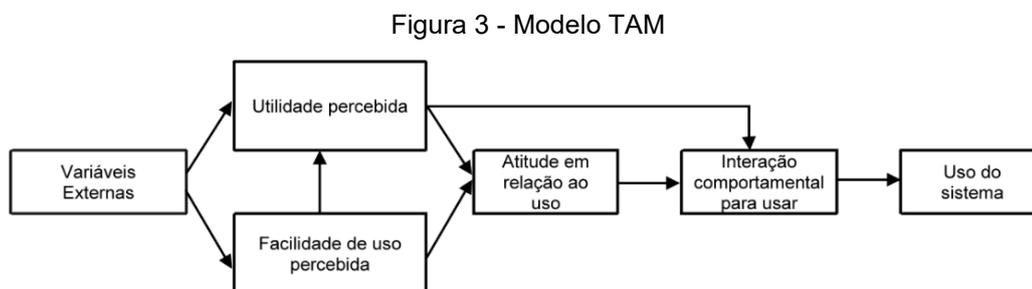
⁴ Tradução livre do texto- "*desire to learn on their personal mobile devices should not be an automatic assumption, as some of them feel strongly about keeping their person-al lives separate from their academics*" (TABOR, 2016).

associadas a outros modelos na formação de novos modelos de análise. Os dois modelos, podem ser utilizados para qualquer tipo de tecnologia e não ficam restritos aos estudos de *m-learning*. Portanto, são apresentados de forma ilustrativa na evolução dos modelos de aceitação.

2.1.1. Modelo TAM

O modelo de aceitação de tecnologia (TAM), procura compreender os porquês de os usuários aceitarem ou rejeitarem determinada tecnologia, de forma a tentar prever e melhorar a aceitação. Ele foca em dois construtos: a utilidade percebida e a facilidade de uso percebida. Em outras palavras, as pessoas buscam a tecnologia como forma de aprimorar seu desempenho em alguma tarefa, porém tendem a abandoná-la caso seja difícil de aprender a usá-la.

O modelo TAM tenta explicar por que alguém escolhe utilizar uma tecnologia e detrimento de outra. Para tanto, parte do princípio de que a tecnologia é escolhida por ser útil e fácil de usar. No entanto, esta percepção não possui ligação com a tecnologia em si, e sim com critérios individuais de cada indivíduo. A Figura 3 apresenta o esquema gráfico do modelo TAM.



Fonte: Adaptado de Davis *et al.* (1989) traduzido pela autora

O modelo foi desenvolvido por Fred Davis, Richard Bagozzi e Paul Warshaw (1989), atualmente apresenta novas versões e é considerado um dos mais utilizados para a análise de aceitação.

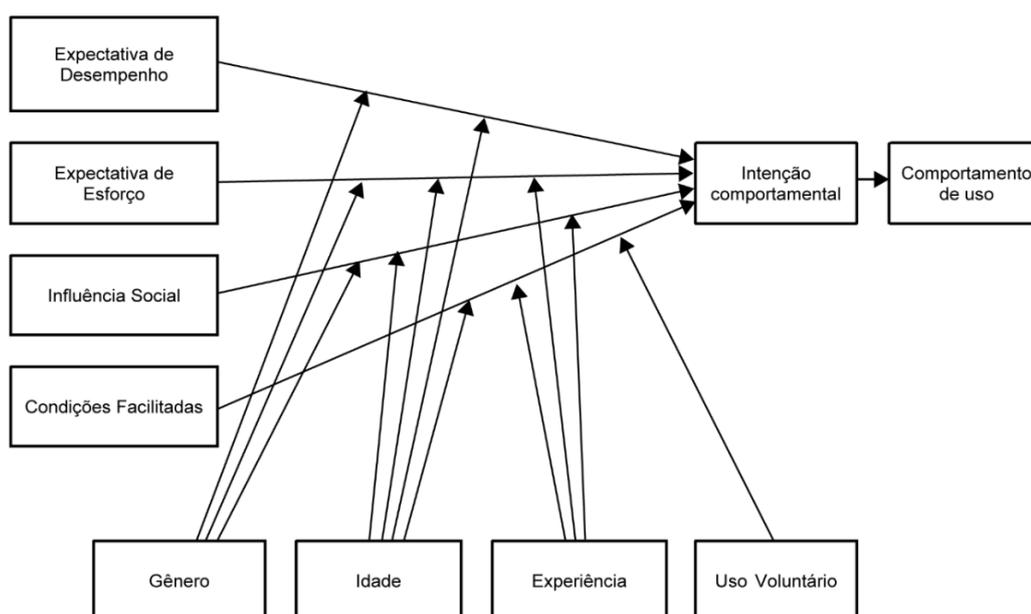
2.1.2. Modelo UTAUT

O modelo UTAUT surgiu da união de oito modelos de aceitação da tecnologia, *Theory of Reasoned Action (TRA)*, *Technology Acceptance Model (TAM)*, *Motivational Model (MM)*, *Theory of Planned Behavior (TPB)*, *Combined TAM and TPB (C-TAM-*

TPB), *Model of PC Utilization* (MPCU) e *Innovation Diffusion Theory* (IDT) (VENKATESH, *et al.*, 2003). É baseado na validação de quatro fatores: expectativa de performance, expectativa de esforço para o uso, influência social e condições facilitadas.

- **Expectativa de performance:** a compreensão do usuário de como a tecnologia aprimora sua performance. Em outras palavras, ele deve compreender a tecnologia como uma forma de melhorar seu desempenho em determinada tarefa.
- **Expectativa de esforço para o uso:** o entendimento do usuário quanto a facilidade de utilização da tecnologia. A tecnologia deve ser simples de ser utilizada.
- **A influência social:** a percepção do usuário de como pessoas influentes o vem positivamente por usar determinada tecnologia. Refere-se ao status social que determinada tecnologia pode promover ao usuário.
- **As condições facilitadas:** como o usuário percebe a infraestrutura técnica e organizacional que apoia a utilização do sistema. A avaliação do usuário quanto a solidez da tecnologia.

Figura 4 - Modelo UTAUT



Fonte: Adaptado de Venkatesh *et al.* (2003) traduzido pela autora

Os fatores de aceitação não são utilizados isoladamente, devem ser regulados por quatro moderadores: gênero, idade, experiência e voluntariedade de uso pelo usuário. A Figura 4 demonstra as inter-relações entre os fatores e moderadores que são detalhadas a seguir.

Expectativa de Desempenho deve ser moderada pelo gênero e idade

Pesquisas indicam que indivíduos que se identificam com o gênero masculino são predispostos a ser orientados para a realização de tarefas, portanto, as expectativas de desempenho, que se concentram na realização de tarefas, provavelmente serão especialmente apreciadas por este grupo (VENKATESH, *et al.*, 2003).

Já em se tratando de idade, trabalhadores mais jovens podem colocar mais importância nas recompensas extrínsecas. No entanto, é importante perceber que a análise de gênero e idade deve ser analisada em conjunto, pois há uma tendência de alteração no papel social para indivíduos de cada gênero em fases de vida diferentes (VENKATESH, *et al.*, 2003).

Expectativa de Esforço deve ser moderada por gênero, idade e experiência

Para Venkatesh, Morris, Grodon Davis e Fred Davis (2003) indivíduos que se identificam com o gênero feminino tendem a ser mais afetados pela expectativa de esforço. Já com relação a idade, há uma tendência de pessoas mais velhas apresentarem dificuldade em processar estímulos complexos e alocar atenção às informações sobre o trabalho. No entanto, este fato pode ser subvertido pela experiência do usuário no uso do sistema. (VENKATESH, *et al.*, 2003).

Influência Social deve ser moderada por gênero, idade, voluntariedade e experiência

Teorias sugerem que pessoas que se identificam com o gênero feminino possuem uma inclinação a ser mais sensíveis a opinião dos outros, à medida que se voluntariam a aprender uma nova tecnologia. Com relação a idade, o engajamento deve ser diretamente proporcional e a experiência inversamente proporcional a idade.

Isso significa que, indivíduos mais velhos valorizam a influência social, porém quanto maior sua experiência na utilização de determinada tecnologia, menor é sua necessidade de aceitação social (VENKATESH, *et al.*, 2003).

Condições Facilitadoras deve se moderada por idade, experiência e voluntariedade.

Devido as limitações cognitivas decorrentes do envelhecimento, trabalhadores mais velhos dão mais importância ao recebimento de ajuda. No entanto, este fato é alterado de acordo com a experiência do usuário no uso da tecnologia. A voluntariedade somente terá efeito nas condições facilitadoras caso não ocorra nenhum treinamento para a utilização da tecnologia (VENKATESH, *et al.*, 2003).

Como mencionado, o modelo UTAUT é uma evolução do modelo TAM a seguir é apresentado o modelo MLAM que se propõem a avaliação de aceitação específica do *mobile learning*.

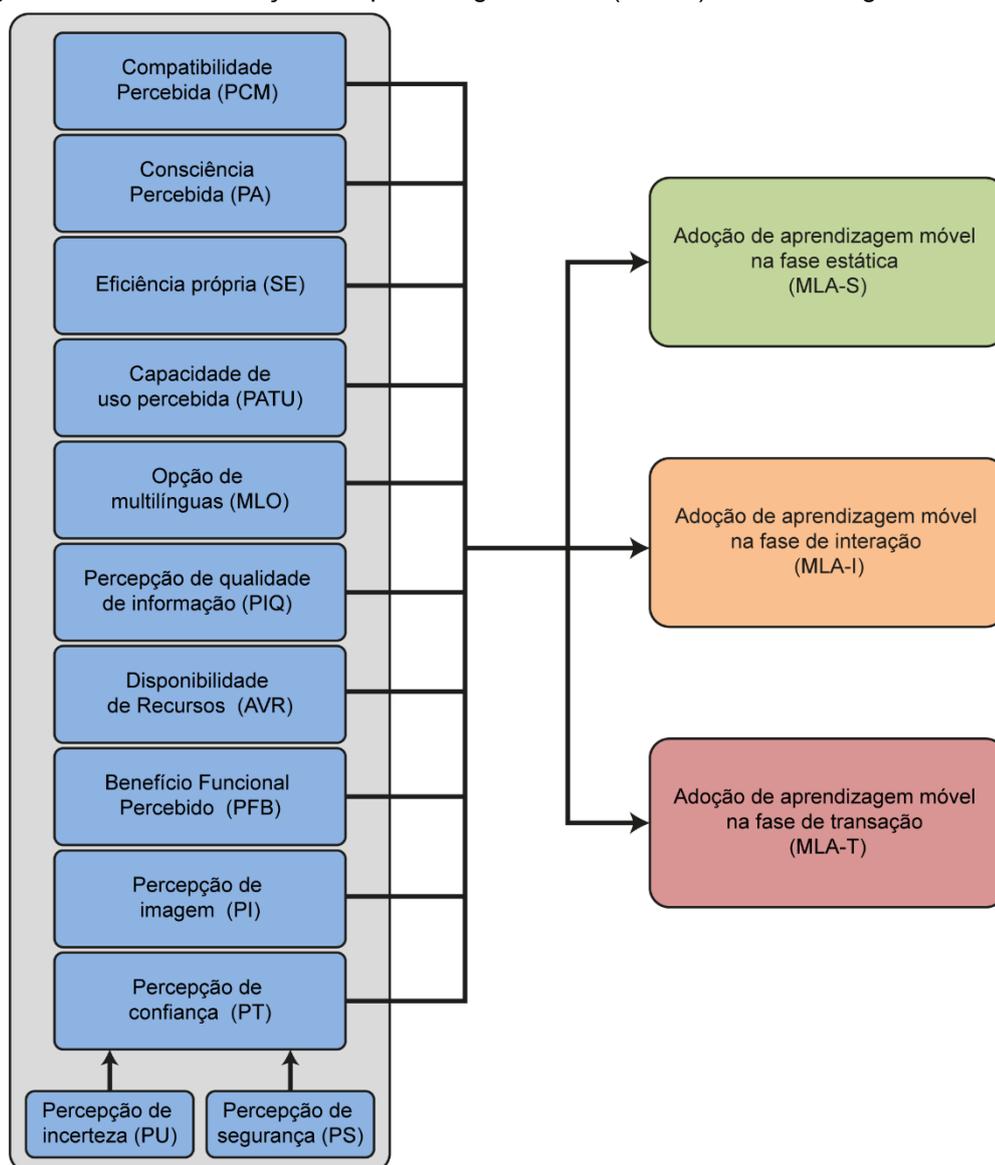
2.1.3. Modelo MLAM

No artigo “*Analysis the Effect of Different Factors on the Development of Mobile Learning Applications at Different Stages of Usage*” os autores Almaiah, Alamri e Al-Rahmi (2019) partem do modelo UTAUT para criar o seu próprio modelo *Mobile Learning Adoption Model* (MLAM) de análise específico para *m-learning*.

No modelo MLAM, são utilizados construtos para a definição da aceitação, assim como no modelo UTAUT, porém estes não são regulados por outros fatores e sim divididos em diferentes fases de uso. Para os autores, modelo MLAM abrange a adoção de aprendizagem móvel em vários aspectos como tecnologia, funções do sistema, requisitos de segurança e fatores humanos. (ALMAIAH, ALAMRI e AL-RAHMI, 2019).

O novo modelo, diferente dos citados anteriormente, leva em consideração características do sistema, o que para os autores alteram a percepção dos usuários quanto sua aceitação. A Figura 5 representa o esquema gráfico do modelo MLAM, com seus construtos e fases que serão detalhados a seguir.

Figura 5 - Modelo de adoção de aprendizagem móvel (MLAM) em três estágios de serviço.



Fonte: Almaiah, Alamri e Al-Rahmi (2019) tradução da autora

Neste modelo, o entendimento do usuário se diferencia de acordo com sua percepção da complexidade do sistema. Por este motivo, Almaiah, Alamri e Al-Rahmi (2019) dividiram o *m-learning* em três fases principais de acordo com sua complexidade e interação: fase estática, fase de interação e fase de transação.

- **Fase estáticas (MLA-S):** nesta fase, os alunos podem acessar informações como datas de inscrição, cursos de aprendizagem, anúncios universitários, datas do exame, atribuições, notas, situação financeira, baixar cursos de aprendizagem e carregar atribuições (ALMAIAH, ALAMRI e AL-RAHMI, 2019).

- **Fase de interação (MLA-I):** nesta fase, os alunos podem se comunicar facilmente com instrutores de qualquer lugar sobre quaisquer consultas usando comunicações bidirecionais, como e-mail e sistema de bate-papo (ALMAIAH, ALAMRI e AL-RAHMI, 2019).
- **Fase de transação (MLA-T):** nesta fase, os alunos podem se inscrever nas aulas virtuais, adicionar e excluir novos cursos, responder aos questionários e pagar taxas de inscrição conectando-se com suas contas bancárias através de tecnologia de autoatendimento, interações virtuais e confirmação de tarefas. (ALMAIAH, ALAMRI e AL-RAHMI, 2019).

Durante sua investigação para a criação do novo modelo, os autores perceberam que cada fase de uso necessita de um grupo determinado de construtos para a realização da análise.

Na fase estática os fatores de maior relevância para a adoção do serviço de aprendizagem móvel foram: qualidade da informação (PIQ), benefício funcional percebido (PFB), confiança percebida (PT), capacidade de uso percebida (PATU), compatibilidade percebida (PCM), percepção de conscientização (PA) e disponibilidade de recursos (AVR).

Na fase de interação foram: compatibilidade percebida (PC), a consciência percebida (PA), a qualidade da informação percebida (PIQ), o benefício funcional percebido (PFB), a autoeficácia tecnológica (SE) e a capacidade de uso percebida (PATU)

Os resultados também indicaram que a confiança percebida (PT), por imagem percebida (PI) e opção multilíngue (MLO), não têm efeito significativo na adoção de serviços de aprendizagem móvel na fase de interação.

Na fase de transação foram: compatibilidade percebida (PC), consciência percebida (PA) segurança percebida (PS), benefício funcional percebido (PFB), incerteza percebida (PU) e capacidade de uso percebida (PATU).

O Quadro 1, apresenta os construtos do modelo MLAM, suas definições conceituais e hipóteses definidas pelos por Almaiah, Alamri e Al-Rahmi (2019).

Quadro 1 - Definição conceitual dos construtos e hipóteses do MLAM

Construtor	Definições conceituais	Hipóteses
Compatibilidade Percebida (PC)	O grau em que um aplicativo de aprendizagem móvel é percebido como consistente com as necessidades e percepções dos usuários potenciais.	Compatibilidade Percebida (PCM) tem uma relação significativa com a adoção de aprendizagem móvel.
Consciência Percebida (PA)	O grau de consciência dos usuários através da aquisição de conhecimentos é suficiente para aprender as características do aplicativo de aprendizagem móvel, usá-lo com habilidade e realizar suas funções, vantagens e desvantagens.	A Consciência Percebida (PA) tem uma relação significativa com a adoção da aprendizagem móvel
Disponibilidade de Recursos (AOR)	Disponibilidade e liberdade de uso de dispositivos móveis, aplicativos móveis e internet com recursos competitivos como velocidade, acesso e custo.	Disponibilidade de Recursos (AOR) tem relação significativa com a adoção da aprendizagem móvel
Eficiência Própria (SE)	O grau de capacidade tecnológica dos usuários para usar, interagir e transacionar com aplicativos de aprendizagem móvel com base em conhecimento, experiência e habilidade prévios à medida que percebem que é necessário fazê-lo.	A Eficiência Própria (SE) tem relação significativa com a adoção da aprendizagem móvel
Capacidade de uso percebida (PATU)	O grau em que um usuário percebe sua competência confortável para usar o aplicativo de aprendizagem móvel tecnologicamente, organizacional e psicologicamente que correspondem aos valores, necessidades sociais e atitudes gerais do indivíduo.	A Capacidade de uso percebida (PATU) tem uma relação significativa com a adoção da aprendizagem móvel
Opção Multilíngue (MLO)	O aplicativo de aprendizagem móvel suporta diferentes idiomas primos para facilitar a visualização, pesquisa, seleção, download, interação e transação com sua linguagem conveniente na ausência de interação humana	Opção Multilíngue (MLO) tem relação significativa com a adoção da aprendizagem móvel
Qualidade da Informação Percebida (PIQ)	A qualidade das informações abrange até que ponto informações precisas, organizadas, compreensíveis, atualizadas e oportunas são fornecidas no aplicativo de aprendizagem móvel para que os usuários obtenham informações sobre qualquer um de seus serviços pretendidos	A Qualidade da Informação Percebida (PIQ) tem uma relação significativa com a adoção da aprendizagem móvel
Confiança Percebida (PT)	A confiança de um usuário na capacidade do aplicativo de aprendizagem móvel de fornecer um serviço confiável e eficiente	Confiança Percebida (PT) tem relação significativa com a adoção da aprendizagem móvel
Incerteza Percebida (PU)	O grau em que os usuários percebem o risco nas transações devido a situações incontroláveis e desconhecidas no ambiente virtual associada ao aplicativo de aprendizagem móvel	A incerteza percebida tem uma relação significativa com a confiança percebida da aprendizagem móvel
Segurança Percebida (PS)	O grau em que os usuários percebem que o nível de privacidade de dados e integridade de dados é eficiente e garante segurança para todas as transações eletrônicas e autenticação de identidade on-line via aplicativo de aprendizagem móvel.	A Segurança Percebida tem uma relação significativa com a confiança percebida
Benefício Funcional Percebido (PFB)	O grau em que os usuários percebem os benefícios funcionais gerais, incluindo custo, tempo, eficiência e eficácia do uso do aplicativo de aprendizagem móvel — em vez de usar funções tradicionais de escritório físico.	Benefício Funcional Percebido (PFB) tem relação significativa com a adoção da aprendizagem móvel
Imagem percebida (PI)	O grau em que os usuários percebem de forma comportamental e cultural essa adoção significativa do aplicativo de aprendizagem móvel melhora e melhora o status social e o prestígio	Imagem Percebida (PI) tem relação com a adoção da aprendizagem móvel

Fonte: Almaiah, Alamri e Al-Rahmi (2019) tradução da autora

Desta forma, o modelo MLAM além de ser voltado ao *mobile learning* procura direcionar a aceitação para as diferentes necessidades de recursos do sistema, pois para os autores, os recursos alteram a percepção do usuário quanto a sua aceitação.

Apesar de os modelos de aceitação apresentarem conceitos importantes sobre a intensão de uso, estes não apresentam diretrizes práticas para a busca de requisitos em aplicativos similares. Por este motivo, no próximo item são debatidas questões que envolvem a usabilidade, iniciando pela definição e chegando a sua utilização em dispositivos moveis. Para esta pesquisa, sua utilização representa uma aproximação com o Design, assim como a busca de requisitos para a análise de similares e a construção da prática do projeto.

2.2. USABILIDADE

Nos próximos item são apresentadas definições, métodos de avaliação de usabilidade de dispositivos fixos e moveis.

Definição

Conhecido como um dos maiores especialistas em usabilidade nos Estados Unidos (CYBIS, BETIOL e FAUST, 2010), Jakob Nielsen (1993) define usabilidade como um atributo qualitativo, que determina a facilidade ou dificuldade de utilização de uma interface. O autor analisa que a usabilidade tem múltiplos componentes e é tradicionalmente associado a esses cinco atributos:

- **Aprendizado:** a facilidade de aprendizagem do sistema para que um usuário iniciante possa começar a trabalhar (NIELSEN, 1993).
- **Eficiência:** o sistema deve ser eficiente a fim de permitir um alto nível de produtividade (NIELSEN, 1993).
- **Memorabilidade:** a capacidade de memorização dos recursos e utilização do sistema permitindo que o usuário casual possa retomar o trabalho mesmo após algum tempo sem utilizá-lo (NIELSEN, 1993).
- **Erros:** baixa taxa de erro, para que os usuários cometam poucos erros durante o uso do sistema e se cometerem erros, possam facilmente recuperar deles (NIELSEN, 1993).

- **Satisfação:** deve ser agradável de usar, de modo que os usuários estejam subjetivamente satisfeitos ao usá-lo (NIELSEN, 1993).

Além da definição e dos componentes que balizam a usabilidade, à Nielsen juntamente com Molich (NIELSEN e MOLICH, 1990), é atribuída a autoria do método de Avaliação Heurística. Mesmo tendo sido criada a partir da experiência dos autores, sem fontes científicas (NIELSEN e MOLICH, 1990) é um dos métodos mais utilizados para encontrar problemas de interface (MACHADO e VERGARA, 2020) (SALGADO e FREIRE, 2014).

2.2.1. Avaliação Heurística

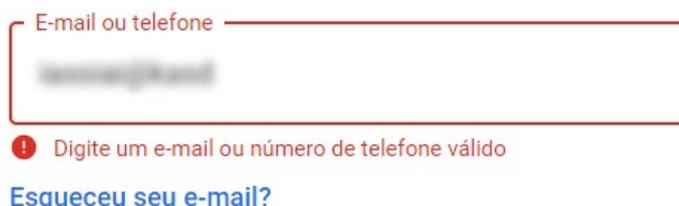
Utilizadas na maior parte do tempo, como critérios para a solução de determinados problemas, as heurísticas são provenientes de decisões não racionais, ligadas a intuição e propensas ao erro que ignoram parte da informação (GIGERENZER e GAISSMAIER, 2011). No entanto, para Gigerenzer e Gaissmaier (2011) mesmo que os modelos lógicos e estatísticos reduzam a incidência de erros, estes aplicados sem o devido rigor decorrem em falhas, além de serem onerosos e dispendiosos. Assim sendo, utilizar adaptações de heurísticas simples e ignorar parte da informação pode conduzir a julgamentos mais precisos para questões de baixa previsibilidade e pequenas amostras (GIGERENZER e GAISSMAIER, 2011). Este fato, pode explicar por que a Avaliação Heurística, mesmo não possuindo um rigor científico, se mantém como o método mais utilizado até os dias atuais.

A Avaliação Heurística consiste em um *checklist*, originalmente pensado com nove itens e posteriormente reestruturado para dez itens (NIELSEN e LORANGER, 2006), é uma avaliação sistemática de critérios (heurísticas) observados na interface a fim de identificar possíveis problemas que podem ser desconfortáveis ao usuário. Os itens que compõem a lista de checagem serão detalhados a seguir.

1. **Visibilidade do Status do Sistema:** o sistema deve manter o usuário informado, em tempo real, sobre o que está acontecendo. *Feedbacks* instantâneos, ou em um tempo razoável, orientarão o usuário sobre o status do sistema de forma que ele compreenda que algo está sendo processado ou que ele necessita realizar uma ação. Como por exemplo: animações de carregamento e campos de formulários

que ficam vermelhos caso não tenham sido preenchidos ou esteja incorreto, assim como mostra a Figura 6 quando o e-mail do google não está correto.

Figura 6 - Feedback formulário



Fonte: Gmail ([s.d.]).

2. **Correspondência entre o sistema e o mundo real:** a linguagem do sistema deve ser adequada ao usuário para reduzir a carga cognitiva durante a interação com o produto. Como por exemplo, a familiaridade dos ícones com objetos do cotidiano pode auxiliar na compreensão de sua finalidade, além disso ícones que são utilizados em diferentes aplicativos representando mesmas funcionalidades são facilmente assimilados, mesmo que os objetos que lhes deram origem já não sejam mais utilizados como no caso do disquete que representa salvamento em diferentes plataformas. Na mostra a Figura 7 os ícones do WhatsApp, do microfone e da máquina fotográfica para mensagem de áudio e imagem.

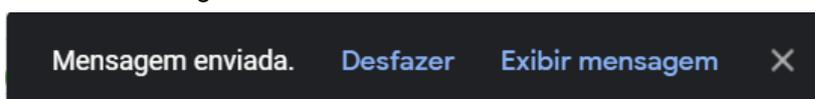
Figura 7 - Ícones do WhatsApp



Fonte: WhatsApp ([s.d.]).

3. **Controle e liberdade do usuário:** formas pelas quais o usuário pode escapar de possíveis erros de suas ações, desta forma os usuários podem se sentir seguros, pois poderão desfazer eventuais erros cometidos durante a utilização de determinada tecnologia, exemplo: botão para cancelar uma ação ou as lixeiras que permitem reciclar arquivos e e-mails deletados. A Figura 8 mostra a opção do Gmail para desfazer o envio de um e-mail.

Figura 8 - Botão cancelar envio do Gmail

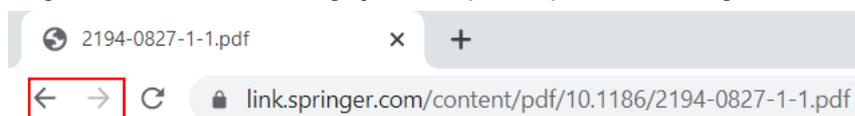


Fonte: Gmail ([s.d.]).

4. **Consistência e padronização:** padrões semelhantes utilizados em diferentes áreas do sistema evitam que o usuário se sinta perdido e necessite reaprender suas funcionalidades em diferentes momentos do uso, exemplo: a posição dos botões de

navegação deve estar sempre no mesmo lugar nas diferentes telas. Os botões de retorno e avanço se encontram sempre no topo a esquerda no navegador Google Chrome, assim como mostra a Figura 9.

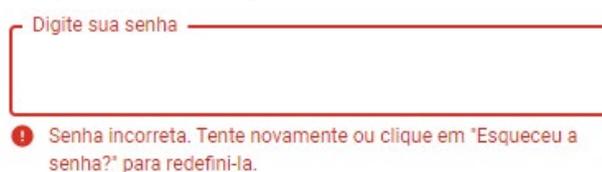
Figura 9 - Botões de navegação no top a esquerda do Google Chrome



Fonte: Navegador Google Chrome ([s.d.]).

5. Apoio ao reconhecimento, diagnóstico e recuperação de erros: mensagens de erro devem ser claras e simples, sem que causem intimidação ao usuário, exemplo: mensagens de erro de senha. A Figura 10 mostra a mensagem de erro de senha do Gmail.

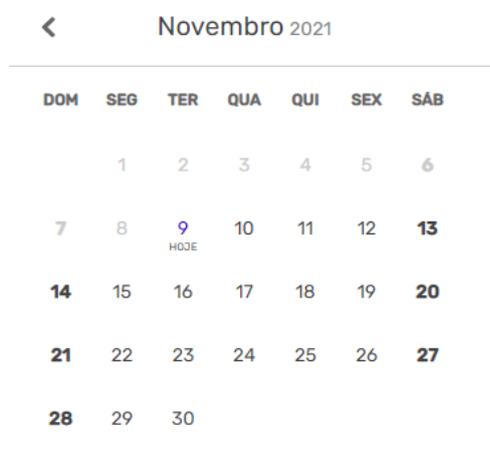
Figura 10 - Mensagem de erro na senha Gmail



Fonte: Gmail ([s.d.]).

6. Prevenção de erros: as interfaces devem ser pensadas de forma a evitar possíveis erros do usuário, exemplo: datas passadas desabilitadas em sites de compras de passagem aérea. No site Decolar apresenta o dia atual em roxo com a palavra hoje abaixo, os dias anteriores aparecem em cinza claro indicando a indisponibilidade para a escolha, assim como mostra a Figura 11.

Figura 11 - Datas passadas desabilitadas no site Decolar



Fonte: Site Decolar ([s.d.]).

7. Reconhecimento em vez de memorização: minimizar a quantidade de informações a serem memorizadas pelo usuário deixando, objetos, ações e opções importantes visíveis, facilitando o reconhecimento de padrões, exemplo: lista de Pix já realizados para evitar que o usuário precise lembrar várias chaves. A Figura 12 apresenta as transações recentes do Pix no aplicativo do Banco do Brasil com nomes e chaves para facilitar a utilização do usuário.

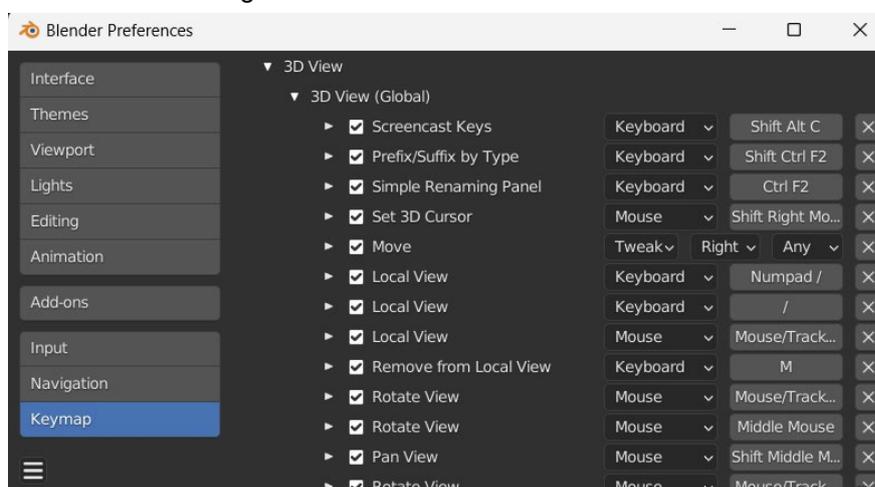
Figura 12 - Pix realizados anteriormente salvos no aplicativo do banco do Brasil



Fonte: Aplicativo Banco do Brasil ([s.d.]).

8. Flexibilidade e eficiência de uso: a interface deve ser idealizada tanto para usuários novatos como para usuários experientes. Para usuários novatos deve ser de simples compreensão e para os mais experientes deve apresentar formas de acelerar o seu trabalho, exemplo: atalhos. Na Figura 13 é possível ver a janela para a confecção de atalhos do programa Blender.

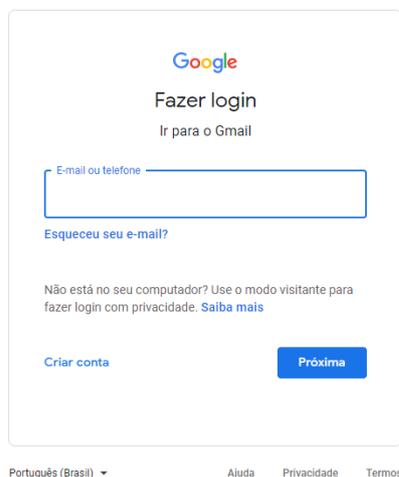
Figura 13 - Lista de atalhos do Blender



Fonte: Blender ([s.d.]).

9. **Projeto estético e minimalista:** as informações na tela devem ser sucintas e devem informar apenas o necessário para a realização de determinada ação, exemplo: a Figura 14 apresenta a tela de login do Gmail somente com informações relevantes ao login ou a uma possível criação de uma nova conta.

Figura 14 - Tela de login do Gmail



Fonte: Gmail ([s.d.]).

10. **Ajuda e documentação:** é importante que um sistema possua telas ou balões com ajuda em caso de dúvidas do usuário, exemplo: pontos de interrogação em formulários que explicam onde encontrar as informações desejadas. Na Figura 15 o site da Polícia Federal apresenta explicações sobre os documentos necessários, para tanto basta clicar nas interrogações ao lado das palavras.

Figura 15 - Interrogações no formulário para passaporte da Polícia Federal

Fonte: Site da Polícia Federal ([s.d.]).

A Avaliação Heurística é considerada um dos principais métodos de avaliação de usabilidade, mesmo tendo sido inicialmente pensada para a avaliação de interfaces de computador, atualmente, também é utilizada para a avaliação de aplicações moveis. No próximo item são debatidos métodos de análise de usabilidade em aplicativos moveis.

2.2.1. Usabilidade Móvel

Machado e Vergara (2020) realizaram uma revisão sistemática para mapear quais os métodos de usabilidade, novos e existentes, são mais utilizados para a avaliação de dispositivos móveis. Os autores partiram de três macro etapas: portfólio bibliográfico, análise bibliométrica e análise sistêmica. Desta forma, determinaram quais as bases de dados mais relevantes para a pesquisa, quais os autores se destacam na área e quais artigos mereciam destaque.

Como resultado, obtiveram a avaliação heurística de Nielsen e Molich (1990) como método existente mais utilizado na avaliação de aplicativos moveis, além de novos métodos de análise (MACHADO e VERGARA, 2020) (SALGADO e FREIRE, 2014).

Os autores pontuaram também os aspectos mais significativos de alguns trabalhos, como a necessidade de avaliações realizadas em ambiente real de uso e não somente em laboratório. Quando os aplicativos moveis são utilizados em ambientes reais, a usabilidade pode ser alterada devido a distrações do ambiente, ruídos e outras pessoas. Desta forma, os métodos de avaliação de usabilidade móvel deveriam se atentar para os contextos de uso (ALSHEHRI e FREEMAN, 2012).

Os testes realizados em laboratório possuem um custo mais acessível e são mais fáceis de serem realizados, podem ser interpretados com mais precisão devido a possibilidade de eliminação de variáveis irrelevantes ou indesejadas. São capazes de definir tarefas e procedimentos para corresponder ao objetivo do estudo e por intermédio de filmagens permitem registro das reações do usuário durante o processo de avaliação (ALSHEHRI e FREEMAN, 2012).

Mesmo que os testes em laboratório possam simular situações reais de uso, não são capazes de mimetizar totalmente o contexto de realização de tarefas

encontrados pelos usuários durante sua vida cotidiana (ALSHEHRI e FREEMAN, 2012). Desta forma, os testes de usabilidade frequentemente tratam o usuário como uma entidade estática, fato que poderia ser subvertido através de avaliações de campo (ISOMÄKI e PEKKOLA, 2011).

Por outro lado, os testes de campo apresentam maior relevância em critérios subjetivos de experiência do usuário (KRONBAUER, SANTOS e VIEIRA, 2012). Porém, dificuldades na coleta de dados, devido à natureza dinâmica do contexto móvel real e a natureza imprevisível das redes sem fio, tornam o processo de avaliação de difícil generalização.

Algumas pesquisas voltadas para a melhoria das avaliações de usabilidade em contexto real, utilizam a automação de testes a partir da captura de dados pelos sensores ou informações da tela (KRONBAUER, SANTOS e VIEIRA, 2012) . Estas pesquisas representam a busca de uma visão holística sobre os elementos de design da tecnologia móvel (ALSHEHRI e FREEMAN, 2012), no entanto confrontam “com outros aspectos éticos da privacidade do usuário e outros fatores subjetivos da interação que são difíceis de mensurar de forma automática” (MACHADO e VERGARA, 2020).

Além das discussões centradas na busca por cenários que melhor contemplem a coleta de dados para análises, alguns autores buscam por novas Heurísticas ou adaptações das Heurísticas proposta por Nielsen e Molich (1990) para o contexto móvel.

Heurísticas para dispositivos móveis

Salgado e Freire (2014) realizaram um mapeamento da literatura na tentativa de determinar os métodos existentes para a avaliação de dispositivos móveis, como resultaram obtiveram vinte e nove Heurísticas até o ano de 2013. Concluíram também, que a Avaliação Heurística ainda era o método mais utilizado para análise de aplicações moveis.

Apesar disso, o método de Nielsen e Molich (1990) deixa de lado questões como: o contexto de uso das aplicações; dificuldades de leitura devido ao tamanho da tela; entrada de dados prejudicada pela dimensão do teclado e recursos existentes

como sensores, câmeras e microfones que podem ser utilizados como facilitadores do uso. Estas questões, importantes para a avaliação e desenvolvimento de aplicações móveis, também são levantadas em pesquisas relacionadas ao *mobile learning*, que identificam fatores como dificuldade de leitura, digitação e contextos de uso como fatores importantes a serem considerados na utilização do *m-learning* (ZHANG, Y.A., 2015).

Na tentativa de atender as diferenças existentes entre a utilização de dispositivos móveis e estáticos⁵, Joyce e Lilley (2014) realizaram um estudo baseado nas Heurísticas de Nielsen e Molich (1990) com a finalidade de compilar um conjunto de heurísticas específicas para dispositivos móveis. Os autores contaram com o *feedback* quantitativo e qualitativo de sessenta especialistas de dezoito países. Como resultado, obtiveram um conjunto de treze heurísticas que foram batizadas de SMART que são apresentadas a seguir:

SMART1: Fornecer notificação imediata do status do aplicativo – Informar imediatamente ao usuário o status da aplicação.

SMART2: Use um tema e termos consistentes, bem como convenções e padrões familiares ao usuário – Usar temas e padronizações para permitir que as telas se pareçam.

SMART3: Previna erros sempre que possível; Auxiliar os usuários caso ocorra um erro – Prevenir a ocorrência de erros o máximo possível e caso estes ocorram, avisar ao usuário da sua existência e como corrigi-lo.

SMART4: Exibir uma sobreposição apontando os principais recursos quando apropriado ou solicitado – Permite que usuários iniciantes se familiarizem com o aplicativo e se tornem experientes rapidamente.

SMART5: Cada interface deve se concentrar em uma tarefa – O foco em uma tarefa garante que as interfaces móveis sejam menos desordenadas e mais simples facilitando a interação e o foco do usuário.

⁵ Notebooks devido a suas dimensões e características de uso são considerados estáticos para esta pesquisa.

SMART6: Projete uma interface visualmente agradável – Interfaces móveis atraentes são mais memoráveis e, portanto, são usadas com mais frequência.

SMART7: Interfaces intuitivas facilitam a navegação do usuário – As interfaces móveis devem ser fáceis de aprender, deixando óbvios os próximos passos.

SMART8: Crie um caminho navegável claro para a conclusão da tarefa – os usuários devem ser capazes de ver imediatamente como eles podem interagir com o aplicativo e navegar em seu caminho para a conclusão da tarefa.

SMART9: Permitir opções de configuração e atalhos – O aplicativo móvel pode permitir opções de configuração e atalhos para as informações mais importantes e tarefas frequentes.

SMART10: Atender a diversos ambientes móveis – Ambientes diversos podem apresentar más condições de iluminação e alto ruído ambiente, desta forma dificultam a utilização dos usuários.

SMART11: Facilitar a entrada de dados – Dispositivos móveis são difíceis de usar do ponto de vista de entrada de conteúdo. Utilizar entrada multimodal e manter os campos de formulário ao mínimo pode facilitar a utilização da aplicação.

SMART12: Use a câmera, o microfone e os sensores quando apropriado para diminuir a carga de trabalho dos usuários – Considere o uso da câmera, microfone e sensores para diminuir a carga de trabalho dos usuários.

SMART13: Crie um ícone estético e identificável – O ícone do aplicativo deve ser estético e identificável de forma a atrair o usuário.

Além de possuírem um número maior de itens as SMARTs levam em consideração recursos como sensores, câmeras e microfones, assim como o contexto de uso, leitura e entrada de dados que não são utilizados na Avaliação Heurística. Desta forma, procura solucionar problemas de usabilidade utilizando recursos disponíveis na tecnologia móvel.

As Heurísticas SMARTs abrangem uma série de fatores importantes a criação de aplicações móveis. No entanto, apesar de considerarem as interfaces em seus

itens, não apresentam questões próprias à navegação por gestos encontradas nos dispositivos móveis.

2.2.2. ESCALA MATCH

MATCH - *Measuring Usability of Touch screen Phone Applications* (SALAZAR, *et al.*, 2013), é uma escala de usabilidade para medir a usabilidade de aplicativos de telefone com tela sensível ao toque por meio de uma única pontuação. Foi elaborada a partir de uma revisão sistemática da literatura e da adaptação das heurísticas de Nielsen para dispositivos sensíveis ao toque, que foram decompostos em itens de medição que representam a usabilidade do construto de qualidade abstrata (UFSC, 2022).

Foi desenvolvida pelo Grupo de Qualidade de Software do Instituto Nacional para Convergência Digital, da Universidade Federal de Santa Catarina, procura medir a usabilidade dos aplicativos *touch screen* através de uma única pontuação, a partir de um questionário (lista de verificação denotada) desenvolvido pelo grupo.

A lista de verificação foi validada por meio de um estudo empírico no qual os resultados de 247 avaliações heurísticas utilizando a lista de verificação foram analisados estatisticamente utilizando a Teoria de Resposta ao Item (TR). (UFSC, 2022)

O questionário possui 48 perguntas de multipla escolha com as opções: sim, parcialmente, não e não se aplica. Após o preenchimento, a pontuação é calculada automaticamente e exposta juntamente com um quadro com as explicações de cada faixa de classificação. A seguir são listadas as perguntas que compõe o questionário.

Heurística 1: Visibilidade do status do sistema

1. Para cada ação do usuário o aplicativo oferece *feedback* imediato e adequado sobre seu status? Por exemplo, após tarefas como envio de e-mail, adição, exclusão e carregamento de arquivo, exibir uma mensagem de confirmação do tipo "e-mail enviado" ou "arquivo excluído".
2. Os componentes interativos selecionados são claramente distintos dos demais? Por exemplo, o estado de botões muda quando são pressionados e destaca a aba do menu que está sendo visualizada.

3. As mensagens sobre o status do aplicativo possuem uma linguagem clara e concisa? Por exemplo, os títulos das telas e das mensagens de erro são de fácil compreensão.

4. Fornece um update do status para operações mais lentas? Por exemplo, uma indicação seja na forma de ícone ou texto sobre o progresso do carregamento do sistema ou de um arquivo.

Heurística 2: Correspondência entre o sistema e o mundo real

5. O significado de símbolos e ícones são compreensíveis e intuitivos? Utilizar ícones e símbolos fáceis de reconhecer e relacionar com a tarefa à qual estão associados.

6. As informações são dispostas em uma ordem lógica e natural? Por exemplo, itens em listas de seleção (nomes, produtos, etc.) são ordenados por um critério adequado (p.ex. alfabeticamente).

Heurística 3: Controle e liberdade do usuário

7. É o usuário quem inicia e encerra tarefas e não o aplicativo? Por exemplo, aguardar o usuário teclar **enter** após preencher o campo de busca para iniciar a tarefa.

8. É possível identificar o número de passos necessários para a realização de uma tarefa? Por exemplo, a partir de uma indicação numérica (1-5) da quantidade de páginas ou passos, da apresentação de um tutorial ou da divisão da tarefa em abas.

9. É possível retornar a tela anterior a qualquer momento? Seja a partir da navegação por abas, de um botão voltar do aplicativo ou do próprio celular.

10. No caso de aplicativos associados a login ou contas de e-mail, permite o fácil acesso de mais de um usuário? Por exemplo, um aplicativo de comércio eletrônico permitir a fácil escolha de qual conta utilizar para realizar a compra.

11. O usuário pode cancelar uma ação em progresso? Por exemplo, cancelar um download em andamento.

12. O aplicativo deixa claro qual o próximo passo para realizar a tarefa? Como a partir de um botão para avançar ou nota de explicação.

Heurística 4: Consistência e padrões

13. As telas com o mesmo tipo de conteúdo possuem o mesmo título? Por exemplo, todas as telas de busca possuem o mesmo título.

14. Controles e botões se distinguem do restante do layout, deixando evidente que são clicáveis? Por exemplo, diferenciar os botões aplicando sombra ou outro recurso para simular relevo.

15. Todas as informações textuais do aplicativo utilizam o mesmo idioma?

16. Funções diferentes são apresentadas de maneira distinta ao usuário? Por exemplo, funções diferentes como salvar e cancelar não são representadas pelo mesmo nome ou ícone.

17. Funções semelhantes são apresentadas de forma similar? Por exemplo, usa o mesmo ícone ou rótulo de botão para a mesma funcionalidade em telas diferentes ou propõe a mesma forma de entrada de dados para uma mesma funcionalidade em diferentes telas.

18. Controles que realizam a mesma função ficam em posições semelhantes na tela? Por exemplo, se em uma tela o botão para avançar fica no lado direito, nas outras telas esse mesmo botão também estará no lado direito.

19. A forma de navegação é consistente entre as telas no aplicativo? Mantêm o mesmo tipo de navegação (rolagem vertical, rolagem horizontal, menus ou abas) em todas as telas.

20. Os links são tratados de forma consistente entre as telas? Mantêm o mesmo tratamento visual em termos de cor, tipo e estilo (p.ex. negrito, sublinhado) de fonte.

21. As informações textuais são apresentadas de forma padronizada? Apresenta informações textuais semelhantes na mesma disposição e com o mesmo tratamento visual (tamanho, tipo e cor da fonte).

22. Os dados e mensagens mais importantes encontram-se na posição padrão dos aplicativos para esta plataforma? Sim, o menu de ferramentas foi reposicionado para evitar que a palma da mão escondesse as informações. As dicas das ferramentas foram posicionadas logo acima dele para permitir uma melhor visualização.

23. Em campos onde existe a necessidade de inserção de dados isso é evidente? Por exemplo, ter uma caixa de texto com cursor.

Heurística 5: Reconhecimento em vez de lembrança

24. O aplicativo utiliza em seus textos e rótulos, uma linguagem habitual e conhecida pelo usuário do aplicativo? Evitando termos técnicos ou muito específicos de determinada área.

25. Os títulos das telas descrevem adequadamente seu conteúdo? Não se aplica, o HyperCAL ^{3D} *mobile* não apresenta mais de uma tela.

Heurística 6: Flexibilidade e eficiência de uso

26. O aplicativo funciona corretamente, sem apresentar problemas durante a interação? Por exemplo, não trava e botões funcionam no primeiro clique.

27. As tarefas são relativamente simples de serem executadas? Por exemplo, uma tarefa pode ser completa em poucos passos.

28. As funções mais utilizadas são facilmente acessadas? As funções mais utilizadas devem ser acessadas sem precisar rolar ou navegar entre muitas telas.

29. O aplicativo utiliza objetos (ícones) em vez de botões? Por exemplo, utilizar um ícone de impressora em vez de utilizar a palavra impressora.

30. Todas as telas mantêm acessíveis menus e funções comuns do aplicativo? Por exemplo, em aplicativos de conta de e-mail a caixa de entrada é acessível a partir de todas as telas do aplicativo.

Heurística 7: Estética e design minimalista

31. São exibidas apenas informações relacionadas a tarefa que está sendo realizada? Por exemplo, na tela de cadastro, outras informações não devem ser exibidas.

32. São usados textos somente quando estes são realmente indispensáveis? Por exemplo, não oferecer instruções textuais muito longas.

33. O menu é esteticamente simples e claro? Com opções fáceis de encontrar, dispostas em uma ordem lógica e com títulos curtos.

34. O aplicativo exibe quantidades pequenas de informações em cada tela? Sem texto ou imagens em excesso.

35. Os títulos de telas/janelas e rótulos de botões/links são curtos? Sim, os únicos rótulos existentes estão nas abas de criação de botões e faces e possuem apenas uma palavra.

36. Em textos, o uso de abreviaturas é evitado? Com opções fáceis de encontrar, dispostas em uma ordem lógica e com títulos curtos.

Heurística 8: Pouca interação homem/dispositivo

37. A navegação do aplicativo é intuitiva? Por exemplo, é fácil chegar à tela desejada.

Heurística 9: Interação física e ergonomia

38. Possui botões com tamanho adequado ao clique? Por exemplo, evitando botões muito pequenos causando a seleção da opção errada.

39. A navegação principal encontra-se na posição padrão dos aplicativos para esta plataforma? Por exemplo, o menu na barra inferior para o iOS e superior para o Android.

40. Os botões e controles podem ser facilmente acessados com qualquer uma das mãos? Especialmente no caso de botões que serão utilizados repetidamente para avançar ou confirmar ações.

41. A área clicável dos botões e links ocupa toda a dimensão dos mesmos?

Heurística 10: Legibilidade e layout

42. O espaçamento entre linhas utilizado favorece a leitura? Nem muito grande, para não aumentar desnecessariamente a rolagem, e nem muito pequeno dificultando a leitura.

43. As fontes utilizadas favorecem a leitura? Em termo de tamanho, tipo e estilo.

44. Os ícones possuem contraste suficiente em relação ao plano de fundo?

45. Os textos têm contraste suficiente em relação ao plano de fundo? Por exemplo, evitando texto cinza claro em um fundo branco.

46. As imagens possuem cor e detalhamento favoráveis a leitura em uma tela pequena? A resolução deve permitir a fácil identificação dos elementos da imagem e os ícones não devem ter muitos detalhes usando uma representação mais abstrata.

47. O aplicativo realça conteúdos mais importantes, deixando-os maiores, mais brilhosos ou em negrito?

48. O alinhamento utilizado favorece a leitura? Por exemplo, dando preferência para alinhamento justificado ou esquerdo para texto corrido.

2.3. INTERFACES NATURAIS

Há algum tempo, as telas de celulares ganharam o espaço que anteriormente era dedicado as teclas. Telas maiores e sensíveis ao toque passaram a ocupar toda a frente do dispositivo e os cliques deixaram de ser a única forma de interação. Diferentes dispositivos passaram a operar com controles gestuais que simulam os movimentos do corpo humano, de jogos de vídeo game que capturam o corpo inteiro

do jogador até simples torneiras de banheiro com sensores de movimento, as interfaces naturais (*Natural User Interface* - NUI) passaram a fazer parte da vida cotidiana. Liu (2010) avalia que esta seja a quarta geração de interfaces desenvolvidas até o momento. Ele classifica as interfaces como:

- Interface em lote – para o autor foram as primeiras interfaces idealizadas que funcionavam a partir de cartões perfurados nas primeiras máquinas existentes.
- Interface em linhas de comando – utilizavam comandos para a realização de tarefas.
- Interfaces gráficas (*Graphic User Interface* GUI) – são utilizadas até hoje em programas diversos, carregam metáforas como *desktop* e na forma de ícones que representam comandos.
- Interfaces Naturais - se concentram em habilidades humanas como toque, visão, voz, movimento e funções cognitivas mais elevadas, como expressão e percepção.

Segundo Wigdor e Wixon (2011) a cada nova geração de interfaces, surgem também novos desafios na construção das aplicações. No entanto, há uma tendência dos designers de tentar recriar conceitos da anterior na nova. “Um risco na construção desses novos aplicativos é que os designers e desenvolvedores dependem de conceitos e abordagens que funcionaram no passado” (WIGDOR e WIXON, 2011).

As interfaces naturais são consideradas interfaces invisíveis, pois o usuário ao imergir na interação, esquece que está lidando com um dispositivo eletrônico. Isso ocorre, pois estes são desenvolvidos de forma a se adaptar as interações humanas em suas formas naturais. A seguir são apresentadas as características das NUIs elencadas por Liu (2010).

Centrado no usuário: diferente de outros tipos de interface, nas quais as pessoas precisam se adaptar as máquinas, as NUI tendem a buscar que as máquinas se adaptem ao usuário, como por exemplo, a partir de reconhecimento de fala e por gestos que traduzem intenções de uso, como mudar de página ou excluir mensagens (LIU, 2010).

Multicanal: as NUI buscam melhorar a naturalidade da interação humano-computador a partir de canais sensoriais como visão, audição, tato, olfato e equilíbrio, e canais motores tipo mãos, boca, olhos, cabeça, pés e corpo (LIU, 2010).

