

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

WAGNER OLIVEIRA DA COSTA

**ANÁLISE DE ESTÁGIOS, TIPOS E TÉCNICAS DE TESTES
UTILIZADAS EM SISTEMAS DE REALIDADE VIRTUAL E
AUMENTADA**

MARÍLIA
2014

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

WAGNER OLIVEIRA DA COSTA

**ANÁLISE DE ESTÁGIOS, TIPOS E TÉCNICAS DE TESTES
UTILIZADAS EM SISTEMAS DE REALIDADE VIRTUAL E
AUMENTADA**

Trabalho de Curso apresentado ao Curso Ciências da Computação da Fundação de Ensino “Eurípides Soares da Rocha”, mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Ciências da Computação.

*Orientador:
Prof. Ms. Adriano Bezerra*

MARÍLIA
2014

Costa, Wagner Oliveira

Análise de Estágios, tipos e técnicas de testes utilizados em sistemas de realidade virtual e aumentada / Wagner Oliveira da Costa; orientador: Adriano Bezerra. Marília, SP: [s.n.], 2014.

86 f.

Trabalho de Curso Ciências da Computação - Curso de Ciências da Computação, Fundação de Ensino “Eurípides Soares da Rocha”, mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, Marília, 2014.

1. Testes 2. Funcional 3. Estrutural 4. Caixa branca 5. Caixa Preta 6. Realidade Virtual 7. Realidade Aumentada

CDD: 006 – Realidade Aumentada



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Wagner Oliveira da Costa

Critérios de testes utilizados em Realidade Virtual e Aumentada

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 9,0 (nove)

Orientador: Adriano Bezerra

1º. Examinador: Mauricio Duarte

2º. Examinador: Leonardo Castro Botega

Marília, 01 de dezembro de 2014.

À Deus, pelo esplendor da vida, presente em todas as atividades;

Aos amigos pelo incentivo; Fabio Oliveira da Costa, Gabriela Oliveira, Rafael Caetano, Rodrigo Wilson Coradi, Monyque Nalia, João Paulo Damasceno, Rafael Gasparetti e João Vitor Passareli.

À minha família.

AGRADECIMENTOS

Agradeço as manifestações de carinho e apreço, recebidas de todos os colegas da Fundação de Ensino “Eurípides Soares da Rocha”.

Ao Prof. Ms. Orientador Adriano Bezerra, braço amigo de todas as etapas deste trabalho.

A minha família, pela confiança e motivação.

Aos amigos e colegas, pela força e pela vibração em relação a esta jornada.

Aos professores e colegas de Curso, pois juntos trilhamos uma etapa importante de nossas vidas.

A todos que, com boa intenção, colaboraram para a realização e finalização deste trabalho.

Ao professor coordenador de TCC que sempre me incentivou a estudar mais para dar maior qualidade à minha monografia.

“A maior recompensa do trabalho não é o que nos pagam por ele, mas aquilo em que nos transforma.”

John Ruskin.

Costa, Wagner Oliveira. **Análise de Estágios, tipos e técnicas de testes utilizados em sistemas de realidade virtual e aumentada**. 2014. 50 f. Trabalho de Curso (Bacharelado em Ciências da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino “Eurípides Soares da Rocha”, Marília, 2014.

RESUMO

Dentre as técnicas de verificação e validação (V&V), a atividade de teste de softwares é um dos elementos mais importantes que visam aprimorar a produtividade e fornecer evidências da confiabilidade e qualidade do produto de software implementado. Este tipo de atividade é considerada uma das atividades mais dispendiosas do processo de desenvolvimento. Sendo assim as atividades de teste deveriam estar presentes em todos os tipos de projetos de software. Inclusive no desenvolvimento de sistemas não convencionais como os que utilizam a Realidade Virtual (RV) e Realidade Aumentada (RA), que possuem particularidades não funcionais a serem testadas, necessitando-se explorar com maior intensidade suas atividades de teste e avaliação. Este trabalho tem como objetivo apresentar a análise dos estágios, tipos e técnicas utilizadas em testes de sistemas de RA e RV ultimamente desenvolvidas no Brasil, independente do segmento do sistema e por fim apresentar quais os testes que mais possuem verificações. Contribuindo na área de RV com informações que poderão ser utilizadas em testes nesta área, na qual hoje é pouco explorada no quesito de engenharia de software.

Palavras-chave: Caixa Preta, Caixa Branca, Realidade Virtual, Realidade Aumentada, Teste Funcional, Teste Estrutural.

Costa, Wagner Oliveira. **Análise de Estágios, tipos e técnicas de testes utilizados em sistemas de realidade virtual e aumentada**. 2014. 50 f. Trabalho de Curso (Bacharelado em Ciências da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino “Eurípides Soares da Rocha”, Marília, 2014.

ABSTRACT

Among the techniques of verification and validation (V&V), the activity of software testing is one of the most important elements that aim to improve productivity and provide evidence of the reliability and quality of the implemented software product. This type of activity is considered one of the most costly activities of the development process. Therefore the testing activities should be present in all types of software projects. Including the development of unconventional systems such as those using the Virtual Reality (VR) and Augmented Reality (AR), which have no functional characteristics to be tested, need to explore more intensely your test and evaluation activities. This work aims to present the analysis of the stages, types and techniques used in AR and VR systems tests recently developed in Brazil, regardless of system segment and finally present what tests that have more checks. Contributing in the RV area with information that could be used in tests in this area , in which today is little explored in the issue of software engineering.

Keywords: Augmented Reality, Black Box, Functional Testing, Structural Testing, Virtual Reality, White box.

LISTA DE ILUSTRAÇÕES

Figura 1 – Estratégia de teste.....	19
Figura 2 – Equipamentos de Realidade Virtual.....	28
Figura 3 – Exemplo de Realidade Virtual.....	28
Figura 4 – Sistema de RA Não Imersivo.....	30
Figura 5 – Sistema de RA Imersivo.....	30
Figura 6 – Diagrama de Classes.....	34
Figura 7 – Classificação de aceitação dos usuários pelo software.....	35
Figura 8 – As localizações dos pontos da medida.....	37
Figura 9 – Os valores médios usando o OLPC. Primeiro andar à esq. e segundo andar à dir.	37
Figura 10 – Arquitetura do SwImax.....	39
Figura 11 – Visão geral da interface do SwImax.....	40
Figura 12 – Relevância do uso do simulador SwImax como ferramenta de auxílio ao ensino de redes sem fio.....	41
Figura 13 – Arquitetura proposta em alto nível.....	43
Figura 14 – Diagrama de classes do jogo.....	43
Figura 15 – Interface gráfica da Aplicação – Curador do Museu.....	45
Figura 16 – Modelo numérico de terreno e curvas de altitude.....	46
Figura 17 – Representação do Nó Multitexture.....	47
Figura 18 – Representação de um AV.....	48
Figura 19 – Screenshots da ferramenta.....	49
Figura 20 – Diagrama de execução da ferramenta.....	49
Figura 21 – Exemplo de uma cascata de classificadores armazenada em um arquivo XML que é utilizada na detecção de objetos pelas funções “cvLoad” e “cvDetectHaarCascade”.....	51
Figura 22 – Representação em 3D do sinal rastreado.....	52
Figura 23 – Mapeamento dos movimentos do anel rastreado para o controle da câmera virtual.....	53
Figura 24 – Calibração do dispositivo de navegação.....	53
Figura 25 – Screenshot do ambiente de testes utilizado no experimento.....	54
Figura 26 – Media de erros em cada dispositivo. M&T e Fingertrax.....	55
Figura 27 – Usuário sendo reconhecido pelo algoritmo de Diferença.....	56
Figura 28 – Cena desenvolvida de exemplo da utilização do protótipo.....	57
Figura 29 – Cena desenvolvida de exemplo da utilização do protótipo.....	57
Figura 30 – Visão do usuário de acordo com o reconhecimento pelo algoritmo de Diferença.....	58
Figura 31 – Visão de diferentes ângulos pelo mesmo usuário.....	58
Figura 32 – Exemplo da execução de aplicação.....	59
Figura 33 – MIT com o motor virtual – RV.....	61
Figura 34 – Motor Expandido – RA.....	61
Figura 35 – Arquitetura do sistema proposto.....	63
Figura 36 – Gráfico latência de Comunicação.....	64
Figura 37 – Gráfico de Escalabilidade.....	65
Figura 38 – Gráfico finalidade de uso da ferramenta.....	65
Figura 39 – Visão da Utilização dos Óculos de Imersão.....	66
Figura 40 – Fluxo do Sistema RA.....	67
Figura 41 – Protótipo estereoscópico transportável.....	71
Figura 42 – Estrutura do Ambiente Proposto.....	71
Figura 43 – Aplicação interativa desenvolvida com Panda 3D.....	72

Figura 44 – Um marcador é utilizado para mover o carro virtual, Vê-se a imagem exibida mudando conforme o observador se move em relação a mesa.....	72
Figura 45 – A - Imagem dos marcadores	73
Figura 46 – B - Imagem com os objetos virtuais sobre os marcadores	74
Figura 47 – Captura de tela da aplicação no contexto da universidade.....	76
Figura 48 – Captura de tela da aplicação exibindo mídia de vídeo.....	76
Figura 49 – Técnicas de Teste (Caixa Branca, Caixa Preta e Baseado em Defeitos)	82
Figura 50 – Estágios de Teste (Unidade, Integração, Sistema e Aceitação)	83
Figura 51 – Tipos de Teste: (Todos) ou os principais.	84
Figura 52 – Técnicas de Teste (Não Utilizaram Testes)	82

LISTA DE TABELAS

TABELA I - Fluxo Principal	67
TABELA II - Tratamento de Exceções	68
TABELA III - Teste Funcional	69
TABELA IV - Tempo De Execução Das Tarefas Pelos Usuários	77
TABELA V - Acertos e Erros dos usuarios na execução das tarefas.....	77
TABELA VI - Pontos de Interesse da Aplicação de RA Móvel no Campus Universitário.....	78
TABELA VII – Classificação dos Trabalhos.....	87

LISTA DE ABREVIATURAS E SIGLAS

INPI: Instituto Nacional de Propriedade industrial

RA: Realidade Aumentada

RV: Realidade Virtual

V&V: Verificação e Validação

FUN: Apenas informações quanto a Funcionalidade

TST: Teste de Sistema

TSG: Teste de Segurança

TSR: Teste de Recuperação

TAD: Teste de Avaliação de Desempenho

TSC: Carga

TEU: Teste Unitário

TEI: Teste de Integração

TED: Teste de Integração descendente

TEA: Teste de Integração ascendente

TAA: Teste de Aceitação Alfa

TAB: Teste de Aceitação Beta

REQ: Requisitos

TER: Teste de Regressão

OpenCV: Open Computer Vision

SUMÁRIO

INTRODUÇÃO	15
CAPÍTULO 1 REVISÃO TEÓRICA	17
1.1 VISÃO GERAL DE TESTE DE SOFTWARE	17
1.2 TESTE FUNCIONAL	20
1.2.1 Teste de Sistema	21
1.2.2 Teste de Segurança	21
1.2.3 Teste de Recuperação	22
1.2.4 Teste de Desempenho	22
1.3 TESTE ESTRUTURAL	22
1.3.1 Teste Unitário	23
1.3.2 Teste Integração	24
1.3.3 Teste Integração descendente	24
1.3.4 Teste Integração ascendente	24
1.3.5 Teste Regressão	24
1.4 IMPLANTAÇÃO	25
1.4.1 Teste de Aceitação Alfa	25
1.4.2 Teste de Aceitação Beta	26
CAPÍTULO 2 REALIDADE VIRTUAL E AUMENTADA.....	27
2.1 REALIDADE VIRTUAL.....	27
2.2 REALIDADE AUMENTADA.....	29
CAPÍTULO 3 ANÁLISE DE ESTÁGIOS, TIPOS E TÉCNICAS DE TESTES UTILIZADAS EM SISTEMAS DE REALIDADE VIRTUAL.....	32
3.1 NAVEGAÇÃO, RECONHECIMENTO DE GESTOS E CONTROLE DE INTERFACE NO SISTEMA ITV UTILIZANDO O DISPOSITIVO KINECT	32
3.2 MODELAGEM DE OBJETOS TRIDIMENSIONAIS PARA UM AMBIENTE INTERATIVO DE INSTRUÇÕES E TÉCNICAS VIRTUAIS	35
3.3 MODELAGEM, SIMULAÇÃO, E VISUALIZAÇÃO IMERSIVA DE REDES SEM FIO	36
3.4 SWIMAX: UM SIMULADOR EM REALIDADE VIRTUAL PARA AUXILIAR O ENSINO DO PADRÃO WIMAX.....	38
3.5 UM AMBIENTE PARA O DESENVOLVIMENTO DE APLICAÇÕES DE REALIDADE VIRTUAL BASEADAS EM AGLOMERADOS GRÁFICOS	41
3.6 UMA ARQUITETURA PARA INTEGRAÇÃO DE SISTEMAS HÁPTICOS E ENGINES DE JOGOS	42
3.7 MUSEU 3I: PUBLICAÇÃO E VISITAÇÃO ONLINE DE ACERVOS TRIDIMENSIONAIS.....	44
3.8 X3D E INTEGRAÇÃO MULTIMÍDIA PARA REPRESENTAÇÃO DE UM SÍTIO ARQUEOLÓGICO	45
3.9 APLICAÇÕES BASEADAS EM GRAFO DE CENA – UMA ABORDAGEM ESTRUTURAL PARA CRITÉRIOS DE TESTE.....	47
3.10 RECONHECIMENTO DE GESTOS COM SEGMENTAÇÃO DE IMAGENS DINÂMICAS APLICADAS A LIBRAS	50
3.11 CONTROLE DE NAVEGAÇÃO EM AMBIENTES VIRTUAIS 3D ATRAVÉS DO RASTREAMENTO DE OBJETOS	52

3.12 VISUALIZAÇÃO DE AMBIENTES VIRTUAIS COM SIMULAÇÃO DE PROJEÇÃO HOLOGRÁFICA.....	55
--	----

CAPÍTULO 4 ANÁLISE DE ESTÁGIOS, TIPOS E TÉCNICAS DE TESTES UTILIZADAS EM SISTEMAS DE REALIDADE AUMENTADA.....59

4.1 DESENVOLVIMENTO DE UM JOGO TRIDIMENSIONAL COM REALIDADE AUMENTADA	59
---	----

4.2 REALIDADE AUMENTADA PARA AUXILIAR O APRENDIZADO DE MOTOR ELÉTRICO.....	60
--	----

4.3 DESENVOLVIMENTO DE UMA ARQUITETURA PARA DISTRIBUIÇÃO DE REALIDADE AUMENTADA NA WEB APLICADA AO ENSINO DE MOTORES DE CORRENTE CONTINUA	62
---	----

4.4 REALIDADE AUMENTADA IMERSIVA NO AMBIENTE TELEVISIVO	66
---	----

4.5 UMA PLATAFORMA PARA VISUALIZAÇÃO ESTEREOSCÓPICA HORIZONTAL	70
--	----

4.6 INTELIGÊNCIA ARTIFICIAL APLICADA A JOGOS DE TABULEIRO COM REALIDADE AUMENTADA.....	73
--	----

4.7 UMA APLICAÇÃO DE REALIDADE AUMENTADA MÓVEL PARA AMBIENTES INDOOR E OUTDOOR.....	74
---	----

CAPÍTULO 5 RESULTADOS.....80

CONCLUSÃO.....89

REFERÊNCIAS.....90

INTRODUÇÃO

O processo de Validação e Verificação V&V de softwares é dividido em técnicas, tipos e estágios definidos pela engenharia de software, estes devem ser aplicados corretamente em cada fase de desenvolvimento visando à qualidade do software final, minimizando os erros na fase inicial que o deixará mais confiável, e o custo para correção de um problema identificado no começo da implementação é bem menor.

O objetivo do trabalho é realizar várias pesquisas e análises de artigos e monografias e então apresentar em formas de gráficos, e discursivamente as maneiras mais utilizadas nos trabalhos apresentados para realização dos testes e seus estágios, tipos e técnicas que se fizeram necessários em cada sistema.

A questão é que, através do estudo teórico será possível avaliar reais situações dos testes feitos em diversos trabalhos de Realidade Virtual (RV) e Realidade Aumentada (RA), e assim servirá como base para trabalhos futuros na área, e também para estudos em relação aos critérios utilizados para a aplicação de testes nestes domínios, que é o ambiente virtual; cada vez mais explorado na atualidade. Sistemas nesta área necessitam de resposta em tempo real, para isso se faz necessário que seja aplicados testes diferenciados para obter resultados satisfatórios.

Foi utilizado para este trabalho de conclusão de curso uma pesquisa bibliográfica em relação a sistemas desenvolvidos na área de RV e RA, estes trabalhos que foram analisados são de Periódicos como, Portal de Periódicos da Capes, Artigos de universidades, como a Universidade de São Paulo (USP), Universidade Federal do Pará (UFPA), Workshop de RA e RV dos anos de 2010, 2013 e 2014 (WRVA e SRV), entre outras que serão citadas posteriormente. Com a finalidade de coletar informações e realizar uma análise quantitativas com os resultados classificando todos os projetos de acordo com a forma que foram realizados os devidos testes de software.

O Capítulo 1 apresenta a caracterização do estudo, incluindo a delimitação do tema, objetivos, justificativa, metodologia e a estrutura do trabalho;

O Capítulo 2 contém uma revisão bibliográfica dos principais conceitos ligados ao teste de software, onde descreve os testes estruturais (caixa branca) e testes funcionais (caixa preta) e suas subdivisões. Coloca-se também de uma forma geral as características e diferenças entre RV e RA;

O Capítulo 3 relata a análise dos trabalhos, descrevendo cada um deles resumidamente, analisando os procedimentos de teste de software que cada um apresenta;

O Capítulo 4 apresenta os resultados analisados de forma gráfica e quantitativa, apresenta todos os trabalhos em uma tabela e analisa as formas utilizadas;

O Capítulo 5 apresenta a conclusão do trabalho e as informações adquiridas através dele, incluso também a proposta de um trabalho futuro, onde poderia centralizar as informações de cada trabalho e cada autor em um servidor de análises, atualizando com seus métodos e contribuindo para que o setor de testes em RA e RV seja mais documentada e concentrada com técnicas diferenciadas.

CAPÍTULO 1 REVISÃO TEÓRICA

A atividade de engenharia de software possui como uma de suas premissas a qualidade do produto disponibilizado ao cliente, e para isso utiliza das atividades de testes para certificar o maior número de variáveis possíveis, que podem implicar em falhas ou erros durante a utilização do produto.

O teste na definição de Pressman (2011, p.402), é um conjunto de atividades planejadas com determinada antecedência e que podem ser executadas de forma sistemática. Por esta razão, é necessário que seja definido um modelo (*template*) de teste para o processo de software, ou seja, um conjunto de etapas que deverão ser executadas no qual poderá colocar técnicas específicas de projeto de caso, de teste e métodos de teste.

A concepção de Shooman trata o teste como um processo individualista e de acompanhamento conforme a progressão do projeto.

De muitas formas, o teste é um processo individualista e o número de tipos diferentes de testes varia tanto quanto as diferentes abordagens de desenvolvimento. Por muitos anos, nossa única defesa contra os erros de programação era um projeto cuidadoso e a inteligência do programador. Estamos agora em uma nova era na qual as modernas técnicas de projeto [e revisões técnicas] estão ajudando a reduzir a quantidade de erros iniciais inerentes ao código. De maneira semelhante, diferentes métodos de teste estão começando a se agrupar em várias abordagens e filosofias distintas. (PRESSMAN, 2011, p. 401 *apud* SHOOMAN, 1983)

O processo de teste busca atingir dois objetivos diferentes, segundo Ian Sommerville (2011, p.144): demonstração ao desenvolvedor, e ao cliente o pleno funcionamento do sistema, e que o mesmo atende aos seus requisitos definidos no projeto, e demonstra que o software não se comporta de maneira incorreta, de forma indesejável ou diferente do que foi especificado.

Desta forma, o desenvolvedor deverá identificar as melhores práticas de teste e aplicar sistematicamente no produto.

1.1 VISÃO GERAL DE TESTE DE SOFTWARE

O teste de Software é submetido a testes na busca de que seja revelado erros cometidos inadvertidamente quando o mesmo foi projetado e em sua construção.

O autor Pressman (2011, p.402), trata o teste de software como um elemento de um tópico mais amplo, também conhecido como verificação e validação. Neste sentido, a verificação trata do conjunto de tarefas, que tem por objetivo garantir que o software implementa corretamente um função específica, e a validação é referente a um conjunto de tarefas que asseguram que o software foi criado e pode ser rastreado segundo os requisitos do cliente.

Boehm (1981) define de outra forma:

Verificação: “Estamos criando o produto corretamente?”

Validação: “Estamos criando o produto certo?”

O teste em si, tem por objetivo apenas garantir que os requisitos ou especificações do negócio foram devidamente implementados. Este na maioria das empresas que os executam, são realizados normalmente pelos próprios desenvolvedores ou pelos próprios usuários. Como este processo de desenvolver um produto para o mercado cria aplicações com defeitos, é necessário descobrir aonde estão localizadas estes “bugs”. Levando em consideração o conceito de Qualidade de Software, este processo de teste é insuficiente.

“Definimos “defeito” (do inglês, fault) como sendo um passo, processo ou definição de dados incorretos e “engano” (mistake) como a ação humana que produz um defeito. Assim, esses dois conceitos são estáticos, pois estão associados a um determinado programa ou modelo e não dependem de uma execução particular.

O estado de um programa ou, mais precisamente, da execução de um programa em determinado instante é dado pelo valor da memória (ou das variáveis do programa) e do apontador de instruções. A existência de um defeito pode ocasionar a ocorrência de um “erro” (error) durante a execução do programa, que se caracteriza por um estado inconsistente ou inesperado. Tal estado pode levar a uma “falha” (failure), ou seja, pode fazer com que o resultado produzido pela execução seja diferente do resultado esperado.” (DELAMARO; MALDONADO; JINO, 2007, p. 02).

Segundo Delamaro (2007) as atividades de testes são complexas, diversos fatores podem colaborar para ocorrência de erros. Por exemplo, a utilização de um algoritmo incorreto para computar o valor das mensalidades a serem pagas para empréstimo, ou a não utilização de uma política de segurança em alguma funcionalidade do software são dois tipos distintos de engano e, de certa forma, encontra-se em níveis diferentes de abstração. O primeiro tipo de erro provavelmente está confinado a uma função ou rotina que implementada de forma incorreta uma dada funcionalidade. No segundo caso mesmo que exista uma certa política de segurança implementada de maneira correta, é preciso verificar se todos os pontos nos quais essa política deveria ser aplicada fazem-no de maneira correta.

Desta forma possuem objetivos bem distintos e de uma forma geral pode-se definir as técnicas de caixa branca que são, por exemplo, o teste de unidade e teste de integração e caixa preta que é o teste de sistemas.

O teste de unidade tem em si o objetivo de testar as menores frações de um programa podendo ser procedimentos, métodos, classes ou funções. Assim tende a encontrar algoritmos incorretos ou mal implementados, estrutura de Dados incorretos, ou simples erros de programação. Como este teste é realizado separadamente do programa, é possível que o

próprio desenvolvedor possa testar à medida que realiza implementações, sem necessidade do sistema estar totalmente realizado.

O teste de integração é feito após serem testadas as unidades individualmente. A ênfase é dada na estrutura do sistema, conectando as unidades para que trabalhem juntas e seja possível verificar se a integração entre elas funciona de maneira adequada, e não leva erros por conta disso. É necessário ter conhecimentos nas estruturas do sistema interno e das interações existentes entre as partes do sistema, por isso, o teste também é realizado pela equipe de desenvolvimento.

Quando todas as portas estão testadas integradas, inicia-se o teste de sistema, no qual tem o objetivo de verificar se todas as funcionalidades especificadas nos documentos de requisitos, estão corretamente implementadas. Alguns aspectos que devem ser verificados e explorados são os fatores de correção, completude e coerência, assim como requisitos não funcionais como: segurança, e performance em robustez; estes testes normalmente são realizados por uma equipe extensa e especializada em testes de sistema.

Pressman (2011, p.404), estabelece uma visão ampla na estratégia de teste de software que pode ser vista e entendida como uma espiral, com o início ao centro e seguindo para a extremidade ilustrando as fases deste processo de software.

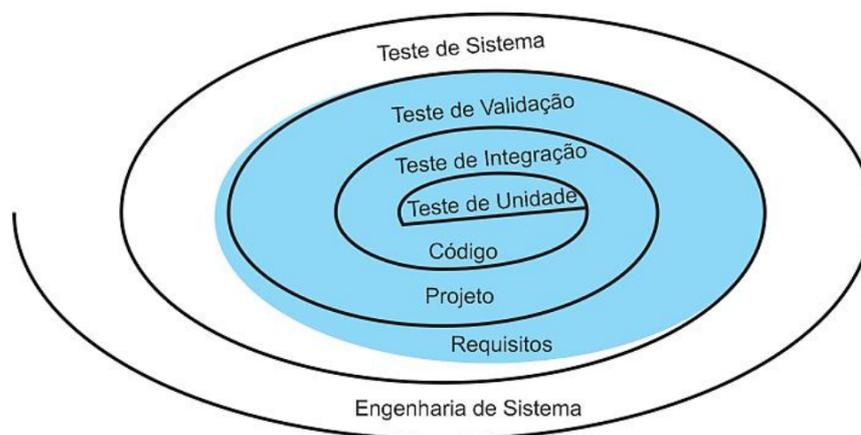


Figura 1 – Estratégia de teste

Fonte: PRESSMAN (2011, p.404)

Os testes unitários ficam ao centro do espiral(figura 1), representando pequenos fragmentos de requisitos que foram implementados (código fonte). Avançando pelo espiral notamos que a fase do projeto relaciona-se com testes de integração, com isso a cada novo *release* integrado é possível verificar e garantir que as novas funcionalidades não resultarão em erros nos fragmentos ou módulos implementados anteriormente. O teste de sistema realiza

a verificação de modo geral, se todas as funções e elementos implementados estão combinando de forma a resultar no software.

