

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



Desenvolvimento de uma arquitetura de distribuição de realidade virtual e aumentada
aplicada em ambientes educacionais

Dissertação apresentada à Universidade Federal de Uberlândia por
KEILA DE FÁTIMA CHAGAS NOGUEIRA

Como parte dos requisitos para obtenção do título de
Mestre em Ciências no Programa de Pós-Graduação em Engenharia Elétrica

Edgard A. Lamounier Junior, PhD – Orientador
Alexandre Cardoso, Dr – Coorientador

UBERLÂNDIA, AGOSTO DE 2010

KEILA DE FÁTIMA CHAGAS NOGUEIRA

Desenvolvimento de uma arquitetura de distribuição de realidade virtual e aumentada aplicada em ambientes educacionais

Dissertação apresentada por Keila de Fátima Chagas Nogueira, à Universidade Federal de Uberlândia (UFU), Faculdade de Engenharia Elétrica, para obtenção do título de Mestre em Ciências.

Área de Concentração

Processamento da Informação

Banca Examinadora:

Edgard A. Lamounier, PhD – Orientador

Alexandre Cardoso, Dr – Coorientador

Claúdio Kirner, PhD

Keiji Yamanaka, PhD

Desenvolvimento de uma arquitetura de distribuição de realidade virtual e aumentada
aplicada em ambientes educacionais

Keila de Fátima Chagas Nogueira

**Dissertação apresentada à Universidade Federal de Uberlândia como parte
dos requisitos para obtenção do título de Mestre em Ciências.**

Prof.Dr. Alexandre Cardoso
Coordenador de Pós-Graduação

Prof.PhD.Edgard Lamounier Júnior
Orientador

À aos meus pais Elias e Fátima e
principalmente para meu amado Kenedy
Lopes Nogueira.

Agradecimentos

- ✓ A Deus, por estar a cada dia me ensinando uma nova maneira de viver e enxergar a vida.
- ✓ Ao Professor Edgard Lamounier pelas orientações, apoio, compreensão e dedicação;
- ✓ Ao Professor Alexandre Cardoso, pelo apoio e orientação.
- ✓ À Coordenação do Curso de Pós-Graduação em Engenharia Elétrica pela oportunidade de realização deste trabalho;
- ✓ Aos professores do curso de pós-graduação pelos conhecimentos que adquiri durante a fase dos créditos;
- ✓ Aos colegas de pesquisa pelo apoio e motivação para vencer os desafios (Kenedy Lopes, Elvio Prado, Luciano Silva, Ezequiel Zorzal, Webert Vieira, Eduardo Souza, Marlus Dias, Daniel Stefany) enfim a todos do laboratório de Computação Gráfica que me apoiaram durante este trabalho.
- ✓ Um agradecimento muito especial a meu amor Kenedy Lopes Nogueira, uma pessoa que está do meu lado sempre apoiando em cada etapa da minha vida. Essa conquista também é sua.
- ✓ Agradecimento especial aos meus pais, Elias e Fátima. O apoio para mim foi muito importante nesta jornada. Obrigada pai e mãe pelo que sou hoje e por terem me auxiliado na conclusão de mais esta etapa em minha vida.
- ✓ A meus sogros Oswaldo e Eliani, a meus irmãos Oswaldo Filho e Selma e todos da minha família que se sempre me apoiaram.
- ✓ A minha amiga Roberta Kelly, pelo apoio de sempre.

- ✓ A meus pequenos afilhados João Lucas e Carlos Eduardo, anjinhos de Deus na minha vida.
- ✓ Enfim a todos que de alguma maneira colaboraram para a realização desta dissertação.

RESUMO

NOGUEIRA, Keila F C. Desenvolvimento de uma arquitetura de distribuição de Realidade Virtual e Aumentada aplicada em Ambientes Educacionais, 2010. 92 P. Dissertação (Mestrado em Ciências) Programa de Pós-Graduação em Engenharia Elétrica, UFU, 2010.

Palavras-Chave: Realidade Virtual, Realidade Aumentada, Sistemas Distribuídos, Aprendizado.

Atualmente, o acesso à informação esta cada dia mais fácil, devido ao avanço da Tecnologia da Informação e Comunicação (TIC), reflexo da popularização do computador e da Internet, tanto nas escolas quanto nas residências.

No mesmo contexto pesquisas e projetos, estão sendo desenvolvidos voltados para novas metodologias de ensino. Esses projetos facilitam novos métodos educacionais quando comparado com métodos tradicionais. Hoje é possível que vários laboratórios (física, química, matemática) possam ser simulados por meio de um único laboratório de informática.

Existem várias tecnologias de suporte à criação destes sistemas, podendo destacar Realidade Virtual que permite criar modelos e ambientes e simulá-los como se fossem reais. Outra tecnologia é a Realidade Aumentada, onde é possível inserir elementos de aprendizagem virtuais em ambientes reais, o que possibilita o aumento de alternativas de ferramentas educacionais.

Contudo, os sistemas distribuídos de Realidade Virtual e Realidade Aumentada desenvolvidos atualmente estão limitados na quantidade de objetos virtuais pertencentes ao ambiente utilizado, outra questão é que nestes sistemas ora utilizam Realidade Virtual ou Aumentada para resolver seus problemas de forma independente, não com completando as duas tecnologias.

Assim, essa dissertação tem a proposta de apresentar uma arquitetura de distribuição de Realidade Virtual e Aumentada, onde qualquer modelo ou ambiente virtual poderá ser distribuído em ambas, o que permite maior flexibilidade da arquitetura proposta.

ABSTRACT

NOGUEIRA, Keila F C. Desenvolvimento de uma arquitetura de distribuição de Realidade Virtual e Aumentada aplicada em Ambientes Educacionais, 2010. 92P. Dissertação (Mestrado em Ciências) Programa de Pós-Graduação em Engenharia Elétrica, UFU, 2010.

Key-words: Virtual Reality, Augmented Reality, Distributed Systems, Learning

Currently, access to information is increasingly easy because of the advancement of Information and Communication Technology (TIC), reflecting the popularization of computers and the Internet, both in schools as at home.

In the same context research projects are being developed toward new teaching methodologies. These projects facilitate new educational methods compared with traditional methods. Today is possible to simulate several laboratories (physics, chemistry, mathematics) by a single computer lab.

There are several technologies that support the creation of these systems, included Virtual Reality that lets you create Virtual models and environments and simulate them as if they were real. Another technology is Augmented Reality, where you can insert elements of virtual learning in real environments, which increases the number of alternative educational tools.

However distributed systems Virtual Reality and Augmented Reality developed today are limited in the amount of virtual objects belonging to the environment used, another issue is that these systems use either Virtual or Augmented Reality to solve their problems independently, not completing the two technologies.

Thus, this dissertation came with the proposal to develop a distribution architecture for Virtual and Augmented Reality, where any model or virtual environment may be distributed, which allows greater flexibility of the proposed architecture

Lista de Publicações

A seguir são apresentadas as publicações no período de realização deste trabalho:

CHAGAS, K. F.; Nogueira, K.L; Lamounier, E. ; Cardoso, A.; Proposta de uma arquitetura de distribuição de Realidade Virtual e Aumentada em Ambientes Educacionais. In: Symposium on Virtual and Augmented Reality, 2010, Natal. Prêmio de Melhor Short Paper da Conferência.

CHAGAS, K. F.; Nogueira, K.L; Cardoso, A.; Lamounier, E. . UMA ARQUITETURA PARA A DISTRIBUIÇÃO DE REALIDADE VIRTUAL E AUMENTADA APLICADA NA EDUCAÇÃO. In: Simpósio Brasileiro de Informática na Educação, 2009, Florianópolis. Uma Arquitetura para a Distribuição de Realidade Virtual e Aumentada aplicada na Educação, 2009.

CHAGAS, K. F.; Nogueira, K.L; Lamounier, E. ; Cardoso, A.. ARQUITETURA DE DISTRIBUIÇÃO DE REALIDADE VIRTUAL E AUMENTADA. In: WRVA - 6º Workshop de Realidade Virtual e Aumentada, 2009, Santos. ARQUITETURA DE DISTRIBUIÇÃO DE REALIDADE VIRTUAL E AUMENTADA, 2009.

CHAGAS, K. F.; Cardoso, A.; Lamounier, E. ; Nogueira, K.L. UM SISTEMA DE COMUNICAÇÃO SÍNCRONA COM SUPORTE A REALIDADE VIRTUAL E AUMENTADA. In: CEEL- Conferência em Estudos de Engenharia Elétrica, 2007, Uberlândia. UM SISTEMA DE COMUNICAÇÃO SÍNCRONA COM SUPORTE A REALIDADE VIRTUAL E AUMENTADA, 2007.

SUMÁRIO

CAPÍTULO 1	17
1 INTRODUÇÃO.....	17
1.1 MOTIVAÇÃO	18
1.2 OBJETIVOS.....	20
1.3 ORGANIZAÇÃO DA DISSERTAÇÃO	21
CAPÍTULO 2	22
2 FUNDAMENTOS TECNOLÓGICOS SISTEMAS DISTRIBUÍDOS	22
2.1 INTRODUÇÃO	22
2.2 MODELOS DE COMPUTAÇÃO DISTRIBUÍDA	24
2.2.1 <i>Cliente- Servidor</i>	24
2.2.2 <i>Peer-to-peer</i>	26
2.3 ARQUITETURA TCP/IP	27
2.4 PROTOCOLOS DE COMUNICAÇÃO TCP E UDP	29
2.5 TECNOLOGIAS DE DISTRIBUIÇÃO	30
2.5.1 <i>Socket</i>	30
2.5.2 <i>RMI e RPC</i>	30
2.5.3 <i>Comparações entre as tecnologias abordadas</i>	31
2.6 CONSIDERAÇÕES FINAIS	32
CAPÍTULO 3	33
3 FUNDAMENTOS TECNOLÓGICOS DE REALIDADE VIRTUAL E AUMENTADA 33	
3.1 INTRODUÇÃO	33
3.2 REALIDADE VIRTUAL	33
3.3 REALIDADE AUMENTADA.....	34
3.3.1 <i>ARToolKit</i>	36
3.4 AMBIENTES VIRTUAIS DISTRIBUÍDOS	38
3.4.1 <i>Sistema de Ambiente Virtual Multiusuário Centralizado</i>	38
3.4.2 <i>Sistema de Ambientes Virtuais Multiusuários Distribuídos</i>	39
3.5 CONSIDERAÇÕES FINAIS	41

CAPÍTULO 4	42
4 TRABALHOS RELACIONADOS	42
4.1 INTRODUÇÃO	42
4.2 SISTEMAS DE REALIDADE VIRTUAL DISTRIBUÍDO	43
4.2.1 DIVE.....	43
4.2.2 Arquitetura para Distribuição de AV Multidisciplinares de Ensino.	43
4.2.3 Ambientes distribuídos em Realidade Virtual com suporte a Aprendizagem Cooperativa para a Resolução de Problemas.	44
4.3 SISTEMAS DE REALIDADE AUMENTADA DISTRIBUÍDOS.....	45
4.3.1 Arquitetura para Distribuição e Colaboração em Ambientes Virtuais de Realidade Aumentada.....	45
4.3.2 NetArtoolkit	46
4.3.3 SACRA	47
4.4 CONSIDERAÇÕES FINAIS	48
CAPÍTULO 5	51
5 ESPECIFICAÇÃO DO SISTEMA	51
5.1 INTRODUÇÃO	51
5.2 ARQUITETURA DO SISTEMA	51
5.3 INTERFACE GRÁFICA COM USUÁRIO – GRAPHICAL USER INTERFACE (GUI).....	54
5.3.1 Recursos	55
5.4 CONSIDERAÇÕES FINAIS	60
CAPÍTULO 6	61
6 DETALHES DE IMPLEMENTAÇÃO	61
6.1 INTRODUÇÃO	61
6.2 ESTÁGIOS DESENVOLVIDOS.....	62
6.2.1 Protocolo de envio de dados.....	62
6.2.2 Servidor	63
6.2.3 Solicitando a abertura de objetos existentes – Cliente.....	66
6.2.4 Novo objeto - Cliente.....	67
6.2.5 Mensagem de texto- Cliente	68
6.2.6 Distribuidor de Eventos.....	69
6.3 ESTUDOS DE CASO	69

6.3.1	<i>Coração Virtual</i>	69
6.3.2	<i>Motor Virtual</i>	71
6.4	CONSIDERAÇÕES FINAIS	72
CAPÍTULO 7		73
7	AVALIAÇÕES E RESULTADOS	73
7.1	INTRODUÇÃO	73
7.2	VERIFICAÇÃO E VALIDAÇÃO DO SOFTWARE	73
7.3	AMBIENTE EXPERIMENTAL.....	74
7.4	RESULTADOS DA ANÁLISE OBTIDA.....	74
7.4.1	<i>Latência de Comunicação</i>	74
7.4.2	<i>Escalabilidade</i>	76
7.5	AVALIAÇÃO DO SISTEMA.....	76
7.5.1	<i>Funcionalidade</i>	77
7.5.2	<i>Usabilidade</i>	78
7.5.3	<i>Confiabilidade</i>	79
7.5.4	<i>Eficiência</i>	80
7.5.5	<i>Portabilidade</i>	81
7.6	CONSIDERAÇÕES FINAIS SOBRE O PROTÓTIPO RVANET	82
CAPÍTULO 8		84
8	CONCLUSÕES E TRABALHOS FUTUROS	84
8.1	CONCLUSÕES	84
8.2	TRABALHOS FUTUROS	85
8.3	CONSIDERAÇÕES FINAIS	86
ANEXO I.....		93

ÍNDICE TABELAS

Tabela 2.1 - Comparação Socket, RMI e RMC	31
Tabela 4.1- Comparação dos trabalhos relacionados ao projeto.....	49
Tabela 6.1 - Sintaxe do protocolo de envio.....	62
Tabela 8.1- Comparativa das características relevantes as a pesquisa	84

ÍNDICE DE FIGURAS

Figura 2.1- Processos Cliente Servidor	25
Figura 2.2- Cliente Servidor	25
Figura 2.3- Modelo Peer-to-Peer	26
Figura 2.4 - Arquitetura TCP/IP	27
Figura 2.5 - Portas de Comunicação.....	28
Figura 3.1- (a) Usuário em laboratório usando capacete HMD (b) Ambiente virtual no qual o usuário está imerso.....	33
Figura 3.2 - (a) colocando um marcador em um sistema e (b) visualizando a malha de elementos finitos para o estudo particular de análises de temperatura (Weidlich. et. al. 2008)..	34
Figura 3.3 - Diagrama de Realidade/Virtualidade Continua (Milgram, ET al., 1994).....	35
Figura 3.4 –(a) Imagem Real (Input Vídeo) e (b) Imagem Binária Capturada	36
Figura 3.5 - Objetos Virtuais sobre o marcador.....	37
Figura 3.6 - Diagrama de funcionamento do ARToolkit adaptada de (Kato, 2000).....	37
Figura 3.7 - Modelos de Sistema de AV Multiusuário Centralizado (Kirner, 1996).	39
Figura 3.8 - Modelos de Sistema de AV Multiusuário Distribuído (Kirner, 1996).....	39
Figura 3.9 - Acesso ao ambiente virtual replicado (Kirner, 1996)	40
Figura 3.10 - Acesso ao ambiente virtual particionado (Kirner, 1996).....	40
Figura 4.1- Áreas Relacionadas – Adaptada de (Silva, 2008)	42
Figura 4.2 - Sistema de RV (Dive, 2008).....	43
Figura 4.3 - Ambiente Virtual de Biologia e Química (Ribeiro, 2006)	44
Figura 4.4 - Sistema LVD Silva (2008).....	45
Figura 4.5 - RA Distribuído (Silva, 2008)	46
Figura 4.6 - Objeto Virtual após o envio do comando set_rotation (Oliveira, 2008).....	47
Figura 4.7 - Exemplo dos marcadores de ações e referência (Santin, 2008)	48
Figura 5.1 - Arquitetura do Sistema.....	51
Figura 5.2 – Módulo Cliente.....	52
Figura 5.3 - Fluxo da Informação.....	53

Figura 5.4 - Arquitetura Cliente/Servidor	54
Figura 5.5 - Inicializar Servidor.....	55
Figura 5.6 - Interface Cliente	56
Figura 5.7 - Editor VRML.....	57
Figura 5.8 - Interface RA	58
Figura 5.9 - Download Modelos VRML	58
Figura 5.10 - Novo Objeto.....	59
Figura 5.11 - Novo Objeto Biblioteca	59
Figura 6.1 - Código Identificador de eventos 1.....	63
Figura 6.2 - Código Identificador de eventos 2.....	64
Figura 6.3 - Código Identificador de eventos 3.....	65
Figura 6.4 - Código enviar objeto existente.....	66
Figura 6.5 - Atualização de objeto existente	67
Figura 6.6 - Código Novo objeto	68
Figura 6.7 - Código Mensagem de Texto	68
Figura 6.8 - Código Distribuidor de Eventos	69
Figura 6.9 - Coração em RV	70
Figura 6.10 - Coração em RV e RA.....	70
Figura 6.11 - Motor em RV.....	71
Figura 6.12 - Motor em RA.....	72
Figura 7.1 - Latência de Comunicação	75
Figura 7.2 - Escalabilidade	76
Figura 7.3 – Funcionalidade	77
Figura 7.4 – Usabilidade.....	78
Figura 7.5 - Confiabilidade	79
Figura 7.6 - Eficiência	80
Figura 7.7 – Portabilidade	81
Figura 7.8 - Dados Gerais Norma ISO 9126.....	82
Figura 7.9 - Avaliações Finais	83

LISTA DE ABREVIATURAS

AV- Ambientes Virtuais

AVD - Ambientes Virtuais Distribuidos

API - Application Programming Interface

ARToolKit - Augmented Reality Tool Kit

CORBA- Commom Object Broker Architeture

IDL - Interface Definition Language

OpenGL - Open Graphics Library

ORB - Object Request Broker

ProInfo - Programa Nacional de Informática na Educação

RA - Realidade Aumentada

RNA - Redes Neurais Artificiais

RV - Realidade Virtual

RMI - Remote Method Invocation

RMC - Remote Procedure Call

SD – Sistemas Distribuídos

SEED- Secretaria de Educação a Distância

SL – Second Life

TICs- Tecnologias de Informática e Comunicações

USP - Universidade de São Paulo

UEL - Universidade Estadual de Londrina

UFU - Universidade Federal de Uberlândia

UNESCO - United Nations Educational, Scientific and Cultural Organization

VRML - Virtual Reality Modeling Language

Capítulo 1

1 Introdução

Anualmente, a UNESCO lança um relatório de Monitoramento de Educação, Ciência e Cultura (UNESCO). A qualidade da educação no Brasil, segundo este relatório é baixa, principalmente no ensino básico. O relatório lançando dia 19 de Janeiro de 2010 aponta que, apesar da melhora apresentada entre 1999 e 2007, o índice de repetência no ensino fundamental brasileiro (18,7%) é o mais elevado na América Latina e fica expressivamente acima da média mundial (2,9%). O alto índice de abandono nos primeiros anos de educação também alimenta a fragilidade do sistema educacional do Brasil. Cerca de 13,8% dos brasileiros largam os estudos já no primeiro ano no ensino básico. Neste quesito, o País só fica à frente da Nicarágua (26,2%) na América Latina e, mais uma vez, bem acima da média mundial (2,2%) (UNESCO, 2010).

Na avaliação da UNESCO, o Brasil poderia se encontrar em uma situação melhor se não fosse à baixa qualidade do seu ensino. Das quatro metas quantificáveis usadas pela organização, o País registra altos índices em três (atendimento universal, igualdade de gênero e analfabetismo), mas um indicador muito baixo no percentual de crianças que ultrapassa o 5º ano. Problemas que a educação brasileira ainda enfrenta: a estrutura física precária das escolas e o número baixo de horas em sala de aula são apontados pelos técnicos da UNESCO como fatores determinantes para a avaliação da qualidade do ensino (UNESCO, 2010).

Para tentar resolver os problemas citados acima, o Ministério da Educação implantou o ProInfo (Programa Nacional de Informática na Educação). O ProInfo é desenvolvido pela Secretaria de Educação a Distância (SEED), por meio do Departamento de Infra-Estrutura Tecnológica (DITEC), em parceria com as Secretarias de Educação Estaduais e Municipais.

O ProInfo é um programa educacional criado para promover o uso pedagógico das tecnologias de informática e comunicações (TICs) na rede pública de ensino fundamental e médio apoiando ao processo de ensino-aprendizagem (ProInfo, 2010).