Imprecisas: diferente de teclados que apresentam inserções precisas de teclas, as interfaces naturais lidam com *inputs* imprecisos que precisam ser interpretados pelo dispositivo como fala e gestos (LIU, 2010).

Alta largura de banda: a velocidade dos *inputs* também é um fator determinante nas NUI, mesmo que o usuário possa digitar rapidamente, ainda não será tão rápido quanto a fala e/ou os gestos (LIU, 2010).

Interação baseada em voz: mais ou menos 75% da comunicação humana é feita por voz, no entanto para que este tipo de interação seja realizado com dispositivos eletrônicos, são necessários o reconhecimento de voz e sintetização de voz. Desta forma, a interação pode ocorrer de forma natural (LIU, 2010).

Interação com bases de imagem: suas principais utilizações são o reconhecimento facial e de escrita. É baseada em três pilares básicos, processamento de imagens que permite a melhora nos efeitos visuais das imagens; reconhecimento de imagem, procura reconhecer padrões e medir objetos para obter informações e transformar a imagem em dados; percepção da imagem, visa a interpretação do significado da imagem (LIU, 2010).

Interação baseada em comportamento: a linguagem corporal pode expressar atitude e significado que não são expressas pela fala. Essa interação é baseada no posicionamento, rastreamento, movimento e características de expressão das partes do corpo humano para entender a ação e o comportamento (LIU, 2010).

Como descrito, as NUIs apresentam diferentes características destinadas a distintas formas de interação. No entanto, este estudo foca na interação por gestos devido à natureza dos dispositivos móveis e suas *touch screen*. Portanto, a seguir são apresentados conceitos de designing de interfaces gestuais, mesmo que em alguns momentos elas sejam referenciadas apenas como NUIs.

Design de Interfaces Gestuais

As interfaces gestuais podem ser *touch screen*, em outras palavras, que exigem que usuário toque diretamente no dispositivo, ou *free-form* que não necessitam de toque. Apesar da diferença, ambas funcionam da mesma forma, a partir de três partes gerais: um sensor que é um componente elétrico ou eletrônico que detecta alterações no ambiente; um comparador que compara o estado atual com o estado anterior; e um atuador que, caso uma determinada condição seja atendida, executa a ação determinada (SAFFER, 2008).

Quando comparadas as interfaces tradicionais, baseadas em mouse e teclado, as interfaces gestuais permitem uma gama maior de ações para manipular o sistema como: rolar, girar e aproximar, além das formas tradicionais teclar e clicar. No entanto, alguns elementos que auxiliam na compreensão das interfaces tradicionais, devido à natureza direta da interação, não fazem sentido nas interfaces gestuais, como: cursores, *mouse-over*, duplo clique e clique com o botão direito. Além desses, outros recursos precisaram ser adaptados ao *touch*, como: menus suspensos, copiar, colar e seleção de vários itens.

A diversidade de combinações de gestos permite uma gama variada de ações. Saffar (2008) analisa, que a complexidade do gesto deve ser diretamente proporcional a tarefa que ele executa. Isso significa, quanto menor a complexidade da função, mais simples deve ser o gesto que a executa. Atividades mais complexas podem ter gestuais mais elaborados. Porém, o autor avalia que mesmo uma atividade complexa pode apresentar um gestual simplificado.

O tamanho da tela é outro fator a ser considerado em dispositivos *touch screen*, pois determina se os gestos podem ser pequenos ou largos, com uma ou duas mãos. Além disso, a palma da mão pode acabar por esconder recursos. “Colocar menus e controles na parte inferior da tela em vez de em seu lugar tradicional na parte superior é útil para evitar a cobertura da tela” (SAFFER, 2008).

Como forma de auxiliar no processo de design Wigdor e Wixon (2011), traçam oito diretrizes para o projeto de interfaces gestuais, que são descritas a seguir.

1. Assim como qualquer novo gênero de interface, as diretrizes de design da NUI são próprias e não devem ser adaptações de outros gêneros. Uma interface NUI não deve ser uma GUI com toque, pois é provável que a equipe não considere os mais novos aspectos do meio. Os autores aconselham a investigação de aplicações *touch screen* similares para o entendimento das possibilidades interação com as interfaces NUI
2. Uma interface natural não é aquela que tenta adivinhar qualquer ação do usuário, e sim promove o desenvolvimento de comportamentos qualificados e gera um sentimento de domínio. Designer de interfaces NUI devem esquecer tudo que sabem sobre interfaces gráficas e iniciar o desenvolvimento como uma folha em branco. Em seguida devem mapear ações provocadas pelo ambiente do app, considerar seu conteúdo e considerar como novas ações podem ser realizadas.
3. Considerar o eixo z para os comportamentos e movimentos dos objetos na tela, neste caso os autores tratam de simulações do espaço 3D e não de ambientes tridimensionais, como por exemplo: como inclinar planos, adicionar sombras e elementos sobrepostos para fazer alguns objetos parecerem como se tivessem profundidade.
4. Analisar a possibilidade do sistema de aceitar o desenvolvimento de tarefas realizadas em conjunto com vários usuários, em duplas e individualmente. Sempre que possível permitir a interação multiusuários, com a possibilidade de trabalho concomitante em atividades distintas e com diferentes graus de dificuldade.
5. Permitir a imersão do usuário a partir da suspensão da descrença, na qual ele esteja disposto a aceitar como verdadeiro ou suficientemente real, algo fantasioso ou impossível no mundo real. Assim, ele deixa de comparar suas ações com um padrão definido e passa a experimentar uma conexão direta entre suas ações e os objetos e operações do sistema. Para tanto, é necessário incentivar a ação fluida de objetos do mundo real, usando recursos do mundo virtual para estender os objetos para além do que é possível no mundo real.

6. Criar experiências fluidas e naturais ao aumentar interações físicas do mundo real, como por exemplo: contrair e expandir uma imagem com dois dedos na tela como se ela fosse de borracha.

Para criar interações naturais, crie a base das interações no mundo real e, em seguida, amplie-as de maneiras intuitivas. Para criar interações super-reais, aproveite as possibilidades de objetos virtuais em ambientes digitais para superar o que é possível no mundo real (WIGDOR e WIXON, 2011).

7. Uma interface NUI promove a aprendizagem autônoma, empregando ações que incentivam os usuários a desenvolver suas próprias habilidades cognitivas, afetivas e psicomotoras. Permite que um usuário iniciante passe a experiente em pouco tempo. Para tanto, é necessário compreender as possibilidades do sistema e dividi-las em desafios menores, como iniciar uma ação ou o que fazer em um segundo momento.
8. Cada interação pode variar de acordo com o usuário que a utiliza e o contexto no qual ele está inserido, por exemplo: um mesmo aplicativo pode ser utilizado por um gerente e um vendedor, e a interface deve ser adaptada para ambos os contextos.

No próximo item são debatidas questões referentes as metodologias de desenvolvimento de aplicações móveis, para tanto um breve histórico do assunto é apresentado de forma a elucidar os métodos existentes na atualidade que são descritos na sequência.

2.4. PROCESSO DE DESENVOLVIMENTO DE SOFTWARES

Se comparado ao processo de desenvolvimento de *software*, a criação de aplicativos móveis pode ser considerada relativamente recente e apesar de existirem algumas diferenças provenientes de características próprias dos dispositivos, seus desenvolvimentos compartilham diversas semelhanças.

Assim, neste item é considerada brevemente a evolução da engenharia de *software*, e as diferentes metodologias encontradas para o seu desenvolvimento.

Engenharia de *software*

O termo Engenharia de *software* foi cunhado em uma conferência da OTAM - Organização do Tratado do Atlântico Norte⁶ em outubro de 1968, a ideia central surgiu da necessidade de voltar o foco para o desenvolvimento de *softwares* de forma sistemática, disciplinada e quantificável, o que permitiria aos desenvolvedores operar, manter e evoluir os sistemas. Algumas décadas antes, os cientistas da computação não apresentavam estas preocupações, pois o foco era o desenvolvimento de máquinas para rodar poucos programas. (VALENTE, 2022).

Com o avanço da tecnologia, os computadores deixaram de ser máquinas enormes com poucas funções e passaram a necessitar de uma gama variada de programas para realizar diferentes atribuições. Este fato, deixou evidente a necessidade de organização dos processos de desenvolvimento de *software*. Inicialmente, estes processos eram ordenados de forma sequencial, iniciavam com a especificação de requisitos e terminavam com sua implementação, aplicação de testes e manutenção do sistema. Esta prática, chamada de *Waterfall*, tornou-se obsoleta devido a dificuldades de manutenção de cronogramas e orçamentos (VALENTE, 2022).

Em resposta ao baixo rendimento dos processos *waterfall* (ou processos em cascata), surgiram as metodologias Ágeis. Elas dividem *softwares* complexos em partes menores, permitindo o desenvolvimento de forma incremental. Isso significa que partes do programa são idealizadas, desenvolvidas, testadas e implementadas em curtos espaços de tempo. Os incrementos permitem que adaptações a possíveis mudanças possam ser inseridas à medida que o sistema é desenvolvido (SANTOS, 2016). As Metodologias Ágeis passaram a ter notoriedade em 2001, quando um grupo de dezessete especialistas em processos de desenvolvimento de *software* se reuniu em Utah, e idealizou o "Manifesto Ágil" (BECK, *et al.*, 2001), que apresenta os seguintes valores:

- **Indivíduos e interações** valem mais que processos e ferramentas;
- **Software em funcionamento** mais que documentação abrangente;

⁶ Surgiu como uma organização político-militar de países europeus e americanos com o objetivo formar uma união militar entre nações alinhadas.

- **Colaboração do cliente** mais que negociação de contratos;
- **Responder a mudanças** mais que seguir um plano.

Além destes valores, os especialistas deixam claro que mesmo valorizando os itens a direita, prezam mais pelos elementos da esquerda, destacados em negrito neste texto.

Existem diferentes Metodologias Ágeis utilizadas no desenvolvimento de sistemas. No entanto, no desenvolvimento de *softwares*, há uma inclinação para a utilização do *Framework Scrum* (KALEEL e HARISHANKAR, 2013) (RAHIMIAN e RAMSIN, 2008) (MARTINEZ, *et al.*, 2020), e da metodologia *Extreme programming* (XP) (ABRAHAMSSON, *et al.*, 2004) (JEONG, LEE e SHIN, 2008) em suas formas originais ou como base para criação de novas metodologias. Portanto, no próximo item é apresentada o *Framework Scrum*, seguida da Metodologia XP.

2.4.1. Framework SCRUM

De acordo com os seus criadores Schwaber e Beedle (2001) no livro “*Agile Software Development with Scrum*” o *Framework Scrum* é baseada no modelo de controle de processo empírico que introduz a flexibilidade, a adaptabilidade e a produtividade no desenvolvimento de sistemas. Em outras palavras, Scrum não espera que um processo inicie e termine sem falhas para que um novo processo se inicie, mas aceita e se adapta à medida que elas aparecem.

O *Framework* surgiu como forma de organizar o trabalho e as equipes de desenvolvimento de sistemas, pois os engenheiros de *software* eram interrompidos em suas atividades com mudanças de escopo que atrasavam a produção e até mesmo inviabilizavam as mesmas. Scrum cria uma figura chave (*Scrum Master*) que prioriza e organiza os processos. Certifica-se que os membros da equipe não se comprometam com compromissos além dos que eles conseguem cumprir dentro de uma *Sprint*.

No entanto, a figura do intermediador não é capaz de evitar todos os problemas inerentes a criação de *softwares*. O *Framework Scrum* avalia que as constantes melhorias nas tecnologias, desenvolvedores inexperientes ou com pouco

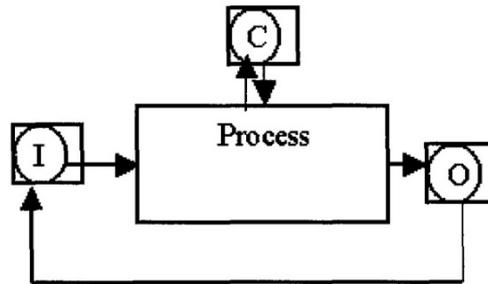
conhecimento da tecnologia utilizada e requisitos com especificações fracas, podem produzir ruídos nos processos. Se os programadores sabem como construir os requisitos usando a tecnologia selecionada, há pouco ruído ou imprevisibilidade e o trabalho flui com pouco ou nenhum retrabalho ou erro. No entanto, se um destes requisitos não for cumprido, ruídos são adicionados a produção (SCHWABER e BEEDLE, 2001).

Existem duas abordagens para lidar com os diferentes ruídos encontrados em sistemas: modelo de controle de processo definido e modelo de controle empírico. O primeiro tende a ser usado para a resolução de ruídos discretos, nos quais a repetição do processo gera sempre os mesmos resultados, em geral servem para controlar sistemas simples. Segundo Schwaber e Beedle (2001) quase nenhum projeto de sistema é tão simples.

O segundo modelo é utilizado para gerenciar e controlar processos complexos, que geram muitos ruídos. Isso significa, que os processos mesmo quando repetidos tendem a gerar resultados diferentes e até mesmo inesperados. Portanto, a gestão e o controle são feitos através de inspeções frequentes e respostas adaptativas. Desta forma, o ruído e a complexidade de cada etapa são reduzidos, o que aumenta as chances de sucesso de cada processo (SCHWABER e BEEDLE, 2001). A Figura 16 representa o modelo de gerenciamento empírico de processos, no qual:

- "I" é a equipe que construirá o incremento a partir dos requisitos e tecnologia.
- "Processo" é a produção que tem um prazo determinado de trinta dias, também chamado de *Sprint*.
- "C" é controle e monitoração do processo de Scrum, realizado a partir de reuniões diárias chamadas de *Daily Scrum* e reunião ao final de cada *Sprint*, *Sprint Review*.
- "O" é o produto incremental construído durante cada iteração.

Figura 16 - Modelo de gerenciamento empírico



Fonte: Schwaber e Beedle (2001)

Como os autores definem, Scrum se assemelha a criação de novos produtos e não a fabricação deles, como era considerada anteriormente a criação de *softwares*. Eles defendem que mesmo que os componentes de uma criação de sistemas sejam os mesmos, seu arranjo e parametrização são únicos para cada projeto. Devido a esta característica, as reuniões com a equipe apresentam um caráter fundamental no desenvolvimento, pois a troca de conhecimento entre os participantes enriquecem e direcionam o processo. Assim, é possível reduzir o ruído inerente a processos complexos, pois o compartilhamento de conhecimento qualifica a equipe.

Até este ponto, foram apresentados os fundamentos que levaram a criação do *Framework Scrum*, a seguir são apresentados de forma esquemática: os membros da equipe e as listas de demandas. Ao final, é apresentado o ciclo de processo Scrum.

A equipe é dividida em três papéis básicos:

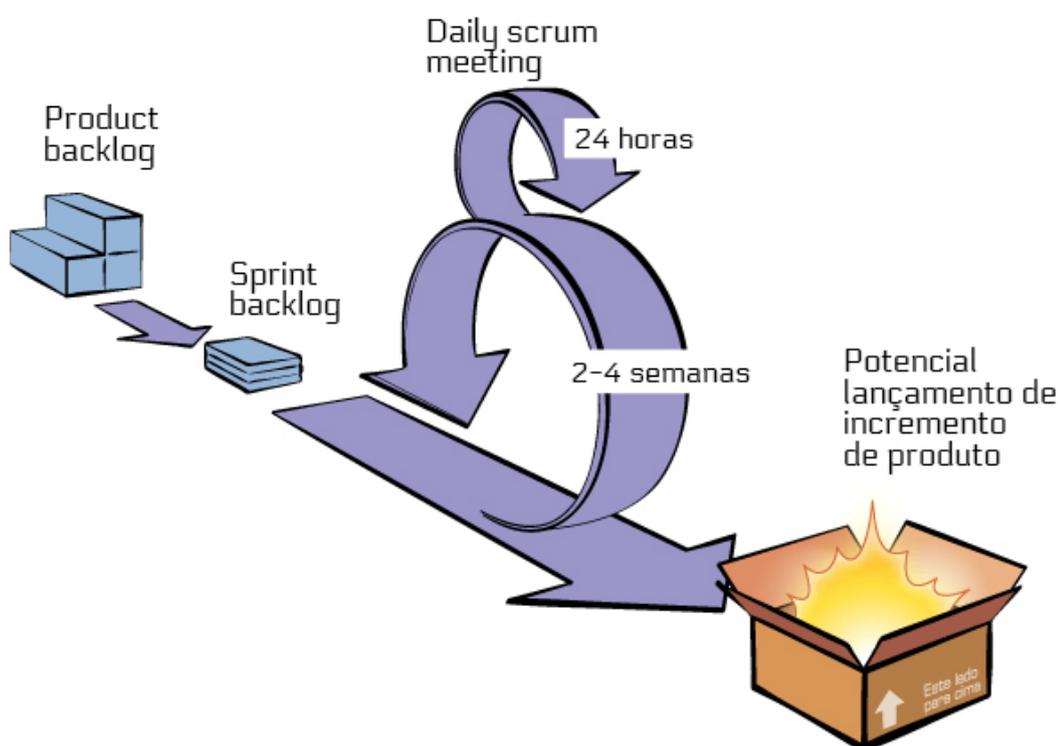
- *Scrum Master* é o líder ou chefe da equipe, intermedia a relação com os demais setores da empresa a fim de impedir que o time de desenvolvimento (*Scrum Team*) perca o foco no *Scrum Sprint*.
- *Scrum Team* é o time de desenvolvimento responsável pela implementação dos requisitos determinados para o *Scrum Sprint*.
- *Product Owner* é o representante do cliente que apresenta as demandas do contratante do sistema.

As demandas do sistema são divididas em listas de processos que precisam ser executados ou finalizados chamados de *Backlog*. No *Framework Scrum*

inicialmente é definido o *Product Backlog*, que sistematiza as demandas necessárias à finalização do projeto. Além deste, cada *Sprint* possui uma lista menor de demandas, derivada do *Product Backlog*, chamada *Sprint Backlog*.

A Figura 17 demonstra um ciclo (*Sprint*) do *Framework Scrum*. A partir do *Product Backlog* são selecionados os itens que serão trabalhados no *Sprint* que pode durar de duas a quatro semanas, com reuniões diárias de quinze minutos, nas quais os desenvolvedores reportam o que fizeram, no que estão trabalhando e quais os próximos passos. Desta forma, todos os envolvidos sabem em que ponto o desenvolvimento está. Ao final do *Sprint* é realizada a *Sprint Review* e um novo *Sprint* se inicia.

Figura 17 – Ciclo do *Framework Scrum*



Fonte: Adaptado de <http://www.desenvolvimentoagil.com.br/scrum/>

2.4.2. Metodologia Extreme Programming (XP)

Assim como o Scrum, a Metodologia XP (BACK, 1999) surgiu para ser aplicada em curtos espaços de tempo e mitigar problemas ao se adaptar a requisitos em evolução e mudança. Ela foi criada sobre valores, princípios e práticas, para a utilização em equipes pequenas e médias. Seus valores correspondem a:

Comunicação: falhas na comunicação da equipe podem gerar atrasos e retrabalho. Já a troca de informações permite a distribuição do conhecimento, o que qualifica os desenvolvedores e acelera a produção.

Simplicidade: é melhor fazer o simples e funcional, que atenda às necessidades e pagar um pouco mais amanhã para mudá-lo se precisar, do que gastar tempo em algo complexo que pode não ser usado.

Feedback: a XP valoriza *feedbacks* rápidos e constantes, pois é capaz de se adaptar as mudanças, portanto se for preciso mudar a direção de um sistema é melhor que seja rápido.

Coragem: verdades, especialmente as desagradáveis, são difíceis, mas permitem que os processos fluam e se adaptem rapidamente em caso de necessidade.

Respeito: todos são dignos e merecem ser respeitados independente de experiência ou técnica.

Além dos valores, a Metodologia XP trabalha em cima de 12 princípios:

- O Jogo de Planejamento — combinar as prioridades e estimativas técnicas para esboçar o escopo da próxima implementação, mantendo o plano sempre atualizado.
- Pequenas versões — programar um sistema simples rapidamente, para a seguir liberar novas versões em ciclos curtos.
- Metáfora — traduzir as palavras do cliente em uma história comum para Equipe de Desenvolvimento e o próprio cliente.
- Design simples — remover a complexidade em todos os pontos do sistema.
- Teste — devem ser realizados testes constantemente para que o desenvolvimento do sistema continue.
- Refatoração – remove a duplicidade de código, melhora a comunicação, simplifica e adiciona flexibilidade.

- Programação de pares — todo o código produzido é implementado por duas pessoas juntas.
- Propriedade coletiva — o código pertence a todos os membros da equipe, desta forma, todos têm acesso e autorização para modificar o código
- Integração contínua — Integre e construa o sistema muitas vezes por dia, toda vez que uma tarefa é concluída.
- 40 horas semanais — respeitar o tempo de trabalho de 40 horas por semana, evitando horas extras pela segunda semana seguida.
- Cliente presente — incluir um cliente real e ao vivo, permite que as dúvidas sejam respondidas em tempo integral.
- Padrões de codificação — padrões de código permitem uma melhor comunicação entre a equipe e que todos possam fazer ajustes no futuro.

Quando colocadas lado a lado, o *Framework* Scrum e a Metodologia XP é possível perceber que mesmo destinadas ao desenvolvimento de *softwares* possuem enfoques diferentes. Scrum se atenta mais na organização do trabalho como um todo, sem ditar como os desenvolvedores devem trabalhar, enquanto a XP procura identificar boas práticas de programação. Assim, as duas metodologias podem ser utilizadas em conjunto.

Apesar do *Framework* Scrum e da Metodologia XP, também serem utilizados para o desenvolvimento de aplicações móveis, peculiaridades dos dispositivos como tamanho de tela, capacidade de processamento, diferenças de *hardware* e interação via *touch screen* representam obstáculos à parte, além da interface gráfica que deve ser enfatizada para recursos de exibição rápidos. O autor Ferdiana (2012), acrescenta a estas os tipos de aplicação móvel que podem ser, segundo ele:

- aplicativo web móvel, utilizam os navegadores dos dispositivos móveis e não necessitam de instalação.

- aplicativo móvel nativo, geralmente desenvolvidos para um tipo específico de dispositivo, são vendidos em lojas virtuais e necessitam de instalação.
- aplicativo móvel híbrido, combinam elementos nativos com web móvel. Possuem integração com sistema de arquivos de um dispositivo, ou sistema baseado na Web, e navegador incorporado.

Para Ferdiana (2012), a escolha do tipo de aplicação impacta diretamente na tecnologia que será utilizada, no grupo de desenvolvedores e no tempo para seu desenvolvimento.

Apesar de existirem metodologias próprias para o desenvolvimento móvel, em geral, utilizam como base o *Framework* Scrum e/ou a Metodologia Xp e adaptam seus métodos para as necessidades oriundas das aplicações a serem desenvolvidas. Desta forma, confirmam a teoria de Schwaber e Beedle (2001) que o desenvolvimento de aplicações móveis é um processo de criação complexo e não pode ser engessado em uma linha de produção.

Outro fator importante no processo de desenvolvimento é a avaliação. No próximo item são debatidas questões referentes aos testes e avaliação de *softwares* e aplicações móveis.

2.4.3. Testes

“Se uma equipe acredita que é ágil, mas nada mudou na maneira como testam, então ainda há muito para aprender”⁷ (LAING e GREAVES, 2015). Como forma de explicar esta frase, Laing e Greaves (2015) utilizam cinco princípios que diferenciam os testes tradicionais dos testes ágeis

Testar é uma atividade, não uma fase

Em métodos tradicionais a avaliação do *software* é realizada ao final do processo, porém as metodologias ágeis dividem a implementação em etapas menores que podem ser testadas individualmente, sem a necessidade de aguardar a finalização do projeto.

⁷ Tradução livre de “If a team believes they are agile, but nothing has changed about the way they test, then there is still much to learn”.

Evite bugs em vez de encontrar bugs

O pensamento tradicional entende os testes como uma forma de encontrar *bugs* (erros), algumas empresas medem a produtividade dos testadores pelo número de erros encontrados. No entanto, estes ocorrem primordialmente devido suposições sobre requisitos, que são implementadas sem esclarecimento. Portanto, para evitar os *bugs* é necessária uma compreensão clara dos requisitos para a implementação.

Não seja um verificador, seja um testador.

Para testar os requisitos é necessário compreender profundamente quem são seus usuários e o que se procura alcançar com o produto, garantindo que o recurso atenda às necessidades reais dos clientes e não apenas a uma lista de critérios.

Não tente quebrar o sistema, em vez disso, ajude a construir o melhor sistema possível

Em sistemas tradicionais de testagem, buscam-se falhas no sistema. No entanto, a busca deveria se concentrar no auxílio a construção dele. Isso significa que o teste deve ser feito de forma a encontrar requisitos que atendam às necessidades dos usuários. A melhor maneira de fazer isso é descobrir como testar o sistema do ponto de vista do usuário e, em seguida, compartilhá-lo com os desenvolvedores antes que eles comecem a codificar.

Toda a equipe é responsável pela qualidade, não apenas pelo testador

Há uma confusão na forma tradicional de testagem, pois o testador, ou a equipe de teste, parece ser a responsável pela qualidade do *software*. No entanto, em ágil, toda a equipe é responsável pela qualidade. Isso ajuda as equipes a perceber que o teste é uma atividade que todos precisam participar e que isso acontece durante todo o trabalho.

A partir destes pontos, é possível compreender a avaliação ou os testes como parte fundamental do processo de construção da aplicação ou *software* e não como uma etapa desconectada que visa avaliar a qualidade final. Desta forma, os testes são realizados de forma a compreender as necessidades do usuário e não para encontrar erros no sistema. Por esse motivo, surgiram alguns métodos de

desenvolvimento baseados em testes como *Test Driven Development* (TDD), que visa a criação de testes anteriormente a geração do código. A ideia por trás deste método, pode ser encontrada nas práticas construtivas da metodologia XP e por este motivo não são debatidas neste trabalho.

No próximo item é apresentado o software gráfico HyperCAL^{3D}, desde sua idealização até sua composição atual, com foco especial nas tecnologias utilizadas nos diferentes estágios do programa e como elas permitiram a evolução do projeto.

2.5. HYPERCAL^{3D}

O HyperCAL^{3D}, juntamente como HyperCAL GD e o HyperCAL *on-line*, fazem parte do conjunto de tecnologias desenvolvidas como apoio ao processo de ensino-aprendizagem desenvolvidos pelo grupo de pesquisa Virtual Design (VID) da Universidade Federal do Rio Grande do Sul. São frutos de uma série de estudos de metodologias de ensino realizados ao longo de vários anos de pesquisas dos professores que compõem o grupo, a fim de promover a manutenção e qualificação do ensino de GD na UFRGS (TEIXEIRA, *et al.*, 2007).

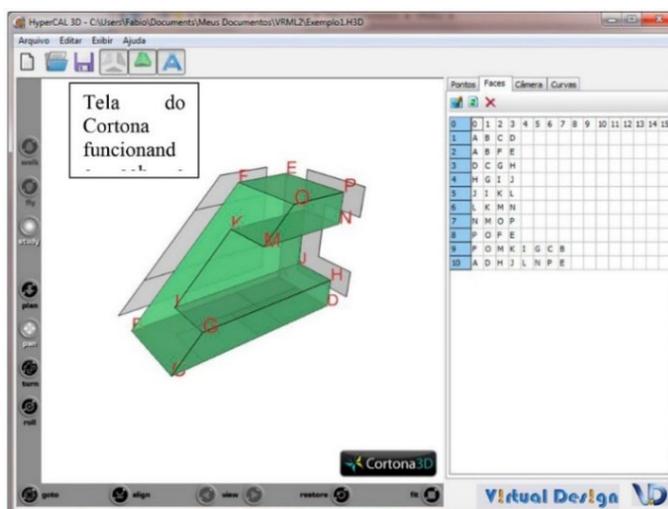
A partir de 2006, o grupo passou a utilizar a aprendizagem baseada em projetos nas aulas de GD, substituindo conceitos abstratos como adimensional, unidimensional e bidimensional que se referem ao ponto, linha e plano respectivamente, pelo estudo a partir de sólidos que contêm os mesmos elementos em sua construção. Para o aluno é mais fácil “entender o porquê de achar a verdadeira grandeza (VG) de uma face de um sólido do que a razão para achar a VG de um plano infinito ou de um triângulo solto no espaço” (TEIXEIRA, *et al.*, 2007).

Os principais objetivos são: reduzir o grau da abstração necessário para aprender os conceitos básicos; aproveitar os conhecimentos prévios dos estudantes; garantir o entendimento dos processos bidimensionais e os seus correspondentes tridimensionais e estimular o processo de aprendizagem a partir de problemas concretos com aplicações práticas (TEIXEIRA e SANTOS, 2013).

Como apoio a metodologia, no segundo semestre do ano de 2006, foi disponibilizada a primeira versão do aplicativo HyperCAL^{3D} desenvolvido na linguagem VRML (*Virtual Reality Modeling Language*) e que necessitava de *plug-in* Cortana da *Parallel Graphics* para a visualização das geometrias. A partir da avaliação

de alunos e professores, ficou claro que a baixa interatividade devido a tecnologia utilizada era um problema que precisava ser resolvido (TEIXEIRA, 2010). A Figura 18 apresenta a primeira versão do programa.

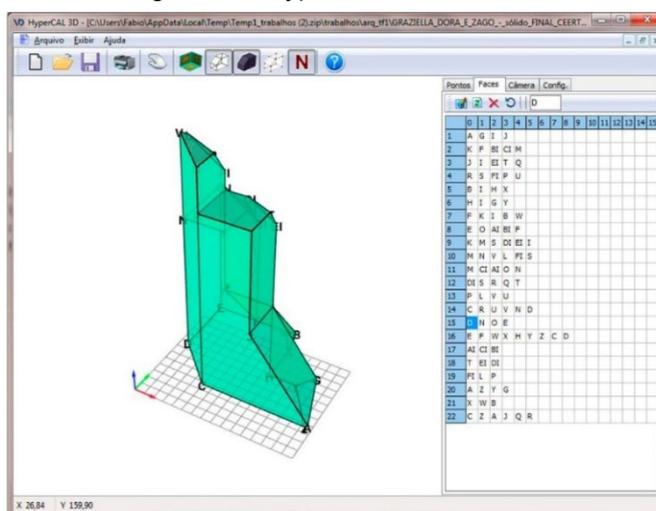
Figura 18 - HyperCAL ^{3D} versão 1.0



Fonte: (TEIXEIRA, 2010)

Como a tecnologia utilizada na primeira versão não comportava as necessidades interativas do software, a versão 2, Figura 19, foi totalmente reescrita utilizando o processador gráfico OpenGL, a linguagem Delphi e a biblioteca GLScene. Desta forma, além do programa se tornar mais interativo, permitiu a possibilidade de impressão dos modelos e escalabilidade do software (TEIXEIRA, 2010).

Figura 19 - HyperCAL ^{3D} versão 2.0



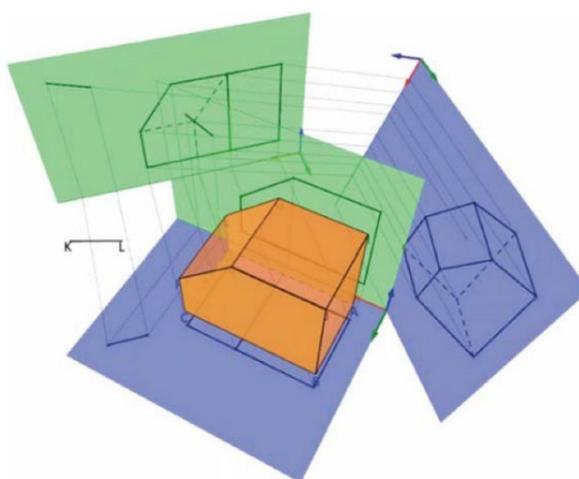
Fonte: (TEIXEIRA, 2010)

A mudança de tecnologia permitiu que a atualização do software pudesse ocorrer com maior frequência, “normalmente a cada semestre, uma nova versão é

distribuída, sempre trazendo melhorias e novas funcionalidades, com base no uso e experiências feitas por alunos e professores da UFRGS” (SANTOS, 2016).

Duas novas funcionalidades inseridas a partir da versão 3.0 do HyperCAL ^{3D}, representaram avanços significativos no programa, a possibilidade de criação de planos auxiliares de projeção a partir da Mudança de Sistema de Referência (MSR) permitiu que o sólido fosse projetado em qualquer orientação e a visualização diferenciada para linhas ocultas (SANTOS, 2016). Esta funcionalidade pode ser vista na Figura 20.

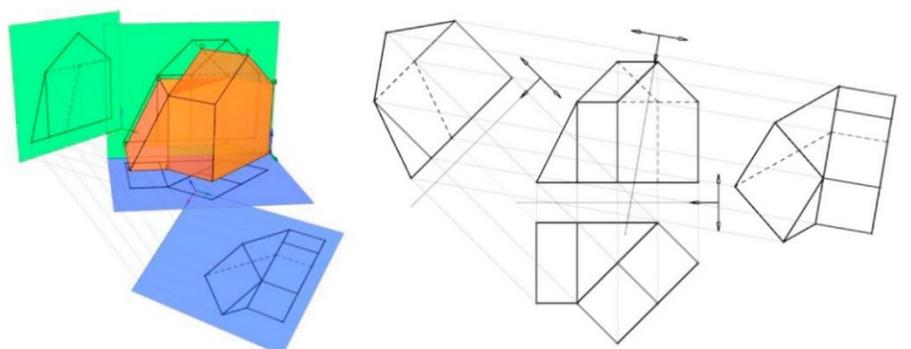
Figura 20 - Mudanças no HyperCAL ^{3D} versão 3.0



Fonte: (SANTOS, 2016)

A versão 4.0 agregou um ambiente bidimensional com o diedro planificado, o acréscimo da épura ⁸ ao programa representou a inserção de uma das funcionalidades fundamentais ao ensino de GD (TEIXEIRA e SANTOS, 2013), Figura 21.

Figura 21 - HyperCAL versão 4.0

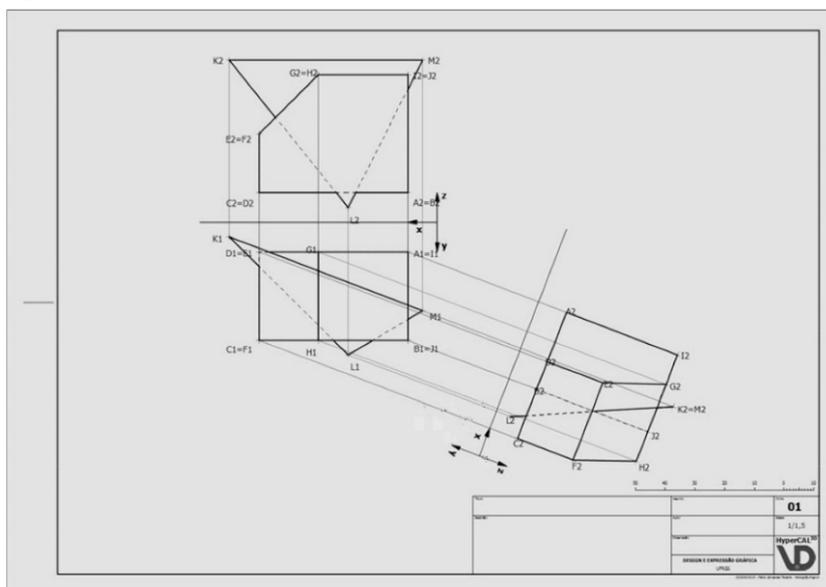


Fonte: adaptada de (TEIXEIRA e SANTOS, 2013).

⁸ Representação das projeções de sólidos geométricos nos planos rebatidos em duas dimensões.

Após a inserção da épora a possibilidade de impressão passou a ser melhor aproveitada para a criação de folhas padronizadas com imagens em alta resolução (SANTOS, 2016). A Figura 22 apresenta uma folha padronizada para a impressão.

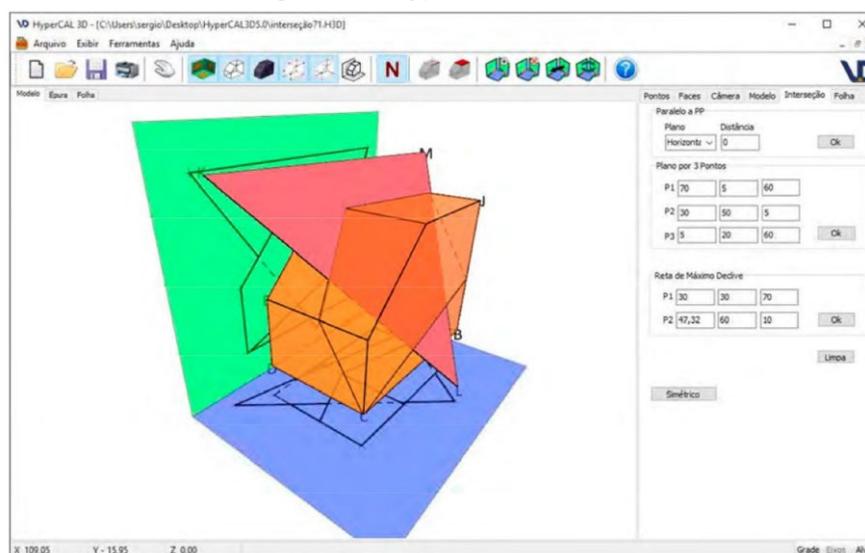
Figura 22 - Folha padronizada da épora confeccionada no HyperCAL^{3D}



Fonte: (SANTOS, 2016)

A Criação de planos de interseção foi adicionada na versão 5.0 do programa, e pode ser feita de três formas distintas. Com a criação de um plano paralelo a um plano existente (horizontal, frontal ou de perfil). A partir da criação de três pontos que definem um plano, Figura 23. Com a definição de um plano baseado em sua reta de máximo declive (SANTOS, 2016).

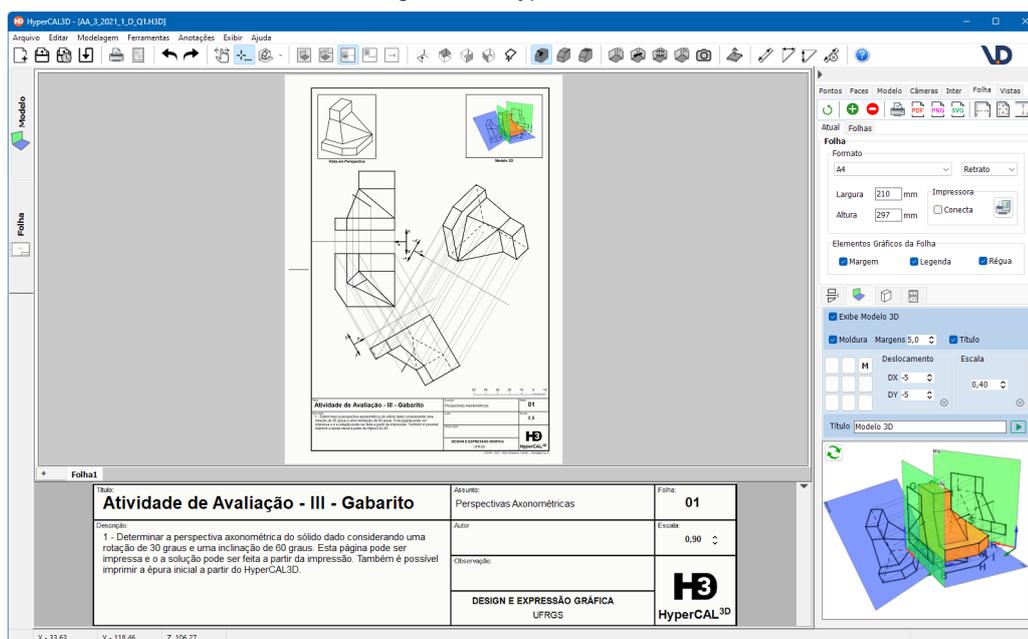
Figura 23 - HyperCAL versão 5.0



Fonte: (SANTOS, 2016)

Atualmente, o HyperCAL^{3D} se encontra na versão 9.0 (Figura 24). O aplicativo conta com uma interface mais moderna e com desempenho gráfico aprimorado. A criação de folhas impressas ganhou novas funcionalidades e uma nova interface, que permite a criação de materiais educacionais com grande agilidade e precisão, incluindo a Épura, Vistas em Perspectiva, Modelo 3D e Dados geométricos.

Figura 24 - HyperCAL 3D versão 9.0



Fonte: TEIXEIRA (2023)

No final de 2020, foi iniciado o desenvolvimento da versão para a navegador. A interface do HyperCAL^{3D} web foi projetada pelo Prof. Dr. Sérgio Leandro dos Santos que realizou toda a organização das classes e estrutura do programa. O mesmo código utilizado na versão web é utilizado na *mobile*, pois o HyperCAL^{3D} *mobile* foi desenvolvido como um aplicativo híbrido, que pode ser utilizado no navegador do celular ou tablet e posteriormente a implementação de todas as suas funcionalidades será compilado para IOS e android. No entanto, para fins desta pesquisa a versão *mobile* foi considerada como aplicativo, diferenciado da versão web, pois necessita da implementação específica da interface gestual. Portanto, o HyperCAL^{3D} web foi considerado como site para utilização em navegadores de computador e HyperCAL^{3D} *mobile* como um aplicativo que pode ser acessado via navegador de dispositivos móveis.

3. METODOLOGIA DA PESQUISA

Neste capítulo são apresentados os procedimentos metodológicos utilizados para o desenvolvimento desta pesquisa.

Durante algum tempo designers buscaram conhecimentos científicos nos métodos das ciências naturais. “Uma ciência natural é um corpo de conhecimento sobre alguma classe de coisas, objetos ou fenômenos no mundo: sobre as características e propriedades que elas possuem; sobre como eles se comportam e interagem entre si”⁹ (SIMON, 1996). Em outras palavras, é a observação de fenômenos na busca por informações que determinem como eles ocorrem. O cientista participa apenas como mero observador e não participa ativamente do experimento (DRESCH, LACERDA e ANTUNES, 2015).

As ciências naturais vêm contribuindo com o crescimento do conhecimento há vários anos e não é absurda a ideia de tentar alinhar a ciência do design com a essa forma de fazer ciência. Por outro lado, a artificialidade passou a fazer parte da vida cotidiana e requer novas formas de pesquisa e desenvolvimento.

Por esse ângulo, o estudo de design deixa de lado os moldes das ciências naturais e passa a ser guiado pela ciência do artificial, na qual o designer participa ativamente e sem se desvincular do método científico, pois é parte fundamental na construção do fenômeno, do artificial e de seus artefatos (DRESCH, LACERDA e ANTUNES, 2015). Mesmo parecendo que as ciências naturais e as ciências do artificial são opostas, para Simon (1996), os artefatos estão inseridos na natureza e desta forma não podem se opor ou ignorar suas leis.

Em meio ao paradigma teórico/ prático surgiu o método de pesquisa *Design Science Research* (DSR) que foi construído com base nos princípios da Design Science¹⁰ ou Ciência do artificial, que apesar de possuir Design em sua nomenclatura não representa somente o estudo do design e sim de todas as áreas que tratam da confecção do artificial ou de ciências balizadas por projetos que aproximem teoria e prática.

⁹ Tradução livre do texto: “A natural science is a body of knowledge about some class of things objects or phenomena in the world: about the characteristics and properties that they have; about how they behave and interact with each other.” (SIMON, 1996)

¹⁰ A nomenclatura Design Science foi sugerida em 1969 por Herbert Simon.

A DSR é multidisciplinar, constitui um processo rigoroso de projetar artefatos para resolver problemas complexos relevantes a um determinado contexto. O conhecimento obtido com sua aplicação não é descritivo, que representa os dados agora, mas sim prescritivo, que visa descrever quais as melhores ações a serem tomadas em cada cenário (LACERDA, *et al.*, 2013)

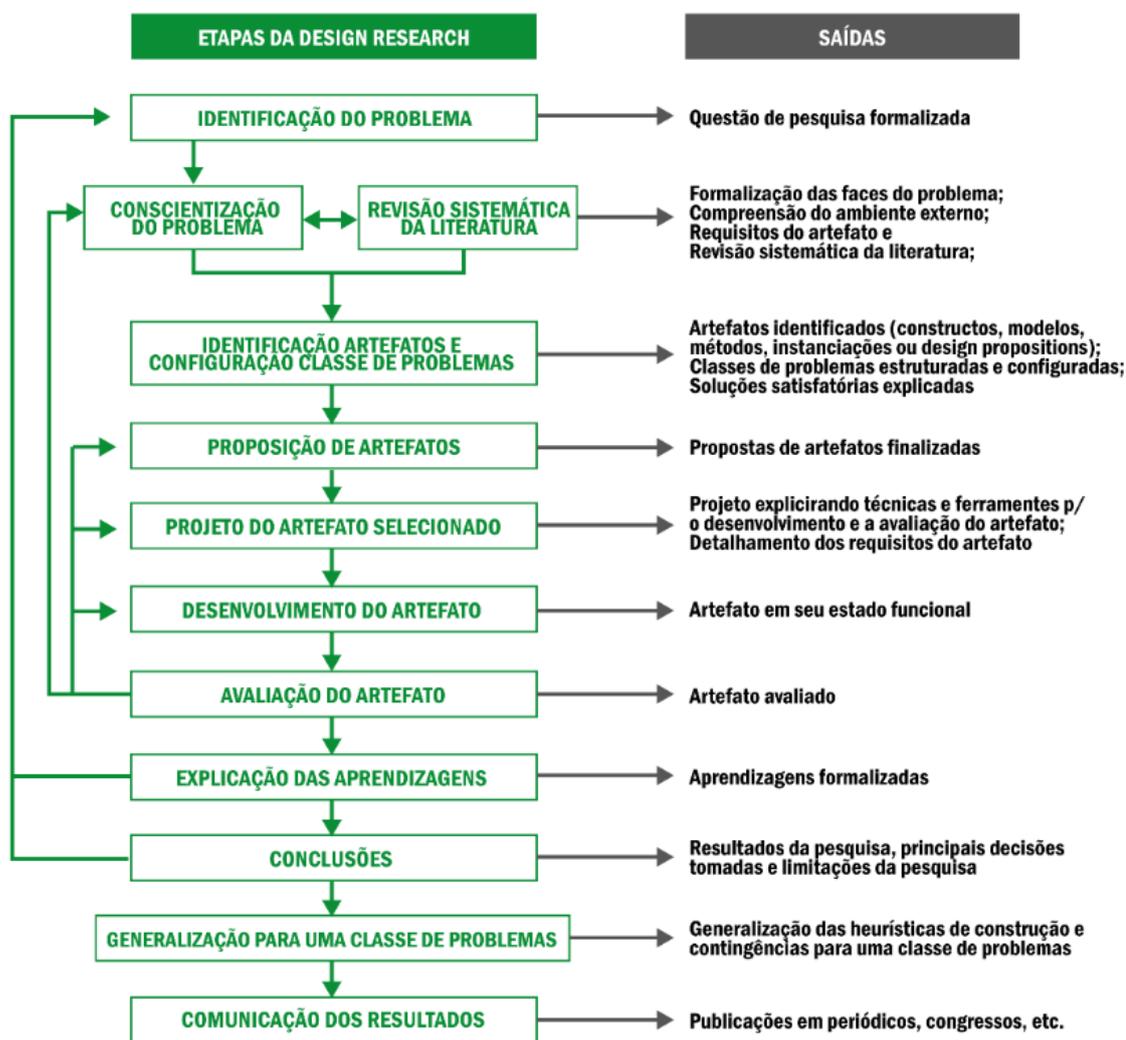
As pesquisas em *m-learning* vêm ganhando espaço no meio acadêmico, em especial no ano de 2020 devido a demanda por ensino a distância e remoto desencadeada pela pandemia de COVID-19. Uma parte considerável da pesquisa de *m-learning* está subjugada pela necessidade de criação de artefatos que atendam suas demandas preexistentes e, posteriormente, abram espaço para novos estudos.

A criação de artefatos digitais é uma tarefa multidisciplinar que requer o conhecimento, organização de quantidade considerável de informações e *expertises*. As pesquisas em *m-learning* devido à complexidade de adaptação aos dispositivos existentes e as rápidas mudanças na tecnologia, acabam por deixar de lado as adaptações e criações de ferramentas de ensino e passaram a analisar questões como as vantagens e desvantagens do uso de dispositivos móveis na educação, a criação de frameworks, formas de assistência ao aprendizado e adoção de novas tecnologias no ensino futuro (ZHANG, Y.A., 2015).

Essa lacuna verificada na literatura é causada, predominantemente, pela necessidade de desenvolvimento de artefatos capazes de testar os postulados teóricos de aprendizagem existentes. A DSR, por sua vez, procura reduzir a distância entre teoria e prática e sua utilização parece apropriada para a resolução do problema desta pesquisa.

Dresch, Lacerda e Antunes (2015), propõem a utilização de doze etapas para o desenvolvimento da pesquisa, que são: identificação do problema, conscientização do problema, revisão sistemática da literatura, identificação dos artefatos e configuração das classes de problemas, proposição de artefatos para resolução do problema, projeto do artefato, desenvolvimento do artefato, avaliação do artefato, explicitação das aprendizagens, conclusão, generalização para uma classe de problemas e comunicação dos resultados. A Figura 25 sintetiza as etapas idealizadas pelos autores.

Figura 25 - Etapas da metodologia Design Science Research



Fonte: Adaptado de Dresch, Lacerda e Antunes Jr. (2015)

A seguir são apresentadas as etapas constituintes da DSR e como foram resolvidas na construção desta pesquisa. Ao final desta apresentação é exposta uma figura com o desenho desta pesquisa baseado nas etapas da DSR.

3.1. IDENTIFICAÇÃO DO PROBLEMA

Surge principalmente do interesse do pesquisador em uma informação ou responder uma pergunta ou uma solução para um problema prático. Após sua identificação e compreensão, deve ser postulado em uma pergunta formalizada (DRESCH, LACERDA e ANTUNES, 2015).

Nesta pesquisa, a identificação do problema surgiu da necessidade de ampliação do acesso ao HyperCAL ^{3D} para diferentes plataformas, em especial

dispositivos móveis. Devido ao fato de vários estudantes possuírem *smartphones* de uso pessoal, a utilização destes pode auxiliar nas aulas a distância e presenciais em salas que não possuem computadores ou estes não são suficientes para a quantidade de alunos. Acrescentando a isso, a motivação pessoal da autora na computação gráfica e programação. A questão formalizada pode ser encontrada no capítulo introdutório deste relatório.

3.2. CONSCIENTIZAÇÃO DO PROBLEMA

O pesquisador deve reunir o máximo de informações possível para garantir uma compreensão extensa de todos os aspectos do problema, entendendo seu contexto e suas causas. Assim como, deve considerar a funcionalidade, o desempenho esperado e os requisitos operacionais do artefato (DRESCH, LACERDA e ANTUNES, 2015).

Para compreender a extensão do problema de pesquisa é necessária a conscientização de onde se pretende chegar. Esta definição permite ao pesquisador encontrar o ponto que pretende atingir como o desenvolvimento do trabalho. Assim, o objetivo geral da pesquisa foi traçado.

Adaptar o *software* de ensino de geometria descritiva HyperCAL^{3D}, utilizado nos cursos de Arquitetura e Urbanismo, Design de Produto e Design Visual e diversos cursos de Engenharia da UFRGS, ao *mobile learning*, tendo como base os conceitos de desenvolvimento de *m-learning* apoiados na usabilidade de dispositivos móveis e interfaces gestuais.

Em um primeiro momento, sem a pretensão de uma revisão sistemática, foi realizada uma pesquisa nas bases de dados SCOPUS (<http://www.scopus.com>) e Science Direct (<http://www.sciencedirect.com>) com a string de busca "*mobile learning*" and "*descriptive geometry*" que não retornou resultados.

Traçar o objetivo geral da pesquisa permite ao pesquisador compreender as etapas necessárias a resolução deste objetivo. Dividir esta meta em etapas menores, leva a conclusão dele. Portanto, os objetivos específicos são os pequenos degraus que levam a conclusão da pesquisa.

Desta forma, foi necessária a compreensão do *mobile learning*, do que se trata, autores, definição etc. Como é um campo de estudos multidisciplinar, a compreensão do desenvolvimento de *m-learning*, foi importante para a identificação das práticas utilizadas. Por este motivo, o primeiro objetivo específico desta pesquisa procurou responder questões referentes ao desenvolvimento da aprendizagem móvel independente de áreas de estudo.