1.2 TESTE FUNCIONAL

O teste funcional é encontrado na literatura desde os anos 70, foi nesta época que começaram a surgir vários métodos em apoio a especificação de sistema, como a Análise e Projeto Estruturado de GANE e SARSON, no qual foi mencionado de forma indireta os aspectos de validação do sistema, relativos a satisfação dos seus requisitos funcionais. Desta forma, algumas técnicas são utilizadas para demonstrar e descrever tais requisitos, de uma forma completa e não ambígua, são técnicas que fazem parte dos critérios funcionais.

A técnica de teste funcional é utilizada no caso de projetos de teste, onde o programa ou sistema é considerado uma caixa preta, e para aplicação do teste é fornecido entradas e é avaliado as saídas geradas, verificando se estão conforme os objetivos especificados. No caso desta técnica os detalhes de implementação são desconsiderados e o software é avaliado segundo o ponto de vista do usuário.

Os autores DELAMARO, MALDONADO e JINO esclarecem e elegem os critérios mais conhecidos da técnica do teste funcional :

Os critérios mais conhecidos da técnica de teste funcional são Particionamento em classes de equivalência, análise do valor limite, grafo causa-efeito e *error-guessing*. Além desses, também existem outros, como, por exemplo, Teste Funcional Sistemático, Syntax Testing, State Transition Testing e Graph Matrix.

Como todos os critérios da técnica funcional baseiam-se apenas na especificação do produto testado, a qualidade de tais critérios depende fortemente da existência de uma boa especificação de requisitos. Especificações ausentes ou mesmo incompletas tornarão difícil a aplicação de critérios funcionais. Além disso, tais critérios também apresentam a limitação de não garantir que as partes essenciais ou críticas do produto em teste sejam exercitadas.

Por outro lado, os critérios funcionais podem ser aplicados em todas as fases de testes e em produtos desenvolvidos com qualquer paradigma de programação, pois não levam em consideração os detalhes de implementação.

(DELAMARO; MALDONADO; JINO, 2007, p. 11)

Uma das vantagens dos critérios da técnica funcional é que eles apenas precisam da especificação do produto para derivar os requisitos do teste, dessa forma eles podem ser aplicados igualmente a qualquer programa, independente de que seja procedimental ou orientado a objetos, ou a componentes de software, em razão do código fonte não ser necessário. No entanto, segundo ROOPER (1994), como os critérios funcionais são baseados

unicamente na especificação e não podem assegurar que os pontos críticos e essenciais do código foram cobertas pelo teste.

Os critérios apresentam limitações, por exemplo, elementos de uma mesma classe de equivalência se comportem de forma igual, na prática isso pode não ocorrer. Portanto, é fundamental que as técnicas de teste sejam compreendidas como um complemento e sejam aplicadas em conjunto para que ocorra a exploração de pontos de vista diferentes no momento do teste do software.

1.2.1 Teste de Sistema

O sistema é composto por diversos elementos que trabalham em conjunto e dependência (exemplo: software, hardware, informações e pessoas) e para o correto funcionamento são executados teste de integração de sistema e de validação.

O desenvolvedor deve se antecipar aos problemas em potencial da interface, Pressman apresenta 4 pontos a serem seguidos para se evitar o clássico problema de “procura do culpado”, onde os desenvolvedores de diversos elementos do sistema trocam acusações devido algum erro.

[...] (1) criar caminhos de manipulação de erro que testem todas as informações vindas de outros elementos do sistema, (2) executar uma série de testes que estimulem dados incorretos ou outros erros potenciais na interface do software, (3) registrar os resultados dos testes para usar como “evidência” se ocorrer à caça ao culpado, e (4) participar do planejamento e projeto de testes do sistema para assegurar que o software seja testado adequadamente. (PRESSMAN, 2011, p.418-419)

O teste de sistema é uma série de diferentes testes com a finalidade principal de exercitar totalmente o sistema, embora que para cada teste tem-se uma finalidade específica, o principal objetivo é que ao final todos os testes funcionaram no sentido de verificar se os elementos do sistema foram integrados corretamente e se executam as funções definidas a eles.

1.2.2 Teste de Segurança

O teste de segurança é baseado na elaboração de casos que possam comprometer as verificações de segurança de um programa, ou seja, protegê-lo de um acesso indevido.

Na conceito de Beizer (1984) sobre o teste de segurança: “A segurança de um sistema deve, naturalmente, ser testada quanto a invulnerabilidade por um ataque frontal – mas deve também ser testada quanto a invulnerabilidade por ataques laterais e pela retaguarda.”

Durante o teste de segurança, o testador faz o papel do indivíduo que tenta invadir o sistema, utilizando-se de todas as ferramentas disponíveis, inclusive software projetado para romper as defesas criadas, sobrecarga do sistema, etc.

Contudo, o principal papel do criador do sistema é tornar o custo da invasão maior que o valor das informações que podem ser obtidas.

1.2.3 Teste de Recuperação

Este tipo de teste busca a recuperação de falhas e a retomada do processamento em pouco ou nenhum tempo de parada. Em alguns casos o sistema tem que ser tolerante e suportar algumas falhas sem que ocorra a paralisação total do sistema, em outros casos a falha deverá ser corrigida dentro de um prazo limite de tempo para evitar perdas financeiras.

O teste de recuperação é um teste do sistema que trabalha para que as falhas ocorram de diversas formas, verificando se a recuperação foi executada corretamente. No caso de recuperação automática o teste irá avaliar a reinicialização, mecanismos de verificação, recuperação de dados e reinício, no caso de recuperação com a intervenção do programador o teste avalia o tempo médio utilizado na recuperação para definir se está dentro dos padrões aceitáveis.

1.2.4 Teste de Desempenho

O teste de desempenho analisa se o software além de executar as funções necessárias, atende os requisitos de desempenho aceitável em tempo de execução dentro de um contexto de software integrado.

A avaliação ocorre ao testar o desempenho de todas as etapas de teste, inclusive no nível de unidade, onde o desempenho do módulo individual poderá ser avaliado no decorrer do teste. No entanto, o desempenho efetivo somente poderá ser avaliado após todos os elementos do sistema estarem totalmente integrados.

Os testes de desempenho normalmente são realizados em conjunto ao teste de esforço e usualmente requer instrumentação de hardware e software, para isso, é necessário que seja monitorada a utilização dos recursos de forma efetiva e precisa, e ao monitorar o sistema o testador irá identificar o que leva a degradação e possível falha do sistema.

1.3 TESTE ESTRUTURAL

A técnica de teste estrutural (ou caixa branca), estabelece os requisitos do teste de acordo com uma determinada implementação e requer a execução de partes ou de componentes do programa.

O software tem os caminhos lógicos submetidos ao teste, utilizando de casos de teste que colocam a prova os conjuntos específicos de condições, laços, pares de definições e uso de variáveis. Os critérios aplicados são classificados com base na complexidade, no fluxo de controle e no fluxo de dados.

A técnica estrutural possui uma série de limitações e desvantagens decorrentes da limitação do teste de programa como estratégia de validação, e estes aspectos inserem sérios problemas na automatização do processo de validação. No entanto, esta técnica é utilizada como complementar ao processo de teste do software.

Segundo DELAMARO, MALDONADO, JINO, os principais problemas são:

Não existe procedimento de teste de propósito geral que possa ser usado para provar a correção de um programa;

Dados dois programas, é indecível se eles computam a mesma função;

É indecível, em geral, se dois caminhos de um programa, ou de programas diferentes computam a mesma função;

É indecível, em geral, se um dado caminho é executável, ou seja, se existe um conjunto de dados de entrada que leve a execução desse caminho.

É importante ressaltar ainda, as limitações inerentes à técnica estrutural:

caminhos ausentes: se o programa não implementa algumas funções, não existirá um caminho que corresponda aquela função; conseqüentemente, nenhum dado de teste será requerido para exercitá-la;

Correção coincidente: o programa pode apresentar, coincidentemente, um resultado correto para um dado em particular de entrada, satisfazendo um requisito de teste e não revelando a presença de um defeito; entretanto, se escolhido outro dado de entrada, o resultado obtido poderia ser incorreto. (DELAMARO, MALDONADO, JINO, 2007, p.47-48)

O teste estrutural é baseado no conhecimento da estrutura interna do programa, dessa forma os aspectos de implementação são fundamentais para que ocorra a geração /seleção dos casos de teste associados. Na aplicação os critérios utilizados no teste são classificados de forma geral em: a) critérios com base na complexidade; b) critérios com base em fluxo de controle e c) critérios baseados em fluxo de dados.

Os critérios estabelecem os requisitos de teste mais rigorosos a serem exercitados, fornecendo medidas objetivas para a adequação de um conjunto de teste de um determinado programa.

1.3.1 Teste Unitário

O teste de unidade tem como foco o esforço de verificação da menor unidade do projeto do software, utilizando do projeto no nível de componente.

Neste teste ocorre o enfoque na lógica interna de processamento e nas estruturas dos dados, respeitando o limite de um componente. Devido à limitação existente, este teste pode ser realizado em paralelo para diversos componentes.

O fluxo dos dados é testado através de uma interface de um componente antes de iniciar os demais testes. No caso dos dados não entrarem e saírem corretamente, todos os

demais testes são discutíveis. Após ensaiar as estruturas de dados o impacto local gerado sobre os dados globais deverão ser apurados no decorrer do teste de unidade.

Para a correta apuração deverá ocorrer um teste seletivo de caminhos de execução durante o teste de unidade. O testador terá que projetar testes para descobrir erros devido a computações errôneas, comparações incorretas ou fluxo de controle inadequado.

1.3.2 Teste Integração

O teste de integração é uma técnica para construir de forma sistemática a arquitetura de software, ao mesmo tempo em que direciona os testes para identificar possíveis erros associados com as interfaces. O objetivo é construir uma estrutura a partir de componentes testados em unidade.

No modelo de integração incremental o programa é estruturado e testado em pequenos incrementos. Nesta sistemática, os erros são mais fáceis de identificar, isolar e corrigir, aumentando a probabilidade de todas as interfaces serem testadas por completo.

1.3.3 Teste Integração descendente

No teste de integração descendente (*top-down*) os módulos são integrados a partir do módulo principal de controle, e segue a integração de cima para baixo ao longo da hierarquia de controle.

O módulo principal é utilizado como testador (*test driver*), e para a simulação de módulos a serem integrados deverão ser criados módulos pseudocontrolados.

A estratégia de integração descendente realiza a verificação dos principais pontos de controle ou decisão antecipada no processo de teste.

1.3.4 Teste Integração ascendente

No teste de integração ascendente (*bottom-up*), os módulos são integrados partindo dos níveis mais baixos da estrutura do programa, e segue a integração de baixo para cima ao longo da hierarquia de controle. Devido a integração ser realizada de baixo para cima os módulos pseudocontroladores são utilizados para realizar as chamadas aos módulos que estão sendo submetidos aos testes.

A medida que ocorre a integração para cima a necessidade dos pseudocontroladores de testes separados diminui.

1.3.5 Teste Regressão

No teste de regressão, a cada novo módulo acrescentado no teste de integração, ocorre mudança no software. Novos caminhos de fluxo de dados são estabelecidos, com isso podem ocorrer novas entradas e saídas.

As alterações realizadas podem interferir na harmonia já existente, causando problemas com funções que funcionavam corretamente. Visualizando estrategicamente o cenário do teste de integração, o teste de regressão nada mais é do que a execução do mesmo subconjunto de testes já executados, para assegurar que as modificações não implicaram em efeitos colaterais.

Na medida em que o avanço do teste de integração evolui, a quantidade de teste de regressão pode crescer muito. Portanto, o conjunto de teste de regressão deve ser projetado para testar apenas uma ou mais classes de erros em cada uma das funções principais do programa.

1.4 IMPLANTAÇÃO

O objetivo desses testes é obter um feedback dos clientes ou usuários do sistema antes do sistema entrar em produção ou ser disponibilizado para comercialização. Estes são os testes de aceitação que na maioria das vezes são realizados pelos usuários do software ou até mesmo pelos seus possíveis clientes (aceitação beta), todavia, na maioria das vezes a equipe de testes trabalha junto ao cliente no intuito de ajudar o usuário a verificar se a ferramenta está funcionando adequadamente.

A única preocupação que os usuários possuem é que o software execute corretamente o que foi especificado por eles, para o analista de requisitos da empresa de desenvolvimento.

1.4.1 Teste de Aceitação Alfa

Os testes de aceitação são realizados quando o produto é personalizado para um cliente, com a finalidade de permitir o cliente validar todos os requisitos. Nesta modalidade os teste são conduzidos pelo usuário final e não pelos engenheiro de software.

No caso de um software desenvolvido para ser utilizado por muitos clientes, os construtores utilizam um processo chamado de Teste Alfa e beta para descobrir os erros que normalmente somente o usuário final conseguirá identificar.

Na definição de Pressman (2011, p.417) teste alfa é conduzido na instalação do desenvolvedor por um grupo representativo de usuários finais. O software é usado em um cenário natural com o desenvolvedor “espiando por cima dos ombros” dos usuários, registrando os erros e os problemas de uso. Os testes alfas são conduzidos em um ambiente controlado.

Neste processo de teste alfa é realizado antes da entrega final ao cliente e o mesmo é executado ate o momento em que o cliente aceite que o software seja entregue.

1.4.2 Teste de Aceitação Beta

Diferente do teste alfa, neste sistema geralmente o desenvolvedor não está presente e consequentemente o ambiente em que ocorre não é controlado pelo desenvolvedor. Os resultados são enviados periodicamente para análise e correção conforme o caso, após as modificações é disponibilizado aos demais clientes.

“O teste beta é conduzido nas instalações de um ou mais usuários finais. Diferentemente do teste alfa, o desenvolvedor geralmente não está presente. Portanto, o teste beta é uma aplicação “ao vivo” do software em um ambiente que não pode ser controlado pelo desenvolvedor. O cliente” registra todos os problemas (reais ou imaginários) encontrados durante o teste beta e relata esses problemas para o desenvolvedor em intervalos regulares. Como o resultado dos problemas relatados durante o teste beta, os engenheiros de software fazem modificações e então preparam a liberação do software para todos os clientes.” (PRESSMAN, 2011, p.418)

CAPÍTULO 2 REALIDADE VIRTUAL E AUMENTADA

Neste capítulo serão abordados os principais aspectos relacionados à Realidade Aumentada e também em relação à Realidade Virtual, que representam técnicas de interface computacional que levam em conta o espaço tridimensional. Sendo assim o usuário atua de forma multissensorial, explorando por meio da visão, audição e tato.

2.1 REALIDADE VIRTUAL

A realidade virtual pode ser definida de várias formas, na definição de KIRNER e SISCOOTTO (2017) propõe que:

“Realidade virtual é uma interface avançada para aplicações computacionais, que permite ao usuário navegar e interagir, em tempo real, com um ambiente tridimensional gerado por computador, usando dispositivos multissensoriais” (KIRNER; SISCOOTTO; 2007)

Na definição de Burdea e Coiffet sobre a RV: “realidade virtual é uma interface computacional avançada que envolve simulação em tempo real e interações, através de canais multissensoriais” (Burdea e Coiffet, 1994).

Pode-se dividir em duas formas, RV imersiva e não imersiva. No primeiro conceito o usuário que esta manipulando a aplicação em RV é transportado para um domínio do software, ou seja, o sistema faz com que ele se sinta completamente imerso no mundo virtual, e assim poderá interagir com os objetos através de objetos multissensoriais como capacete de realidade virtual (HMD - Head Mounted Display) ou salas de multi-projeção por exemplo. Na segunda forma, o usuário apenas interage com a aplicação, mas é preservado o senso de presença no mundo real.

De acordo com a definição de Romero Tori, Kirner e Siscoutto:

A Realidade Virtual (RV) é, antes de tudo, uma interface avançada do usuário para acessar aplicações executadas no computador, tendo como características a visualização de, e movimentação em, ambientes tridimensionais em tempo real e a interação com elementos desse ambiente. Além da visualização em si a experiência do usuário de RV pode ser enriquecida pela estimulação dos demais sentidos como tato e audição. (TORI, KIRNER, SISCOOTTO, 2006, p.6).

A RV proporciona a sensação de que o individuo está dentro do ambiente e o grau desta sensação depende da ativação dos sentidos humanos, neste ambiente os sentidos que mais são provocados são a visão, a audição e o tato, o primeiro é o que tem maior influência na percepção do ambiente, e na ordem apresentada permanece a sequência aos estímulos humanos, na Figura 2 são apresentados os equipamentos utilizados na RV.



Figura 2 – Equipamentos de Realidade Virtual
Fonte: TAVARES, 2009, p.18

RV é uma interface aplicada a comunicação entre o ser humano e a máquina. A utilização pode ser realizada por qualquer aplicação executável por um software. Este ambiente virtual normalmente é tridimensional, com a finalidade de simular com o máximo de fidelidade o mundo real, embora não há nada que impeça que seja utilizado elementos 2D, como menus e botões, a Figura 3 demonstra um exemplo de RV.



Figura 3 – Exemplo de Realidade Virtual
Fonte: TAVARES, 2009, p.17

Segundo TAVARES (2009), neste ambiente de RV os sentidos são provocados através de dispositivos multissensoriais, que geram a interação e a imersão do usuário no ambiente virtual. O termo multissensorial está relacionado diretamente com a possibilidade de despertar diversos sentidos, ou seja, quanto mais sentidos forem utilizados pelo sistema, mais realístico será o ambiente virtual.

O autor STEUER (1993), estabelece que a interatividade nos sistemas de RV pode ser entendida como a capacidade do próprio sistema reagir a ações realizadas pelo usuário, sendo capaz de modificar o ambiente.

Os avanços no desenvolvimento dos processadores a partir da metade e que seguiu até o final dos anos 80, tornou mais viável os sistemas de alta performance e fez surgir novos dispositivos de exibição mais acessível, com execução mais leve e com resoluções melhores. Com a evolução para a tecnologia capaz de detectar movimentos e a posição o resultado foi a melhora na precisão e na ergonomia.

A utilização do termo Realidade Virtual de uma forma aberta ocorreu a partir das indústrias de filmes, em especial os filmes de ficção científica, onde eram utilizados na projeção de pessoas, informações, mapas, etc., que interagiam com algum personagem, essas projeções eram conhecidas por hologramas.

2.2 REALIDADE AUMENTADA

A Realidade Aumentada trata-se de um enriquecimento de elementos virtuais (pode não ser necessariamente tridimensionais) a outro ambiente, neste caso o ambiente físico.

RA pode ser definida de várias formas, na definição de KIRNER e SISCOOTTO demonstra algumas:

- Enriquecimento do ambiente real com objetos virtuais, usando algum dispositivo tecnológico, funcionando em tempo real;
- Melhoria do mundo real com textos, imagens e objetos virtuais, gerados por computador;
- Mistura de mundos reais e virtuais em algum ponto da realidade/virtualidade contínua, que conecta ambientes completamente físicos a ambientes completamente virtuais (KIRNER, SISCOOTTO, 2007, p.10)

Neste contexto, o usuário permanece no mundo real e ao contrario da realidade RV, a aplicação tem o intuito de transportar o ambiente virtual para o espaço do usuário, por meio de dispositivos tecnológicos, com o auxílio de câmeras, por exemplo.

A RA tenta combinar os recursos de multimídia com a RV para prover interação em tempo real, apresentando elementos de boa qualidade de imagem. Como o foco deste é manter o senso de presença do usuário no mundo real, tem uma necessidade de usar recursos tecnológicos invisíveis ao usuário o deixando livre no ambiente que está. Tais como recursos de rastreamento ótico, projeções, interações multimodais, etc.

TORI, KIRNER E SISCOOTTO (2006), apresentam os tipos e os componentes de um sistema de RA:

A realidade aumentada pode ser classificada de duas maneiras, dependendo da forma que o usuário vê o mundo misturado. Quando o usuário vê o mundo misturado apontando os olhos diretamente para as posições reais com cena óptica ou por vídeo, a realidade aumentada é de visão direta (imersiva). Quando o usuário vê o mundo misturado em algum dispositivo, como monitor ou projetor, não alinhado com as posições reais, a realidade aumentada é de visão indireta (não imersiva), [...].

Na visão direta, as imagens do mundo real podem ser vistas a olho nu ou trazidas, através de vídeo, enquanto os objetos virtuais gerados por computador podem ser projetados nos olhos, misturados ao vídeo do mundo real ou projetados ao cenário real. Na visão indireta, as imagens do mundo real e do mundo virtual são misturadas em vídeos e mostradas ao usuário. (TORI, KIRNER, SISCOOTTO; 2006 p. 27-28)

Na Figura 4, é apresentado a visão do Sistema de RA Não-Imersivo e na Figura 5 a Visão do Sistema da RA Imersivo.

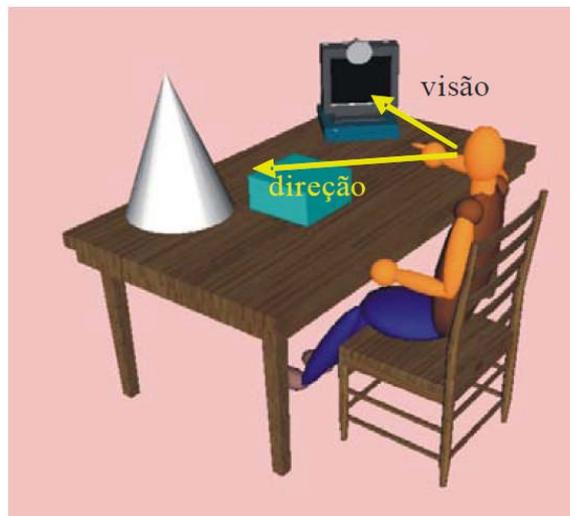


Figura 4 – Sistema de RA Não Imersivo

Fonte: TAVARES, 2009, p.29

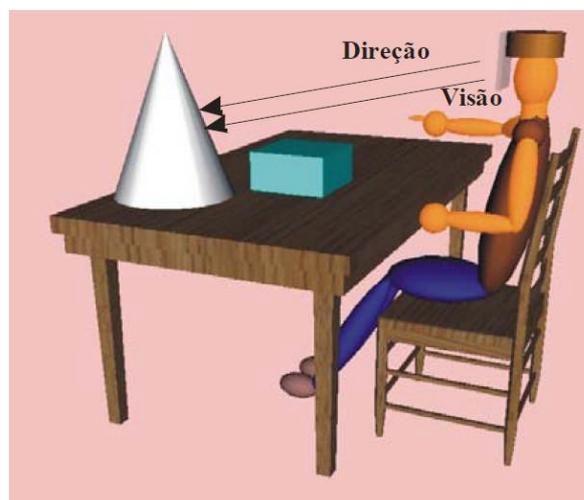


Figura 5 – Sistema de RA Imersivo

Fonte: TAVARES, 2009, p.29

Na visão de Milgran (1994) e Isdale (2000), o sistema de RA pode ser classificado com base na tecnologia de visualizadores em: visão óptica direta, visão direta baseada em vídeo; visão baseada em monitor e visão baseada em projetor.

Na composição básica da RA pode-se elencar o hardware e o software conforme TORI, KIRNER, SISCOOTTO (2006; p.28).

“O hardware envolve os dispositivos de entrada, displays, processadores e redes. O software inclui o software de realidade virtual, mais posicionadores e misturadores de imagens, funções de interação e interfaces multimodais”.

Os sistemas de RA no início enfatizavam a visualização, e não se preocupavam em como iria ocorrer a interação dos usuários com esses sistemas. Alguns desses sistemas limitaram-se apenas a reproduzir no ambiente a RA gráficos de interfaces já conhecidas no sistema 2D e 3D, como por exemplo menus de tela, reconhecimento de gestos, etc.

A interface equipada com os dispositivos engloba mecanismos de hardware e software especializados (drivers de dispositivos), que dão suporte para que ocorra as interações. A RA tem uma tendência a utilizar dispositivos que acabem passando despercebidos pelo usuário, com isso as ações ganham mais naturalidade.

Segundo Azuma (2001): Além de usar as interfaces gráficas, os sistemas de RA vêm apresentando duas tendências: explorar diferentes tipos de visualizadores e dispositivos; e integrar o mundo virtual com o mundo real, usando interfaces tangíveis.

Outro termo muito conhecido seria a Virtualidade Aumentada, esta se refere ao mundo virtual como vários elementos reais pré-cadastrados e por meio de dispositivos multissensoriais poderem manipular ou interagir no mundo virtual. Podem ser utilizados objetos estáticos, como móveis, edifícios, etc. ou até Avatares referentes a pessoas, mãos, animais, processados através de escaneamento 3D, este tipo de reconstrução real tem o intuito de ser realizada em tempo real para preservar os movimentos e sincronizar como o mundo real aonde está sendo executado a interação.

CAPÍTULO 3 ANÁLISE DE ESTÁGIOS, TIPOS E TÉCNICAS DE TESTES UTILIZADAS EM SISTEMAS DE REALIDADE VIRTUAL

Neste capítulo serão apresentados os principais trabalhos analisados na área de RV e no próximo capítulo na área de RA. Foram analisados dezenove artigos e monografias retirados entre periódicos de universidades e nos WorkShop de Realidade Virtual e Aumentada dos anos de 2010, 2013 e 2014 (WRVA e SRV), além destes inúmeros trabalhos que foram lidos e analisados, somente os que apresentavam algum fundamento para análise e possuíam tipos de testes relacionados ao tema foram citados detalhadamente.

Ao realizar a primeira leitura de cada artigo ou monografia, foi identificado se o assunto seria pertinente a este trabalho de análise. Assim, realizou-se um estudo do que se tratava cada um e resumidamente será apresentado em cada subcapítulo as informações referentes a eles.