1.1 Motivação

Através do ProInfo poucos são os centros educacionais do país que não possuem recursos computacionais. As tecnologias computacionais aplicáveis à educação são normalmente enquadradas em quatro categorias (Fernando, 2008), apresentadas abaixo com os respectivos exemplos:

- ✓ tutorial: ensinam através de demonstrações e simulações em seqüências predefinidas pelo sistema, cujos exemplos mais comuns são: sistemas multimídia em CD-ROM;
- ✓ exploratório: facilitam a aprendizagem ao fornecerem informações, demonstrações ou simulações quando requeridas pelo estudante. Exemplos relevantes são: web, enciclopédias multimídia em CD-ROM;
- ✓ aplicativos: usados para edição de texto e figuras, e análise de dados. Os exemplos mais comuns são: processadores de texto, planilhas, gerenciadores de bancos de dados e sistemas de gravação/edição de vídeo;
- ✓ comunicação: conjuntos de software e hardware usados para intercomunicação em redes locais, acesso à Internet e seus serviços, correio eletrônico, chats, etc.

Uma quinta categoria que se pode destacar é o uso de Ambientes Virtuais. Pantelidis (Shulman, 1999), co-diretora do Virtual Reality and Education Laboratory da East Carolina University em Greenville, North Carolina, e editora da VR in the Schools, descreve algumas situações em que a RV pode ser muito útil na sala de aula. Primeiro, a RV deveria ser usada para ensinar assuntos nos quais a forma tradicional pode ser perigosa (estudar a estrutura de uma tempestade), prejudicial ao ambiente (simular um desastre químico em um rio) ou cara (um mergulho subaquático). Segundo, as aplicações de RV fazem sentido por permitirem que os estudantes aprendam em ambientes que não podem ser experimentados. Como visitar Marte, viajar dentro do corpo humano, ou mover-se entre as moléculas de um composto químico. Pantelidis afirma, por fim, que os projetistas devem focar as aplicações de RV para a visualização de um conjunto de informações ou conceitos para que eles possam ser mais facilmente compreendidos.

Segundo Rodello et.al (2002), apesar da teoria sobre RV não ser muito nova, somente nos últimos anos esta vem apresentando uma grande evolução, onde suas implementações tecnológicas já estão sendo utilizadas por diversas áreas do conhecimento, como por exemplo: medicina, arquitetura, educação, astronomia, simulação, entre outros.

Ambientes Virtuais Distribuídos (AVDs) vêm crescendo e apresentando um elevado potencial de aplicação (Kirner, 2004). Eles são caracterizados por cenários tridimensionais com alto índice de interação onde usuários, dispersos geograficamente, podem cooperar e compartilhar dados e informações em tempo real. Para tal, é necessário o suporte de rede de computadores, visando à melhoria do desempenho coletivo (Benford, 1994). São encontradas algumas limitações nos sistemas atuais:

- ✓ Distribuição somente em rede locais: ambientes virtuais distribuídos em rede local implica na utilização somente de computadores agrupados em um limitado espaço como, por exemplo, um laboratório. Isto limita o uso de recursos de ensino, via internet, com maior alcance populacional.
- ✓ Distribuição simultânea de RV e RA: não contemplando as duas tecnologias ao mesmo tempo (Silva, 2008). A vantagem de utilizar as duas tecnologias é o melhor aproveitamento ou entendimento de um modelo virtual, como por exemplo, em modelos atômicos deixam de ser apenas descritos ou ilustrados pelos professores através da lousa. A interatividade efetiva das técnicas de RV e RA propiciam ao estudante acesso, facilitando a compreensão em nível teórico. Em outras palavras, um sistema de RV e RA permite uma melhor representação (visualização) dos conceitos científicos (teorias, modelos atômicos) e, dessa forma, propicia melhores condições de aprendizagem. (ROGADO, 2008)

Este trabalho propõe uma arquitetura flexível, tanto para o uso de ambientes RV e RA distribuídos e promover a distribuição não só em Redes Locais (LAN), como também na Internet, a fim de contribuir para o aumento do uso desta tecnologia.

Uma das motivações para o desenvolvimento de ambientes virtuais de RV e RA distribuídos na Web está na possibilidade de transformar o mesmo em um lugar social onde uma comunidade de usuários possa compartilhar e interagir com essa informação (Kirner, 2004). Sendo que a Educação deve ser um processo de construção de

conhecimento ao qual ocorrem em condição de complementaridade: de um lado, os alunos e professores e, do outro, os problemas sociais atuais e o conhecimento já construído (Becker, 2002). Tem-se, como exemplo, o uso do software Second Life, que é um ambiente distribuído capaz de suportar comunidades virtuais.

O fundamento dos ambientes está em incentivar cada usuário, conhecido como um avatar, a encontrar um meio de sobreviver, aprendendo e desenvolvendo atividades lucrativas, as quais irão refletir diretamente em seu poder aquisitivo dentro do jogo.

Vários filmes mostram a tendência científica dos avatares em ambientes virtuais, como por exemplo, os filmes: Avatar, Os Substitutos e Game (Cine Pop, 2010) e (Cine Menu, 2010).

Entretanto, se comparado com sistemas de jogos distribuído os sistemas de RV e RA hoje estão muito aquém, pois os jogos possuem uma escalabilidade de usuários bem superior a estes, além de interface gráfica rica em detalhes, não comprometendo a performance em tempo real. Pode-se citar, por exemplo, o jogo World of Warcraft que possui mais de 11 milhões de usuários (World of Warcraft, 2010). Contudo, estes jogos não têm a flexibilidade de permitir que o usuário altere as regras e o ambiente do jogo e também não permite inserir elementos externos, como um novo objeto virtual ou o uso de RA, o que possibilitaria uma faixa de aplicação e investigação muito mais ampla do sistema.

1.2 Objetivos

Nesse sentido, o objetivo desse projeto é investigar uma arquitetura que suporte simultaneamente a distribuição de RV e RA, com isso acredita-se que haja uma melhor entendimento de um modelo virtual. Ainda poderá inserir novos modelos virtuais na biblioteca comum e alterar uma quantidade maior de parâmetros do modelo virtual. Neste sentido, justifica-se uma proposta de uma interface de RV e RA que possua um nível favorável de interação. As contribuições deste trabalho encontram-se mediante a interseção das quatro áreas de pesquisa e conhecimento: Realidade Virtual e Realidade Aumentada, Sistemas Distribuídos e Aprendizagem.

1.3 Organização da Dissertação

A presente dissertação consiste de 8 capítulos, dentre os quais a introdução, descritos a seguir.

O Capítulo 1 apresenta a motivação os objetivos e a estrutura do trabalho. O Capítulo 2 e 3 apresentam uma visão geral sobre conceitos de Sistemas Distribuídos, Modelos de Computação Distribuída, Tecnologias de Distribuição, Realidade Virtual e Realidade Aumentada. O Capítulo 4 apresenta o estado da arte e mostra alguns sistemas distribuídos de Realidade Virtual e Realidade Aumentada, suas estruturas, suas abrangências, potencialidades e limitações.

A descrição do funcionamento do sistema concebido e sua respectiva arquitetura são apresentadas no capítulo 5. O Capítulo 6 apresenta os detalhes de implementação dos algoritmos utilizados. As avaliações e os resultados do sistema são apresentados no capítulo 7. O Capítulo 8 apresenta as considerações finais, conclusões e as sugestões para trabalhos futuros.

2 Fundamentos Tecnológicos Sistemas Distribuídos

2.1 Introdução

Este capítulo aborda os conceitos fundamentais de Sistemas Distribuídos, destacando, principalmente, modelos de computação distribuída e tecnologias de distribuição.

De acordo com Tanenbaum (2003), um sistema distribuído é uma coleção de processadores acoplados e interconectados por uma rede de comunicação que se apresenta ao usuário como um sistema único e consistente. Para Coulouris (2001), um sistema distribuído é uma coleção de computadores autônomos interligados através de uma rede de computadores e equipados com software que permita o compartilhamento dos recursos do sistema: hardware, software e dados.

Umas das principais características que um sistema distribuído deve ter é a heterogeneidade, que consiste na capacidade do sistema operar nas diferentes variedades de redes, sistemas operacionais, hardware e linguagens de programação (Coulouris, 2001). Isso é um grande desafio, mas diversos outros são encontrados:

- ✓ **Heterogeneidade:** Sistemas distribuídos devem permitir acesso a um conjunto heterogêneo de computadores e rede, se aplica a: Redes, Hardware, Sistemas Operacionais e Linguagens de programação.
- ✓ **Segurança:** Consiste em proteger de forma adequada os recursos compartilhados e manter em segredo as informações que são transmitidas em mensagens pela rede. A segurança é baseada em três componentes: confidencialidade, integridade e disponibilidade.
- ✓ **Escalabilidade:** Um sistema é dito escalável se permanecer eficiente, quando há um aumento significativo de usuários ou de recursos. A Internet é um exemplo

claro de um sistema distribuído, com alto índice de escalabilidade, pois possui um número significativo de computadores e um crescente aumento dos recursos.

- ✓ **Latência:** Do ponto de vista do ambiente virtual distribuído, é a latência que controla a natureza interativa e dinâmica do sistema. Determina se o sistema pode ser estendido e re-implementado em várias formas, considerando grandes desafios impostos na integração de diferentes componentes e existentes nos diversos tipos de hardware e software.
- ✓ **Tratamento de falhas:** possibilita que o sistema tenha capacidade de operar mesmo na presença de falhas. Para isso, cada componente do sistema precisa reconhecer possíveis falhas e suas conseqüências, para atuarem de forma apropriada. O tratamento de falhas deve ser parcial, ou seja, alguns componentes falham e outros continuam funcionando.
- ✓ **Concorrência:** consiste na capacidade do sistema atender a requisições de diversos usuários, procurando acessar recursos compartilhados ao mesmo tempo. Para isso, cada recurso deve ser projetado para operar de modo seguro em um ambiente concorrente.
- ✓ **Transparência:** consiste no fator que possibilita obter a imagem de um único sistema, escondendo toda a distribuição do usuário, até mesmo de alguns programas. O conceito de transparência pode ser aplicado para vários aspectos de um sistema distribuído.
 - **Localização:** o usuário desconhece a localização dos recursos.
 - **Migração:** os recursos podem mover-se sem modificar seus nomes.
 - **Replicação:** os usuários desconhecem a quantidade de cópias existentes.
 - **Concorrência:** múltiplos usuários podem compartilhar recursos automaticamente.
 - **Paralelismo:** atividades que podem ocorrer em paralelo, sem o conhecimento do usuário.

Os desenvolvedores de sistemas distribuídos devem estar atentos a essas propriedades durante o desenvolvimento do projeto.

Vale ressaltar que toda e qualquer função disponível em um computador desktop é funcional em uma rede distribuída. Tais funções incluem transferência de arquivos, conexão, correio, navegação *Web* e chamadas de procedimentos remotos.

Segundo Silberschatz (2004) a vantagem de um sistema distribuído é que essas funções podem ser executadas sobre grandes distâncias. Computadores distantes geograficamente podem colaborar em uma rede utilizando as mais diversas operações. Com isso, as redes de sistemas distribuídos têm ampla tendência, pois o tamanho dos computadores pode ser reduzido tendo em vista que mainframes podem ser substituídos por estações de trabalho ou computadores pessoais, que têm melhor relação custo-benefício, com interfaces melhores para os usuários e fácil manutenção.

2.2 Modelos de Computação Distribuída

Um aspecto relevante no projeto do sistema distribuído é o modelo arquitetural a ser adotado, pois é a estrutura desse modelo, que poderá garantir o seu funcionamento perante as exigências requisitadas tanto no presente, quanto as que serão demandadas no futuro.

O modelo arquitetural define a maneira que os componentes dos sistemas interagem entre si, além da forma como são mapeados em uma rede de computadores. As maiores preocupações da estrutura da arquitetura de um sistema consistem em torná-lo confiável, gerenciável, adaptável e a custos efetivos. A seguir, serão apresentados dois modelos arquiteturais de Sistemas Distribuídos (Coulouris, 2001): o modelo *cliente-servidor* e o *peer-to-peer*.

2.2.1 Cliente- Servidor

O modelo cliente-servidor é uma tecnologia computacional que separa clientes e servidores, sendo interligados entre si utilizando uma rede de computadores (Coulouris, 2001).

Servidores são programas que respondem as solicitações por serviços compartilhados; ele é um processo reativo, disparado pela chegada de pedidos de seus clientes. Geralmente, o processo servidor roda o tempo todo, oferecendo serviços a

muitos clientes. Cada instância de um cliente pode enviar requisições de dado para algum dos servidores conectados e esperar pela resposta. Por sua vez, algum dos servidores disponíveis pode aceitar tais requisições, processá-las e retornar o resultado para o cliente. O modelo cliente-servidor possui grande importância no cenário dos sistemas distribuídos e apesar de o conceito ser aplicado em diversos usos e aplicações, a arquitetura é praticamente a mesma, como pode ser observado na Figura 2.1.

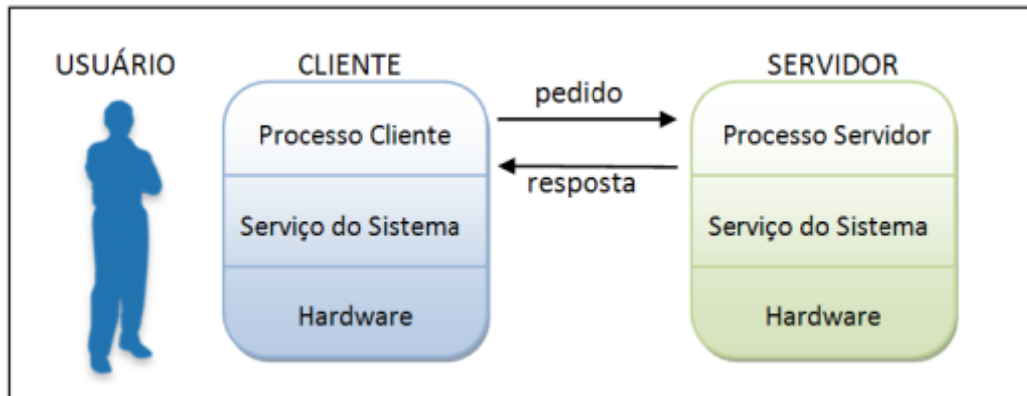


Figura 2.1- Processos Cliente Servidor

Existem vários exemplos desse modelo, na Web como serviços de Internet (*FTP*, e-mails e outros) (Coulouris, 2001). A Figura 2.2 mostra um exemplo do modelo cliente-servidor.

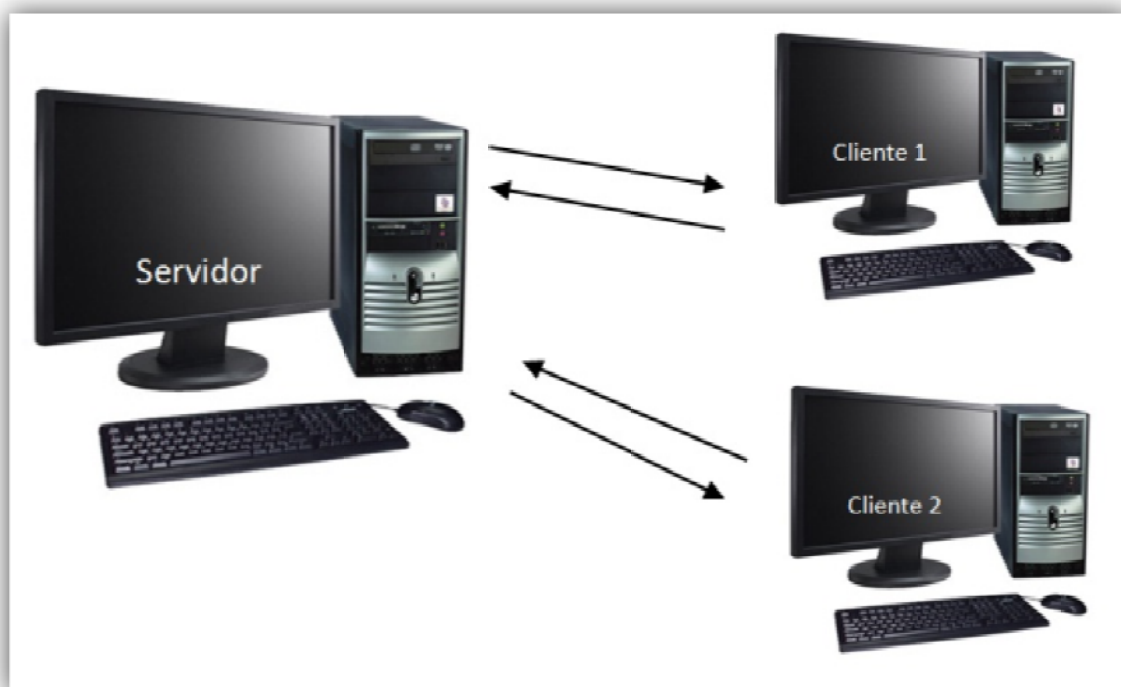


Figura 2.2- Cliente Servidor

2.2.2 Peer-to-peer

O P2P ou Peer-to-Peer (Par-a-Par) é uma tecnologia para estabelecer uma espécie de rede de computadores virtual, onde cada estação possui capacidades e responsabilidades equivalentes. Difere da arquitetura cliente/servidor, no qual alguns computadores são dedicados a servirem dados a outros.

No modelo peer-to-peer, todos os processos desempenham papel semelhante, interagindo de forma cooperativa, como peers para executar tarefas distribuídas, sem qualquer distinção entre clientes e servidores. A eliminação do processo servidor reduz o atraso da comunicação interprocessos para acesso aos objetos locais (Coulouris, 2001). A Figura 2.3 demonstra uma aplicação distribuída baseada no modelo peer-to-peer. No Capítulo 6 será discutido estas tecnologias, justificando sua escolha da tecnologia utilizada.

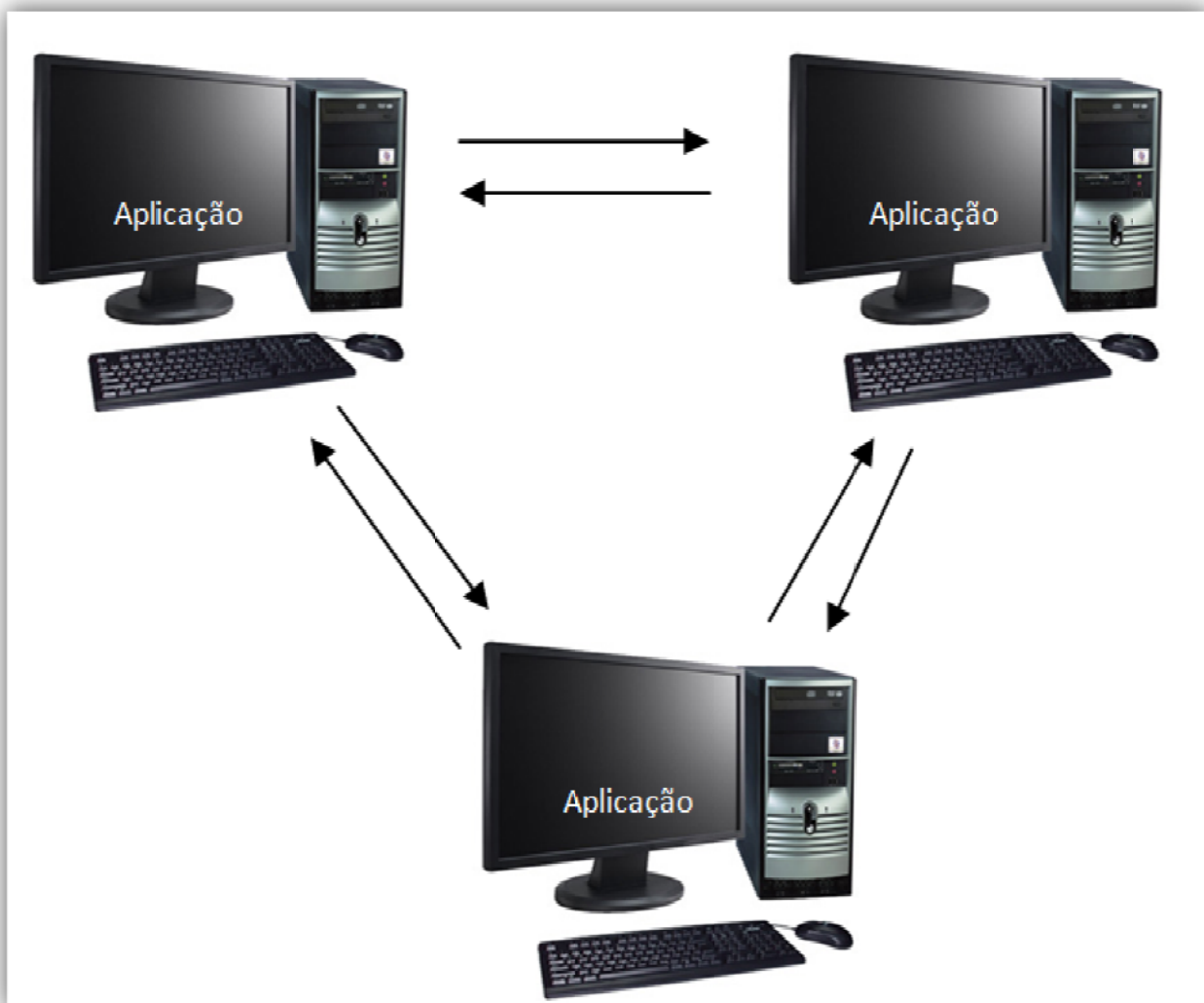


Figura 2.3- Modelo Peer-to-Peer

2.3 Arquitetura TCP/IP

Segundo Tanenbaum (2003), a arquitetura Internet *TCP/IP* (*Transmission Control Protocol/Internet Protocol*) é organizada em quatro camadas conceituais, como mostra a Figura 2.4.