3.3. REVISÃO SISTEMÁTICA DA LITERATURA

Auxilia na compreensão do conhecimento existente, “independentemente do tipo de ciência que o gerou, ajuda o pesquisador a justificar tanto a importância da construção de um artefato quanto por que ele funcionará” (DRESCH, LACERDA e ANTUNES, 2015, p. 113).

Foi realizada uma revisão sistemática da literatura, baseada nas etapas inicialmente propostas por Ellwanger (2018), com o objetivo de compreender o *mobile learning* e identificar estratégias para o desenvolvimento de aplicações com esta finalidade. As etapas de realização da revisão sistemática estão detalhadas no capítulo 4 deste relatório.

Após a conclusão da revisão, ficou evidente a necessidade de aproximação com a área do Design, visto que os artigos encontrados tratavam em geral de sistemas de informação e modelos de aceitação, portanto foi utilizada a revisão sistemática realizada por Machado e Vergara (2020) sobre a usabilidade em dispositivos móveis, por ser uma revisão atualizada.

3.4. IDENTIFICAÇÃO DOS ARTEFATOS E CONFIGURAÇÃO DAS CLASSES DE PROBLEMAS

A partir das compreensões obtidas na revisão sistemática e os conceitos de usabilidade, assim como das funcionalidades do HyperCAL ^{3D}, foi realizado um cruzamento teórico com a finalidade de encontrar requisitos para a análise de aplicações similares ao HyperCAL ^{3D}. Este cruzamento pode ser visto na íntegra no item 4.2.2 deste relatório, as questões desenvolvidas para a análise são listadas abaixo:

- Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)?
- O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?
- Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?
- O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

Na busca por ferramentas similares, inicialmente foi realizada uma pesquisa por aplicativos de ensino de geometria descritiva na loja Google Play e Apple Store, que retornou alguns aplicativos, porém voltados para cálculos geométricos. Estes foram descartados pois não apresentam similaridade com o HyperCAL ^{3D}.

Por esse motivo, foram realizadas mais duas buscas nas mesmas plataformas. A primeira buscava identificar aplicativos para modelagem 3D, por apresentarem similaridades com as funcionalidades da *viewport* 3D do HyperCAL ^{3D}. Enquanto a segunda buscou apps de desenho técnico, devido à similaridade com a *épura*. Inicialmente foi realizada uma filtragem dos resultados a partir da descrição do desenvolvedor na plataforma, aplicativos de escultura, esboço a mão livre, impressão e livros, entre outros foram dispensados. Foram selecionados aplicativos de modelagem e edição 3D e desenho CAD. A lista completa pode ser vista no Apêndice A deste relatório.

Após a realização das análises, as aprendizagens obtidas e os estudos das ferramentas do software HyperCAL ^{3D}, foi possível identificar a classe de problemas na qual esta pesquisa está inserida, como: “utilização de recursos e interfaces gestuais em aplicativos de modelagem 3D e desenho 2D”.

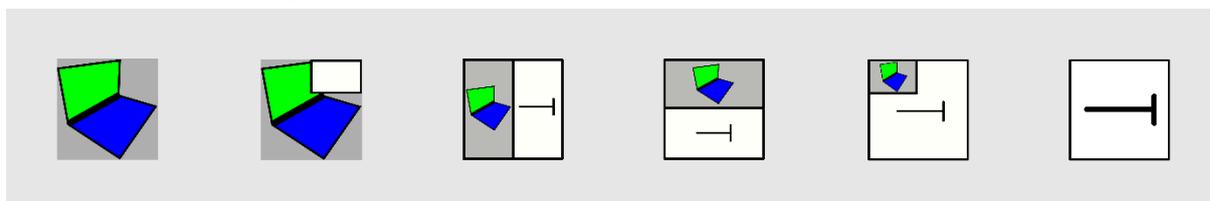
3.5. PROJETO E DESENVOLVIMENTO DO ARTEFATO

Nesta etapa devem ser consideradas as organizações internas dos componentes do artefato e a relação delas com o ambiente externo dele. Também é importante a descrição dos procedimentos de construção e avaliação (DRESCH, LACERDA e ANTUNES, 2015).

Nesta pesquisa o projeto e desenvolvimento ocorreram em pequenos ciclos. É possível diferenciá-los em termos metodológicos no *Framework* Scrum e na Metodologia XP, mesmo que não tenham sido utilizadas de forma direta. Tendo em vista que, para esta pesquisa, as *viewport* 3D e *épura* são fundamentais para o desenvolvimento das demais etapas do projeto, elas foram priorizadas no início da construção do artefato.

Como já foi citado, o professor Dr. Sérgio Santos iniciou a confecção da versão para a internet que serviu de base para a versão móvel. Quando a autora ingressou no projeto, a interface gráfica, as *viewport* 3D e *épura* já estavam inseridas no layout e podiam ser reorganizadas de acordo com a seleção dos botões ilustrados na Figura 26.

Figura 26 - Botões de layout das viewport 3d e *épura* do HyperCAL^{3D} web



Fonte: HyperCAL^{3D} web

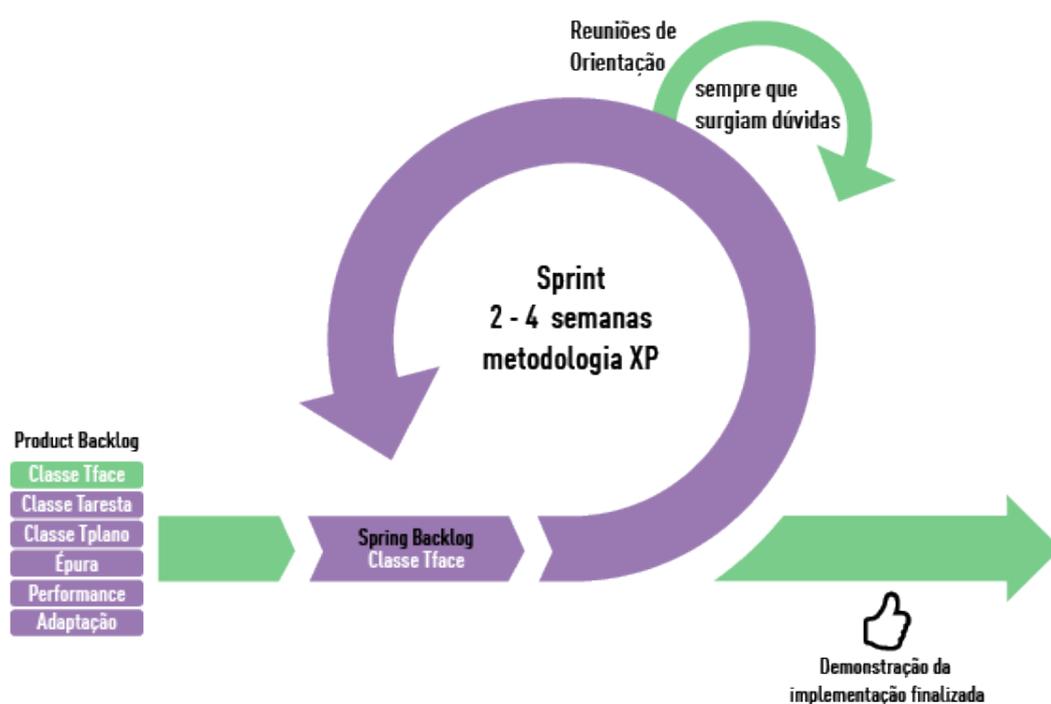
Além destes itens, a classe *Tponto*, responsável pela representação dos pontos nas *viewport* 3D e *épura*, já havia sido implementada. Devido ao conhecimento extensivo dos orientadores desta pesquisa do *software* HyperCAL^{3D}, as etapas de desenvolvimento ou o *Product Backlog* foi dividido em ciclos menores e interdependentes. Dentro desta analogia, é possível colocá-los simultaneamente na posição de *Scrum Master* e *Product Owner*, enquanto a autora se enquadra na posição de *Scrum Team*.

Em cada pequeno *Sprint*, foi sugerido à autora a implementação de um *Sprint Backlog*. Ao todo foram seis ciclos: desenvolvimento da classe *Tface*,

desenvolvimento da classe Taresta, desenvolvimento da classe Tplano, desenvolvimento da épura, reparo do desempenho e adaptações às proposições do artefato desta pesquisa, que é fruto desta pesquisa. Os primeiros ciclos apresentam funcionalidades similares ao HyperCAL ^{3D} *desktop*, com alterações na linguagem e funções para a adaptação a nova tecnologia. No entanto, o sexto e último ciclo, por se tratar dos conhecimentos adquiridos com esta pesquisa, apresentam as necessidades elencadas no item de proposição do artefato. Uma diferença importante a ser destacada, é que não foram realizadas reuniões diárias, mas as reuniões de orientação foram fundamentais no processo do desenvolvimento, pois sanaram dúvidas importantes.

É possível compreender que o *Framework* Scrum serviu de base o projeto do artefato, pois organizou o processo de implementação. Por outro lado, o desenvolvimento apresenta ligação direta com os conhecimentos de programação da autora, que foram sendo aprimorados no decorrer da implementação de cada etapa. A Figura 27 ilustra um ciclo do projeto e desenvolvimento do artefato. No item 1534.5 são descritos os ciclos do projeto, assim como o desenvolvimento da Metodologia XP na implementação dos códigos.

Figura 27 - *Framework* Scrum e Metodologia XP



Fonte: a autora (2022)

3.6. AVALIAÇÃO DO ARTEFATO

Após o desenvolvimento do artefato, o pesquisador deve considerar se ele atende a solução do problema, revisitando os requisitos definidos na conscientização do problema e comparando com os resultados apresentados (DRESCH, LACERDA e ANTUNES, 2015).

Durante todo o estudo ficou evidente a necessidade de avaliação com enfoque no usuário. Portanto, é utilizada para esta avaliação a escala de usabilidade MATCH - *Measuring Usability of Touch screen Phone Applications* descrita no item 2.6 deste relatório. As respostas justificadas do questionário, assim como a pontuação final do HyperCAL ^{3D} *mobile* são apresentadas no item 4.6 deste relatório.

3.7. EXPLICITAÇÃO DAS APRENDIZAGENS

Neste ponto, são identificados os principais aspectos do estudo e do artefato desenvolvido. Esta etapa, corresponde à identificação de fatores que contribuíram para o sucesso da pesquisa, assim como os elementos que não funcionaram (DRESCH, LACERDA e ANTUNES, 2015).

3.8. CONCLUSÃO

São expostos os resultados encontrados, assim como as limitações do estudo e apresentadas as possibilidades para trabalhos futuros (DRESCH, LACERDA e ANTUNES, 2015).

3.9. GENERALIZAÇÃO PARA UMA CLASSE DE PROBLEMAS

Nesta fase o pesquisador busca generalizar a solução encontrada para uma classe de problemas. (DRESCH, LACERDA e ANTUNES, 2015). Esta etapa é apresentada juntamente com a conclusão do estudo no capítulo 6 deste relatório.

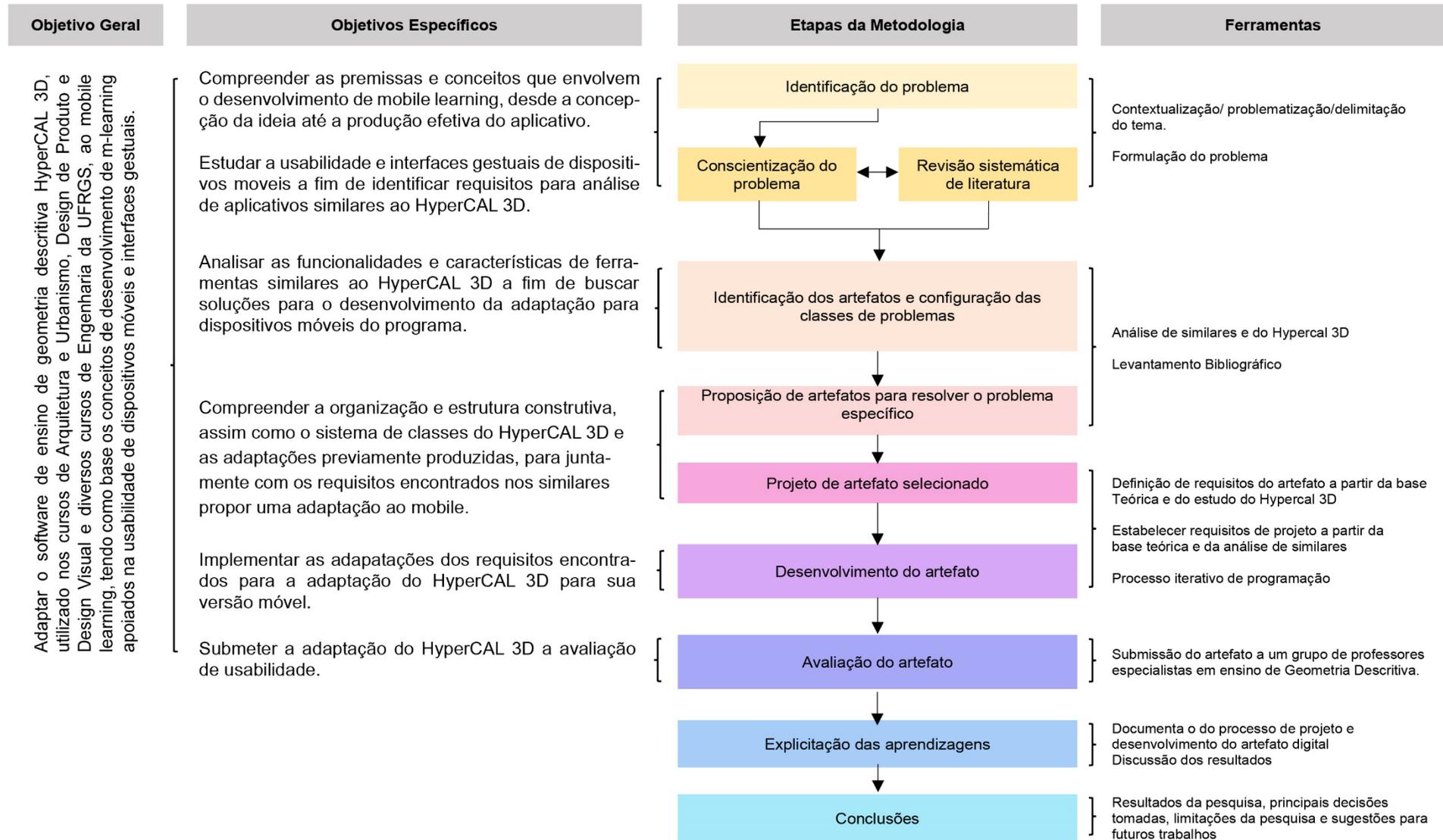
3.10. COMUNICAÇÃO DOS RESULTADOS

Tem o propósito de divulgar os resultados obtidos, com a publicação do trabalho em periódicos, congressos, seminários etc. (DRESCH, LACERDA e

ANTUNES, 2015). Nesta pesquisa, os resultados são divulgados na conclusão da tese de doutorado, assim como na publicação de artigos oriundos dela e a publicação do HyperCAL ^{3D} *mobile*. Esta etapa não é apresentada no desenho da pesquisa, pois a conclusão dela já atende a seus critérios.

De acordo com o exposto acima, a Figura 28 representa o desenho desta pesquisa seguindo as etapas da metodologia de pesquisa DSR idealizadas por Dresch, Lacerda e Antunes (2015).

Figura 28 - Desenho da pesquisa



Fonte: a autora (2022)

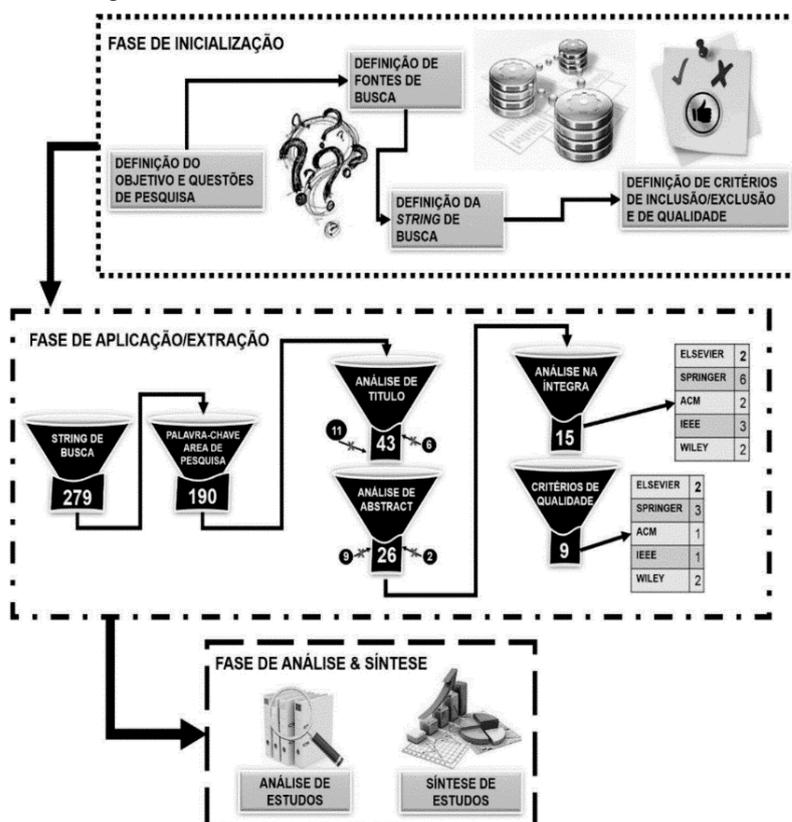
4. DESENVOLVIMENTO DA PESQUISA

Neste item são apresentadas as etapas de desenvolvimento que levaram à síntese de soluções necessárias a proposição do artefato. Em um primeiro momento, foi realizada uma revisão sistemática com a finalidade de compreender o *mobile learning* e encontrar requisitos para o desenvolvimento de aplicações *m-learning*.

4.1. REVISÃO SISTEMÁTICA DE LITERATURA

Como já citado, a conscientização do problema partiu da transformação do problema de pesquisa em objetivo geral. Assim, foi possível compreender a onde o estudo visa chegar. Posteriormente, foram elencados os objetivos específicos, que são os pequenos passos que devem ser alcançados na busca pelo objetivo final. Em leituras preliminares ficou evidente a dimensão do *mobile learning* e a necessidade de focar a pesquisa em uma área específica deste campo de estudo. Portanto, o primeiro degrau neste processo, foi a necessidade de compreensão do desenvolvimento de *mobile learning*. Para tanto, foi realizada uma revisão sistemática de Literatura em etapas sintetizadas na Figura 29, que são detalhadas a seguir.

Figura 29 - Fases da Revisão Sistemática de Literatura



Fonte: (ELLWANGER, 2018)

Uma revisão sistemática visa verificar e expor estudos relevantes em um relatório de investigação sistemático. Apoiada em métodos explícitos e sistemáticos para a coleta, seleção, análise e avaliação de estudos, busca responder uma ou mais perguntas específicas (ELLWANGER, 2018). Existem diversas formas de se produzir uma revisão sistemática, para a realização desta foram utilizadas as orientações da autora Ellwanger (2018) que propõem o desenvolvimento em três etapas fundamentais: Inicialização, Aplicação/Extração e Análise & Síntese.

Fase de Inicialização

Na fase de Inicialização foi definido o objetivo da revisão sistemática, assim como as questões de pesquisa para as quais se buscavam respostas, as fontes de dados que foram utilizadas, a *string* de busca e os critérios de inclusão, exclusão e de qualidade utilizados. Estes parâmetros são apresentados nas próximas seções.

Objetivo e questões de pesquisa

Esta revisão foi realizada de forma a responder o primeiro objetivo específico desta pesquisa, que busca a compreensão do cenário de desenvolvimento de *mobile learning*, desde o projeto até a confecção dos aplicativos. Para tanto foram definidas as seguintes questões de pesquisa:

- Quais características são consideradas indispensáveis em um aplicativo que visa o *m-learning*?
- Como é feito o planejamento para o desenvolvimento?
- Existem lacunas no desenvolvimento de *mobile learning*?
- Como organizar a programação do aplicativo?

Definição das fontes de busca

Como base de dados, inicialmente foram adotadas as bases SCOPUS (<http://www.scopus.com>), IEEE Xplore Digital Library (<http://ieeexplore.ieee.org>) e Science Direct (<http://www.sciencedirect.com>). No entanto em uma pesquisa preliminar por “Mobile Learning development” foi obtido apenas um artigo na base de dados Science Direct, que foi descartado.

Definição da *string* de busca

Para a definição da *string* de busca, inicialmente a pesquisa foi direcionada para o desenvolvimento *mobile* a partir das palavras: *development* e *design*. As palavras *framework* e *model* foram posteriormente acrescentadas, pois são comumente utilizadas no início do desenvolvimento de ferramentas, em forma de método, ou ao final, como resultado de um processo de desenvolvimento. Para finalizar foi utilizado ensino superior ou *higher education* pois esta pesquisa visa alunos universitários.

A *string* de busca finalizada ficou: "Mobile Learning development" or "mobile learning design" or "Mobile Learning model" or "Mobile Learning framework" or "m-learning development" or "m-learning design" or "m-learning model" or "m-learning framework" and "higher education"

Após a definição das bases de dados e da *string* de busca, foram definidos os critérios para a inclusão e exclusão dos estudos encontrados.

Definição dos critérios de inclusão/exclusão e qualidade

Os critérios de inclusão foram baseados atualidade, acesso, área de pesquisa, em que língua foram escritos. Como critérios de inclusão foram definidos os seguintes filtros:

- foi determinado um período de 10 anos para a busca de estudos. Portanto, artigos publicados no período de 2011 a 2021,
- artigos disponíveis para acesso (*open access*),
- trabalhos que se vinculam às áreas foco desta pesquisa *mobile learning development* e Design.
- a disponibilidade dos trabalhos em língua portuguesa ou inglesa.

Para exclusão de artigos foram considerados, integralidade do conteúdo, título e palavras-chave, *abstract*, reincidência, conteúdo repetido e se tratava-se de revisões sistemáticas. Como critérios de exclusão foram definidos:

- a indisponibilidade do trabalho em sua íntegra,

- trabalhos que não apresentem indícios de relação com o foco desta pesquisa identificáveis pelo título e palavras chaves,
- artigos que não façam referência ao desenvolvimento de *mobile learning* (verificável pela leitura do abstract),
- artigos repetidos (devido a utilização de mais de uma base de consulta),
- trabalhos de um mesmo autor que apresentem títulos diferentes, mas com conteúdo igual e
- revisões sistemáticas de literatura por representarem buscas do autor e não necessariamente o foco desta pesquisa.

Também foram definidos, na forma de pergunta para facilitar a análise, critérios de qualidade que foram aplicados durante a leitura dos trabalhos na íntegra, como critérios de qualidade foram elencados:

(C1) Apresenta alguma proposta, aplicação prática ou experimento com validação?

(C2) Em termos metodológicos, o estudo se apresenta de forma clara e passível de replicação?

(C3) Imagens, gráficos ou tabelas são claros e compreensíveis?

(C4) A conclusão ou as considerações finais apresentam os benefícios e limitações do estudo e oferecem direcionamentos para trabalhos futuros?

No próximo item são descritos os procedimentos da segunda etapa da revisão elencados por Ellwanger (2018).

Fase de Aplicação/ Extração

Nesta etapa, a *string* de busca foi aplicada às bases de dados selecionadas. Devido a diferenças sistêmicas nas fontes de dados, a busca precisou ser adaptada a cada site. No site Scopus, a pesquisa foi realizada com uma busca por títulos, resumos e palavras chaves. Para tanto, foi apenas necessário colocar a *string* completa no campo de pesquisa e selecionar “*Article title, Abstract, Keyword*” no menu “*Search within*”.

Na base de dados IEEE não é possível pesquisar títulos, resumos e palavras chaves de uma única vez, portanto foram realizadas três pesquisas. Assim, foi necessária a inserção manual de cada termo em um campo específico, a indicação de busca, primeiramente por título, logo após por abstract e posteriormente por palavras chaves.

Desta forma obtiveram-se três *strings* que foram utilizadas para a busca de artigos que possuem os termos no título, resumo e palavras-chave que são descritas respectivamente a seguir:

String 1 ("Document Title": Mobile Learning development) OR ("Document Title": mobile learning design) OR ("Document Title": Mobile Learning model) OR ("Document Title": Mobile Learning framework) OR ("Document Title": m-learning development) OR ("Document Title": m-learning design) OR ("Document Title": m-learning model) OR ("Document Title": m-learning framework) AND ("Document Title": higher education)

String 2 ("Abstract": Mobile Learning development) OR ("Abstract": mobile learning design) OR ("Abstract": Mobile Learning model) OR ("Abstract": Mobile Learning framework) OR ("Abstract": m-learning development) OR ("Abstract": m-learning design) OR ("Abstract": m-learning model) OR ("Abstract": m-learning framework) AND ("Abstract": higher education)

String 3 ("Author Keywords": Mobile Learning development) OR ("Author Keywords": mobile learning design) OR ("Author Keywords": Mobile Learning model) OR ("Author Keywords": Mobile Learning framework) OR ("Author Keywords": m-learning development) OR ("Author Keywords": m-learning design) OR ("Author Keywords": m-learning model) OR ("Author Keywords": m-learning framework) AND ("Author Keywords": higher education)

Com a finalidade de reduzir o número de publicações, alguns filtros foram selecionados ainda nas bases de dados, pois eles confirmavam os critérios de inclusão/exclusão. A busca foi inicialmente reduzida pelo período de 10 anos e foi selecionada a opção *Open Access*. Desta forma, foram obtidos os resultados descritos no Quadro 2.

Quadro 2 - Resultados obtidos na busca pelas *Strings*

Base de dados	Local de busca	Número de artigos
Scopus	Título, resumo e palavras-chave	26
IEEE Xplore	Título	24
IEEE Xplore	Resumo	460
IEEE Xplore	Palavras-chave	26

Fonte: a autora (2022)

Os resultados obtidos foram enviados para na ferramenta Rayyan (<https://rayyan.ai>) (OUZZANI, *et al.*, 2016) para a aplicação dos demais critérios de inclusão e exclusão. Devido a busca ter sido realizada mais de uma vez em um mesmo site, o primeiro critério utilizado foi determinar se havia artigos duplicados. Dos 536 estudos, 74 estavam duplicados e foram excluídos.

Foram observados diversos títulos relacionados a aprendizagem de máquinas, que não possui ligação com esta pesquisa, por esse motivo foi realizada uma filtragem com os termos *mobile learning* e *m-learning*. Neste ponto restavam 462 artigos que foram reduzidos a 52 que seguiram para a leitura de títulos. Após a leitura dos títulos dos trabalhos, foram selecionados 22 artigos para a leitura do resumo, devido sua similaridade com o foco da revisão. Ao final da leitura dos resumos, 16 estudos que foram selecionados para a análise de qualidade.

Os trabalhos que atendem plenamente o critério observado recebe a pontuação 1, os que atendem parcialmente recebem a pontuação 0.5 e os que não atendem os critérios recebem uma pontuação 0. O somatório das notas de cada pergunta determina a classificação do estudo. As notas finais de cada artigo ficarão entre 0 e 4 pontos sendo:

- 0.0 até 1.0 pontos para fraco
- 1.5 até 2.5 pontos médio
- 3.0 até 4.0 pontos bom

No total dois estudos não atingiram pelo menos a pontuação total média e foram descartados por falta de qualidade. O Quadro 3 apresenta as pontuações obtidas para cada pergunta, assim como os artigos que foram dispensados por não atenderem aos critérios estabelecidos riscados em vermelho.

Quadro 3 Avaliação de trabalhos por critérios de qualidade

AUTORES	TÍTULO	C1	C2	C3	C4	TOTAL
Todoranova, L.; Penchev, B. (2020)	A Conceptual Framework for Mobile Learning Development in Higher Education	0.5	0	0.5	0	4
Almaiah, M. A.; Alamri, M. M.; Al- Rahmi, W. M. (2020)	Analysis the Effect of Different Factors on the Development of Mobile Learning Applications at Different Stages of Usage	0.5	1	1	0.5	3.5
Chiu, P.-S.; et all (2018)	An authentic learning-based evaluation method for mobile learning in Higher Education	0.5	1	1	1	3.5
Almaiah, M. A.; Alamri, M. M.; Al- Rahmi, W.; (2019)	Applying the UTAUT Model to Explain the Students' Acceptance of Mobile Learning System in Higher Education	0.5	1	1	0.5	3
Malandrino, D.; et all (2015)	A Tailorable Infrastructure to Enhance Mobile Seamless Learning	1	1	1	1	4
Kumar, J. A.; et all (2020)	Behavioral Intention to Use Mobile Learning: Evaluating the Role of Self-Efficacy, Subjective Norm, and WhatsApp Use Habit	0.5	1	1	1	3.5
Alotaibi, F.D.; Siraj, S.; Ismail, W.M.A.S. (2019)	Design and development of mobile learning model for teaching arabic language reading skills to Non- Arab speakers in higher education institutions	0.5	0	0.5	0	4
Alowayr, A. (2021)	Determinants of mobile learning adoption: extending the unified theory of acceptance and use of technology (UTAUT)	0.5	1	1	1	4
A. Mukminin et all (2020)	Exploring the Drivers Predicting Behavioral Intention to Use m-Learning Management System: Partial Least Square Structural Equation Model	0.5	0.5	1	1	3.5
Sun, G.; Shen, J. (2014)	Facilitating Social Collaboration in Mobile Cloud- Based Learning: A Teamwork as a Service (TaaS) Approach	1	0.5	1	0.5	3.5
Alvarado, L. A. R. et all (2018)	Layered Software Architecture for the Development of Mobile Learning Objects With Augmented Reality	1	1	1	0.5	3.5
Sitar-Taut, D.-A.; Mican, D. (2021)	Mobile learning acceptance and use in higher education during social distancing circumstances: an expansion and customization of UTAUT2	0.5	0.5	1	0.5	3
Rodríguez, J. M. R.; et all (2020)	Mobile Learning in Higher Education: Structural Equation Model for Good Teaching Practices	0.5	1	1	0.5	3
Al-Adwan, A.S.; Al- Madadha, A.; Zvirzdinaite, Z. (2018)	Modeling students' readiness to adopt mobile learning in higher education: An empirical study	0.5	1	1	1	3.5
Sarrab, M.; et all (2018)	Toward Educational Requirements Model for Mobile Learning Development and Adoption in Higher Education	0.5	1	1	0.5	2.5
Bikanga Ada, M.; Stansfield, M.; Baxter, G. (2017)	Using mobile learning and social media to enhance learner feedback: Some empirical evidence	0.5	1	1	1	3.5

Legenda: **(C1)** - Proposta, aplicação prática ou experimento com validação, **(C2)** - metodologia clara e passível de replicação, **(C3)** Imagens, gráficos ou tabelas claros e compreensíveis, **(C4)** - Conclusão ou considerações finais apresentam os benefícios e limitações do estudo e oferecem direcionamentos para o desenvolvimento de trabalhos futuros.

Fase de Análise e Síntese

Nesta Fase, os trabalhos são analisados com maior atenção a fim de identificar mecanismos que possam ser utilizados na compreensão e resolução dos objetivos desta pesquisa. Os artigos “*A Conceptual Framework for Mobile Learning Development in Higher Education*” e “*Design and development of mobile-learning model for teaching arabic language reading skills to Non-Arab speakers in higher education institutions*” foram desconsiderados por não apresentar pontuação significativa. O Quadro 4 apresenta quantidade de artigos em cada critério de qualidade.

Quadro 4 – Pontuação final dos Critérios de qualidade

Perguntas	fracos	médios	bons
(C1) Apresenta alguma proposta, aplicação prática ou experimento com validação?	0	13	3
(C2) Em termos metodológicos, o estudo se apresenta de forma clara e passível de replicação?	2	5	9
(C3) Imagens, gráficos ou tabelas são claros e compreensíveis?	0	2	14
(C4) A conclusão ou as considerações finais apresentam os benefícios e limitações do estudo e oferecem direcionamentos para trabalhos futuros?	2	7	7

Fonte: a autora (2022)

Dos 16 estudos analisados, apenas três atingiram totalmente o critério C1. Durante a leitura dos artigos uma característica foi observada, a grande maioria dos artigos descreviam análises, mesmo isso caracterizando experimentos o que atende totalmente ao critério C1, estes artigos foram considerados como parciais. Isso se deve, a tentativa de separá-los dos artigos que representam desenvolvimentos efetivos de aplicações, que são o foco desta revisão. Este resultado não era esperado e, portanto, precisou ser incorporado a pesquisa. Desta forma, por representarem a grande maioria dos artigos, as análises foram aceitas como um fato de grande relevância para o estudo. Diante do exposto, apenas 19% dos estudos atenderam totalmente ao critério C1, enquanto 81% atenderam parcialmente e são analisados mais adiante no relatório.

Na segunda pergunta (C2) sobre a metodologia aplicada na pesquisa, 56% atendem totalmente, 31% parcialmente, pois em geral os estudos que foram considerados desta forma, apresentam conceitualmente a metodologia, porém não descrevem os passos para sua aplicação. Os outros 13% não expõem de forma clara

as questões metodológicas. Diante do exposto, é possível perceber que as bases de dados escolhidas para a revisão prezam, ainda que parcialmente, pela qualidade metodológica de seus estudos. Vale ressaltar que, dos cinco artigos da base Scopus, um não atingiu satisfatoriamente o critério metodológico, enquanto de onze estudos da IEEE, um também não atingiu a qualidade esperada.

De forma geral, todos os artigos apresentam imagens, gráficos ou tabelas claros. Apenas dois estudos apresentam gráficos confusos, porém possuem tabelas e imagens claras o que justifica estarem como parcialmente aceitos no critério C3.

No último critério de qualidade (C4), foram analisadas as conclusões e considerações finais a partir de três parâmetros fundamentais: benefícios do estudo, limitações e trabalhos futuros. Portanto, estudos que atendem apenas a um destes itens são considerados fracos, enquanto os que atendem a dois ou três são respectivamente médios e fortes. Em geral, todos os estudos apresentam seus benefícios; 44% apresentavam benefícios e trabalhos futuros, mas não apresentavam limitações; e outros 44% dos estudos atenderam a todos os parâmetros.

A seguir são apresentadas as análises e sínteses dos artigos que foram aprovados nos critérios de qualidade.

Fase de Análise & Síntese

Nesta fase de Análise & Síntese, os estudos que foram considerados significativos são examinados em profundidade, a fim de identificar como eles respondem as perguntas determinadas na Fase de Inicialização desta revisão:

- Quais características são consideradas indispensáveis em um aplicativo que visa o *m-learning*?
- Como é feito o planejamento para o desenvolvimento?
- Existem lacunas no desenvolvimento de *mobile learning*?
- Como organizar a programação do aplicativo?

Como foi mencionado na análise dos critérios de qualidade, uma parte considerável dos estudos apresenta análises. Desta forma, o Quadro 5 sinaliza os

estudos por uma separação de cores, sendo verde para os que representam o desenvolvimento de aplicações e azul para os que representam análises.

Quadro 5 – Divisão dos artigos entre análises e desenvolvimento

AUTORES	TÍTULO	C1	C2	C3	C4	TOTAL
Almaiah, M. A.; Alamri, M. M.; Al-Rahmi, W. M. (2020)	<i>Analysis the Effect of Different Factors on the Development of Mobile Learning Applications at Different Stages of Usage</i>	0.5	1	1	0.5	3.5
Chiu, P.-S.; et all (2018)	<i>An authentic learning-based evaluation method for mobile learning in Higher Education</i>	0.5	1	1	1	3.5
Almaiah, M. A.; Alamri, M. M.; Al-Rahmi, W.; (2019)	<i>Applying the UTAUT Model to Explain the Students' Acceptance of Mobile Learning System in Higher Education</i>	0.5	1	1	0.5	3
Malandrino, D.; et all (2015)	<i>A Tailorable Infrastructure to Enhance Mobile Seamless Learning</i>	1	1	1	1	4
Kumar, J. A.; et all (2020)	<i>Behavioral Intention to Use Mobile Learning: Evaluating the Role of Self-Efficacy, Subjective Norm, and WhatsApp Use Habit</i>	0.5	1	1	1	3.5
Alowayr, A. (2021)	<i>Determinants of mobile learning adoption: extending the unified theory of acceptance and use of technology (UTAUT)</i>	0.5	1	1	1	4
A. Mukminin et all (2020)	<i>Exploring the Drivers Predicting Behavioral Intention to Use m-Learning Management System: Partial Least Square Structural Equation Model</i>	0.5	0.5	1	1	3.5
Sun, G.; Shen, J. (2014)	<i>Facilitating Social Collaboration in Mobile Cloud-Based Learning: A Teamwork as a Service (TaaS) Approach</i>	1	0.5	1	0.5	3.5
Alvarado, L. A. R. et all (2018)	<i>Layered Software Architecture for the Development of Mobile Learning Objects With Augmented Reality</i>	1	1	1	0.5	3.5
Sitar-Taut, D.-A.; Mican, D. (2021)	<i>Mobile learning acceptance and use in higher education during social distancing circumstances: an expansion and customization of UTAUT2</i>	0.5	0.5	1	0.5	3
Rodríguez, J. M. R.; et all (2020)	<i>Mobile Learning in Higher Education: Structural Equation Model for Good Teaching Practices</i>	0.5	1	1	0.5	3
Al-Adwan, A.S.; Al-Madadha, A.; Zvirzdinaite, Z. (2018)	<i>Modeling students' readiness to adopt mobile learning in higher education: An empirical study</i>	0.5	1	1	1	3.5
Sarrab, M.; et all (2018)	<i>Toward Educational Requirements Model for Mobile Learning Development and Adoption in Higher Education</i>	0.5	1	1	0.5	2.5
Bikanga Ada, M.; Stansfield, M.; Baxter, G. (2017)	<i>Using mobile learning and social media to enhance learner feedback: Some empirical evidence</i>	0.5	1	1	1	3.5

Fonte: a autora (2022)

A análise dos estudos em verde, que buscam o desenvolvimento de aplicações levou as seguintes descobertas. O artigo “*Facilitating Social Collaboration in Mobile Cloud-Based Learning: A Teamwork as a Service (TaaS) Approach*” (SUN e SHEN, 2014) apresenta a criação de um algoritmo capaz de identificar características dos alunos para a organização de grupos em nuvem. No entanto, o algoritmo foi estruturado de forma analógica e seus resultados computados em um *software* de terceiros MATLAB, portanto foi desconsiderado para a análise & síntese.

O estudo “*Toward Educational Requirements Model for Mobile Learning Development and Adoption in Higher Education*” (SARRAB, *et al.*, 2018) promete a da criação de requisitos para a criação de *M-learning* baseado em Modelos de Design Instrucional, no entanto estes requisitos não são utilizados para o desenvolvimento de aplicações e sim adaptação dos conteúdos para programas como Moodle, Quadro Negro e *Schoology*.

Os demais artigos, “*A Tailorable Infrastructure to Enhance Mobile Seamless Learning*” (MALANDRINO, *et al.*, 2015) e “*Layered Software Architecture for the Development of Mobile Learning Objects With Augmented Reality*” (ALVARADO, *et al.*, 2018), apresentam arquiteturas organizadas em camadas, nas quais as camadas inferiores fornecem serviços para as superiores. Em outras palavras ambos utilizam banco de dados, interface customizável e potencial para serem escalonáveis (acréscimo de funções).

O primeiro estudo busca a criação de uma aplicação para uso geral (não para uma determinada matéria) que possa ser utilizada para qualquer dispositivo e, portanto, analisa que não é necessário o conhecimento dos dispositivos pertencentes aos alunos para sua confecção, assim utiliza uma linguagem web para resolver os problemas de compatibilidade com diferentes dispositivos. Já o segundo, visa a criação de objetos de aprendizagem móvel com realidade aumentada para dispositivos androide.

Considerando as questões definidas para a pesquisa desta revisão:

- Quais características são consideradas indispensáveis em um aplicativo que visa o *m-learning*?

Apesar dos estudos se preocuparem com as questões pedagógicas que envolvem o *m-learning*, não deixam claro como conseguiram atender a este requisito. Além disso, ambos analisam o contexto de uso dos aplicativos como essencial para o desenvolvimento das aplicações.

- Como é feito o planejamento para o desenvolvimento?

O planejamento é feito em camadas e possuem em suas estruturas banco de dados, interfaces customizáveis e podem ser escalonados para implementações futuras.

- Existem lacunas no desenvolvimento de *mobile learning*?

Nenhum dos estudos busca a solução para uma demanda específica, aparentemente ambos buscam a criação de plataformas genéricas como A.V.A Moodle. Portanto, será necessária a avaliação dos demais artigos para a definição de lacunas.

- Como organizar a programação do aplicativo?

Os estudos comentam quais as linguagens foram utilizadas e como foram organizadas as estruturas constitutivas, porém sem deixar claro como a programação foi realizada

Vale ressaltar que, o HyperCAL ^{3D} já possui uma arquitetura definida, no entanto na versão *desktop* os arquivos são armazenados localmente e não conta com um banco de dados para o armazenamento de arquivos e login de usuários.

Mesmo as perguntas tendo sido respondidas, a quantidade de artigos e as diferenças em seu desenvolvimento deixaram dúvidas sobre questões fundamentais para o desenvolvimento do *mobile learning*. Portanto, a avaliação dos estudos que tratam de análises, serviu para determinar outros fatores relevantes a criação de aplicações para *m-learning*.

Durante a avaliação, foi observado que a maior parte dos artigos se tratava de análises a partir de modelos de aceitação de tecnologia. No total, dez estudos realizaram análises, 70% deles utilizavam Modelos de Aceitação da Tecnologia. O que representa 50% de todos os artigos que passaram na análise de qualidade.

Dentre os modelos utilizados, destaca-se o emprego dos modelos TAM e UTAUT, assim como variações destes. Um estudo apresenta a criação de um novo modelo chamado MLAM, baseado no modelo UTAUT, porém que utiliza características da aplicação para a análise. Estes modelos podem ser encontrados no capítulo 2 deste relatório. O Quadro 6 apresenta quais modelos foram utilizados nos estudos e os coloca ordem temporal, sendo os primeiros mais atuais e os últimos menos.

Quadro 6 – Modelos utilizados em ordem temporal

AUTORES	TÍTULO	Desenvolvimento/ Modelos de aceitação / outras análises
Alowayr, A. (2021)	<i>Determinants of mobile learning adoption: extending the unified theory of acceptance and use of technology (UTAUT)</i>	Modelo de Aceitação UTAUT
Sitar-Taut, D.-A.; Mican, D. (2021)	<i>Mobile learning acceptance and use in higher education during social distancing circumstances: an expansion and customization of UTAUT2</i>	Modelo de Aceitação UTAUT2
Almaiah, M. A.; Alamri, M. M.; Al-Rahmi, W. M. (2020)	<i>Analysis the Effect of Different Factors on the Development of Mobile Learning Applications at Different Stages of Usage</i>	Modelo de Aceitação MLAM
Kumar, J. A.; et all (2020)	<i>Behavioral Intention to Use Mobile Learning: Evaluating the Role of Self-Efficacy, Subjective Norm, and WhatsApp Use Habit</i>	Modelo de Aceitação TAM e TPB
A. Mukminin et all (2020)	<i>Exploring the Drivers Predicting Behavioral Intention to Use m-Learning Management System: Partial Least Square Structural Equation Model</i>	Modelo de Aceitação TAM
Rodríguez, J. M. R.; et all (2020)	<i>Mobile Learning in Higher Education: Structural Equation Model for Good Teaching Practices</i>	Outras Análises
Almaiah, M. A.; Alamri, M. M.; Al-Rahmi, W.; (2019)	<i>Applying the UTAUT Model to Explain the Students' Acceptance of Mobile Learning System in Higher Education</i>	Modelos de Aceitação UTAUT
Al-Adwan, A.S.; Al-Madadha, A.; Zvirzdinaite, Z. (2018)	<i>Modeling students' readiness to adopt mobile learning in higher education: An empirical study</i>	Modelo de Aceitação TAM/ UTAUT modificados
Sarrab, M.; et all (2018)	<i>Toward Educational Requirements Model for Mobile Learning Development and Adoption in Higher Education</i>	Desenvolvimento
Alvarado, L. A. R. et all (2018)	<i>Layered Software Architecture for the Development of Mobile Learning Objects With Augmented Reality</i>	Desenvolvimento
Chiu, P.-S.; et all (2018)	<i>An authentic learning-based evaluation method for mobile learning in Higher Education</i>	Outras Análises
Bikanga Ada, M.; Stansfield, M.; Baxter, G. (2017)	<i>Using mobile learning and social media to enhance learner feedback: Some empirical evidence</i>	Outras Análises
Malandrino, D.; et all (2015)	<i>A Tailorable Infrastructure to Enhance Mobile Seamless Learning</i>	Desenvolvimento
Sun, G.; Shen, J. (2014)	<i>Facilitating Social Collaboration in Mobile Cloud-Based Learning: A Teamwork as a Service (TaaS) Approach</i>	Desenvolvimento

Fonte: a autora (2022)

O estudo dos artigos de análise deixou evidente uma grande lacuna que deve ser considerada na elaboração de aplicações para *mobile learning*, a intensão dos alunos de utilizarem ou não seus dispositivos pessoais para o ensino. Mesmo parecendo que a aceitação é automática, uma série de intenções comportamentais é abordada para minimizar a rejeição dos estudantes. Este fato, é de extrema importância quando comparado com o tempo e esforço necessários ao desenvolvimento de aplicações que visam o *m-learning*.

Quanto aos três estudos que não apresentam modelos de aceitação, possuem foco em três características distintas do *m-learning*. Os autores Chiu, Pu *et al.* (2018), analisam a aprendizagem autêntica em *mobile learning* a partir do ensino em situações reais ou simuladas, assim como na aprendizagem baseada em projetos. Para Ada, Stansfield e Baxter (2017) uma questão de suma importância é o retorno dos alunos quanto à aprendizagem móvel. Mesmo o artigo não apresentando modelos de aceitação, possui uma preocupação com aceitação por parte dos alunos. Por fim, Rodriguez *et al.* (2020) apresentam a preocupação com critérios que levam a boas práticas, por parte dos professores, de *m-learning*.

Como última análise foi realizada a avaliação temporal de forma a compreender quais teorias são mais atuais. Como pode ser observado (Quadro 6), os artigos mais atuais utilizam modelos de aceitação de tecnologia enquanto os menos atuais representam desenvolvimentos efetivos de aplicações e outras formas de análise.

Foram obtidas as seguintes respostas as perguntas desta revisão:

- Quais características são consideradas indispensáveis em um aplicativo que visa o *m-learning*?

É necessário levar em consideração as questões pedagógicas para o desenvolvimento de *m-learning*, como foi possível observar pelo menos um artigo apresenta como solução para isso, a aprendizagem baseada em projetos, que já é utilizada nas aulas de GD dos cursos de Design, Engenharia e Arquitetura da UFRGS.

É importante determinar o contexto de uso para a determinar as melhores práticas de desenvolvimento. Contextos formais de ensino apresentam maior disponibilidade de recursos que os contextos

informais, neste caso as aplicações acabam servindo de suporte ao ensino, mas não representam sua totalidade.

- Como é feito o planejamento para o desenvolvimento?

A determinação dos dispositivos é fundamental para a escolha das linguagens de programação. Aplicações mobile baseadas na web, podem utilizar os navegadores de internet para atingir o maior número possível de dispositivos.

Nestes casos a aplicação pode ser feita apenas no nível do cliente, porém a criação de um banco de dados pode auxiliar no desenvolvimento. Planejar a aplicação em camadas auxilia na sua escalabilidade. Além disso, interfaces customizáveis são bem-vindas.

- Existem lacunas no desenvolvimento de *mobile learning*?

A vontade dos alunos em utilizar seus dispositivos pessoais pode significar o sucesso ou fracasso de uma aplicação, desta forma identificar que parâmetros podem determinar esta aceitação podem diminuir esta lacuna.

- Como organizar a programação do aplicativo?

Esta pergunta não foi respondida de forma direta na revisão sistemática, no entanto os estudos de desenvolvimento demonstraram que cada aplicação deve ser pensada de forma individual. Como este estudo trata da adaptação do HyperCAL ^{3D} ao *mobile*, a estrutura utilizada será a baseada na versão *desktop*, porém com os recursos existentes nas linguagens de internet.

Após a conclusão da revisão, devido à aproximação com a área do Design, foi utilizada a revisão sistemática de Machado e Vergara (2020) sobre a usabilidade em dispositivos móveis para a identificação de possíveis requisitos para análise de similares. Para tanto, foi necessário o cruzamento teórico entre as teorias de *mobile learning*, usabilidade em dispositivos móveis e características do HyperCAL ^{3D} que é apresentado a seguir.

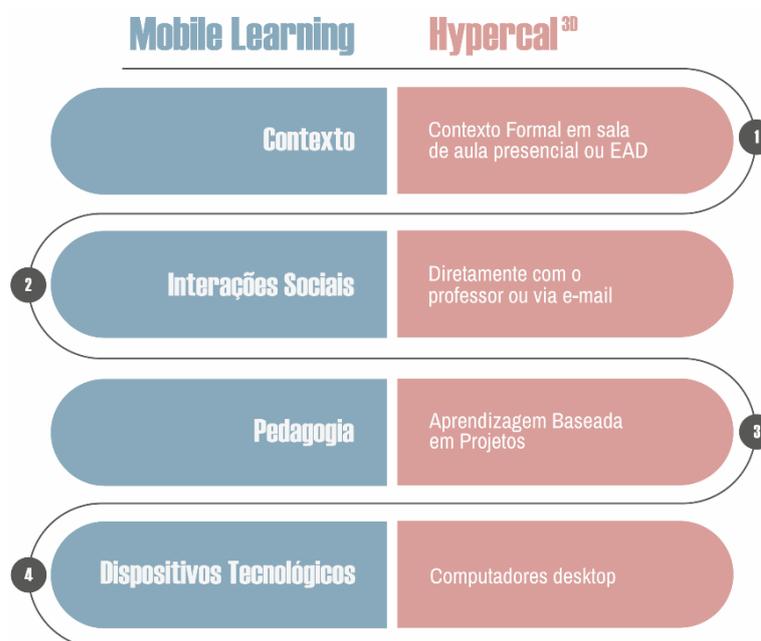
4.2. CRUZAMENTO TEÓRICO

Em um primeiro momento são apresentadas as características do *mobile learning* em conjunto com o HyperCAL^{3D}. Este cruzamento pretende compreender como o programa gráfico é utilizado atualmente e quais características precisam ser repensadas para a adaptação ao *mobile learning*.

4.2.1. *Mobile learning* e HyperCAL^{3D}

Como citado anteriormente, o *mobile learning* ainda não possui uma definição consensual e os estudos existentes utilizam abordagens diferentes, desta forma é difícil encontrar um fio condutor que una o *m-learning*. Portanto, para a realização desta pesquisa optou-se por partir de um quadro geral baseado nos construtos de Crompton (2013a): pedagogia, dispositivos tecnológicos, contexto e interações sociais. A Figura 30 representa um comparativo entre os construtos de *mobile learning* de Crompton (2013a) e seus correspondentes no HyperCAL^{3D}. Esta abordagem visa compreender quais construtos são contemplados pelo programa gráfico e quais necessitam alguma intervenção para adequar-se ao ensino móvel.

Figura 30 - Comparativo entre pilares do *mobile learning* e HyperCAL^{3D}



Fonte: a autora (2022)

Para Crompton (2013a) o contexto em *m-learning* pode ser formal, em sala de aula e/ou informal autogerenciado pelo estudante, como por exemplo é possível citar cursos de idiomas em aplicativos ou sites de cursos como Udemy. Por outro lado, a autora também defende a mutabilidade do contexto quando é avaliado a partir da

interação dos usuários com outros usuários, o ambiente no qual estão inseridos e as ferramentas do cotidiano (CROMPTOM, 2013b).

Como é possível perceber, a autora analisa o contexto por dois pontos de vista distintos. No primeiro, busca uma definição mais ampla centrada nas formas de “acesso” ao ensino. Na segunda abordagem, parece estar mais focada no dispositivo e nas formas de interação com ele. Interessante perceber, que a usabilidade também analisa o contexto de uso relacionado ao dispositivo, portanto o contexto como pilar constitutivo do *m-learning*, será analisado como formal ou informal. Este fato, coloca o HyperCAL ^{3D} em um contexto formal de ensino, com interações sociais realizadas em sala de aula diretamente com o professor e os demais colegas ou por e-mails e aplicativos de mensagens.

Com relação às pedagogias, Crompton (2013a) analisa a evolução pedagógica desde a década de 30, quando o ensino era unilateral e os alunos deveriam aprender os fatos sem questionar, até o desenvolvimento de pedagogias centradas no aluno, como é o caso da aprendizagem baseada em problemas. “Os alunos são incentivados a serem ativos em sua própria aprendizagem, a serem auto pensados e consumidores ativos de conhecimento” (CROMPTON, 2013a).