Foram considerados como requisitos para os trabalhos analisados, sistemas que possuíam detalhes de implementações, de informações quanto a utilização da ferramenta ou referente aos dispositivos (hardwares), que se fizeram necessários para obter um desempenho em tempo real da aplicação.

3.1 NAVEGAÇÃO, RECONHECIMENTO DE GESTOS E CONTROLE DE INTERFACE NO SISTEMA ITV UTILIZANDO O DISPOSITIVO KINECT

Pesquisa analisada: Navegação, Reconhecimento de gestos e controle de interface no sistema ITV utilizando o dispositivo Kinect, de: CARVALHO (2013).

Esta dissertação apresenta a implementação de navegação no ambiente virtual, reconhecimento de gestos e controle de interface, feitos através do dispositivo Kinect, Utilizando o Sistema ITV: (Instrução Técnica Virtual) um sistema de treinamento de operadores e mantenedores de usinas hidrelétricas e subestações elétricas. São mostrados, também, determinados aperfeiçoamentos recentes, como conversão em vídeo, telas de alarmes sonoros e visuais, ambientação sonora em três dimensões e narração do processo.

Além da apresentação do Sistema ITV, é exposto o dispositivo Kinect e o algoritmo utilizado na comparação dos padrões de movimento, o DTW. Como estudo de caso, é exibida uma ITV que foi elaborada especialmente para testar e avaliar a nova interface proposta. O Sistema ITV é utilizado para treinamento de operadores e mantenedores de usinas hidrelétricas e subestações elétricas em RV, combinado com o Kinect é possível o reconhecimento de gestos e controle da interface utilizando a comparação de movimento com o algoritmo DTW.

No trabalho são definidos os requisitos do sistema, definindo os métodos de entradas, os dados são utilizados fazendo uma comparação para gerar uma tabela com as informações obtidas pelo algoritmo DTW que lê os movimentos do dispositivo Kinect. Com estas informações é possível refinar os dados de entrada que serão utilizados aplicação.

No trabalho também é apresentada parte da implementação, detalhando que foi utilizado o teste unitário, conforme a informação que o mesmo passa em relação ao clique do mouse. Em seguida é mostrado o funcionamento da aplicação combinado com o Kinect, ele defini no artigo o diagrama de atividades. Os testes realizados foram em funcionamento e também de aceitação pelos usuários, onde através do dispositivo Kinect foi possível simular a utilização de ferramentas no ambiente de realidade implementado.

Pode-se concluir que CARVALHO (2013), utilizou tanto testes estruturais (caixa branca) utilizando os testes unitários, e também o teste funcional de sistema (caixa preta), com o teste de sistema e de aceitação Alfa que testado no próprio ambiente, a seguir (figura 6) está o diagrama de classe da aplicação, ele também apresentou a indivíduos que não tinham conhecimento do funcionamento da ferramenta para realizarem o teste de aceitação Beta.

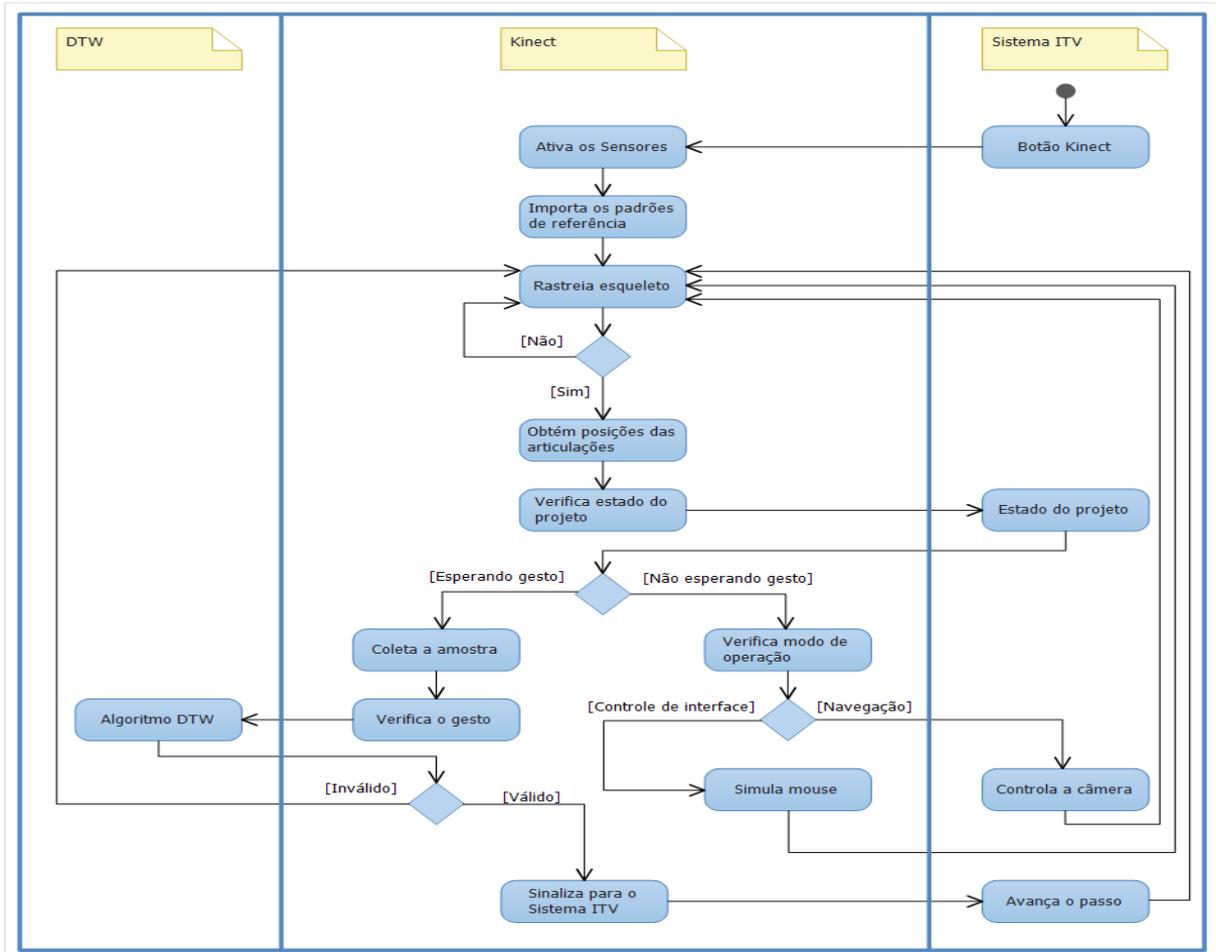


Figura 6 – Diagrama de Classes

Fonte: CARVALHO (2013, pag. 61)

No teste de aceitação foi verificado também se algum usuário apresentava familiaridade com o Sistema ITV e quais o desconheciam. Gerando assim o gráfico a seguir, podendo perceber que mais da metade dos entrevistados (53.85%) não possuía familiaridade alguma com o sistema (figura 7).

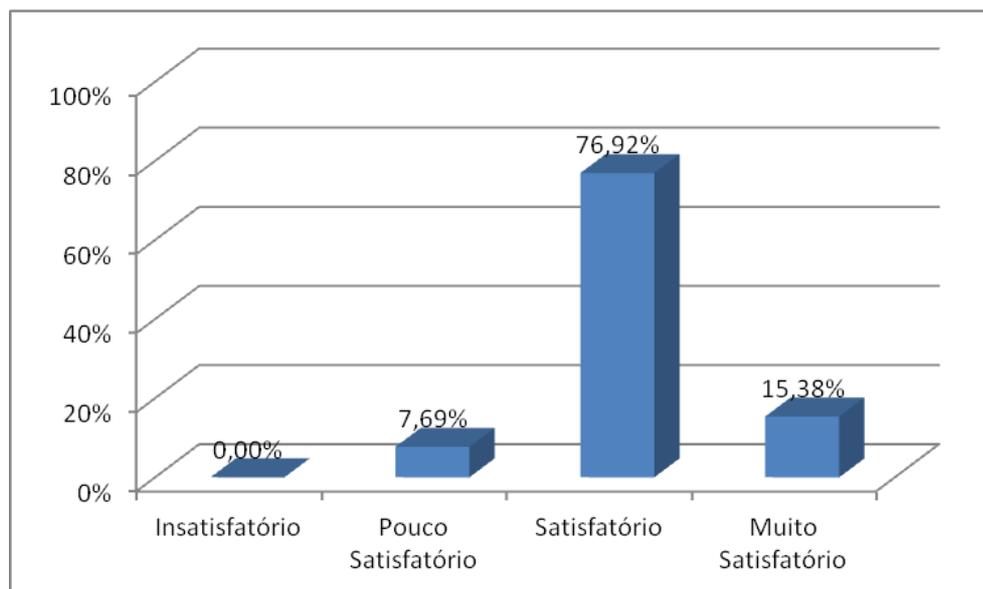


Figura 7 – Classificação de aceitação dos usuários pelo software

Fonte: CARVALHO (2013, p.72)

3.2 MODELAGEM DE OBJETOS TRIDIMENSIONAIS PARA UM AMBIENTE INTERATIVO DE INSTRUÇÕES E TÉCNICAS VIRTUAIS

Pesquisa analisada: Modelagem de objetos tridimensionais para um ambiente interativo de instruções técnicas virtuais, de: NASCIMENTO (2010).

Este trabalho utiliza também o sistema ITV (Instruções Técnicas Virtuais), em si este trabalho não entra no contexto de implementação, as aplicações 3D são voltadas ao treinamento e simulação industrial pelo ITV, ele é um ambiente de criação e execução de animações virtuais 3D interativas, que representam instruções de manutenção (montagem/desmontagem) e operação de maquinário da Usina Hidrelétrica de Tucuruí. Este trabalho faz uma abordagem sobre a metodologia utilizada na implementação do sistema ITV, com ênfase à criação dos objetos 3D, considerando requisitos, objetivos e limitações do processo.

Esse trabalho aborda somente os conceitos da modelagem tridimensional, entrando somente nos quesitos funcionais/aplicação. Detalham as formas geométricas, texturas e como são criados esses objetos utilizando diversas ferramentas de aplicações de modelagem 3D. Também compara as linguagens de desenvolvimento em 3D, porém neste não é tratado de implementação ou código, apenas das características de cada ferramenta.

3.3 MODELAGEM, SIMULAÇÃO, E VISUALIZAÇÃO IMERSIVA DE REDES SEM FIO

Pesquisa analisada: Modelagem, simulação, e visualização imersiva de redes sem fio, de: BENDZ (2008).

Esta pesquisa aborda o uso com visualização imersiva em campos eletromagnéticos, especificamente em campos gerados por redes sem fio, como as redes IEEE 802.11 (Wi-Fi), que são muito utilizadas no cotidiano. Uma versão aprimorada do método de diferenças finitas no domínio do tempo (FDTD) é desenvolvido: o método FDTD de alta ordem e malha grosseira (Coarse Grid Higher Order FDTD, CGHO-FDTD), que realiza os cálculos numéricos para a representação, estes podem ser ainda mais rápidos pelo uso de computação paralela, em um aglomerados de computadores. Alguns fenômenos de tempo são possíveis serem criados através do ambiente como a difração, reflexão, e atenuação.

O objetivo é pesquisar e desenvolver técnicas de visualização imersiva baseadas nos princípios básicos da física para a melhoria do entendimento, por meios da modelagem, simulação, e visualização, da propagação de ondas eletromagnéticas em geral, um exemplo da aplicação esta nas figuras 8 e 9.

BENDZ (2008), descreveu que na sua implementação realizou diversos testes de integração durante o desenvolvimento do ambiente imersivo em RV, utilizando os testes de "caixa branca", pode-se validar que o mesmo dedicou um capítulo da sua monografia para falar do teste funcional, realizando um teste de sistema real e simulando um cenário de uma residência, foi apresentando os valores e resultados obtidos (pag. 82), são detalhados os equipamentos utilizados para medir o desempenho.

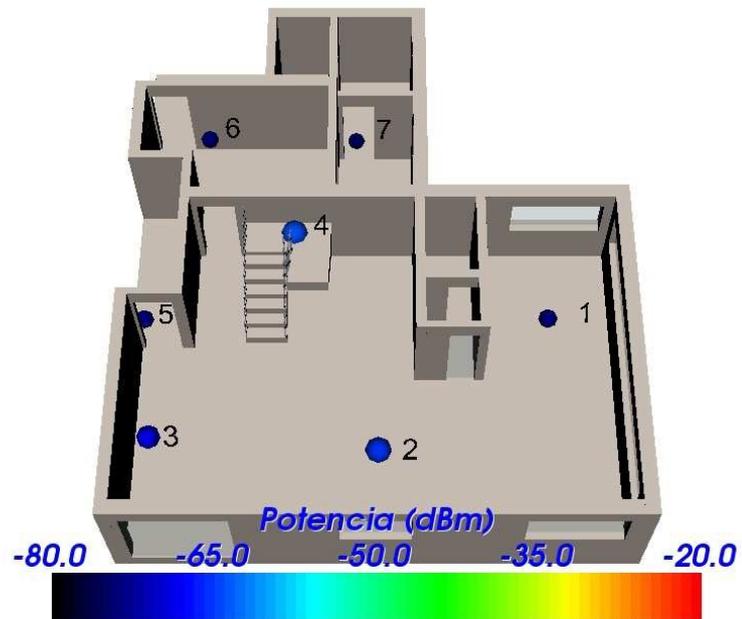


Figura 8 – As localizações dos pontos da medida

Fonte: BENDZ (2008, p.84)

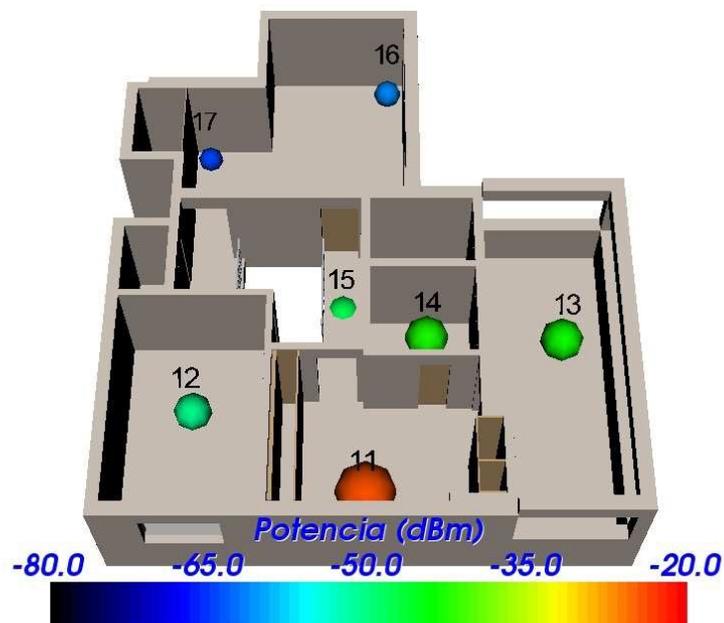


Figura 9 – Os valores médios usando o OLPC. Primeiro andar à esq. e segundo andar à dir.

Fonte: BENDZ (2008, p.84)

3.4 SWIMAX: UM SIMULADOR EM REALIDADE VIRTUAL PARA AUXILIAR O ENSINO DO PADRÃO WIMAX

Pesquisa analisada: SwImax: Um simulador em Realidade Virtual para auxiliar o ensino do padrão WiMAX, de: NEGRÃO (2012).

O padrão IEEE 802.16 chamado de WiMAX, tecnologia de rede banda larga sem fio para áreas metropolitanas, utilizado como alternativa para transmissão de sinal de Internet em regiões que não possuem infraestrutura de rede cabeada.

Por conta do ensino desta tecnologia nas salas de aulas serem meramente teórico, o que dificulta a compreensão dos alunos com relação a determinadas funcionalidades do WiMAX, este trabalho vem abordar um projeto de desenvolvimento de um simulador em RV, SwImax também será voltado para auxiliar no ensino deste.

Esta aplicação simula algumas características do funcionamento deste padrão IEEE 802.16, as quais são: faixas de frequência de operação, área de cobertura, procedimento de handover, transmissão sem linha de visada, entre outros (figura 11).

Para implementação foi utilizado as ferramentas 3Ds Max (3D Studio) para modelagem 3D, Adobe Photoshop CS5 para manipulação das imagens através de camadas, UNITY 3D que seria uma *game engine* (motor de jogo) *software* que visa facilitar e abstrair o desenvolvimento de *games* e aplicações em RV, através de recursos criados para aumentar a eficiência de determinadas tarefas, como o carregamento dos modelos 3D, a inclusão de animações, de imagens, sons e etc.

Foram utilizados também as tecnologias Asset, Prefabs que tratam-se de elementos criados com a finalidade de reutilização, normalmente contendo *scripts* de lógica para o funcionamento de determinado conteúdo, foi apresentado também a portabilidade para acesso a aplicação via WEB e a arquitetura do projeto logo a seguir (figura 10).

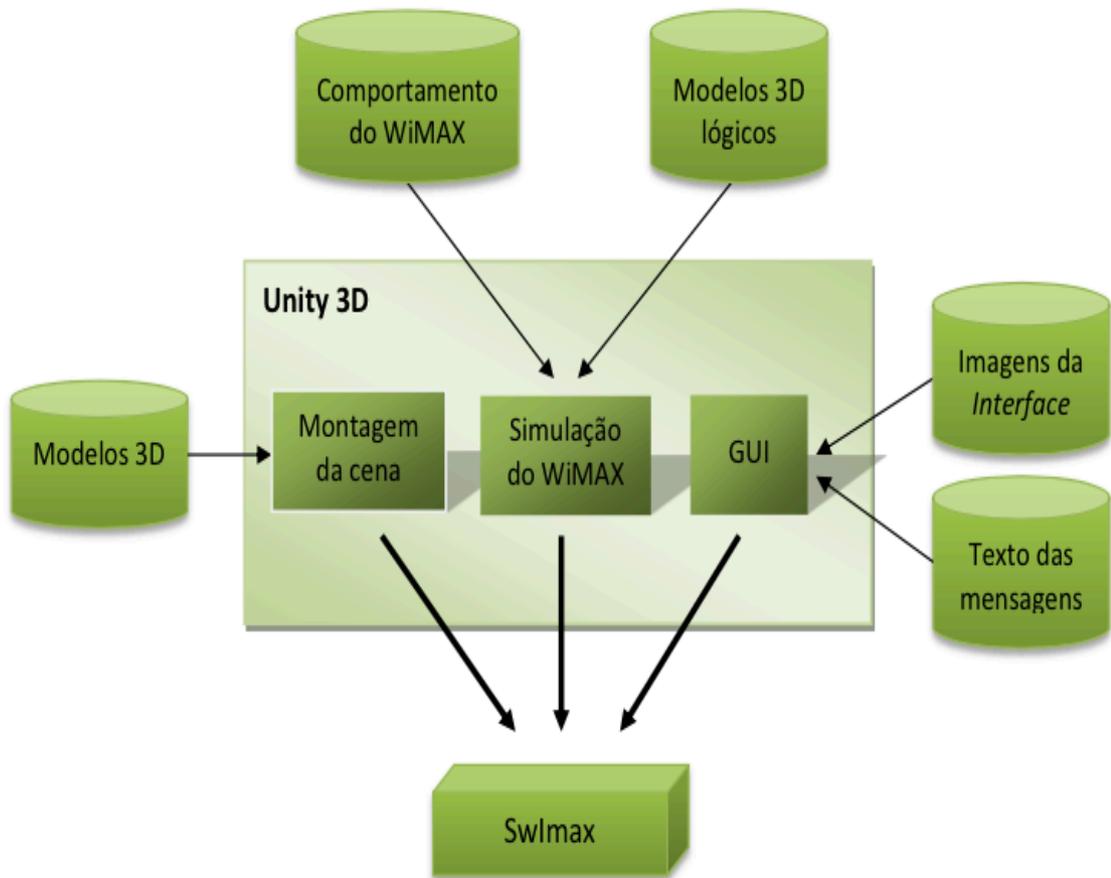


Figura 10 – Arquitetura do SwImax

Fonte: NEGRÃO (2012, p.46)



Figura 11 – Visão geral da interface do Swlmax

Fonte: NEGRÃO (2012, p.52)

Foram implementadas várias características, porém, não entra em detalhes dos testes realizados na implementação. Ele detalha a forma de avaliação do software, onde realizou testes de aceitação beta, que após os usuários testarem a aplicação, responderam um questionário para que fosse avaliada a ferramenta implementada, em seguida gerou-se diversos gráficos com os resultados obtidos, apresentado a seguir (figura 12), em relação a relevância da aplicação onde 70 % disse que a mesma era adequada e 30 % bastante adequada.

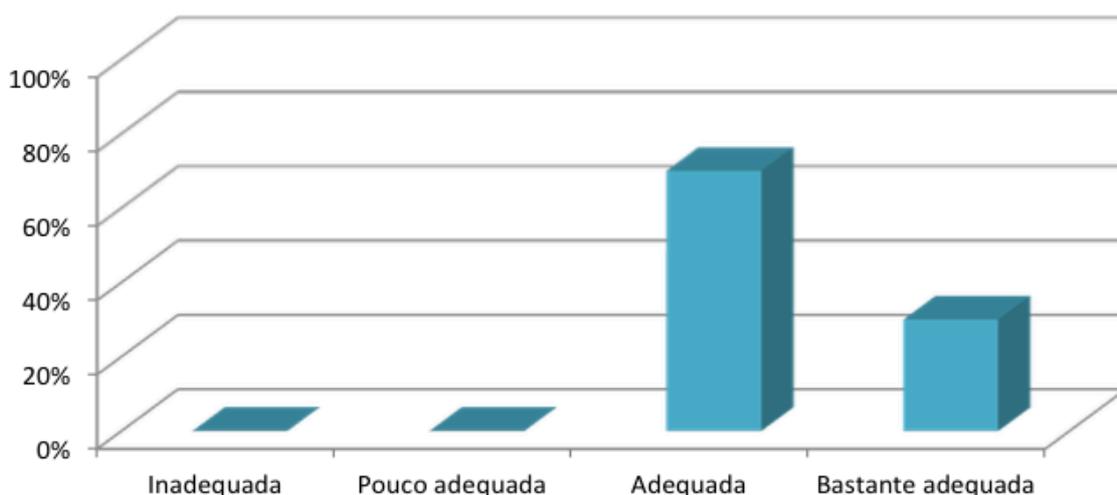


Figura 12 – Relevância do uso do simulador SwImax como ferramenta de auxílio ao ensino de redes sem fio

Fonte: NEGRÃO (2012, p.62)

Apresenta os resultados em seguida, informando que a ferramenta é capaz de confrontar para o aluno, conceitos previamente adquiridos em sala de aula e proporcionar que realmente facilite o processo de ensino e aprendizagem.

Concluimos que os testes mais utilizados na aplicação foram os de aceitação Beta de acordo com as respostas dadas pelos usuários.

3.5 UM AMBIENTE PARA O DESENVOLVIMENTO DE APLICAÇÕES DE REALIDADE VIRTUAL BASEADAS EM AGLOMERADOS GRÁFICOS

Pesquisa analisada: Um ambiente para o desenvolvimento de aplicações de realidade virtual baseadas em aglomerados gráficos, de: GUIMARÃES, (2004).

Esta pesquisa envolve o desenvolvimento de aglomerados gráficos de computadores, bibliotecas de desenvolvimento e de estratégias de desenvolvimento de aplicações de RV. O ambiente desenvolvido é voltado principalmente para a construção de aplicações de multi-projeção, como as executadas em CAVERNAs™ Digitais(CAVETM - CAVE Automatic Virtual Environment) primeiro sistema de multi-projeção totalmente imersivo e interativo desenvolvido na América Latina. Para testar o ambiente criado, foram desenvolvidas diversas aplicações. A Glass, que é a biblioteca desenvolvida e foi apresentada neste projeto; e as estratégias para o desenvolvimento das aplicações.

No trabalho de PAIVA GUIMARÃES, (2004, pag. 73) vemos a realização dos testes, é possível constatar que realizou um teste estrutural utilizando o teste unitário para testar a classe Glass e após o teste integração descendente para testar a aplicação.

Ele cita a aplicação Perfect Cluster (pag. 39) que foi utilizada para a realização dos testes através dos aglomerados gráficos, onde é necessário utilizar bibliotecas de desenvolvimento que facilitem a criação e execução das aplicações. O sistema de interconexão é baseado em comutadores Gigabit Ethernet NetStructure 470F IntelTM disponibilizando 8 portas ópticas 1000 Base-SX. Os nós são interconectados através de placas adaptadoras de redes ópticas GigaBit Ethernet, assim foi possível constatar que o mesmo realizou teste de carga (testes de análise de desempenho).

3.6 UMA ARQUITETURA PARA INTEGRAÇÃO DE SISTEMAS HÁPTICOS E ENGINES DE JOGOS

Pesquisa analisada: Uma Arquitetura para Integração de Sistemas Hápticos e Engines de Jogos, de: RODRIGUES, MACHADO (WRVA, 2010).

Aqui é apresentada um planejamento e o desenvolvimento de uma arquitetura para integrar uma engine de jogos com sistemas hápticos, estes oferecem um conceito de interação de forma avançada, ampliando o nível de realismo e oferecendo uma maior imersão no ambiente de RV.

Como método de implementação o desenvolvedor adicionou todos os objetos que fazem parte da cena do jogo a uma classe denominada GameManager, e as posições. E definiu a classe Object todos os objetos que fariam parte da cena do jogo (figuras 13 e 14).