Alguns autores, como por exemplo, Stallings (2007), trazem cinco camadas, pois fazem a correspondência da arquitetura do *TCP/IP* com o modelo *OSI* ¹chamado de modelo híbrido.



Figura 2.4 - Arquitetura TCP/IP

A camada “Enlace” transforma um canal de transmissão bruto em uma linha de transmissão. A camada de “Rede” é responsável pelo envio de pacotes de um computador a outro, independente de sua localização na rede. Esta camada realiza a comunicação entre máquinas vizinhas através do protocolo *IP*. Para identificar cada máquina e a própria rede onde estas estão situadas, é definido um identificador, chamado endereço *IP*. A camada de “Transporte” define as regras para a entrega dos pacotes de dados, podendo ser basicamente *TCP*, que garante a sobrecarga da rede e conversas apenas ponto a ponto, e *UDP* que não realiza a confirmação da entrega dos dados, com a

¹ O modelo OSI explica o funcionamento da rede, dividindo-a em 7 camadas. Segundo Tanenbaum (2003) o modelo é chamado de **Modelo Referencia ISO OSI** (*Open Systems Interconnection*), pois ele trata de inter conexão de sistemas abertos, ou seja, sistemas que está aberto à comunicação com outros sistemas.

vantagem de ser mais rápido e permitir conversas ponto a ponto ou pontos a vários pontos.

Os protocolos da camada de “Transporte” utilizam o conceito de Portas para identificar as aplicações. Cada processo de aplicação é identificado por uma porta diferente, que é o endereço de transporte (número inteiro de 16 bits). A Figura 2.5 representa o que foi mencionado.

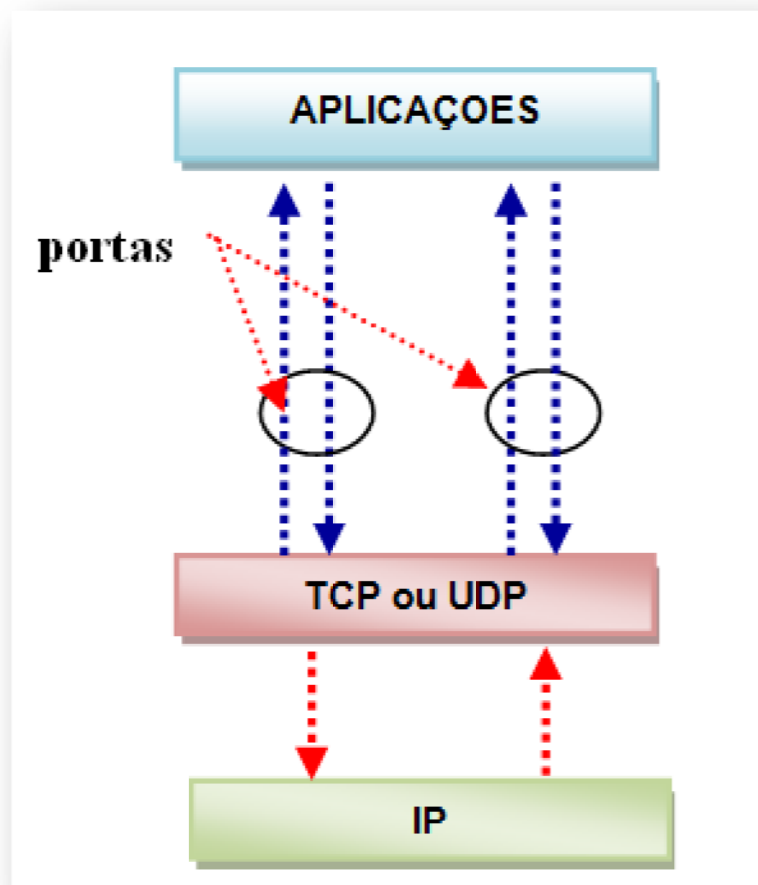


Figura 2.5 - Portas de Comunicação

Os números das portas obedecem a uma padronização. Portas reservadas para serviços do sistema vai até 1023 como, por exemplo, *FTP File Transfer Protocol* (Protocolo de Transferência de Arquivos) utiliza portas 20 e 21, *Telnet*²: 23 , *SSH*³: 22,

² *Telnet*: é um protocolo cliente-servidor usado para permitir a comunicação entre computadores ligados numa rede baseado em TCP.

*SMTP*⁴: 25, *POP3*⁵:110 e *HTTP Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertexto): 80. Já para qualquer aplicação utiliza porta efêmeras de 1024 até 65535, desde que estejam livres.

A camada “Aplicação” reúne os protocolos que fornecem serviços de comunicação ao sistema ou ao usuário.

2.4 Protocolos de Comunicação TCP e UDP

A troca de mensagens entre processos de sistemas distribuídos é gerenciada por protocolos de comunicação, oferecendo um conjunto de regras e formatos para a comunicação entre diferentes processos.

Existem diversos protocolos, sendo os mais conhecidos o *Transmission Control Protocol* (TCP) e o *User Datagram Protocol* (UDP), que são protocolos do nível de transporte. O *TCP* consiste num protocolo confiável, pois oferece garantia de entrega das mensagens. Já o *UDP* consiste num protocolo não confiável, visto que não garante a entrega de mensagens, não ordena mensagens de entrada e não fornece informações para o controle da velocidade com que as informações são transmitidas entre as máquinas (Coulouris, 2001) e (Commer, 1998).

Um ponto importante, na comparação desses protocolos, consiste na possibilidade de aplicações, que utilizam o protocolo confiável, o *TCP*, executarem dez vezes mais lento que aplicações, utilizando o *UDP* (Gossweiler, 1994).

A única vantagem da utilização do *UDP* é a velocidade, porém ele não garante a sequência de dados enviadas, por isso foi escolhido o protocolo *TCP* para desenvolvimento do protótipo.

Os protocolos *TCP* e *UDP* suportam comunicação interprocessos, podendo ser realizada através da passagem de mensagens via *Socket*. Esta tecnologia é explanada na próxima seção.

³ *SSH*: um programa de computador e um protocolo de rede que permite a conexão com outro computador na rede, de forma a executar comandos de uma unidade remota. Possui as mesmas funcionalidades do *TELNET*, com a vantagem da conexão entre o cliente e o servidor ser criptografada.

⁴ *SMTP*: é o protocolo padrão para envio de e-mails através da Internet.

⁵ *POP3*: é um protocolo utilizado no acesso remoto a uma caixa de correio eletrônico

2.5 Tecnologias de Distribuição

Para justificar a escolha de uma tecnologia de distribuição serão abordadas algumas destas, ressaltando testes comparativos entre as mesmas e resultados.

2.5.1 *Socket*

No contexto de redes de computadores *Socket* quer dizer um “ponto” de comunicação entre dois programas executados na rede Silberschatz (2000). Esse ponto tanto envia quanto recebe dado, possui um porta e um número, para cada camada de protocolo saiba para qual aplicação entregar os dados recebidos. Segundo Silberschatz (2000), um *Socket* é definido como a extremidade da comunicação entre dois processos. Dessa forma, para um processo enviar uma mensagem, o seu *Socket* deve referenciar a porta e o endereço Internet do computador de destino (Coulouris, 2001).

Os *Sockets* viabilizam uma forma de comunicação comum, robusta e eficiente entre processos, mas considerada de baixo nível. Um dos motivos dessa classificação consiste no fato da troca de mensagens, via *Socket*, ser um fluxo não estruturado de *bytes* entre processos. Dessa maneira, a estrutura de dados é definida pela aplicação. Assim, um dos cuidados a serem tomados na sua implementação consiste em possíveis problemas relacionados à heterogeneidade dos sistemas computacionais (Silberschatz, 2000) e (Matthiesen, 2006).

2.5.2 *RMI e RPC*

Devido aos possíveis problemas encontrados em técnicas de comunicação de baixo nível, foram desenvolvidas técnicas que viabilizam a transmissão de representações das estruturas de dados ou valores primitivos. Essas técnicas são utilizadas para realizar invocações de métodos remotos (*Remote Method Invocation*, RMI), ou chamadas de procedimentos remotos (*Remote Procedure Call*, RPC) (Coulouris, 2001).

O *RMI* permite a um objeto invocar um método contido em outro objeto num processo remoto. Dois exemplos de sistemas que possibilitam a invocação de métodos remotos são o *Common Object Broker Architecture* (CORBA) e Java RMI. Esses sistemas permitem instanciar e acessar objetos, através de uma interface orientada a objetos bem definida. O CORBA, por exemplo, fornece a *Interface Definition Language* (IDL), que possibilita implementar interfaces em independentes linguagens de programação,

permitindo a comunicação de aplicações escritas em diferentes linguagens. A invocação de métodos remotos em CORBA é realizada, através do *Object Request Broker* (ORB), responsável em localizar o objeto remoto no sistema distribuído (Coulouris, 2005) (Silverschatz, 2000) (Yang, 1996).

O *RPC* possui características muito similares ao *RMI*, visto que um programa cliente chama um procedimento em outro programa, em execução no servidor. Porém, ao contrário do *RMI*, o *RPC* não realiza a comunicação entre objetos distribuídos. Oferece a programação procedimental possibilitando apenas as chamadas remotas de funções ou procedimentos. Um exemplo de sistema que possibilita chamadas de procedimentos é o *Sun RPC* (Coulouris, 2005) (Silverschatz, 2000).

O *RPC* e *RMI* permitem estabelecer a comunicação de processos distribuídos em alto nível. Dessa maneira, aplicações que implementam essas tecnologias podem oferecer serviços avançados, como a interoperabilidade entre aplicações escritas em diferentes linguagens.

2.5.3 Comparações entre as tecnologias abordadas

Os benefícios obtidos com as abstrações dessas tecnologias acabam, entretanto, acarretando custos no desempenho. Um estudo realizado em (Tilley, 1999) comparou o desempenho na transmissão de *callbacks* entre uma *GUI* cliente e uma aplicação servidora, utilizando *Socket* em C++, *Socket* em Java, *RMI* do Java e *RMI* do CORBA. Os resultados dos testes comprovaram que a transmissão de mensagens utilizando *Socket* em C++ foi cerca de quatro vezes mais rápida, em relação ao *RMI* do Java e do CORBA, e cerca de duas vezes mais rápida do que o *Socket* em Java.

Outro estudo comparativo feito por Tramontina (2002) mostrou que *Socket* foi mais rápido comparado com as outras tecnologias em relação ao tempo; os dados transmitidos foram os mesmos, como pode ser visto na Tabela 2.1.

Tabela 2.1 - Comparação Socket, RMI e RMC

Tecnologia	Socket	RMI	RMC
Tempo gasto	0,0027 ms	0,37 ms	1,5 ms

O procedimento era iniciar o servidor, iniciar o cliente e fazer com que este último requisitasse ao primeiro à realização de 100.000 operações.

Em um mecanismo muito simples, usando chamadas diretas do sistema operacional, o *Socket* saiu melhor em tempo de execução.

2.6 Considerações Finais

Neste capítulo, foram apresentadas tecnologias de distribuição de ambientes virtuais, destacando-se os conceitos relevantes à pesquisa, à tecnologia de distribuição usando Sockets.

No próximo capítulo, serão mostrados conceitos sobre Realidade Virtual e Aumentada fundamentais para o entendimento do projeto.

Capítulo 3

3 Fundamentos Tecnológicos de Realidade Virtual e Aumentada

3.1 Introdução

Este capítulo trata dos conceitos fundamentais de Realidade Virtual e Aumentada necessários à compreensão do projeto proposto.

3.2 Realidade Virtual

Dentre as várias definições dadas à Realidade Virtual (RV), pode-se destacar a seguinte: “Interface avançada para aplicações computacionais, onde o usuário pode navegar e interagir, em tempo real, em um ambiente tridimensional gerado por computador, usando dispositivos multisensoriais” (Cardoso et. al., 2007).

Essencialmente, esta tecnologia permite a criação de um ambiente artificial, no qual o usuário tem a impressão de não somente estar dentro deste ambiente, mas também munido da capacidade de navegar no ambiente e interagir com os seus objetos, podendo até alterar a realidade que o envolve (Lamounier e Cardoso, 2009).

A fim de ilustrar este conceito, note na Figura 3.1a um usuário em um laboratório com um equipamento multisensorial (capacete HMD) e que através de técnicas de Realidade Virtual, é transportado para um ambiente virtual similar ao de uma cozinha (Figura 3.1).



Figura 3.1- (a) Usuário em laboratório usando capacete HMD (b) Ambiente virtual no qual o usuário está imerso (Weidlich. et. al. 2008).

3.3 Realidade Aumentada

Realidade Aumentada (RA) é a inserção de objetos virtuais no ambiente físico, mostrada ao usuário, em tempo real, com o apoio de algum dispositivo tecnológico, usando a interface do ambiente real, adaptada para visualizar e manipular os objetos reais e virtuais (Kirner, 2004).

Pode-se definir Realidade Aumentada (RA) como a amplificação da percepção sensorial por meio de recursos computacionais. Assim, associando dados computacionais ao ambiente real, a Realidade Aumentada permite uma interface mais natural com dados e imagens geradas por computador (Cardoso, 2007). Desta forma, entende-se que um sistema de Realidade Aumentada deve prover ao usuário condições de interagir com estes dados de forma mais natural possível. Esta definição é ilustrada na Figura 3.2, onde um engenheiro pode visualizar as camadas de temperatura numa peça com técnicas de RA (Lamounier e Cardoso, 2009).



Figura 3.2 - (a) colocando um marcador em um sistema e (b) visualizando a malha de elementos finitos para o estudo particular de análises de temperatura (Weidlich. et. al. 2008).

Milgram et. al, (1944) define Realidade Aumentada como a sobreposição de objetos virtuais tridimensionais, gerados por computador, com um ambiente real, por meio de algum dispositivo tecnológico. Entretanto, esta conceituação é muito geral, e fica clara com sua inserção em um contexto mais amplo: o da Realidade Misturada.

De acordo com Kirner (2004), Realidade Misturada, mistura o real com o virtual, abrange duas possibilidades: a Realidade Aumentada, cujo ambiente predominante é o ambiente real, e a Virtualidade Aumentada, cujo ambiente predominante é o ambiente

virtual. Pode-se dizer, então, que a Realidade Aumentada é uma particularização da Realidade Misturada. A Figura 3.3 apresenta o diagrama adaptado de realidade/virtualidade contínua, mostrando as possibilidades gradativas de sobreposição do real com o virtual e vice-versa.



Figura 3.3 - Diagrama de Realidade/Virtualidade Contínua (Milgram, ET al., 1994).

A Realidade Aumentada hoje é aplicada como ferramenta de visualização nas mais variadas áreas, como visualização médica, simulação, jogos, pode ser aplicada também à pesquisa e ainda à educação.

Entende-se que as soluções de Realidade Aumentada envolvem a geração de elementos virtuais que são inseridos no ambiente real, de tal forma que o usuário crê que os mesmos são partes do meio no qual está inserido. Assim, Cardoso (2007) afirma que são características básicas de sistemas de Realidade Aumentada: Processamento em tempo real; Combinação de elementos virtuais com o ambiente real; Uso de elementos virtuais concebidos em 3D.

Desta forma, observa-se que a concepção de soluções de Realidade Aumentada necessita de componentes que permitam avaliar a posição de quem interage, o ponto de vista e gerar os elementos virtuais para, finalmente, combiná-los com o ambiente real por

meio de um sistema de projeção. Para tanto, os elementos reais e virtuais necessitam ser alinhados corretamente, um em relação ao outro (Azuma, 1997).

3.3.1 *ARToolKit*

É uma biblioteca que usa técnicas de Visão Computacional para calcular o ponto de vista real da câmera em relação a um marcador no ambiente real. Foi desenvolvida na Universidade de Washington e desenhada para facilitar o desenvolvimento de aplicações em Realidade Aumentada.

Utilizam-se técnicas computacionais para calcular a posição e orientação de uma câmera em relação a marcas (que chamamos de marcadores), para que objetos virtuais em 3D possam ser sobrepostos. Há vários passos como se mostra a seguir.

Primeiro, a imagem de vídeo (Figura 3.4 (a)) é transformada em uma imagem binária (Figura 3.4 (b)) baseada no valor do limiar de intensidade. Depois, busca-se nesta imagem por regiões quadradas (Kato, 2007).

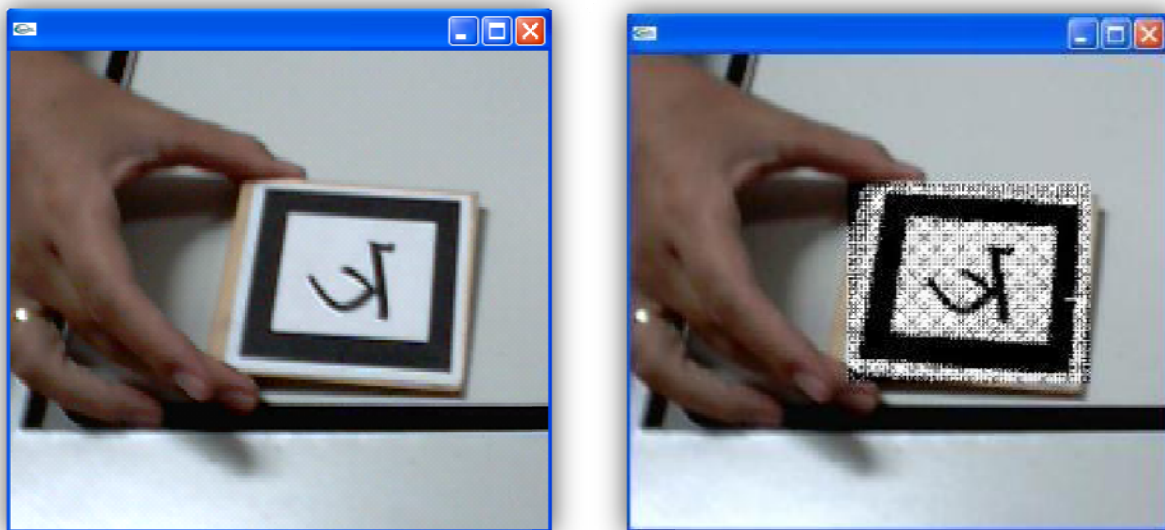


Figura 3.4 –(a) Imagem Real (Input Vídeo) e (b) Imagem Binária Capturada

O ARToolKit encontra todos os quadrados na imagem binária, muitos dos quais não correspondem a marcadores de referência. Para cada quadrado, o desenho padrão dentro dele é capturado e comparado com alguns gabaritos pré-treinados. Se houver alguma similaridade, considera-se que um dos marcadores de referência foi encontrado. O ARToolKit usa então o tamanho conhecido do quadrado e a orientação do padrão

encontrado para calcular a posição real da câmera em relação à posição real do marcador. Uma matriz 3x4 conterá as coordenadas reais da câmera em relação ao marcador. Esta matriz é usada para calcular a posição das coordenadas da câmera virtual. Se as coordenadas virtuais e reais da câmera forem às mesmas, o objeto virtual pode ser desenhado precisamente sobre o marcador real (Figura 3.5).