Como citado anteriormente, o grupo de pesquisa ViD da Universidade Federal do Rio Grande do Sul foi o responsável por utilizar a aprendizagem baseada em projetos no ensino de geometria descritiva, portanto o pilar construtivo da pedagogia de *m-learning* está totalmente alinhado com o HyperCAL ^{3D}.

Como último ponto, mas não menos importante construto do *mobile learning*, os dispositivos tecnológicos em especial nas universidades públicas brasileiras necessitam de investimentos governamentais para a compra dos equipamentos disponibilizados aos alunos em sala de aula. Por outro lado, de acordo com a pesquisa realizada durante a pandemia de COVID com os alunos dos cursos (FACULDADE DE ARQUITETURA, 2020), 92.3% dos estudantes que responderam ao questionário, possuem dispositivos móveis para acesso à internet. Desta forma, a adaptação para *mobile* do HyperCAL ^{3D} não apenas atende ao quarto construto como reduz a necessidade de recursos para as aulas de GD.

Como pode ser visto, o HyperCAL^{3D} consegue atender, a praticamente todos os pilares do *m-learning*, servindo de apoio ao ensino de GD, nas aulas da Universidade Federal do Rio Grande do Sul. Porém, há uma disparidade na utilização de recursos tecnológicos, pois se por um lado, os recursos para compra de equipamentos dependem do governo, por outro, há uma abundância de recursos não explorados que já pertencem aos alunos.

Como a pesquisa teórica deste relatório demonstra, os estudos sobre desenvolvimento de *m-learning* se concentram, em sua maioria, em modelos de aceitação das tecnologias, pois pesquisas indicam que os alunos tendem a não querer utilizar seus equipamentos em contextos de ensino. Estes modelos, em geral demonstram as necessidades e convicções dos usuários, mas não apresentam requisitos práticos para a criação do projeto.

O HyperCAL^{3D} vem ao longo dos anos passando por diversas atualizações, como forma de prover recursos às necessidades de alunos e professores. Adaptar o programa ao *mobile*, no entanto, requer compreender quais destas necessidades são fundamentais e podem ser adequadas às características dos dispositivos móveis. Para tanto, situar o programa nas fases do MLAM (ALMAIAH, ALAMRI e AL-RAHMI, 2019) e encontrar os construtos para a aceitação, é necessário para o direcionamento da adaptação do HyperCAL^{3D} ao *mobile learning*.

De acordo com o modelo MLAM, entender em que fase da utilização do aplicativo o ensino será utilizado é de fundamental importância para a compreensão de quais fatores são efetivamente relevantes na aceitação da tecnologia. A seguir é apresentado o enquadramento do HyperCAL^{3D} nas fases descritas no modelo.

- **Fase estáticas (MLA-S):** é utilizada na busca por informações, poderia ser exemplificada como o site de UFRGS, pois não é necessário realizar login, para receber notícias da Universidade, acessar o calendário acadêmico, fazer pesquisas na biblioteca etc.
- **Fase de interação (MLA-I):** requer troca de informações entre o aluno e, por exemplo um professor, portanto podem ser considerados como representantes da MLA-I e-mail pessoal ou o e-mail institucional da UFRGS.

- **Fase de transação (MLA-T):** permite que o aluno interaja diretamente com as ferramentas em tempo real a partir de cálculos dinâmicos, manipulação de dados e buscas. Assim, requer um sistema mais robusto, este fato coloca o HyperCAL ^{3D} nesta fase.

De acordo com o modelo MLAM, os construtos mais significativos na fase de transação são: Compatibilidade Percebida (PC), Consciência Percebida (PA), Segurança Percebida (PS), Benefício Funcional Percebido (PFB), Incerteza Percebida (PU) e Capacidade de Uso Percebida (PATU).

Quadro 7 - Construtos da fase de transação destacados devido ao grau de adoção da aceitação

Construtor	Definições conceituais	Hipóteses
Compatibilidade Percebida (PC)	O grau em que um aplicativo de aprendizagem móvel é percebido como consistente com as necessidades e percepções dos usuários potenciais.	Compatibilidade Percebida (PCM) tem uma relação significativa com a adoção de aprendizagem móvel.
Consciência Percebida (PA)	O grau de consciência dos usuários através da aquisição de conhecimentos é suficiente para aprender as características do aplicativo de aprendizagem móvel, usá-lo com habilidade e realizar suas funções, vantagens e desvantagens.	A Consciência Percebida (PA) tem uma relação significativa com a adoção da aprendizagem móvel
Segurança Percebida (PS)	O grau em que os usuários percebem que o nível de privacidade de dados e integridade de dados é eficiente e garante segurança para todas as transações eletrônicas e autenticação de identidade on-line via aplicativo de aprendizagem móvel.	A Segurança Percebida tem uma relação significativa com a confiança percebida (PA)
Benefício Funcional Percebido (PFB)	O grau em que os usuários percebem os benefícios funcionais gerais, incluindo custo, tempo, eficiência e eficácia do uso do aplicativo de aprendizagem móvel.	Benefício Funcional Percebido (BPF) tem relação significativa com a adoção da aprendizagem móvel
Incerteza Percebida (PU)	O grau em que os usuários percebem o risco nas transações devido a situações incontroláveis e desconhecidas no ambiente virtual associada ao aplicativo de aprendizagem móvel	A incerteza percebida tem uma relação significativa com a confiança percebida da aprendizagem móvel
Capacidade de uso percebida (PATU)	O grau em que um usuário percebe sua competência e capacidade confortável para usar o aplicativo de aprendizagem móvel tecnologicamente, organizacional e psicologicamente que correspondem aos valores, necessidades sociais e atitudes gerais do indivíduo.	A Capacidade de uso percebida (PATU) tem uma relação significativa com a adoção da aprendizagem móvel

Fonte: Almaiah, Alamri e Al-Rahmi (2019) tradução da autora

É importante ressaltar, que cada construto apresenta também algumas hipóteses que os ligam e/ou o tornam mais ou menos significativos na adoção da aprendizagem móvel. Este fato é relevante, pois o HyperCAL^{3D} já se provou no tempo como um programa fundamental para o ensino de GD na UFRGS. Além disso, questões de segurança não serão abordadas inicialmente, pois não será necessário o cadastro dos usuários e inclusão de informações pessoais. Portanto, neste trabalho, serão considerados apenas os construtos ligados diretamente a adoção da aprendizagem móvel. O Quadro 7 apresenta na cor verde os construtos que apresentam hipóteses com relação significativa a adoção de aprendizagem móvel e em branco os construtos ligados a segurança que não foram utilizados.

Por este motivo, são considerados apenas os construtos: Compatibilidade Percebida (PC), Consciência Percebida (PA), Benefício Funcional Percebido (PFB), e Capacidade de Uso Percebida (PATU). A Figura 31 apresenta em destaque o enquadramento do HyperCAL^{3D} na fase do modelo MLAM e os construtos pertinentes à sua aceitação.

Figura 31 - Fase do modelo MLAM e construtos utilizados



Fonte: a autora (2022)

Como citado anteriormente, não serão utilizados todos os recursos existentes na versão *desktop* para a versão móvel. Além disso, esta pesquisa possui como foco a criação da *viewport* 3D e *épura* do *software*, portanto para a busca de requisitos em aplicativos similares são levadas em consideração apenas características ligadas as *views* de aplicativos 2D e 3D.

A seguir, são alinhados os construtos do modelo MLAM considerados relevantes para este trabalho com as heurísticas SMART para a definição de requisitos para a análise de aplicações similares.

4.2.2. Modelo MLAM e Usabilidade SMART

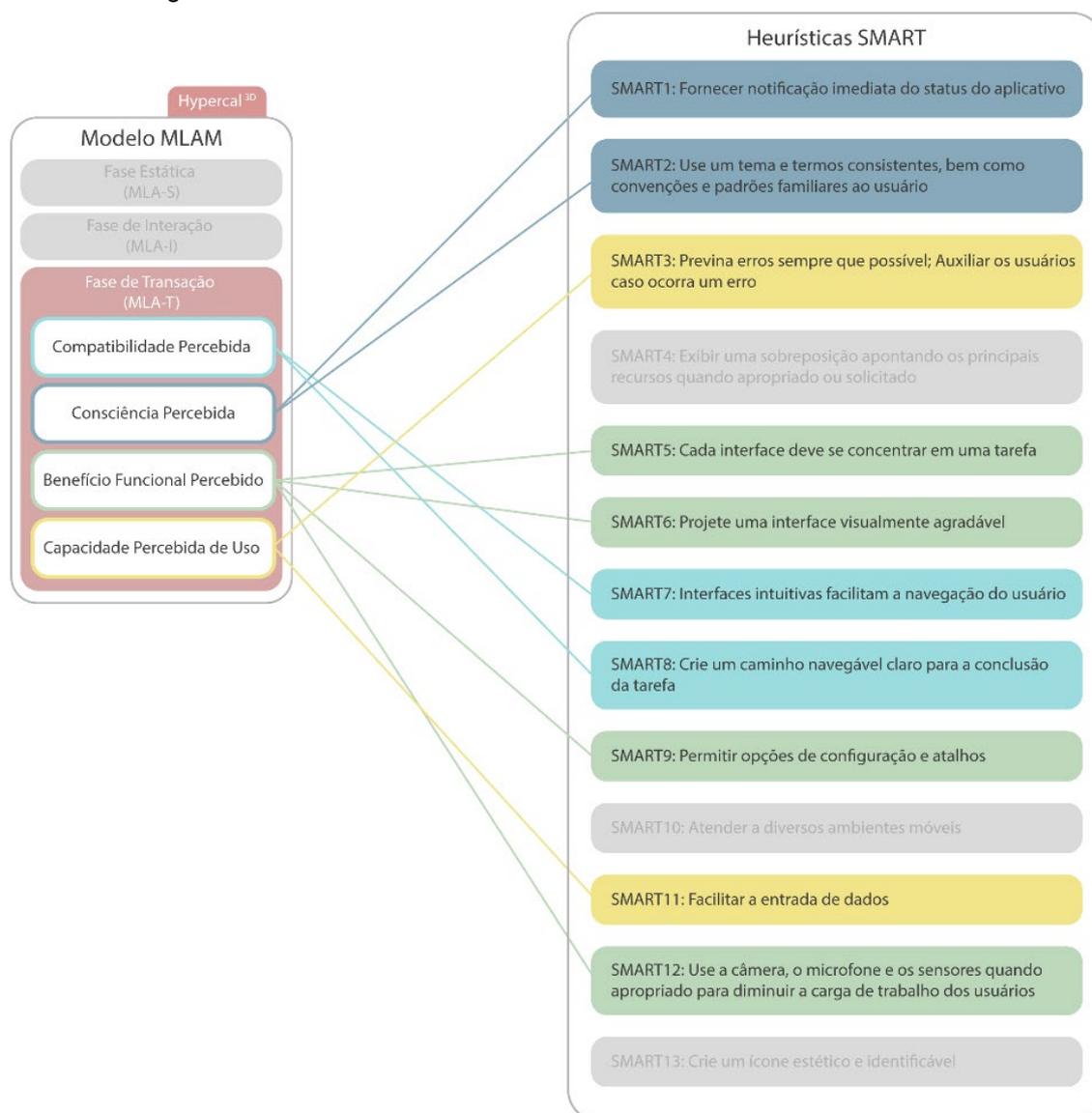
Neste ponto, são traçados paralelos entre o modelo MLAM e as heurísticas de usabilidade para dispositivos móveis desenvolvidos no modelo SMART, levando em consideração as características da *viewport* 3D e *épura* encontradas no HyperCAL^{3D} *desktop* que serão utilizadas na adaptação *mobile* descritas abaixo.

Viewport 3D: tem por finalidade demonstrar os sólidos criados, seus vértices, arestas e planos, assim como suas projeções no sistema de referência. Também é possível interagir com a *view* com movimentos de rotação em três dimensões e *pan*, assim como a criação de novos sistemas de referência e seleção de arestas e planos.

Épura: apresenta as projeções planificadas do sólido, pontos, planos e arestas em tempo real à medida que são criados nos sistemas de referência existentes ou em novos sistemas enquanto são posicionados na *viewport* 3D ou nela mesma. Também apresenta movimentos de rotação bidimensional e *pan*.

Em se tratando de dispositivos móveis, alguns atributos devem ser levados em consideração: o tamanho da tela, manipulação via *touch screen*, capacidade de processamento e conectividade. Estas particularidades afetam diretamente a área de visualização, a manipulação dos elementos, o desempenho da aplicação e o tamanho final da aplicação e dos arquivos gerados, respectivamente. Portanto, estes aspectos são levados em consideração para a realização do cruzamento teórico a seguir. A Figura 32 apresenta um esquema gráfico do cruzamento dos construtos de MLAM relevantes a adaptação para *mobile learning* do programa gráfico HyperCAL^{3D}, com as heurísticas de usabilidade SMART. As explicações referentes a estes cruzamentos são descritas na sequência.

Figura 32 - Cruzamento entre o modelo MLAM e as Heurísticas SMART



Fonte: a autora (2022)

Compatibilidade Percebida (PC)

É descrita no modelo MLAM como o grau em que um aplicativo de aprendizagem móvel é percebido como consistente com as necessidades e percepções dos usuários potenciais (ALMAIAH, ALAMRI e AL-RAHMI, 2019).

O HyperCAL^{3D}, como já citado, é uma ferramenta de apoio ao ensino aprendizagem de GD dos cursos de Design, Engenharia e Arquitetura da UFRGS que possuem a aprendizagem baseada em projetos como base pedagógica. O *software* auxilia no ensino de conceitos difíceis de serem trabalhados em sala de aula, por métodos tradicionais, como: modelagem de sólidos, sistema de projeção, projeções

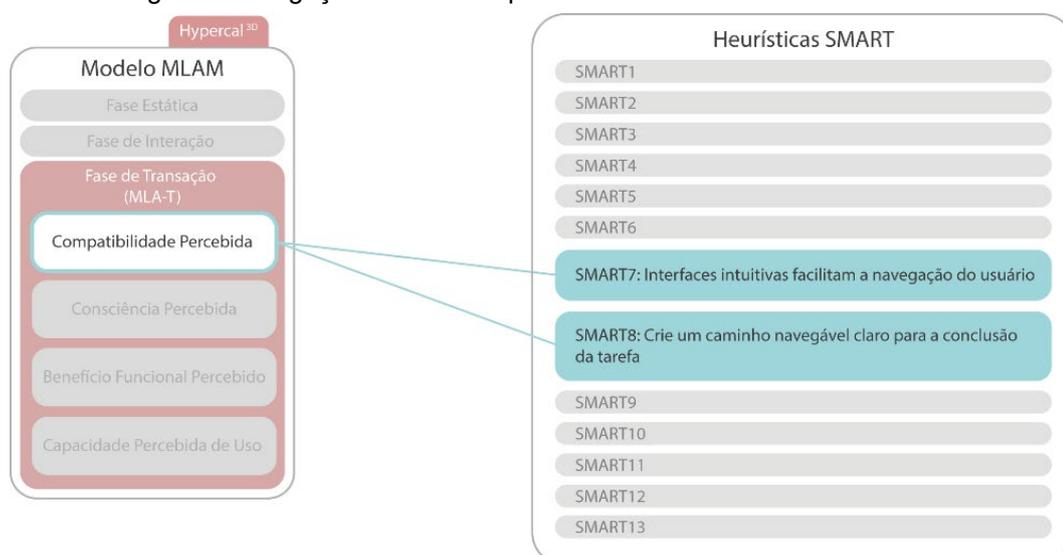
acumuladas, mudanças de sistema de referência primária e mudanças de sistema de referência sucessivas (SANTOS, 2016).

Assim, o programa já se consolidou como compatível com as necessidades dos usuários. No entanto, a visualização em telas pequenas, as dificuldades na navegabilidade e interação das *views*, devido à falta de periféricos, como teclado e mouse e a necessidade de segurar o celular quando manipulado, podem causar desconforto com a versão *mobile* levando o usuário a entender o app como não compatível com suas necessidades.

Portanto, para a versão móvel se manter consistente com as necessidades do usuário é necessário analisar questões referentes a *layout* de telas e as formas de interação, em especial para a *viewport* 3D, pois, em geral, as rotações de objetos tridimensionais são baseadas em dois pontos: posição da câmera em relação ao *target* e *target* em relação ao objeto. Ao associar a rotação com a possibilidade de pan, o *target* é facilmente deslocado do solido, e desta forma perdido em uma segunda rotação. Este fato, aliado à utilização do *touch screen*, pode dificultar a navegação no aplicativo móvel e provocar o sentimento de incompatibilidade.

As Heurísticas SMART7 e SMART8 falam diretamente às questões de navegabilidade e podem ser avaliadas em aplicativos similares, como forma de compreender como os desenvolvedores lidaram com estes “problemas”. A Figura 33 apresenta a ligação entre o construto Compatibilidade Percebida do modelo MLAM e as Heurísticas SMART7 e SMART8.

Figura 33 – Ligações entre Compatibilidade Percebida e SMATS 7 e 8



Fonte: a autora (2022)

SMART7: Interfaces intuitivas facilitam a navegação do usuário – As interfaces móveis devem ser fáceis de aprender, por meio das quais os próximos passos são óbvios. Isso permite que os usuários completem mais facilmente suas tarefas (JOYCE e LILLEY, 2014).

Além da interface, as telas *touch* apresentam uma série de gestos de navegação que são amplamente utilizados em diferentes aplicações. Por outro lado, com exceção dos jogos que possuem uma navegabilidade própria, não é comum aplicações que utilizam três dimensões. Desta forma, a análise de similares sobre a utilização de gestos na navegação pode prover informações importantes para o desenvolvimento do HyperCAL^{3D} *mobile*.

SMART8: Crie um caminho navegável claro para a conclusão da tarefa – os usuários devem ser capazes de ver imediatamente como eles podem interagir com o aplicativo e navegar em seu caminho para a conclusão da tarefa (JOYCE e LILLEY, 2014).

Em geral os gestos utilizados nas interfaces *touch* simulam situações cotidianas, como por exemplo passar imagens arrastando para o lado ou para cima. No entanto, em aplicações 3D e 2D a navegação acontece no mesmo local de edição do objeto, portanto é necessário identificar como aplicativos similares lidam de forma fluida com a constante alteração entre edição do objeto e navegação na *view*, necessárias a realização das tarefas.

Portanto, o primeiro critério de análise dos aplicativos similares é a utilização dos gestos. Para tanto, foi desenvolvida a seguinte questão de análise:

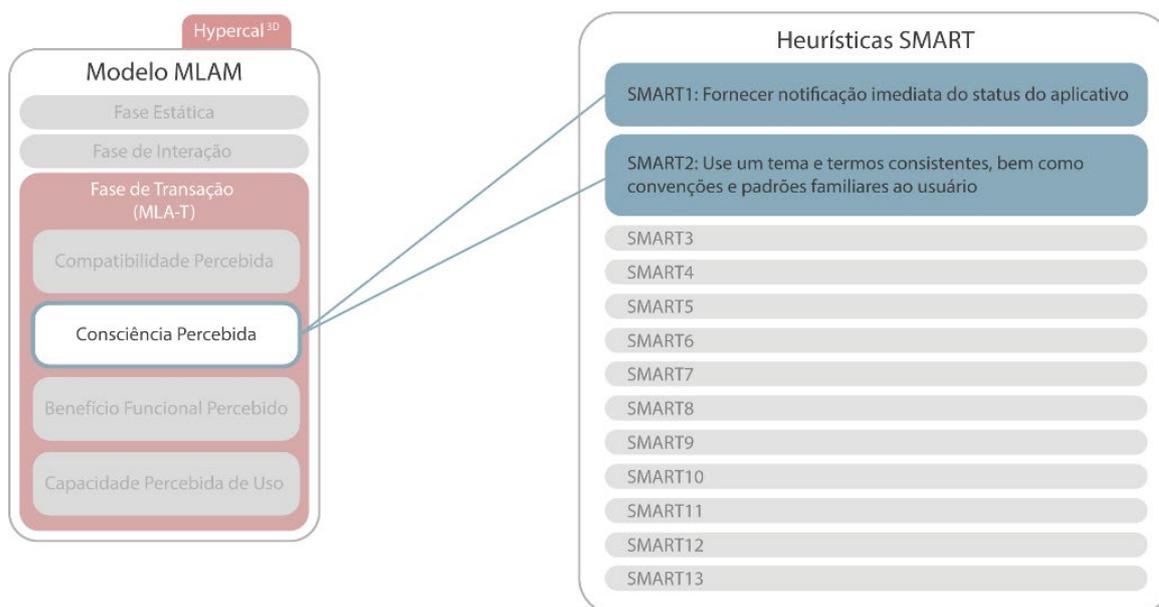
Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)?

Consciência Percebida (PA)

O grau de consciência dos usuários através da aquisição de conhecimentos é suficiente para aprender as características do aplicativo de aprendizagem móvel, usá-lo com habilidade e realizar suas funções, vantagens e desvantagens (ALMAIAH, ALAMRI e AL-RAHMI, 2019).

Este construto tem a ver com a capacidade de compreensão da tecnologia e, portanto, heurísticas voltadas a aprendizagem dos recursos podem auxiliar na aceitação do aplicativo. As Heurísticas SMART1 e SMART2, têm por objetivo facilitar a utilização do aplicativo, desta forma podem auxiliar usuário a compreender suas funcionalidades de forma dinâmica. Apesar da Heurística SMART 7 também poder ser utilizada para este construto, como já foi utilizada no anterior não será necessária sua utilização novamente. A Figura 34 apresenta as ligações entre a Consciência Percebida e as SMARTS 1 e 2.

Figura 34 - Ligações entre Consciência Percebida e SMARTS 1 e 2



Fonte: a autora (2022)

SMART1: Fornecer notificação imediata do status do aplicativo – Certifique-se de que o usuário do aplicativo móvel seja informado do status do aplicativo imediatamente e enquanto for necessário. Se for o caso, faça isso de forma não intrusiva, como exibir notificações dentro da barra de status (JOYCE e LILLEY, 2014).

Diretamente ligada ao ensino da tecnologia, esta heurística pode auxiliar na busca por soluções em aplicações similares de recursos utilizados para as notificações de forma não invasiva.

SMART2: Use um tema e termos consistentes, bem como convenções e padrões familiares ao usuário – Use um tema para o aplicativo móvel para garantir que diferentes telas se pareçam. Crie também um guia de estilo a partir do qual

palavras, frases e conceitos familiares ao usuário são aplicados consistentemente em toda a interface, usando uma ordem natural e lógica. Use convenções e padrões de plataforma que os usuários esperam em um aplicativo móvel, como os mesmos efeitos quando os gestos são usados (JOYCE e LILLEY, 2014).

O HyperCAL ^{3D} desktop já possui uma série de convenções de cores e padrões que serão mantidos na versão móvel, no entanto a análise de similares pode indicar padrões necessários à compreensão da navegação móvel que não necessitam ser utilizados na versão *desktop*. Assim, para a análise de similares é utilizada a pergunta:

O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?

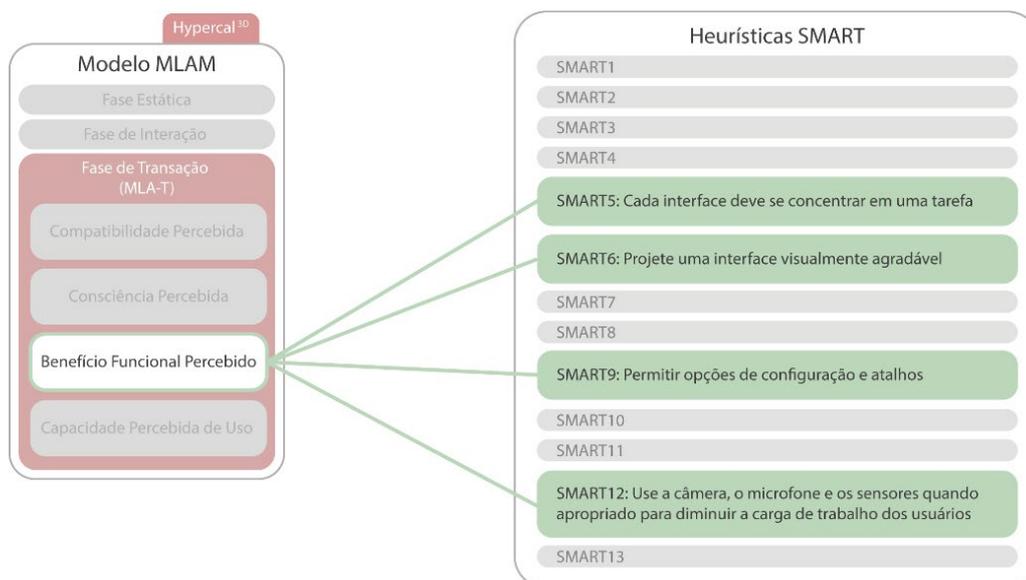
Benefício Funcional Percebido (PFB): o grau em que os usuários percebem os benefícios funcionais gerais, incluindo custo, tempo, eficiência e eficácia do uso do aplicativo de aprendizagem móvel, em vez de usar funções tradicionais de escritório físico (ALMAIAH, ALAMRI e AL-RAHMI, 2019).

Para analisar este construto, é considerada a quantidade de recursos existentes na UFRGS e a disponibilidade aos alunos de seus próprios dispositivos móveis. Como forma de aproveitar estes recursos, além da disponibilização do HyperCAL ^{3D} para *mobile* há a necessidade de aceitação dos estudantes em utilizar seus aparelhos pessoais para o ensino de GD.

Portanto, é necessário tornar o aplicativo interessante a ponto do aluno, na falta de um computador disponível, em vez de se juntar em duplas, optar por utilizar a versão móvel. Vale ressaltar, que a versão móvel não apresentará todos os recursos disponíveis na versão *desktop* e por esse motivo será útil em determinadas atividades para desafogar a utilização dos computadores da universidade em algumas tarefas, mas não no total da utilização do programa.

Este construto fala diretamente sobre as facilitações de uso, que permitem agilidade nas execuções de tarefas por parte do usuário. As heurísticas SMART5, SMART6, SMART9 e SMART12, buscam auxiliar o usuário na agilidade de uso do aplicativo. Como pode ser observado na Figura 35.

Figura 35 - Ligações entre Benefício Funcional Percebido e SMARTS 5, 6, 9 e 12



Fonte: a autora (2022)

SMART5: Cada interface deve se concentrar em uma tarefa – Estar focando em uma tarefa garante que as interfaces móveis sejam menos desordenadas e simples a ponto de ter apenas os elementos necessários na tela para completar essa tarefa. Isso também permite que a interface seja visível para usuários que são interrompidos com frequência (JOYCE e LILLEY, 2014).

Com a utilização de seus próprios dispositivos, os estudantes podem se sentir tentados a abrir notificações de outros aplicativos, desta forma a SMART 5 busca fixar o usuário na tarefa que está sendo realizada.

SMART6: Projete uma interface visualmente agradável – Interfaces móveis que são atraentes são muito mais memoráveis e, portanto, são usadas com mais frequência. Os usuários também são mais tolerantes com interfaces atraentes (JOYCE e LILLEY, 2014).

Durante a análise de similares, é importante perceber quais recursos tornam suas interfaces atraentes de forma a instigar o usuário a utilização.

SMART9: Permitir opções de configuração e atalhos – Dependendo do usuário-alvo, o aplicativo móvel pode permitir opções de configuração e atalhos para as informações mais importantes e tarefas frequentes, incluindo a capacidade de configurar de acordo com contextuais necessidades.

Neste caso, os atalhos não são como os de computadores, baseados em teclas e sim formas de acesso facilitado a ferramentas muito utilizadas. Analisar como os aplicativos similares lidam com a criação de atalhos pode auxiliar na confecção do HyperCAL ^{3D} *mobile*.

SMART12: Use a câmera, o microfone e os sensores quando apropriado para diminuir a carga de trabalho dos usuários – Considere o uso da câmera, microfone e sensores para diminuir a carga de trabalho dos usuários. Por exemplo, usando GPS para que o usuário saiba onde está e como ir até determinado local ou usando OCR e a câmera para capturar digitalmente as informações que o usuário precisa inserir, permitindo o uso do microfone para inserir conteúdo em vez de digitar no teclado pequeno (JOYCE e LILLEY, 2014).

Recursos como sensores, mobilidade de câmera e microfone são encontrados apenas em dispositivos móveis, portanto entender como os programas similares utilizam estes recursos pode indicar uma tendência para a criação de aplicações móveis.

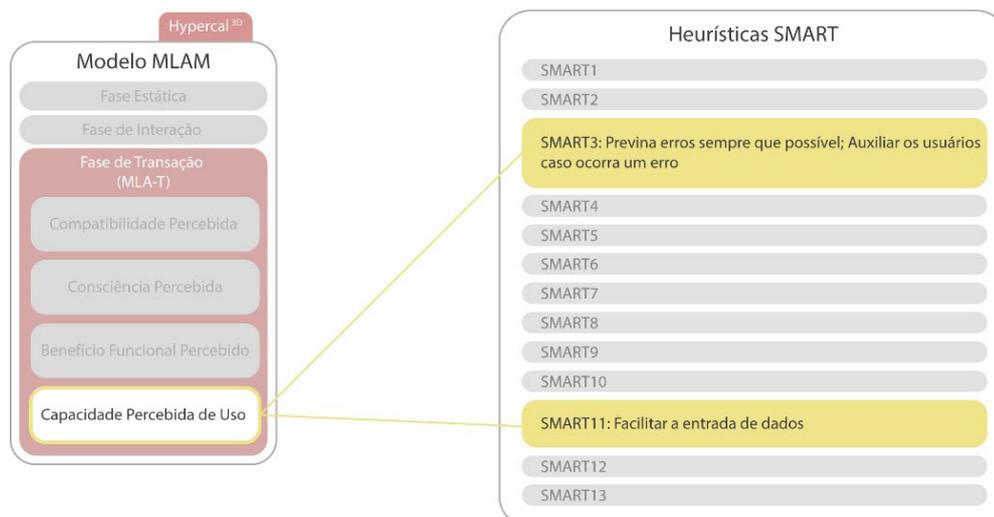
Este ponto, fala diretamente a facilitação de uso de forma a tornar o aplicativo mais versátil, dinâmico e interessante para evitar que possíveis distrações possam chamar a atenção do usuário. Desta forma, foi elaborada a seguinte pergunta:

Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?

Capacidade de Uso Percebida (PATU): o grau em que um usuário percebe sua competência e habilidade confortável para usar o aplicativo de aprendizagem móvel tecnologicamente, organizacional e psicologicamente que correspondem aos valores, necessidades sociais e atitudes gerais do indivíduo (ALMAIAH, ALAMRI e AL-RAHMI, 2019).

A prevenção de erros e a facilitação do uso do aplicativo podem significar uma melhor compreensão do usuário quanto a sua própria habilidade durante o uso. Portanto, as Heurísticas SMART 3 e SMART 11 podem auxiliar na aceitação deste construto. A Figura 36 ilustra as ligações.

Figura 36 - Ligações entre Capacidade de Uso Percebida e SMARTS 3 e11



Fonte: a autora (2022)

SMART3: Previna erros sempre que possível; Auxiliar os usuários caso ocorra um erro – Certifique-se de que o aplicativo móvel seja à prova de erros o máximo possível. Caso ocorra um erro, deixe o usuário saber qual é o erro de uma forma que ele entenderá e ofereça conselhos sobre como ele pode corrigir o erro ou proceder de outra forma.

Analisar como aplicativos similares lidam com os erros e/ou os evitam, pode auxiliar na criação do HyperCAL^{3D} *mobile*, pois permitirão que o usuário compreenda que é capaz de utilizar o aplicativo.

SMART11: Facilitar a entrada de dados – Dispositivos móveis são difíceis de usar do ponto de vista de entrada de conteúdos. Certifique-se de que os usuários podem inserir conteúdos com mais facilidade e precisão, por exemplo, exibindo botões de teclado que são tão grandes quanto possível, além de permitir entrada multimodal e mantendo os campos de formulário ao mínimo.

Entender como os aplicativos 3D utilizam a entrada de dados pode contribuir com a criação de soluções eficientes para o HyperCAL^{3D} *mobile*.

Para contemplar este ponto foi estruturada a seguinte pergunta para a análise de aplicativos similares:

O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

Como apresentado, não foram utilizadas todas as Heurísticas SMART, somente aquelas que apresentam ligação com os construtos identificados como relevantes para a aceitação do HyperCAL^{3D} *mobile*. A seguir são apresentadas as análises realizadas nos aplicativos similares.

4.3. IDENTIFICAÇÃO DOS ARTEFATOS E CONFIGURAÇÃO DAS CLASSES DE PROBLEMAS

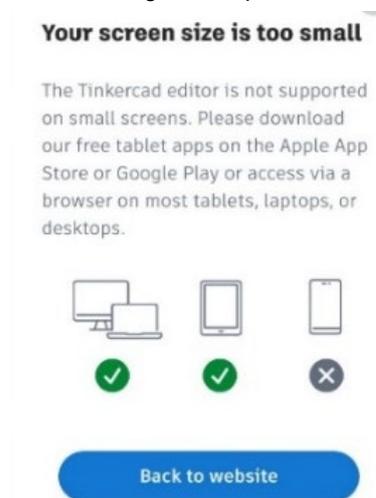
Como citado no item 3.4, não foram encontrados aplicativos *m-learning* sobre geometria descritiva. Por esse motivo, foi desenvolvida uma busca por aplicativos de modelagem 3D e desenho técnico que apresentam similaridades com a *viewport* 3D e *épura* do HyperCAL^{3D}.

Foram encontrados ao todo 45 aplicativos (Apêndice A) com o termo 3D *modeling*. Ao final da filtragem restaram nove destinados a modelagem e edição de objetos 3D: 3D Modeling App, OnShap 3D CAD, Wuweido, Qubism 3D modeling, Tinkercad, SDF 3D, 3d, EDS 3D Modeling tool, Tetra – Easy 3D Creation e Sketch 3D. Mesmo selecionados na primeira etapa, alguns aplicativos não puderam ser analisados ou foram dispensados pelos motivos descritos em sequência:

Sketch 3D: o aplicativo não possui uma versão gratuita e apresenta uma avaliação negativa sobre a falta de ferramentas para modelagem, portanto o investimento no mesmo foi desconsiderado.

Tinkercard: apresentou a mensagem: “o tamanho de sua tela é muito pequeno”, como mostra a Figura 37 e por este motivo foi dispensado da análise.

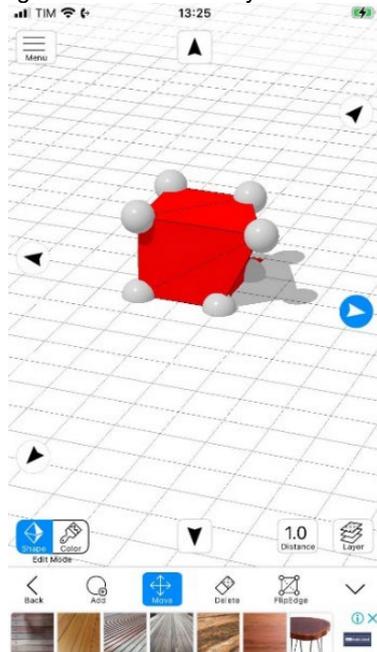
Figura 37 - Mensagem do aplicativo Tinkercard



Fonte: Tinkercard ([s.d.]).

Tetra – Easy 3D Creation: a proposta do aplicativo é diferente do HyperCAL ^{3D}, mesmo possuindo a possibilidade de mover os vértices, não permite a movimentação, rotação, escalonamento do objeto como um todo, além de ser de difícil manipulação e todos os objetos partirem de um cubo (Figura 38).

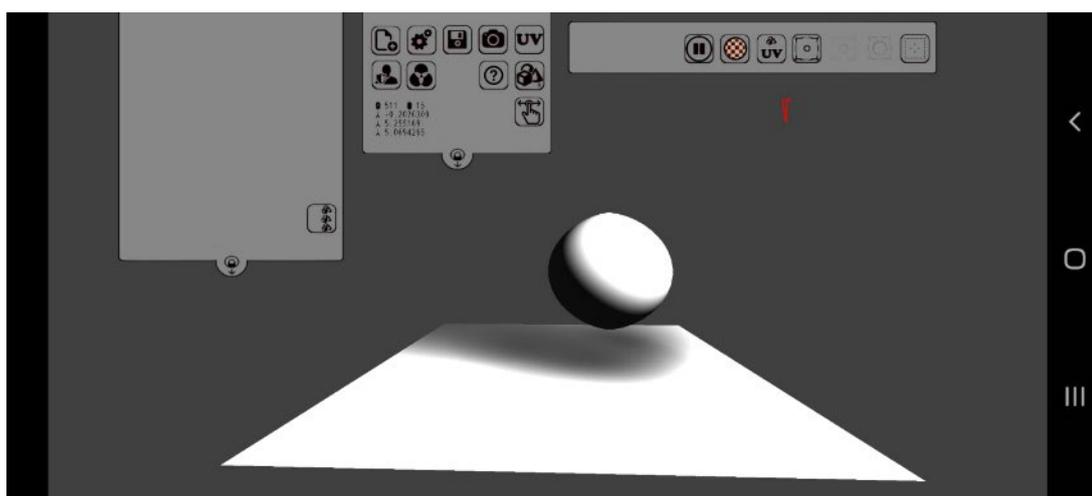
Figura 38 - Tetra – Easy 3D Creation



Fonte: Tetra – Easy 3D Creation ([s.d.]).

EDS 3D Modeling tool: o aplicativo não apresenta instruções de uso e não é intuitivo. Devido a estas dificuldades de compreensão das ferramentas foi desconsiderado. A Figura 39 apresenta o aplicativo.

Figura 39 - EDS 3D Modeling tool



Fonte: EDS 3D Modeling tool ([s.d.]).

Na segunda busca, foram encontrados 41 aplicativos, dos quais oito possuem a finalidade de desenho CAD, são eles: AutoCAD, UVCAD, CAD Desenhe um

desenho mais 3D, OnShap 3D CAD, GnaCAD, CorelCAD Mobile e FingerCAD. Alguns aplicativos não puderam ser analisados ou foram dispensados pelos motivos descritos em sequência:

CorelCAD Mobile: A versão de avaliação apresenta as ferramentas de desenho desativadas.

OnShap 3D CAD: foi analisado no item modelagem 3D.

CAD Desenhe um desenho mais 3D: apesar de o aplicativo não ser intuitivo, possui um vídeo explicativo no youtube. No entanto, sua usabilidade é precária, pois necessita de várias etapas para a criação e edição de linhas e formas, o que atrapalha a fluência de criação e edição. As interações são feitas primordialmente por botões deixando as interações via gestos em segundo plano. Desta forma sua análise não acrescenta parâmetros significativos a pesquisa.

FingerCAD: o aplicativo não possui uma versão de avaliação portanto foi dispensado para a análise.

Após a análise dos aplicativos, foi realizada uma busca no Google por 3D *modeling on-line*, entre os diferentes sites que surgiram, alguns se destacaram quando abertos no computador com: vectary e selfcad. No entanto, mesmo utilizando o mesmo navegador no celular, ambos apresentaram diversos problemas de adaptação a tela e impossibilidades de utilização dos recursos via *touch*. Este fato inviabilizou a análise de ambos. Além destes, mesmo não tendo surgido na pesquisa, o editor da biblioteca Three.js foi submetido à análise.

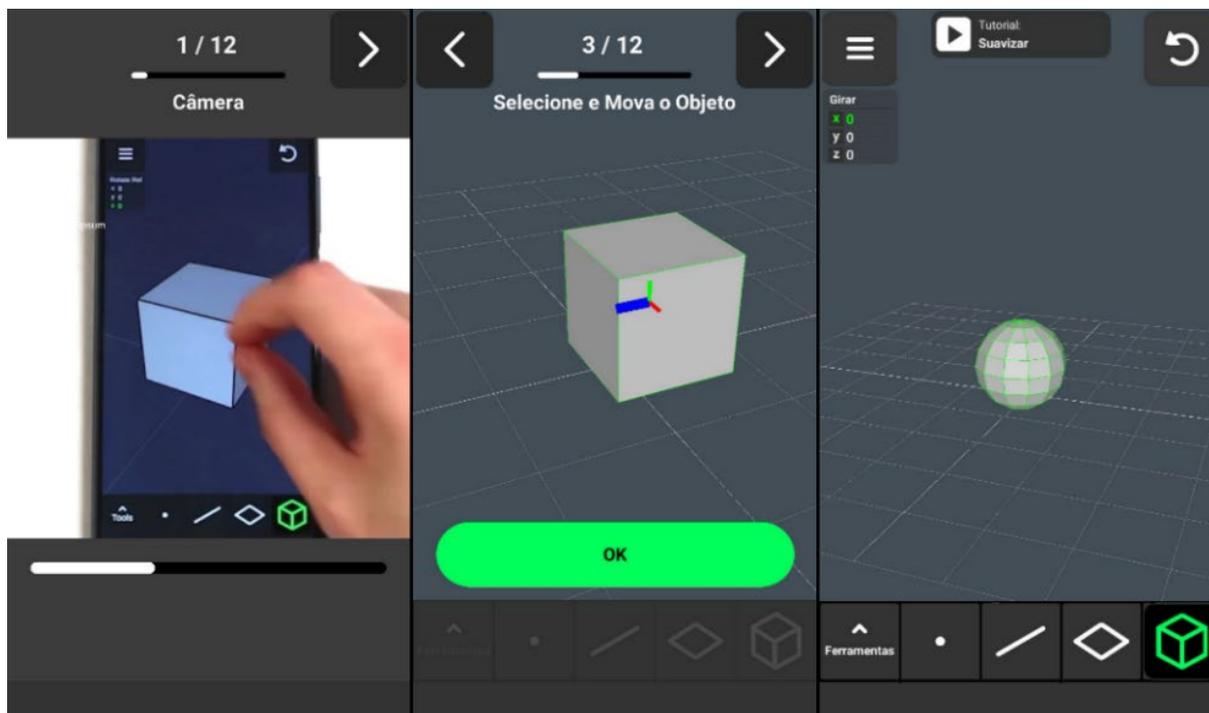
A seguir são apresentadas as análises dos aplicativos encontrados iniciando pelos tridimensionais.

Aplicativos tridimensionais analisados são: 3D Modeling App, OnShap 3D CAD, Wuweido, Qubism 3D Modeling, SDF 3D e 3d. Para a diferenciar a interface gestual da interface gráfica, todos os gestos são representados por desenhos de mãos, enquanto a interface gráfica foi descrita na forma de ferramentas, como por exemplo: ferramenta mover. Os gráficos que são utilizados como eixos sobre os objetos são chamados de *gizmos*.

4.3.1. Análise do aplicativo 3D Modeling App

Desenvolvido pela companhia Core five, Inc., não é um aplicativo CAD, porém apresenta ferramentas para inserção de medidas. Apresenta a possibilidade de utilizar a câmera ortográfica e perspectiva. Possui uma versão de avaliação de 7 dias.

Figura 40 – Tutorial 3D Modeling app



Fonte: 3D Modeling App ([s.d.]).

O aplicativo possui vídeos tutoriais (Figura 40) de todas as ferramentas, que são apresentadas na primeira visualização e podem ser revistas a após a seleção da ferramenta com o clique no botão correspondente que aparece no topo da tela.

Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)?

O Quadro 8 apresenta os gestos utilizados no 3D Modeling app. O aplicativo apresenta também ferramentas de escultura e pintura, além de outras que não possuem relação com o HyperCAL^{3D} e, portanto, não foram consideradas.

Quadro 8 - Navegação, seleção, desseleção e edição 3D Modeling App

<p>Navegação: os gestos de navegação são realizados por meio de arraste dos dedos na tela, um dedo para rotação, dois dedos para pan e dois dedos em pinça para zoom. Além destes, foi possível observar que o arraste em rotação com dois dedos permite rotacionar ao redor do eixo vertical.</p>	<p style="text-align: center;">Navegação 3D Modeling App</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Rotação</p>  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>Pan</p>  <p>ARRASTE DUPLO</p> </div> <div style="text-align: center;"> <p>Zoom/ rotação</p>  <p>ARRASTE PINÇA</p> </div> </div>
<p>Seleção: É feita com um toque simples para selecionar um objeto e tocar e segurar a partir do segundo objeto. Para selecionar arestas, vértices e planos é necessário acionar os botões correspondentes e tocar sobre os elementos desejados. Também é possível selecionar em janela com tocar e segurar em seguida arrastando para a seleção.</p>	<p style="text-align: center;">Seleção 3D Modeling App</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Simples</p>  <p>TOQUE SIMPLES</p> </div> <div style="text-align: center;"> <p>Janela</p>  <p>TOCAR E ARRASTE SIMPLES SEGURAR</p> </div> </div> <div style="text-align: center; margin-top: 10px;"> <p>Multisseleção</p>  <p>TOQUE SIMPLES + TOCAR E SEGURAR</p> </div>
<p>Desselecionar: toque simples sobre o objeto já selecionado ou toque na área livre para desselecionar todos juntos.</p>	<p style="text-align: center;">Desselecionar 3d Modeling App</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Individual</p>  <p>TOQUE SIMPLES sobre o objeto</p> </div> <div style="text-align: center;"> <p>Total</p>  <p>TOQUE SIMPLES área livre</p> </div> </div>
<p>Editar: após a seleção do objeto ou de alguma de suas partes é possível, mover com arraste sobre ele; rotacionar com o arraste de dois dedos em rotação e escalonar em todos os eixos ao mesmo tempo com arraste em pinça, em apenas um eixo arrastar com dois dedos em paralelo.</p>	<p style="text-align: center;">Edição 3D Modeling App</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Mover</p>  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>Rotacionar</p>  <p>ARRASTE DE DOIS DEDOS COM ROTAÇÃO</p> </div> <div style="text-align: center;"> <p>Escalonar</p>  <p>ARRASTE PINÇA</p> </div> </div>

Fonte: a autora (2022)

O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?

Feedbacks: as arestas do objeto mudam para a cor verde quando selecionadas, assim como os vértices e planos em caso de seleção destes. Durante a edição, um *gizmo*¹¹ surge no ponto de origem do objeto e se deforma ao longo do eixo que está sendo manipulado. As edições mover, rotacionar e escalonar acontecem em tempo real, sem a utilização de outros indicativos de posição, rotação e escalonamento iniciais, apenas com a mudança do estado do modelo.

¹¹ Ferramenta gráfica que apresenta os eixos de rotação, movimentação e escala e permite a manipulação dos objetos nos mesmos eixos.

Temas e padrões: de forma geral o aplicativo utiliza as mesmas cores em toda a navegação, não apresenta muitos ícones, mas os que apresenta são comuns a outros aplicativos como: lixeira e desfazer. Utiliza palavras para identificar os botões.

Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?

Clareza, quantidade e qualidade de informações na tela: o aplicativo possui poucas informações na tela, a maior parte das ferramentas está “escondida” em menus acessíveis ao toque de um botão (Figura 41).

Figura 41 - menus escondidos 3D Modeling App



Fonte: 3D Modeling App ([s.d.]).

Atalhos: apresenta a possibilidade de edição do último parâmetro alterado, clicando-se sobre o botão abaixo do menu sanduiche no top esquerdo da tela.

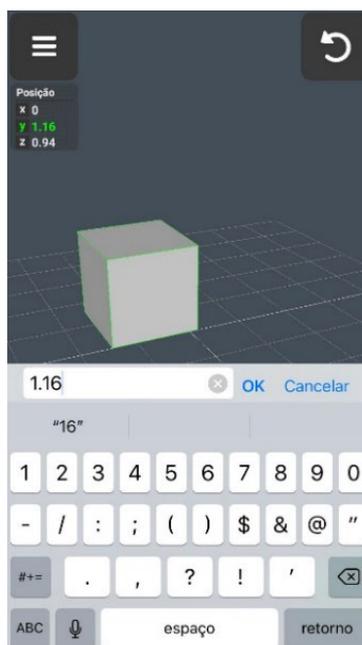
Utilização de sensores: não apresenta a utilização de sensores.

O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

Prevenção de erros: o app apresenta tutoriais introdutórios das ferramentas existentes, além de banner que indicam por exemplo: que a capacidade máxima de vértices foi atingida. Outro mecanismo de prevenção é a palavra multisseleção que surge no topo da tela quando o segundo objeto é selecionado.

Entrada de dados: é possível inserir coordenadas clicando no botão abaixo do menu sanduiche via teclado virtual (Figura 42). O teclado virtual apresenta um botão para acesso ao microfone que permite ditar o valor a ser inserido.

Figura 42 - Teclado Virtual 3D Modeling App

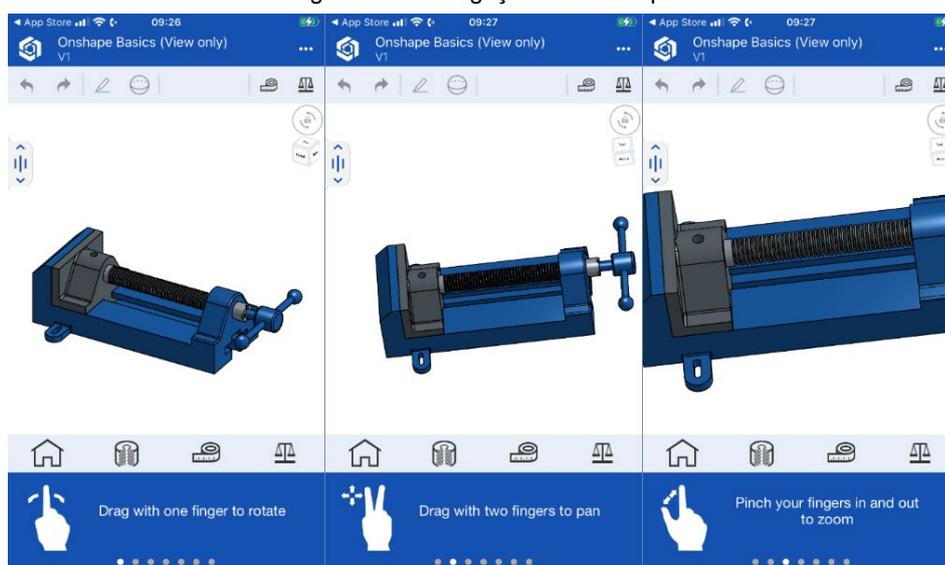


Fonte: 3D Modeling App ([s.d.]).

4.3.2. Análise do aplicativo OnShape 3D CAD

Desenvolvido pela companhia OnShape, pode ser utilizado no navegador de internet ou baixado para celular IOS ou Androide como um aplicativo. O *software* permite o trabalho compartilhado por meio de serviços em nuvem. Apresenta uma versão gratuita para estudantes.

Figura 43 - Navegação no OnShape



Fonte: OnShape ([s.d.]).

Na primeira utilização do aplicativo é oferecido ao usuário um tutorial dos gestos que podem ser utilizados para navegação e seleção de partes do sólido apresentado na tela. Assim, como demonstra a Figura 43.

Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)?

Além das ferramentas de edição descritas no Quadro 9, o aplicativo apresenta uma série de possibilidades de edição que não foram analisadas pois no HyperCAL ^{3D} não são utilizadas.

Quadro 9 - Navegação, seleção, desseleção e edição OnShape

<p>Navegação: os gestos de navegação são realizados por meio de arraste dos dedos na tela, um dedo para rotação, dois dedos para pan e dois dedos em pinça para zoom. Além destes, foi possível observar que o arraste em pinça permite rotacionar a view focando os objetos, que se encontram entre os dedos.</p>	<p style="text-align: center;">Mavegação OnShape</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Rotação</p>  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>Pan</p>  <p>ARRASTE DUPLA</p> </div> <div style="text-align: center;"> <p>Zoom/rotação</p>  <p>ARRASTE PINÇA</p> </div> </div>
<p>Seleção: pode ser feita com toque simples sobre o objeto ou para ativar a seleção de precisão é necessário, tocar e segurar, até o cursor mudar, e arrastar. Para seleção em janela basta tocar e segurar com dois dedos e, após o aparecimento da janela de seleção, arraste em pinça.</p>	<p style="text-align: center;">Seleção OnShape</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Simples</p>  <p>TOQUE SIMPLES</p> </div> <div style="text-align: center;"> <p>Precisão</p>  <p>TOCAR E SEGURAR ARRASTE SIMPLES</p> </div> </div> <div style="text-align: center; margin-top: 10px;"> <p>Janela</p>  <p>TOCAR E SEGURAR ARRASTE PINÇA DUPLA</p> </div>
<p>Desselecionar: para deselegonar um objeto, basta dar um toque sobre ele. Para deselegonar todos os objetos selecionados em conjunto, são necessários dois toques na área livre da view.</p>	<p style="text-align: center;">Desselecionar OnShape</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Individual</p>  <p>TOQUE SIMPLES</p> </div> <div style="text-align: center;"> <p>Total</p>  <p>DOIS TOQUES</p> </div> </div>
<p>Editar: após a seleção do objeto, é necessário acionar o comando de edição, que libera o botão de <i>transform</i> que permite mover, rotacionar e escalar a partir do arraste de alguma parte do <i>gizmo</i>.</p>	<p style="text-align: center;">Edição OnShape</p> <div style="display: flex; justify-content: space-around; align-items: center; margin-bottom: 10px;"> <div style="text-align: center;">  <p>FERRAMENTA EDIÇÃO</p> </div> <div style="text-align: center;"> <p>+</p> </div> <div style="text-align: center;">  <p>FERRAMENTA TRANSFORMAÇÃO</p> </div> <div style="text-align: center;"> <p>+</p> </div> </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>GIZMO</p> </div> <div style="text-align: center;"> <p>+</p> </div> <div style="text-align: center;">  <p>ARRASTE SIMPLES</p> </div> </div>

Fonte: a autora (2022)

O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?