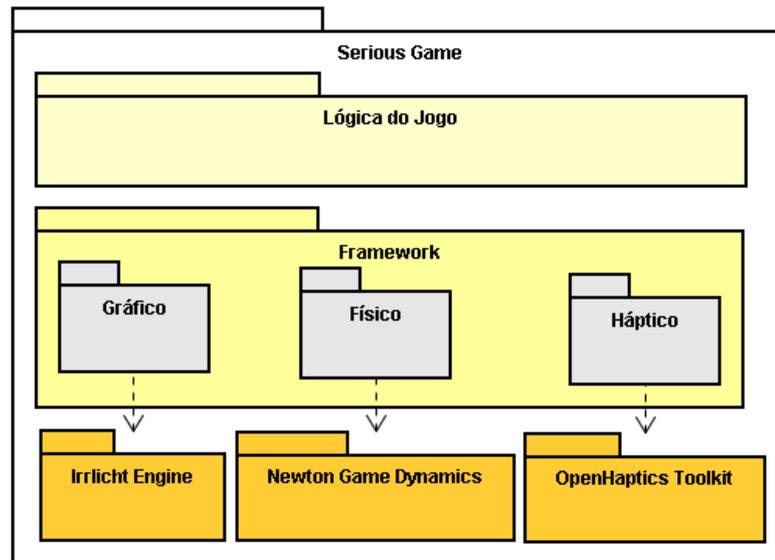


Figura 13 – Arquitetura proposta em alto nível

Fonte: RODRIGUES, MACHADO (WRVA, 2010, p.28)

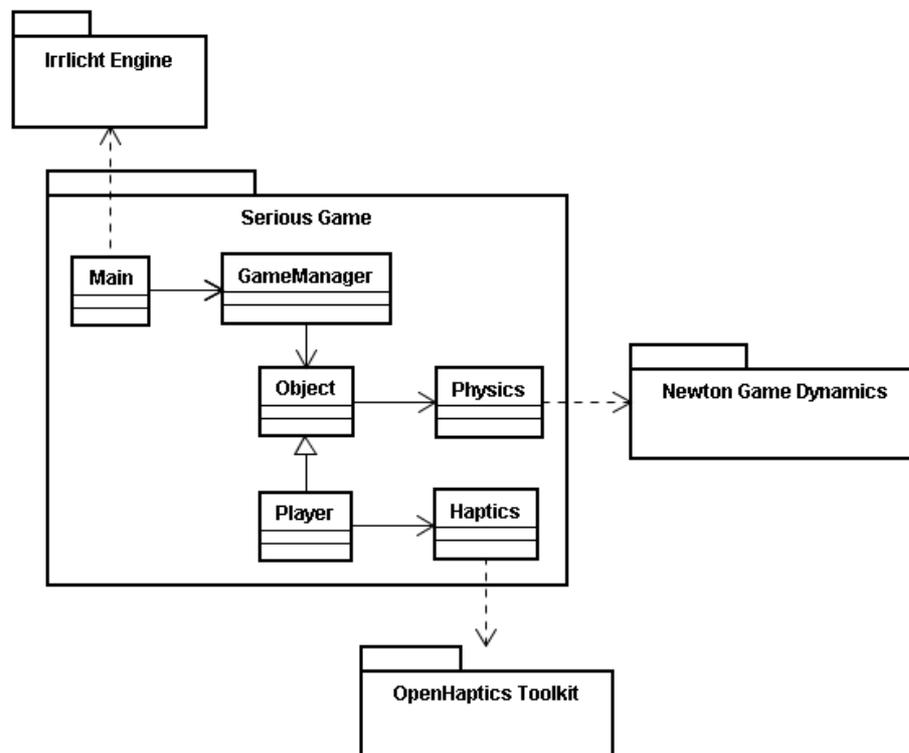


Figura 14 – Diagrama de classes do jogo

Fonte: RODRIGUES, MACHADO (WRVA, 2010, p.28)

O funcionamento do jogo é apresentado da seguinte forma:

"Para o jogo proposto foi criado um conjunto de classes. A classe Main inicializa a engine gráfica e possui o loop principal do jogo. A classe GameManager adiciona todos os objetos que farão parte da cena do jogo, como também suas posições. A classe Object define todas as variáveis e métodos necessários para a interação dos objetos na cena. Já as classes Physics e Haptics representam os módulos Físico e Háptico. A classe Player, uma das principais classes do jogo, representa o dispositivo háptico na cena do jogo." (RODRIGUES, MACHADO, WRVA, 2010, p. 31)

Conclui-se que tiveram problemas realizando o teste de desempenho, onde utilizou o Teste de Carga e de contenção, ele cita ainda que devido a alta necessidade de atualização, para estabilizar o laço de controle do dispositivo háptico e através da classe Player, foi possível calcular os retornos de força, com os dados geométricos fornecidos pelo Módulo Físico, sendo assim identificamos também que ele utilizou testes unitários e de integração.

3.7 MUSEU 3I: PUBLICAÇÃO E VISITAÇÃO ONLINE DE ACERVOS TRIDIMENSIONAIS

Pesquisa analisada: Museu 3I: Publicação e Visitação Online de Acervos Tridimensionais, de: FALCÃO, MACHADO (WRVA, 2010).

O projeto realizado por FALCÃO, MACHADO (WRVA, 2010), é um museu virtual, onde o usuário poderá escolher as obras que deseja visualizar enquanto navega por este museu tridimensional, outro fator é que se o visitante possuir um objeto no formato X3D, este poderá enviar ao Curador para ser exposta como base de acervos. Os Museus virtuais 3D são ambientes virtuais que buscam representar tridimensionalmente museus reais ou imaginários (figura 15).

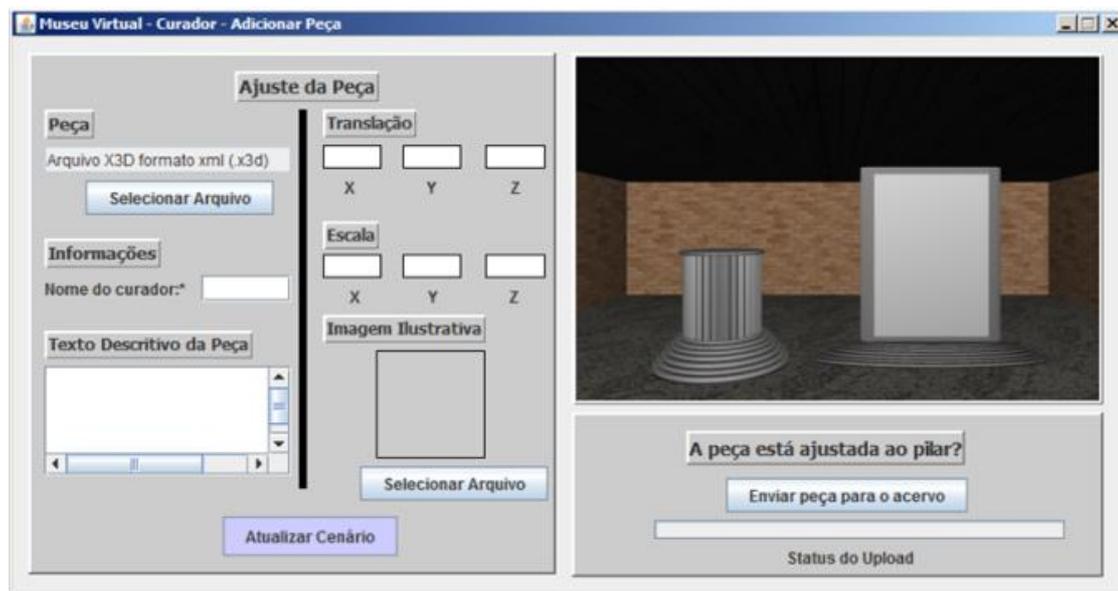


Figura 15 – Interface gráfica da Aplicação – Curador do Museu.

Fonte: FALCÃO, MACHADO (WRVA, 2010, p.80)

Primeiramente ele realizou a modelagem tridimensional do museu utilizando Blender (software livre de modelagem 3D) e foi considerado também a qualidade, para não ser excessiva por se tratar de uma aplicação WEB, após realizou o desenvolvimento da aplicação e interface gráfica para a ser utilizado pelo visitante. Utilizaram Java em conjunto com a API SAI proveniente do browser Xj3D para instanciar a visualização e incorporar a interface gráfica desenvolvida com a API Swing.

Na questão da implementação em relação ao perfil Curador da aplicação, somente a opção adicionar peça é testada funcionalmente como mostrado no artigo, porém o autor não entra em detalhes dos tipos de testes utilizados, apenas demonstra com algumas imagens como deve ser o ambiente tridimensional, esta aplicação ainda passaria por implementações antes de ser disponibilizada em uma applet na Web e após os usuários poderão interagir com o mesmo e realizar os testes de aceitação do projeto.

3.8 X3D E INTEGRAÇÃO MULTIMÍDIA PARA REPRESENTAÇÃO DE UM SÍTIO ARQUEOLÓGICO

Pesquisa analisada: X3D e Integração Multimídia para Representação de um Sítio Arqueológico, de: MEDEIROS, MACHADO (WRVA, 2010)

O Sítio Arqueológico Itacoatiara do Ingá é um dos mais importantes sítios arqueológicos do país e do mundo, está localizado no município de Ingá, no interior da

Paraíba, sendo tombado pela União através do Instituto do Patrimônio Histórico e Artístico Nacional - IPHAN. Este trabalho tem o intuito de preservar o patrimônio histórico e cultural do sítio arqueológico através do ambiente virtual. Foi proposta também a inclusão de informações importantes a respeito do mesmo através de diferentes mídias, como textos, imagens e vídeos. Assim o ambiente virtual passa a representar não apenas o ambiente real, mas adiciona a estes materiais complementares com informações relacionadas a tradição popular, arqueologia e geografia do local.

Pela necessidade de disponibilização do conteúdo na Web foi escolhida o padrão X3D para ser utilizado na aplicação, ele é utilizado para construir ambientes virtuais tridimensionais complexos. As formas são descritas por figuras geométricas e os comportamentos das cenas podem ser controlados internamente pelo arquivo X3D e externamente por linguagens de programação ou *script*.

Primeiramente foi feito uma modelagem do cenário (figura 16), inclusive do relevo local do terreno, apos montar o cenário foi implementado a navegação do cenário utilizando o modo GAME do X3D (FPS - *First Person Shooter*) oferecendo movimentação ao usuário em primeira pessoa.

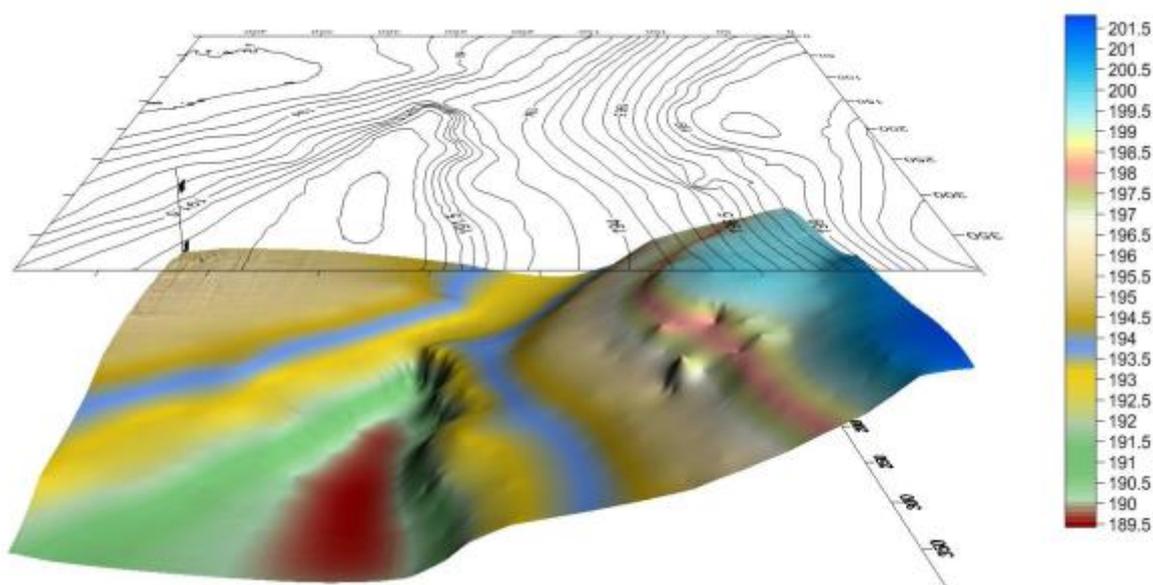


Figura 16 – Modelo numérico de terreno e curvas de altitude

Fonte: MEDEIROS, MACHADO (WRVA, 2010, pag. 66)

Os autores não entram em detalhes da implementação e programação, somente da montagem do cenário através da ferramenta X3D, e apresentam informações referentes ao browser escolhido para executar a aplicação, *browser* Xj3D perante a necessidade de processamento e por conta que seria disponibilizado o acesso via WEB, detalham a criação de um “nó” contendo as coordenadas de textura que seria feito em cada imagem o MultiTexture (figura 17).

```
<MultiTexture mode='\"DOTPRODUCT3\" \"MODULATE\" '>
    <ImageTexture DEF='\"IM_Tex\" url='\"tex/painelinteiro03.png\" />
    <ImageTexture DEF='\"IM_Tex_003\" url='\"tex/dsc02305.jpg\" />
</MultiTexture>
```

Figura 17 – Representação do Nó Multitexture

Fonte: MEDEIROS, MACHADO (WRVA, 2010, pag. 66)

Como no máximo foi apresentado informações sobre o nó, pode-se classificar que somente foi apresentado informações referentes ao teste de caixa branca (unitário), já que não informam também se está disponível a aplicação para utilização e aceitação de usuários.

3.9 APLICAÇÕES BASEADAS EM GRAFO DE CENA – UMA ABORDAGEM ESTRUTURAL PARA CRITÉRIOS DE TESTE

Pesquisa analisada: Aplicações baseadas em Grafo de Cena – uma abordagem estrutural para critérios de teste, de: BEZERRA, NUNES, DELAMARO (USP).

Este trabalho apresenta uma aplicação denominada VETesting que é utilizada para realizar testes estruturais em aplicações com RV e RA através de Grafos de Cena GC sendo que este entra como um teste estrutural unitário e de integração, texturas, objeto, forma, iluminação são alguns dos aspectos que podem ser acionados aos GC acíclico.

Nestes grafos podem ser encontrados os nós intermediários e nós folhas que contém a descrição geométrica do objeto, já os nós intermediários representam as transformações tridimensionais como translação, rotação e escala. Com a análise do GC do Ambiente Virtual AV é possível derivar os requisitos de teste baseados nas características do ambiente criado (figura 18).

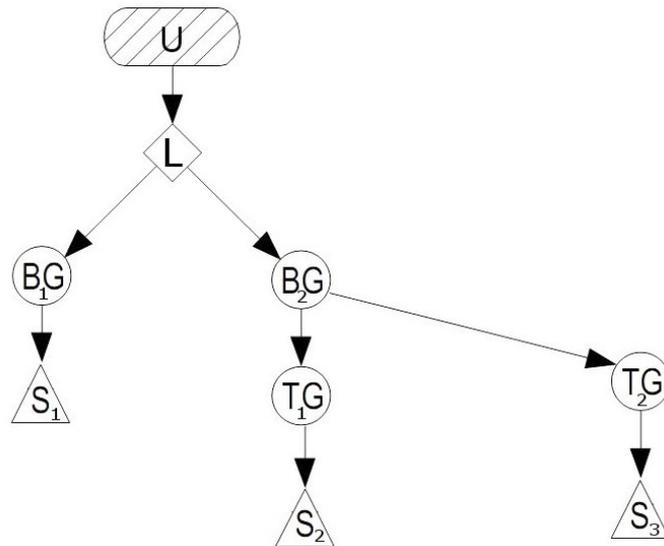


Figura 18 – Representação de um AV

Fonte: BEZERRA, NUNES, DELAMARO (USP).

Na Figura 19 a seguir é mostrado um exemplo de funcionamento da aplicação, através do diagrama que descreve a execução da ferramenta, que trata aplicações desenvolvidas utilizando a API Java3D, como descrito pelo autor. Inicialmente é carregado a classe (universo virtual) a ser testada por meio de um ClassLoader específico. Após ser carregada a classe, salvam-se todos os requisitos que serão utilizados no teste e a visualização do GC atual é apresentada em uma interface gráfica (figura 20).

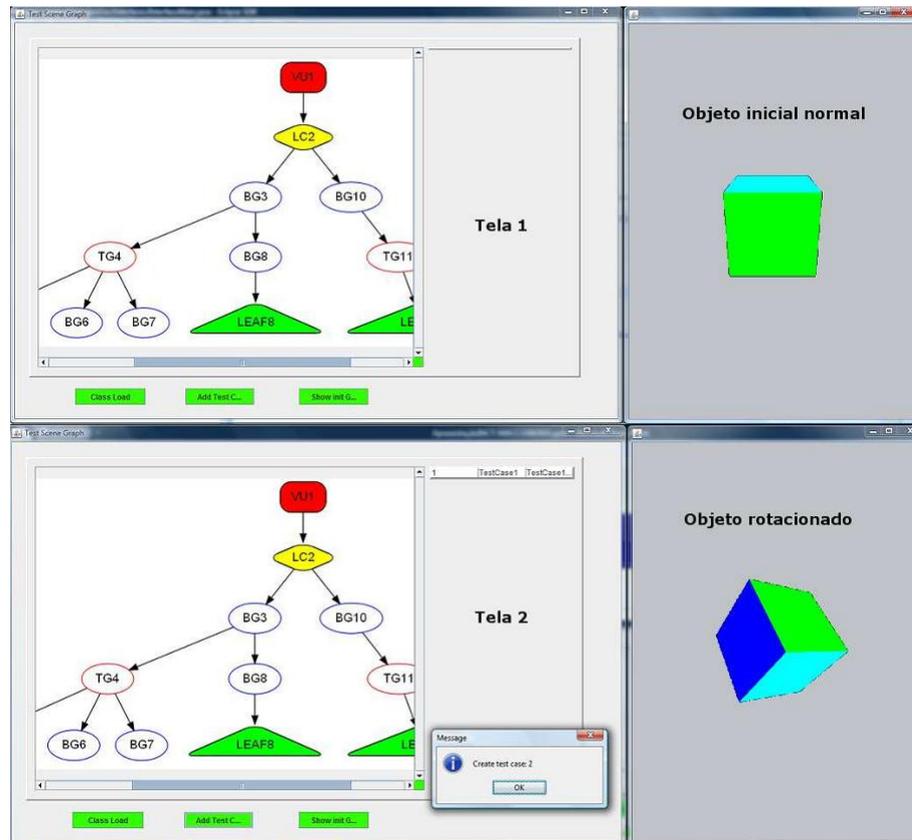


Figura 19 – Screenshots da ferramenta.

Fonte: BEZERRA, NUNES, DELAMARO (USP).

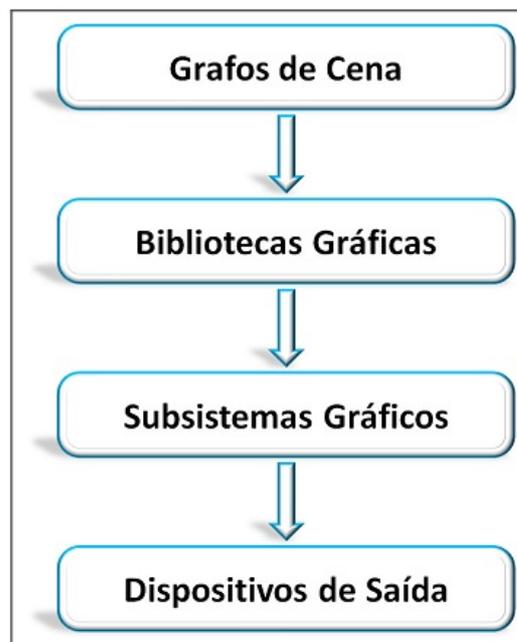


Figura 20 – Diagrama de execução da ferramenta

Fonte: BEZERRA, NUNES, DELAMARO (USP).

3.10 RECONHECIMENTO DE GESTOS COM SEGMENTAÇÃO DE IMAGENS DINÂMICAS APLICADAS A LIBRAS

Pesquisa analisada: Reconhecimento de Gestos com Segmentação de Imagens Dinâmicas Aplicadas a Libras, de: PAVAN, MODESTO (WRVA, 2010).

Este projeto de implementação tem como principal objetivo criar uma técnica de rastreamento do sinais de libras, utilizando dispositivos de baixo custo, como no caso de webcam, sendo mais acessível e tornando a técnica aplicável à diversas áreas.

Foi utilizado a biblioteca OpenCV (Open Computer Vision) da Intel para o desenvolvimento do algoritmo de reconhecimento baseados e, contornos e dos classificadores, onde oferece melhor desempenho em tempo real.

Esta biblioteca fornece por exemplo Algoritmos para detecção de bordas, captura de imagens da *webcam* e comparação com as características definidas nos classificadores. Como o PAVAN, MODESTO (WRVA, 2010) descreve:

"As etapas de reconhecimento consistem basicamente em capturar as imagens dinâmicas pela *webcam*, utilizar o classificador responsável por realizar o reconhecimento da mão fechada para definir a área de interesse, aplicar o algoritmo de *Sobel*, responsável por extrair o contorno da imagem e realizar a análise com os classificadores treinados para identificar os sinais de Libras. "PAVAN (WRVA, 2010)

Os classificadores citados em questão são os responsáveis por analisar as imagens após ser aplicado o algoritmo de Sobel. Porém, deverá ser realizado vários treinamentos funcionais das imagens a serem reconhecidas. Também vale ressaltar que "para realizar o treinamento do classificador é necessário reunir um conjunto de imagens contendo o objeto de interesse, em diversas condições de iluminação e em posições diferentes." denominado de amostras positivas.

O termo "cascata" que o autor cita é utilizado por conta do algoritmo de treinamento, que cria uma série de classificadores que são analisados em sequência, até que um padrão seja identificado. Como mostrado no exemplo a seguir (figura 21), a estrutura do código utilizado na implementação como teste estrutural, pode-se citar esta estrutura.

```

<?xml version="1.0"?>
<opencv_storage>
<haarcascade_frontalface_alt type_id="opencv-haar-classifier">
  <size>20 20</size>
  <stages>
    <_>
      <!-- stage 0 -->
        <_>
          <_>
            <!-- tree 0 -->
              <_>
                <!-- root node -->
                  <trees>
                    <_>
                      <_>3 7 14 4 -1.</_>
                      <_>3 9 14 2 2.</_><!-- tree 0 -->
                    <_>0<!-- root node -->
                  <feature>0.6374819874763489<rects>
                    >
                      <_>3 6 14 8 -1.</_>
                      <_>3 10 14 4 2.</_><
                    0><_>
                      <_>0 6 2 14 -1.</_>
                      <_>0 13 2 7 2.</_><
                    00.4165507853031158
                    >0.0767832621932030
                    0.4795796871185303<
                    0.8047714829444885></_></_>

```

Figura 21 – Exemplo de uma cascata de classificadores armazenada em um arquivo XML que é utilizada na detecção de objetos pelas funções “cvLoad” e “cvDetectHaarCascade”.

Fonte: PAVAN, MODESTO (WRVA, 2010, pag 98)

Para treinar a letra “C” foi necessário diversas amostras e configurações de parâmetros classificando neste caso como um teste de caixa branca, onde é necessário realizar um treinamento muitas vezes cansativo da aplicação (figura 22).

O sistema desenvolvido captura o centro das coordenadas do sinal realizado e passa as informações através de um arquivo XML, para sincronizar essas coordenadas com o aplicativo responsável por construir o objeto 3D.

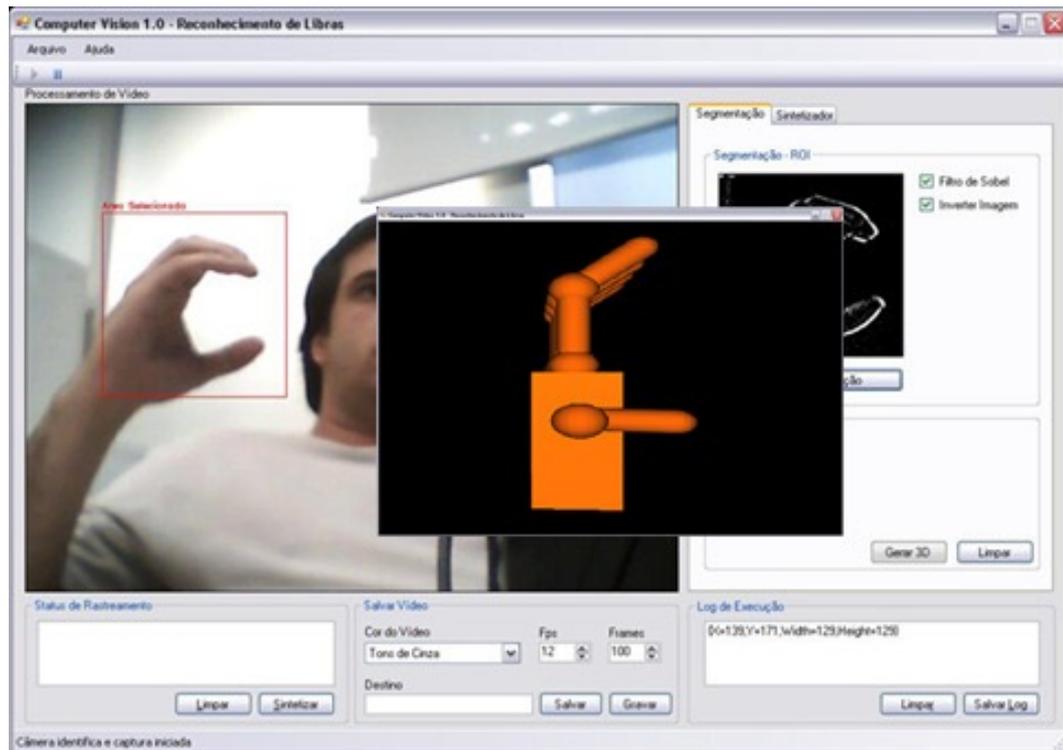


Figura 22 – Representação em 3D do sinal rastreado

Fonte: PAVAN, MODESTO (WRVA, 2010, pag 98)

Porém, não foram criados diversos classificadores, foi rastreado somente a letra C por ser um processo um tanto trabalhoso. Este trabalho em si realizou teste de caixa preta e pouca informação referente a implementação, entra somente no âmbito de rastreamento das formas e funções da aplicação porém, não mostra o sistema traduzindo e também não mostra se o mesmo será disponibilizado a usuários para realizar futuros testes de aceitação.