Figura 3.5 - Objetos Virtuais sobre o marcador

Todo o processo por ser resumido na Figura 3.6.



Figura 3.6 - Diagrama de funcionamento do ARToolkit adaptada de (Kato, 2000)

3.4 Ambientes Virtuais Distribuídos

As aplicações de Realidade Virtual e Aumentada podem ser vistas sob um aspecto bastante amplo, variando de uma única pessoa, usando um único computador e até muitos usuários, usando um sistema distribuído (Rinaldi et al., 2006).

De acordo com Rinaldi et al. (2006) em AVD's os usuários podem compartilhar um mesmo espaço tridimensional virtual de trabalho (*workspace*), onde poderão se auxiliar na execução de uma determinada tarefa, baseando-se nos princípios de trabalho cooperativo baseado em computador (*CSCW - Computer Supported Cooperative Work*).

A Realidade Virtual pode empregar técnica para reproduzir o ambiente real e imaginário e possibilitar a manipulação e visualização de informações no computador como se fosse o ambiente real (Ribeiro, 2006). Assim, entende-se que a complexidade desses ambientes virtuais aumenta na medida em que essas informações tornam-se comuns a uma série de usuário, ou seja, esses ambientes são distribuídos (Ribeiro, 2006).

Rinaldi et. al. (2006) destaca que este espaço compartilhado representa um local comum, podendo ser real ou fictício. O local compartilhado deve apresentar as mesmas características a todos os participantes.

Resumidamente, um AVD (Ambiente Virtual Distribuído) pode ser definido de forma simplificada como um sistema que permite vários usuários interagirem tanto com o ambiente, quanto entre “eles” em tempo real, mesmo que estes estejam em diferentes localidades geográficas (Ribeiro, 2006).

3.4.1 Sistema de Ambiente Virtual Multiusuário Centralizado

Um sistema de ambiente virtual multiusuário pode ser centralizado ou distribuído (Goussweiler, 1994). Na Figura 3.7 pode-se visualizar um sistema multiusuário centralizado.

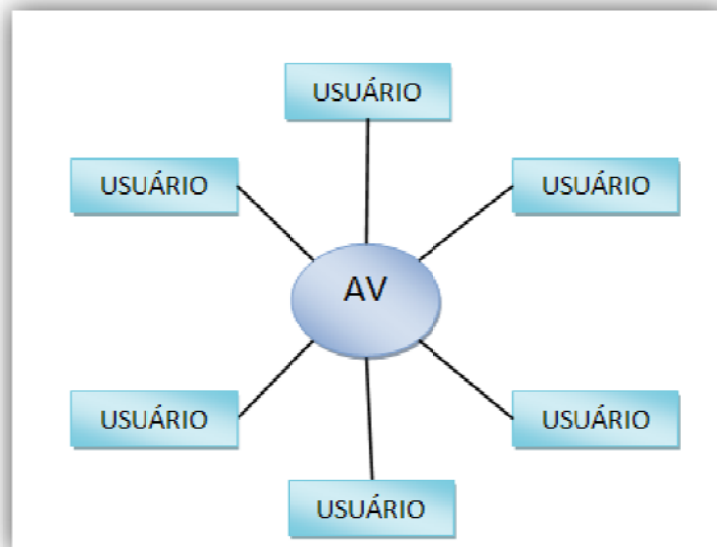


Figura 3.7 - Modelos de Sistema de AV Multiusuário Centralizado (Kirner, 1996).

Como pode ser observado, no modelo centralizado, todos os usuários compartilham o ambiente virtual.

3.4.2 Sistema de Ambientes Virtuais Multiusuários Distribuídos

No Sistema de AV Multiusuário Distribuído o ambiente virtual é acessado e distribuído através de uma rede, como pode ser visto na Figura 3.8.

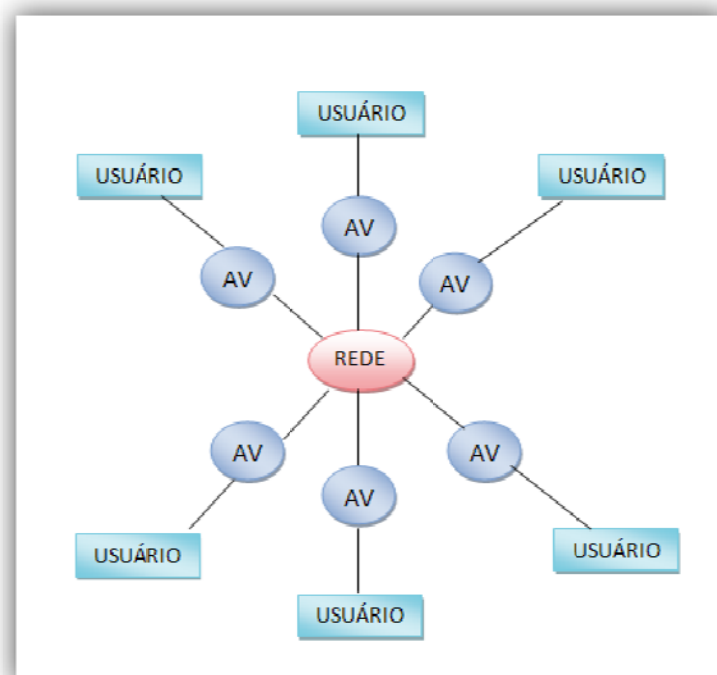


Figura 3.8 - Modelos de Sistema de AV Multiusuário Distribuído (Kirner, 1996)

Neste modelo distribuído, o ambiente virtual pode ser replicado (para ambientes pequenos) conforme a Figura 3.9 ou particionado (para ambientes virtuais de grande porte), de acordo com (Goussweiler, 1994) como se vê na Figura 3.9.

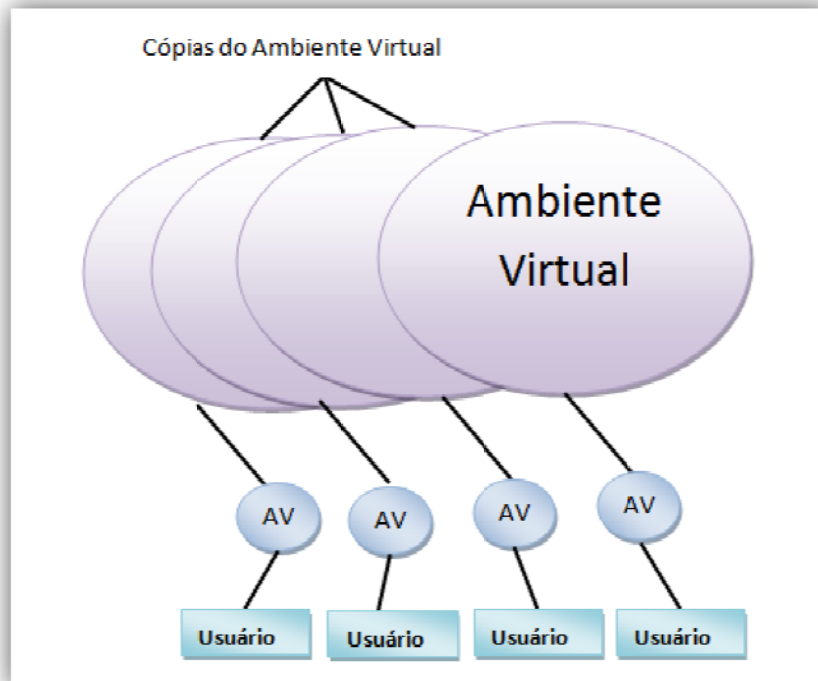


Figura 3.9 - Acesso ao ambiente virtual replicado (Kirner, 1996)

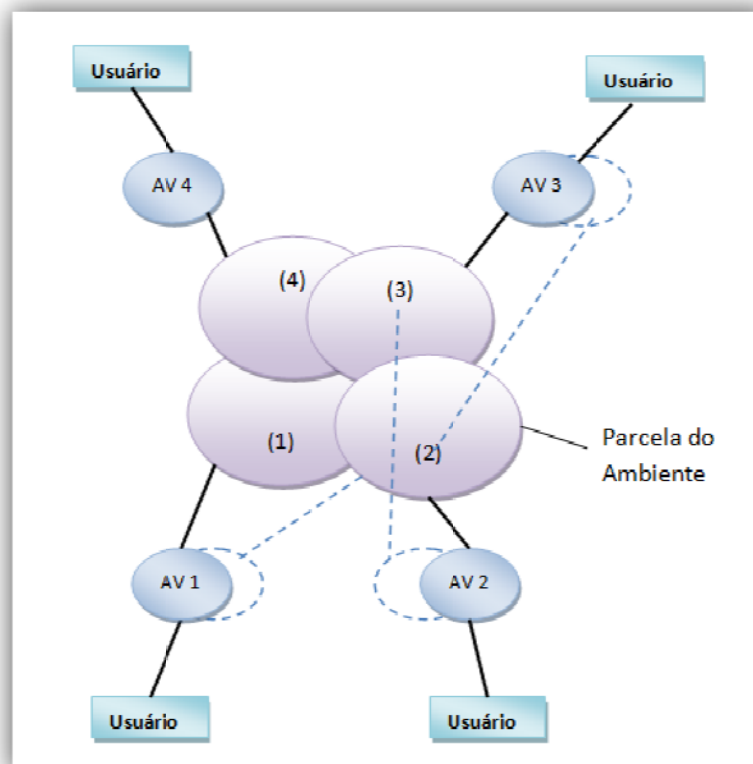


Figura 3.10 - Acesso ao ambiente virtual particionado (Kirner, 1996)

Em um sistema replicado com n usuários, quando um usuário fizer qualquer alteração no ambiente virtual, isto deverá ser comunicado para todas as **($n-1$)** versões do ambiente virtual, onde estão os outros usuários, constituindo a difusão (*broadcast*⁶) (Kirner, 1996).

Já em sistemas particionados com n usuários, a situação é mais complexa, uma vez que o ambiente virtual é dividido em várias partes e cada máquina ficará encarregada de uma delas. Como o usuário pode navegar no ambiente virtual, ele poderá penetrar em outras regiões, de forma que sua máquina ou servidor deverá receber uma réplica da região, onde ele se encontra. Assim cada máquina estará cuidando de uma região fora da sua parcela. Se existirem vários usuários em uma mesma região do ambiente virtual, esse grupo de usuários receberá uma cópia dessa região. Qualquer alteração no ambiente virtual, feita por um membro do grupo, será retransmitida para o restante do grupo, constituindo a retransmissão por grupo (*multicast*⁷) (Kirner, 1996).

3.5 Considerações Finais

Neste capítulo, foi abordado conceitos de Realidade Virtual e Aumentada destacando conceitos relevantes de tecnologias de distribuição de ambientes virtuais.

No protótipo a ser desenvolvido utilizará a tecnologia de distribuição de Ambientes Multiusuários Distribuído, onde o ambiente virtual é acessado e distribuído através de uma rede, caracterizando um sistema onde existe um tutor (professor/aluno) que gerencia o conteúdo e os clientes (alunos) que interagem com o ambiente. Este modelo foi escolhido, pois possibilita que todos os usuários compartilham o mesmo ambiente virtual.

No próximo capítulo, serão demonstrados trabalhos relacionados com o objetivo de mostrar as tecnologias utilizadas e destacar a contribuição desta dissertação.

⁶ Sistema de envio de mensagem, onde a mensagem é enviada para todos os computadores conectados a uma rede

⁷ É um método ou técnica de transmissão de um pacote de dados para múltiplos destinos ao mesmo tempo. Durante uma transmissão *Multicast*, o transmissor envia os pacotes de dados somente uma vez, ficando a cargo dos receptores captarem esta transmissão e reproduzi-la.

4 Trabalhos Relacionados

4.1 Introdução

É promissora a pesquisa na área de sistemas virtuais distribuídos. Foi realizado um estudo do estado da arte em relação à distribuição de ambientes de Realidade Virtual e Aumentada para verificar o que tem sido desenvolvido, com o propósito de buscar novas tecnologias que auxiliassem na criação do sistema proposto. Ao final do capítulo, é apresentada uma tabela que reflete o estado da arte de interesse e norteia os encaminhamentos adotados neste projeto de pesquisa.

Este capítulo objetiva apresentar uma visão dos principais trabalhos encontrados na literatura, mostrando suas características mais importantes. Os trabalhos encontrados foram divididos em dois módulos: *Sistemas de Realidade Virtuais Distribuídos* e *Sistema de Realidade Aumentada Distribuídos*, como pode ser visto na Figura 4.1.



Figura 4.1- Áreas Relacionadas – Adaptada de (Silva, 2008)

4.2 Sistemas de Realidade Virtual Distribuído

4.2.1 DIVE

O *DIVE* (*Distributed Interactive Virtual Environment*) foi desenvolvido pelo *Swedish Institute of Computer Science* (Dive, 2008). É um ambiente de realidade virtual com n usuários com o suporte de uma rede. Os participantes navegam em um cenário 3D e há uma interação com os demais participantes. O sistema não utiliza a plataforma cliente-servidor, cada participante conectado a um ambiente virtual, possui localmente uma cópia com a descrição deste ambiente e as atualizações são continuamente enviadas na rede da qual o usuário está fazendo parte (Dive, 2008). A Figura 4.2 ilustra este sistema.



Figura 4.2 - Sistema de RV (Dive, 2008)

4.2.2 Arquitetura para Distribuição de AV Multidisciplinares de Ensino.

Ribeiro (2006) relata em sua tese, uma arquitetura para distribuição de ambientes virtuais multidisciplinares de ensino. Ambientes Virtuais de Biologia (paisagem com plantas, água, luz e terra) e Química (membrana de uma folha) foram utilizados tendo o fenômeno da fotossíntese como estudo de caso e relação entre os dois ambientes na Figura 4.6 e 4.7 ilustram o funcionamento do aplicativo e suas interações. Foi desenvolvido a partir da arquitetura proposta por Ribeiro e a distribuição ocorre em redes locais utilizando a plataforma CORBA.

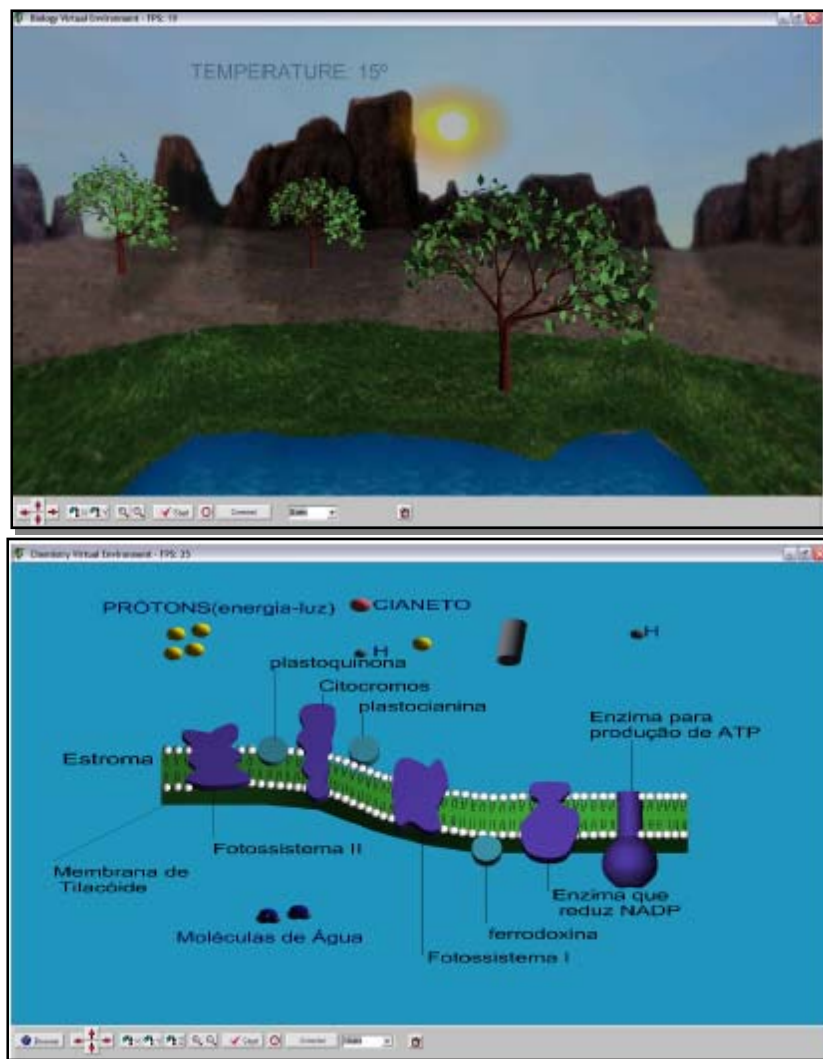


Figura 4.3 - Ambiente Virtual de Biologia e Química (Ribeiro, 2006)

4.2.3 Ambientes distribuídos em Realidade Virtual com suporte a Aprendizagem Cooperativa para a Resolução de Problemas.

Silva (2008) desenvolveu um sistema LVD que viabiliza o uso de RV para oferecer suporte à resolução de problemas por meio de aprendizagem cooperativa. Foram considerados os fundamentos da aprendizagem por experimentação e a efetividade de teorias e princípios pedagógicos construtivista. Como estudo de casos, optou-se por desenvolver o projeto no âmbito da Física.

O trabalho contribui para as áreas multidisciplinares envolvidas no desenvolvimento deste projeto, Computação, Física e Educação. A Figura 4.4 ilustra o sistema.

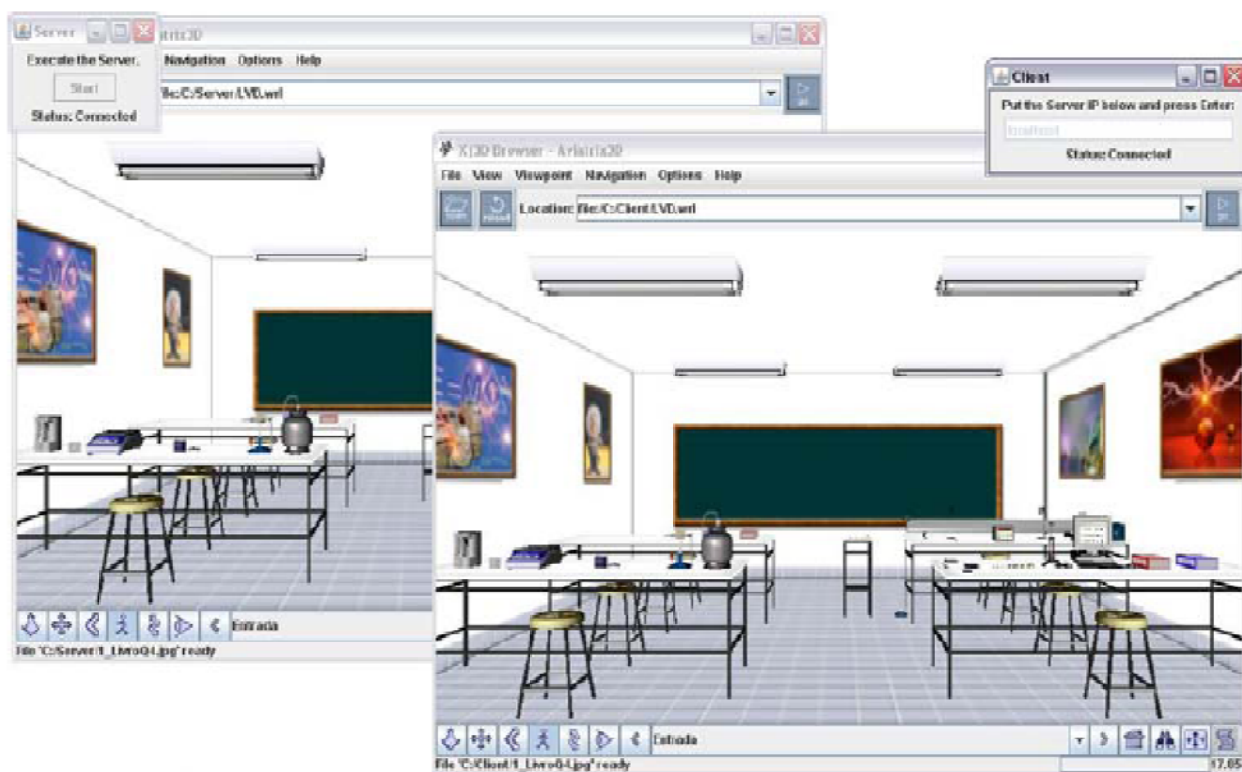


Figura 4.4 - Sistema LVD Silva (2008)

No projeto LVD a distribuição ocorre somente entre dois clientes e um servidor, limitando a quantidade de usuários para utilizar o sistema. Ele também não permite a inserção de novos objetos no ambiente virtual.