Feedbacks: na view é possível observar a mudança de cor nos elementos selecionados. Durante a edição com o *gizmo*, são mantidos na tela dois objetos iguais com intensidades de cor diferentes, o menos intenso indica o *status* antes da alteração

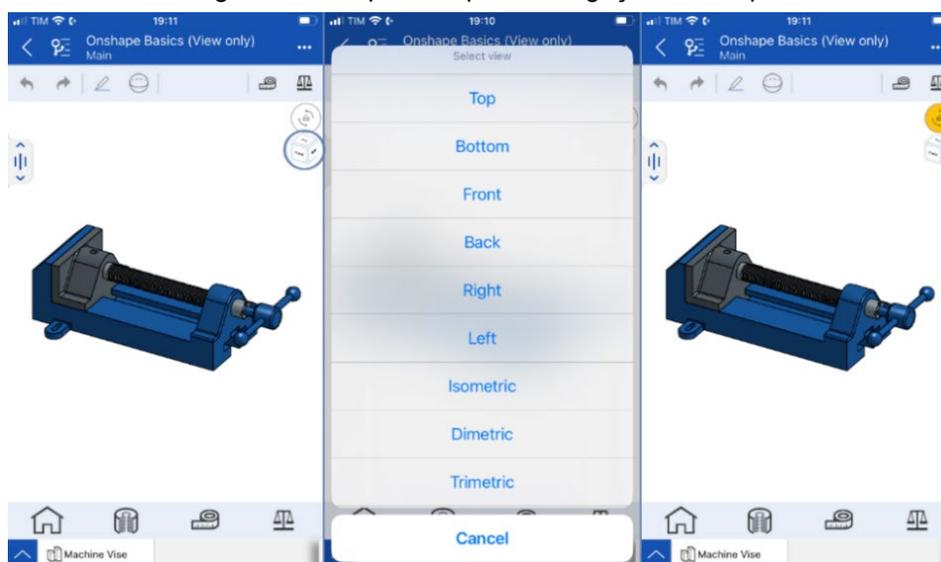
e o outro a mudança em tempo real. São utilizados também, cotas que indicam os valores modificados, exemplo: cota da distância de deslocamento ou ângulo modificado.

Além destes indicadores, é comum o aparecimento de *banners* que indicam possibilidades de interação não intuitivas com o aplicativo. Um exemplo disso ocorre quando o usuário utiliza um clique na área livre do aplicativo, um banner é inserido na parte superior da tela indicando a possibilidade de um duplo clique para desselecionar todos os objetos. No entanto, este só é apresentado na primeira vez que o usuário clica na área livre.

Temas e padrões: de forma geral o aplicativo utiliza as mesmas cores em toda a navegação e ícones comuns a outros aplicativos, como lápis para a edição, setas para desfazer e refazer entre outros.

Além destes ícones comuns, o app possui um cubo na parte superior direita, semelhante a outros programas gráficos tridimensionais como inventor, Maya e 3d Max, que permite a mudança para vistas específicas como topo, frente e perspectiva isométrica. Outro ícone interessante, é a utilização de um cadeado com setas que indicam rotação, localizado acima do cubo. O mesmo pode ser utilizado para o travamento da rotação da view, o que facilita algumas edições que não utilizam o *gizmo*. Estes mecanismos de rotação e travamento podem ser observados na Figura 44.

Figura 44 - Dispositivo para navegação OnShape



Fonte: OnShape ([s.d.]).

Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?

Clareza, quantidade e qualidade de informações na tela: Em geral é um aplicativo bem limpo, o que o torna visualmente agradável, sem informações desnecessárias. As ferramentas estão distribuídas em diferentes menus *dropdown*, acessíveis ao toque de um botão.

Atalhos: há pelo menos um atalho, os comandos existentes para uma determinada seleção, podem ser acessados a partir do toque duplo na tela, que apresenta um menu flutuante de ferramentas.

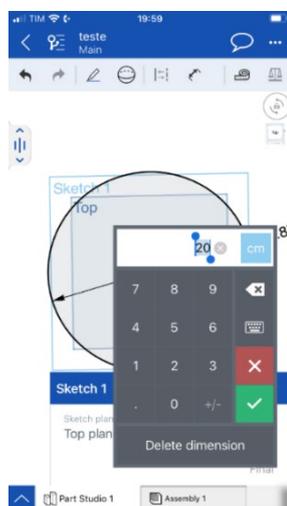
Utilização de sensores: o giro do aparelho não apresenta alterações na orientação da *view*, além disso o aplicativo não parece aproveitar os sensores do dispositivo.

O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

Prevenção de erros: o aplicativo possui tutorial para a compreensão dos gestos que podem ser utilizados, botões de desfazer e refazer e banners indicam funcionalidades que podem ser desconhecidas ao usuário.

Entrada de dados: o aplicativo permite a inserção de dimensões a partir de um teclado virtual numérico que aparece na tela quando uma cota é selecionada. Conforme a Figura 45.

Figura 45 - Entrada de dados no OnShape

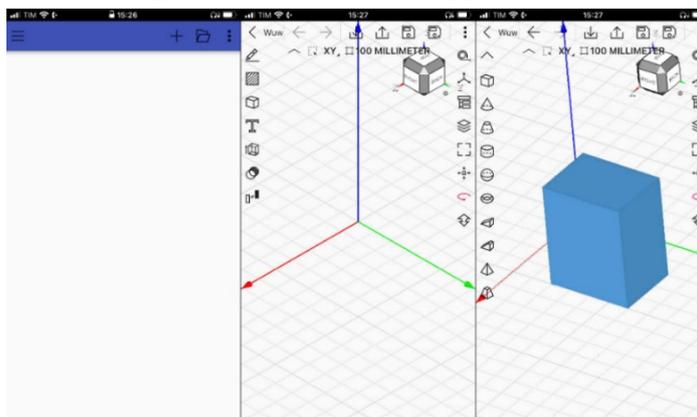


Fonte: OnShape ([s.d.]).

4.3.3. Análise do aplicativo Wuweido

Desenvolvido pela companhia Mozongsoft Co. Ltd., não apresenta tutorial introdutório. A tela inicial apresenta uma barra no topo e um espaço vazio abaixo. Quando o ícone com o sinal mais é acionado, o aplicativo apresenta uma propaganda, após o fechamento desta segue para a tela de modelagem (Figura 46).

Figura 46 - Telas iniciais do app Wuweido



Fonte: Wuweido ([s.d.]).

Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)? A resposta para esta pergunta se encontra no Quadro 10.

Quadro 10 -Navegação, seleção, desseleção e edição Wuweido

<p>Navegação: os gestos de navegação são realizados por meio de arraste dos dedos na tela, um dedo para rotação, dois dedos para pan e dois dedos em pinça para zoom.</p>	<p style="text-align: center;">Navegação Wuweido</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Rotação</p>  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>Pan</p>  <p>ARRASTE DUPLO</p> </div> <div style="text-align: center;"> <p>Zoom</p>  <p>ARRASTE PINÇA</p> </div> </div>
<p>Seleção: os objetos podem ser selecionados com toque simples. A seleção em janela é feita com tocar e segurar com um dedo seguido de arraste. A multisseleção pode ser realizada em janela ou com toques simples sobre os objetos.</p>	<p style="text-align: center;">Seleção Wuweido</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Simples</p>  <p>TOQUE SIMPLES</p> </div> <div style="text-align: center;"> <p>Janela</p>  <p>TOCAR E ARRASTE SIMPLES SEGURAR</p> </div> </div> <p style="text-align: center;">Multisseleção</p> <div style="text-align: center;">  <p>TOQUE SIMPLES</p> </div>
<p>Desselecionar: tocar e segurar ou dois toques na área livre. O aplicativo não apresenta desseleção individual.</p>	<p style="text-align: center;">Desselecionar Wuweido</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>TOCAR E SEGURAR</p> </div> <div style="text-align: center;">  <p>DOIS TOQUES</p> </div> </div>
<p>Editar: o aplicativo não apresenta possibilidade de edição de vértices, arestas e faces, mas permite a movimentação, rotação e escalonamento via <i>gizmo</i>.</p>	<p style="text-align: center;">Edição Wuweido</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>GIZMO</p> </div> <div style="text-align: center;"> <p>+</p>  <p>ARRASTE SIMPLES</p> </div> </div>

Fonte: a autora (2022)

O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?

Feedbacks: há uma mudança de cores das arestas dos objetos selecionados e o aparecimento do *gizmo* indica que eles podem ser modificados. Durante a alteração o objeto fica na posição original, porém um modelo transparente com arestas coloridas indica a nova alteração.

Temas e padrões: o aplicativo apresenta uma tela inicial com layout e cores diferentes da tela de modelagem e navegação. Por outro lado, possui ícones similares a outros aplicativos e *softwares* como: disquete para salvamento e setas para desfazer e refazer. Além disso, apresenta um cubo de navegação, no canto superior esquerdo, similar a outros programas tridimensionais.

Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?

Clareza, quantidade e qualidade de informações na tela: os ícones que servem de botões são transparentes, desta forma é possível enxergar o grid através deles, o que acrescenta ruído aos ícones. Além disso, as informações e botões se sobrepõe causando confusão na visualização.

Atalhos: não apresenta atalhos

Utilização de sensores: a tela muda de vertical para horizontal e vice-versa indicando a utilização do giroscópio.

O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

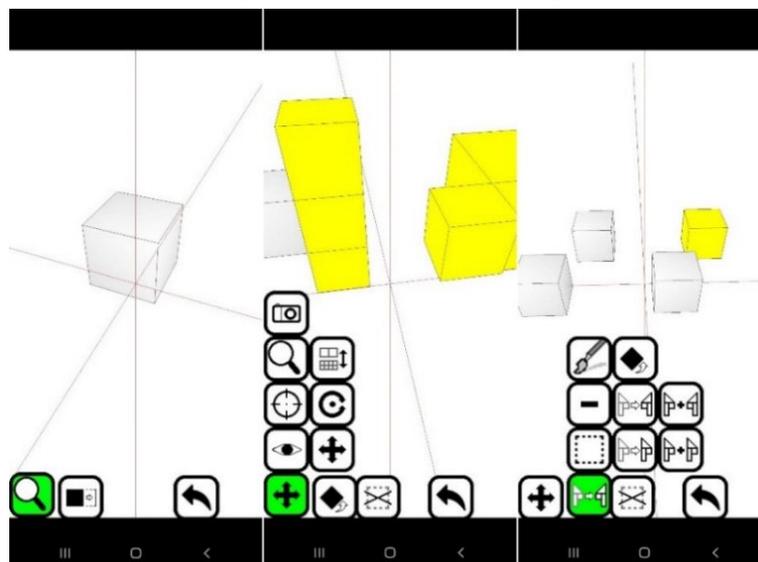
Prevenção de erros: as únicas ferramentas encontradas para lidar com eventuais erros, são os botões de desfazer, refazer e indicações escritas na parte superior da tela com instruções dos próximos passos.

Entrada de dados: é possível inserir coordenadas de forma numérica a partir do teclado virtual.

4.3.4. Análise do aplicativo Qubism 3D Modeling

Desenvolvido por Jonathon Quinn, não é um aplicativo base CAD. É de difícil compreensão e não apresenta ferramentas de instrução. Os eixos aparecem apenas na cor vermelha na tela e não acompanham a navegação dos objetos (Figura 47).

Figura 47 - Qubism 3D Modeling



Fonte: Qubism 3D Modeling ([s.d.]).

Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)? Respostas no Quadro 11

Quadro 11 - Navegação, seleção, desseleção e edição Qubism 3D Modeling

<p>Navegação: os gestos de navegação são realizados por meio de arraste dos dedos na tela, um dedo para rotação, dois dedos para pan e dois dedos em pinça para zoom.</p>	<p>Navegação Qubism 3D modeling</p> <p>Rotação Pan Zoom</p>  <p>ARRASTE SIMPLES ARRASTE DUPLO ARRASTE PINÇA</p>
<p>Seleção: a seleção pode ser feita em janela ou toque simples, porém é necessário utilizar uma ferramenta de seleção. O app não permite a seleção de arestas, vértices e planos.</p>	<p>Seleção Qubism 3D modeling</p> <p>Simples</p>  <p>FERRAMENTA DE SELEÇÃO TOQUE SIMPLES</p>
<p>Desselecionar: para desselecionar individualmente é necessário um toque simples sobre o modelo selecionado, a desseleção total é feita com o clique no botão desselecionar.</p>	<p>Desselecionar Qubism 3D modeling</p> <p>Individual Total</p>  <p>TOQUE SIMPLES FERRAMENTA DE DESSELEÇÃO</p>
<p>Editar: a edição é feita a partir da seleção dos objetos e de ferramentas específicas para mover e rotacionar seguida de arraste simples. O aplicativo não parece possuir ferramenta para escalonamento.</p>	<p>Edição Qubism 3D modeling</p>  <p>FERRAMENTA MOVER ARRASTE SIMPLES</p>  <p>FERRAMENTA ROTACIONAR ARRASTE SIMPLES</p>

Fonte: a autora (2022)

A rotação da câmera é baseada no ponto central do mundo, desta forma toda a rotação ocorre ao redor deste ponto independente do objeto que está em foco. Este fato, dificulta a navegação da *view*.

O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?

Feedbacks: os sólidos e as ferramentas mudam de cor ao serem selecionados. As edições mover, rotacionar e escalonar acontecem em tempo real, sem a utilização de outros indicativos de posição, rotação e escalonamento iniciais, apenas com a mudança do estado do modelo.

Temas e padrões: o aplicativo segue os mesmos padrões em todas as ferramentas. Apresenta alguns ícones similares a outros apps, como lupa e desfazer.

Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?

Clareza, quantidade e qualidade de informações na tela: o aplicativo não apresenta muitas informações na tela, somente alguns botões que abrem novos menus. Os ícones não são intuitivos e as etapas para a edição são de difícil compreensão.

Atalhos: o aplicativo não apresenta atalhos.

Utilização de sensores: não é possível girar a tela, o app não parece utilizar sensores

O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

Prevenção de erros: a única forma de prevenção de erros encontradas é o botão desfazer.

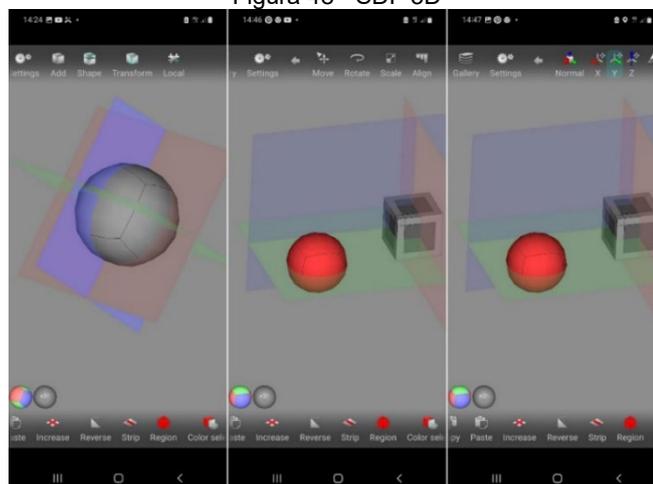
Entrada de dados: o aplicativo não apresenta possibilidade de entrada de dados.

4.3.5. Análise do aplicativo SDF 3D

Desenvolvido pela Antomind Group, não é um aplicativo CAD. Apresenta ferramentas simples (Figura 48), porém é de difícil navegação. Isso se deve ao fato

de o *target* da câmera estar sempre conectado ao centro do mundo, provocando a rotação da tela sempre baseada neste ponto e não no sólido em foco. Não apresenta tutorial de introdução.

Figura 48 - SDF 3D



Fonte: SDF 3D ([s.d.]).

Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)? Resposta no Quadro 12.

Quadro 12 - Navegação, seleção, desseleção e edição SDF 3D

<p>Navegação: os gestos de navegação são realizados por meio de arraste dos dedos na tela, um dedo para rotação, dois dedos para pan e dois dedos em pinça para zoom na área livre.</p>	<p style="text-align: center;">Navegação SDF 3D</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Rotação</p>  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>Pan</p>  <p>ARRASTE DUPLO</p> </div> <div style="text-align: center;"> <p>Zoom</p>  <p>ARRASTE PINÇA</p> </div> </div>
<p>Seleção: é possível selecionar com toque simples as faces, arestas e vértices, porém é necessário indicar qual destas partes será selecionada antes do toque, clicando no botão indicativo. Não é possível selecionar totalmente vários objetos, a multisseleção de partes do sólido é feita com toque simples e não apresenta seleção em janela.</p>	<p style="text-align: center;">Seleção SDF 3D</p> <hr/> <p style="text-align: center;">Simples</p> <div style="text-align: center;">  <p>TOQUE SIMPLES</p> </div>
<p>Desselecionar: para desselecionar é necessário um toque simples sobre a parte selecionada ou tocar e segurar sobre o objeto ou na área livre para desselecionar todos os itens selecionados de uma vez.</p>	<p style="text-align: center;">Desselecionar SDF 3D</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Individual</p>  <p>TOQUE SIMPLES</p> </div> <div style="text-align: center;"> <p>Total</p>  <p>TOCAR E SEGUIR</p> </div> </div>
<p>Editar: a edição é feita a partir da seleção de ferramentas específicas como mover, rotacionar e escalonar. Após a seleção da ferramenta é necessário selecionar os eixos para a alteração e posteriormente arrastar o dedo sobre a tela.</p>	<p style="text-align: center;">Edição SDF 3D</p> <div style="text-align: center;">  <p>FERRAMENTA TRANSFORMAÇÃO</p> </div> <div style="display: flex; justify-content: center; align-items: center; gap: 20px;"> <div style="text-align: center;">  <p>FERRAMENTA MOVER</p> </div> <div style="text-align: center;"> <p>ou</p>  <p>FERRAMENTA ROTACIONAR</p> </div> <div style="text-align: center;"> <p>ou</p>  <p>FERRAMENTA ESCALA</p> </div> </div> <div style="text-align: center; margin-top: 10px;">  <p>FERRAMENTA EIXO</p> </div> <div style="text-align: center; margin-top: 5px;">  <p>ARRASTE SIMPLES</p> </div>

Fonte: a autora (2022)

O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?

Feedbacks: os objetos selecionados e/ou as partes dele mudam de cor ao serem selecionadas. As edições mover, rotacionar e escalonar acontecem em tempo real, sem a utilização de outros indicativos de posição, rotação e escalonamento iniciais, apenas com a mudança do estado do modelo.

Temas e padrões: o app apresenta o mesmo padrão de cores e ícones em todas as telas e possui ícones semelhantes a outros aplicativos como copiar, colar, desfazer e refazer.

Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?

Clareza, quantidade e qualidade de informações na tela: o aplicativo apresenta dois menus principais, no topo e base da tela, que são alterados à medida que as ferramentas são selecionadas. Não apresenta muitas informações ao mesmo tempo na tela. É possível colocar o app em tela cheia e para retornar ao modo normal é necessário tocar duas vezes na tela.

Atalhos: não apresenta atalhos

Utilização de sensores: não é possível girar a tela com o giroscópio. O app não apresenta a utilização de sensores.

O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

Prevenção de erros: as únicas formas de prevenção de erros encontradas são os botões de desfazer e refazer.

Entrada de dados: não apresenta formas de inserção de dados como teclado digital.

De forma geral, os aplicativos analisados apresentam funcionalidades interessantes. No entanto, dois aplicativos ficaram abaixo das expectativas: Wuweido e Qubism. A seguir são analisados os aplicativos de desenho técnico.

4.3.6. Análise do aplicativo AutoCAD

O aplicativo desenvolvido pela Autodesk pode ser baixado na versão *mobile* para IOS e Androide. Só apresenta possibilidades de trabalho 2D, porém permite criar e abrir desenhos salvos em **dwg** na conta do usuário, desta forma os arquivos desenvolvidos no programa para computador podem ser abertos no celular e vice-versa.

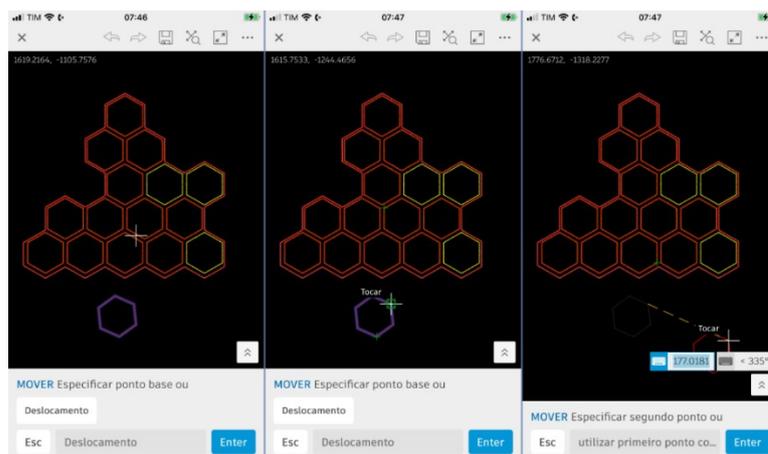
Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)? Para esta avaliação é utilizado um quadro (Quadro 13), com a explicação na coluna esquerda e gráficos ilustrativos na direita.

Quadro 13 - Navegação, seleção, desseleção e edição AutoCAD

<p>Navegação: O aplicativo não apresenta movimento de rotação 2D ou 3D na navegação. O Pan pode ser realizado com o arraste de apenas um dedo na tela enquanto o zoom com arraste em pinça de dois dedos.</p>	<p style="text-align: center;">Navegação AutoCAD</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Pan</p>  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>Zoom</p>  <p>ARRASTE PINÇA</p> </div> </div>
<p>Seleção: pode ser feita com toque simples sobre o objeto. A seleção em janela pode ser feita de duas formas: selecionar um objeto e arrastar um dedo ou utilizar a ferramenta de seleção e arrastar um dedo, durante a seleção, a navegação em pan fica desativada. Um fato interessante é, quando a janela é feita da direita para a esquerda, ela fica verde e permite a seleção dos objetos que forem tocados por ela. Porém, quando a janela é feita da esquerda para a direita fica azul e só permite a seleção de objetos que estejam totalmente inseridos na mesma.</p>	<p style="text-align: center;">Seleção AutoCAD</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Simples</p>  <p>TOQUE SIMPLES</p> </div> <div style="text-align: center;"> <p>Janela</p>  <p>TOQUE SIMPLES SOBRE DESENHO ARRASTE SIMPLES</p> <p>OU</p>  <p>FERRAMENTA DE SELEÇÃO ARRASTE SIMPLES</p> </div> </div>
<p>Desselecionar: é necessário clicar duas vezes na área livre para deselegionar os objetos, o app não apresenta uma forma de deselegionar os objetos um a um.</p>	<p style="text-align: center;">Desselecionar AutoCAD</p> <div style="text-align: center;"> <p>Total</p>  <p>DOIS TOQUES</p> </div>
<p>Editar: após a seleção do objeto, é necessária a escolha da forma de edição: mover, dimensionar ou girar, entre outros. Todas ocorrem da mesma forma (Figura 48), surge uma cruz (<i>gizmo</i>) na tela que deve ser arrastada até o ponto que servirá de ancora para o objeto, para confirmar o local é necessário um click na tela. A partir deste ponto o objeto passa a seguir o arraste do dedo com uma indicação tracejada do deslocamento, quando o arraste é finalizado aparece a indicação da palavra tocar para a confirmação da posição.</p>	<p style="text-align: center;">Edição AutoCAD</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>FERRAMENTA MOVER</p> </div> <div style="text-align: center;"> <p>ou</p>  <p>FERRAMENTA ROTACIONAR</p> </div> <div style="text-align: center;"> <p>ou</p>  <p>FERRAMENTA ESCALA</p> </div> </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 10px;"> <div style="text-align: center;">  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>+</p>  <p>TOQUE SIMPLES</p> </div> <div style="text-align: center;"> <p>+</p>  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>+</p>  <p>TOQUE SIMPLES</p> </div> </div>

Fonte: a autora (2022)

Figura 49 - Edição de desenhos AutoCAD



Fonte: AutoCAD ([s.d.]).

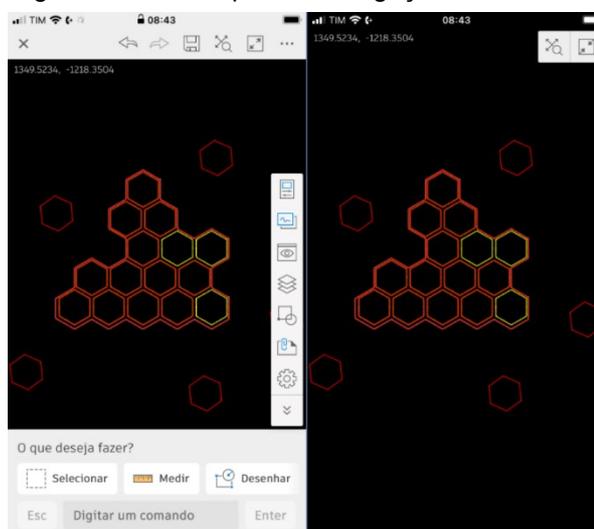
A Figura 49 apresenta a edição no aplicativo AutoCAD.

O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?

Feedbacks: o aplicativo apresenta uma mudança nas cores dos desenhos selecionados e/ou partes do desenho. Além disso, o aparecimento do *gizmo* de inserção da ancora, as indicações de toque, o tracejado da trajetória e a indicação do objeto em seu local inicial, facilitam o entendimento do usuário quanto as ações necessárias e/ou que estão sendo executadas.

Temas e padrões: utiliza as mesmas cores em toda a navegação e ícones comuns a outros aplicativos, como disquetes para salvamento, setas para desfazer e refazer entre outros.

Figura 50- Botões para a navegação no AutoCAD



Fonte: AutoCAD ([s.d.]).

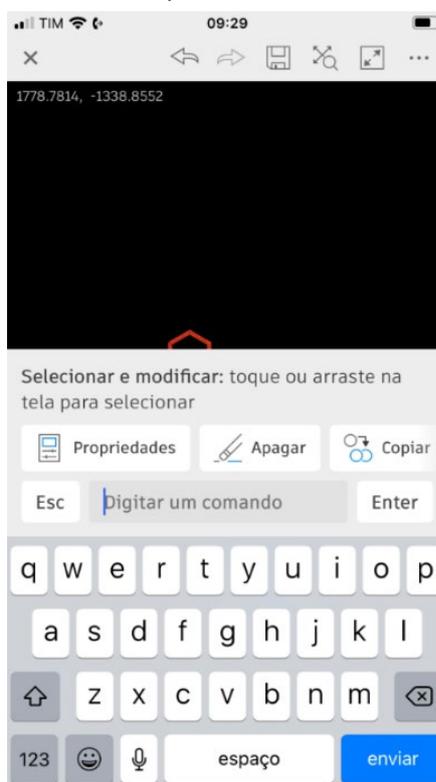
O AutoCAD apresenta dois botões no canto superior direito, o primeiro centraliza e mostra todo o desenho na *view* e o segundo expande a *view* (Figura 50) escondendo os menus, estes itens podem ser observados em diferentes aplicativos que necessitam de espaço para a visualização e edição.

Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?

Clareza, quantidade e qualidade de informações na tela: Em geral é um aplicativo bem limpo, com menus retrateis e a possibilidade de expansão da *view* para a melhor utilização do espaço.

Atalhos: o app apresenta a possibilidade de digitar os comandos desejados ao invés de procurá-los na tela, assim como na versão de computador, o que permite a entrada de dados pelo teclado virtual (Figura 51). Além disso, ao selecionar um desenho a ferramenta de seleção em janela é acionada.

Figura 51 - Atalho para comandos do AutoCAD



Fonte: AutoCAD ([s.d.]).

Utilização de sensores: o único sensor que foi constatada a utilização foi o giroscópio, devido a possibilidade de mudar a orientação da tela ao girar o celular.

O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

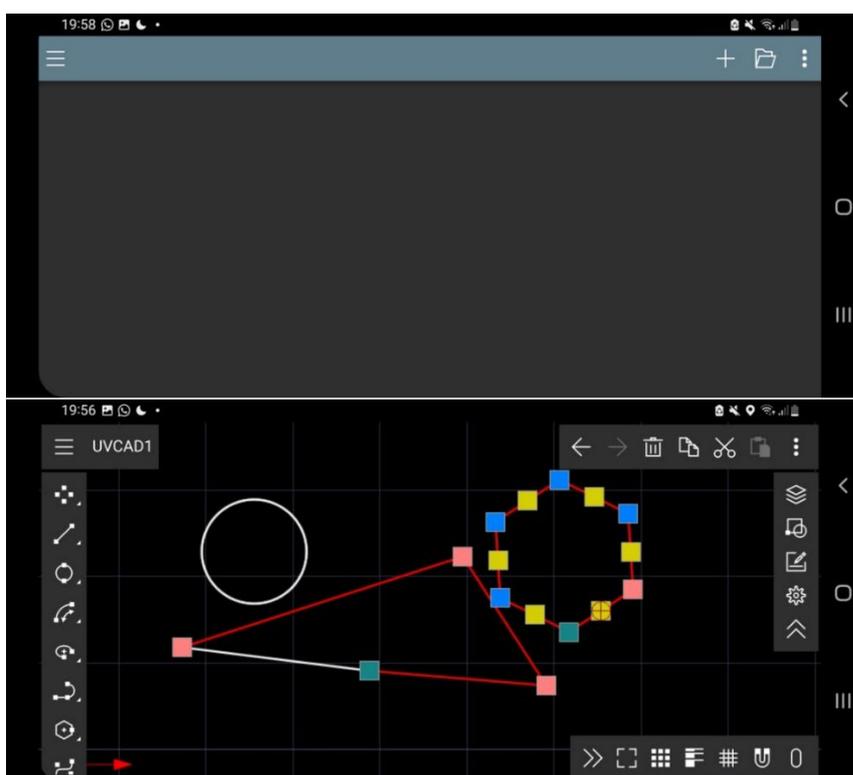
Prevenção de erros: a única possibilidade perceptível, relacionada ao tratamento de erros, é a possibilidade de refazer e desfazer.

Entrada de dados: a entrada de dados é feita a partir de teclado virtual que apresenta um botão de microfone e permite que o nome do comando seja ditado.

4.3.7. Análise do aplicativo UVCAD

Desenvolvido por Mozongsoft Co. Ltd., é diferente dos demais apps analisados pois, utiliza a tela em posição horizontal com menus em todas as bordas do app (Figura 52).

Figura 52 – UVCAD



Fonte: UVCAD ([s.d.]).

Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)? No Quadro 14, na coluna da esquerda estão inseridas as explicações da navegação e na coluna direita são apresentados gráficos ilustrativos dos gestos e ferramentas utilizados.

Quadro 14 - Navegação, seleção, desseleção e edição UVCAD

<p>Navegação: O aplicativo não apresenta movimento de rotação 2D ou 3D na navegação. O Pan pode ser realizado com o arraste de dois dedos na tela enquanto o zoom com arraste em pinça de dois dedos.</p>	<p style="text-align: center;">Mavegação UVCAD</p> <hr/> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Pan</p>  <p>ARRASTE DUPLIO</p> </div> <div style="text-align: center;"> <p>Zoom</p>  <p>ARRASTE PINÇA</p> </div> </div>
<p>Seleção: é necessário tocar e segura para ativar a ferramenta de seleção individual, arrastar o dedo sobre o objeto e tirar o dedo da tela, para seleção individual. É possível selecionar vários objetos a partir do arraste em janela da direita para esquerda.</p>	<p style="text-align: center;">Seleção UVCAD</p> <hr/> <p style="text-align: center;">Simples</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>TOCAR E SEGUIRAR</p> </div> <div style="text-align: center;">  <p>ARRASTE SIMPLIS</p> </div> <div style="text-align: center;">  <p>TIRAR O DEDO DA TELA</p> </div> </div> <p style="text-align: center;">Janela</p> <div style="text-align: center;">  <p>ARRASTE SIMPLIS</p> </div>
<p>Desselecionar: não apresenta possibilidade de desselecionar individualmente. É possível desselecionar todos os objetos com um toque simples ou arrastando um dedo em janela da esquerda para a direita.</p>	<p style="text-align: center;">Desselecionar UVCAD</p> <hr/> <p style="text-align: center;">Total</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>TOQUE SIMPLIS</p> </div> <div style="text-align: center;"> <p>OU</p>  <p>ARRASTE SIMPLIS</p> </div> </div>
<p>Editar: o aplicativo não parece ter uma lógica de edição, em alguns momentos a manipulação dos vértices move o objeto como um todo e em outros arrasta somente o vértice. Nos círculos há um eixo central e o objeto é movido a partir dele, enquanto a alteração do raio é feita movendo-se os vértices. No entanto, os polígonos aparentemente podem ser movidos separadamente ou em conjunto com objeto, mas não há uma distinção clara de como alternar entre estes estados. O app não apresenta ferramentas de movimentação, rotação e escala. Estes fatos são confirmados nos comentários do app na google play.</p>	<p style="text-align: center;">Edição UVCAD</p> <hr/> <p style="text-align: center;">Inconclusivo</p>

Fonte: a autora (2022)

O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?

Feedbacks: as formas selecionadas mudam de cor e apresentam vértices coloridos para a edição. Durante a criação das formas, um banner indica o próximo passo a ser realizado. As edições mover, rotacionar e escalonar acontecem em tempo real, sem a utilização de outros indicativos de posição, rotação e escalonamento iniciais, apenas com a mudança do estado do modelo.

Temas e padrões: a única diferença entre temas é da tela inicial para a tela de edição, porém utiliza as mesmas cores em toda a navegação e ícones comuns a outros aplicativos, como disquetes para salvamento, setas para desfazer e refazer entre outros.

Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?

Clareza, quantidade e qualidade de informações na tela: apresenta menus em todas as bordas com botões que abrem outros menus, porém os da esquerda podem ser fechados totalmente para aumentar a área de edição.

Atalhos: não apresenta atalhos

Utilização de sensores: não utiliza sensores.

O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

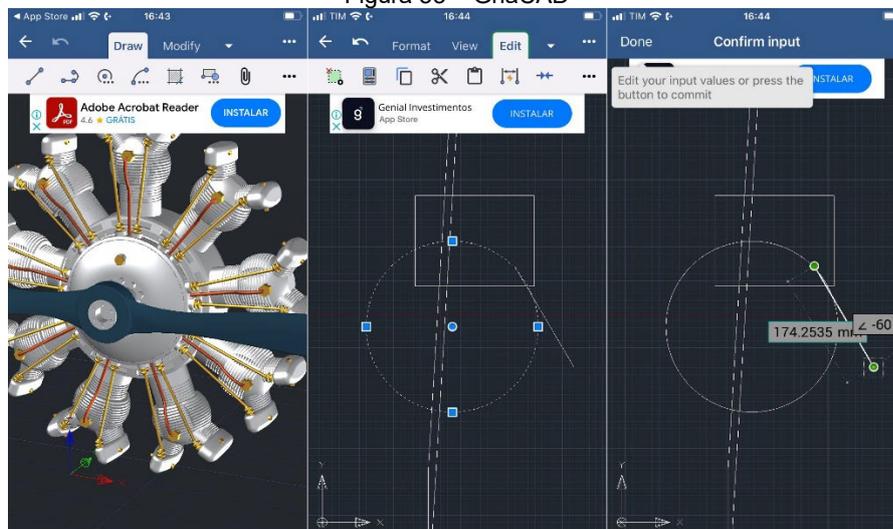
Prevenção de erros: banner no canto superior esquerdo que indica os próximos passos, e botão desfazer e refazer.

Entrada de dados: permite a inserção de coordenadas na criação das formas via teclado virtual numérico.

4.3.8. ANÁLISE DO APLICATIVO GNACAD

O aplicativo foi desenvolvido por Luidmila Zaitceva (Figura 52) e apresenta a possibilidade de editar gráficos CAD em duas e três dimensões. No entanto não apresenta ferramentas para a criação de sólidos tridimensionais. O app apresenta uma versão gratuita, porém com propagandas em banners e vídeos.

Figura 53 – GnaCAD



Fonte: GnaCAD ([s.d.]).

Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)?

O Quadro 15 apresenta os controles de navegação, seleção, deselegionar e edição do GnaCAD.

Quadro 15 - Navegação, Seleção e Edição GnaCAD

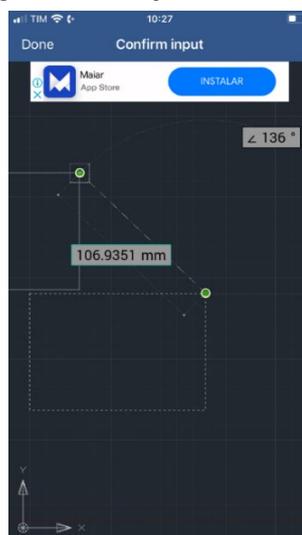
<p>Navegação: na navegação de desenhos 2D, o movimento de Pan é realizado com o arraste de um dedo na <i>view</i> e o zoom com o movimento de pinça. A navegação em 3D não é feita de forma intuitiva, os movimentos de Pan e Zoom são os mesmos dos desenhos em 2D, porém as rotações precisam ser ativadas via menu. Após a ativação da rotação, o movimento de rotação passa a ser feito com o arraste de um dedo, o de Pan com o arraste de dois dedos e o zoom e rotação do objeto em foco com o arraste em pinça. Também é possível ativar a rotação de constrição que utiliza um <i>gizmo</i> para rotação nos eixos específicos.</p>	<p style="text-align: center;">Navegação 2D GnaCAD</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Pan</p>  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>Zoom</p>  <p>ARRASTE PINÇA</p> </div> </div> <p style="text-align: center;">Navegação 3D GnaCAD</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Rotação</p>  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>Pan</p>  <p>ARRASTE DUPLO</p> </div> <div style="text-align: center;"> <p>Zoom/ Rotação</p>  <p>ARRASTE PINÇA</p> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;"> <p>Constrição de Rotação</p>  <p>GIZMO</p> </div> <div style="text-align: center;"> <p>Pan</p>  <p>ARRASTE DUPLO</p> </div> <div style="text-align: center;"> <p>Zoom/ Rotação</p>  <p>ARRASTE PINÇA</p> </div> </div>
<p>Seleção: pode ser feita com toque simples sobre o objeto. A seleção em janela pode ser feita tocando e segurando seguido de arraste. Semelhante ao AutoCAD, apresenta diferença na seleção de acordo com a direção da criação da janela. Quando a janela é feita da direita para a esquerda, ela fica verde e permite a seleção dos objetos que forem tocados por ela. Porém, quando a janela é feita da esquerda para a direita fica azul e só permite a seleção de objetos que estejam totalmente inseridos na mesma.</p>	<p style="text-align: center;">Seleção GnaCAD</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Simples</p>  <p>TOQUE SIMPLES</p> </div> <div style="text-align: center;"> <p>Janela</p>  <p>TOCAR E SEGURAR</p> </div> <div style="text-align: center;"> <p>Janela</p>  <p>ARRASTE SIMPLES</p> </div> </div>
<p>Desselecionar: é necessária a utilização de uma ferramenta para liberar todos os objetos selecionados de uma vez. O app não apresenta formas de deselegionar individualmente os objetos.</p>	<p style="text-align: center;">Desselecionar GnaCAD</p> <p style="text-align: center;">Total</p>  <p style="text-align: center;">FERRAMENTA DE DESSELEÇÃO</p>
<p>Editar: com o objeto selecionado, é necessária a escolha da forma de edição: mover, dimensionar ou girar, entre outros. Todas ocorrem da mesma forma, com um toque o <i>gizmo</i> é inserido no ponto de ancora e o arraste desloca o objeto para o novo <i>status</i>. Para encerrar é preciso acionar o botão Done.</p>	<p style="text-align: center;">Edição GnaCAD</p> <div style="display: flex; justify-content: space-around; margin-bottom: 10px;"> <div style="text-align: center;">  <p>FERRAMENTA MOVER</p> </div> <div style="text-align: center;"> <p>ou</p>  <p>FERRAMENTA ROTACIONAR</p> </div> <div style="text-align: center;"> <p>ou</p>  <p>FERRAMENTA ESCALA</p> </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>TOQUE SIMPLES</p> </div> <div style="text-align: center;"> <p>+</p>  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>+</p>  <p>TOQUE SIMPLES</p> </div> </div>

Fonte: a autora (2022)

O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?

Feedbacks: após a seleção os objetos ficam transparentes, as edições apresentam o objeto na posição original em linhas cheias e tracejado na posição final. Além disso uma linha tracejada, uma cota e uma marcação e de ângulo indicam o montante da alteração que está sendo realizada em tempo real. A Figura 54 demonstra as indicações que aparecem na tela.

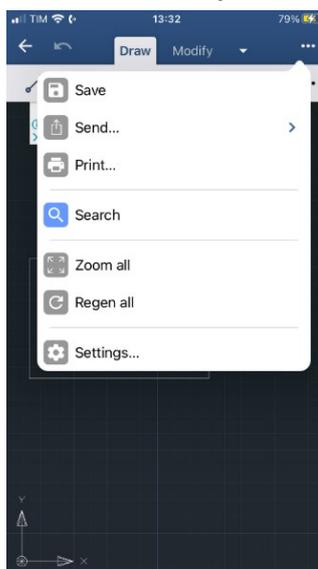
Figura 54 - Edição no GnaCAD



Fonte: GnaCAD ([s.d.]).

Temas e padrões: o app utiliza ícones usuais como, disquete para salvamento, lupa para pesquisa e engrenagem para configurações. Também apresenta cores iguais durante toda a navegação e existe um botão de zoom para centralizar todo o desenho, no entanto fica escondido em um menu sanduiche no canto superior direito. Conforme pode ser visto na Figura 55

Figura 55- Zoom e centralização do desenho no GnaCAD

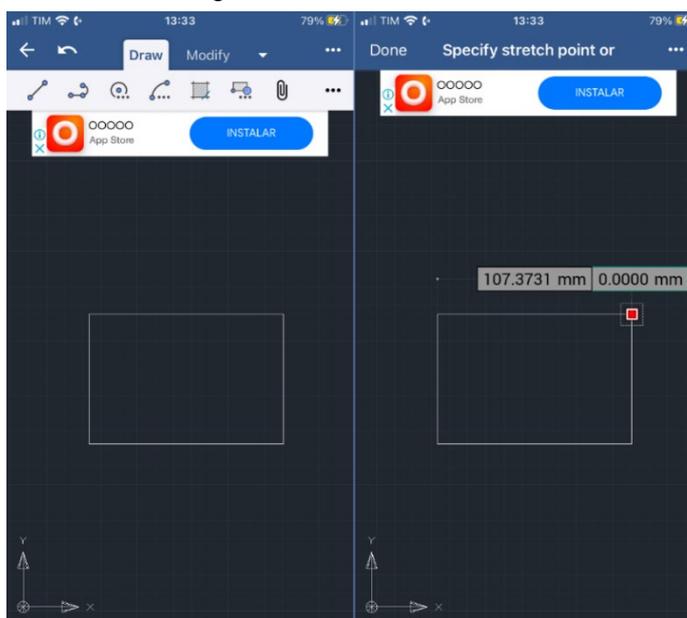


Fonte: GnaCAD ([s.d.]).

Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?

Clareza, quantidade e qualidade de informações na tela: apresenta poucas informações na tela, no entanto os banners de propaganda atrapalham a utilização. Todos os menus estão disponíveis ao mesmo tempo em tabs na barra superior, no entanto são escondidos quando uma ação os sobrepõe com a indicação do próximo passo a ser executado. A Figura 56 ilustra esta afirmação.

Figura 56 - Menus GnaCAD



Fonte: GnaCAD ([s.d.]).

Atalhos: ao tocar e segurar com um dedo o aplicativo disponibiliza um menu com ferramentas.

Utilização de sensores: utiliza o giroscópio para modificar a orientação da interface e da *viewport*.

O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

Prevenção de erros: a única possibilidade perceptível, relacionada ao tratamento de erros, é a possibilidade de refazer e desfazer.

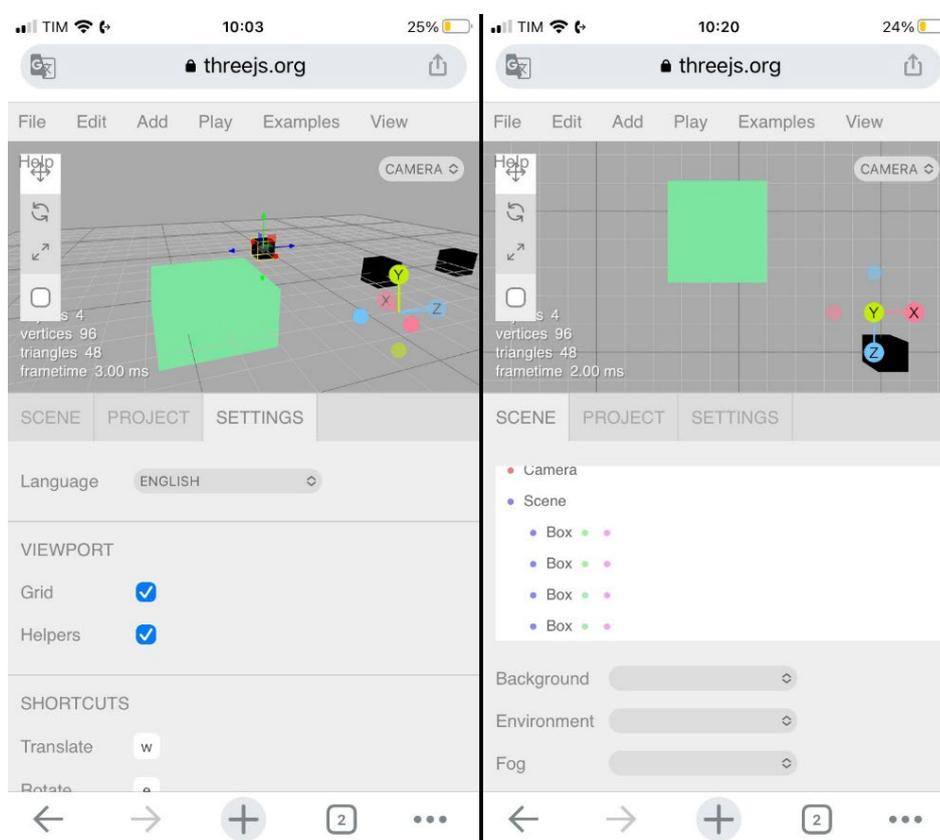
Entrada de dados: a entrada de dados é feita a partir de teclado virtual, assim como os outros aplicativos.

No próximo item será analisado o site do editor Three.js no navegador móvel, foram utilizados para análise os *browsers* Safari e Chrome na tentativa de melhorar o desempenho e as ferramentas. No entanto, não foram observadas mudanças.

4.3.9. Análise do site Three.js Editor

Three.js é uma biblioteca *open source* (código aberto) baseada em WebGL, que foi criada para mostrar gráficos tridimensionais diretamente no navegador. O editor da biblioteca roda em *browser* de computador e celular. A Figura 57 apresenta a interface do site em dispositivo móvel.

Figura 57 - Editor Three.js



Fonte: a autora (2022)

Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)?

No Quadro 16 encontram-se as descrições das formas de navegação, seleção, desseleção e edição. Além da navegação descrita no quadro, também é possível navegar a partir da ferramenta que se encontra no canto inferior direito da *view* tridimensional.

Quadro 16 - Navegação, seleção, desseleção e edição do Editor Three.js

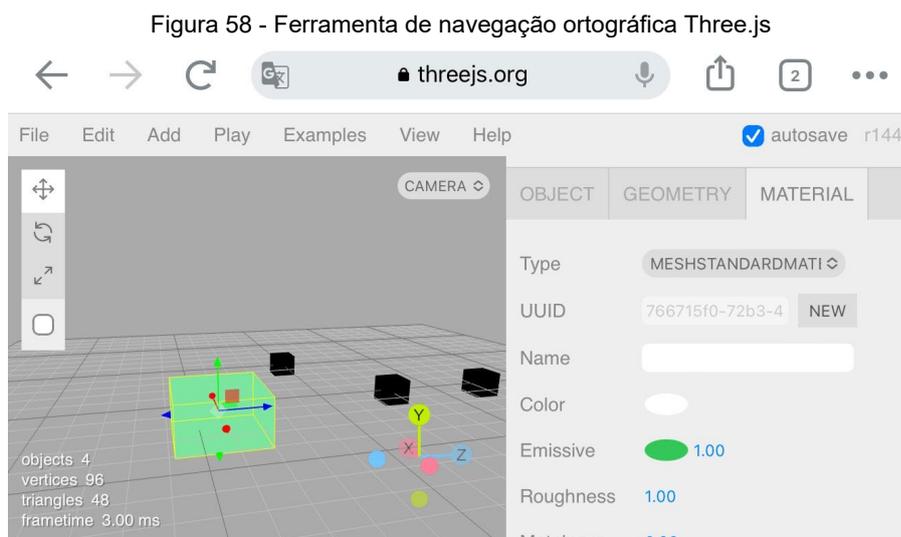
<p>Navegação: os gestos de navegação são realizados por meio de arraste dos dedos na tela, um dedo para rotação, dois dedos para pan e dois dedos em pinça para zoom.</p>	<p style="text-align: center;">Mavegação Editor Three.js</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Rotação</p>  <p>ARRASTE SIMPLES</p> </div> <div style="text-align: center;"> <p>Pan</p>  <p>ARRASTE DÚPLO</p> </div> <div style="text-align: center;"> <p>Zoom</p>  <p>ARRASTE PINÇA</p> </div> </div>
<p>Seleção: não apresenta a possibilidade de multisseleção, para selecionar individualmente é necessário o toque simples sobre o modelo.</p>	<p style="text-align: center;">Seleção Editor Three.js</p> <p style="text-align: center;">Simples</p>  <p style="text-align: center;">TOQUE SIMPLES</p>
<p>Desselecionar: é feito com um toque simples na área livre ou sobre um modelo não selecionado.</p>	<p style="text-align: center;">Desseleção Editor Three.js</p> <p style="text-align: center;">Simples</p>  <p style="text-align: center;">TOQUE SIMPLES</p>
<p>Editar: a edição é feita a partir da seleção das ferramentas mover, rotacionar e escalonar a partir do <i>gizmo</i> e com arraste simples.</p>	<p style="text-align: center;">Edição Editor Three.js</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p>FERRAMENTA MOVER</p> </div> <div style="text-align: center;"> <p>ou</p>  <p>FERRAMENTA ROTACIONAR</p> </div> <div style="text-align: center;"> <p>ou</p>  <p>FERRAMENTA ESCALA</p> </div> </div> <div style="display: flex; justify-content: center; align-items: center; margin-top: 10px;">  <p>GIZMO</p> +  <p>ARRASTE SIMPLES</p> </div>

Fonte: a autora (2022)

O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?

Feedbacks: as arestas dos sólidos selecionados mudam de cor e o *gizmo* aparece para a edição. As edições mover, rotacionar e escalonar acontecem em tempo real, sem a utilização de outros indicativos de posição, rotação e escalonamento iniciais, apenas com a mudança do estado do modelo.

Temas e padrões: o site somente apresenta ícones para as ferramentas de edição (mover, rotacionar e escalonar) semelhantes a outros *softwares* 3D, para os demais botões são utilizadas palavras. Apresenta a mesma paleta de cores durante toda a navegação. Além destes, o editor possui uma ferramenta de navegação para as vistas ortográficas, no canto inferior esquerdo da *view*, similar à do *software* Blender (Figura 58)



Fonte: a autora (2022)

Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?