3.11 CONTROLE DE NAVEGAÇÃO EM AMBIENTES VIRTUAIS 3D ATRAVÉS DO RASTREAMENTO DE OBJETOS

Pesquisa analisada: Controle de Navegação em Ambientes Virtuais 3D através do Rastreamento de Objetos, de: GADELHA, SANTOS (WRVA, 2010)

Este artigo propõe uma aplicação para um deslocamento em ambientes virtuais, denominado Fingertrax, ele tem em vista apresentar um baixo custo, ser de fácil adaptação a computadores de uso pessoal, propiciar um desempenho na tarefa de deslocamento, pelo menos, tão eficiente quanto a utilização de mouse e teclado em ambientes virtuais de mesa e ser um mecanismo de utilização simples e fácil. Para avaliação da aplicação utilizaram o teste de aceitação beta ao aplicar a usuários e pontuar os resultados obtidos, nos testes foi realizado

também uma comparação com sistema de RV que utilizam tanto o teclado como o mouse para comparar o desempenho e as funcionalidades do mesmo.

Para este trabalho não é utilizado o uso de periféricos de entrada como Mouse e Teclado, é utilizado a webcam para ler o posicionamento da mão que indica a direção (figura 23 e 24), para isso foi necessário ser utilizado um anel de plástico para que fosse possível realizar a leitura pela webcam, como informado pelo autor e ainda calibrar o dispositivo para o reconhecimento do anel ou do objeto utilizado, como no exemplo mostrado na figura a seguir.

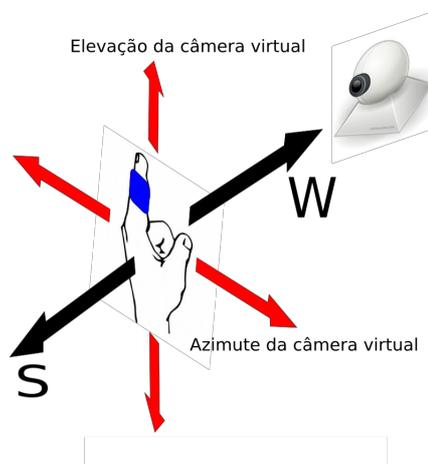


Figura 23 – Mapeamento dos movimentos do anel rastreado para o controle da câmera virtual

Fonte: GADELHA, SANTOS (WRVA, 2010, pag 104)

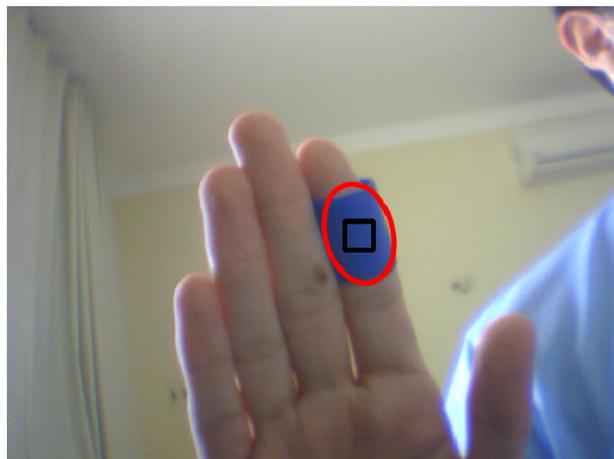


Figura 24 – Calibração do dispositivo de navegação

Fonte: GADELHA, SANTOS (WRVA, 2010, pag 104)

Para implementação da aplicação foi utilizado também a biblioteca OpenCV, após calibrar e identificar o objeto o sistema utiliza o algoritmo *Continuously Adaptive Mean Shift* (*CamShift*) [1], da OpenCV pela da função *cvCamShift* que foi originalmente projetado para rastrear faces humanas.

Para a implementação do ambiente de testes foi utilizado a linguagem C++ com a biblioteca OpenGL. Nele foi criado um ambiente com várias esferas vermelhas como alvo, que ao usuário atingi-las a cor muda para verde, foi implementado também um neblina para esconder os alvos na aplicação de teste.

O autor entra também nos detalhes quanto ao teste de desempenho realizado, aonde informa que os testes foram realizador em um computador com processador Intel Pentium 4 3.00 GHz, 3 GB de memória RAM, placa de vídeo NVIDIA GeForce 6200 TurboCache e webcam de 1,3 megapixels.

Já para o teste de aceitação beta, este foi aplicado em 23 pessoas na faixa etária de 18 à 25 anos, todos já haviam navegados em ambientes de RV com o mouse ou teclado em outras aplicações.

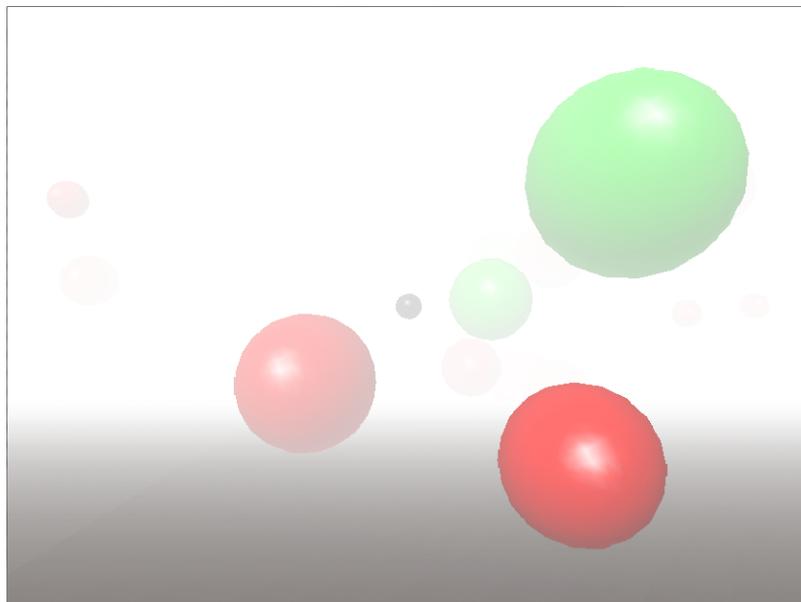


Figura 25 – Screenshot do ambiente de testes utilizado no experimento

Fonte: GADELHA, SANTOS (WRVA, 2010, pag 104)

Após os usuários realizarem um treinamento em relação a tarefa que deveriam executar (figura 25), ele dispunha de 2 minutos para locomover-se pelo cenário e acertar a

maior quantidade de alvos possíveis com a aplicação Fingertex. E em seguida ele deveria executar a mesma navegação utilizando o M&T (periféricos como mouse e teclados).

Por fim os resultados são apresentados (figura 26), neles é possível apurar que o maior desempenho se deu quando realizado o uso do mouse e teclado, do que quando utilizado a aplicação por meio da webcam. Apesar da aplicação não ter resultado positivo, vale salientar que os usuários não estavam acostumados a manipular objetos com a webcam e fica uma alternativa para a navegação de ambientes virtuais.

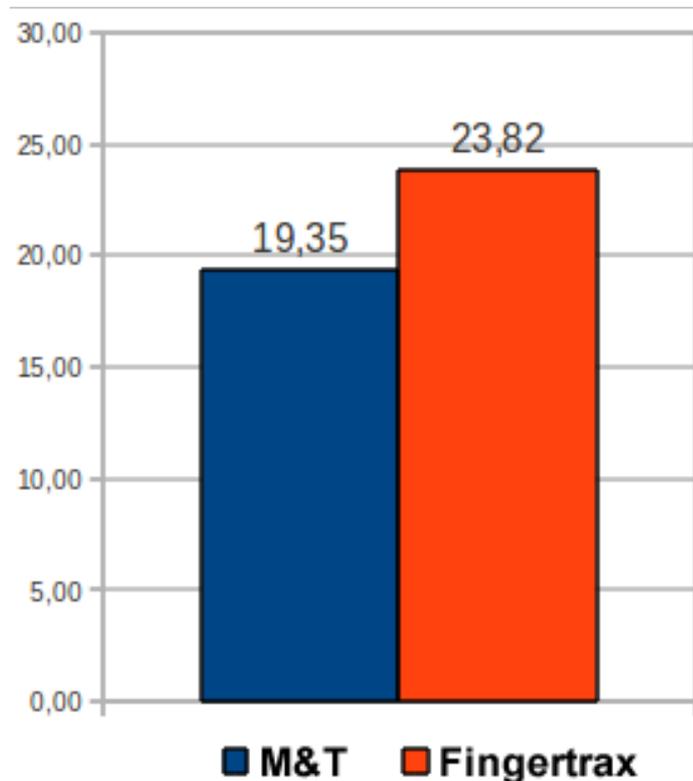


Figura 26 – Média de erros em cada dispositivo. M&T e Fingertrax.

Fonte: GADELHA, SANTOS (WRVA, 2010, pag 104)

3.12 VISUALIZAÇÃO DE AMBIENTES VIRTUAIS COM SIMULAÇÃO DE PROJEÇÃO HOLOGRÁFICA

Pesquisa analisada: Visualização de ambientes virtuais com simulação de projeção holográfica, de: BUCIOLI, LAMOUNIER (WRVA, 2014).

Este trabalho tem como objetivo principal simular em um protótipo projeções holográficas em dispositivos comuns como monitores e televisores, fornecendo uma plataforma para aplicações em RA e RV. Neste contexto, o trabalho visa imitar através de uma aplicação a forma como o ser humano enxerga objetos reais, onde que de acordo com o

ponto de vista, possa visualizar de diferentes ângulos a imagem projetada nos dispositivos assim como no mundo real.

Logo no início do projeto já descrevem uma problemática na implementação que seria em relação ao ângulo de visualização, onde para se aproximar da visão em tempo real precisariam de motores para mover a tv/monitor.

As informações em relação a arquitetura é descrita passo a passo sobre o desenvolvimento, desde a separação do usuário do ambiente real. E ainda descrevem parte do código gerado para realizar o procedimento de separação:

```
diferenca = abs(r1-r2)+abs(g1+g2)+abs(b1+b2);  
if(diferenca > limiar) pixeldif=0;  
else pixeldif=1;
```

Onde o valor de limiar serve para estabelecer uma tolerância da diferença entre os ambientes reais e virtuais. Este seria o módulo rastreador da aplicação.

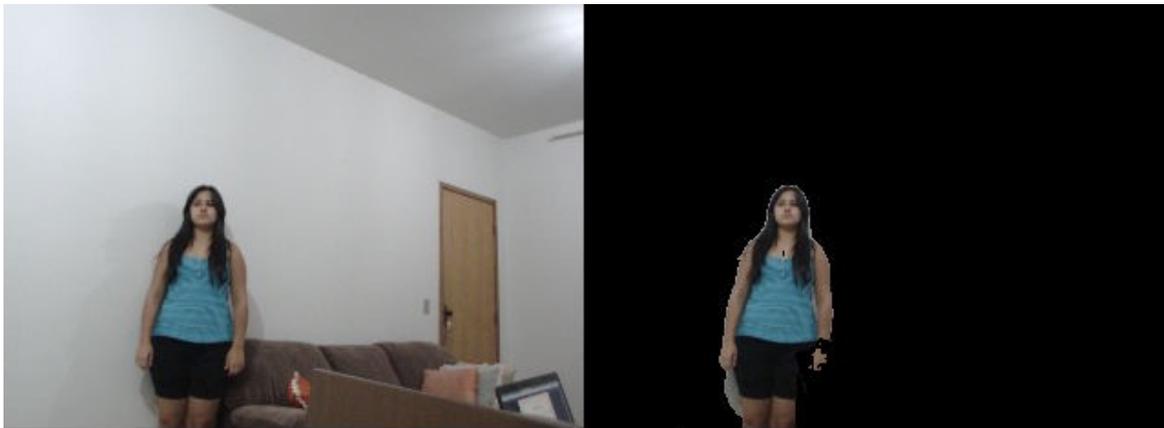


Figura 27 – Usuário sendo reconhecido pelo algoritmo de Diferença

Fonte: BUCIOLI, LAMOUNIER (WRVA, 2014, pag 144).

Entram em detalhes do cenário virtual desenvolvido (figura 27, 28, 29 e 30), onde conforme acionavam objetos, realizavam os testes e já apresentavam para documentação.



Figura 28 – Cena desenvolvida de exemplo da utilização do protótipo.

Fonte: BUCIOLI, LAMOUNIER (WRVA, 2014, pag 144).

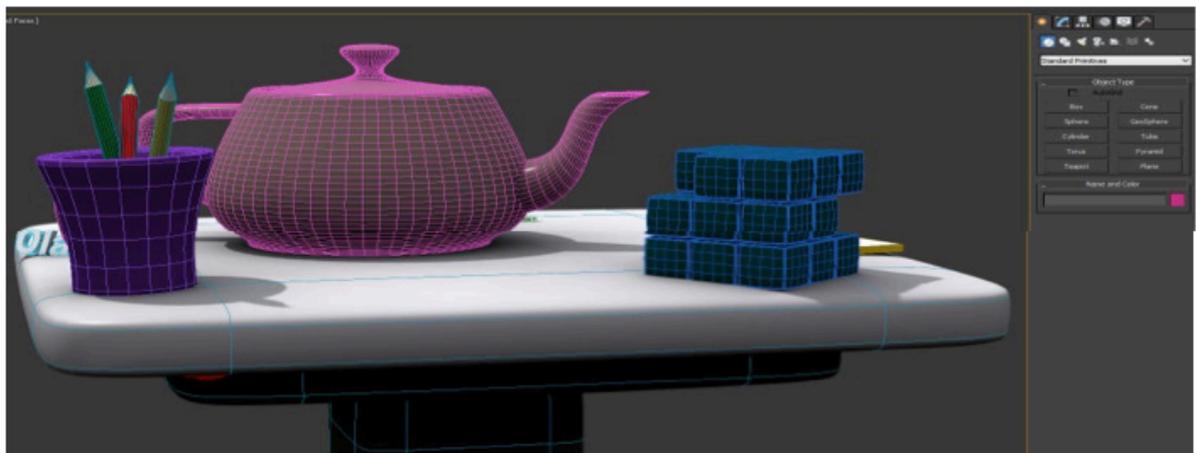


Figura 29 – Cena desenvolvida de exemplo da utilização do protótipo.

Fonte: BUCIOLI, LAMOUNIER (WRVA, 2014, pag 144).

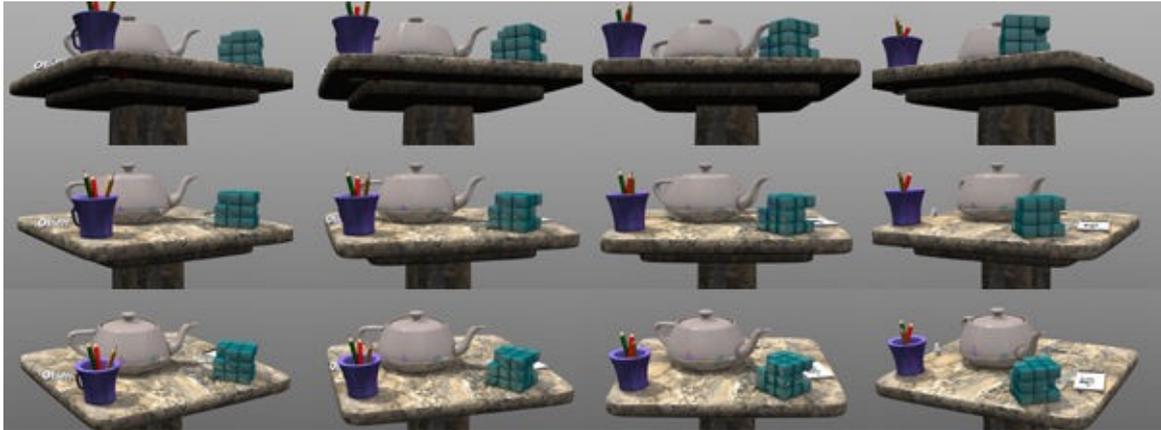


Figura 30 – Visão do usuário de acordo com o reconhecimento pelo algoritmo de Diferença.

Fonte: BUCIOLI, LAMOUNIER (WRVA, 2014, pag 144).

É realizado alguns testes com mudança de perspectiva quanto a visualização (figura 31), onde obtém êxito ao verificar que a aplicação se adaptava conforme a mudança de perspectiva, conforme o objetivo inicial.

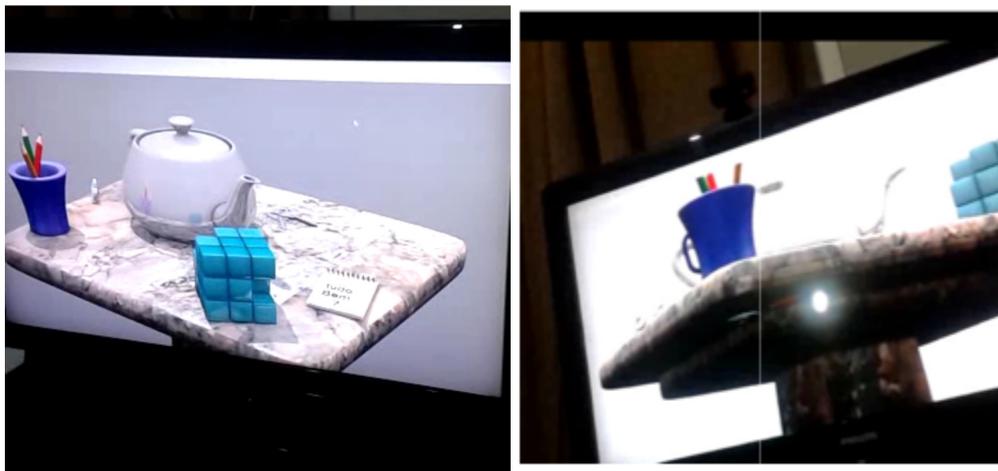


Figura 31 – Visão de diferentes ângulos pelo mesmo usuário

Fonte: BUCIOLI, LAMOUNIER (WRVA, 2014, pag 144).

Ao final eles ressaltam que o projeto ainda não foi finalizado, apresentando somente um protótipo da versão final a ser desenvolvida e que todos os testes foram realizados pela própria equipe. Caracterizando como testes estruturais, ou seja, de caixa branca.

CAPÍTULO 4 ANÁLISE DE ESTÁGIOS, TIPOS E TÉCNICAS DE TESTES UTILIZADAS EM SISTEMAS DE REALIDADE AUMENTADA

Neste capítulo serão apresentados os trabalhos que possuem especificação de testes na área de RA e respectivas informações referentes aos testes que foram utilizados em cada trabalho.

4.1 DESENVOLVIMENTO DE UM JOGO TRIDIMENSIONAL COM REALIDADE AUMENTADA

Pesquisa analisada: Desenvolvimento de um Jogo Tridimensional com Realidade Aumentada, de: SILVA, KIRNER (WRVA, 2010).

Este jogo desenvolvido foi batizado como Slidetrix. Possui o objetivo de testar a lógica e agilidade do participante. Ele possui uma base composta por vinte e quatro cubos espalhados sobre a superfície de numa matriz 3x3x3, como mostrado na Figura 02, vazada no eixo central vertical. Entre os cubos existe um espaço vazio. Esta característica é importante para permitir a mobilidade das peças no jogo. Sempre que uma peça for clicada, se houver um espaço vago em sua vizinhança ela poderá mover para esta nova posição (figura 32).

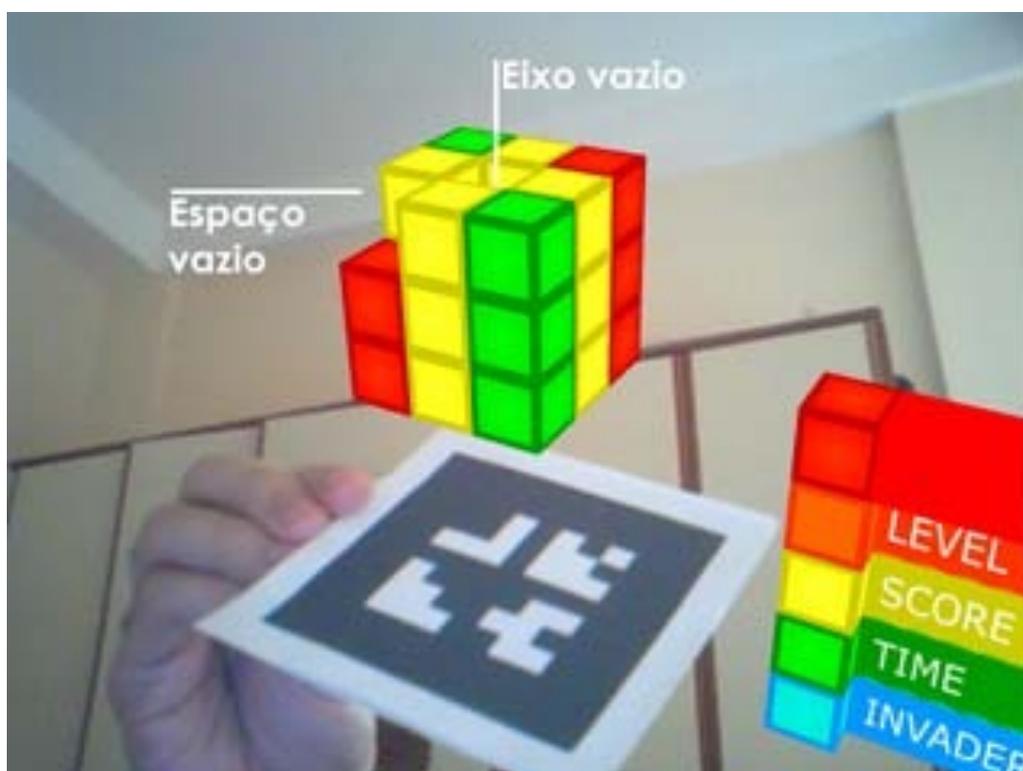


Figura 32 – Exemplo da execução de aplicação

Fonte: SILVA, KIRNER (WRVA, 2010, p.46)

Para desenvolvimento foi utilizado a ferramenta FlartoolKit que se trata de um projeto feito por Saqoosha adaptando o NyARToolkit, Utilizando Java para ActionScript 3.0 para criar o ambiente virtual.

O ambiente utilizando Actionscrip 3.0 com uma estrutura de programação orientada a objeto, foi importante para dar estabilidade, clareza e controle. Foi utilizado o teste de integração, onde o mesmo cita que um problema encontrado foi o não suporte a Threads pelo Flash. Em relação a testes de desempenho tiveram inconsistências com o problema de velocidade sem as Threads que não são suportados pelo Flash.

Durante os testes, ele informa que foi verificado queda de velocidade, na taxa de quadros por segundo, em momentos em que foram apresentados muitos objetos anexados ao cenário principal. Em máquinas mais antigas, a jogabilidade pode ser prejudicada por este motivo. E que como o flash não possui um tratamento nativo dos elementos básicos tridimensionais, esta queda de processamento foi constatada.

O desenvolvedor ainda cita que alguns fatores devem ser considerados para a aplicação executar bem, caracterizando os testes de carga e de avaliação de desempenho, tais como uma câmera com boa resolução, necessidade de uma máquina com níveis medianos de poder computacional. A lentidão de processamento na presença de muitos objetos virtuais e o baixo nível de tratamento de erros no ambiente de desenvolvimento do IDE Flash devem ser reduzidas, gradativamente, com o aprimoramento das tecnologias envolvidas. Porém, nenhum desses itens impediu a realização do projeto.

4.2 REALIDADE AUMENTADA PARA AUXILIAR O APRENDIZADO DE MOTOR ELÉTRICO

Pesquisa analisada: Realidade Aumentada para Auxiliar o Aprendizado de Motor Elétrico, de: COSTA, SILVA, CERQUEIRA (WRVA, 2010).

Neste artigo é apresentado como a RA pode ser utilizada para gerar uma ferramenta de auxílio ao aprendizado de motores de indução trifásicos (MIT), que são os mais utilizados industrialmente para acionar máquinas de qualquer potência, em razão de sua robustez, custos e, simplicidade operacional e de manutenção. Com a RA é possível salientar seu potencial de utilização e apresenta as vantagens e desvantagens desta tecnologia no processo de aprendizagem.

Para desenvolvimento da aplicação foi utilizado ARToolKit, porém esta ferramenta denominada SACRA foi desenvolvida por KIRNER (2011), Este trabalho é referente

reutilização da ferramenta criada por KIRNER, realizando customizações por COSTA, SILVA e CERQUEIRA (UFVJM, 2011), mas não entram em detalhes do código desenvolvido, somente nos testes funcionais da aplicação (figuras 33 e 34), não demonstrando nenhum teste de aceitação também. Apenas apresenta a aplicação e suas funcionalidades em RV e com RA caracterizando o teste de caixa preta de sistema.



Figura 33 – MIT com o motor virtual – RV

Fonte: COSTA, SILVA, CERQUEIRA (WRVA, 2010, pag. 54).