4.3 Sistemas de Realidade Aumentada Distribuídos

4.3.1 *Arquitetura para Distribuição e Colaboração em Ambientes Virtuais de Realidade Aumentada.*

Silva (2008) desenvolveu uma arquitetura para distribuição e colaboração em ambientes virtuais de Realidade Aumentada. O objetivo foi apresentar uma abordagem computacional/algorítmica que seja suficiente para suportar o funcionamento de um Ambiente de Realidade Aumentada, disponibilizando um nível razoável de colaboração entre vários usuários conectados em uma rede local de computadores.

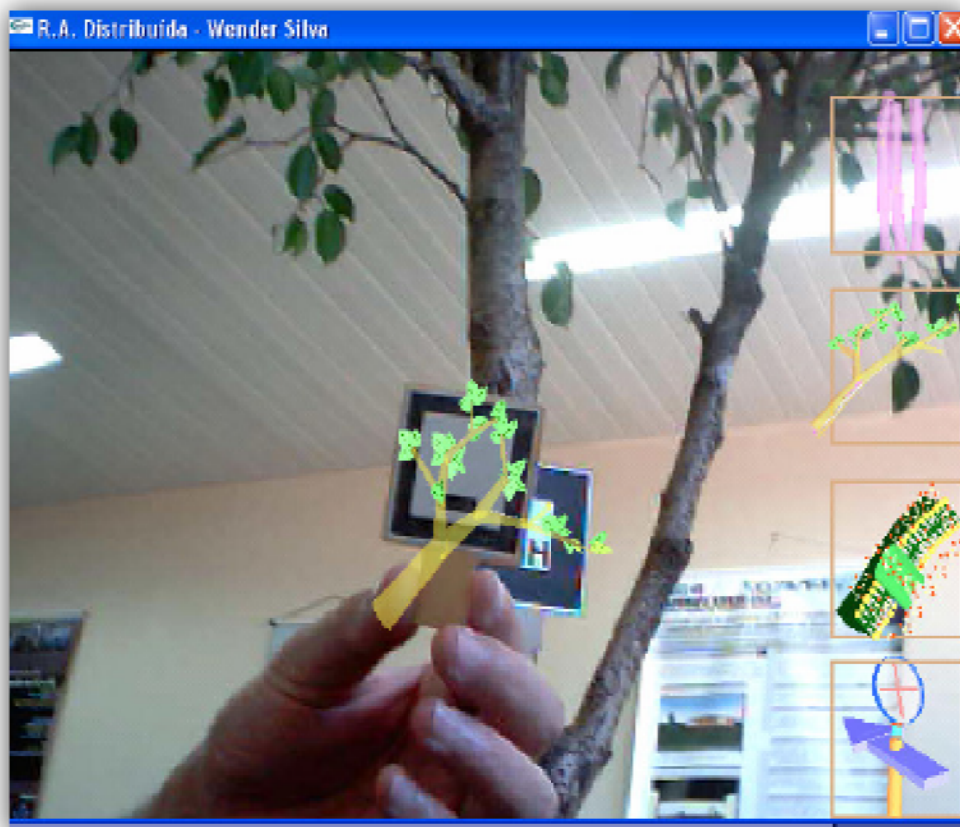


Figura 4.5 - RA Distribuído (Silva, 2008)

No projeto citado, a distribuição ocorre somente em redes locais e especificamente com um modelo da biologia, isto é, para manipular outro, deve-se programar o modelo RA, seguindo as especificações da arquitetura, restringindo seu uso.

4.3.2 *NetArtoolkit*

Oliveira (2008) desenvolveu um sistema distribuído de realidade aumentada, utilizando a ferramenta NetArtoolkit, que é uma biblioteca escrita na linguagem C++ de código livre que é responsável por fazer o rastreamento de marcadores pré-definidos pela ferramenta, e associar um elemento virtual ao marcador.

Os usuários remotamente poderão trabalhar de forma colaborativa para atender objetivos em comum. A realidade aumentada será a interface usada entre o usuário e o computador, sendo uma interface intuitiva, isto é, os movimentos corporais são os próprios “dispositivos” de interação.



Figura 4.6 - Objeto Virtual após o envio do comando `set_rotation` (Oliveira, 2008)

4.3.3 SACRA

Santin (2008) desenvolveu um sistema de autoria colaborativa de realidade aumentada (SACRA), que incorpora técnicas de autoria e colaboração à interface de realidade aumentada altamente interativa, oferecendo a seus usuários novas formas de interação para a construção de ambientes virtuais.

O SACRA oferece um ambiente de RA tangível, para a autoria de ambientes virtuais, possibilitando a colaboração face a face e remota.

A interação face a face ocorre no mesmo local e ao mesmo tempo. A interação assíncrona é realizada no mesmo local, mas em tempos diferentes; nesse caso técnicas

para salvar e recuperar os trabalhos foram desenvolvidas. Na interação síncrona distribuída, usuários remotos realizam o trabalho colaborativo em tempo real. Para ocorrer à interação assíncrona distribuída, o usuário pode enviar ou receber por email o trabalho de autoria desenvolvido, independentemente de tempo e local para a elaboração do trabalho.

A interação no SACRA é realizada por meio dos marcadores de ações, que implementam funcionalidades especializadas para a manipulação de objetos virtuais. Os marcadores de ações possibilitam que os usuários controlem a presença e alterem algumas características desses objetos virtuais.

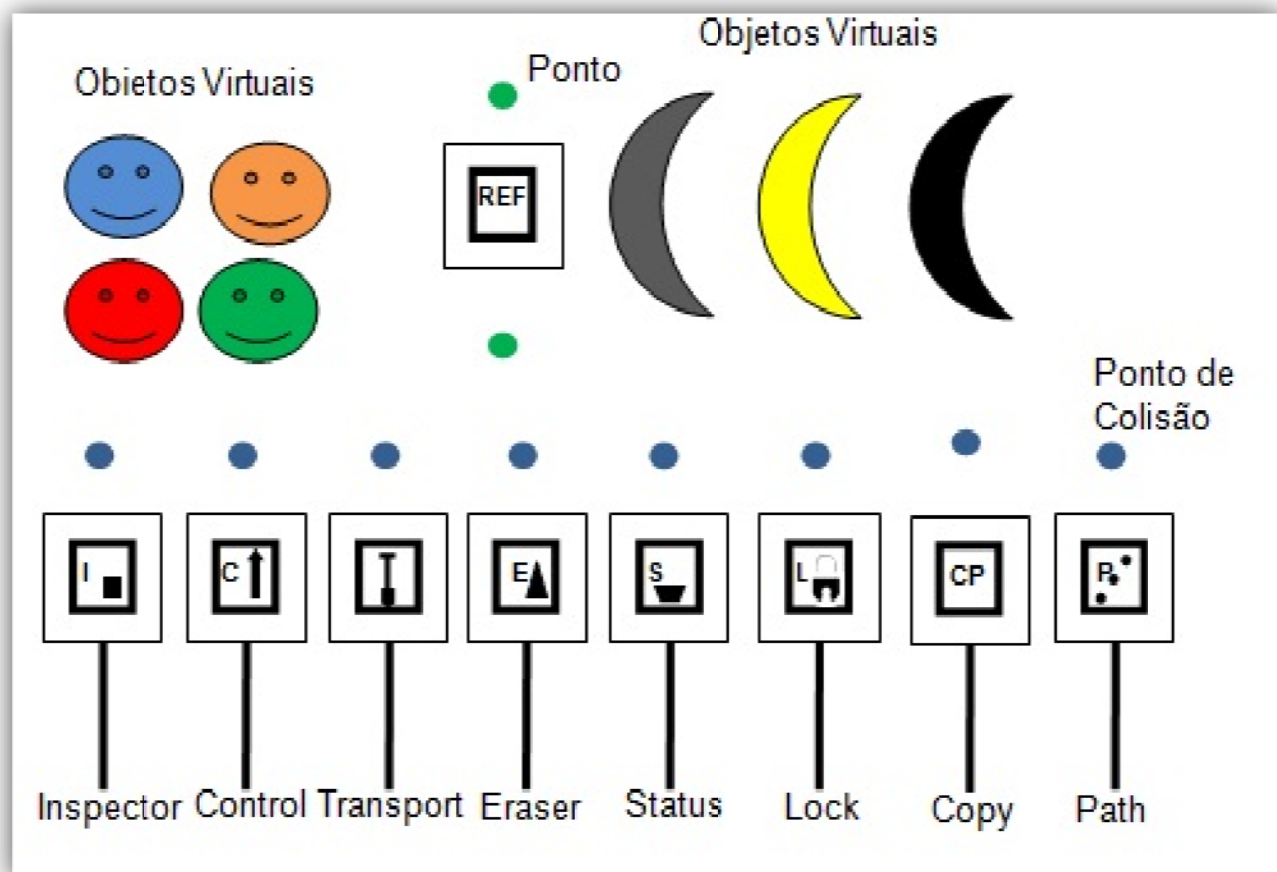


Figura 4.7 - Exemplo dos marcadores de ações e referência (Santin, 2008)

4.4 Considerações Finais

A Tabela 1 ilustra as comparações dos trabalhos relacionados ressaltando suas características e limitações.

Projetos	Distribuição				Visualização	
	Abrangência		Ambientes Comuns		Novo Modelo RV / RA	Ambientes de RV / RA
	LAN	Internet	RV	RA		
DIVE	✓	✗	✓	✗	✗	✗
LVD	✓	✓	✓	✗	✗	✗
Arquitetura Distribuição de AV Multidisciplinares de Ensino	✓	✗	✓	✗	✗	✗
Arquitetura Distribuição e Colaboração em Ambientes Virtuais de RA	✓	✗	✗	✓	✗	✗
NetArtoolkit	✓	✓	✗	✓	✗	✗
Sacra	✓	✓	✗	✓	✓	✗

Tabela 4.1- Comparação dos trabalhos relacionados ao projeto

Assim, observando a Tabela 4.1, percebe-se que os trabalhos desenvolvidos estão focados em RA ou RV, não contemplando a distribuição simultânea de ambos. Pode-se destacar ainda que nem todos os sistemas permite à distribuição pela na Internet. E em nenhum dos casos ocorre à visualização de ambientes de RV e RA. Cada característica é melhor descrita abaixo:

Distribuição de ambientes comuns de RV e RA- percebe-se que os trabalhos desenvolvidos estão focados em RV ou RA, não contemplando a distribuição simultânea de ambos.

Distribuição LAN e Internet - dentre os trabalhos selecionados que são relacionados à pesquisa do sistema nem todos possuem o recurso de distribuição pela Internet.

Permite a distribuição de um novo modelo RV/RA - a limitação de distribuição de objetos RV é bem clara, sendo que em todos os projetos a distribuição está ligada a

um único estudo de caso, ou seja, se o usuário quiser inserir um novo modelo no ambiente ele não conseguirá.

Visualização simultânea de modelos de RV e RA – nenhuns dos sistemas analisados possuem este recurso.

Analisando as características de cada sistema na Tabela 4.1 observa-se que cada projeto não contempla os itens abaixo, isto conduz a necessidade de se projetar um protótipo que:

- ✓ Possibilite a distribuição na Rede Local e na Internet, possibilitando uma interação não só num mesmo local físico, mas também em locais diferentes. Isso contribuirá para o ensino de Educação à Distância.
- ✓ Distribuição de um novo modelo virtual que não pertence à biblioteca comum. Este modelo pode ser distribuído por qualquer usuário, sendo que posteriormente será inserido na biblioteca compartilhada. Isto permite a flexibilidade do sistema tornando aplicável em qualquer área do conhecimento.
- ✓ Visualização de modelos em RV e RA, facilitando assim o entendimento do aluno. Acredita-se que a utilização da Realidade Aumentada conduzirá a uma melhor compreensão, como por exemplo, ver um a projeção de um coração virtual no tamanho real. Além da visualização, o usuário poderá modificar o código do modelo virtual através de um editor VRML que será desenvolvido. Essa alteração poderá ser repassada para todos os usuários conectados ao sistema.

O próximo capítulo aborda a especificação do protótipo proposto.

5 Especificação do Sistema

5.1 Introdução

A partir do estudo realizado nos Capítulos 2,3 e 4, foi possível conceber uma arquitetura de Realidade Virtual e Aumentada distribuída que ofereça ao usuário condições de interagir com objetos virtuais; em ambos ambientes a proposta permite, por meio de uma interface gráfica, a manipulação de objetos existentes numa biblioteca comum e ainda suporta a inserção de novos objetos, que podem ser distribuídos.

5.2 Arquitetura do Sistema

A arquitetura proposta permite a conexão de vários (n) usuários conectados a um Ambiente de Distribuição de Realidade Virtual e Aumentada, utilizando de uma interface que permita a manipulação de objetos virtuais. A Figura 5.1 apresenta esta arquitetura.

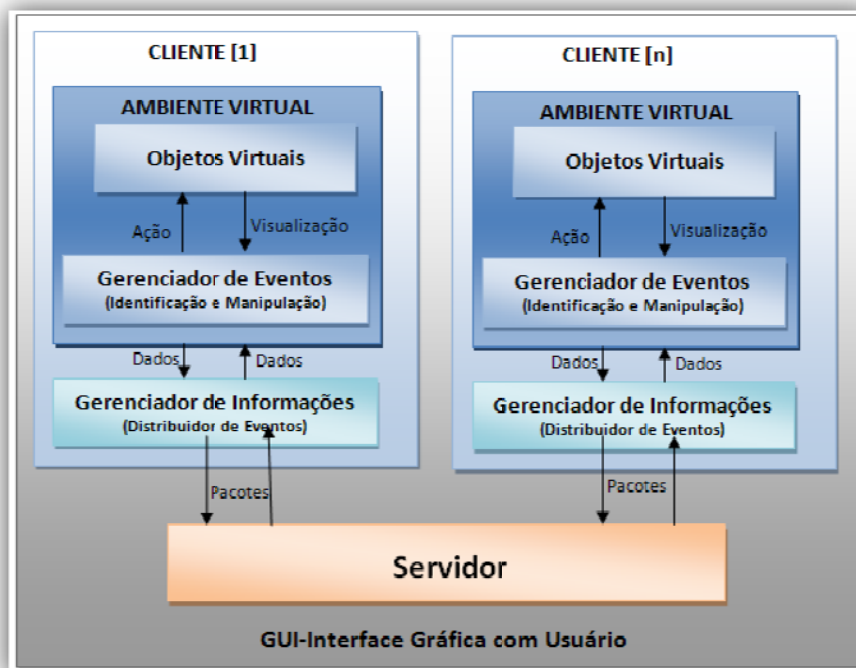


Figura 5.1 - Arquitetura do Sistema

A arquitetura proposta nesta dissertação é composta pelos seguintes módulos:

Interface Gráfica com o usuário (GUI): possui o módulo de conexão do sistema, e a interface de visualização RV e RA.

Servidor: recebe/envia pacotes dos clientes e distribui na rede as informações sobre os objetos.

Cliente: neste módulo encontra-se a seguinte subdivisão como pode ser vista na Figura 5.2.

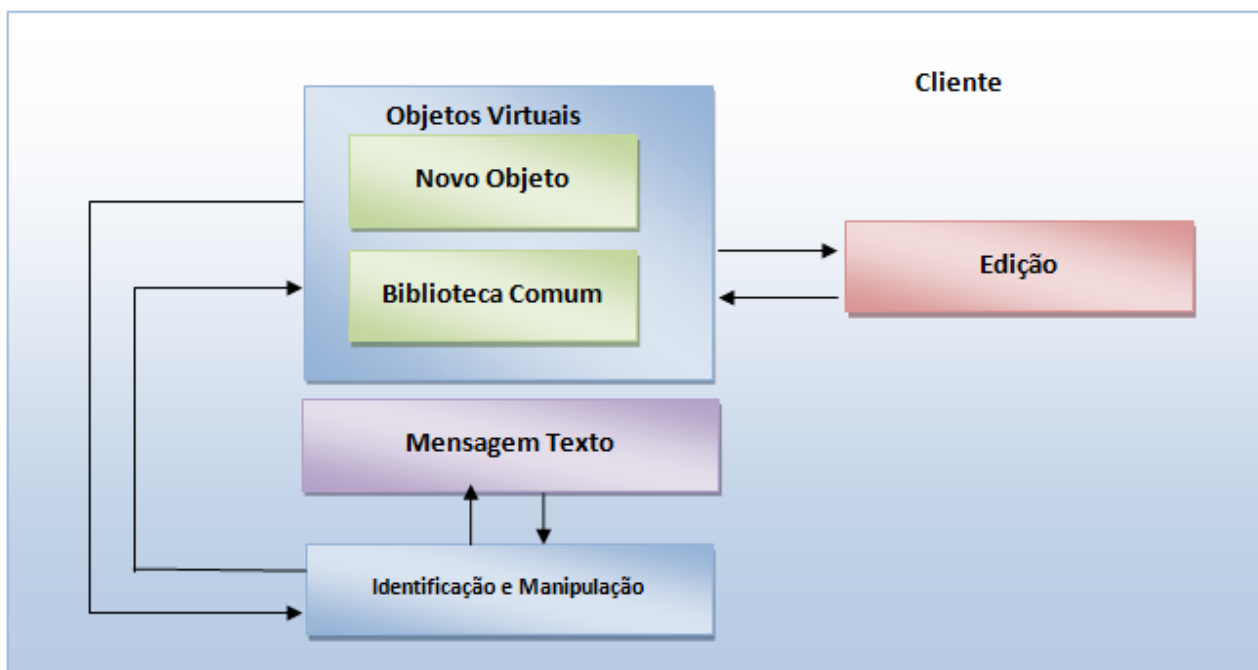


Figura 5.2 – Módulo Cliente

O módulo Cliente possui os seguintes sub-módulos:

Biblioteca comum: possui objetos virtuais comuns a todos os usuários, faz parte do pacote de instalação.

Novo Objeto: um objeto por ser inserido por qualquer usuário a qualquer momento.

Mensagem de Texto: comunicação via chat.

Edição: neste módulo, os objetos da biblioteca comum e os novos objetos podem ser editados. O usuário deve possuir conhecimento em VRML para estar criando e editando o modelo.

Distribuidor de Eventos: Utilizando funções multi-thread, dados provenientes do sistema RV são coletados e comparados com os dados provenientes da rede. Havendo interação no ambiente virtual, é propagada pela rede uma mensagem de atualização para os demais computadores

A Figura 5.3 apresenta uma visão mais detalhada do mesmo sistema, ilustrando o fluxo de informação.