Clareza, quantidade e qualidade de informações na tela: a tela apresenta uma quantidade considerável de elementos, pois o painel lateral ocupa a metade da tela no dispositivo móvel (no computador a proporção é menor). Existe o modo tela cheia, porém não funciona em *smartphones*. Por outro lado, as informações são diretas e objetivas. No topo da tela existe um menu *dropdown* que permite a abertura de novas ferramentas.

Atalhos: possui a possibilidade de utilização de Ctrl z para desfazer e Ctrl shift z para refazer, no entanto no dispositivo móvel não podem ser utilizados e não apresentam substitutos.

Utilização de sensores: é possível mudar a orientação da tela de vertical para horizontal e vice-versa a partir do giroscópio.

O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

Prevenção de erros: foram encontrados somente os botões desfazer e refazer.

Entrada de dados: é possível digitar os valores necessários a partir do teclado virtual.

Análise dos aplicativos demonstrou algumas tendências na navegação e seleção dos objetos nas *views*. Assim como, possibilidades interessantes de atalhos e edição que facilitam a utilização dos apps, no próximo item são investigadas as diferentes soluções encontradas.

4.3.10. Conclusões das análises

Neste item são comparados os resultados encontrados em cada aplicativo. Na primeira pergunta de análise foram avaliados a navegação, seleção, desseleção e edição. Devido as diferenças na navegação 3D e 2D, na análise de navegação, os aplicativos foram separados nestas categorias.

Para melhor visualização entre gestos de toque de gestos de arraste, assim como ferramentas específicas e *gizmos*, foi elaborada uma legenda de cores (Figura 59) que é utilizada nos quadros comparativos.

Figura 59 - Legenda de cores



Fonte: a autora (2022)

Como os aplicativos utilizam os gestos de navegação (rotação, pan e zoom), seleção (simples, específica e em janela), desseleção (individual e em conjunto) e edição (mover, rotacionar e escalonar)?

Navegação: Rotação, Pan e Zoom

Os gestos navegação em todos os aplicativos 3D são os mesmos (Quadro 17). Há uma tendência à movimentos de arraste, intuitivos e largamente utilizados em outros aplicativos, para realização de tarefas de navegação. São utilizados: arraste simples para a rotação, arraste duplo para **pan** e arraste em pinça para **zoom**. Os aplicativos 3D Modeling App, OnShape 3D CAD e GnaCAD 3D apresentam também

formas específicas de rotação a partir do movimento de pinça como: rotacionar apenas em um eixo ou rotação com foco na região entre os dedos. O app Gna 3D também apresenta rotação por constrição mediante *gizmo*.

Quadro 17 - Gestos de navegação dos aplicativos 3D analisados

Navegação nos aplicativos 3D					
	Rotação	Pan	Zoom	Rotação específica	Constrição Rotação
3D Modeling App					
OnShape 3D CAD					
Wuweido					
Qubism					
SDF 3D					
GnaCAD 3D					
Three.js					
<p>LEGENDA</p> <p> GESTOS DE TOQUE  GESTOS DE ARRASTE  FERRAMENTAS  GIZMO</p>					

Fonte: a autora (2022)

A navegação nos aplicativos 2D também utiliza movimentos de arraste, no entanto foi observado que os apps não utilizam navegação para a rotação, mantendo sempre a direção da tela. Os três aplicativos analisados, utilizam os mesmos gestos para a navegação, com exceção do Pan no UVCAD que é realizado com o arraste duplo (Quadro 18).

Quando comparadas as navegações 3D e 2D, parece haver uma tendência de utilização de arraste simples para a navegação mais utilizada. Isso quer dizer que em aplicativos tridimensionais, os movimentos de rotação da tela, em geral, são mais acessados, pois a modelagem requer a visualização constante de diferentes ângulos do modelo, enquanto o pan e zoom auxiliam os movimentos de rotação. Já nos

aplicativos 2D, a necessidade de visualização recai sobre a movimentação em pan devido a questões de enquadramento do desenho na tela. Desta forma, é possível compreender que o gesto mais simples seja utilizado para a navegação de maior incidência.

Quadro 18 - Gestos de navegação dos aplicativos 2D analisados

Navegação nos aplicativos 2D		
	Pan	Zoom
AutoCAD	 ARRASTE SIMPLES	 ARRASTE PINÇA
UVCAD	 ARRASTE DUPLO	 ARRASTE PINÇA
GnaCAD 2D	 ARRASTE SIMPLES	 ARRASTE PINÇA
<p>LEGENDA</p> <p> GESTOS DE TOQUE  GESTOS DE ARRASTE  FERRAMENTAS  GIZMO</p>		

Fonte: a autora (2022)

Seleção: Simples, específica, multisseleção e janela

Inicialmente não havia sido cogitada a análise da multisseleção, no entanto no decorrer das análises, ficou evidente a necessidade de inserir esta nova possibilidade, pois a maioria dos aplicativos, desseleciona o modelo anteriormente selecionado quando um próximo sólido ou desenho é selecionado, no entanto isso não ocorre nos apps 3D Modeling App e Wuweido. Nestes, os sólidos permanecem selecionados enquanto novos sólidos podem ser acrescentados a seleção.

No Quadro 19 é possível observar a tendência na utilização de toques simples para a seleção individual (simples), enquanto para a seleção em janela, há a predominância de tocar e segurar seguido de arraste. Este fato, parece acontecer devido a necessidade de acionar a ferramenta de seleção em janela a partir de tocar e segurar e posteriormente realizar o arraste para selecionar. Estes gestos poderiam ser considerados como atalhos, porém os aplicativos não apresentam outras ferramentas ou botões para a seleção, isso significa que eles substituem as interfaces gráficas totalmente por interfaces gestuais, o que confirma a visão de Wigdor e Wixon (2011), quando estes defendem que as interfaces gestuais não devem ser pensadas como adaptações de outras interfaces.

Quadro 19 -Gestos de seleção

Seleção				
	Simple	Específica	Multisseleção	Janela
3D Modeling App	 TOQUE SIMPLES	✗	 +  TOQUE SIMPLES + TOCAR E SEGURAR	 +  TOCAR E SEGURAR + ARRASTE SIMPLES
OnShape 3D CAD	 TOQUE SIMPLES	 +  TOCAR E SEGURAR + ARRASTE SIMPLES	✗	 +  TOCAR E SEGURAR DUPLA + ARRASTE PINÇA
Wuweido	 TOQUE SIMPLES	✗	 TOQUE SIMPLES	 +  TOCAR E SEGURAR + ARRASTE SIMPLES
Qubism	 FERRAMENTA DE SELEÇÃO + TOQUE SIMPLES	✗	✗	✗
SDF 3D	 TOQUE SIMPLES	✗	✗	✗
GnaCAD 3D	 TOQUE SIMPLES	✗	✗	 +  TOCAR E SEGURAR + ARRASTE SIMPLES
Three.js	 TOQUE SIMPLES	✗	✗	✗
AutoCAD	 TOQUE SIMPLES	✗	✗	 +  TOQUE SIMPLES + ARRASTE SIMPLES OU  +  FERRAMENTA DE SELEÇÃO + ARRASTE SIMPLES
UVCAD	 +  TOCAR E SEGURAR + ARRASTE SIMPLES +  TIRAR O DEDO DA TELA	✗	✗	 ARRASTE SIMPLES
LEGENDA  GESTOS DE TOQUE  GESTOS DE ARRASTE  FERRAMENTAS  GIZMO				

Fonte: a autora (2022)

Desselecionar: individual e conjunto

A desseleção apresenta predominância de toques, sendo eles simples, duplos ou prolongados (Quadro 20). Em alguns casos, os mesmos toques da seleção são utilizados, sendo diferenciados apenas pelo local de ocorrência (área livre ou sobre o objeto).

Quadro 20 – Gestos para desselecionar

Desselecionar		
	Individual	Conjunto
3D Modeling App		
OnShape 3D CAD		
Wuweido	 ou 	
Qubism		
SDF 3D		
GnaCAD 3D		
Three.js		
AutoCAD		
UVCAD		 ou 
LEGENDA  GESTOS DE TOQUE  GESTOS DE ARRASTE  FERRAMENTAS  GIZMO		

Fonte: a autora (2022)

Edição: mover, rotacionar e escalonar

Para as edições ocorre um predomínio da utilização de gestos de arraste, assim como a necessidade de utilização de ferramentas (Quadro 21). Em alguns casos é necessário selecionar várias ferramentas para conseguir editar o objeto.

O único aplicativo que não necessita a seleção de ferramentas para a edição é o 3D Modeling App que utiliza os mesmos gestos de navegação na edição dos sólidos. Isso ocorre, pois após a seleção os gestos parecem deixar de ser associados aos movimentos da câmera e passam a ser associados as alterações do objeto selecionado. Assim como descrito na seleção de objetos, este aparenta ser outro caso de utilização das NUI sem a adaptação de interfaces gráficas.

Quadro 21 - Gestos para editar

Edição			
	Mover	Rotacionar	Escalonar
3D Modeling App	 ARRASTE SIMPLES	 ARRASTE DE DOIS DEDOS COM ROTAÇÃO	 ARRASTE PINÇA
OnShape 3D CAD	 FERRAMENTA EDIÇÃO +  FERRAMENTA TRANSFORMAÇÃO  +  ARRASTE SIMPLES	 FERRAMENTA EDIÇÃO +  FERRAMENTA TRANSFORMAÇÃO  +  ARRASTE SIMPLES	 FERRAMENTA EDIÇÃO +  FERRAMENTA TRANSFORMAÇÃO  +  ARRASTE SIMPLES
Wuweido	 GIZMO +  ARRASTE SIMPLES	 GIZMO +  ARRASTE SIMPLES	 GIZMO +  ARRASTE SIMPLES
Qubism	 FERRAMENTA MOVER +  ARRASTE SIMPLES	 FERRAMENTA ROTACIONAR +  ARRASTE SIMPLES	
SDF 3D	 FERRAMENTA TRANSFORMAÇÃO +  FERRAMENTA MOVER +  FERRAMENTA EIXO  ARRASTE SIMPLES	 FERRAMENTA TRANSFORMAÇÃO +  FERRAMENTA ROTACIONAR +  FERRAMENTA EIXO  ARRASTE SIMPLES	 FERRAMENTA TRANSFORMAÇÃO +  FERRAMENTA ESCALA +  FERRAMENTA EIXO  ARRASTE SIMPLES
GnaCAD 3D	 FERRAMENTA MOVER +  TOQUE SIMPLES +  ARRASTE SIMPLES +  TOQUE SIMPLES	 FERRAMENTA ROTACIONAR +  TOQUE SIMPLES +  ARRASTE SIMPLES +  TOQUE SIMPLES	 FERRAMENTA ESCALA +  TOQUE SIMPLES +  ARRASTE SIMPLES +  TOQUE SIMPLES
Three.js	 FERRAMENTA MOVER +  GIZMO +  ARRASTE SIMPLES	 FERRAMENTA ROTACIONAR +  GIZMO +  ARRASTE SIMPLES	 FERRAMENTA ESCALA +  GIZMO +  ARRASTE SIMPLES
AutoCAD	 FERRAMENTA MOVER +  ARRASTE SIMPLES +  TOQUE SIMPLES +  ARRASTE SIMPLES +  TOQUE SIMPLES	 FERRAMENTA ROTACIONAR +  ARRASTE SIMPLES +  TOQUE SIMPLES +  ARRASTE SIMPLES +  TOQUE SIMPLES	 FERRAMENTA ESCALA +  ARRASTE SIMPLES +  TOQUE SIMPLES +  ARRASTE SIMPLES +  TOQUE SIMPLES
UVCAD	inconclusivo	inconclusivo	inconclusivo
<p>LEGENDA</p>  GESTOS DE TOQUE  GESTOS DE ARRASTE  FERRAMENTAS  GIZMO			

Fonte: a autora (2022)

A seguir são comparados os temas e mecanismos de *feedback*.

O aplicativo apresenta um tema, padrões visuais e mecanismos para *feedback*?

Foram identificados seis mecanismos utilizados nos aplicativos que servem de *feedback* ao usuário, são eles: troca de cor após seleção, alterações dos modelos sem a utilização de outros mecanismos em tempo real, aparecimento do *gizmo*, indicação de deslocamento via tracejado, transparência no deslocamento e aparecimento de cotas. Apesar do aparecimento de *banners*, que poderiam ser considerados como *feedback* em alguns aplicativos, estes são considerados na prevenção de erros, pois neste item são avaliados *feedbacks* apenas de seleção e edição. O Quadro 22 apresenta os mecanismos utilizados em cada aplicativo.

Quadro 22 – Mecanismos de *feedback*

Feedback						
	Troca de cor	Tempo real	Gizmo	Tracejado	Transparência	Cotas
3D Modeling App						
OnShape 3D CAD						
Wuweido						
Qubism						
SDF 3D						
GnaCAD 3D						
Three.js						
AutoCAD						
UVCAD						
Legenda						
 utiliza  não utiliza						

Fonte: a autora (2022)

É possível observar, a maior incidência de *feedbacks* a partir da utilização de mudança de cor nos objetos selecionados. Mais de cinco dos nove aplicativos, utilizam

edições em tempo real que pode ou não ser acompanhada de *gizmo*, enquanto o restante utiliza outros artifícios para indicar as edições durante a sua execução.

Temas e padrões

A maior parte dos aplicativos apresentam a mesma paleta de cores durante toda a navegação, apenas dois: UVCAD e Wuweido, possuem telas iniciais com cores diferentes. Além das cores, é comum a utilização de ícones usuais como setas para refazer e desfazer, disquetes para salvamento, lápis para a edição. Apps como: OnShape 3D, Wuweido e Three.js, possuem sistema de navegação largamente utilizados por programas gráficos tridimensionais.

Como o aplicativo é organizado quanto a clareza, quantidade e qualidade das informações? São utilizados sensores e/ou atalhos?

Clareza, quantidade e qualidade de informações na tela

Em geral todos aos aplicativos apresentam poucas informações na tela, pois apresentam menus retráteis para armazenar diferentes tipos de ferramentas, o que aumenta a área útil para a edição. Alguns aplicativos possuem a possibilidade de expansão da *view* ao acessar a tela cheia, contudo no site do editor Three.js na versão móvel não funciona.

Atalhos

Dos nove aplicativos investigados, quatro possuem atalhos funcionais em dispositivos móveis. Não há uma forma comum de atalho a todos os aplicativos, alguns utilizam tocar e segurar para acessar um menu de opções, outros permitem a ativação da ferramenta a partir da seleção de um primeiro objeto ou desenho. E por último o aplicativo AutoCAD permite o acesso a ferramentas pela digitação de seu nome em uma barra no menu inferior da tela.

Sensores

O único sensor observado nos aplicativos foi o giroscópio para a mudança de orientação da tela de vertical para horizontal e vice-versa, esta utilização foi percebida em quatro dos nove apps analisados.

O aplicativo apresenta ferramentas para a prevenção de erros e facilitação na entrada de dados?

Prevenção de erros

As formas comuns a todos os aplicativos analisados de prevenção de erros, são os botões desfazer e refazer. Os aplicativos 3D Modeling App e OnShpe 3D CAD possuem vídeos tutoriais que apresentam suas ferramentas e interface gestual. Além destas há a utilização de banner e/ou palavras e frases que indicam os próximos passos.

Entrada de dados

É feita na maior parte dos casos pela utilização de teclado virtual numérico, no entanto no aplicativo 3D Modeling App e AutoCAD é possível ditar o comando com o microfone que aparece no teclado. Os aplicativos Qubism e SDF 3D não apresentam possibilidade de entrada de dados.

4.3.11. Configuração das classes de problemas

Para identificar a classe de problemas que esta pesquisa visa enfrentar, é necessária a compreensão do todo envolvido. Partindo do problema de pesquisa, foi possível identificar a necessidade de entendimento do *mobile learning* como área de estudo. Neste ponto, ficou evidente a complexidade e as diferentes abordagens encontradas na pesquisa e desenvolvimento do *m-learning*.

Por este motivo, o estudo passou a investigar o desenvolvimento de *mobile learning*, como forma de encontrar suporte para a adaptação do HyperCAL^{3D} à tecnologia móvel. A revisão sistemática de literatura deixou evidente a utilização de modelos de aceitação da tecnologia como ponto primordial para este desenvolvimento. No entanto, estes modelos possuem aproximação com o desenvolvimento de *software* e não apresentam diretrizes para o Design de soluções. Assim, o estudo aproximou as necessidades identificadas com o HyperCAL^{3D} *desktop* e as heurísticas de usabilidade e encontrou critérios para a análise de aplicativos similares.

As análises dos aplicativos similares trouxeram esclarecimentos interessantes, em especial, sobre como as interfaces gestuais são utilizadas em aplicações

tridimensionais e bidimensionais. Além da utilização de recursos, próprios da tecnologia móvel, como sensores, câmeras e microfones. Estes fatores podem influenciar a aceitação dos aplicativos por seus usuários.

HyperCAL^{3D} é o resultado de anos de pesquisa e desenvolvimento do grupo ViD. Sua utilidade e eficiência no ensino já foi comprovada pelo uso e melhorias ao longo dos anos. Portanto, sua adaptação ao *mobile learning* esbarra apenas no fator tecnológico e próprio dos dispositivos. A partir deste pensamento, seria possível compreender a classe de problemas desta pesquisa como “a adaptação tecnológica de ferramentas de ensino a novas tecnologias”.

No entanto, após a análise dos aplicativos similares, foi possível compreender que a classe de problemas, devido à natureza do HyperCAL^{3D}, pode ser direcionada para aplicativos destinados a modelagem 3D e desenho técnico 2D, pois se por um lado faltam recursos como *mouse* e teclado para a interação, por outro abundam recursos não utilizados como gestos, sensores e microfone, que de acordo com os estudos apresentados podem proporcionar uma melhor aceitação dos aplicativos, pois tendem a melhorar sua usabilidade.

Portanto, a classe de problemas desta pesquisa é “usabilidade em aplicativos de modelagem 3D e desenho 2D”.

4.4. PROPOSIÇÃO DE ARTEFATOS PARA RESOLUÇÃO DO PROBLEMA

A adaptação parte do processo de ajustar uma coisa à outra. Portanto, para que uma adaptação seja bem-sucedida, é necessária a compreensão tanto do que será adequado, quanto ao que será ajustado. Por esse motivo, para a melhor adaptação do HyperCAL^{3D} à tecnologia móvel, e ao *mobile learning*, é importante ajustar as ferramentas existentes no programa aos recursos disponíveis no dispositivo.

Além disso, como pode ser observado na revisão sistemática de literatura e no cruzamento teórico desta pesquisa, alguns aspectos devem ser considerados para a aceitação do HyperCAL^{3D} *mobile*. Portanto, para que a adaptação se aproxime ou solucione o problema desta pesquisa, ela precisa:

- Ajustar a navegação, seleção, desseleção e edição da *viewport* 3D e épura às interfaces gestuais;
- Alinhar os mecanismos de *feedback*, temas e padrões ao HyperCAL ^{3D} *desktop*;
- Adaptar a quantidade de informações ao tamanho reduzido das telas;
- Utilizar, quando possível, atalhos e sensores;
- Minimizar os erros dos usuários;
- Facilitar a entrada de dados.

4.5. PROJETO E DESENVOLVIMENTO DO ARTEFATO

Anteriormente à descrição das etapas de desenvolvimento, é necessária a explicação de como foi elaborada a versão para internet do HyperCAL ^{3D}, pois esta serve de base para o desenvolvimento *mobile*. Vale ressaltar que esta pesquisa representa apenas uma parte do desenvolvimento da versão móvel do aplicativo e que, após conclusão do app, este poderá ser compilado para os dispositivos móveis IOS e android, ou ser acessado via navegador.

HyperCAL ^{3D} web

A versão web foi iniciada em 2020, pelo professor Dr. Sérgio Leandro Santos, membro do grupo de pesquisa ViD, que desenvolveu a interface e algumas funcionalidades do programa.

No início de 2021, a autora passou a fazer parte do time de produção da versão para a internet do programa e desenvolveu as classes Tplano, Taresta e Tplano. Desta forma, os sólidos passaram a ser visualizados na *viewport* tridimensional, assim como passou a ser possível a criação de novos sistemas de referência, que posteriormente puderam ser rebatidos na épura.

É importante ressaltar que as versões web e a *mobile* não pretendem substituir a versão *desktop* e, sim, contemplar o maior número possível de recursos disponíveis. Como é difícil precisar quais as características dos dispositivos existentes, inicialmente não serão implementadas todas as funcionalidades do *software desktop*

para a versão web e *mobile*. Foram implementadas as funcionalidades de abertura de arquivos com a extensão H3D, oriunda do HyperCAL ^{3D}, inserção de vértices e faces, mudança no sistema de projeção, seleção de arestas e faces, visualização 3d e épura em diferentes configurações. Posteriormente novas funções poderão ser inseridas no aplicativo.

Para versão web o HyperCAL ^{3D} precisou ser totalmente reescrito com o auxílio do *framework* Quasar Vue, responsável pela organização da interface e a biblioteca Three.js para o desenvolvimento da *viewport* 3D e épura. A seguir são detalhadas as tecnologias utilizadas para a versão web e que será adaptada para versão *mobile*.

Quasar é um *framework* de código aberto derivado do Vue.js, utiliza apenas uma base de código que pode ser compilada para diferentes plataformas, como web browsers, aplicações moveis e aplicativos *desktop*. Além da possibilidade de criação para diferentes dispositivos como androide, IOS, Mac, Windows e Linux. (STOENESCU, 2015)

Vue.js também é um *framework* de código aberto desenvolvido em Javascript, direcionado ao desenvolvimento de interfaces de usuários. É utilizado na criação de páginas únicas, o que significa que o carregamento da página é feito apenas uma vez e seus componentes podem ser carregados dinamicamente e inseridos à medida que são necessários (YOU, 2014). As linguagens Html ¹² e CSS ¹³ são utilizadas na criação dos componentes da página, assim como a página principal, no entanto seguem uma estrutura própria do Vue.js

Three.js é uma biblioteca Javascript utilizada para criação e visualização de gráficos tridimensionais diretamente no *browser* de internet sem a necessidade de instalação de *plugins* de terceiros, pois utiliza a API (*Application Programming Interface*) WebGL ¹⁴ (CABELLO, 2010).

Para a utilização das tecnologias citadas é utilizado o node.js, um *software* de código aberto baseado em eventos assíncronos, executados a partir de requisição e

¹² HTML é uma linguagem utilizada para a construção de páginas para web que podem ser interpretadas pelos navegadores.

¹³ CSS *Cascading Style Sheets* é um código utilizado para adicionar estilo a página da internet como cor, tamanho, bordas entre outros.

¹⁴ WebGL é uma API do novo elemento Canvas disponível no HTML5 que permite a renderização de gráficos 2D e 3D diretamente no browser.

não na ordem do *script*. O programa utiliza o interpretador V8 do Google, como uma máquina virtual Javascript, que transforma o código em linguagem binária antes de executá-lo, desta forma acelera o desempenho da aplicação. Possui um gerenciador de pacotes que o confere potencial de reutilização em inúmeras situações diferentes (DAHI, 2009).

Como já citado, a versão para a internet não pretende substituir a versão *desktop* do programa e sim visa fornecer mais uma possibilidade de acesso a algumas ferramentas e recursos do *software*. Em um futuro próximo, todas as versões poderão ser conectadas de forma a se complementarem.

Para tanto, será necessário o desenvolvimento de uma tela de login, que permitirá que os usuários da ferramenta possam acessar seus arquivos online nas diferentes plataformas existentes. Eles poderão modelar os sólidos, na versão *desktop* e acessá-los no celular ou vice-versa. A versão para internet já conta com a possibilidade de abrir arquivos desenvolvidos na versão *desktop* e salvar os arquivos na mesma extensão, no entanto somente de forma local.

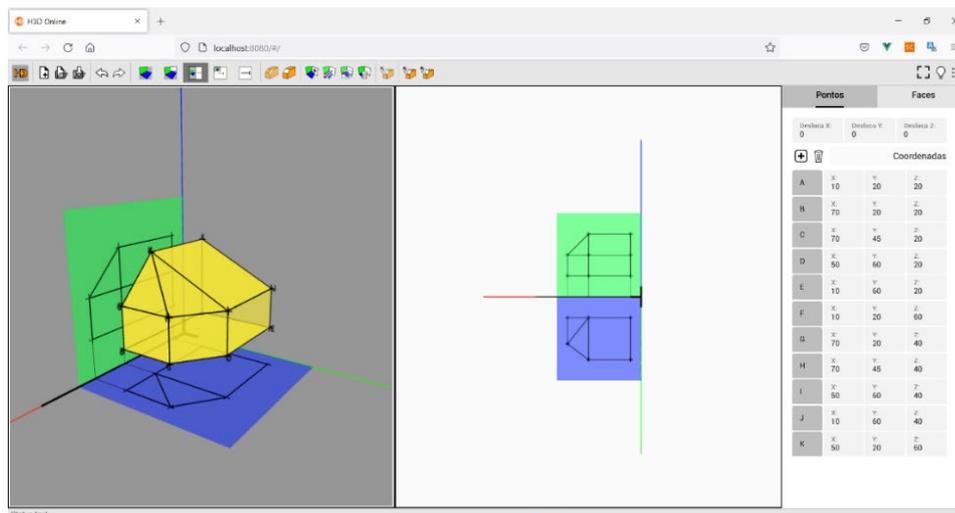
Para melhor organização do relatório, inicialmente são apresentadas as funcionalidades do programa gráfico na versão web, posteriormente é esquematizada sua arquitetura e, por fim, as implementações necessárias ao *mobile learning* que atendem as análises do capítulo 4 desta pesquisa.

4.5.1. Funcionalidades da versão web

A primeira etapa na criação de um sólido é a inserção dos pontos que dão origem às suas faces. Para tanto, é necessário abrir um menu sanduiche localizado no canto superior esquerdo da interface, que apresenta duas *tabs*, uma para a criação dos pontos e a outra para a criação das faces.

Na *tab* Pontos, é possível criar pontos a partir de suas coordenadas, assim como deletá-los. Quando um ponto é inserido, passa automaticamente a ser visualizado na *viewport* 3D. As projeções do ponto recém introduzido, podem ser visualizadas simultaneamente em 3D e na Épura. Conforme ilustra a Figura 60.

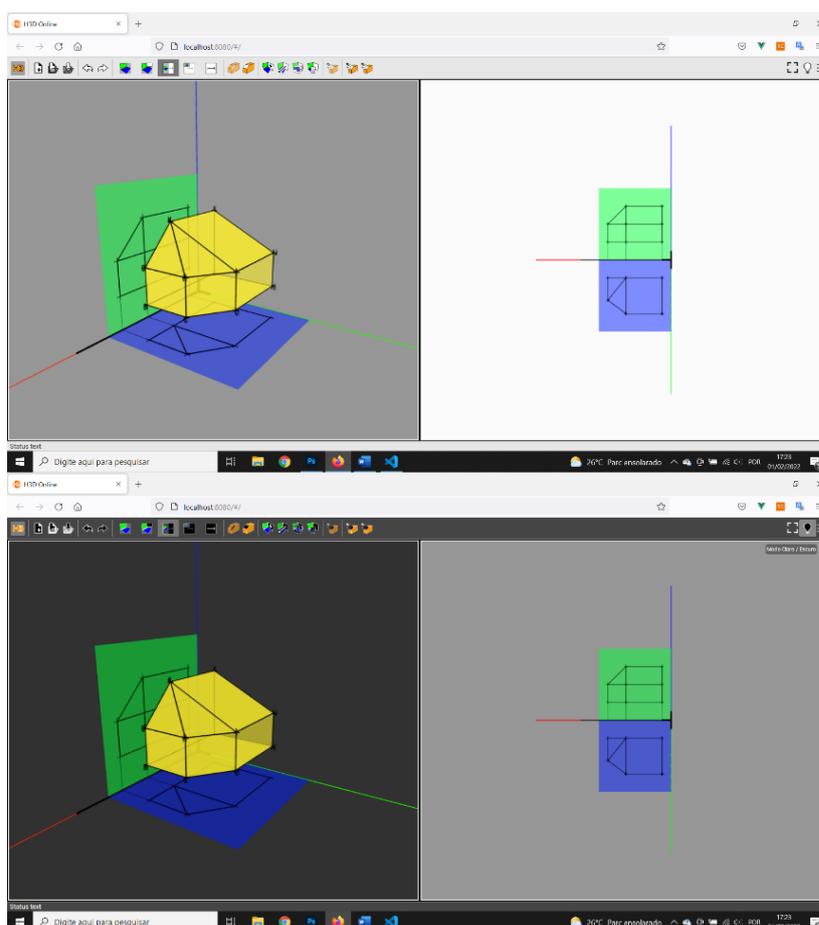
Figura 60 - Ferramentas de criação de sólidos no HyperCAL 3D web



Fonte: a autora (2022)

Ao lado do menu sanduiche, encontram-se dois botões responsáveis por alterações gerais na interface do programa. O primeiro, representado por uma lâmpada é capaz de transformar a interface de modo escuro para claro e vice-versa (Figura 61).

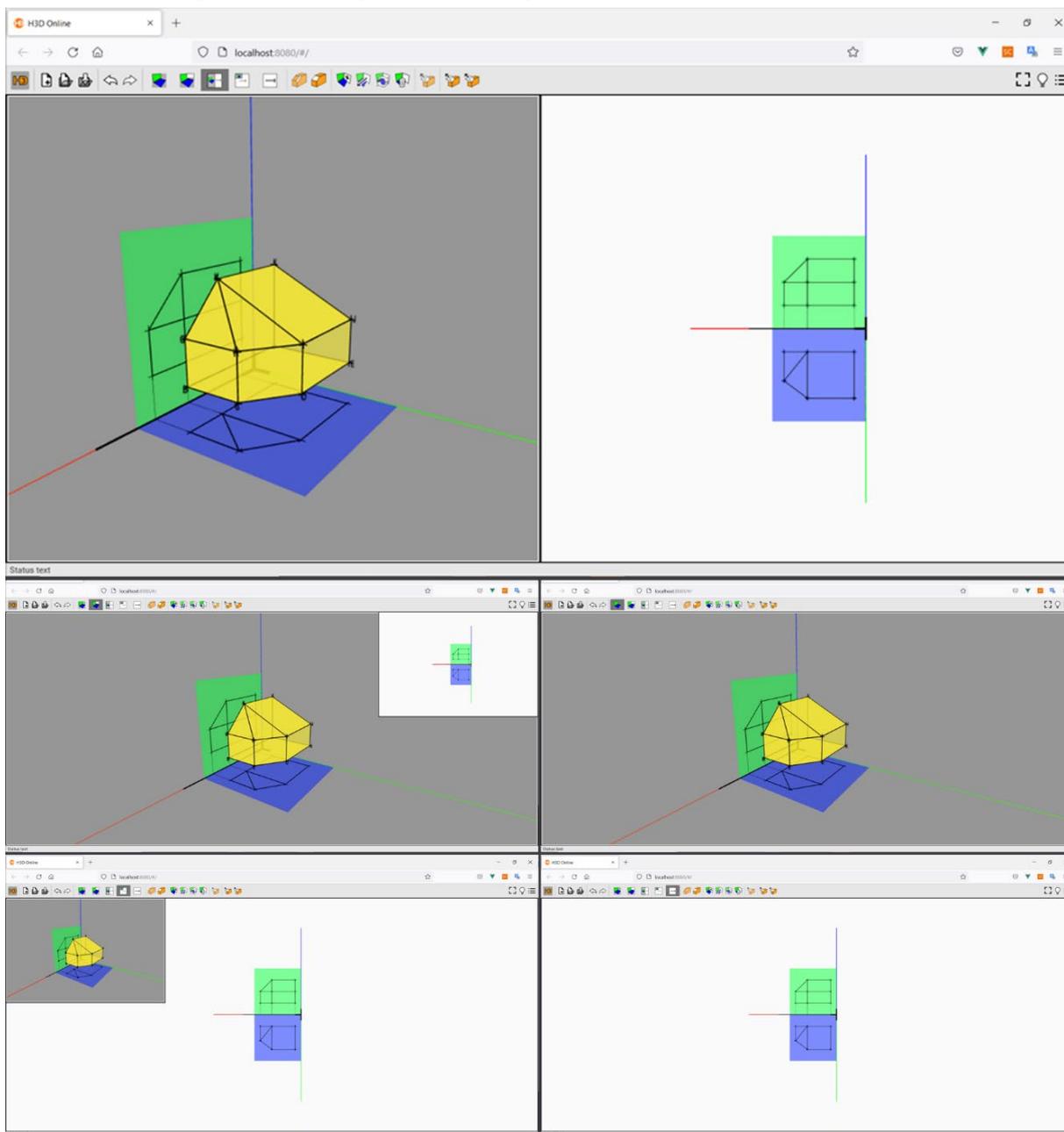
Figura 61 - Modo Claro/ Escuro HyperCAL 3D web



Fonte: a autora (2022)

O segundo botão, permite expandir o site e esconder as barras do navegador e do Windows. Além destas funcionalidades, é possível modificar a dinâmica de visualização da *viewport* tridimensional e é pura, podendo mantê-las individualmente na tela, juntas dividindo meio a meio o espaço disponível ou com tamanhos diferentes ao mesmo tempo. A Figura 62 demonstra estas possibilidades.

Figura 62 - Alteração da visualização das views no HyperCAL 3D web

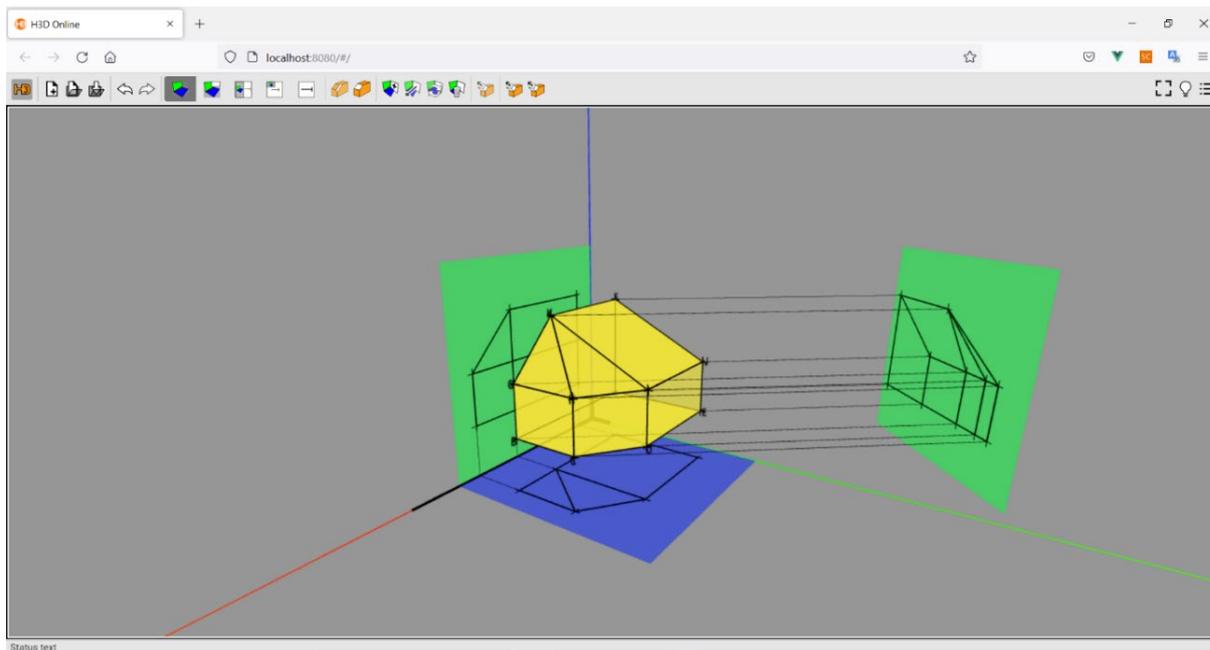


Fonte: a autora (2022)

As faces e arestas do sólido podem ser selecionadas, porém até a finalização desta pesquisa não podem ser extrudadas ou modificadas via seleção. É possível ainda, criar planos de projeção clicando no botão de mudança de sistema de projeção

e selecionando um plano pai a partir da *viewport* tridimensional. O novo sistema de referência é inserido a noventa graus do plano selecionado e pode ser arrastado e posicionado em tempo real, enquanto suas projeções são recalculadas (Figura 63). Com um segundo clique é finalizado o arraste e o plano é também inserido na *épura*. A visualização em tempo real na *épura*, assim como a possibilidade de criação do sistema a partir dela e a edição dos planos após a sua inserção, até a conclusão desta pesquisa não haviam sido implementadas.

Figura 63 - Mudança de sistema de projeção



Fonte: a autora (2022)

Como citado anteriormente, também é possível salvar e abrir arquivos com a extensão H3D oriunda do HyperCAL ^{3D} *desktop*, porém somente localmente.

4.5.2. Arquitetura da versão web HyperCAL 3D

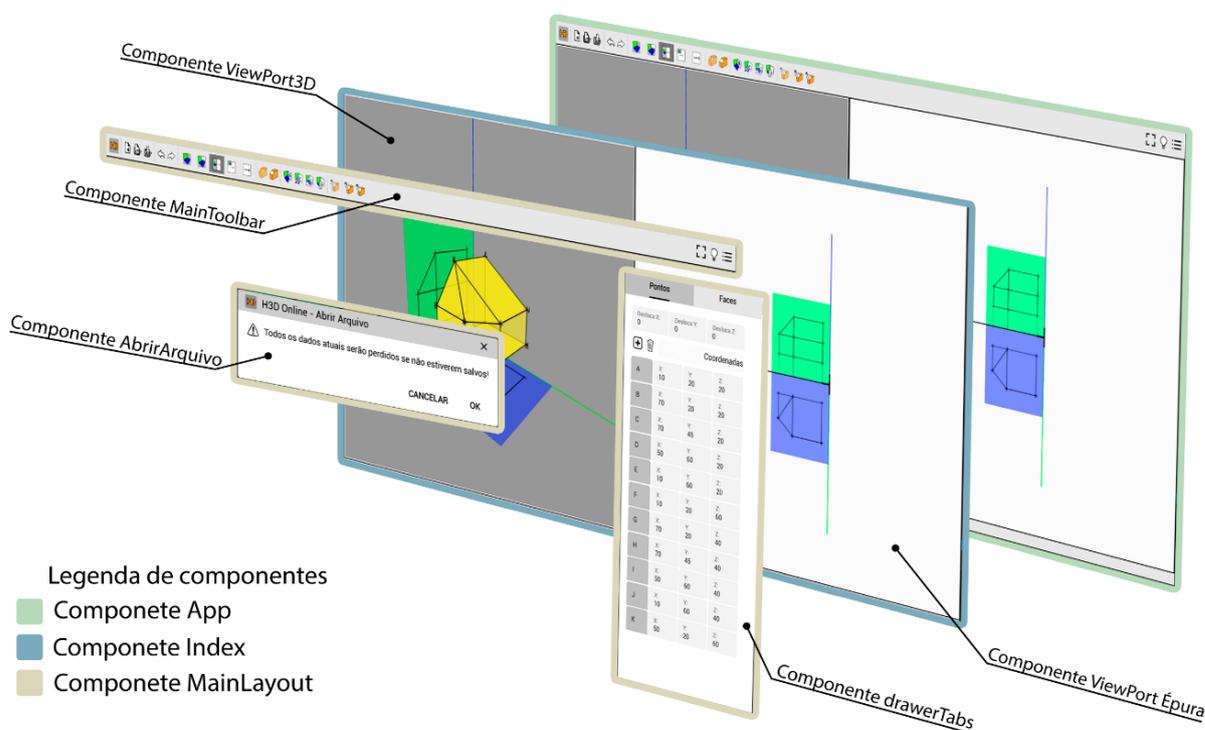
Como já citado, o HyperCAL ^{3D} web está sendo desenvolvido com o *framework* Quasar, que permite que o programa seja compilado para diferentes dispositivos, o *framework* Vue, para a criação de páginas únicas com componentes carregados por demanda e a biblioteca gráfica Three.js, responsável pela visualização dos elementos tridimensionais e a manipulação em tempo real das *views*.

Para a adaptação do programa, estão sendo utilizadas três estratégias, as duas primeiras possuem ligação com os *frameworks* Quasar e Vue, que são: a utilização de componentes e de *stores*. E a terceira, sistema de classes, possui relação com a programação baseada em objetos. Estas estratégias são explicadas a seguir.

Utilização de componentes

De forma geral, os componentes podem ser desde botões, menus completos e o *layout* completo da página. Este sistema, permite que o programa carregue, por demanda, apenas as partes necessárias ou, na nomenclatura correta, componentes necessários, sem precisar recarregar toda a página, como ocorre nos sites baseados apenas em HTML e CSS. Para o desenvolvimento de aplicações desta natureza, há apenas um arquivo HTML, que serve para a comunicação com os navegadores e carregar o componente principal do site, normalmente chamado de App. Este componente possui as rotas ou caminhos para a inserção dos componentes principais. A Figura 64 identifica os componentes principais do HyperCAL 3D web.

Figura 64 - Estrutura principal do HyperCAL 3D web



Fonte: a autora (2022)

De forma simplificada, o componente App possui uma única *div*¹⁵ com um componente que identifica as rotas para a inserção dos dois componentes principais, que formam a interface do programa: Index e *MainLayout*. O Index é responsável pela inserção e manipulação de outros dois componentes: *Viewport* 3D e *Viewport* Épura. Ele possui as rotinas para as mudanças de tamanho e posicionamento que podem ser acessadas pela barra de ferramentas principal. O *MainLayout* posiciona os

¹⁵ Divisões virtuais realizadas para a organização do layout de um site.

componentes barra de ferramentas, *cards* de abertura, salvamento e novo arquivo e menu com as *tabs* “pontos” e “planos”, que também são componentes.

Utilização de Stores

O site também conta com *stores*, que de forma simplificada, são arquivos Javascript que armazenam os estados, mutações e ações que podem afetar as variáveis acessadas pelos componentes. Em outras palavras, cada evento, como por exemplo um clique em um botão, chama uma ação da *store*, que desencadeia uma mutação no estado de determinada variável. Desta forma, o site pode acessar as variáveis a partir de diferentes componentes, sem precisar replicá-las. É possível criar quantas *stores* forem necessárias, mas no HyperCAL ^{3D} web são utilizadas três: UI_store, h3d_store e Three_store.

A UI_store é responsável pelos estados, mutações e ações da interface, enquanto a h3d_store, armazena, entre outros, os pontos, as faces e os planos de projeção. Assim, quando um novo ponto é inserido seus dados são armazenados no *array* de pontos da h3d_store. Na Three_store são guardadas as interações da *Viewport* 3D e *épura*, como por exemplo o plano selecionado para a criação do novo sistema de projeção.

Programação baseada em objetos

As classes são responsáveis pela criação dos objetos: pontos, arestas, faces e planos de projeção nas *views*. Utilizando uma metáfora, são como fábricas que possuem dados e procedimentos necessários para a criação de objetos do mundo real. As classes utilizadas no HyperCAL ^{3D} web são: Tponto, Tface, Taresta e Tplano.

Classe Tponto: é responsável por criar os pontos e inserir na cena da *viewport* 3D. Ela identifica cada ponto e insere marcadores tridimensionais na posição referente as suas coordenadas.

Classe Tface: cria a representação gráfica das faces do sólido, para tanto identifica quais pontos que compõem a face e se ela possui furos.

Classe Taresta: identifica as arestas do sólido a partir de suas faces, avalia se há repetição de arestas e projeta linhas nas arestas do sólido na *viewport* 3D.

Classe Tplano: cria as projeções do sólido (pontos e arestas) e as linhas de chamada nos sistemas de referência, além de recalculá-las enquanto o plano está sendo arrastado na *view*.

4.5.3. Viewport 3D e Épura

As *views* são componentes que utilizam o *canvas*¹⁶ para a inserção da renderização das cenas tridimensionais desenvolvidas com a biblioteca Three.js. Estas cenas possuem três métodos principais, responsáveis pela maioria de suas ações: *main*, *update* e *render*.

Main: é responsável pela inicialização da cena e colocação dos elementos principais que a compõem, como: primeiro diedro, câmera e seus movimentos orbitais e luz. Esse método é chamado quando o programa é acessado pela primeira vez e é responsável também pela requisição do método *render* (*webGLRenderer*).

Update: este método analisa as alterações realizadas na *h3d_store*, quando um arquivo é aberto ou um novo ponto é inserido, ele insere cada ponto, face, aresta e plano na tela. Possui em sua construção, um método capaz de identificar o que já foi inserido, para evitar objetos repeditos.

Render: as imagens colocadas na tela só podem ser geradas a partir deste método. De forma simplificada, ele calcula o tamanho do *canvas*, as luzes e objetos, baseados na posição da câmera. Desta forma, é capaz de desenhar na tela as imagens da cena idealizada. No HyperCAL^{3D} web, possui ainda uma função de animação, uma função *callback* que faz com que o método reinicie até que todas as interações com a *view* terminem. Isso permite que as imagens sejam renderizadas em sequência, dando a impressão de movimento, a partir das interações.

Além dos métodos principais, o site possui ainda, o método *raycaster*, que calcula a posição do mouse no ambiente tridimensional, partindo do *canvas* e da posição da câmera. E um método que identifica qual navegador está sendo utilizado.

Para permitir a interação, são utilizados eventos que identificam: sobreposições baseadas no *raycaster*, cliques simples e duplos, toques e arrastes do mouse. Estes eventos, permitem as manipulações pelo usuário das cenas tridimensionais.

¹⁶ É um elemento do html 5 responsáveis pela área de renderização

Tanto a *viewport* 3D quando a *Épura* foram desenvolvidas com os mesmos recursos, no entanto na *Épura*, a navegação é feita apenas com rotações bidimensionais, diferente da *view* 3D onde é permitida a rotação em três dimensões.

4.5.4. Adaptação do HyperCAL^{3D} ao *mobile*

Como mencionado no item 3.5 deste relatório, a metodologia utilizada no desenvolvimento da adaptação foi a metodologia Scrum e devido ao extenso conhecimento dos orientadores desta pesquisa do programa HyperCAL^{3D}, estes foram considerados nos papéis de Scrum *Masters* e *Product Owner* ao mesmo tempo.

Vale ressaltar que a lista de necessidades para a adaptação é extensa e continuará após a conclusão desta pesquisa. No entanto, as etapas desenvolvidas até o momento já podem ser aplicadas de forma local.

O desenvolvimento partiu de seis etapas definidas como *product backlog*: implementação das classes *Tface*, *Taresta* e *Tplano*; implementação da *épura*; correção do desempenho e adaptações as proposições do artefato desta pesquisa. Vale ressaltar que, mesmo que os primeiros ciclos de desenvolvimento não tivessem relação direta com o desenvolvimento desta pesquisa, foram de grande importância para a realização dela, pois sem eles não seria possível a realização do ciclo final. Além disso, devido à falta de conhecimento da autora em programação, da arquitetura do HyperCAL^{3D}, do *framework* Quasar Vue e da biblioteca Three.js, no início deste projeto, estes ciclos foram fundamentais para o desenvolvimento da autora.

Durante cada *Sprint*, foram utilizados também alguns princípios da metodologia XP, enquanto outros, como metáfora e programação em pares, foram dispensados, por representarem aplicações para equipes de desenvolvimento. Os princípios “propriedade coletiva” e “padrão de codificação” foram pensados juntamente com a refatoração, na qual a busca por um código mais limpo e a adição de comentários foram acrescentados de forma a tornar o código mais claro para futuros desenvolvedores. As quarenta horas de trabalho semanais também foram dispensadas, pois este projeto não representa um regime de trabalho convencional. Portanto, para cada ciclo de *sprint* foram pensados e desenvolvidos os seguintes princípios.

- O Jogo de Planejamento – determinação das prioridades e estimativas do escopo da implementação.

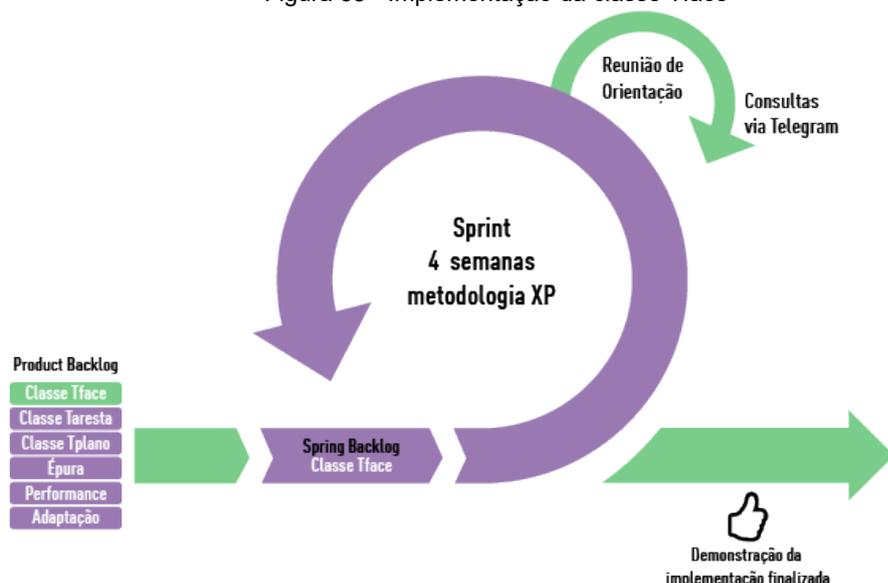
- Pequenas versões — programar em pequenas etapas contínuas e constantes até atingir a implementação do *Sprint backlog*.
- Design simples — descomplexificar etapas com programação simples.
- Teste — realizar testes constantemente para evitar que erros sejam acumulados em etapas posteriores.
- Refatoração – remover a duplicidade de código, melhora a comunicação, simplifica e adiciona flexibilidade.
- Integração contínua — integrar o sistema toda vez que uma tarefa é concluída.
- Cliente presente — tirar dúvidas com os orientadores para evitar perda de tempo.

A seguir são apresentados os ciclos de Sprint realizados, assim como as formas de implementação de cada princípio.

Implementação da classe Tface

Como já foi dito, devido à inexperiência da autora com as tecnologias envolvidas na criação do HyperCAL ^{3D} web, o primeiro *sprint*, que correspondia a criação da classe Tface, durou aproximadamente quatro semanas, do dia nove de março de 2021 até o dia primeiro de abril de 2021.

Figura 65 - Implementação da classe Tface



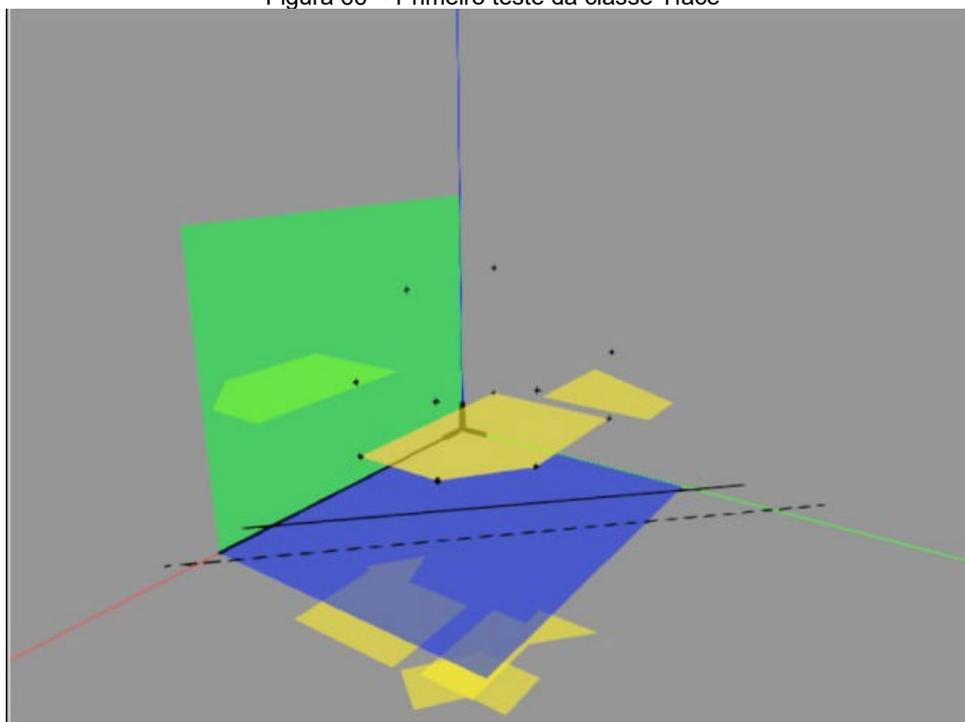
Fonte: a autora (2022)

Durante o desenvolvimento, foi necessária pelo menos uma reunião de orientação com o professor Dr. Sérgio Santos e diversas consultas via Telegram (Figura 65).