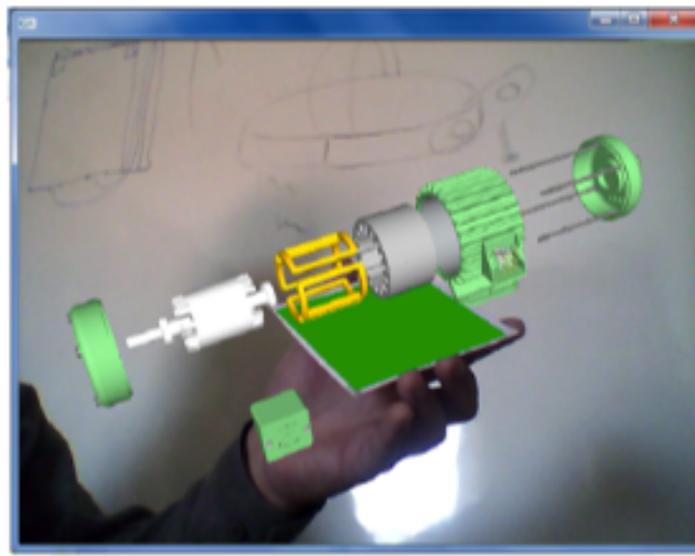


Figura 34 – Motor Expandido – RA

Fonte: COSTA, SILVA, CERQUEIRA (WRVA, 2010, pag. 54).

4.3 DESENVOLVIMENTO DE UMA ARQUITETURA PARA DISTRIBUIÇÃO DE REALIDADE AUMENTADA NA WEB APLICADA AO ENSINO DE MOTORES DE CORRENTE CONTINUA

Pesquisa analisada: Desenvolvimento de uma arquitetura para distribuição de realidade aumentada na web aplicada ao ensino de motores de corrente contínua, de: SILVA, SANTOS, entre outros (WRVA, 2010).

Um motor de corrente contínua converte a energia elétrica em energia mecânica, a diferença dos demais motores é que ele é alimentado com corrente de tensão contínua, como o próprio nome diz, esta pode ser fornecida por pilhas e baterias no caso de motores pequenos, ou de uma rede alternada após retificação no caso de motores maiores. Seus principais componentes são o estator e rotor. Este trabalho tem como objetivo mostrar uma arquitetura de distribuição de objetos com de RA na *web*, permitindo que vários usuários, conectados a um servidor, possam interagir com o mesmo objeto virtual que no caso específico é o motor de corrente contínua.

Para o desenvolvimento deste trabalho foi utilizado também o *FLARToolkit* que é um conjunto de classes desenvolvidas em AS3, que juntamente com o *Papervision3D* possibilita o desenvolvimento de aplicações de RA utilizando o Flash Player. Já para a comunicação cliente/servidor foi utilizado o BlazeDS que é um *middleware* responsável por estabelecer a comunicação entre ambos, utilizando para isso objetos remotos. A aplicação cliente pode ser desenvolvida em linguagem de programação Flex/Flash(*Action Script 3*) como neste caso ou poderia ser implementada por Ajax e o servidor foi desenvolvido em Java.

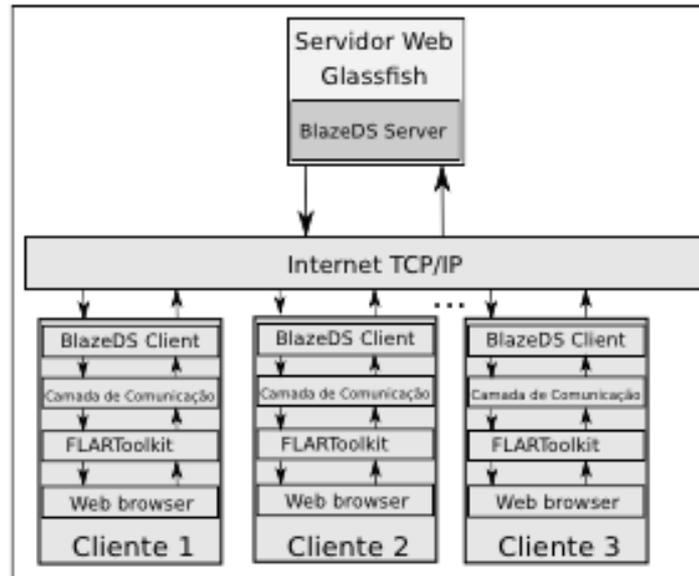


Figura 35 – Arquitetura do sistema proposto

Fonte: SILVA, SANTOS, entre outros (WRVA, 2010, p.54)

A arquitetura (figura 35) foi desenvolvida utilizando várias camadas de software, tanto no lado do cliente quanto do lado do servidor, com o propósito de facilitar a manutenção do software, o autor detalhou no item 6 do seu artigo os testes realizados.

Foram realizados testes de latência e escalabilidade (figura 36), onde ele testou primeiramente o tempo que o pacote enviado pelo cliente demora para chegar no servidor, no segundo teste utilizou o servidor com 3 clientes, assim foi possível medir o tempo de envio do pacote do cliente para o servidor e do servidor para outro cliente, por ultimo ele utilizou o servidor e 6 clientes, calculando o tempo de envio dos pacote também. Resultando no gráfico apresentado:

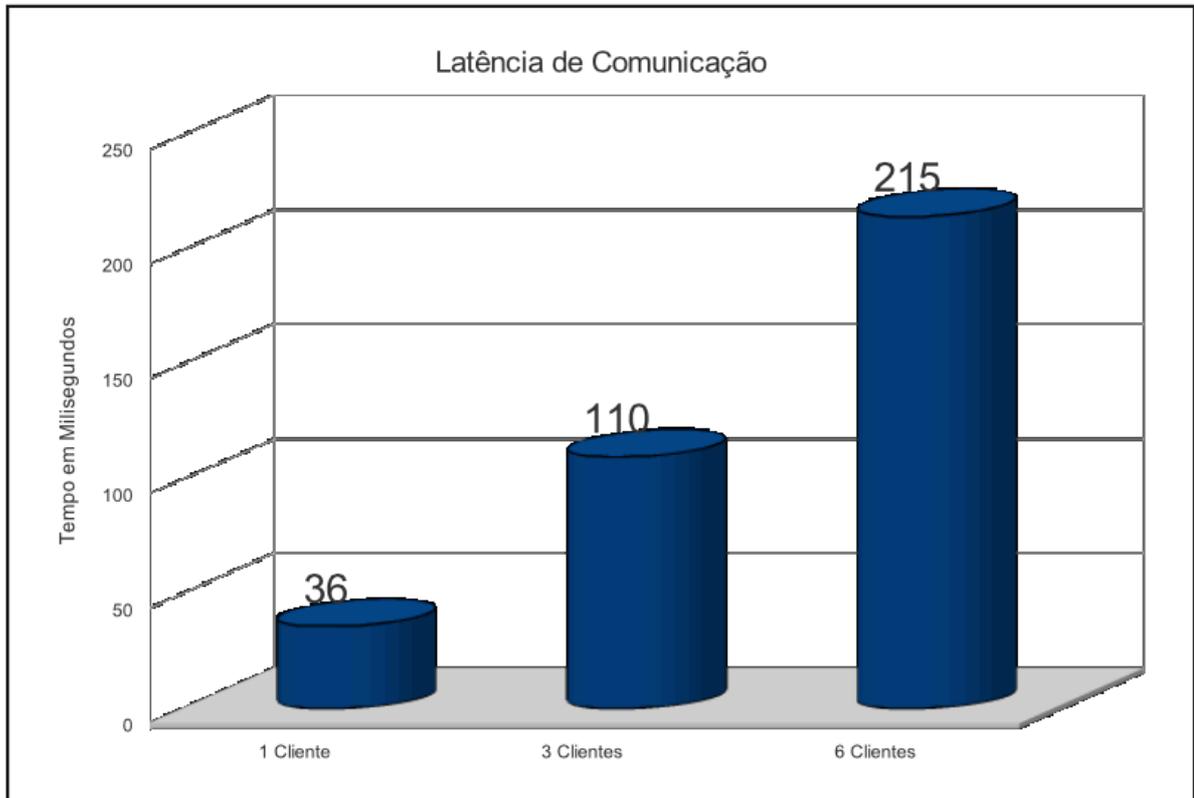


Figura 36 – Gráfico latência de Comunicação

Fonte: SILVA, SANTOS, entre outros (WRVA, 2010, p.54)

A escalabilidade (figura 37), foi medida em função do resultado da análise anterior de latência, levando-se em conta a degradação do tempo de comunicação com o aumento do número de clientes. A escalabilidade foi testada, aumentando esta quantidade de 6 clientes anteriormente para 12, 18, 24 e 30 clientes, respectivamente. Resultando no gráfico abaixo.

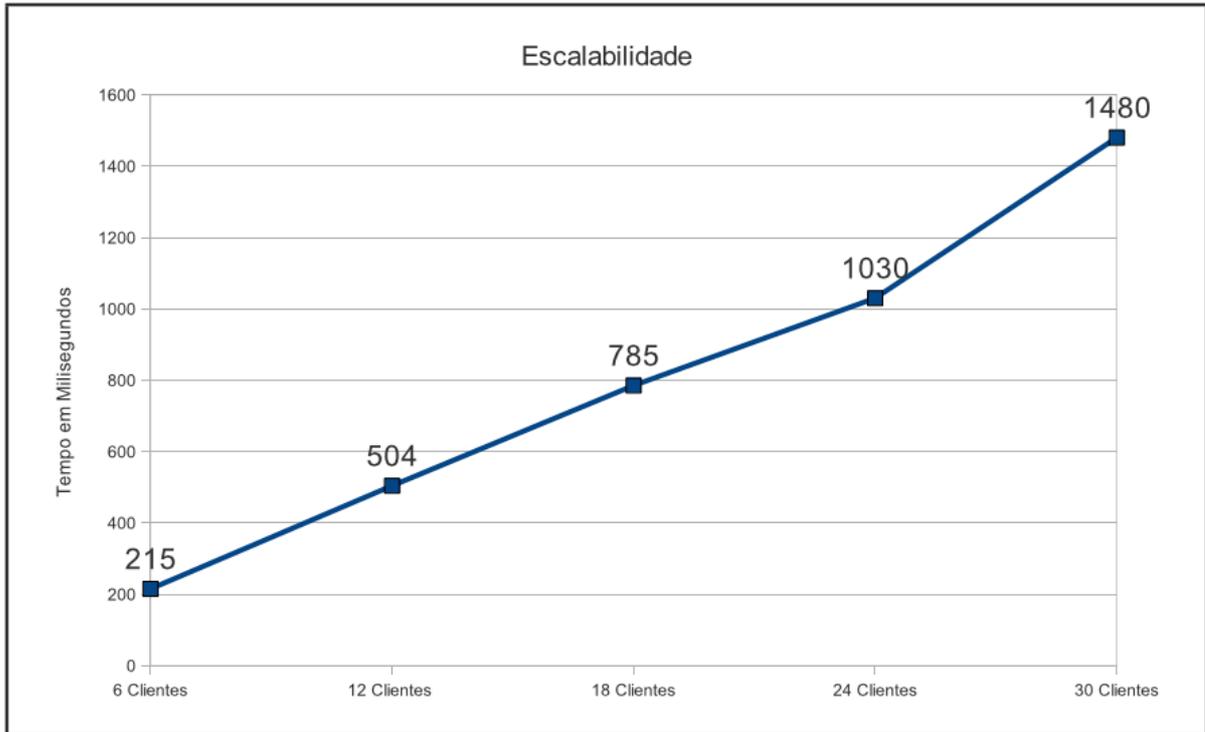


Figura 37 – Gráfico de Escalabilidade

Fonte: SILVA, SANTOS, entre outros (WRVA, 2010, p.54)

Em seguida referente realizaram o teste de aceitação beta (figura 38), apresentando a uma turma de 35 alunos divididos em grupos, que por final responderam um questionário e com as informações do mesmo, foi possível gerar o seguinte gráfico.

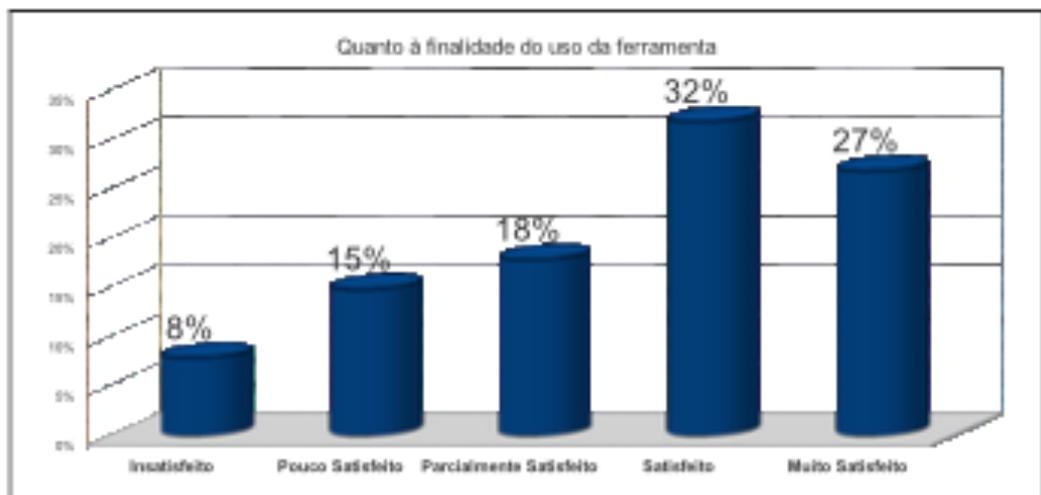


Figura 38 – Gráfico finalidade de uso da ferramenta

Fonte: SILVA, SANTOS, entre outros (WRVA, 2010, p.54)

A maioria considerou satisfeita com o sistema. Porém, algumas pessoas ressaltaram que para operar o *software*, as pessoas necessitam de certo conhecimento de informática, ainda alguns usuários comentaram que o software poderia suportar mais objetos virtuais na cena. Estes seriam os principais resultados do teste de implantação (aceitação beta).

4.4 REALIDADE AUMENTADA IMERSIVA NO AMBIENTE TELEVISIVO

Pesquisa analisada: Realidade Aumentada Imersiva no Ambiente Televisivo, de: FERREIRA, FILHO (2011).

Este projeto aborda o desenvolvimento de um ambiente de RA no ambiente televisivo, bem como as aplicabilidades publicitárias, interativas, informativas e de entretenimento, ele busca a interatividade que a televisão digital poderá proporcionar em termos de qualidade no envio da transmissão, obtendo assim resultados em relação aos resultados no recebimento das imagens com alta definição para o uso de serviços de RA.

O software, através desses equipamentos, identificará os símbolos gráficos chamados de marcadores fiduciais, gerando a interação em tempo real no ambiente televisivo (figura 39 e 40).

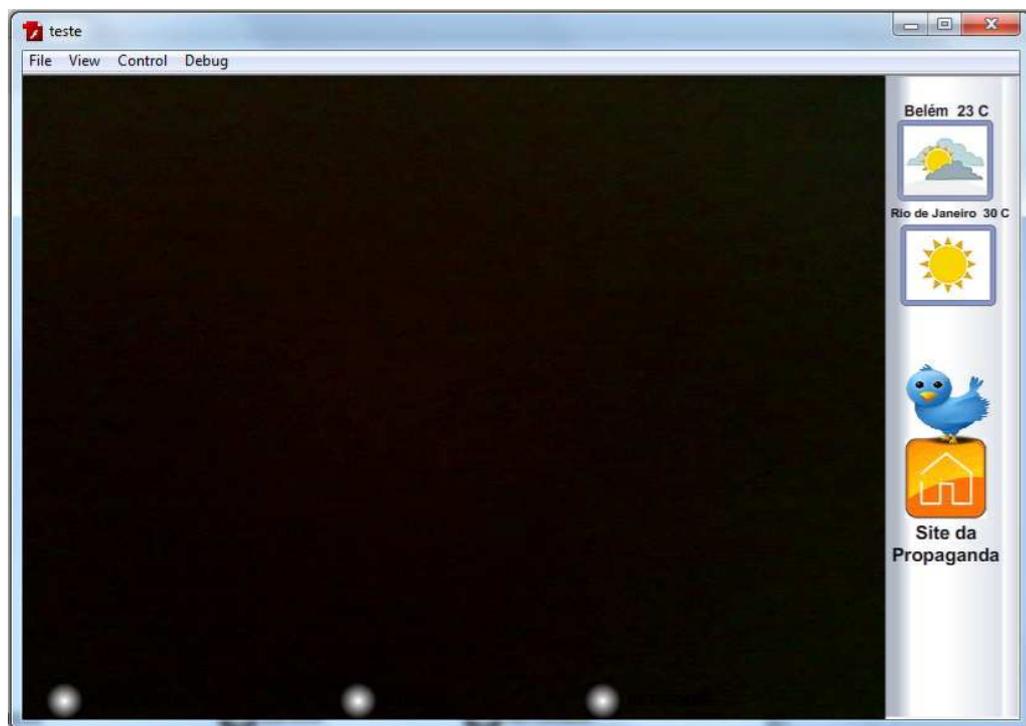


Figura 39 – Visão da Utilização dos Óculos de Imersão

Fonte: FERREIRA, FILHO (2011, pag. 02)

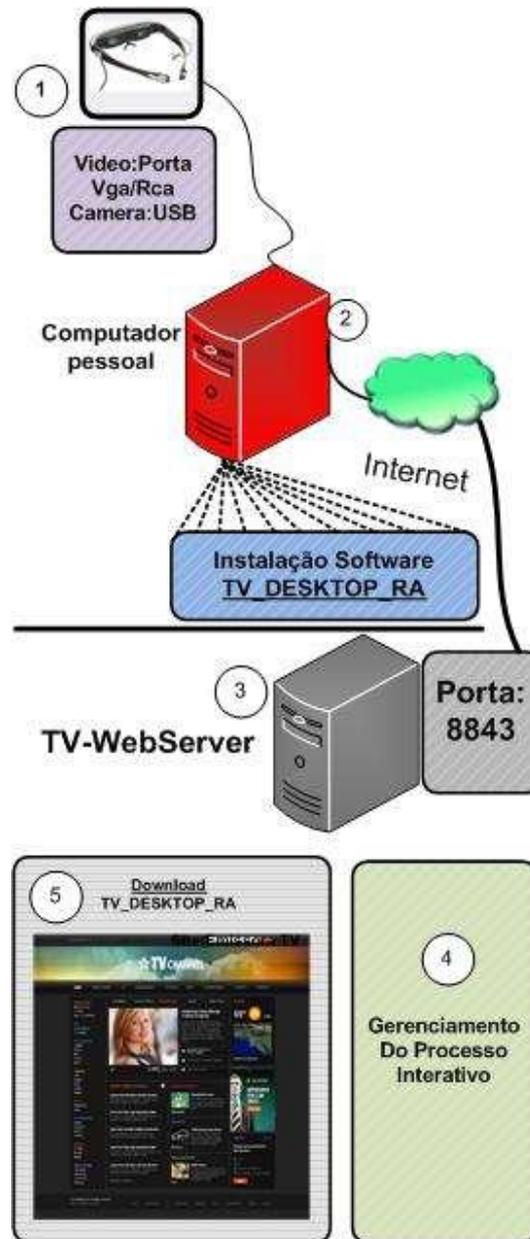


Figura 40 – Fluxo do Sistema RA

Fonte: FERREIRA, FILHO (2011, pag. 02)

Apresentam os testes de funcionalidade do software (Tabela I), onde informam que para o desenvolvimento do software são necessários estudos para analisar suas possíveis falhas de funcionalidades e quais ações deverão ser executadas, caso ocorra alguma situação inesperada e assim descreve através da Tabela - I o fluxo principal que é constituído pela fase de interação do software com conexão em busca dos dados.

TABELA I - FLUXO PRINCIPAL

<ol style="list-style-type: none">1- O usuário aciona os óculos de imersão virtual.2- O sistema inicia acionando a webcam para análise imersiva.3- O sistema se registra no servidor onde se encontra o software de simulação da realidade aumentada.4- O sistema sincroniza com o banco de dados TV-webserver a programação da televisão.5- O sistema recebe as informações do banco de dados TV-webserver.6- O sistema atualiza a base local de imagens e interações para o usuário.7- O usuário recebe imagens em 3d com interações de movimentos.

Fonte: FERREIRA, FILHO (2011, pag. 02)

Ele apresenta também no teste de funcionalidade do sistema o desenvolvimento do tratamento de exceções como mostra na Tabela II, para que se possa analisar os cenários e definir ações. Em seguida ele mostra como deve ser feito os tratamentos de exceções, caracterizando o teste de regressão, onde estes devem ser sistematicamente utilizados durante todas as fases do ciclo de vida do software. Informa também que no presente trabalho "foi utilizado a descrição arquitetural, devido ser uma das melhores formas de se obter uma visão global dos fluxos excepcionais e de como eles são tratados pelos elementos da arquitetura."

TABELA II- TRATAMENTO DE EXCEÇÕES

- 2- O sistema não possui o webcam correta para a análise imersiva.
 - 2.1- Conectar equipamento correto.
 - 2.2- Retornam ao fluxo principal no passo 2.
- 3- O sistema não consegue se conectar ao servidor de realidade aumentada.
 - 3.1- Cadastro no servidor não efetuado.
 - 3.2- O usuário se cadastra no site do canal de televisão que necessita conectar com o respectivo servidor de RA.
 - 3.3- Retorna fluxo principal no passo 4.
- 4- O sistema não sincroniza com o banco de dados *TV-webserver* a programação da televisão.
 - 4.1- Excessos de usuários utilizando o banco de dados.
 - 4.2- Apertar o botão de esperar sincronização de bando de dados.
 - 4.3- Retornar fluxo principal no passo 5.
- 5- O sistema não recebe as informações do banco de dados *TV-webserver*.
 - 5.1- Sistemas locais desatualizado.
 - 5.2- Atualizar sistema local.
 - 5.3- Retornar fluxo principal no passo 6.
- 6- O sistema não executa a base local de imagens e interações para o usuário.
 - 6.1- *Plugins* desatualizados ou não instalados.
 - 6.2- Instalar *plugins* de imagens em 3d.
 - 6.3- Retornar fluxo principal no passo 7.

Fonte: FERREIRA (pag. 02)

Os testes funcionais realizados são para verificação de todas as funcionalidades especificadas pelos requisitos do cliente (Tabela III). Ele demonstra as funções operacionais do software, ou seja, a entrada que é que adequadamente deve ser aceita e a saída corretamente produzida. Na Tabela-III ele mostra o teste funcional final, necessário para verificação da qualidade do produto de software que será carregado no computador, ciente de onde será executado o ambiente de RA.

TABELA III- TESTE FUNCIONAL

Possíveis Cenários	WE	OI	CS	PA	SFS	I3D	Ações do Software
Cenário 1	V	V	V	V	V	V	Imersão feita com sucesso.
Cenário 2	F	V	V	V	V	V	Webcam Instalada. Ir ao passo 3.
Cenário 3	V	V	F	V	V	V	Pendências Resolvidas Ir ao passo 4.
Cenário 4	V	V	V	F	V	V	Servidor receptivo a novas conexões Ir ao passo 5.
Cenário 5	V	V	V	V	F	V	Atualizar Sistema Local Ir ao passo 6.
Cenário 6	V	V	V	V	V	F	Atualizar Plugins para Execução das animações 3D;
WE- Webcam OI- Óculos de Imersão CS- Cadastro no Site PA- Programação Atualizada SFS- Sincronização feita com sucesso I3D-Imagens em 3D e Interação							

Fonte: FERREIRA, FILHO (2011, pag. 02)

4.5 UMA PLATAFORMA PARA VISUALIZAÇÃO ESTEREOSCÓPICA HORIZONTAL

Pesquisa analisada: Uma plataforma para visualização estereoscópica horizontal, MADEIRA (WRVA, 2010).

Este trabalho tem o objetivo da criação de um sistema e um hardware capaz de exibir objetos virtuais posicionados sobre uma superfície plana disposta horizontalmente. Para a estrutura física foi desenvolvido dois protótipos, um mais simples que seria possível ser transportado e outro mais sofisticado mas que foi montado permanentemente no laboratório Visgraf do IMPA.

O mais simples possui um monitor CRT na horizontal sobre uma estrutura de aço, madeira e ainda possui um óculos estereoscópico (figura 41).



Figura 41 – Protótipo estereoscópico transportável

Fonte: MADEIRA (WRVA, 2010, pag 152)

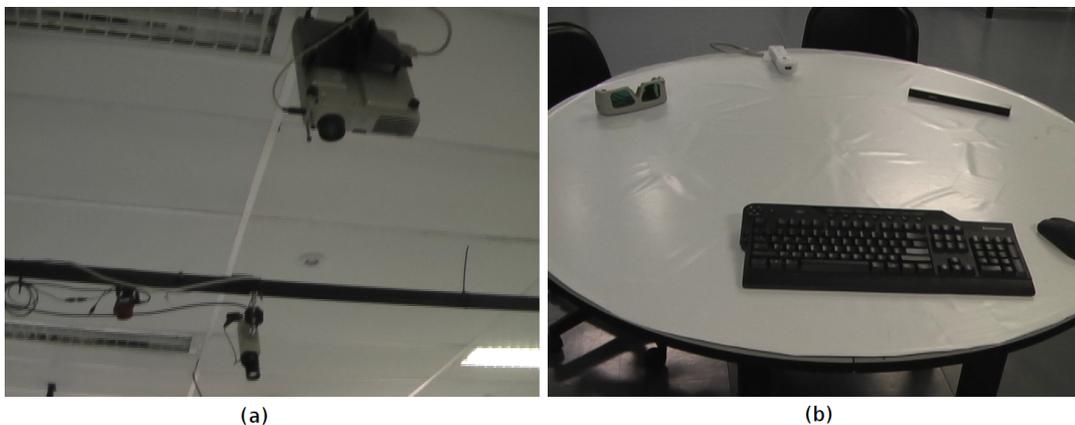


Figura 42 – Estrutura do ambiente proposto.