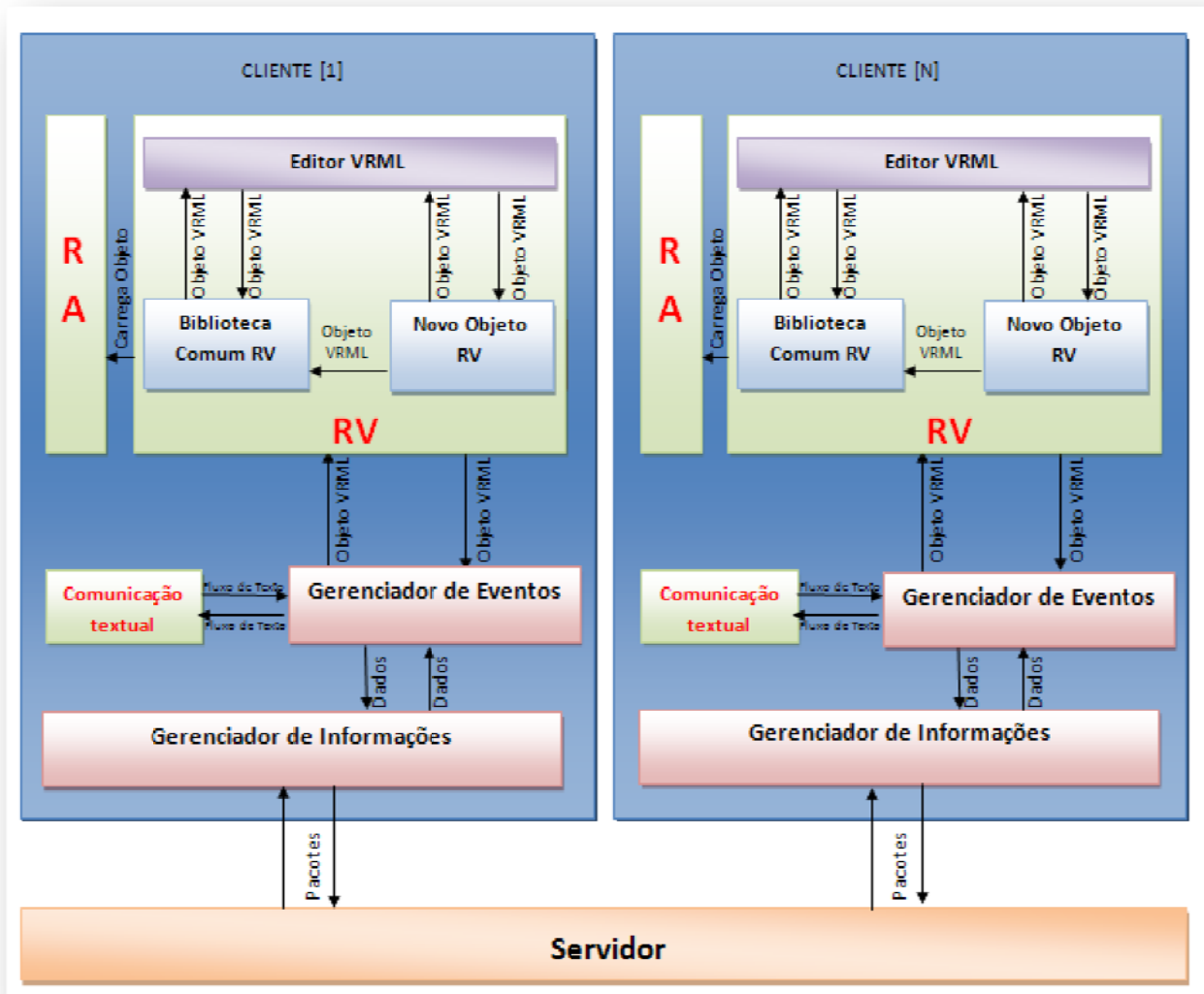


Figura 5.3 - Fluxo da Informação

RV: possui uma biblioteca de objetos comuns em RV, que pode ser compartilhada, ainda possibilita a inserção de um novo objeto e a edição de objetos utilizando um editor VRML.

RA: neste módulo o usuário pode visualizar o objeto VRML em RA.

Gerenciador de Informações: módulo responsável por receber todo tipo de dados da rede

Gerenciador de Eventos: recebe os dados provenientes do Gerenciador de informações e classifica-os em dados de controle, objetos VRML e fluxo de texto, em seguida direciona cada tipo de dado para o módulo correspondente.

5.3 Interface Gráfica com Usuário – Graphical User Interface (GUI)

O módulo GUI - Interface Gráfica do Usuário possibilita a comunicação do usuário com todos os outros módulos desenvolvidos.

Utilizando a tecnologia cliente/servidor, onde o processamento da informação é dividido em módulos ou processos distintos, um processo é responsável pela manutenção da informação (Servidor) e outros responsáveis pela obtenção dos dados (Clientes).

Os processos cliente enviam mensagem para o processo servidor, e este por sua vez processa e envia as mensagens para todos os outros clientes conectados. A arquitetura cliente/servidor é praticamente a mesma, como pode ser observada na Figura 5.4.

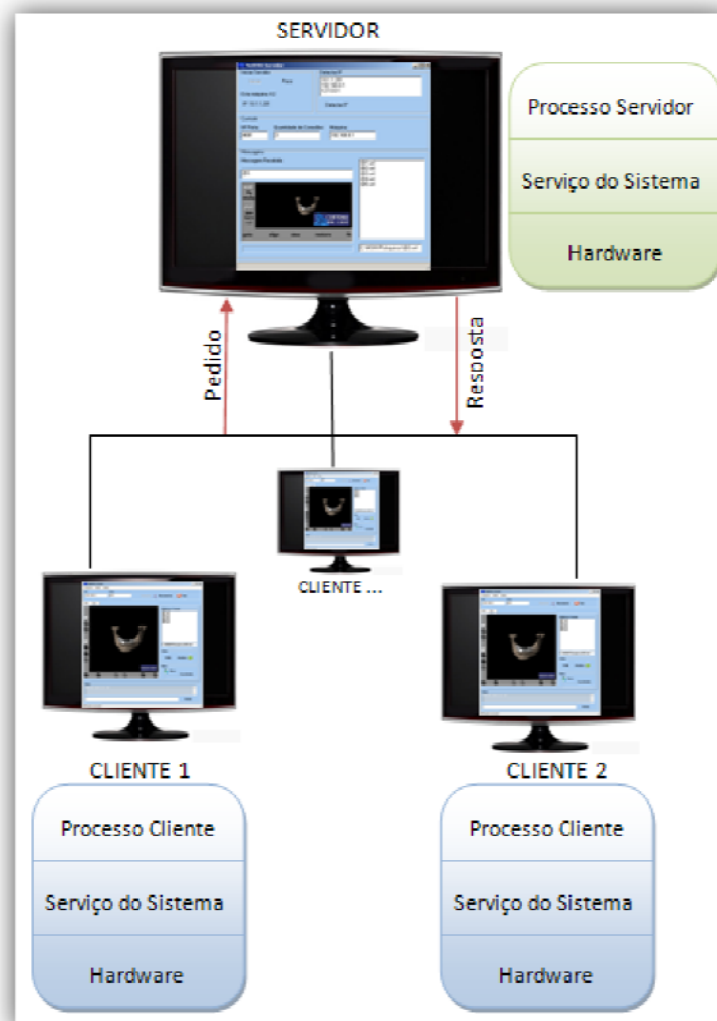


Figura 5.4 - Arquitetura Cliente/Servidor

5.3.1 Recursos

A interface do servidor é subdividida em três partes, a primeira é responsável por (iniciar ou finalizar o Servidor), a segunda controla a quantidade de clientes conectados e a terceira tem a função de receber as mensagens e propagar para todos os clientes conectados ao servidor. A Figura 5.5 mostra cada parte.

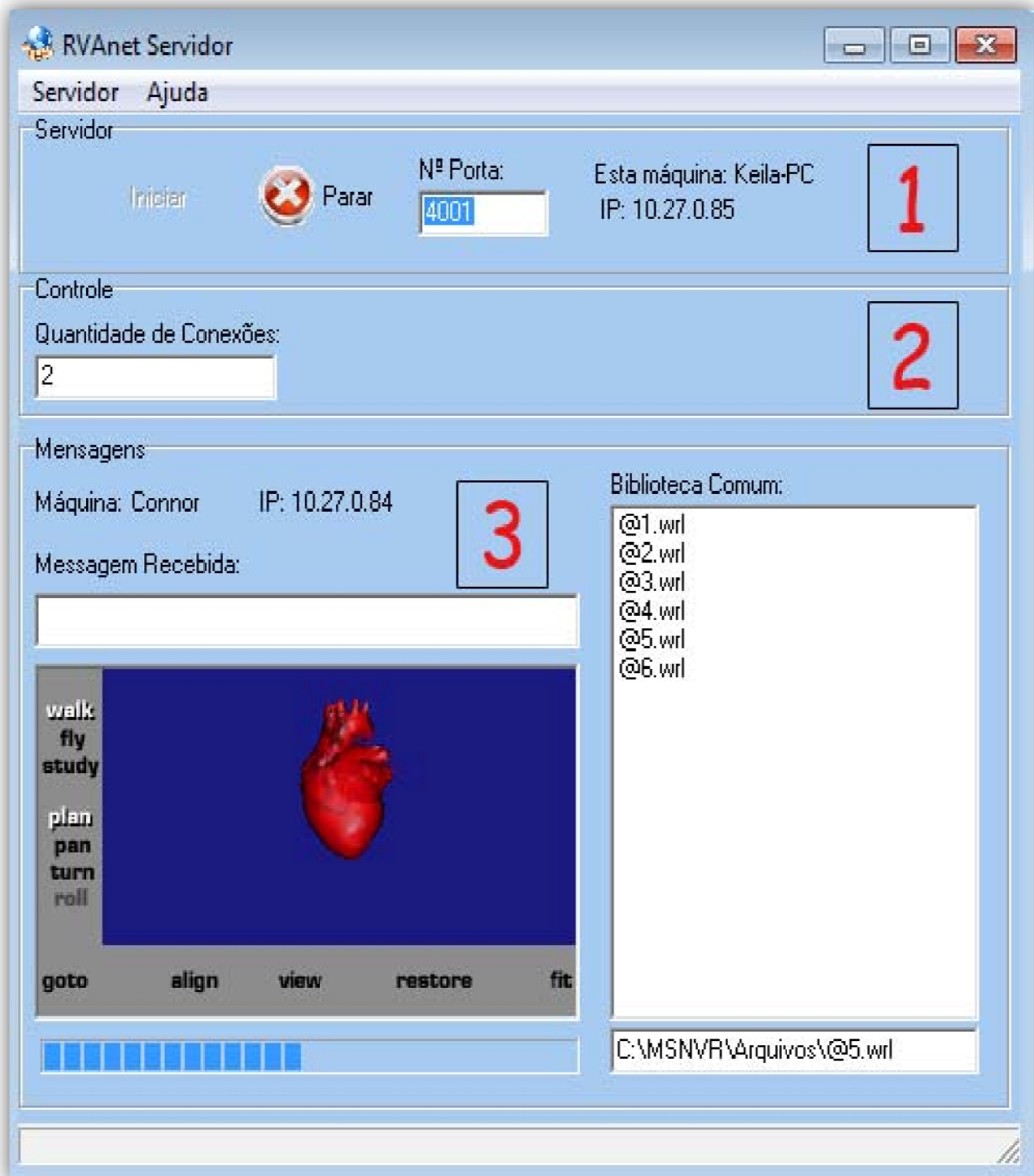


Figura 5.5 - Inicializar Servidor

Para o cliente foi desenvolvida a seguinte interface:

- 1- **Interface Gráfica com o Usuário**- possui a parte de conexão do cliente através do endereço IP do servidor (1), uma aba de visualização da interface RV e RA (2) e ainda um chat que permite a comunicação via texto (3), como pode ser visualizado na Figura 5.6.

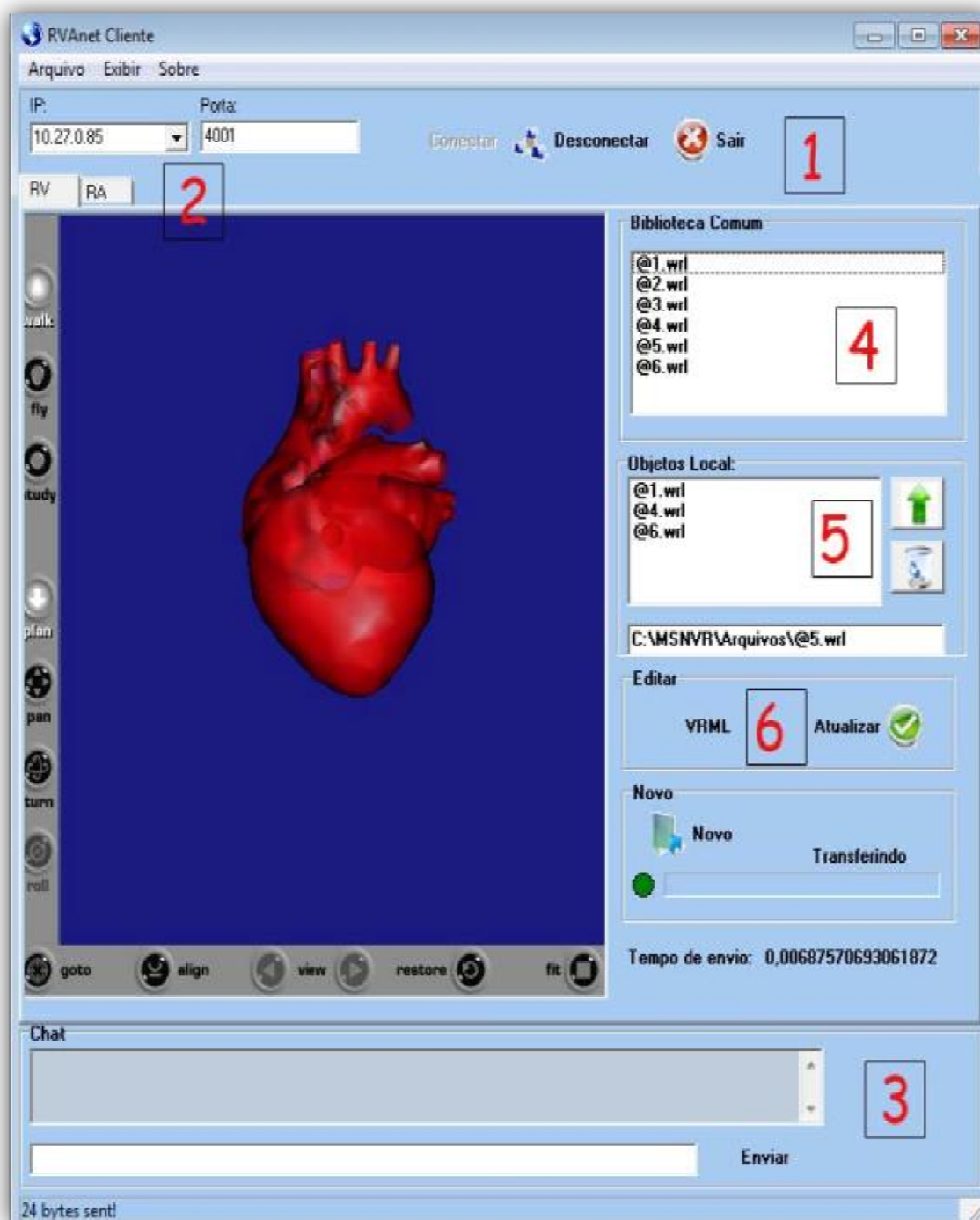


Figura 5.6 - Interface Cliente

2- **Ambiente Virtual**- a aba de visualização da interface RV (2) possui uma biblioteca comum (4) contendo todos os modelos cadastrados. Essa biblioteca é comum a todos os usuários, faz parte do pacote de instalação.

No módulo de edição (6), é possível fazer a manipulação do código do objeto VRML. A biblioteca comum contém modelos cadastrados, que é acessível por todos os usuários, fazendo parte do pacote de instalação. Estes modelos podem ser editados. Ao clicar no botão VRML abrirá outra janela que será responsável pela manipulação do código do objeto, como mostra a Figura 5.7.

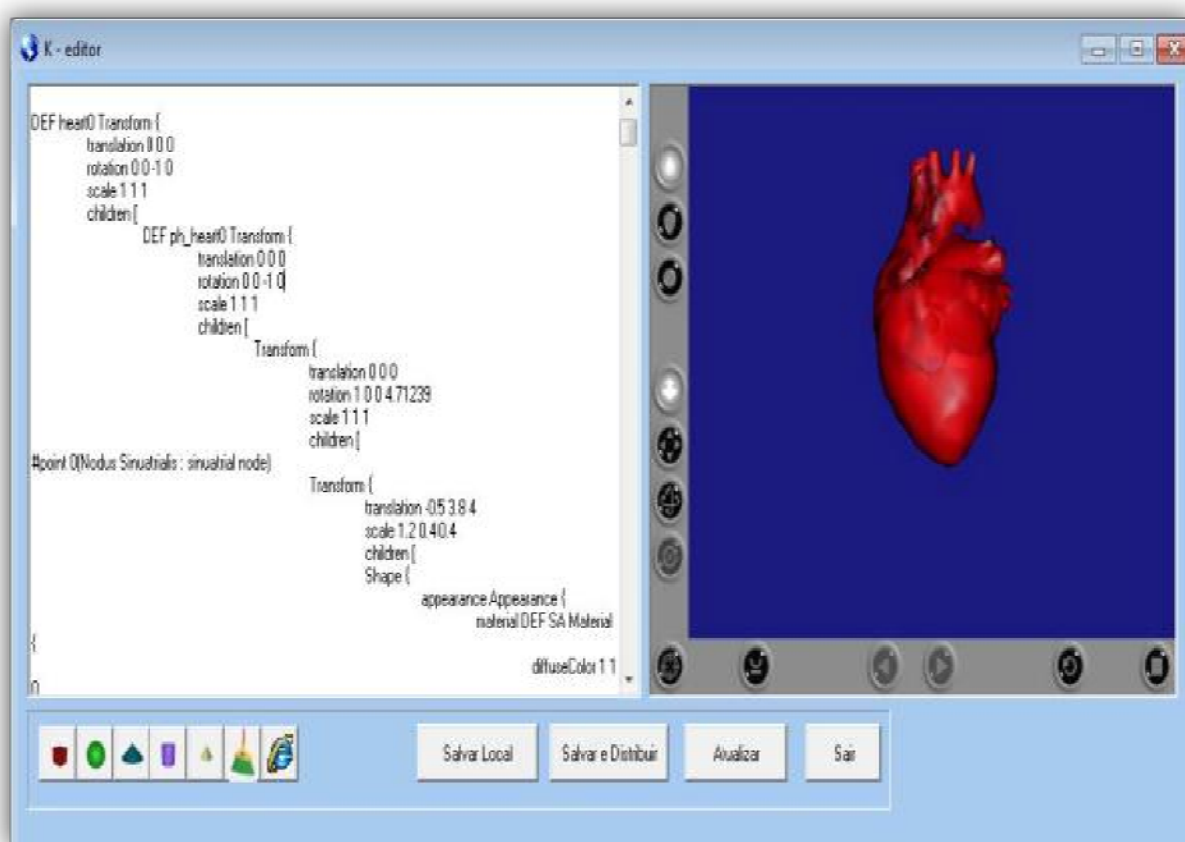


Figura 5.7 - Editor VRML

No editor, o usuário poderá tanto alterar o código do objeto, como criar novos objetos através de algumas primitivas que o editor fornece. Após criar ou alterar o objeto ele poderá salvar local e depois distribuir.

Na aba de visualização do RA, o usuário encontrará três pontos principais (1) a área de visualização do RA, (2) botão converter RV/RA onde o modelo virtual distribuído em RV será visualizado em RA, e ainda (3) o botão RA, responsável por ativar o módulo de Realidade Aumentada.

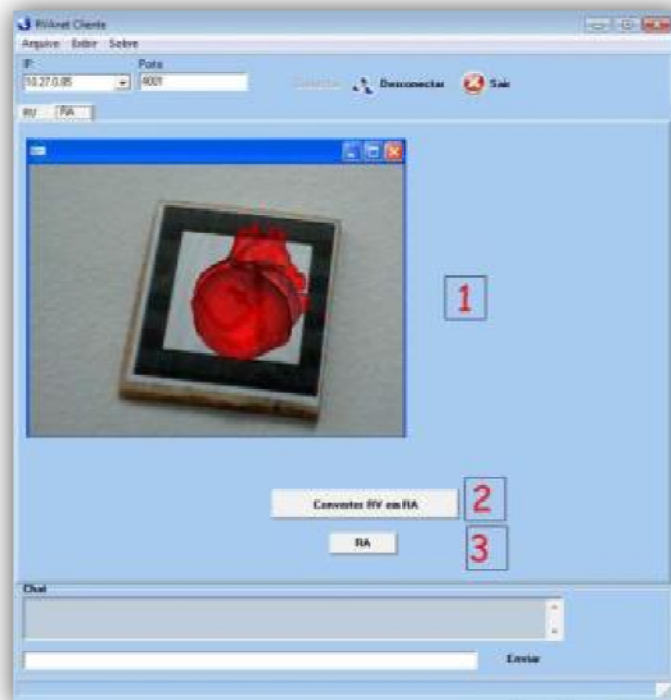


Figura 5.8 - Interface RA

- 3- **Novo Objeto:** Ao clicar no botão novo o usuário poderá estar enviando um novo objeto para todos estarem compartilhando. Este objeto pode estar na máquina local do usuário ou poderá ser baixado de algum site, que possua modelos VRML, como mostra a Figura 5.9.