Para a implementação da classe, era necessária a identificação das coordenadas dos pontos que compunham a face a ser criada. Estas coordenadas foram utilizadas para a criação de uma *shape* (método que cria formas) que posteriormente foi adicionada, juntamente com o material para formar uma malha que foi inserida na *viewport*.

Nas primeiras tentativas de implementar o código, já foi possível enxergar as faces na cena, no entanto todas as faces apareciam na horizontal e separadas do objeto (Figura 66).

Figura 66 - Primeiro teste da classe Tface



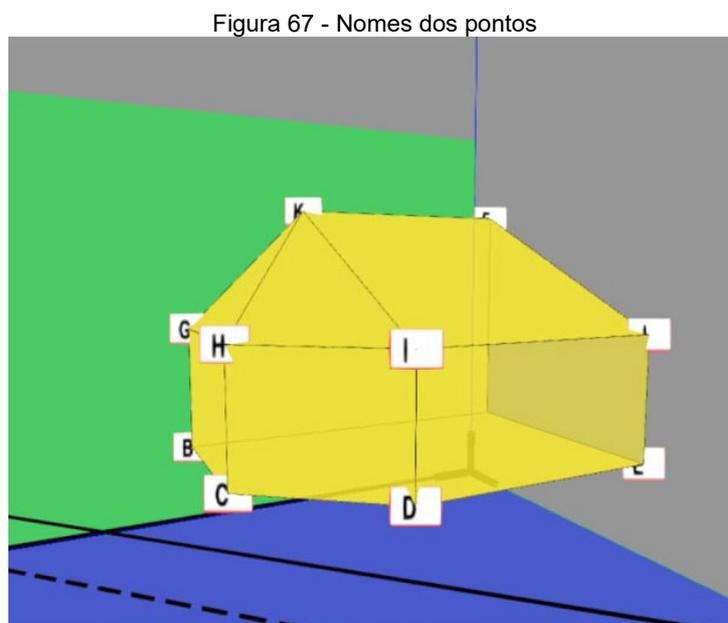
Fonte: a autora (2022)

Mesmo após uma extensa pesquisa sobre a possível causa do problema, não foram encontradas respostas. No entanto, a solução para o problema foi obtida através de uma diferença na cor do código. As cores das palavras *quaternion*¹⁷ estavam diferentes das demais variáveis e indicavam que ela é uma palavra reservada. Portanto, ao declará-la como variável, nada acontecia. Com a alteração do nome da variável o código funcionou.

¹⁷ É o quociente de dois vetores que indica a rotação dos objetos tridimensionais

Para identificar se as faces possuíam furos, foi implementado um laço de repetição que percorre o *array* de pontos da face, procurando identificar um hífen e seu index. Em caso de resposta afirmativa, o *array* é dividido em *array* face e *array* furo. Neste momento foram identificados dois problemas: a biblioteca Three.js não aceita a subtração de dois *shapes* (formas), somente um *path* (caminho) de um *shape*; E o segundo no HyperCAL ^{3D}, as faces são criadas em sentido anti-horário e os furos, em sentido horário, o que causava uma separação do furo e da face. Para resolver o problema, foi alterado o sinal do *quaternion* do furo.

Ainda no período de desenvolvimento da classe Tface, foram acrescentados os nomes dos pontos a classe Tponto. Inicialmente, foi inserido um retângulo branco no fundo do rótulo que atravessava, juntamente com as letras, as faces do sólido (Figura 67).



Fonte: a autora (2022)

Mesmo removendo os retângulos brancos, os nomes dos pontos ainda apareciam cruzando as faces do sólido. Para solucionar o problema, foi calculado um vetor entre o ponto central do sólido e o ponto inserido, que somado a um valor fixo de afastamento, direciona a posição final do rótulo.

Neste *sprint*, o foco foi a criação da classe Tface. Diferentes aprendizados foram adquiridos pela autora que, posteriormente, serviram de base para a execução dos outros ciclos. Como forma de demonstrar os processos e desafios encontrados durante a implementação deste ciclo, optou-se pela descrição narrativa dos fatos ocorridos durante o processo e, posteriormente, a apresentação sistematizada da

metodologia XP do desenvolvimento. O Quadro 23 sintetiza seus princípios e como estes foram utilizados neste *sprint*.

Quadro 23 - Metodologia XP no Sprint Tface

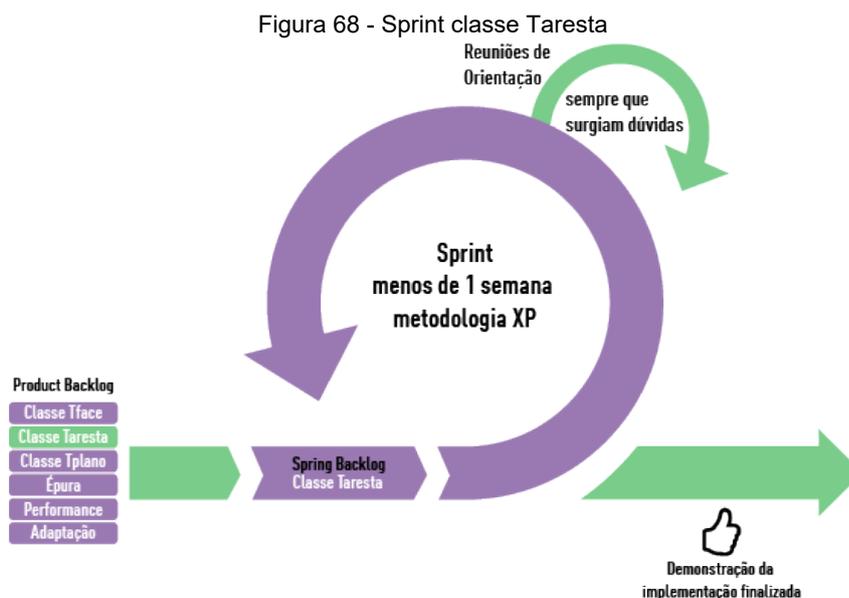
Princípio metodologia XP	Síntese no Sprint
O Jogo de Planejamento	<ul style="list-style-type: none"> • Buscar no <i>array</i> de pontos as coordenadas dos pontos que compõem as faces • Criar <i>shapes</i> com os pontos encontrados. • Transformar as shapes em malhas e inseri-las na <i>viewport</i> 3D • Identificar se as faces apresentam furos e separar em <i>arrays</i> faces e furos • Subtrair furos das faces • Apresentar resultado na <i>view</i>.
Pequenas versões	<ul style="list-style-type: none"> • Criar <i>shapes</i> com os pontos da face e utilizá-las para gerar a malha da face. • Identificar se existem furos na face e criar malhas furadas
Design simples	As implementações deste ciclo por serem relativamente simples, não apresentavam formas mais complexas de implementações, portanto se mantiveram simplificados do início ao final.
Teste	Durante todo o ciclo diferentes testes foram executados, testes de console, teste de visualização e navegação.
Refatoração	Devido a inexperiência da autora foram geradas muitas tentativas e erros, portanto ela optou por primeiramente fazer o código funcionar e posteriormente ajustá-lo.
Integração contínua	As etapas de desenvolvimento ocorreram de forma contínua, a única quebra foi a inserção dos nomes dos pontos, que era uma demanda existente. Porém, que foi implementada no meio do <i>sprint</i> para proporcionar um afastamento dos problemas do <i>quaternion</i> que ainda não haviam sido resolvidos.
Cliente presente	Durante todo <i>sprint</i> , diversas consultorias foram realizadas com os orientadores a fim de sanar dúvidas existentes.

Fonte: a autora (2022)

Implementação da classe Taresta

Para a criação da classe Taresta (Figura 68) era preciso identificar os pontos de cada face e juntá-los em pares. Inicialmente foi realizado um laço de repetição em cada face para a identificação das coordenadas dos pontos que as compunham. Como este comando havia sido implementado na classe Tface, inclusive com a identificação dos furos, estes valores foram resgatados do *array* faces da cena e enviados para a classe Taresta.

Desta forma, a classe inicialmente recebe um *array* com os índices dos pontos que compõem cada face ou furo. Estes são organizados em pares em um novo *array* de arestas. Para evitar a duplicação de arestas, antes da inserção as arestas são testadas, a partir da análise de seus pontos. Caso já exista uma aresta com os mesmos pontos, a mesma não é inserida no *array*, para evitar duplicidade.



Fonte: a autora (2022)

Além destes comandos que constituem o construto da classe, um método busca os índices de pontos de cada aresta e cria cilindros entre eles que são inseridos na cena. Posteriormente as arestas são armazenadas na store, a partir de um laço de repetição que as resgata da cena. O Quadro 24 apresenta a síntese da metodologia XP deste ciclo.

Quadro 24 – Metodologia XP no Sprint Taresta

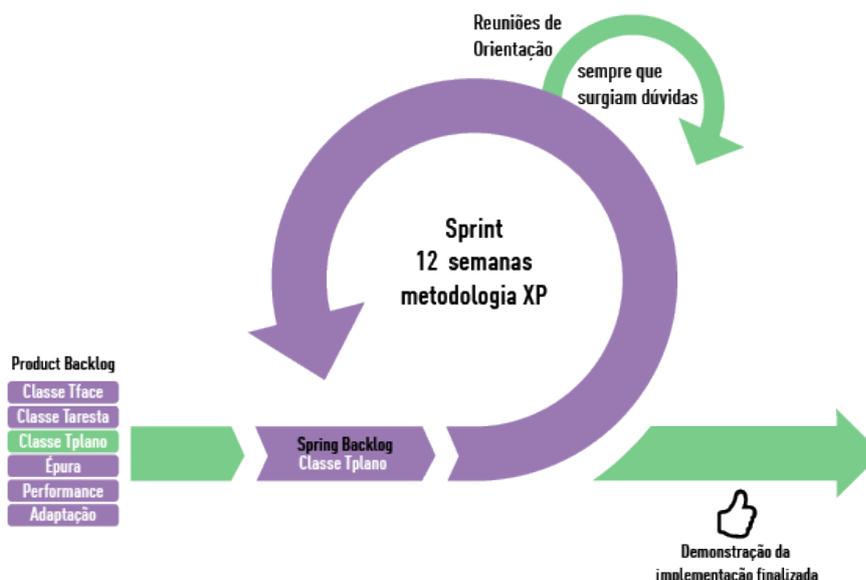
Princípio metodologia XP	Síntese no Sprint
O Jogo de Planejamento	<ul style="list-style-type: none"> • Identificar os pontos de cada aresta • Criar um <i>array</i> de arestas sem repetições • Gerar a geometria e malha das arestas • Inserir cada aresta na <i>view</i>.
Pequenas versões	<i>Array</i> com arestas Inserir arestas nas <i>views</i>
Design simples	O método que cria as arestas a partir de pontos já havia sido desenvolvido pelo professor Dr. Sérgio Santos
Teste	O <i>array</i> de arestas foi testado diversas vezes a partir do console, pois apresentava problemas de repetição de arestas. Foi identificado que a posição do <i>array</i> no looping estava causando o problema.
Refatoração	Foi realizada após a finalização da classe.
Integração contínua	As etapas progrediram de maneira contínua
Cliente presente	Os professores orientadores desta pesquisa acompanharam o progresso do trabalho.

Fonte: a autora (2022)

Implementação da classe Tplano

Na visão da autora, este foi um dos sprints mais difíceis e mais longos de implementar (Figura 69), devido à complexidade envolvida, ao *update* em tempo real, ao tamanho dos planos em relação às projeções, o rastreamento do mouse, entre outros.

Figura 69 - Sprint da classe Tplano



Fonte: a autora (2022)

Mesmo levando mais de quatro semanas, este foi considerado como um sprint único. A classe Tplano precisava de um plano com cor diferente do plano pai. Por exemplo, se o pai era azul o filho deveria ser verde; apresentar as projeções dos pontos e arestas em tempo real; criar as linhas de chamada em duas versões: deslocamento em tempo real e nos planos de projeção; ser inserido a noventa graus do plano pai; ser inserido dinamicamente e ter o tamanho alterado de acordo com o sólido projetado.

O primeiro passo para a criação da classe, foi habilitar a ferramenta “novo sistema de referência” e programar a classe para receber as informações de posição, rotação e cor do plano pai, após a seleção deste na *viewport* 3D. Para tanto, foi utilizado o método *Raytrace* que retorna para posição do mouse a partir do *Canvas* e identifica os objetos interseccionados.

Com os dados recebidos, a classe foi programada para inserir um plano (combinação de geometria, material e malha) na mesma posição, rotação acrescida de noventa graus no eixo x e de cor diferente da cor do plano pai do novo sistema de projeção.

Após a confirmação do funcionamento destas etapas, foi implementado um método na *viewport* 3D que identifica a projeção do mouse a partir da criação de um vetor normalizado (chamado *plane*), sobre o plano infinito no qual o pai está

posicionado. Assim, a posição do plano gerado pela classe poderia ser alterada em tempo real seguindo a posição da projeção do mouse.

Para que o plano se mantivesse sempre voltado para o sólido, foi adicionada a função *look at* para o centro dele. No entanto, isto ocasionou a inclinação do novo sistema, pois o eixo dele se encontrava em seu centro (sobre o plano pai) e o centro do modelo em uma altura diferente. Foram realizadas diversas tentativas para evitar que este fenômeno ocorresse. A solução encontrada para resolver o problema foi mover o vetor normalizado, que servia para a projeção do mouse de acordo com o plano pai, para a posição do centro do modelo. Desta forma, centro do modelo e eixo do plano filho passaram a ser coplanares e a função *look at* passou a direcionar o plano filho em apenas duas dimensões, impedindo a inclinação.

Após a resolução do problema, foi implementada a alteração do tamanho do plano para que este se adaptasse às dimensões do modelo tridimensional. Assim como um novo *EventListener* (escuta de eventos), para identificar um novo clique responsável por parar o arraste do plano. Os próximos passos seriam a codificação das projeções.

As projeções foram idealizadas a partir de um novo vetor normalizado alinhado e posicionado ao plano de projeção. Em um primeiro momento, os pontos do modelo são projetados um a um, salvos em um *array*, e suas coordenadas inseridas na posição de marcadores em formato de cruz e rotacionados de acordo com o plano. Durante o deslocamento do plano, a projeção dos pontos é recalculada e a posição dos marcadores alterada.

Para a inserção das projeções das arestas, são localizados, no *array* de projeções dos pontos, os índices salvos, em pares, no *array* de arestas e um cilindro, então, é inserido entre ambos. Da mesma forma, é realizado o cálculo das linhas de chamada durante o deslocamento do plano, porém são utilizados os pontos no espaço do modelo e suas respectivas projeções no plano.

Inicialmente, foi projetada a alteração das arestas e das linhas de chamada em tempo real, a partir de sua remoção e reinserção durante o arraste do plano. No entanto, durante os testes, várias arestas e linhas eram inseridas e nenhuma era removida. Assim, algumas tentativas visavam a alteração de suas malhas, o que se provou de difícil controle, pois, por exemplo: para escalonar a linha de chamada de

maneira precisa, é necessário alinhar o eixo de escalonamento com a linha, calcular a variação da distância e transformar o cálculo em percentual de escalonamento. Para resolver este obstáculo, o enfoque passou a ser direcionado para a geometria dos cilindros, pois ela poderia ser recalculada a partir dos mesmos pontos, ou de suas novas projeções. Isso significa, que a forma de cálculo seguiria a mesma base, porém a geometria da malha precisava ser modificada como um todo. Após alguns testes a solução mostrou-se satisfatória.

O último ponto de desenvolvimento deste *sprint* foi a alteração das linhas de chamada, pois, após o clique que libera o arraste, as linhas deveriam sair das projeções até a linha de terra e ir da linha de terra até as projeções do plano pai. Um novo vetor normalizado, que foi posicionado e rotacionado com o plano pai, permitiu projetar os pontos do sólido sobre ele e as projeções do plano filho sobre a linha de terra. Neste ponto, foram criadas linhas (cilindros) entre as projeções do plano filho e suas projeções na linha de terra e as projeções do plano pai até suas projeções também na linha de terra. O Quadro 25 apresenta a síntese da metodologia XP deste *sprint*.

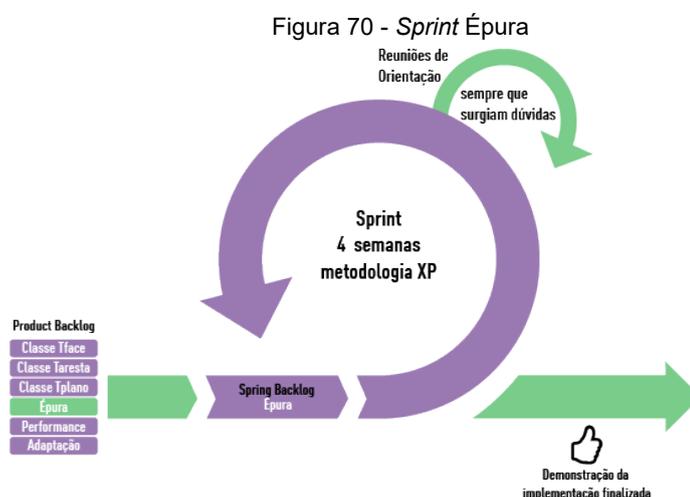
Quadro 25 - Metodologia XP *Sprint* Tplano

Princípio metodologia XP	Síntese no Sprint
O Jogo de Planejamento	<ul style="list-style-type: none"> • Selecionar os planos pai na <i>viewport</i> 3D e enviar sua posição, rotação e cor para a classe Tplano • Criar um plano na posição, rotacionado a 90 graus em x e de cor diferente ao plano pai • Arrastar o plano criado em tempo real enquanto ele aponta para o centro do modelo • Criar as projeções dos pontos e atualizá-las em tempo real de arraste • Inserir linhas de chamada e arestas nas projeções • Alterar linhas de chamada e arestas durante arraste do plano • Parar o arraste e substituir linhas de chamada por linhas de chamada no plano.
Pequenas versões	Diferentes versões de cada etapa foram implementadas na tentativa de subverter os problemas decorrentes das alterações em tempo real. Algumas se mostraram complexas e ineficientes e foram substituídas por versões mais simples e eficientes.
Design simples	O melhor design acabou por ser o mais simples, em especial, para o cálculo em tempo real das linhas de chamada e arestas.
Teste	As soluções encontradas neste <i>sprint</i> foram satisfatórias, no entanto demonstraram problemas de desempenho, devido a manipulação em tempo real, que ainda não haviam sido observados. Estes problemas foram abordados em um novo <i>sprint</i> .
Refatoração	A classe Tplano sofreu diversas refatorações, devido a quantidade significativa de testes para a solução dos problemas encontrados.
Integração contínua	As etapas progrediram de maneira contínua
Cliente presente	Os professores orientadores desta pesquisa acompanharam o progresso do trabalho.

Fonte: a autora (2022)

Implementação da épura

Para a criação da épura, uma rotina de rebatimento de planos a noventa graus precisava ser implementada. Esta deveria rotacionar o plano pai e seus descendentes e posteriormente rotacionar cada descendente da mesma forma. Na primeira tentativa de rotação um fato curioso foi detectado: apesar dos planos de projeção terem sido posicionados e rotacionados em tempo real, suas posições e rotações se encontravam zeradas na *viewport* 3D. Depois de algum estudo percebeu-se que, como a classe *Tplano* recebia as informações e internamente posicionava o plano, o objeto gerado na *view* não sofria alteração de posição e rotação e durante o arraste o mesmo ocorria, pois o plano do interior do objeto era rotacionado e não o objeto como um todo.



Fonte: a autora (2022)

Este fato levou à total reestruturação da classe *Tplano*, o que se mostrou ineficiente, pois o vetor normalizado que possui o método para a projeção dos pontos precisa ser alinhado a uma malha (*mesh*) e a classe *Tplano*, após a inserção na *view*, não configurava um plano e, sim, um novo tipo de objeto, o que invalidava as projeções. Outras tentativas foram realizadas para a resolução deste obstáculo, porém percebeu-se que a cada tentativa a classe *Tplano* ficava mais confusa. Portanto, optou-se por retomar a versão original da classe *Tplano* e buscar uma nova solução para o posicionamento do ponto de rotação do plano de projeção da épura. Por estes motivos, este *Sprint* acabou tomando mais tempo do que era cogitado no início de seu desenvolvimento. A Figura 70 demonstra o tempo final do *Sprint*.

A nova estratégia consistia em posicionar um objeto. Neste caso, foi utilizado um *axeHelper* (insere os eixos na *view*) no ponto central da linha de terra que compõe

o plano a ser rotacionado. A partir deste ponto, todo o sistema usa sua posição para realizar a rotação. Com a rotação resolvida, foi possível implementar a rotina de rotação e finalizar a Épura. No local da linha de terra dos novos sistemas de referência, constam os eixos. Porém, estes futuramente serão deslocados e a linha de terra inserida. O Quadro 26 sintetiza a metodologia XP na implementação da Épura.

Quadro 26 - Metodologia XP na implementação da Épura

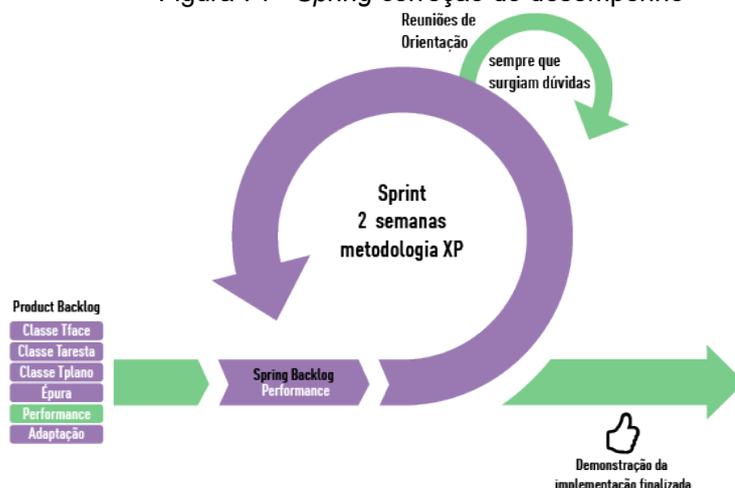
Princípio metodologia XP	Síntese no Sprint
O Jogo de Planejamento	<ul style="list-style-type: none"> • Rotina para a rotação dos planos • Reposicionar o ponto de rotação e reaplicar a rotina
Pequenas versões	Na tentativa de ajustar a rotação dos planos, diversas tentativas incorreram em erros e precisaram ser refeitas, todas foram executadas a partir deste conceito.
Design simples	O melhor design acabou por ser o mais simples: inserir um ponto de rotação no local desejado e utilizá-lo para girar o plano.
Teste	Diversos testes foram realizados para correção da rotação do plano e ao final do processo para fechamento do <i>sprint</i> .
Refatoração	Ao final do <i>sprint</i> a épura foi refatorada para eliminar repetições no código.
Integração contínua	Houve uma quebra na continuidade devido a reestruturação da classe Tplano que, no entanto, foi retomada da versão anterior.
Cliente presente	Os professores orientadores desta pesquisa acompanharam o progresso do trabalho.

Fonte: a autora (2022)

Correção do desempenho

Durante a implementação da classe Tplano, foi observada uma demora no tempo de resposta enquanto o arraste era executado, assim como uma piora à medida que mais planos eram acrescentados. Para a resolução deste problema, foi realizado um novo *sprint* com enfoque no desempenho do aplicativo (Figura 71).

Figura 71 - *Sprint* correção do desempenho



Fonte: a autora (2022)

A resposta para este obstáculo veio a partir da observação do método *Update* da *viewport* 3D. A cada alteração realizada, este era chamado e removia todos os

itens da *view* e os reinseria. Enquanto os objetos apresentavam apenas pequenas alterações, como inserção de novos pontos ou faces, ou até mesmo alteração nestes, a dificuldade de processamento não era sentida. No entanto, à medida que novos e mais complexos elementos como planos e projeções eram inseridos, a visualização era prejudicada. Para evitar a variação de desempenho, o método de inserção da classe *Tplano* foi removido do *Update*. Assim, quando um novo plano é inserido, somente ele é recalculado durante o arraste. O mesmo processo foi repetido na época. O Quadro 27 apresenta a síntese da metodologia XP durante o desenvolvimento do *Sprint*.

Quadro 27 - Metodologia XP na correção do desempenho

Princípio metodologia XP	Síntese no Sprint
O Jogo de Planejamento	<ul style="list-style-type: none"> • Identificar a fonte do problema de desempenho • Corrigir o problema
Pequenas versões	Em um primeiro momento, foi preciso identificar a causa do problema de desempenho e somente então atacar o problema.
Design simples	Neste caso, foram realizadas somente algumas reestruturações
Teste	Alguns testes foram realizados anteriormente a implementação da solução, como forma de confirmar a origem do obstáculo. Durante o desenvolvimento da solução, novos testes foram realizados para identificar se o problema havia sido resolvido.
Refatoração	Durante a análise do método <i>Update</i> , algumas repetições no código foram refatoradas para torná-lo mais dinâmico.
Integração contínua	Neste caso foi apenas uma implementação pontual.
Cliente presente	Os professores orientadores desta pesquisa acompanharam o progresso do trabalho.

Fonte: a autora (2022)

Adaptações às proposições do artefato

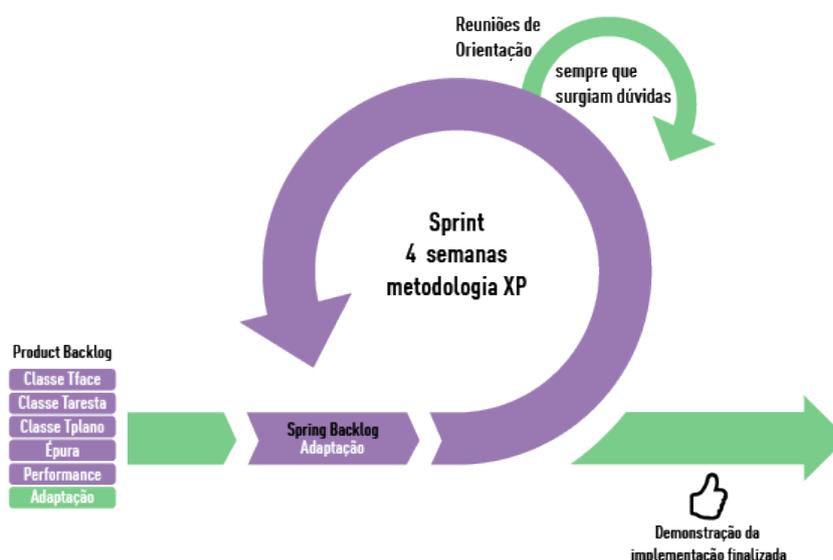
Os demais *sprints* realizados, com exceção da correção do desempenho, tratavam de adaptações de classes e *views* que já existiam na versão desktop do HyperCAL^{3D}. Desta forma, as necessidades de implementação eram bem definidas. Por outro lado, as implementações realizadas neste *Sprint* são fruto dos itens observados na revisão sistemática de literatura, cruzamento teórico e análise de similares desta pesquisa, que trouxeram a luz questões significativas quanto às interfaces utilizadas nos aplicativos móveis.

Como observado, a maior parte das perguntas de análise recaem sobre questões de interface: a quantidade de informações na tela, atalhos, navegação, seleção, edição, prevenção de erros, utilização de sensores, entrada de dados e temas e padrões. Portanto, em um primeiro momento são avaliadas as alternativas encontradas e se estas poderiam servir às necessidades do HyperCAL^{3D} *mobile*. Para

tanto, são avaliados quais recursos poderiam ser adaptados à interface gestual do aplicativo e quais podem ser implementados em sua interface gráfica, levando em consideração as questões sobre NUI levantadas no capítulo 2 deste relatório.

Foram selecionadas para a implementação, as questões avaliadas nos aplicativos analisados. Por se tratar de questões relativas à interface, sendo ela gráfica ou gestual, e por serem debatidas questões decorrentes dos estudos realizados nesta pesquisa, este foi considerado como um único *Sprint* (Figura 72). No entanto, devido à diversidade de temas e implementações necessárias, cada item apresenta sua própria metodologia XP.

Figura 72 - *Sprint* Adaptação



Fonte: a autora (2022)

Para facilitar a compreensão, itens que demandaram mais atenção foram divididos em dois momentos, inicialmente são apresentados os requisitos a serem atendidos e os motivos para a implementação, em seguida são descritas as implementações realizadas e os itens da metodologia XP. O item cliente, presente na metodologia XP, foi suprimido, pois o aplicativo foi apresentado aos professores orientadores ao final do *sprint*, para suas análises quanto aos gestos utilizados. As alterações necessárias, após suas análises, foram acrescentadas nos itens subsequentes.

Além das implementações dos requisitos estudados nesta pesquisa, foi necessário o desenvolvimento da identificação dos toques para a interface gestual do aplicativo. Mesmo existindo algumas bibliotecas com as funcionalidades, a autora

optou pela não utilização destas, pois, ao compreender como os gestos podem ser utilizados, uma gama maior de possibilidades de toques pode ser implementada no futuro.

Identificação dos toques

A partir de estudos sobre a linguagem Javascript no site w3schools¹⁸, foi possível compreender a existência de quatro tipos de eventos relacionados ao toque: *ontouchcancel*, *ontouchend*, *ontouchmove* e *ontouchstart*. Além dos eventos, é possível identificar quantos dedos tocaram a tela. O Javascript também possui o método *setTimeout* que determina o tempo final de uma função ou método.

Com estes elementos, uma grande diversidade de toques e combinações de toques pôde ser feita. Portanto, o primeiro passo para a implementação foi a identificação do evento de início do toque, que armazena em duas variáveis a posição inicial x e y do toque e inicia a contagem do tempo. Neste ponto, foi inserida a checagem de interrupção do toque, como mostra a Figura 73. De um lado, encontra-se possibilidade de interrupção do toque e de outro a manutenção dele. Caso ocorra a interrupção do toque e um novo início de toque foi identificado em menos de 200 milissegundos, o método dois toques é acionado, caso contrário o método toque simples é chamado.

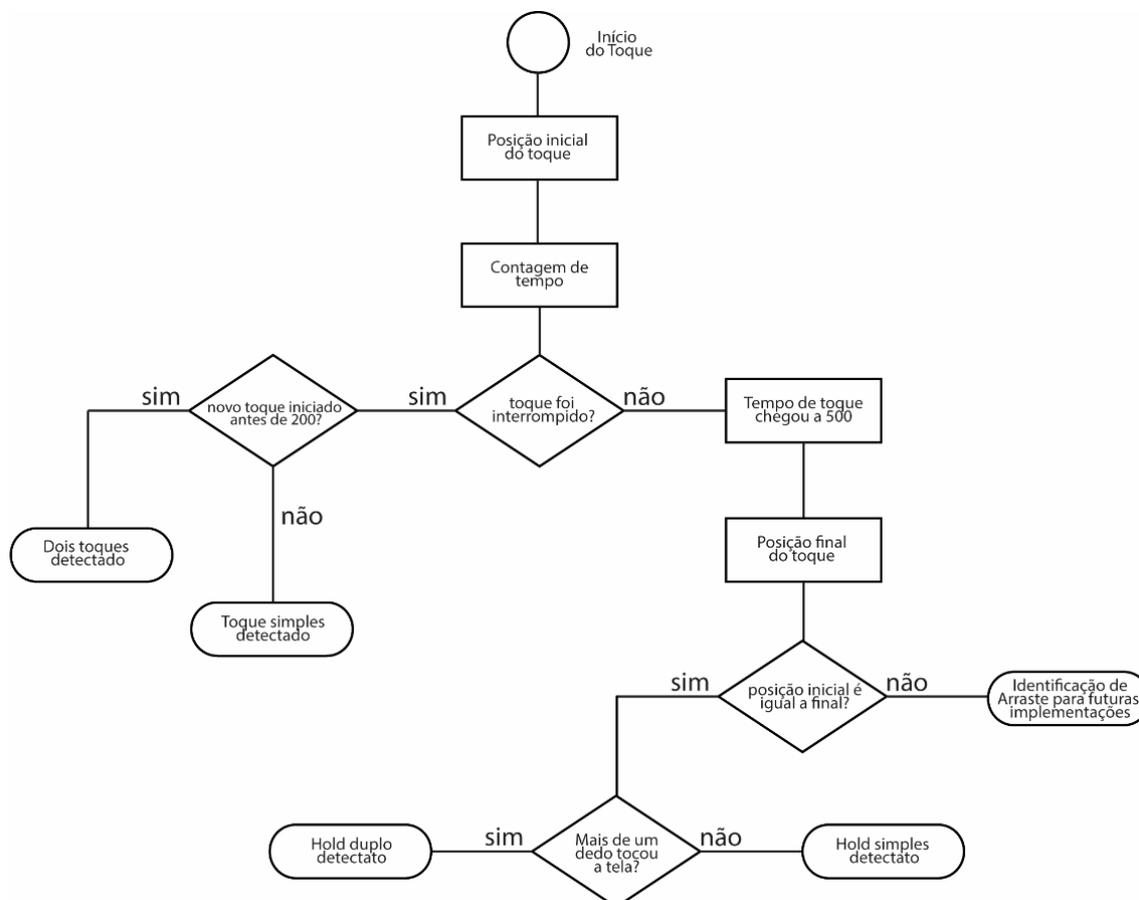
Se a manutenção do toque for identificada, é verificado o tempo de manutenção até que este chegue a 500 milissegundos, se esta condição for atendida, há uma verificação da posição do dedo ao final do tempo e uma comparação com a posição inicial dele, o que determina o gesto de *hold* (tocar e segurar) que pode ser simples ou duplo, devido ao tamanho da tela toques triplos ou maiores não foram considerados.

Esta implementação permite, ainda, a identificação de toque simples com dois dedos, duplo toque com dois dedos, arraste em direções específicas, como para os lados, para cima e/ou para baixo. Assim, novas possibilidades de toques podem ser implementadas no futuro para auxiliar usuários do HyperCAL^{3D} *mobile* a utilizá-lo sem a interface gráfica, que poderá ser totalmente escondida e aumentar a área útil de utilização.

¹⁸ https://www.w3schools.com/jsref/obj_touchevent.asp acesso em 8/8/2022

Vale ressaltar que os gestos de arraste simples, arraste duplo e arraste em pinça, não foram implementados da mesma forma, pois estes já são utilizados como controle de câmera pela biblioteca Three.js.

Figura 73 - Fluxograma de identificação de toques HyperCAL^{3D} *mobile*



Fonte: a autora (2022)

Requisitos para Navegação

Há uma predominância das interfaces gestuais na navegação, mesmo que em alguns casos seja possível observar um cubo de navegação, comum em *softwares* tridimensionais, estes em geral são utilizados para o acesso a vistas ortográficas, o que não é utilizado no HyperCAL^{3D}.

O arraste simples foi identificado como o gesto de navegação mais empregado para os movimentos preferenciais da *view*, pan para interfaces 2D e rotação para interfaces 3D. Estes fatos confirmam, as ideias de Saffer (2008) sobre a complexidade do gesto ser diretamente proporcional à tarefa que ele executa.

Mesmo os aplicativos 2D analisados não utilizando a rotação, o HyperCAL^{3D} utiliza este recurso, portanto ele será mantido. Assim, para alinhar o HyperCAL^{3D}

mobile às tendências gestuais dos demais aplicativos, são usados os gestos para a navegação mostradas no Quadro 28.

Quadro 28 -Gestos da navegação HyperCAL ^{3D} *mobile*

Navegação HyperCAL ^{3D} <i>mobile</i>			
	Rotação	Pan	Zoom
Viewport 3D	 ARRASTE SIMPLES	 ARRASTE DUPLO	 ARRASTE PINÇA
Épura	 ARRASTE DUPLO	 ARRASTE SIMPLES	 ARRASTE PINÇA

Fonte: a autora (2022)

Implementação dos Requisitos de Navegação

A biblioteca Three.js possui controles de navegação padrão: arraste simples para rotação, arraste duplo para **pan** e arraste em pinça para **zoom**. Para alterar os gestos da épura, foi necessária apenas a troca de controle. No entanto, durante as análises dos professores, ficou evidente a dificuldade de realizar o zoom devido à rotação, que ocorre em conjunto ao pinçamento, pois a rotação também utiliza dois dedos. Por esse motivo, os gestos de navegação foram mantidos na forma original. Devido à simplicidade da implementação, neste ponto não são apresentados os itens da metodologia XP.

Requisitos de Seleção

O HyperCAL ^{3D} *desktop* permite a seleção de faces, arestas e planos. Porém, até a finalização desta pesquisa, a seleção somente permite a criação de novos sistemas de referência que não alteram as características do plano selecionado. Portanto, a seleção destes não é considerada nesta pesquisa.

Na interface gráfica, existem ferramentas que devem ser selecionadas para que sejam habilitadas. Após seleção da ferramenta correspondente, é possível selecionar mais de um objeto (faces ou arestas), desde que sejam do mesmo tipo.

Quanto aos aplicativos analisados, foi observada a predominância do toque simples para a seleção simples, sem a necessidade de seleção de ferramentas. Além desta, apenas dois aplicativos apresentaram multisseleção e ambos utilizavam toque simples.

Para manter o padrão utilizado no HyperCAL ^{3D} *desktop*, as ferramentas de seleção permanecerão ativas seguidas de toque simples. Como forma de aprimorar a interface gestual do aplicativo, são usados toques simples para a seleção diretamente sobre os objetos na *viewport* 3D. A multisseleção é ativada de acordo com o primeiro item selecionado (Quadro 29).

Quadro 29 - Seleção HyperCAL ^{3D} *mobile*

Seleção HyperCAL ^{3D} <i>mobile</i>		
Interface Gráfica	Interface Gestual	
Individual/Multisseleção	Individual	Multisseleção
		

Fonte: a autora (2022)

Implementação dos Requisitos de Seleção

Inicialmente foram implementadas as duas ferramentas de seleção da interface gráfica, semelhantes às do HyperCAL ^{3D} *desktop*, que permitem a seleção de arestas e faces. Após o clique na ferramenta, um toque sobre os objetos (arestas ou faces) modifica sua cor. A possibilidade de isolamento dos itens selecionados até a finalização desta pesquisa, ainda não havia sido implementada.

A seleção dos objetos é realizada com um método de *raytrace*, que identifica, a partir da posição do toque e de um *array* de objetos da cena, quais objetos se encontram abaixo deste. Como forma de diferenciar faces de arestas, foram criados *arrays* separados para cada um dos objetos, o que aumenta o desempenho da seleção.

Depois do toque, o método *raytrace* armazena, em um *array*, os objetos que cruzaram com o toque. Os primeiros itens armazenados são os que se encontram mais próximos da câmera e os últimos, os mais afastados. Desta forma, o primeiro item do *array* é o que está logo abaixo do dedo. Por este motivo, ele é selecionado e sua cor, alterada, indicando a seleção. Devido à utilização da interface gráfica, a informação do tipo de item a ser selecionado precisava ser enviada de um componente para outro, pois o botão de seleção é um componente da *Toolbar* e a seleção é feita, até este momento, na *viewport* 3D. Portanto, a variável que contém a informação de qual item deve ser selecionado foi colocada na *store* Three.

As *Stores*, como já citado, possuem os *status* de objetos e/ou variáveis, ações e métodos para alteração destes *status* e métodos que enviam os valores para outros componentes. Desta forma, quando a ferramenta é acionada, envia uma ação para um método de mutação do estado da variável, que armazena o tipo de objeto a ser selecionado e, em seguida, grava o novo valor.

No componente *viewport 3D*, um método *getter*¹⁹ busca a informação na *store* e retorna o valor que identifica qual o *array* deve ser utilizado no *raytrace* para a identificação do cruzamento.

Para a implementação de seleção somente com a interface gestual, inicialmente foi habilitado o *raytrace* com uma busca em um *array* com faces e arestas. Caso ele identifique o cruzamento com uma face ou uma aresta, ele detecta que tipo de item é (face ou aresta) e passa a permitir a seleção somente de itens semelhantes. No entanto, isso só é possível, caso a ferramenta de seleção não tenha sido ativada. O Quadro 30 sintetiza os princípios utilizados da metodologia XP na implementação da Seleção no HyperCAL^{3D} *mobile*.

Quadro 30 -Metodologia XP na implementação da Seleção

Princípio metodologia XP	Síntese da Seleção
O Jogo de Planejamento	<ul style="list-style-type: none"> • Iniciar o método de cruzamento a partir da ferramenta • Identificar o tipo de item a ser selecionado • Identificar o primeiro objeto do cruzamento • Alterar a cor do objeto identificado • Identificar a partir do primeiro item qual o tipo de objeto que deve ser selecionado • Utilizar os métodos criados para a seleção via ferramenta.
Pequenas versões	Desde as primeiras versões já foi identificada a necessidade de diferenciar faces, arestas e planos, portanto, foram implementadas na store métodos que identificam e gravam os tipos de itens a serem selecionados pelo cruzamento
Design simples	Não havia necessidade de implementar um design complexo em funcionalidades simples como as de seleção, portanto, o design final é bem simples.
Teste	Vários testes durante a implementação foram realizados para testar as funcionalidades da seleção, novos testes serão necessários futuramente, após a implementação do isolamento das faces e arestas.
Refatoração	Várias partes do código foram repetidas nos testes iniciais e refatoradas em métodos únicos ao final do processo.
Integração contínua	Futuramente este código será utilizado para a implementação do isolamento dos objetos selecionados.

Fonte: a autora (2022)

¹⁹ Métodos para a busca de valores

Requisitos para a Deseleção

O HyperCAL ^{3D} *desktop* permite desselecionar todos os itens selecionados, de uma vez, com um clique na mesma ferramenta utilizada para a seleção. Já os aplicativos analisados apresentam formas diferenciadas de desseleção individual e múltipla. No entanto, enquanto a maioria dos aplicativos permite desselecionar individualmente, com um toque simples sobre o objeto selecionado, para desselecionar todos ao mesmo tempo, estes não apresentam um gesto comum.

Como forma de não descaracterizar as ferramentas da versão *desktop*, foi mantida ferramenta de seleção para a desseleção total. Porém, são adicionadas as possibilidades de desseleção por gestos: toque simples para desseleção individual e duplo toque para desseleção total (Quadro 31).

Quadro 31 - Deseleção HyperCAL ^{3D} *mobile*

Deseleção HyperCAL ^{3D} <i>mobile</i>		
Interface Gráfica	Interface Gestual	
Deseleção Total	Deseleção Simples	Deseleção Total
		

Fonte: a autora (2022)

Implementação dos Requisitos de Deseleção

Como já citado, as ferramentas de desseleção do HyperCAL ^{3D} *desktop* foram mantidas. A desseleção total pode ser acessada a partir da ferramenta de seleção de face e/ou aresta. O aplicativo identifica se o *array* de objetos selecionados está vazio e, se não estiver, retorna os objetos selecionados para a cor original e limpa o *array* de objetos selecionados.

Para a desseleção a partir da interface gestual, o método *raytrace* verifica se o *array* de objetos selecionados está vazio. Se estiver, adiciona o objeto que cruzou com o toque no mesmo *array* e modifica sua cor. Caso contrário, se não verifica se o objeto interceptado consta no *array* e, caso afirmativo, retira o objeto do *array* e retorna sua cor à cor original, se não, acrescenta no *array* e modifica sua cor para a cor de seleção.

Para a desseleção total via *touch screen*, foram utilizados os dois toques, a partir da identificação destes o aplicativo busca o *array* de objetos selecionados e retorna um a um para a cor original e limpa o *array*.

O Quadro 32 apresenta um resumo das etapas de implementação, assim como das estratégias utilizadas previstas na metodologia XP.

Quadro 32 -Metodologia XP na implementação da Deseleção

Princípio metodologia XP	Síntese da Seleção
O Jogo de Planejamento	<ul style="list-style-type: none"> • Criar método de desseleção que retorna os objetos para a cor original • Limpar o <i>array</i> de selecionados • Identificar se existem objetos selecionados a partir do clique na ferramenta de seleção • Se existem, método de desseleção é acionado • Se não, a seleção é acionada • Interface gestual • Identificar pelo toque sobre o objeto se este consta no <i>array</i> de selecionados • Se sim, ativar o método de desseleção • Se não, método selecionar objeto • Ativar o método de desseleção a partir do duplo toque
Pequenas versões	O código implementado foi pensado para ser complexificado de maneira gradual. Assim, inicialmente foram pensadas as etapas as desseleções totais e as mudanças de cor, para em um segundo momento identificar objetos selecionados para a desseleção.
Design simples	As interações pós seleção ainda não foram desenvolvidas, desta forma o design pode ser simplificado.
Teste	Diversos testes foram realizados durante as implementações para checar a viabilidade de implementação das etapas subsequentes.
Refatoração	Assim como as demais implementações, devido a inexperiência da autora, inicialmente foram elaboradas estratégias para o funcionamento do aplicativo. Portanto, não foram analisadas questões de repetição de código e sim se este funciona. Devido a isso, após as finalizações da implementação, diversos pontos do código apresentam repetições, que posteriormente são transferidas para métodos ou funções.
Integração contínua	Assim como as demais implementações, esta etapa será importante para implementações futuras.

Fonte: a autora (2022)

Edição

No HyperCAL ^{3D} *desktop*, os pontos e faces podem ser editados no mesmo local de inserção. Por esse motivo, a única forma de edição é a movimentação dos novos sistemas de projeção. Para tanto, é utilizada uma ferramenta que habilita a edição, em seguida um clique sobre o plano permite seu arraste. Contudo, a funcionalidade de arraste do plano não foi implementada até a finalização deste trabalho, portanto não serão pensadas estratégias para a edição neste momento.

Feedbacks

Assim como nas análises, o HyperCAL ^{3D} *desktop* utiliza a mudança de cores para indicar a seleção dos objetos. Além disso, o arraste em tempo real dos planos

de projeção indicam as alterações deles. Estas implementações foram desenvolvidas no item “seleção” e no *sprint* Tplano respectivamente.

Temas e padrões

Os temas e padrões do HyperCAL ^{3D} *mobile* foram idealizados na versão para internet e seguem padrões de cor e ícones como disquete para salvamento, setas para desfazer e refazer, entre outros, portanto não precisam ser reajustados.

Requisitos de Clareza, quantidade e qualidade de informações na tela

Na versão para a internet, que serve de base para esta adaptação, é encontrado apenas um menu na parte superior da tela e um menu sanduiche, que abre a possibilidade de inserção e edição de pontos e planos, na lateral direita do site. Além desse recurso, existe um botão que permite o carregamento da *viewport* em tela cheia.

Da mesma maneira, as análises indicaram que os principais recursos para a quantidade de informações na tela são o recurso *fullscreen* e menus retráteis. No entanto, segundo Saffer (2008) a colocação de menus na base dos aplicativos, impede que as informações sejam escondidas pela palma da mão, devido à natureza *touch screen* do dispositivo. Portanto, para a adaptação aos dispositivos móveis, o menu do topo (*maintoolbar*) foi transferido para a base da tela.

Implementação dos requisitos de clareza, quantidade e qualidade de informações na tela

Devido à utilização do mesmo código para a versão de internet, o aplicativo precisa identificar o dispositivo no qual está sendo utilizado, para a modificação da posição do menu. Desta forma, o aplicativo reconhece se o dispositivo utilizado é *mobile* (tablet ou *smartphone*) ou não. Em caso afirmativo, armazena o valor *true* na variável *ismobile* da *store three*, que fica acessível a todos os componentes. Desta forma, o componente *mainlayout* interpreta que o componente *maintoolbar* deve ser colocado na base do aplicativo.

O *framework* Quasar vue, por padrão apresenta a rolagem do menu. Portanto, não foram implementadas retrações extras para o menu de ferramentas. Por se tratar de uma implementação simples, não é apresentada a metodologia XP deste desenvolvimento.

Requisitos de Atalhos

A versão *desktop* do HyperCAL ^{3D} apresenta atalhos para salvar arquivo (Ctrl + s), criar novo arquivo (Ctrl + n), desfazer (Ctrl + z) e refazer (Shift + Ctrl + z). Estes atalhos não foram implementados na versão móvel, pois estas são as primeiras ferramentas disponíveis na *maintoolbar* e ficam visíveis sem a necessidade de rolar a barra. No entanto, devido ao tamanho dos dispositivos móveis, a utilização de atalhos pode significar um avanço na qualidade da seleção de ferramentas, que poderão ser acessadas de forma rápida.

Foi observado nas análises, o gesto de tocar e segurar para acessar ferramentas escondidas na interface gráfica. Portanto, este recurso é utilizado para o acesso a duas ferramentas no HyperCAL ^{3D} *mobile*: retornar à posição, rotação e escala original da *viewport* 3D e é pura e a criação de um novo plano de referência. O mesmo tipo de toque com um dedo é utilizado para ambas as funcionalidades. Porém, para diferenciar uma de outra, é analisado se o toque foi realizado sobre um plano de referência da *view* (Quadro 33).

Quadro 33 - Atalhos HyperCAL ^{3D} mobile

Atalhos	
Ferramenta	Gesto
Retornar câmera a posição, rotação e escala originais	 TOCAR E SEGURAR sobre vazio
Criar novo sistema de referência	 TOCAR E SEGURAR sobre plano de projeção

Fonte: a autora (2022)

Implementação dos Requisitos de Atalho

A partir do reconhecimento do gesto “tocar e segurar” e da identificação do cruzamento, pelo método de *raytrace*, do toque com um plano de projeção, um novo sistema de projeção é acionado e cria um novo Tplano na tela que pode ser arrastado em tempo real. Caso o mesmo gesto “tocar e segurar” não identifique o cruzamento com um plano de projeção, a *view* retorna as configurações originais.

Utilização de sensores

A versão de internet foi desenvolvida com a utilização do giroscópio para a alteração da disposição da *viewport* 3D e é pura de vertical para horizontal e vice-versa. As análises não demonstraram a necessidade do emprego de outros sensores, portanto não serão implementados outros na adaptação ao *mobile*.

Requisitos para a Prevenção de erros

Os recursos de desfazer e refazer estão acessíveis na versão de internet do HyperCAL^{3D}, que também exibe o que cada ferramenta faz a partir do recurso *mouse hover*. No entanto, este recurso não pode ser acessado em dispositivos com *touch screen*, pois eles não reconhecem a aproximação sem o toque. Portanto, após o toque na ferramenta, a descrição precisa aparecer e permanecer por um tempo visível e depois desaparecer. Além disso, a barra de ferramentas precisa estar na parte inferior da tela e a dica acima desta, pois, de outra forma, o recurso fica escondido pela mão do usuário.

Outro recurso para a prevenção de erros é um tutorial de gestos, exibido em um banner ao início da aplicação, que pode ser escondido a qualquer momento.

Implementação dos Requisitos para a Prevenção de Erros

A barra de ferramentas foi alterada no item clareza, quantidade e qualidade de informações na tela. Para a visualização da dica das ferramentas, foi ajustado um atraso no seu desaparecimento. Assim, quando o toque é realizado sobre a ferramenta, a dica permanece por alguns segundos e depois desaparece.

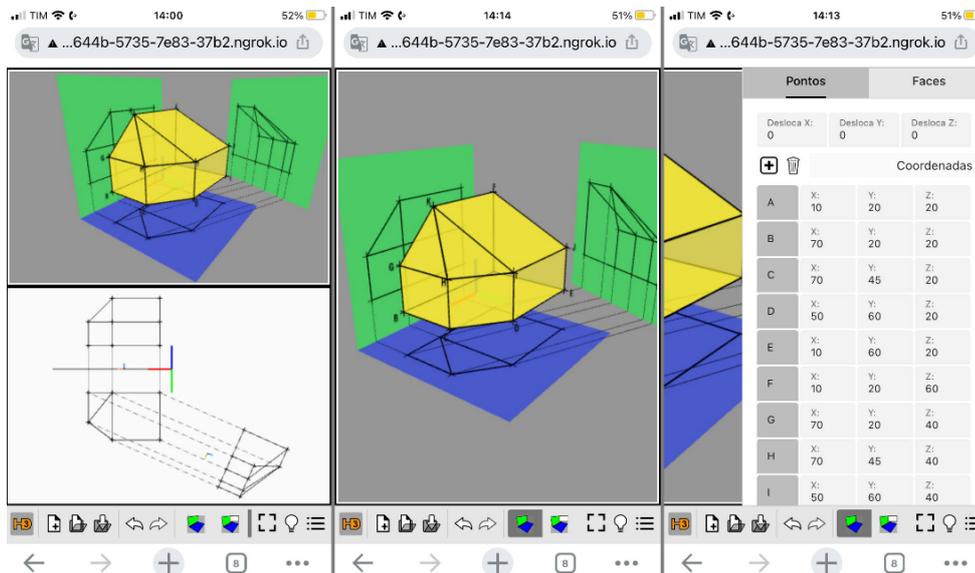
Um tutorial de *cards* com imagens dos gestos e funcionalidades deles foi inserido em um banner que aparece com o carregamento do aplicativo. Para sua implementação, foram utilizados *cards* disponíveis no *framework* Quasar vue.