Fonte: MADEIRA (WRVA, 2010, pag 152)

Já o outro dispositivo é composto por uma projetor estereoscópico fixado no teto do laboratório (figura 42), uma câmera de vídeo, um controle do videogame Wii, óculos 3D e um emissor de luz infravermelha.

Para os softwares implementados foi utilizado na maioria das vezes a linguagem Python, já para a computação gráfica foi utilizado a biblioteca de desenvolvimento de jogos Panda3D, para a integração com a RV foi utilizado o ARToolKit, que já é integrada com o

Panda3D, e também foi utilizado o CWiid para leitura do controle de Wii. Para criarem objetos sobre os protótipos de imagens foi utilizado a OpenGL, da linguagem C. O autor entre em detalhe ainda do funcionamento do dispositivo para realização de testes sobre ambas as mesas. Porém ele ressalta que os resultados interessantes foram os provenientes da linguagem Python com Panda3D (figuras 43 e 44).

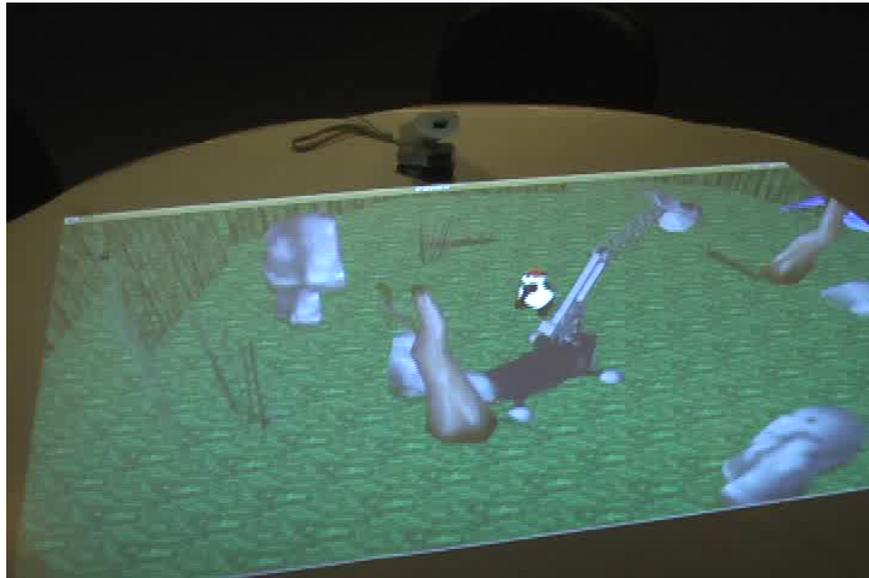


Figura 43 – Aplicação interativa desenvolvida com Panda 3D

Fonte: MADEIRA (WRVA 2010, pag 152)

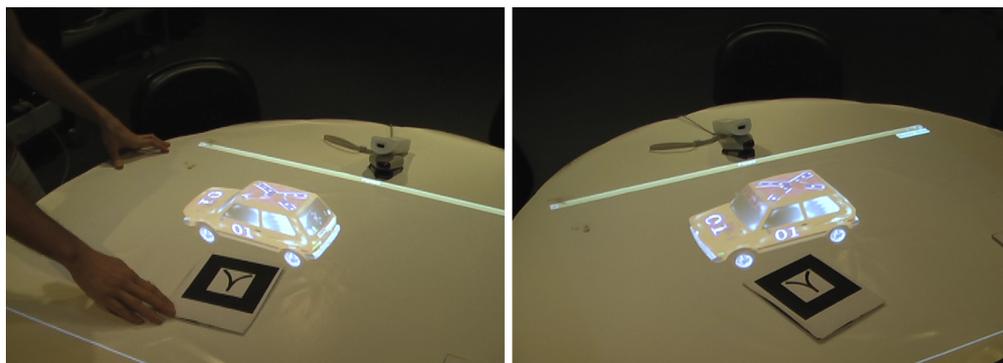


Figura 44 – Um marcador é utilizado para mover o carro virtual, Vê-se a imagem exibida mudando conforme o observador se move em relação a mesa.

Fonte: MADEIRA (WRVA 2010, pag 152)

Não entram em detalhes referentes aos testes realizados no desenvolvimento, nem quanto ao teste de aceitação por usuários, cita apenas os procedimentos realizados quanto ao

funcionamento da aplicação em situações diferentes, podendo assim caracterizar um teste funcional, como o de caixa preta.

4.6 INTELIGÊNCIA ARTIFICIAL APLICADA A JOGOS DE TABULEIRO COM REALIDADE AUMENTADA

Pesquisa analisada: Inteligência Artificial aplicada a Jogos de Tabuleiro com Realidade Aumentada, FERNANDES, CÂMARA, entre outros (WRVA, 2014).

Este trabalho tem a pretensão de analisar um algoritmo de Inteligência Artificial e utilizá-lo em um Jogo Eletrônico de Tabuleiro com RA. O jogo em questão utilizará marcadores para representar um herói e um vilão (figuras 45 e 46), aonde o objetivo do jogo que utilizara o algoritmo de busca A* (A Estrela), é que o vilão capture o herói pelo melhor caminho, passando pelos marcadores que representarão o cenário, cada um com um peso diferente.

No nosso caso foi utilizado a ferramenta ARToolKit, SudaRA com vários modelos visuais, áudio e botões, além de vários exemplos prontos, para implementação foi utilizado a linguagem C com a biblioteca OpenGL. O SudaRA ainda possui suporte a web utilizando cliente/servidor permitindo a conexão por rede, podendo ser criado jogos On-line e de multijogadores implementada em C++.



a)

Figura 45 – A - Imagem dos marcadores

Fonte: FERNANDES, CÂMARA, entre outros (WRVA, 2014, pag 50)



b)

Figura 46 – B - Imagem com os objetos virtuais sobre os marcadores
 Fonte: FERNANDES, CÂMARA, entre outros (WRVA, 2014, pag 50)

Os autores realizaram testes com os algoritmos de Busca Cega e Busca Gulosa, dentro do ambiente do SudaRA, que não mostrou resultados satisfatórios, assim na utilização dos marcadores com os objetos virtuais deixaram a aplicação lenta na execução dos testes, apresentando como resultado um jogo confuso, onde o objetivo era apenas calcular o caminho com menor peso, com base nos dados representados por cada objeto virtual dos terrenos, assim alteraram para Busca A*.

Foram avaliados a qualidade da WebCam, distância de tabuleiro e hardwares utilizados para o teste de carga, aonde foi necessário diminuir os cenários e objetos de 22 marcadores iniciais para apenas 10 sanando o problema. Futuramente pretende trocar os dispositivos para poder aumentar novamente o numero de objetos no jogo e também poder aplicar outras heurísticas de busca e ainda utilizarem NPC e adicionando ao cenário vários obstáculos fazendo-o ainda mais desafiador.

4.7 UMA APLICAÇÃO DE REALIDADE AUMENTADA MÓVEL PARA AMBIENTES INDOOR E OUTDOOR

Pesquisa analisada: Uma Aplicação de Realidade Aumentada Móvel para Ambientes Indoor e Outdoor, de: SANTOS, CARNEIRO, entre outros (WRVA, 2014).

Neste artigo é apresentado uma aplicação RAM que agrega diversas tecnologias que possibilitam ao usuário explorar locais tanto *indoor* quanto *outdoor*. Algumas características

são a navegação em RA, como em Magic Lens descrita por CAWOOD (NC, 2007); foi desenvolvido um módulo para identificação de padrões de imagens na cena real utilizando o QRCode, e também reconhecimento de imagens pré-cadastradas. É possível ainda visualizar e interagir com conteúdos digitais associados há um determinado Ponto de Interesse (POI), do tipo texto, vídeo, áudio, imagem, objetos 2D e 3D.

Esta aplicação foi desenvolvida para a plataforma Android 4.2.2, utilizando rastreamento através do sensor de GPS, Barcode Scanner para leitura do QR Code e ORB (Oriented FAST and Rotated BRIEF) e por final foi realizado um teste de aceitação por usuários Alfa.

O projeto em questão foi implementado utilizando o estilo arquitetural Web de RA (cliente/servidor). Assim a aplicação terá acesso direto a rede, através de protocolos independentes de provedor de plataforma, dividindo a lógica da exibição de conteúdo entre a aplicação móvel (cliente) e servidor.

Para a aplicação foram criados 3 módulos principais, o gerenciador de comunicação que é responsável por codificar as requisições, comunicar com o servidor e por fim realizar o tratamento dos retornos. Também foi implementado o Núcleo que é uma aplicação que intermédia a comunicação dos demais componentes controlando quais modelos serão responsáveis por cada operação, ele ainda designa a quem deve ser encaminhada a resposta do servidor. Por último é apresentado o Misturador de Mundos que é responsável por manter a relação entre as imagens capturadas pela câmera (Mundo Real) e as informações recebidas do servidor (Mundo Virtual), também é responsável pela leitura e sincronização dos marcadores em QRCode. Na figura a seguir, a aplicação está com a função de navegação de RA ativa e o marcador de localização está mostrando a direção da biblioteca central da universidade (figuras 47 e 48).

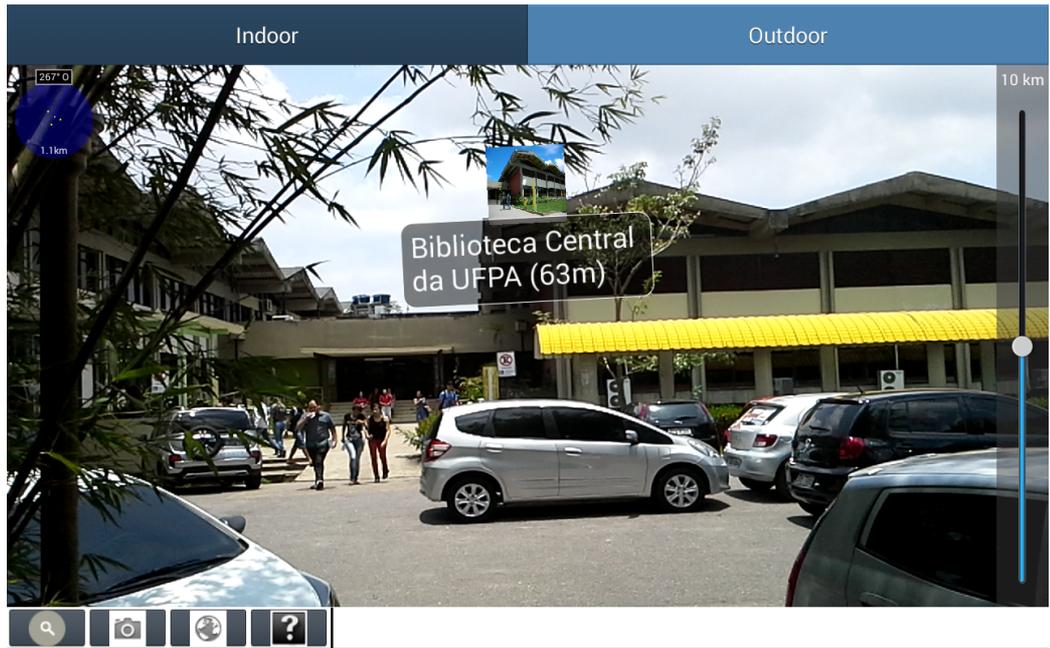


Figura 47 – Captura de tela da aplicação no contexto da universidade.

Fonte: SANTOS, CARNEIRO, entre outros (WRVA, 2014, pag 120)

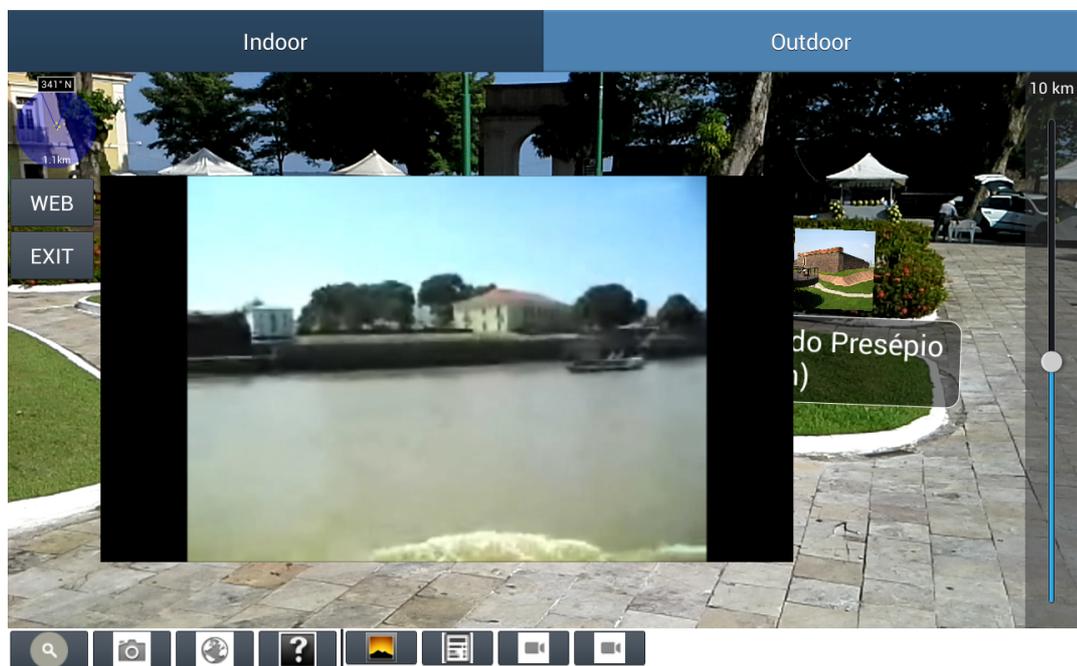


Figura 48 – Captura de tela da aplicação exibindo mídia de vídeo.

Fonte: SANTOS, CARNEIRO, entre outros (WRVA, 2014, pag 120)

Ao apresentar o teste de usabilidade para 3 homens e 2 mulheres do curso de ciências da computação e sistemas de informação da FACOMP (Universidade Federal do Pará). Quanto as informações do dispositivos, os testes foram realizados em um Galaxy Tab 3 da Samsung de 8” com Android 4.2.2 individualmente para os usuários. No final foi gerado algumas tabelas com os resultados baseados nas informações que o avaliador pontuou durante o procedimento dos usuários ao executar as tarefas (tabelas IV, V e VI).

TABELA IV - Tempo De Execução Das Tarefas Pelos Usuários

Tarefa	Usuário 1	Usuário 2	Usuário 3	Usuário 4	Usuário 5	Média
1	4s	5s	7s	5s	4s	5s
2	3s	4s	6s	5s	4s	4,4s
3	9s	8s	9s	8s	7s	8,2s
4	9s	9s	10s	9s	8s	9s
5	25s	30s	35s	28s	30s	29,6s
6	20s	25s	60s	24s	22s	30,2
Média	11,67s	13,5s	21,17s	13,17s	12,5s	x

Fonte: SANTOS, CARNEIRO, entre outros (WRVA, 2014, pag 120)

TABELA V - Acertos e Erros dos usuarios na execução das tarefas

Tarefa	Usuário 1	Usuário 2	Usuário 3	Usuário 4	Usuário 5
1	acertou	acertou	acertou	acertou	acertou
2	acertou	acertou	acertou	acertou	acertou
3	acertou	acertou	acertou	acertou	acertou
4	acertou	acertou	acertou	acertou	acertou
5	acertou	acertou	acertou	acertou	acertou
6	acertou	acertou	não finalizou	acertou	errou

Fonte: SANTOS, CARNEIRO, entre outros (WRVA, 2014, pag 120)

TABELA VI - Pontos de Interesse da Aplicação de RA Móvel no Campus Universitário.

Pontos de Interesse	Texto	Imagem	Vídeo	Link
Biblioteca Central	Sim	Sim	Não	Sim
Centro de Convenções	Não	Sim	Não	Sim
Reitoria	Não	Sim	Não	Sim
Centro de Convivência	Não	Sim	Não	Sim
Capela Universitária	Não	Sim	Não	Sim
Ginásio	Sim	Não	Não	Não
Restaurante Universitário	Sim	Sim	Não	Sim
Hospital Universitário	Sim	Sim	Não	Sim

Fonte: SANTOS, CARNEIRO, entre outros (WRVA, 2014, pag 120)

Foram relatadas pelos usuários algumas questões que deveriam ter mais mensagens apresentando erro/falha ou confirmação de certas operações na aplicação, também que deveria haver uma forma de marcar os ícones já utilizados para não confundirem.

Também relataram dificuldades em utilizar os serviços de reconhecimento de imagem, por conta de lentidão na aplicação ou por conta da posição adequada para o reconhecimento. Tiveram dificuldade em seguir uma trajetória até o ponto de interesse, alguns botões de mídia aparecem sem que o usuário perceba em algumas vezes.

Sendo assim este trabalho apresenta algumas informações em relação ao desempenho da aplicação no dispositivo, mas o teste que mais teve foco foi no quesito de aceitação Alfa por usuários da faculdade.

CAPÍTULO 5 RESULTADOS

Os resultados foram classificados e serão apresentados neste capítulo da seguinte forma: estágios de teste, técnicas de teste e tipos de teste.

Após apresentar um breve resumo, identificaram-se quais as ferramentas utilizadas para o desenvolvimento, como as bibliotecas ARToolKit, OpenGL, ou softwares como o ITV. Em alguns trabalhos também foram identificados os dispositivos que processaram as informações de entrada e saída, que realizam iterações com o meio externo, como webcams, projetores e monitores de vídeo.

Conforme eram apresentadas informações referentes a hardwares e softwares, que deveriam ser utilizados para obter um ótimo processamento de imagens ou para que obtivesse uma ótima performance, estes foram pontuados para gerar os gráficos com os resultados.

Foi verificado que os tipos de teste que mais tiveram foco foram em relação a desempenho, carga e distribuição para multi-plataformas. Na maioria dos trabalhos os autores tinham a intenção de disponibilizar o sistema desenvolvido, como por exemplo, um sistema de RA via Web por meio de uma arquitetura cliente/servidor, podendo ser dividido o processamento no envio das requisições e pelo servidor.

Um dos focos da análise é verificar os dispositivos utilizados e identificados em cada artigo para navegação no ambiente virtual ou em RA. Em trabalhos de RA é possível identificar dispositivos para imersão e interação com a aplicação com a aplicação, como Kinect que faz a leitura dos movimentos do usuário que estará manipulando a aplicação para interagir com o ambiente ou por meio de uma simples webcam, também foi identificado as formas de apresentação do software ao usuário nestes ambientes como projetores e monitores de vídeo/tv.

Ainda sobre RA, estas aplicações normalmente necessitam de um processamento maior que em RV, por conta de ter a necessidade de capturar a imagem com a movimentação/interação, e realizar o processamento, identificar cada objeto daquela cena e realizar o devido tratamento na aplicação, estas na maioria das vezes são implementadas em dispositivos com um maior grau de processamento.

Tanto em RA e RV foi identificado que há uma grande preocupação quanto ao desempenho por conta dos dois tratarem imagens, figuras ou objetos em 3D ou ainda simularem um universo virtual.

Em RV cada autor que identificou informações quanto a análise de desempenho, testes de cargas, e sistema se preocuparam também em tentar disponibilizar a aplicação

através de uma página Web, utilizando o tratamento de cliente/servidor, onde o usuário (cliente) ao acessar a aplicação poderia processar em seu próprio computador as informações de entrada, por exemplo, e ficando de responsabilidade do servidor apenas a disponibilização das informações solicitadas, dividindo assim o desempenho e contribuindo para uma qualidade e velocidade da aplicação no software.

Foram especificados testes de latência e escalabilidade, onde testaram o tempo de envio pelo cliente do pacote e do servidor para o cliente, aumentando o número de usuários gradativamente para fazer requisições ao servidor.

Pode-se verificar que alguns autores destes trabalhos tinham conhecimento em engenharia de software, e além de passar informações quanto aos testes de desempenho também especificaram os diagramas de classes, e os requisitos com as funcionalidades do sistema e a arquitetura do projeto..

Quanto as ferramentas, aplicações, bibliotecas e linguagens de programação que foram utilizadas, constatou-se que a maioria das implementações utilizaram bibliotecas que possuem uma documentação e alta aceitação por desenvolvedores, como AR Toolkit (para linguagem C), NYARToolkit (para Java) ou FLARToolkit (para Flash) que são um conjunto de classes desenvolvidas e distribuídas como *opensource*, ou seja, possuem o código aberto, com isso qualquer desenvolvedor pode utilizá-los para implementação de seus próprios sistemas. Também foram utilizadas ferramentas como 3Ds Max que é utilizada para a modelagem 3D, Adobe Photoshop CS5 para manipulação das imagens, entre outras.

Em partes dos trabalhos é possível identificar as classes que foram implementadas, suas funcionalidades e como devem se comportar no momento que são executadas individualmente ou quando chamam outras classes e métodos, ou ainda recebem seus respectivos dados de retornos e variáveis pertinentes a elas, caracterizando o chamado teste de caixa branca ou teste estrutural, especificaram ainda como era o funcionamento do teste de integração, descendente ou ascendente como por exemplo no projeto de implementação de testes de Grafos de Cena que testam cada nó do objeto em cada cena da aplicação de RA ou RV.

Citando o teste de Caixa Preta ou teste funcional, foram identificados-se dezenove trabalhos neste quesito, onde está incluso os trabalhos que foram especificados suas funcionalidades ou modo que as aplicações devem se comportar conforme cada entrada ou movimento e também os requisitos não-funcionais.

Alguns artigos especificaram o uso de mais de uma técnica de teste de software na aplicação, onde foi possível identificar também os testes baseados em defeitos que seriam

testes de desempenho como o de carga, stress, avaliação de desempenho, e contenção, além do teste de recuperação. Sendo assim, geramos um gráfico que apresenta as técnicas de testes que foram apresentadas até agora, caixa branca, caixa preta e baseado em defeitos. Na figura 49 é possível verificar a porcentagem das técnicas de teste utilizadas nas atividades de teste no desenvolvimento de Sistemas de RV e RA encontrados nos trabalhos.



Figura 49 – Técnicas de Teste (Caixa Branca, Caixa Preta e Baseado em Defeitos)
Fonte: Elaborada pelo autor

Avaliando o teste de aceitação, pode-se afirmar que estes são os testes que mais são visados em sistemas de RV e RA, no presente trabalho reunimos 5 destes onde se dividiram entre teste de aceitação Alfa e o teste Beta (Quando usuários que não participaram do desenvolvimento que realizam os testes, na maioria das vezes estes são os clientes) e no final realizam um questionário ao usuário ou equipe de testes, que avalia como o software se comportou em relação ao que era esperado, também verificam se os usuário acha o software útil ou se é muito complexo não conseguindo manuseá-lo.

Pode-se classificar nestes trinta e sete por cento de testes e caixa branca, que a maioria utilizou as ferramentas ARToolKit (C/C++), NYARToolkit (Java/C#/AS3) e FLARToolkit (AS3/Flash Web) estas são todas *opensource*, ou seja, possuem o código livre e qualquer desenvolvedor pode utilizá-las para suas implementações, onde se torna uma grande contribuição para a área de RA e RV. Estas ferramentas estão no conceito da GPL (General Public License), assim, todas as aplicações que utilizam estas ferramentas devem ser publicadas para serem utilizadas em trabalhos futuras e implementadas por outros desenvolvedores.

Classificou-se através do gráfico de barras a seguir (Figura 50) quanto aos estágios de testes, está dividido em testes estruturais de unidade e integração, quanto ao teste funcional que seria o teste de sistema e o teste de implantação que seria referente aos testes de aceitação dos usuários ou clientes e pela equipe teste (Alfa).

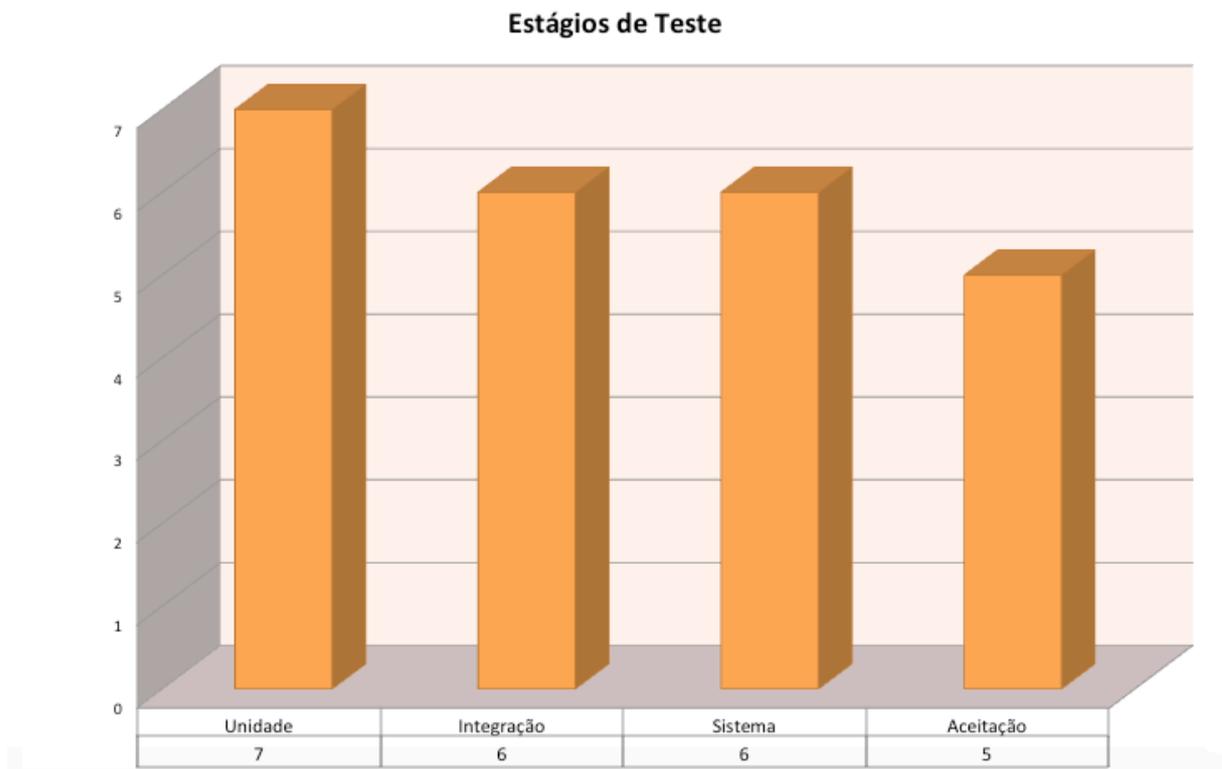


Figura 50 – Estágios de Teste (Unidade, Integração, Sistema e Aceitação)

Fonte: Elaborada pelo autor

Na figura 51 a seguir será apresentado quanto aos tipos de teste, contendo a quantidade de trabalhos classificados para cada um e em seguida a tabelas classificando cada artigo e quais testes foram realizados em cada um, os tipos de teste foram separados

respectivamente em testes de funcionalidade, testes estruturais e por último em teste de implantação (aceitação).