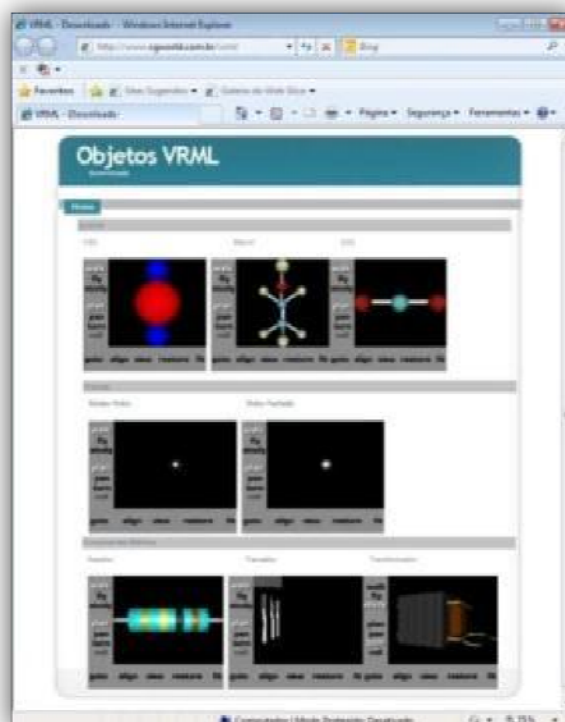


Figura 5.9 - Download Modelos VRML

Neste exemplo, o usuário fez o *download* do modelo e salvou na sua máquina, depois clicou no botão Novo, como mostra a Figura 5.10 abaixo.

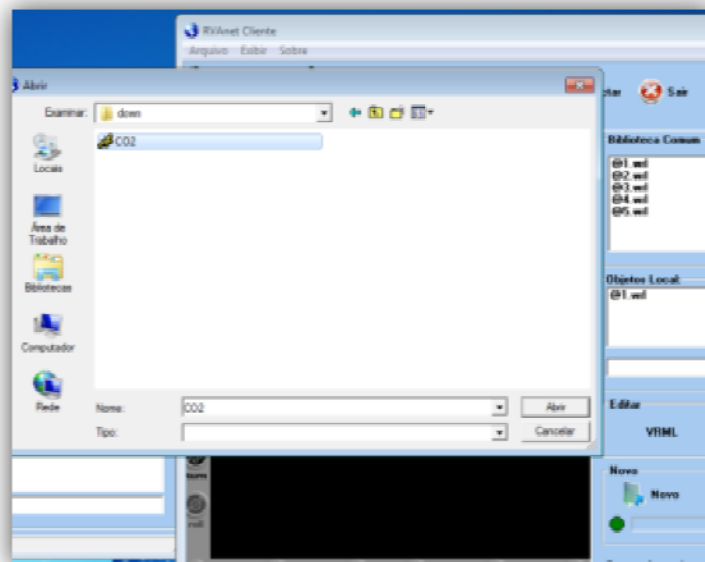


Figura 5.10 - Novo Objeto

Na Figura 5.10, pode-se observar que a biblioteca comum possui apenas 5 objetos, já na Figura 5.11 abaixo, após o envio do novo objeto ela possui 6 objetos. O tempo de envio foi de 5 milissegundos.

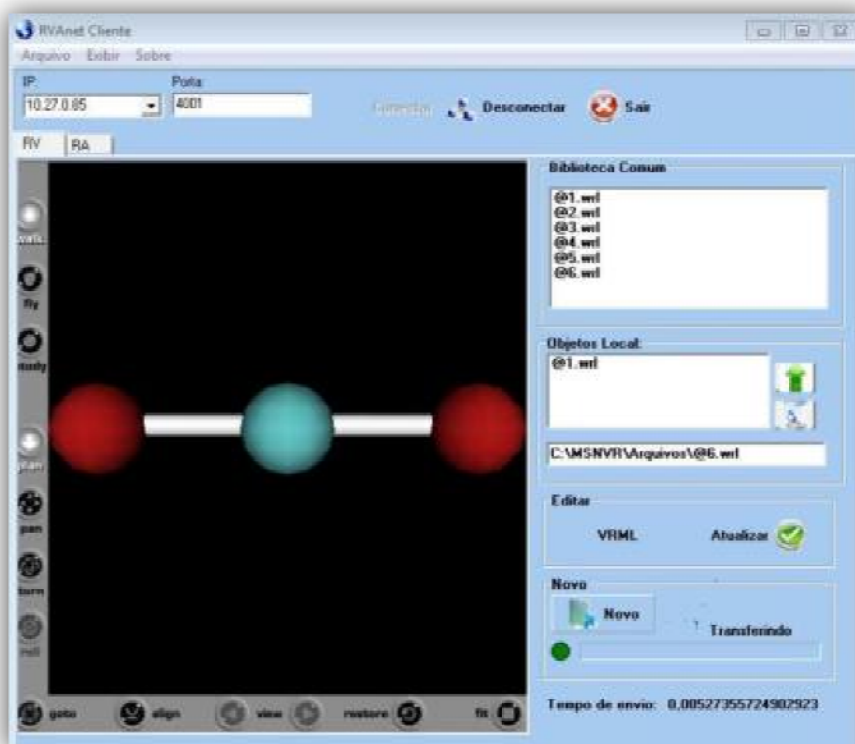


Figura 5.11 - Novo Objeto Biblioteca

5.4 Considerações Finais

Neste Capítulo, foi apresentada a arquitetura do protótipo proposto mostrando a funcionalidade de cada módulo. Dentre os módulos apresentados, destaca-se o **Gerenciador de Informações** que é o módulo responsável pelo envio de pacotes do Servidor para os Clientes, e também o **Gerenciador de Eventos** que tem a finalidade de classificar os dados e direcionar para os módulos correspondentes.

Os detalhes de desenvolvimento do protótipo serão apresentados no próximo capítulo.

6 Detalhes de Implementação

6.1 Introdução

A partir da Arquitetura de Realidade Virtual e Aumentada distribuída proposta, foi desenvolvido um protótipo o qual recebeu o nome de RVAnet.

RVAnet foi desenvolvido para desktop, utilizando a plataforma Windows. Essa seção apresenta as tecnologias de apoio e ainda são abordados os detalhes de desenvolvimento do protótipo RVAnet.

- ✓ **VRML (*Virtual Reality Modeling Language*)** é uma linguagem de modelagem que possibilita a criação de ambientes virtuais tridimensionais, com alta qualidade, aplicáveis a Internet, utilizando apenas um browser e um plug-in para realizar a visualização. VRML permite a descrição de um espaço tridimensional, tornando possível construir ambientes correspondentes a uma dada realidade ou completamente imaginários. Fundamentalmente, o browser gera em tempo real a representação visual equivalente à descrição textual da cena.
- ✓ **ARToolkit:** Como já referenciado no Capítulo 2 desta dissertação, Kirner (2007) destaca que o ARToolKit é um sistema que viabiliza o desenvolvimento de interfaces de Realidade Aumentada. Disponível gratuitamente no site do laboratório HITL da Universidade de Washington, o ARToolKit emprega métodos de visão computacional para detectar marcadores na imagem capturada por uma câmera. O rastreamento óptico do marcador permite o ajuste de posição e orientação para realizar a renderização do objeto virtual, de modo que esse objeto pareça estar “atrelado” ao marcador (Kirner, 2007).
- ✓ **Cliente-Servidor:** É importante ressaltar que os modelos de computação distribuída (Cliente-Servidor e P2P) citadas no Capítulo 2 não são tecnologias concorrentes e nem que uma surgiu para substituir à outra, pois cada uma foi projetada

com um foco de aplicação, portanto elas são necessárias em soluções específicas (Coulouris, 2001).

Enquanto que a tecnologia Cliente-Servidor prioriza seus esforços na garantia de uma melhor gerência de seus serviços e recursos através da centralização dos mesmos, a tecnologia P2P focaliza seus esforços na garantia de uma melhor distribuição destes serviços e recursos (ocasionando uma maior dificuldade de controle e gerenciamento) (Coulouris, 2001).

Em ambientes educacionais normalmente existe a pessoa do tutor que supervisiona e controla as atividades educacionais logo pode se notar a necessidade de um módulo servidor (tutor/professor), que controla e administra os clientes (alunos) na rede.

✓ **Sockets:** Após comparações realizadas no Capítulo 2, foi definida a utilização de Sockets, que é um método utilizado em aplicações distribuídas, por isso compõe a parte de distribuição dessa arquitetura proposta na dissertação.

6.2 Estágios Desenvolvidos.

O protótipo possui alguns estágios relevantes que serão expostos os códigos abaixo, tanto para o Servidor e para o Cliente.

6.2.1 Protocolo de envio de dados

Para o envio de dados, foi desenvolvido um protocolo, composto de um identificador e conteúdo. Para cada evento existe um código especial descrito abaixo na Tabela 6.1.

Tabela 6.1 - Sintaxe do protocolo de envio

Comando	Código	Descrição
Abre objeto da biblioteca	@(Número)	Abre um objeto que já existe
Atualiza objeto da biblioteca	#(Número)	Se um objeto é editado
Acrescenta novo objeto a biblioteca	&	Um novo objeto é compartilhado
Fluxo de texto	Se nenhum dos outros	Mensagem de texto é enviada aos clientes

O código correspondente a cada comando ocupa a primeira posição do vetor do protocolo enviado. Cada evento descrito acima será detalhado nos tópicos a seguir.

6.2.2 Servidor

O servidor monitora a rede e recebe dados de todos os clientes, uma vez que um novo dado seja recebido, este o salva em memória (Figura 6.1) e repassa o mesmo a todos os clientes inclusive ao cliente que enviou o dado.

```
tamanho = Socket->ReceiveLength();  
// pega o tamanho do pacote de Dados  
buffer2 = new UCHAR[tamanho];  
// apaga os arquivos que chegaram na ultima vez  
remove("C:\\MSNVR\\Arquivos\\Server\\Arq_Entrada.txt");  
// recria o arquivo para salvar os dados recebidos  
arquivo = fopen("C:\\MSNVR\\Arquivos\\Server\\Arq_Entrada.txt","ab");  
// recebe os dados  
Socket->ReceiveBuf(buffer2,tamanho);  
// grava os dados em memoria  
fseek(arquivo,0,SEEK_END);  
// posiciona curso de gravação de arquivo no fim do mesmo  
fwrite(buffer2,tamanho,1,arquivo);  
// Descarrega conteudo da memoria para arquivo  
Label7->Caption=Socket->RemoteHost;  
// Exibe informações do host que enviou pacote  
Label8->Caption=Socket->RemoteAddress;  
// Exibe informações do host que enviou pacote  
TamanhoBuffer += tamanho;  
// atualiza variavel que controla progressbar  
ProgressBar1->Position=TamanhoBuffer;  
// atualiza progressbar  
fclose(arquivo);  
// envia os dados recebidos a todos os clientes conectados  
for(int i=0;i<ServerSocket1->Socket->ActiveConnections;i++)  
ServerSocket1->Socket->Connections[i]->SendBuf(buffer2,tamanho);  
delete[] buffer2;  
Id->LoadFromFile("C:\\MSNVR\\Arquivos\\Server\\Arq_Entrada.txt");  
LinhaComandos=Id->Strings[0];  
identificador = LinhaComandos.c_str();
```

Figura 6.1 - Código Identificador de eventos 1

Uma vez que os dados foram encaminhados aos clientes, o servidor executa uma rotina denominada identificador de eventos que é responsável por classificar qual ação deve ser realizada com o dado recebido. Um protocolo para o envio da mensagem foi desenvolvido, onde a primeira linha do dado contém um identificador reservado e também um numero decimal associado ao objeto. Se o identificador for igual "&" o servidor e os clientes interpretam como o recebimento de um novo objeto que ainda não está na biblioteca, mas que deve ser inserido nela. Se o identificador for "@", a interpretação é que um objeto já existente na biblioteca está sendo carregado no navegador (Figura 6.2).

```
if(identificador[0] == '&') //Se for recebido um novo objeto
{
remove("C:\\MSNVR\\Arquivos\\Server\\Temp\\TempServer.wrl");
CopyFile("C:\\MSNVR\\Arquivos\\Server\\Arq_Entrada.txt","C:\\MSNVR\\Arquivos\\Server\\Temp\\TempServer.wrl",true);
//copia objeto
arquivo = fopen("C:\\MSNVR\\Arquivos\\Server\\Temp\\TempServer.wrl","r+");
fseek(arquivo, 0,SEEK_SET);
fputs("#VRML V2.0 utf8",arquivo);
close(arquivo);
ListaObjetos->LoadFromFile("C:\\MSNVR\\Arquivos\\Server\\ServerObjs.txt");
Obj = StrToInt(ListaObjetos->Strings[0]);
Obj++;
Edit5->Text= Caminho+"@"+Obj+".wrl";
remove(Edit5->Text.c_str());
CopyFileA("C:\\MSNVR\\Arquivos\\Server\\Temp\\TempServer.wrl",Edit5->Text.c_str(),true);
//copia objeto
ListaObjetos->Insert(0, Obj);
ListaObjetos->SaveToFile("C:\\MSNVR\\Arquivos\\Server\\ServerObjs.txt");
CppWebBrowser1-> Navigate (WideString
("C:\\MSNVR\\Arquivos\\Server\\Temp\\TempServer.wrl"));
}
else if(identificador[0] == '@'){ // Abrir um objeto que ja pertence a biblioteca //
Obj = StrToInt(identificador[1]);
Edit5->Text= Caminho+"@"+Obj+".wrl";
CppWebBrowser1->Navigate (WideString(Edit5->Text));
}
```

Figura 6.2 - Código Identificador de eventos 2

E ainda se o identificador for igual “#” o dado é interpretado como um novo objeto que será adicionado ao final da biblioteca em todos os usuários clientes e também no servidor (Figura 6.3).

```
else if(identificador[0] == '#'){ // Atualizar um arquivo existente para outro
com mesmo nome
remove("C:\\MSNVR\\Arquivos\\Server\\Temp\\TempServer.wrl");

CopyFile("C:\\MSNVR\\Arquivos\\Server\\Arq_Entrada.txt", "C:\\MSNVR\\Arquivos\\
\\Server\\Temp\\TempServer.wrl", true);
//copia objeto
arquivo=fopen("C:\\MSNVR\\Arquivos\\Server\\Temp\\TempServer.wrl", "r+");
fseek(arquivo, 0, SEEK_SET);
fputs("#VRML V2.0 utf8", arquivo);
fclose(arquivo);
Obj = StrToInt(identificador[1]);
Edit5->Text=Caminho+"@"+Obj+".wrl";
remove(Edit5->Text.c_str());
CopyFile("C:\\MSNVR\\Arquivos\\Server\\Temp\\TempServer.wrl", Edit5->
Text.c_str(), true); //copia objeto
CppWebBrowser1-
>Navigate(WideString("C:\\MSNVR\\Arquivos\\Server\\Temp\\TempServer.wrl"));
}
else // Comandos de texto simples
Memo1->Text=LinhaComandos;
FileListBox1->Update();
```

Figura 6.3 - Código Identificador de eventos 3

No cliente, os dados podem ser enviados de várias formas dependendo de seu tipo, dado de atualização, dado de texto, dado para abertura de objeto pré-existente, dado para um novo objeto.

6.2.3 Solicitando a abertura de objetos existentes – Cliente

O código da Figura 6.4 serve para solicitar a abertura de um objeto já existente. Se o objeto já estiver carregado na biblioteca comum não é necessária a retransmissão do mesmo por completo, somente de um sinal de controle, indicando que o objeto já existe e qual é o objeto.

```
char *buffer = new char[30];
char *buf = new char[4];
int len;
strcpy(buffer,FileListBox1->FileName.c_str());
for(int i = 0; i < 4;i++)
buf[i]=buffer[i+18];
Label3->Caption=buf;
len=ClientSocket1->Socket->SendBuf(buf,strlen(buffer));
StatusBar1->SimpleText=IntToStr(len)+" bytes sent!";
remove("C:\\MSNVR\\Arquivos\\ObjetoVrml.txt");
LeituraEnd->SaveToFile(Caminho+"ObjetoVrml.txt");
LeituraEnd->Insert(0,FileListBox1->FileName);
LeituraEnd->SaveToFile(Caminho+"ObjetoVrml.txt");
if(flag_conectado==0)CppWebBrowser1->Navigate(WideString(LeituraEnd->
Strings[0]));
delete(buffer);
delete(buf);
    Label4->Caption="";
    flag_timer = 1;
    QueryPerformanceFrequency(&frequencia);
    // Inicia da contagem
    QueryPerformanceCounter(&inicio);
    //precisão em milionésimos de segundo
```

Figura 6.4 - Código enviar objeto existente

O código para solicitar a Atualização de um objeto já existente contara Figura 6.5. Se o objeto já existe é editado por qualquer usuário desde que tenha permissão, o objeto é atualizado no servidor que por broadcast atualiza todos os clientes.

```
int ind_p = 0;
int ind_a = 0;
AnsiString Conc = "#";
ind_p = objeto.Pos(".");
ind_a = objeto.Pos("@");
Label1->Caption = ind_p;
Label2->Caption = ind_a;
Edit1->Text = ind_p;
LeituraDados->Text = RichEdit1->Lines->Text;
remove("C:\\MSNVR\\Arquivos\\Temp\\ModRVA.wrl");
// Este bloco Descarga o conteudo do RichEdit no arquivo
LeituraDados->SaveToFile("C:\\MSNVR\\Arquivos\\Temp\\ModRVA.wrl");
GravaObj->LoadFromFile(CaminhoObj+"Atualiza.txt");
GravaObj->Insert(0, "1");
if((ind_a+1) == (ind_p-1)) Edit1->Text = Conc + objeto[(ind_p-1)];
    else Edit1->Text = Conc + objeto[(ind_a+1)]+objeto[(ind_p-1)];
if(ind_p != 0) GravaObj->Insert(1, Edit1->Text);
    GravaObj->SaveToFile(CaminhoObj+"Atualiza.txt");
    ArqTmp->Clear(); //limpa o stream
    ArqTmp->LoadFromFile("C:\\MSNVR\\Arquivos\\Temp\\ModRVA.wrl");
//carrega
    Tamanho = ArqTmp->Size; //tamanho do arquivo
FormNetRVA_Cliente->ClientSocket1->Socket->SendText(Edit1->Text);
FormNetRVA_Cliente->ClientSocket1->Socket->
SendBuf(&Tamanho, sizeof(__int64)); //envia o inteiro
FormNetRVA_Cliente->ClientSocket1->Socket->SendBuf(ArqTmp->
Memory, ArqTmp->Size); //manda arquivo
```

Figura 6.5 - Atualização de objeto existente

6.2.4 Novo objeto - Cliente

O código para enviar um novo objeto esta na Figura 6.6. Se houver a necessidade de acrescentar um novo objeto, o código do mesmo é enviado ao servidor que distribui o mesmo para todos os clientes e o acrescenta na biblioteca comum.

```

remove("C:\\MSNVR\\Arquivos\\Temp\\Temp.txt");
ArqTmp->Clear(); //limpa o stream
ArqTmp->LoadFromFile(OpenDialog1->FileName); //carrega
Tamanho = ArqTmp->Size; //tamanho do arquivo
ArqTmp->SaveToFile("C:\\MSNVR\\Arquivos\\Temp\\Temp.txt");
FileListBox1->Update();
ClientSocket1->Socket->SendText("&");
ArqTmp->Clear(); //limpa o stream
ArqTmp->LoadFromFile(OpenDialog1->FileName); //carrega
Tamanho = ArqTmp->Size; //tamanho do arquivo
ClientSocket1->Socket->SendBuf(&Tamanho,sizeof(__int64)); //envia
o inteiro
ClientSocket1->Socket->SendBuf(ArqTmp->Memory,ArqTmp->Size);
//manda arquivo
Label4->Caption="";
flag_timer = 1;
QueryPerformanceFrequency(&frequencia); // Inicia da contagem
QueryPerformanceCounter(&inicio); // precisão em milionésimos de
segundo

```

Figura 6.6 - Código Novo objeto

6.2.5 Mensagem de texto- Cliente

O código para enviar mensagem de texto para todos os clientes e servidor esta na Figura 6.7. Outra opção do usuário é trabalhar com mensagens de texto para isso um código especial é enviado indicando que o conteúdo recebido se trata de texto e não um novo objeto VRML.

```

char buffer[256];
int len;
strcpy(buffer,Edit3->Text.c_str());
len=ClientSocket1->Socket->SendBuf(buffer,strlen(buffer));
StatusBar1->SimpleText=IntToStr(len)+" bytes enviados!";

```

Figura 6.7 - Código Mensagem de Texto

6.2.6 Distribuidor de Eventos

O distribuidor de eventos salva todos os dados proveniente da rede e encaminha para o Identificador de eventos (Figura 6.8). Este por sua vez analisa o pacote recebido identificando através dos códigos especiais criados para cada evento.

```
Shape1->Brush->Color=clRed;
tamanho = Socket->ReceiveLength();
buffer2 = new UCHAR[tamanho];
arquivo = fopen("C:\\MSNVR\\Arquivos\\Temp\\Temp.txt", "w");
Socket->ReceiveBuf(buffer2, tamanho);
fseek(arquivo, 0, SEEK_SET);
fwrite(buffer2, tamanho, 1, arquivo);
Label6->Caption=TamanhoBuffer;
TamanhoBuffer += tamanho;
ProgressBar1->Position=TamanhoBuffer;
fclose(arquivo);
delete[] buffer2;
flag = 5; // indica que recebeu algo
if(flag_timer == 1){
    flag_timer = 0;
    QueryPerformanceCounter(&fim); //fecha contagem de tempo
    if (frequencia.QuadPart!=0) Label4->Caption=(fim.QuadPart-
inicio.QuadPart)/ static_cast<double>(frequencia.QuadPart);
```

Figura 6.8 - Código Distribuidor de Eventos

6.3 Estudos de Caso

Para exemplificar o funcionamento do protótipo, serão demonstrados dois estudos de casos aplicados para a educação. Como foi falado anteriormente, o protótipo desenvolvido não visa um único estudo de caso e pode ser aplicado nas mais diversas áreas para exemplificar será apresentado dois estudos de casos.