Entrada de dados

Foi utilizado o teclado numérico para a entrada dos valores das coordenadas. Assim, são evitadas as possibilidades de inserção de letras pelos usuários, evitando erros.

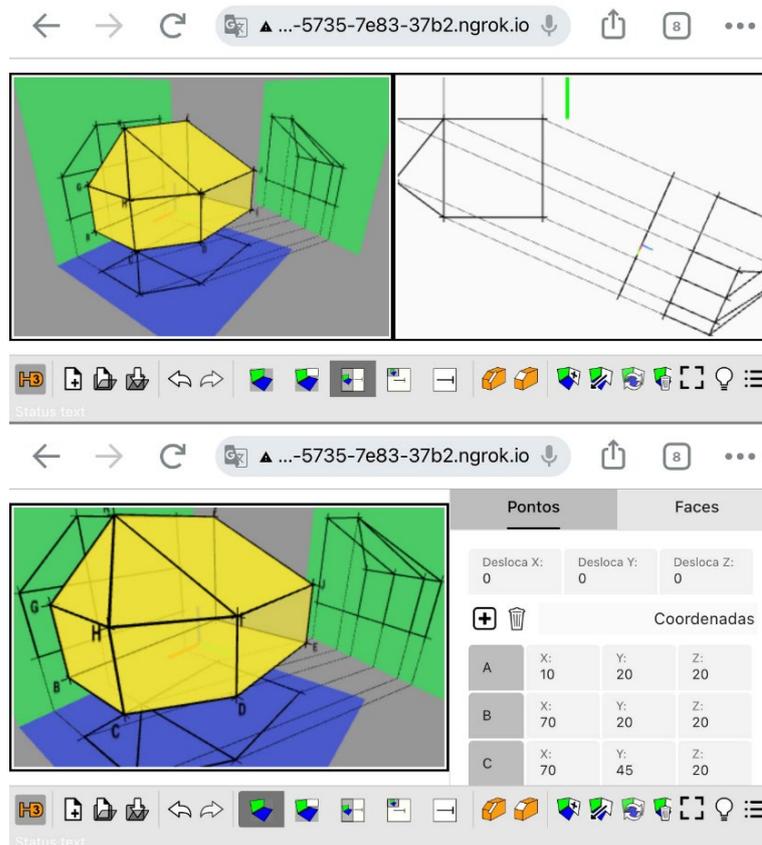
Neste primeiro momento, o HyperCAL ^{3D} *mobile* foi utilizado no navegador de internet. Posteriormente à conclusão da programação de todas as ferramentas, o aplicativo será compilado para Android e iOS. As Figura 74 e Figura 75 apresentam o aplicativo no navegador Chrome no Iphone SE na posição vertical e horizontal respectivamente.

Figura 74 - HyperCAL ^{3D} *mobile* no navegador Chrome vertical



Fonte: a autora (2022)

Figura 75 - HyperCAL ^{3D} *mobile* no navegador Chrome horizontal



Fonte: a autora (2022)

4.6. AVALIAÇÃO DO ARTEFATO

Como citado no item 3.6 deste relatório, foi utilizada a escala de usabilidade MATCH - *Measuring Usability of Touch screen Phone Applications* (SALAZAR, *et al.*, 2013), desenvolvida pelo Grupo de Qualidade de Software do Instituto Nacional para Convergência Digital, da Universidade Federal de Santa Catarina. A partir das respostas ao questionário disponível no site do grupo²⁰, uma pontuação para a usabilidade de aplicativos *touch screen* é apresentada juntamente com a escala e suas descrições.

Ao todo são 48 perguntas, divididas em 10 Heurísticas, com três possíveis respostas: sim, não e não se aplica. A seguir são apresentadas as perguntas, as respostas fornecidas e a justificativa para cada resposta.

Heurística 1: Visibilidade do status do sistema

1. Para cada ação do usuário o aplicativo oferece *feedback* imediato e adequado sobre seu status? Por exemplo, após tarefas como envio de e-mail, adição, exclusão e carregamento de arquivo, exibir uma mensagem de confirmação do tipo "e-mail enviado" ou "arquivo excluído".

Sim, o *feedback* é um dos pontos estudados nesta pesquisa e foi desenvolvido durante o *Sprint* de adaptação do HyperCAL^{3D} ao *mobile*.

2. Os componentes interativos selecionados são claramente distintos dos demais? Por exemplo, o estado de botões muda quando são pressionados e destaca a aba do menu que está sendo visualizada.

Sim, os botões mudam de cor quando selecionados e apresentam dicas das ferramentas.

3. As mensagens sobre o status do aplicativo possuem uma linguagem clara e concisa? Por exemplo, os títulos das telas e das mensagens de erro são de fácil compreensão.

²⁰ <http://match.inf.ufsc.br:90/index.html> acesso em 8/10/2022

Não, o aplicativo apresenta somente uma tela e não apresenta mensagens de erro.

4. Fornece um update do status para operações mais lentas? Por exemplo, uma indicação seja na forma de ícone ou texto sobre o progresso do carregamento do sistema ou de um arquivo.

Não se aplica, nesta versão os arquivos só podem ser abertos se já estiverem no dispositivo, assim, seu carregamento não demora tempo suficiente para necessitar demonstrar o carregamento.

Heurística 2: Correspondência entre o sistema e o mundo real

5. O significado de símbolos e ícones são compreensíveis e intuitivos? Utilizar ícones e símbolos fáceis de reconhecer e relacionar com a tarefa à qual estão associados.

Sim, os ícones utilizados são comumente associados as mesmas funções em outros aplicativos e *softwares* como por exemplo as pastas para abrir os arquivos e as setas para desfazer e refazer. Além disso, os demais ícones apresentam similaridades com as ferramentas que ativam.

6. As informações são dispostas em uma ordem lógica e natural? Por exemplo, itens em listas de seleção (nomes, produtos, etc.) são ordenados por um critério adequado (p.ex. alfabeticamente).

Sim, os únicos itens que apresentam listas, são os pontos e as faces, que são dispostos em ordem alfabética e numérica respectivamente.

Heurística 3: Controle e liberdade do usuário

7. É o usuário quem inicia e encerra tarefas e não o aplicativo? Por exemplo, aguardar o usuário teclar **enter** após preencher o campo de busca para iniciar a tarefa.

Não, durante a inserção de um novo ponto as coordenadas vão sendo atualizadas em tempo real e não aguardam a conclusão da tarefa.

8. É possível identificar o número de passos necessários para a realização de uma tarefa? Por exemplo, a partir de uma indicação numérica (1-5) da quantidade de páginas ou passos, da apresentação de um tutorial ou da divisão da tarefa em abas.

Sim, um tutorial com os gestos utilizados na interface gestual foi adicionado na implementação do *Sprint* de adaptação do HyperCAL^{3D} ao *mobile*.

9. É possível retornar a tela anterior a qualquer momento? Seja a partir da navegação por abas, de um botão voltar do aplicativo ou do próprio celular.

Não se aplica, o HyperCAL^{3D} *mobile* só apresenta uma tela com um menu de ferramentas e um menu sanduiche para a inserção de pontos.

10. No caso de aplicativos associados a login ou contas de e-mail, permite o fácil acesso de mais de um usuário? Por exemplo, um aplicativo de comércio eletrônico permitir a fácil escolha de qual conta utilizar para realizar a compra.

Não se aplica, esta versão ainda não apresenta login

11. O usuário pode cancelar uma ação em progresso? Por exemplo, cancelar um download em andamento.

Não se aplica, o usuário pode cancelar uma ação após a execução no botão desfazer, mas não durante a ação. Além disso, o aplicativo ainda não possui um banco de dados online que permita o download de arquivos.

12. O aplicativo deixa claro qual o próximo passo para realizar a tarefa? Como a partir de um botão para avançar ou nota de explicação.

Sim, as ferramentas apresentam dicas de suas funcionalidades quando um toque sobre elas é identificado.

Heurística 4: Consistência e padrões

13. As telas com o mesmo tipo de conteúdo possuem o mesmo título? Por exemplo, todas as telas de busca possuem o mesmo título.

Não se aplica, o HyperCAL ^{3D} *mobile* só apresenta uma tela.

14. Controles e botões se distinguem do restante do layout, deixando evidente que são clicáveis? Por exemplo, diferenciar os botões aplicando sombra ou outro recurso para simular relevo.

Sim, a barra de ferramentas se destaca com um cinza claro de fundo e vários botões quadrados que indicam as ferramentas existentes, diferente dos demais componentes dispostos na tela.

15. Todas as informações textuais do aplicativo utilizam o mesmo idioma?

Sim

16. Funções diferentes são apresentadas de maneira distinta ao usuário? Por exemplo, funções diferentes como salvar e cancelar não são representadas pelo mesmo nome ou ícone.

Sim, os ícones utilizados para cada ferramenta são diferentes entre si.

17. Funções semelhantes são apresentadas de forma similar? Por exemplo, usa o mesmo ícone ou rótulo de botão para a mesma funcionalidade em telas diferentes ou propõe a mesma forma de entrada de dados para uma mesma funcionalidade em diferentes telas.

Não se aplica, o HyperCAL ^{3D} *mobile* não apresenta mais de uma tela.

18. Controles que realizam a mesma função ficam em posições semelhantes na tela? Por exemplo, se em uma tela o botão para avançar fica no lado direito, nas outras telas esse mesmo botão também estará no lado direito.

Não se aplica, o HyperCAL ^{3D} *mobile* não apresenta mais de uma tela.

19. A forma de navegação é consistente entre as telas no aplicativo? Mantém o mesmo tipo de navegação (rolagem vertical, rolagem horizontal, menus ou abas) em todas as telas.

Sim, a navegação da *viewport* 3D e da *épura* apresentam os mesmos gestos de navegação.

20. Os links são tratados de forma consistente entre as telas? Mantêm o mesmo tratamento visual em termos de cor, tipo e estilo (p.ex. negrito, sublinhado) de fonte.

Não se aplica, o HyperCAL ^{3D} *mobile* não apresenta mais de uma tela.

21. As informações textuais são apresentadas de forma padronizada? Apresenta informações textuais semelhantes na mesma disposição e com o mesmo tratamento visual (tamanho, tipo e cor da fonte).

Sim, todas as dicas são inseridas da mesma maneira, com a mesma fonte e do mesmo tamanho. Além disso, todo o aplicativo apresenta as mesmas fontes, de tamanhos e cores semelhantes de acordo com suas funções.

22. Os dados e mensagens mais importantes encontram-se na posição padrão dos aplicativos para esta plataforma?

Sim, o menu de ferramentas foi reposicionado para evitar que a palma da mão escondesse as informações. As dicas das ferramentas foram posicionadas logo acima dele para permitir uma melhor visualização.

23. Em campos onde existe a necessidade de inserção de dados isso é evidente? Por exemplo, ter uma caixa de texto com cursor.

Sim, para a inserção de pontos, faces e salvar arquivo são apresentadas caixas de texto que mudam de cor ao serem selecionadas e apresentam o cursor piscando indicando a possibilidade de digitação.

Heurística 5: Reconhecimento em vez de lembrança

24. O aplicativo utiliza em seus textos e rótulos, uma linguagem habitual e conhecida pelo usuário do aplicativo? Evitando termos técnicos ou muito específicos de determinada área.

Sim, o HyperCAL ^{3D} *mobile* não apresenta rótulos e textos desconhecidos aos usuários, pois segue as nomenclaturas de geometria descritiva.

25. Os títulos das telas descrevem adequadamente seu conteúdo?

Não se aplica, o HyperCAL ^{3D} *mobile* não apresenta mais de uma tela.

Heurística 6: Flexibilidade e eficiência de uso

26. O aplicativo funciona corretamente, sem apresentar problemas durante a interação? Por exemplo, não trava e botões funcionam no primeiro clique.

Sim, o desempenho do aplicativo foi ajustada em um *sprint* exclusivo para seu ajuste.

27. As tarefas são relativamente simples de serem executadas? Por exemplo, uma tarefa pode ser completa em poucos passos.

Sim, em especial após a implementação da interface gestual e dos atalhos.

28. As funções mais utilizadas são facilmente acessadas? As funções mais utilizadas devem ser acessadas sem precisar rolar ou navegar entre muitas telas.

Sim, atualmente é possível selecionar planos, faces e arestas sem a necessidade de acionar nenhuma ferramenta além da interface gestual e com apenas um gesto (dois se contabilizado o arraste do novo sistema de projeção).

29. O aplicativo utiliza objetos (ícones) em vez de botões? Por exemplo, utilizar um ícone de impressora em vez de utilizar a palavra impressora.

Sim, todos os botões apresentam ícones.

30. Todas as telas mantêm acessíveis menus e funções comuns do aplicativo? Por exemplo, em aplicativos de conta de e-mail a caixa de entrada é acessível a partir de todas as telas do aplicativo.

Não se aplica, o HyperCAL ^{3D} *mobile* não apresenta mais de uma tela.

Heurística 7: Estética e design minimalista

31. São exibidas apenas informações relacionadas a tarefa que está sendo realizada? Por exemplo, na tela de cadastro, outras informações não devem ser exibidas.

Sim, o aplicativo só apresenta uma tela.

32. São usados textos somente quando estes são realmente indispensáveis? Por exemplo, não oferecer instruções textuais muito longas.

Sim, somente são utilizados textos, nas dicas, pontos, faces e nos banners de salvamento e criar novo arquivo.

33. O menu é esteticamente simples e claro? Com opções fáceis de encontrar, dispostas em uma ordem lógica e com títulos curtos.

Sim, além de poucas ferramentas, os ícones são claros, com poucos detalhes e posicionados por similaridade, exemplo: todas as ferramentas relacionadas a arquivos estão colocadas juntas.

34. O aplicativo exhibe quantidades pequenas de informações em cada tela? Sem texto ou imagens em excesso.

Sim, apenas o menu e as *views* ficam visíveis.

35. Os títulos de telas/janelas e rótulos de botões/links são curtos?

Sim, os únicos rótulos existentes estão nas abas de criação de botões e faces e possuem apenas uma palavra.

36. Em textos, o uso de abreviaturas é evitado? Com opções fáceis de encontrar, dispostas em uma ordem lógica e com títulos curtos.

Sim, não são utilizadas abreviaturas.

Heurística 8: Pouca interação homem/dispositivo

37. A navegação do aplicativo é intuitiva? Por exemplo, é fácil chegar à tela desejada.

Sim, mesmo o HyperCAL ^{3D} *mobile* não apresentando outras telas, o item navegação foi apresentado no *Sprint* de adaptação, portanto foi considerado como sim

Heurística 9: Interação física e ergonomia

38. Possui botões com tamanho adequado ao clique? Por exemplo, evitando botões muito pequenos causando a seleção da opção errada.

Sim, apresentam tamanho ideal, nem pequeno, nem grande demais.

39. A navegação principal encontra-se na posição padrão dos aplicativos para esta plataforma? Por exemplo, o menu na barra inferior para o iOS e superior para o Android.

Sim, o menu foi adicionado na barra inferior da tela, porém para todos os dispositivos móveis *touch screen*, devido aos conhecimentos obtidos no item interfaces gestuais do capítulo 2.

40. Os botões e controles podem ser facilmente acessados com qualquer uma das mãos? Especialmente no caso de botões que serão utilizados repetidamente para avançar ou confirmar ações.

Sim, a disposição da barra de ferramentas na parte inferior da tela facilita o acesso com ambas as mãos.

41. A área clicável dos botões e links ocupa toda a dimensão dos mesmos?

Sim

Heurística 10: Legibilidade e layout

42. O espaçamento entre linhas utilizado favorece a leitura? Nem muito grande, para não aumentar desnecessariamente a rolagem, e nem muito pequeno dificultando a leitura.

Não se aplica, não são utilizados textos longos que necessitem espaçamento entre linhas.

43. As fontes utilizadas favorecem a leitura? Em termo de tamanho, tipo e estilo.

Sim, todas as fontes são legíveis.

44. Os ícones possuem contraste suficiente em relação ao plano de fundo?

Sim, todos os ícones são bem destacados.

45. Os textos têm contraste suficiente em relação ao plano de fundo? Por exemplo, evitando texto cinza claro em um fundo branco.

Sim, nos casos de fundo claro (pontos e faces) o texto é preto e em casos de fundo escuro (dicas) o texto é branco.

46. As imagens possuem cor e detalhamento favoráveis a leitura em uma tela pequena? A resolução deve permitir a fácil identificação dos elementos da imagem e os ícones não devem ter muitos detalhes usando uma representação mais abstrata.

Sim, os ícones apresentam poucos detalhes.

47. O aplicativo realça conteúdos mais importantes, deixando-os maiores, mais brilhosos ou em negrito?

Não se aplica, todos as ferramentas apresentam igual importância.

48. O alinhamento utilizado favorece a leitura? Por exemplo, dando preferência para alinhamento justificado ou esquerdo para texto corrido

Não se aplica.

Resultado

O HyperCAL ^{3D} *mobile* recebeu 61.4 pontos - Usabilidade muito alta. O valor mais alto da escala MATCH. O Quadro 34 apresenta a escala e o resultado obtido.



Quadro 34 Pontuação obtida na escala MATCh

MATCh

Checklist para Avaliação da Usabilidade de Aplicativos para Celulares Touchscreen

Início

Resultado: 61.4 pontos - Usabilidade muito alta

Nível	Características que os aplicativos para celular touchscreen quase sempre ou sempre possuem...
Até 30	<p>Usabilidade muito baixa</p> <p>Somente iniciam as tarefas ao comando do usuário, evidenciam a necessidade de inserção de dados, possuem botões e links com área clicável do tamanho dos mesmos, evitam abreviaturas, além disso, são consistentes, utilizam o mesmo idioma em seus textos, apresentam os links de forma consistente entre as telas e funções semelhantes de forma similar.</p>
30 - 40	<p>Usabilidade baixa</p> <p>Além de possuir as características do nível anterior, fornecem um update do status para operações mais lentas por meio de mensagens claras e concisas, mantêm o mesmo título para telas com o mesmo tipo de conteúdo, utilizam títulos de telas que descrevem adequadamente seu conteúdo, exibem apenas informações relacionadas a tarefa que esta sendo realizada, apresentam ícones e informações textuais de forma padronizada com contraste suficiente em relação ao plano de fundo, e imagens com cor e detalhamento favoráveis a leitura em uma tela pequena, possuem navegação consistente entre suas telas, permitem retornar a tela anterior a qualquer momento, mantêm controles que realizam a mesma função em posições semelhantes na tela, permitem que as funções mais utilizadas sejam facilmente acessadas e possuem botões com tamanho adequado ao clique.</p>
40 - 50	<p>Usabilidade razoável</p> <p>Além de possuir as características dos níveis anteriores, dispõem as informações em uma ordem lógica e natural, apresentam as mensagens mais importantes na posição padrão dos aplicativos para a plataforma, oferecem uma navegação intuitiva e um menu esteticamente simples e claro, contêm títulos e rótulos curtos, possuem fontes, espaçamento entrelinhas e alinhamento que favorecem a leitura, realçam conteúdos mais importantes, possuem tarefas simples de serem executadas que deixam claro qual seu próximo passo, oferecem feedback imediato e adequado sobre seu status a cada ação do usuário, evidenciam que controles e botões são clicáveis, distinguem claramente os componentes interativos selecionados, utilizam objetos (ícones) ao invés de botões, com significados compreensíveis e intuitivos e não apresentam problemas durante a interação (trava, botões que não funcionam no primeiro clique, etc).</p>
50 - 60	<p>Usabilidade alta</p> <p>Além de possuir as características dos níveis anteriores, exibem pequenas quantidades de informação em cada tela, mantêm acessíveis menus e funções comuns do aplicativo em todas as telas, evidenciam o número de passos necessários para a realização de uma tarefa, permitem que o usuário cancele uma ação em progresso, possuem navegação de acordo com os padrões da plataforma a que se destinam e possibilitam fácil acesso de mais de um usuário no caso de aplicativos associados a cadastro de login.</p>
Acima de 60	<p>Usabilidade muito alta</p> <p>Tem ainda maior probabilidade, que os níveis anteriores, de possuir todas as características descritas acima, possuindo um alto nível de usabilidade.</p>

Grupo de Qualidade de Software- GQS
 Instituto Nacional para Convergência Digital - INCOD
 Departamento de Informática e Estatística - INE
 Universidade Federal de Santa Catarina - UFSC
 88049-200 Florianópolis - SC Brasil
 gqs@incod.ufsc.br

APOIO



[Sobre](#) | [Política de Privacidade](#) | [Termos de Serviço](#)

Fonte: GQS UFSC (2022)

5. EXPLICITAÇÃO DAS APRENDIZAGENS

Nesta etapa, são salientados os pontos fortes e fracos do estudo e do artefato desenvolvido, assim como os fatores que auxiliaram na conclusão deste e questões relevantes encontradas.

Várias etapas foram realizadas até chegar a este ponto. Em um primeiro momento foi importante compreender o *mobile learning* e, em especial, como pode ser desenvolvido. Ficou evidente a necessidade de uma equipe multidisciplinar para a elaboração de um projeto que pudesse atender ao ensino móvel. Por outro lado, a experiência adquirida com os anos de estudo do grupo ViD e a evolução do HyperCAL ^{3D} *desktop* permitiram que diversas etapas do processo pudessem ser desenvolvidas.

A revisão sistemática de literatura, realizada para compreensão das etapas necessárias ao desenvolvimento, trouxe à luz questões sobre a aceitação da tecnologia. No entanto, os modelos de aceitação, mesmo apresentando diferentes parâmetros e construtos, tendem a demonstrar um cenário genérico e não específico de necessidades, como por exemplo: “O grau em que um usuário percebe sua competência confortável para usar o aplicativo de aprendizagem móvel” (ALMAIAH, ALAMRI e AL-RAHMI, 2019).

A solução encontrada para aproximar o modelo MLAM com requisitos mais específicos para as análises de similares, foi cruzar seus princípios relevantes com a heurística de usabilidade móvel SMART e questões pertinentes à adaptação do HyperCAL ^{3D} *desktop*, para sua versão *mobile*.

Durante o cruzamento teórico, foi possível observar que a interface gestual possui um papel fundamental na usabilidade de dispositivos móveis. No entanto, seu uso ainda não atingiu todo seu potencial. As interfaces gráficas ainda são predominantes em aplicativos *touch screen*. Este é um fato relevante, pois, se por um lado, a falta de periféricos prejudica a manipulação das *views* e dos objetos, por outro, o manuseio gestual destes é intuitivo e dinâmico. Além disso, a diversidade de gestos possíveis pode levar a interação a patamares ainda não explorados e representar um acréscimo para a aceitação da tecnologia móvel. Por outro lado, não é possível confirmar esta afirmação sem a implementação de mais gestos e a avaliação destes.

Além disso, a realização de testes com usuários pode determinar novos caminhos, que não foram observados por esta pesquisa, a serem explorados na implementação de interfaces gestuais de aplicativos destinados a modelagem 3D e desenho técnico.

Outro fator importante de ser mencionado, é que até mesmo a escala de usabilidade MATCH - *Measuring Usability of Touch screen Phone Applications* utilizada para a avaliação do artefato desta pesquisa, mesmo possuindo questões relacionadas a interfaces gestuais, apresenta uma tendência as questões que se referem a interface gráfica.

Portanto, é importante ressaltar que a interface gestual, em especial em aplicativos tridimensionais que visam a modelagem, manipulação e edição de objetos 3D, assim como aplicativos que se destinam ao desenho técnico e/ou que apresentem características semelhantes ao HyperCAL^{3D} *mobile*, pode significar a melhora na usabilidade, em especial de usuários experientes, assim como os atalhos em *softwares* da mesma natureza auxiliam a utilização das ferramentas.

6. CONSIDERAÇÕES FINAIS

Este capítulo apresenta o fechamento deste estudo e foi estruturado em três partes. Na primeira parte, são apresentadas as conclusões do estudo a partir da hipótese estabelecida na introdução do relatório. O item 6.2, discorre sobre a generalização para uma classe de problemas e a última parte do capítulo propõe futuros estudos.

6.1. CONCLUSÕES

A tecnologia vem moldando a vida cotidiana ao longo do tempo, novas ou melhoradas formas de interação constantemente emergem até serem substituídas em um ciclo constante, e no ensino não é diferente. Contudo, a adaptação das ferramentas de ensino às novas possibilidades, nunca é uma tarefa simples. Além do conhecimento do conteúdo e do conhecimento pedagógico, os educadores necessitam de conhecimento tecnológico para adaptar o ensino e até mesmo a tecnologia requer diferentes conhecimentos.

A tecnologia móvel (*smartphones* e *tablets*) difundiu-se no país. Mesmo que não seja possível afirmar que cem por cento da população brasileira possui um *smartphone*, já existem mais aparelhos que pessoas no Brasil (MEIRELLES, 2022). Por outro lado, os estudos sobre *mobile learning* ainda são muito recentes. Diferentes correntes de pensamento buscam formas de explicar os elementos que compõem o *m-learning*. No entanto, os estudos deste campo somente poderão ser aprofundados com o aumento no desenvolvimento de aplicações na área.

A falta de estudos relacionada ao desenvolvimento de *m-learning* se justifica pela necessidade de uma equipe multidisciplinar para a confecção dos aplicativos. Desta forma, é importante frisar que este estudo só foi possível devido à quantidade de pesquisas realizada pelo grupo ViD sobre o ensino de geometria descritiva, mesmo diante de adversidades, como a da Pandemia.

Este estudo buscou auxiliar na ampliação do acesso ao HypeCAL^{3D} aos alunos dos cursos de Engenharia, Arquitetura e Design da Universidade Federal do Rio Grande do Sul, uma instituição pública que depende de recursos governamentais para a manutenção do ensino e para suprir demandas em tempos incertos como os vividos

em 2020 e 2021. Ao aproveitar recursos abundantes, como celulares, que já pertencem aos alunos, é possível ampliar o acesso ao ensino qualificado.

Desta forma, em um primeiro momento de pesquisa foi importante compreender o cenário de desenvolvimento de *mobile learning*. A partir da revisão sistemática de literatura, foi possível perceber a quantidade de trabalhos que utilizam modelos de aceitação, o que representa um *gap* nos estudos, pois a busca pela aceitação da tecnologia, não produz requisitos práticos para o desenvolvimento desta.

Como forma de ultrapassar este obstáculo, foi realizado um cruzamento teórico entre o modelo de aceitação da tecnologia MLAM, desenvolvido especificamente para *mobile learning*, com as heurísticas de usabilidade SMART, próprias para dispositivos móveis. Deste cruzamento, apoiado nas características encontradas no HyperCAL^{3D}, foram idealizadas perguntas com requisitos para a avaliação de aplicativos destinados a modelagem 3D e ao desenho técnico, que são características do HyperCAL^{3D}.

Durante as análises de aplicativos similares, ficou evidente a utilização das interfaces gestuais como forma de subverter a ausência de periféricos e a pequena dimensão das telas para acomodar as diferentes ferramentas dos aplicativos. Para a melhor compreensão das práticas mais comuns, foram identificados os gestos mais utilizados para cada requisito avaliado, os feedbacks, atalhos, utilização de sensores e tutoriais, que posteriormente foram implementados no Hypercal^{3D} *mobile*.

Assim, foi possível determinar características importantes a serem implementadas na confecção da versão móvel do aplicativo.

Para o desenvolvimento do artefato, produto desta pesquisa, foi necessária a implementação das classes Tface, Taresta e Tplano, assim como a épura e uma correção de desempenho, anteriormente à inserção dos requisitos encontrados neste estudo. Estas etapas de desenvolvimento, permitiram a ampliação dos conhecimentos de programação por parte da autora, o que significou maior entendimento do processo de desenvolvimento de aplicativos moveis híbridos e permitiu a criação de um fluxograma de identificação de gestos para a implementação das interfaces gestuais presentes no último *sprint* de desenvolvimento.

Durante a produção do artefato, diversos testes foram realizados, desde pequenas consultas ao console, testes intermediários para checagem de

funcionalidades e testes de conclusão dos ciclos. Estes testes foram fundamentais, para a identificação de possíveis erros no código e aprimoramento de seus métodos.

A avaliação do aplicativo foi a última etapa realizada. Nela foi possível identificar uma pontuação significativa de usabilidade no HyperCAL ^{3D} *mobile*. No entanto, foi evidenciado o peso da interface gráfica na avaliação, pois, esta possui a maioria das perguntas direcionadas a ela, enquanto poucas perguntas são dirigidas para interface gestual. Este fato, também é percebido nas Heurísticas de usabilidade.

Além disso, devido ao fato de nem todas as funcionalidades do HyperCAL ^{3D} *mobile* terem sido implementadas até a conclusão deste estudo, o aplicativo não foi testado em sua integralidade. Futuros estudos poderão determinar o potencial total de sua interface gestual.

Portanto, é possível concluir, que mesmo que todos os objetivos de pesquisa tenham sido atingidos, a hipótese de pesquisa foi parcialmente verificada, pois a adaptação do HyperCAL ^{3D} à tecnologia móvel pode promover a ampliação do acesso a ferramenta de ensino de geometria descritiva, auxiliando a visualização e o entendimento das operações gráficas principais, podendo potencializar os estudos no campo de *mobile learning*, a reflexão dos alunos sobre a disciplina. No entanto, testes com usuários só poderão ser realizados após a conclusão das implementações faltantes ao HyperCAL ^{3D}.

Além disso, as pesquisas levaram ao entendimento da necessidade de maiores estudos relacionados a usabilidade de interfaces gestuais em especial para aplicativos de modelagem 3D e desenho técnico.

6.2. GENERALIZAÇÃO DA CLASSE DE PROBLEMAS

A criação de aplicações móveis voltadas a modelagem tridimensional e desenho técnico, têm aumentado gradativamente. *Softwares* desenvolvidos para computadores vêm sendo adaptados para a tecnologia móvel. Algumas destas adequações permanecem na forma de visualizadores, como é o caso do *sketchup*. Por outro lado, alguns aplicativos ultrapassam as dificuldades provenientes da ausência de periféricos com a utilização de interfaces gestuais.

Portanto, ao buscar respostas para a classe de problemas desta pesquisa “usabilidade em aplicativos de modelagem 3D e desenho 2D” o estudo concluiu que as interfaces gestuais, em especial em aplicativos desta natureza, podem auxiliar na usabilidade, prejudicada pela falta de periféricos. Assim, podem permitir uma maior aceitação da tecnologia por parte dos usuários.

6.3. SUGESTÃO PARA TRABALHOS FUTUROS

O desenvolvimento do trabalho levantou algumas questões que podem ser investigadas mais a fundo em estudos futuros. Inicialmente a finalização de todas as funcionalidades do HyperCAL ^{3D} *mobile* e a implementação de outros gestos para a composição da interface gestual, podem significar a melhora na experiência de usuários mais experientes. Além disso, a verificação de usabilidade com usuários finais do artefato de diferentes níveis de experiência, pode significar uma melhor compreensão dos fatores necessários à interface gestual e quais podem ser descartados.

Outro fator a ser estudado são heurísticas de usabilidade focadas nas interfaces gestuais, levando em consideração que os gestos podem reduzir as informações na tela. Para tanto, seria interessante compreender até que ponto as interfaces gestuais podem substituir as interfaces gráficas sem que o usuário se perca nas funcionalidades ou precise guardar muitas informações na cabeça. Por outro lado, a possibilidade de combinação de gestos pode tornar estas interações intuitivas e simples de serem utilizadas.

Na área de *mobile learning*, após a conclusão das implementações, outros estudos podem ser realizados, como a utilização do HyperCAL ^{3D} *mobile* em um estudo de campo dirigido com estudantes em diferentes locais da Universidade Federal do Rio Grande do Sul. Este estudo apresenta diferentes conceitos estudados no *m-learning*, pois mesmo sendo aplicado em um contexto formal de ensino, o afastamento da sala de aula e a inserção do ruído podem ser considerados positivos e/ou negativos para a aprendizagem. Desta forma, a mobilidade do dispositivo permite que os estudantes possam aprender a qualquer momento, por outro lado, um comparativo entre o *m-learning* em um contexto formal e informal ainda não foi realizado.

Além destes, diversos outros estudos podem ser realizados para ampliar os conhecimentos do *mobile learning*. Porém, outro passo importante para permitir a ampliação dos estudos da área é a criação de um banco de dados *online* para a integração da versão *desktop* com a versão *mobile*. Assim, a troca de informações entre elas pode ampliar os estudos de geometria descritiva e as formas de interação com os objetos de estudo.

Referências

ABRAHAMSSON, P. *et al.* **Mobile-D: An Agile Approach for Mobile Application Development**. Companion to the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications. Vancouver: [s.n.]. 2004. p. 24-28.

ADA, M.; STANSFIELD, M.; BAXTER, G. **Using mobile learning and social media to enhance learner feedback Some empirical evidence**. Journal of Applied Research in, 2017. 70-90.

ALMAIAH, M. A.; ALAMRI, M. M.; AL-RAHMI, W. M. **Analysis the Effect of Different Factors on the Development of Mobile Learning Applications at Different Stages of Usage**. IEEE Access, v. 8, p. 16139 - 16154, December 2019. ISSN 19313581.

ALRASHEEDI, M.; CAPRETZ, ; RAZA, **A Systematic Review of the Critical Factors for Success of Mobile Learning in Higher Education (University Students' Perspective)**. Journal of Educational Computing Research, p. 257–276, 2015.

ALSHEHRI, F.; FREEMAN, M. **Methods for usability evaluations of mobile devices**. 23rd Australasian Conference on Information Systems. Geelong: [s.n.]. 2012. p. 1-10.

ALVARADO, L. A. R. *et al.* **Layered Software Architecture for the Development of Mobile Learning Objects With Augmented Reality**. IEEE Access, 9 Agosto 2018. 57897-57909.

BACK, K. **Extreme Programming Explained: Embrace Change**. 1. ed. [S.l.]: Addison Wesley , 1999.

BECK, K. *et al.* **Manifesto para Desenvolvimento Ágil de Software**. **Manifesto Ágil**, 2001. Disponível em: <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 04 Agosto 2022.

BRUNO, F. B. **Learning Design Aplicado ao Projeto de Unidades de Aprendizagem**. Universidade Federal do Rio Grande do Sul. Porto Alegre. 2019.

CABELLO, R. **three.js**, 2010. Disponível em: <<https://threejs.org/>>. Acesso em:

11 janeiro 2022.

CHIU, P.-S. *et al.* **An authentic learning based evaluation method for mobile learning in Higher Education.** *Innovations in Education and Teaching International*, 03 Janeiro 2018. 336-347.

CHURCHILL, D.; FOX, B.; KING, . **Framework for Designing Mobile Learning Environments.** *In: CHURCHILL, D., et al. Mobile Learning Design: Theories and Application.* Singapore: Springer, v. 1, 2016. Cap. 1, p. 3-24.

CROMPOTON, H. **A Historical Overview of M-learning Toward Learner-Centered Education.** *In: BEGE, Z. L.; MUILENBURG, L. Y. Handbook of Mobile Learning.* Nova York: Routledge, 2013a. Cap. 1, p. 3-14.

CROMPTON, H. **Mobile Learning: New Approach, New Theory.** *In: BERGE, Z. L.; MUILENBURG, L. Y. Handbook of Mobile Learning.* Nova York: Routledge, 2013b. Cap. 5, p. 47-57.

CYBIS, W.; BETIOL, A. H.; FAUST, R. **Ergonomia e Usabilidade: conhecimentos, netodos e aplicações.** 1. ed. São Paulo: Novatec, 2010.

DAHI, R. **Node.js**, 2009. Disponível em: <<https://nodejs.org/>>. Acesso em: 11 janeiro 2022.

DAVIS, F. D.; BAGOZZI, R. P.; WARSHAW, P. R. **User acceptance of computer technology: a comparison of two theoretical models.** *Management Science*, v. 35, p. 839-859, Agosto 1989. ISSN 8.

DRESCH, A.; LACERDA, D. P.; ANTUNES, J. A. V. **Design Science Research Método de Pesquisa para Avanço da Ciência e Tecnologia.** Porto Alegre: Bookman, 2015.

EL-HUSSEIN, M. O. M.; CRONJE, J. C. **Defining Mobile Learning in the Higher Education Landscape.** *Educational Technology & Society*, v. 3, p. 12-21, Julho 2013.

ELLWANGER, C. **Modelagem e Simulação no Design Experiencial: Uma abordagem sistêmica para avaliar o ideação na experiência do usuário.**

Universidade Federal do Rio Grande do Sul. Porto Alegre. 2018.

FACULDADE DE ARQUITETURA, U. **Questionário sobre as condições de retorno para as atividades. FACULDADE DE ARQUITETURA UFRGS**, Porto Alegre, p. 1-18, 2020. Disponível em: <<https://www.ufrgs.br/arquitetura/resultado-do-questionario-sobre-as-condicoes-de-retorno-para-as-atividades-presenciais/>>. Acesso em: 3 mar. 2022.

FERDIANA, R. **Agile Software Engineering Framework for Evaluating Mobile Application Development**. International Journal of Scientific and Engineering Research , Dezembro 2012. 89-93.

FULLER, R.; JOYNES, V. C. T. **Should mobile learning be compulsory for preparing students for learning in the workplace?** Universidade de Sheffield. Sheffield. 2015.

GIGERENZER, G.; GAISSMAIER, W. **Heuristic Decision Making**. The Annual Review of Psychology, 1 janeiro 2011. 451–482.

ISOMÄKI, H.; PEKKOLA, S. **Reframing Humans in Information Systems**. London: Springer, 2011.

JEONG, Y.; LEE, J.; SHIN, G. **Development Process of Mobile Application SW Based on Agile Methodology**. International Conference on Advanced Communication Technology (ICACT). Gangwon: IEEE. 2008. p. 362-366.

JOYCE, G.; LILLEY, M. **Towards the Development of Usability Heuristics**. Third International Conference, Duxu. Crete: Siproinger. 2014. p. 465-474.

KALEEL, S. B.; HARISHANKAR, S. **Applying Agile Methodology in Mobile Software Engineering: Android Application Development and its Challenges**. Computer Science Technical Reports, p. Artigo 4, Janeiro 2013.

KRONBAUER, A. H.; SANTOS, C. A. S.; VIEIRA, V. **Smartphone Applications Usability Evaluation: A Hybrid Model and Its Implementation**. HCSE: International Conference on Human-Centred Software Engineering. [S.l.]: [s.n.]. 2012. p. 146-163.

LACERDA, D. P. *et al.* **Design Science Research: método de pesquisa para**

a engenharia de produção.Gestão & Produção, São Carlos, 26 novembro 2013. 741-761.

LAING, S.; GREAVES, K. **Growing Agile: A Coach's Guide to Agile Testing.** [S.I.]: Growing Agele, 2015.

LAURILLARD, D. **Teaching as a Design Science: Building Pedagogical Patterns for Learning and Technology.** New York: Routledge, 2012.

LIU, W. **Natural User Interface- Next Mainstream Product User Interface.** 11th International Conference on Computer-Aided Industrial Design & Conceptual Design 1. [S.I.]: IEEE. 2010. p. 203-205.

MACHADO, L.; VERGARA, L. G. L. **Uma análise sistemática da literatura acerca dos métodos de usabilidade aplicáveis a dispositivos móveis.**Gepros Gestão da Produção, Operações e Sistemas, v. 15, p. 42-70, 2020. ISSN DOI: 10.15675.

MALANDRINO, D. *et al.* **A Tailorable Infrastructure to Enhance Mobile Seamless Learning.**IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, Março 2015. 18-30.

MARTINEZ, D. *et al.* **An Agile-Based Integrated Framework for Mobile Application Development Considering Ilities.**IEEE Access, v. 8, p. 72461 - 72470, abril 2020.

MEIRELLES, F. S. **33ª Pesquisa Anual do FGVcia: Uso da TI nas Empresas.** Fundação Getúlio Vargas. São Paulo. 2022.

MISHRA, P.; KOEHLER, M. J. **Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge.**Teachers College Record, Nova York, Junho 2006. 1017–1054.

NIELSEN, J. **Usability Engineering.** Mountain View: Morgan Kaufmann, 1993.

NIELSEN, J.; LORANGER, H. **Prioritizing Web Usability.** Berkeley,: New Riders, 2006.

NIELSEN, J.; MOLICH , R. **Heuristic Evaluation of User Interfaces.**

PROCEEDINGS OF THE SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS. [S.l.]: [s.n.]. 1990. p. 249 - 256.

OUZZANI, M. *et al.* **Rayyan — a web and mobile app for systematic reviews.** **Rayyan**, 2016. ISSN DOI: 10.1186/s13643-016-0384-4. Disponível em: <<https://www.rayyan.ai/>>. Acesso em: 10 setembro 2021.

PEREIRA, M. **Geometria na Arquitetura: novas tecnologias e ensino remoto.** PROARQ, Rio de Janeiro, Junho 2021. 241-262.

RAHIMIAN, V.; RAMSIN, R. **Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach.** International Conference on Research Challenges in Information Science, RCIS. Marrakech, Morocco : IEEE. 2008.

RIKALA, J. **Designing a Mobile Learning Framework for a Formal Educational Context.** Universidade de JYVÄSKYLÄ. YVÄSKYLÄ. 2015.

RODRÍGUEZ, J.-M. R. *et al.* **Mobile Learning in Higher Education: Structural Equation Model for Good Teaching Practices.** IEEE Access, 15 Maio 2020. 91761 - 91769.

SAFFER, D. **Designing Gestural Interfaces.** 1. ed. Beijing: O'Reilly , 2008.

SALAZAR, L. H. A. *et al.* **Systematic Literature Review on Usability Heuristics for Mobile Phones.** International Journal of Mobile Human Computer Interaction, 2013.

SALGADO, A. D. L.; FREIRE, A. P. **Heuristic Evaluation of Mobile Usability: A Mapping Study.** International Conference on Human-Computer Interaction. Heraklion: Springer. 2014. p. 178-188.

SANTOS, S. L. **Interface Interativa Bidimensional em um software para ensino de geometria descritiva.** Universidade Federal do Rio Grande do Sul. Porto Alegre. 2016.

SARRAB, M. *et al.* **Toward Educational Requirements Model for Mobile Learning Development and Adoption in Higher Education.** AECT Association for

Educational Communications & Technology, 19 Setembro 2018. 635–646.

SCHWABER, K.; BEEDLE, M. **Agile Software Development with Scrum**. New Jersey: Prentice Hall, 2001.

SHAHABADIA, M. M.; UPLANE, M. **Synchronous and asynchronous e-learning styles and academic performance of e-learners**. *Procedia - Social and Behavioral Sciences*, Pune - India, 2015. 129 – 138.

SHNEIDERMAN, B.; PLAISANT, C. . **Designing the User Interface: Strategies for Effective Human-Computer Interaction**. 5. ed. [S.l.]: [s.n.], 2009.

SHULMAN, L. S. **Those Who Understand: Knowledge Growth in Teaching**. *Educational Researcher*, fevereiro 1986. 4-14.

SIMON, H. A. **The Sciences of the artificial**. Cambridge, Massachusetts: MIT Press, 1996.

SITAR-TAUT, A.; MICAN, D. **Mobile learning acceptance and use in higher education during social distancing circumstances: an expansion and customization of UTAUT2**. *Emerald*, Reino Unido, v. 45, n. 5, p. 1000-1019, Abril 2021. ISSN DOI 10.1108/OIR-01-2021-0017.

STOENESCU, R. **quasar.dev**, 2015. Disponível em: <<https://quasar.dev/>>. Acesso em: 11 janeiro 2022.

SUN, G.; SHEN, J. **Facilitating Social Collaboration in Mobile Cloud-Based Learning: A Teamwork as a Service (TaaS) Approach**. *IEEE Transactions on Learning Technologies*, Julho 2014. 207-220.

TABOR, S. W. **Making Mobile Learning Work: Student Perceptions and Implementation Factors**. *Journal of Information Technology Education: Innovations in Practice*, Idaho, v. 15, p. 75-98, 2016.

TEIXEIRA, F. G. **HyperCAL 3D 2.D – A segunda geração do modelador 3D para geometria descritiva**. XXXVIII COBENGE: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA. Fortaleza: [s.n.]. 2010.

TEIXEIRA, F. G. **HyperCAL3D - Manual do Usuário**. Grupo de pesquisa

Vitual Design, Porto Alegre, 2023. Disponível em: <<https://www.ufrgs.br/hypercal3d>>.

TEIXEIRA, F. G. *et al.* **Geometria Descritiva: Uma Abordagem Concreta**. Simpósio Nacional de Geometria Descritiva e Desenho Técnico / International Conference on Graphics Engineering for Arts and Design. Curitiba: [s.n.]. 2007.

TEIXEIRA, F. G.; SANTOS, **HyperCAL 3D, UM SISTEMA INOVADOR PARA AUXÍLIO AO PROCESSO DE ENSINO DE GEOMETRIA DESCRITIVA**. Simpósio Nacional de Geometria Descritiva e Desenho Técnico e International Conference on Graphics for Arts and Design. Florianópolis/SC: [s.n.]. 2013.

TEIXEIRA, F. G.; SANTOS, S. L. **HyperCAL 3D, uma Ferramenta Computacional para Apoio do Processo de Ensino-Aprendizagem de Geometria Descritiva**. *Design & Tecnologia* 6, p. 20-32, 2013.

TRAXLER, J. **Defining, Discussing, and Evaluating Mobile Learning: The moving finger writes and having writ....** International Review of Research in Open and Distance Learning, Wolverhampton, UK, Junho 2007. 22-33.

UFSC, G. **GQS Grupo de Qualidade de Software. GQS.UFSC**, 2022. Disponível em: <<http://www.gqs.ufsc.br/usability-engineering/match/>>. Acesso em: 8 Outubro 2022.

UFSC, G. **MATCH – Measuring Usability of Touchscreen Phone Applications. GQS UFSC**, 2022. Disponível em: <<http://www.gqs.ufsc.br/usability-engineering/match/>>. Acesso em: 10 out. 2022.

VALENTE, M. T. **Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade**. Minas Gerais: Independente, v. 1, 2022.

VENKATESH, V. *et al.* **USER ACCEPTANCE OF INFORMATION.MIS Quarterly**, v. 27, p. 425-478, Setembro 2003. ISSN 3.

WIGDOR, D.; WIXON, D. **Brave NUI World Designing Natural User Interfaces for Touch and Gesture**. Burlington: Elsevier Inc., 2011.

YOU, E. **Vuejs.org**, 2014. Disponível em: <<https://vuejs.org/>>. Acesso em: 11 janeiro 2022.

ZHANG, Y.A. **Design of Mobile Teaching and Learning in Higher Education**. In: ZHANG, Y. A. **Handbook of Mobile Teaching and Learning** and. 1. ed. Wollongong, NSW, Australia: Springer, v. 1, 2015. Cap. 1, p. 3-10.

Apêndice A

Lista completa de aplicativos encontrados nas plataformas Google Play e Apple Store com a pesquisa *3D modeling*

Aplicativo	Para que se destina	Plataforma de busca
3D Modeling App	Modelagem e edição 3D	Google Play/ Apple Store
d3D Sculptor - 3D	Escultura	Google Play/ Apple Store
Nomad Sculpt	Escultura	Google Play
Sculpt+	Escultura	Google Play
OnShap 3D CAD	Modelagem e edição 3D	Google Play/ Apple Store
Wuweido	Modelagem e edição 3D	Google Play/ Apple Store
Moblo - 3D furniture modeling	Modelagem de moveis e RA	Google Play
Prisma3D	Modelagem e edição de 3D	Google Play
SketchUp Viewer	Visualização de projetos	Google Play/ Apple Store
Qubism 3D modeling	Modelagem e edição 3D	Google Play
Tinkercad	Modelagem e edição 3D	Google Play/ Apple Store
SDF 3D	Modelagem e edição 3D	Google Play
Mega Voxels	Modelagem voxel	Google Play
Sketchfab	Modelos prontos para VR e RA	Google Play
Makers Empire 3D - 3D Printing	Impressão 3D	Google Play
Emb3D 3D Model Viewer	Visualizador de modelos 3D	Google Play/ Apple Store
Fusion 360	Otimização do trabalho em equipe	Google Play/ Apple Store
3D Designer - Modelagem 3d	Personagem 3D	Google Play
Magic Poser	Posicionar Personagens	Google Play/ Apple Store
EI Pose 3D	Posicionar Personagens	Google Play
3D Printing Models Free - Thinger	Impressão 3D	Google Play
Learn 3D Modeling	Video Tutorial de Modelagem 3D	Google Play
EDS 3D Modeling tool	Modelagem e edição 3D	Google Play
3DScanner - Photos to 3D model	Scanner	Google Play
Creality Cloud	Impressão 3D	Google Play
WIDAR - 3D Scan & Edit	Scanner	Google Play
3D Engineering Animation	Modelo de motor animado	Google Play
3D Model Editor for Minecraft	Modelos para Minicraft	Google Play
3D Mannequins	Modelos 3d de animais	Google Play
Polycam - 3D Scanner	Scanner	Google Play/ Apple Store
Shaperyard	Modelagem e edição 3D	Apple Store
Live Home 3D: Projetos de Casa	Design de casas	Apple Store
uMake	Maquetes	Apple Store
Tetra – Easy 3D Creation	Modelagem e edição 3D	Apple Store
Sketch 3D	Modelagem e edição 3D	Apple Store
Pose Croquis	Posicionar Personagens	Apple Store
Poseit	Posicionar Personagens	Apple Store
Blender Keys	Atalhos do Blender	Apple Store
Modelos CAD 3D engenharia	Modelos prontos	Apple Store
BIMx	Visualização de modelos	Apple Store
Scaniverse – Lidar 3D Scanner	Scanner	Apple Store
Keyplan 3D lite	Modelagem de ambientes	Apple Store
Sculpted	Escultura	Apple Store
Sketshbook	Desenho	Apple Store
Putty 3D	Escultura	Apple Store

Lista completa de aplicativos encontrados nas plataformas Google Play e Apple Store com a pesquisa *Technical drawing*

Aplicativo	Para que se destina	Plataforma de busca
Sketch Box Free (Easy Drawing)	Esboços	Google Play
Technical Drawing	Tutoriais	Google Play
Technical Drawing Mock Tests for Best Results	Simulação de desenho técnico	Google Play
Concepts: esboço/notas/desenho	Esboço e notas	Google Play
LineDraw : Engineering Drawing	Ensino de desenho técnico	Google Play

Engineering Drawing App	Imagens de desenhos técnicos	Google Play
AutoCAD – Editor de DWG	Desenho CAD	Google Play / Apple Store
UVCAD - CAD 2D Drawing & Drafting Editor & Viewer	Desenho CAD	Google Play
Technical Drawing	Jogo	Google Play
Engineering drawing app	Livros	Google Play
DWG FastView-CAD Viewer	Visualizador de arquivos	Google Play
CCS Technical Drawings	Diagramas	Google Play
Desenho de engenharia	Coleção de desenhos	Google Play
Engineering Drawing App	Livros	Google Play
ED1	Modelos prontos	Google Play
Symbols - Engineering, Drawing	Simbolos de engenharia	Google Play
CAD Desenhe um desenho mais 3D	Desenho CAD	Google Play
Engineering drawing MCQ	Quiz de engenharia	Google Play
Engineering drawing books	Livros	Google Play
Cad Cam Drawing	Lista de desenhos CAD	Google Play
OnShap 3D CAD	Desenho CAD	Google Play
Engenharia Mecânica Desenho	Imagens e textos de desenhos técnicos	Google Play
Online Technical Drawing Test	Testes de desenho com ranking	Google Play
Desenhos Simples: Rascunho	Esboço	Google Play
Learn Engineering Drawing	Imagens e textos de desenhos técnicos	Google Play
Skedio: Desenho vetorial fácil	Desenho vetor	Google Play
Engineering Drawing Books	Livros	Google Play
GnaCAD	Desenho CAD	Google Play
CorelCAD Mobile	Desenho CAD	Apple Store
Rollercoast Builder Builder Travel	Jogo	Apple Store
CCS Technical Drawings	Modelos pontos	Apple Store
Blueprints and Scan App for Technical	Scanner	Apple Store
Kingdom Draw	Jogo	Apple Store
Ultimate Technical Desing	Pintura	Apple Store
AirDiffuser	Venda de Ar condicionado	Apple Store
DuctSizeCalc	Ar condicionado	Apple Store
Iso2Ortho	Projeções de modelos	Apple Store
FingerCAD	Desenho CAD	Apple Store
A Simple PhotoEditor	Edição de Fotos	Apple Store
MathCanvas	Calculos	Apple Store
Simplified!	Ensino Autocad	Apple Store