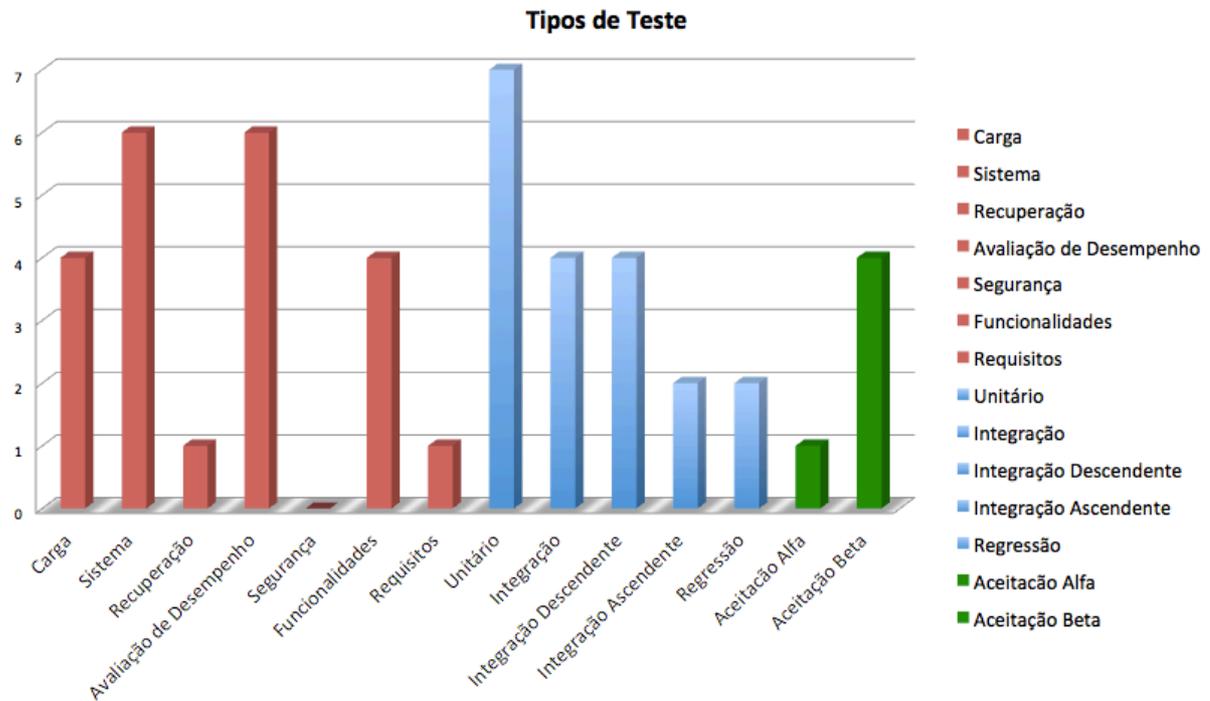


Figura 51 – Tipos de Teste: (Todos) ou os principais.
Fonte: Elaborada pelo autor

Com este gráfico pode-se analisar que os testes mais utilizados são os testes estruturais unitários, teste de desempenho e carga que envolvem latência e escalabilidade e os testes de aceitação (alfa e beta).

Nos testes unitários, o desenvolvedor detalha funcionalidade que são implementadas referentes a uma classe ou objeto específico com seus métodos, atributos e funções na aplicação. Mostra-se como a classe é e como deve se comportar com os retornos dos movimentos vindos do mundo real.

Um informação importante em relação a testes se faz pelos teste de segurança, concluiu-se que não há preocupação com esta etapa em momento algum, quem desenvolve a aplicação se preocupa somente em finaliza-la e disponibilizar via Web por exemplo, para usuários poderem utilizar, mas a questão de segurança da aplicação pode ficar vulnerável desta forma, pois não há um *CheckList* realizado antes da liberação testando as funções e como se comportaram em um ataque, ou com um código malicioso que for inserido na aplicação.

O teste de desempenho que envolve o teste de carga, stress e de esforço, pode-se considera-lo como o mais importante para a área de RA e RV, por serem sistema de tempo real no qual ao usuário realizar uma movimentação, a aplicação deve fazer a leitura imediatamente para ter esta resposta em tempo real. Testes de latência e escalabilidade são úteis para medir a eficiência e desempenho da aplicação, prevenindo mal funcionamento em ambientes de execução desconhecidos.

Outro método que costumam utilizar é o teste de aceitação, em relação ao gráfico da figura 51, vemos a utilização em alguns trabalhos. O teste de aceitação alfa é importante principalmente para liberação para usuários finais, antes que seja liberados para eles, deve-se realizar este tipo de teste e validar se o ambiente que foi definido não possui *bugs*, que poderiam ocasionar erros desconhecidos como o usuário se perder no ambiente virtual proposto. Após este tipo de teste, pode-se liberar o teste de aceitação beta, para usuários que não tem conhecimento algum da ferramenta, após os testes proposto, dão o *feedback* em relação a navegação em RA e RV, classificam se o sistema é pertinente a área que foi proposta, se possui fundamento e se a ferramenta é útil. Pode-se nos dois casos classificar e pontuar as tarefas propostas durante a utilização do sistema, como exemplo se ao utilizar a ferramenta tiveram dificuldade.

Na tabela VII visualiza-se a classificação inicial utilizada para gerar os gráficos apresentados neste trabalho, nela foi relacionado cada trabalho e assinalado quais os tipos de testes se fazem pertinentes a cada aplicação de acordo com a numeração dos capítulos. Um trabalho pode estar relacionado a um teste ou mais, que se foram aplicados e informados na documentação final. A coluna “NDA” representa os trabalhos que não possuem testes especificado na monografia ou artigo, aonde foi mostrado apenas a especificação da funções do sistema, estes trabalhos não foram relacionados neste artigo, identificou-se eles como “Outros Trabalhos”.

Na figura 52 é apresentando as técnicas de testes comparando-as com trabalhos que não foram detalhados neste estudo, onde verificou-se que não demonstrava nenhum tipo de teste, nestes sistemas definidos como “Não Utilizaram Técnicas”, pode-se analisar apenas informações sobre o sistema ou sobre o que é cada projeto, não passou-se dados quanto a implementação ou disponibilidade do software para usuários.

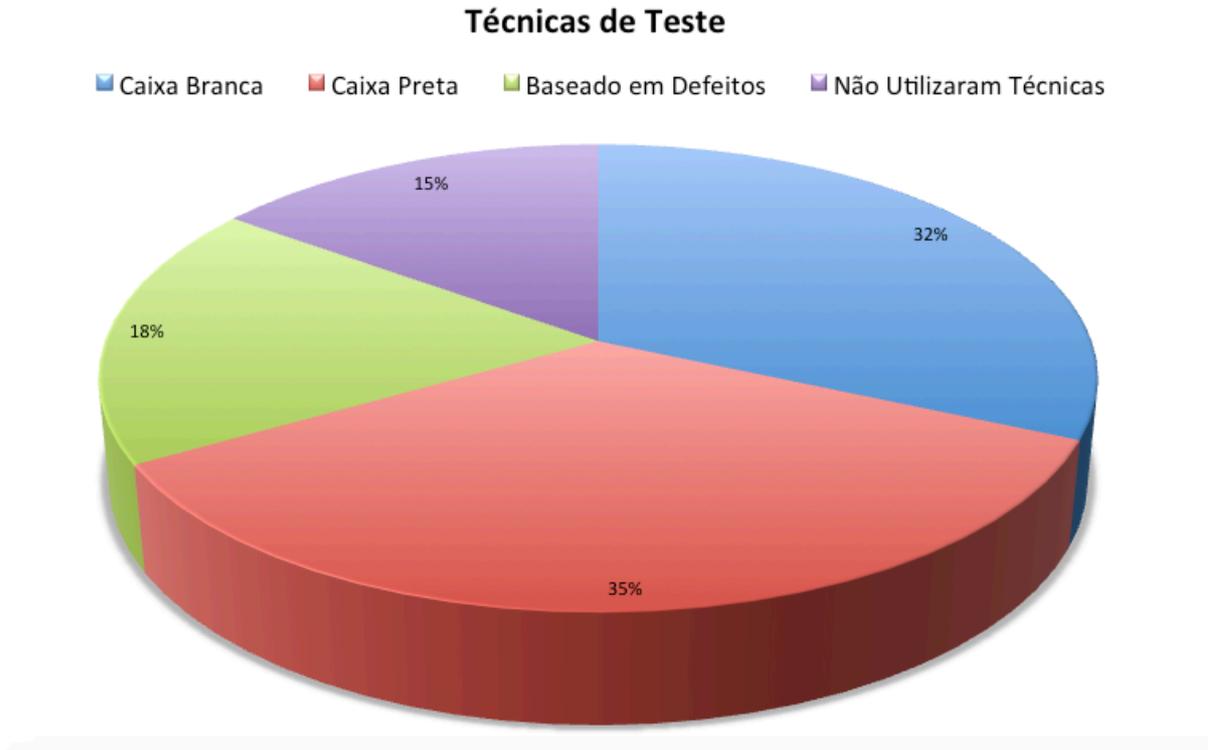


Figura 52 – Tipos de Teste: (Todos) ou os principais.
Fonte: Elaborada pelo autor

Em adição, pode-se concluir também pela análise de outros trabalhos de RA e RV que não foram citados neste artigo, que muitos não possuem nenhuma especificação em momento algum em relação aos testes, somente apresentam informações referentes ao que se trata o trabalho.

Estes trabalhos deveriam ao menos utilizar o teste de desempenho, que como citado anteriormente, é utilizado para que o sistema tenha respostas aos usuários em tempo real, aumentando a experiência com o ambiente virtual tanto em RV, como em RA.

TABELA VII – Classificação dos Trabalhos

Tipos/Técnicas	Caixa Preta							Caixa Branca					Aceitação		NDA
	TSC	TST	TSG	TSR	TAD	FUN	REQ	TEU	TEI	TED	TEA	TER	TAA	TAB	
3.1		■						■						■	
3.2						■									■
3.3		■			■										
3.4														■	
3.5								■	■	■					
3.6	■							■	■	■					
3.7						■		■				■			
3.8								■							
3.9								■	■	■	■				
3.10		■			■	■									
3.11	■				■									■	
3.12								■		■	■	■			
4.1	■				■				■						
4.2		■					■								
4.3		■												■	
4.4				■											
4.5		■				■									
4.6	■				■										
4.7					■								■		
Total	4	6	0	1	6	4	1	7	4	4	2	2	1	4	1
Outros Trabalhos															9

Fonte: Elaborada pelo autor

Legenda da Tabela VII:

TSC: Carga,

TST: Teste de Sistema

TSG: Teste de Segurança

TSR: Teste de Recuperação

TAD: Teste de Avaliação de Desempenho

FUN: Apenas informações quanto a Funcionalidade

REQ: Requisitos

TEU: Teste Unitário

TEI: Teste de Integração

TED: Teste de Integração descendente

TEA: Teste de Integração ascendente

TER: Teste de Regressão

TAA: Teste de Aceitação Alfa

TAB: Teste de Aceitação Beta

NDA: Nenhum tipo de teste

Procurou-se fundamentar conceitos sobre as tecnologias envolvidas e apresentar trabalhos na área em foco, bem como mostrar a importância dessas tecnologias na busca de novas formas de interação com o computador e dispositivos eletrônicos.

Por fim, pode-se afirmar pelos resultados obtidos que atualmente a atividade dos testes de software realizados tanto no desenvolvimento de sistemas de RV como RA, ainda não seguem uma documentação padrão de engenharia de software e análise de requisitos, muitos dos trabalhos somente citam informações referentes a testes para conseguirem finalizar suas monografias e artigos de uma forma que mostre que a ferramenta está em funcionamento ou que foi executada em algum momento, não se preocupando com a definição destes dados corretamente.

CONCLUSÃO

Foi apresentado neste trabalho a análise dos estágios, tipos e técnicas de testes utilizados em RV e RA. Pode-se concluir que o desenvolvimento de sistemas que utilizam RA e RV ainda não seguem um padrão ou um processo de teste muito bem definido, tão pouco os que já existem na bibliografia sobre Engenharia de Software. As Atividades de Verificação, Validação e Teste de Software (VV&T), ainda são pouco exploradas nesta área.

Sistemas de RV são sistemas de tempo real por este motivo, todos os sistemas deveriam realizar ao menos teste de desempenho, para assegurar a característica deste tipo de sistema.

A partir deste pode-se propor que seja feita a documentação específica para cada estágio e fase de teste, para que seja identificado problemas que podem ocorrer desde a análise de requisitos, funcionalidades, implementações e implantação.

Propor também que para trabalhos futuros nesta área sejam especificados quais os testes de desempenho que foram realizados, visando identificar se a performance da aplicação irá atingir o tempo real. Há necessidade também de realizar a documentação de como a aplicação deve se comportar em diferentes ambientes e verificando se a aplicação irá se adaptar a iluminação por exemplo.

Deve ser levado em consideração a questão de testes de segurança e de stress que seriam interessantes serem aplicados nestes software, principalmente nos que possuem interface WEB com comunicação baseada em cliente/servidor.

Para estas aplicações seria interessante ainda que fosse desenvolvido uma base de dados centralizada, onde poderiam subir aplicações *opensource* ou informações em relação a RA e RV, especificando os testes realizados e como suas aplicações se comportam a diferentes ambientes distintos, para que contribuíssem com melhorias em softwares nesta área, sendo mais documentada e concentrada com técnicas diferenciadas neste domínio.

Como teste de desempenho seria interessante que utilizassem também aplicações que necessitem de grande quantidade de recursos de rede e processamento, como por exemplo, aplicações de vídeo em tempo real, para realização de testes em todos os trabalhos, e no final pontuar o desempenho.

Outra consideração importante de ser verificada futuramente está relacionada com a implementação de diferentes níveis de segurança. Seria interessante na construção dos sistemas, uma forma que auxilie na determinação do nível de segurança requerido por determinado usuário ou aplicação, visando a segurança da informação.

REFERÊNCIAS

AZUMA, R. et al. (2001) “Recent Advances in Augmented Reality”. IEEE Computer Graphics and Applications, v 21, n.6, p. 34-47.

BEIZER, B. Software System Testing and Quality Assurance, Van Nostrand-Reinhold, 1984.

BENDZ, Jon Eskil. Modelagem, simulação, e visualização imersiva de redes sem fio. 2008. Tese (Doutorado em Sistemas Eletrônicos) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2008. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3142/tde-30092008-144928/>>. Acesso em 23 de Novembro de 2014.

BEZERRA, Adriano; NUNES, Fátima L. S; DELAMARO, Marcio E. Aplicações baseadas em Grafo de Cena – uma abordagem estrutural para critérios de teste - Universidade de São Paulo (EACH-USP) São Paulo - SP – Brasil. Universidade de São Paulo (ICMC-USP) São Carlos - SP - Brasil adriano@icmc.usp.br. Universidade de São Paulo (ICMC-USP) São Carlos - SP - Brasil delamaro@icmc.usp.br <http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Trabalho?id=11879>. Acesso em 23 de Novembro de 2014.

BINDER, R. V. *Testing object-oriented systems: models, patterns, and tools*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

BOEHM, B., Software Engineering Economics, Prentice Hall, 1981.

BURDEA, G. C.; COIFFET, P. Virtual Reality Technology, Wiley-Interscience, USA, 1994.

BUCIOLI, Arthur Augusto Bastos; LAMOUNIER, Edgard A Jr. Visualização de ambientes virtuais com simulação de projeção holográfica. Professor Bucioli EBTT IFTM Campus Patrocínio. arthurbuccioli@gmail.com - Edigar Jr Coordenador Pós-Graduação Engenharia Elétrica Universidade Federal de Uberlândia. Uberlândia-MG, Brasil. elamounier@gmail.com, Publicado em WRVA, 2014.

CARVALHO, Caio Moreira. Navegação, Reconhecimento de gestos e controle de interface no sistema ITV utilizando o dispositivo Kinect. Universidade Federal do Pará, 2013. <http://repositorio.ufpa.br/jspui/bitstream/2011/4605/1/Dissertacao_NavegacaoReconhecimen%20toGestos.pdf> Acesso em 23 de Novembro de 2014.

CAWOOD, S. M. Fiala, and D. H. Steinberg, Augmented reality: a practical guide. Pragmatic Bookshelf Raleigh, NC, 2007.

COSTA, Junior, R.A.; SILVA, R.C. ; CERQUEIRA, C.S.; Almeida, Realidade Aumentada para Auxiliar o Aprendizado de Motor Elétrico. <<http://www.ckirner.com/sacra/>> Acesso em 23 de Novembro de 2014. Publicado em WRVA, 2010.

DELAMARO, M.E.; MALDONADO, J.C.; JINO, M. Introdução ao Teste de Software. Rio de Janeiro: Campos, 2007.

FALCÃO, Eduardo de Lucena; MACHADO, Liliane dos Santos. Museu 3I: Publicação e Visitação Online de Acervos Tridimensionais. Laboratório de Tecnologias para o Ensino Virtual e Estatística Universidade Federal da Paraíba – CCEN. Publicado em WRVA,2010.

FERNANDES, Anderson Mine; CÂMARA, Bruno Henrique Pachulski; PASCHOAL, Alexandre Rossi; DAMASCENO, Eduardo Filgueiras. Inteligência Artificial aplicada a Jogos de Tabuleiro com Realidade Aumentada. PPGI - Programa de Pós-graduação em Informática. Mestrado em Informática - Universidade Tecnológica Federal do Paraná - Campus Corneio Procópio. Publicado em WRVA, 2014.

FERREIRA, Ronedo S.; FILHO, Emilio Arruda. Realidade Aumentada Imersiva no Ambiente Televisivo. Instituto de Estudos Superiores da Amazônia, Laboratório de Pesquisas Avançadas Belém – PA – Brasil, ronedo_ap@hotmail.com , 2011.<<http://www3.iesampa.edu.br/ojs/index.php/computacao/article/viewArticle/509>> Acesso em 23 de Novembro de 2014.

GADELHA, Matheus Abrantes; SANTOS Selan Rodrigues dos. Controle de Navegação em Ambientes Virtuais 3D através do Rastreamento de Objetos. matheusabrantegadelha@gmail.com, selan@dimap.ufrn.br; Departamento de Informática e Matemática Aplicada — Dia Universidade Federal do Rio Grande do Norte — UFRN Campus Lagoa Nova, 59072-970, Natal/RN, Brasil. Publicado em WRVA, 2010.

GUIMARÃES, Marcelo de Paiva. Um ambiente para o desenvolvimento de aplicações de realidade virtual baseadas em aglomerados gráficos. 2004. Tese (Doutorado em Sistemas Eletrônicos) - Escola Politécnica, University of São Paulo, São Paulo, 2004. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3142/tde-10032005-111137/>>. Acesso em 23 de Novembro de 2014.

ISDALE, J. (2000) “Augmented Reality”. Disponível em <http://vr.isdale.com/vrTechReviews/AugmentedReality_Nov2000.html> Acesso em 20 de Novembro de 2014

KIRNER, Claudio; Gonçalves Kirner, Tereza. Evolução e Tendências da Realidade Virtual e da Realidade Aumentada. Pré-Simpósio SVR, 2011.

KIRNER, Claudio; Santin, Rafael. Funcionamento e Utilização do Sistema de Autoria Colaborativa com Realidade Aumentada – SACRA, UNIFEI, 2011. Normas de Apresentação Tabular (1993). Disponível em <<http://www.ckirner.com/sacra/>> Acesso em: 21 nov. 2014.

KIRNER, C.; SISCOOTTO, R. A. Fundamentos de Realidade Virtual e Aumentada. IX Symposium on Virtual and Augmented Reality. Petrópolis-RJ: SBC. 2007. p. 2-21.

MADEIRA, Bruno Eduardo. Uma plataforma para visualização estereoscópica horizontal. Instituto Militar de Engenharia Praça Gen. Tibúrcio 80 Urca, Rio de Janeiro, RJ madeira@ime.eb.br - Luiz Velho. Instituto Nacional de Matemática Pura e Aplicada Laboratório Visgraf, Rio de Janeiro, RJ lvelho@impa.br. Publicado em WRVA, 2010.

MEDEIROS, Daniel P. S.; MACHADO, Liliane S. X3D e Integração Multimídia para Representação de um Sítio Arqueológico. Laboratório de Tecnologias para o Ensino Virtual e Estatística Universidade Federal da Paraíba - CCEN {[danielpires,liliane](mailto:danielpires@di.ufpb.br)}@di.ufpb.br, Publicado em WRVA, 2010.

MILGRAM, P. et al. (1994) “Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum”. *Telematic and Telepresence Technologies*, SPIE, V.2351, P. 282-292.

NASCIMENTO, Messias José Amador do. Modelagem de objetos tridimensionais para um ambiente interativo de instruções técnicas virtuais, UFPA / ITEC / PPGEE. Campus Universitário do Guamá Belém-Pará-Brasil – 2010. <<http://repositorio.ufpa.br/jspui/handle/2011/2621>> Acesso em 23 de Novembro de 2014.

NEGRÃO, Nathália mendes. SwImax: Um simulador em Realidade Virtual para auxiliar o ensino do padrão WiMAX. - UFPA / ITEC / PPGEE. Campus Universitário do Guamá Belém-Pará, Brasil. Setembro de 2012. < <http://repositorio.ufpa.br/jspui/handle/2011/3381>>. Acesso em 23 de Novembro de 2014.

PAVAN, Adilson Roberto; MODESTO, Fábio Alexandre Caravieri. Reconhecimento de Gestos com Segmentação de Imagens Dinâmicas Aplicadas a Libras. Anhanguera Educacional adilson.pavan@gmail.com fabiomodesto@uol.com.br, Publicado em WRVA, 2010.

PRESSMAN, Roger S. Engenharia de Software: Uma Abordagem Profissional. 7. ed. Porto Alegre: AMGH Bookman, 2011. 780 p.

RODRIGUES, Herbet Ferreira; MACHADO, Liliane dos Santos. Uma Arquitetura para Integração de Sistemas Hápticos e Engines de Jogos. Laboratório de Tecnologias para o Ensino Virtual e Estatística – LabTEVE. Universidade Federal da Paraíba – UFPB. Cidade Universitária s/n – 58051-900 – João Pessoa – PB – Brasil. Publicado em WRVA, 2010.

ROPER, M. Software Testing. McGraw-Hill Book Company Europe, 1994.

SANTOS, Carlos; CARNEIRO, Nikolas; MIRANDA, Brunelli; SERIQUE, Bianchi. Uma Aplicação de Realidade Aumentada Móvel para Ambientes Indoor e Outdoor. Programa de Pós-graduação em Ciência da Computação – PPGCC. Curso de Bacharelado em Ciência da Computação, Faculdade de Computação – FACOMP. Universidade Federal do Pará, Belém, Pará. 66075-110 . Publicado em WRVA, 2014

SHOUMAN, M. L., Software Engineering, McGraw-Hill, 1983.

SILVA, Kleber Anderson Corrêa; KIRNER, Claudio. Desenvolvimento de um Jogo Tridimensional com Realidade Aumentada - Publicado em WRVA 2010

SILVA, Marlus Dias; SANTOS, Eduardo de Souza; CARDOSO, Renato Oliveira Abreu; CARDOSO, Alexandre; LAMOUNIER, Edgard. Desenvolvimento de uma arquitetura para distribuição de realidade aumentada na web aplicada ao ensino de motores de corrente contínua. Universidade Federal de Uberlândia, Uberlândia, MG, Brasil. Publicado em WRVA2010

SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: PEARSON, v. I, 2011.

STEUER, J. Defining Virtual Reality: Dimensions Determining Telepresence. Stanford University, 1993.

TAVARES, A. C. M. Aplicação de um Ambiente Virtual Colaborativo Utilizando Realidade Aumentada. Recife-PE: Escola Politécnica de Pernambuco – Universidade de Pernambuco. 2009.

TORI, Romero; KIRNER, Cláudio; SISCOUTO, Robson. Fundamentos e Tecnologia de Realidade Virtual e Aumentada. Porto Alegre: SBC, 2006, 422 p.

TORI, R.; KIRNER, C.; SISCOOTTO, R. Fundamentos e Tecnologia de Realidade Virtual e Aumentada. Livro do Pré Simpósio VIII Symposium on Virtual Reality, Belém – PA, 02 de

maio de 2006. Versão eletrônica. Disponível em:
<http://www.ckirner.com/download/capitulos/Fundamentos_e_Tecnologia_de_Realidade_Virtual_e_Aumentada-v22-11-06.pdf> Acesso em 20 de Novembro de 2014.