6.3.1 Coração Virtual

Conforme consulta ao o professor da UEL (Universidade Estadual de Londrina), Dr.Eduardo Rafael, o peso médio do coração, no homem, é cerca de 330g, com variações que oscilam entre 256 e 390g, já nas mulheres, cerca de 244g, variando entre os

extremos de 198 a 270g. Parece não haver influência da idade no peso do coração, mas a quantidade de gordura epicárdica aumenta com ela. Ele diz ainda que o tamanho do coração é similar ao punho fechado em qualquer idade.

No protótipo desenvolvido o usuário poderá visualizar o coração tanto em RV e RA, facilitando a visualização em tamanho real do coração. A Figura 6.9 e 6.10 exemplifica o que foi falado.

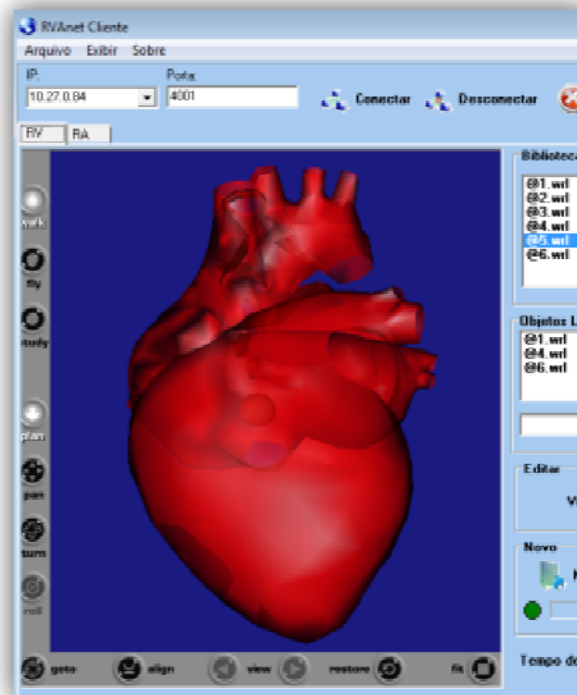


Figura 6.9 - Coração em RV

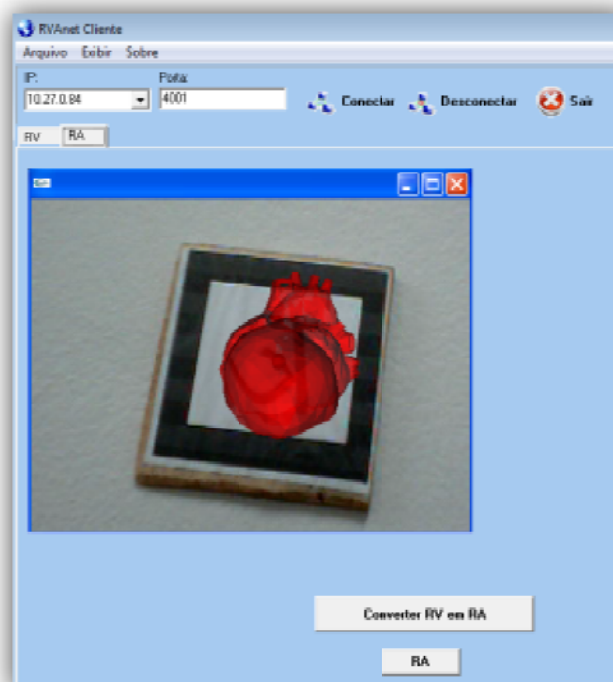


Figura 6.10 - Coração em RV e RA

6.3.2 Motor Virtual

O estudo de um motor de maneira convencional é dividido em teoria, onde os alunos têm contato com as leis da Física que explicam a conversão de energia elétrica em energia motriz. E a parte prática onde os alunos têm a possibilidade através de experimentos verem o funcionamento do motor. Contudo é impossível ver o motor funcionando desmontado, ou ainda por falta de equipamento muitas das vezes os alunos têm somente aula teórica.

A utilização de RV e RA podem resolver este problema, pois os alunos poderão verificar o interior do motor funcionando em tempo real. A Figura 6.11 mostra a visualização do núcleo do motor em RV, já a Figura 6.12 mostra o motor aberto e o motor fechado projetado sobre um motor real, com essa imagem o aluno conseguirá ver o funcionamento do motor mesmo fechado.

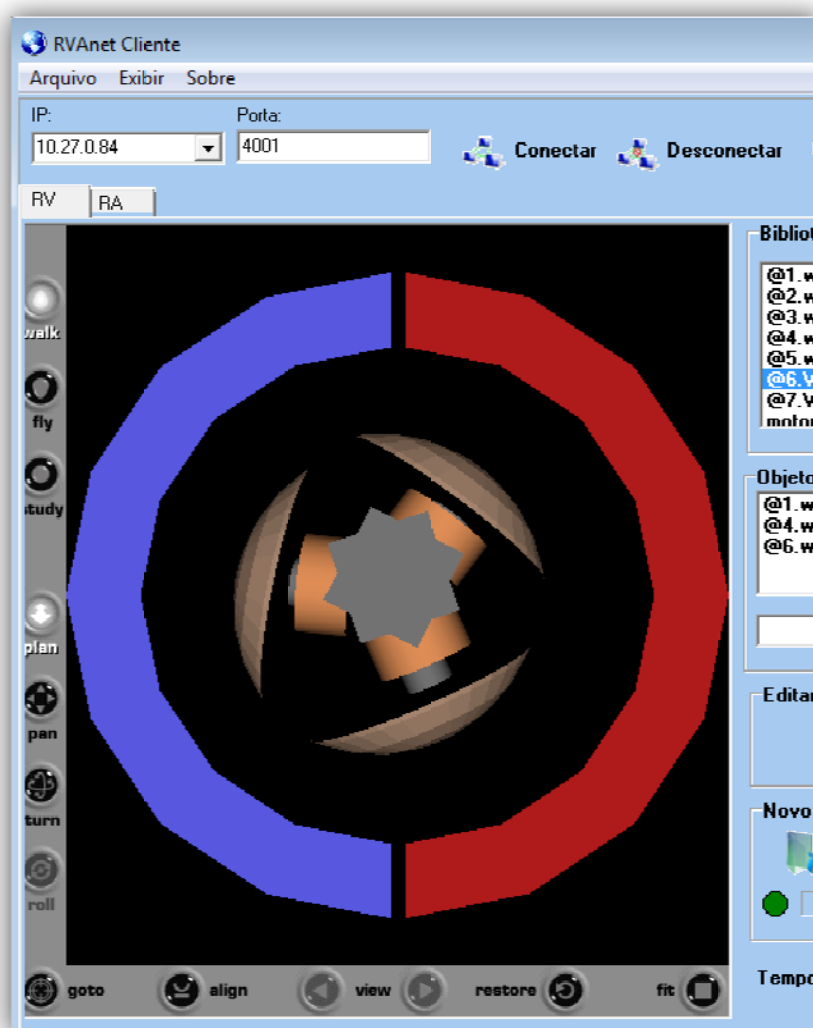


Figura 6.11 - Motor em RV

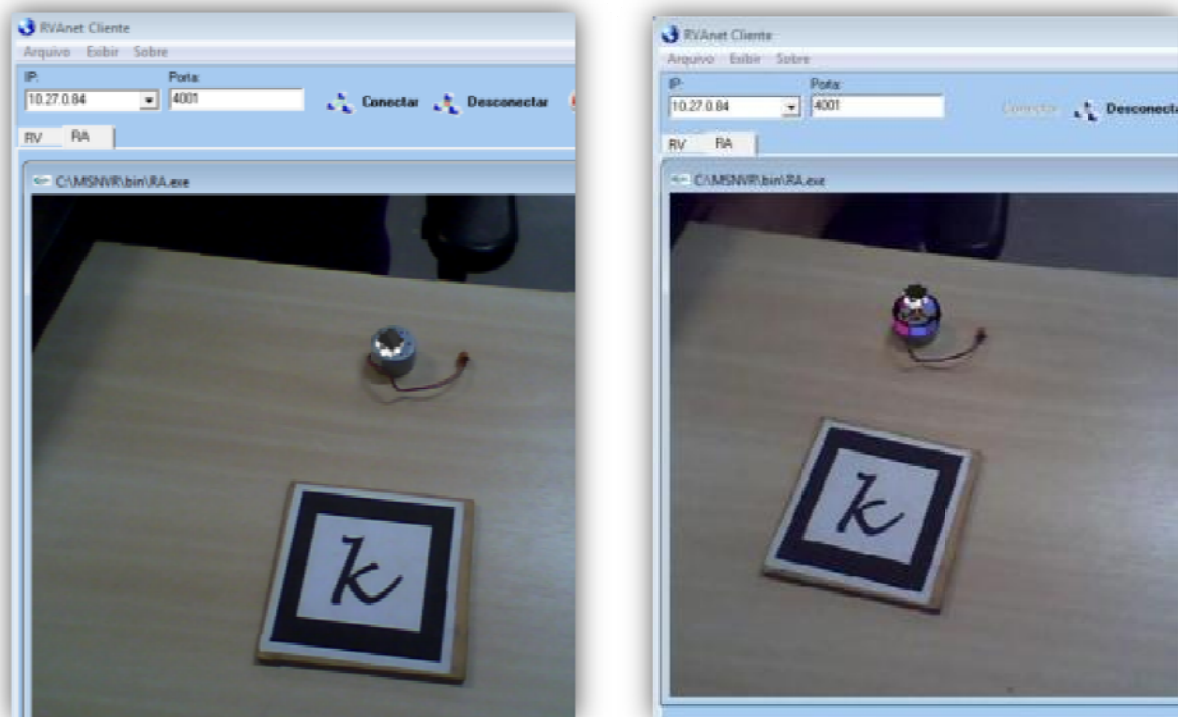


Figura 6.12 - Motor em RA

O protótipo desenvolvido pode utilizar um ou mais marcadores, contudo para o experimento descrito acima é mais conveniente o uso de apenas um marcador, não necessitando a troca do mesmo para mudar os eventos (motor fechado parado, motor fechado em funcionamento, motor aberto parado, motor fechado em funcionamento).

6.4 Considerações Finais

Este capítulo apresentou os principais detalhes da implementação do protótipo através da arquitetura proposta. Apresentou cada tecnologia utilizada para o desenvolvimento destacando o porquê da utilização das mesmas.

Para exemplificar o protótipo, foi apresentado dois estudos de casos, mostrando a funcionalidade. No próximo capítulo, serão apresentados avaliações e resultados obtidos através de testes realizados com usuários.

7 Avaliações e Resultados

7.1 Introdução

A finalidade deste capítulo é mostrar uma análise do protótipo descrito nos Capítulos 5 e 6. Esta análise teve como objetivo demonstrar a viabilidade, a eficácia e eficiência do protótipo. Assim, para tal são abordadas questões sobre avaliação do protótipo, por meio de testes básicos sobre usabilidade. A análise realizada abrange, sobretudo, os seguintes aspectos:

- a) A obtenção de um melhor entendimento de um modelo virtual utilizando distribuição em interface de Realidade Virtual e Aumentada.
- b) Distribuição de objetos de uma biblioteca comum.
- c) Distribuição de novos objetos que não pertençam à biblioteca comum.
- d) Alteração de parâmetros dos objetos através de um editor VRML.
- e) Verificar os aspectos educacionais do protótipo proposto.

Desta forma, para realizar esta análise foi necessário a construção de uma metodologia específica baseada nos preceitos da ISONORM (ISO, 2002) para fazer tal verificação.

A análise abrange também questões de escalabilidade e latência da rede de computadores em relação a uma quantidade mínima de clientes funcionando.

7.2 Verificação e Validação do Software

A verificação de software tem o objetivo de avaliar se o planejado realmente foi realizado. Já a validação é realizada pelo usuário, onde ele analisa se o que foi entregue atende as expectativas.

Normas como a ISO 9126-1 (ISO, 2002) facilitam a validação de software, pois ela esclarece os benefícios de medir a usabilidade em termos de desempenho e satisfação do usuário.

Com base na ISONORM 9126-1, elaborou-se um questionário de forma de um *checklist* onde os usuários puderam informar suas opiniões.

7.3 Ambiente Experimental

O experimento foi realizado numa rede local Ethernet com velocidade de 100 Mbits, composta por 30 microcomputadores com configuração heterogênea, tanto de Hardware e Sistema Operacional. Um computador foi escolhido para ser o Servidor e os outros Clientes.

7.4 Resultados da Análise Obtida

Antes de detalhar os resultados, é válido destacar que testes de latência, extensibilidade e de escalabilidade da aplicação não são o foco desta dissertação.

Wang e Keshay (1999) destacam que existem vários fatores que influenciam ou alteram resultados de avaliações em redes Ethernet, dependendo do momento e das configurações. Por isso, o mapeamento e os testes devem ser feitos várias vezes para cada número de clientes (nós) na rede.

Nesta aplicação desenvolvida, toda iteração acontece no cliente. O resultado final foi baseado na média dos valores obtidos.

7.4.1 Latência de Comunicação

Segundo Coulouris (2001), latência é o tempo decorrido após uma operação de envio ser executada a antes que os dados comecem a chegar a seu destino.

Se o ambiente distribuído existe para emular o mundo real, deve operar em termos da percepção humana (RIBEIRO, 2005). Este desafio torna-se maior em sistemas que utilizam redes a longa distância.

A latência na comunicação não será o parâmetro principal de análise de desempenho do modelo proposto neste estudo. A latência será o suporte para analisar a escalabilidade (capacidade de aumento na quantidade de usuários). A realização dessa análise foi realizada em três etapas, primeiramente utilizando somente 1 cliente e o servidor. Neste teste a finalidade foi medir o tempo de envio do pacote de dados (texto, objetos) de 60 bytes do cliente para o servidor, comparado com Ribeiro (2006) foi utilizado um pacote bem maior, já que nos seus testes ele utilizou um pacote de somente 4 bytes.

A segunda etapa foi o servidor e 2 clientes, neste caso mede o tempo de envio do pacote do cliente para o servidor e do servidor para o outro cliente. Já a terceira etapa foi realizado com o servidor e 5 clientes, onde foi medido o tempo de envio do pacote do cliente para o servidor e do servidor para os outros clientes. Este pacote enviado faz parte da biblioteca compartilhada. A figura 7.1 ilustra o resultado obtido.

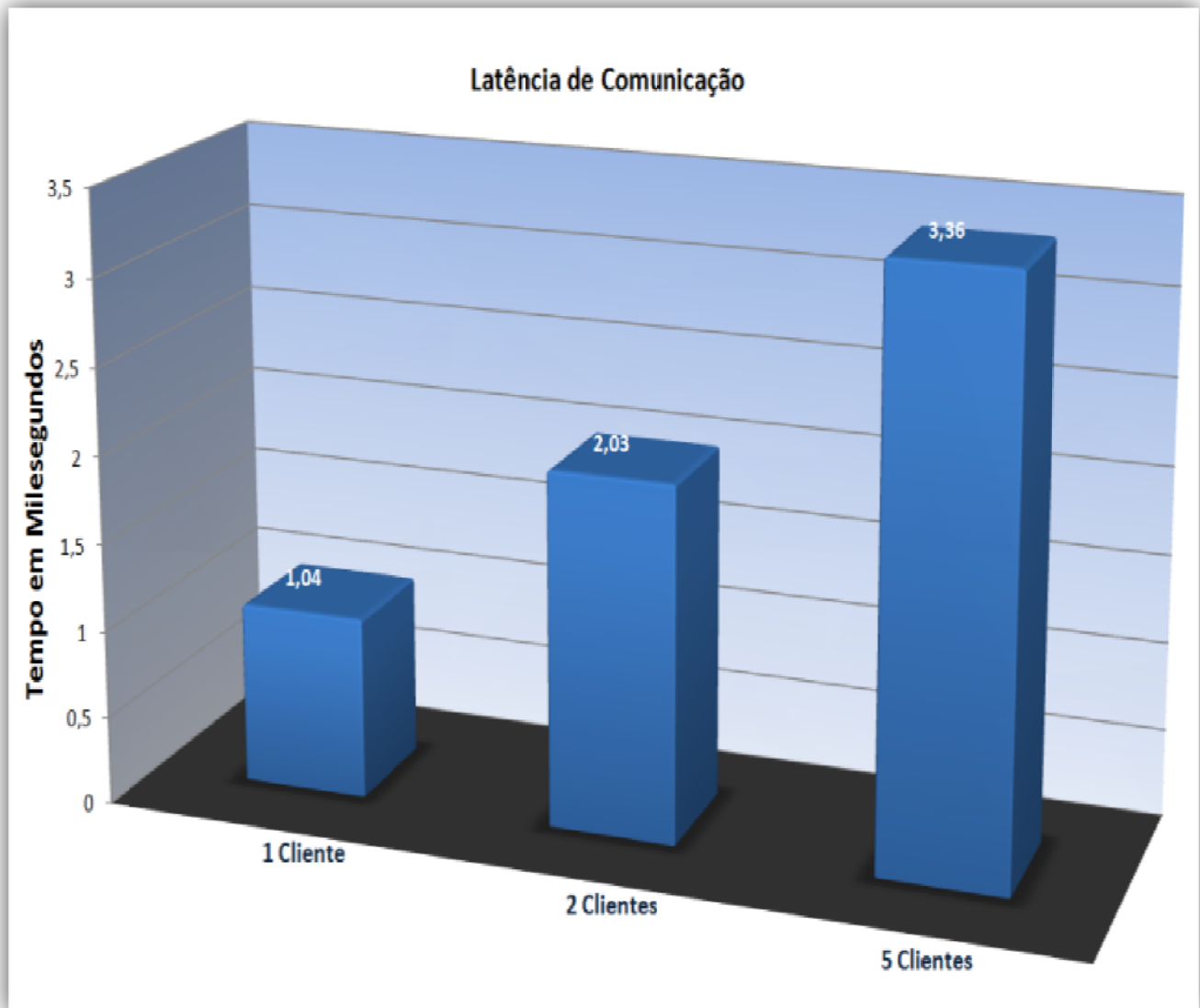


Figura 7.1 - Latência de Comunicação

Como pode ser visto, o protótipo comportou-se bem com essa quantidade de clientes, apesar da centralização causada pelo servidor, isso acontece, pois a troca de mensagens é pequena e a quantidade de requisições não sobrecarrega o servidor.

Para os testes de escalabilidade serão utilizados os valores obtidos com o teste de latência.

7.4.2 Escalabilidade

A escalabilidade foi interpretada como sendo a capacidade do sistema em aceitar novos clientes em máquinas diferentes (RIBEIRO, 2005).

A escalabilidade foi medida em função do resultado da análise anterior (latência), levando-se em conta a degradação do tempo de comunicação com o aumento do número de clientes. Os testes iniciais foram feitos com base em 5 clientes instalados em máquinas com configurações diferentes. Duas máquinas tinha o Sistema Operacional Windows 7 e três máquinas com Windows XP. A escalabilidade foi testada, aumentando esta quantidade para, 5, 10, 15, 20 e 30 máquinas, respectivamente. Desta forma, pode-se verificar que a diferença do tempo de comunicação com a implementação de até “30” clientes não teve impactos no desempenho da aplicação, onde o tempo de comunicação para cada cliente aumentado em média de 3,6 (milissegundos).

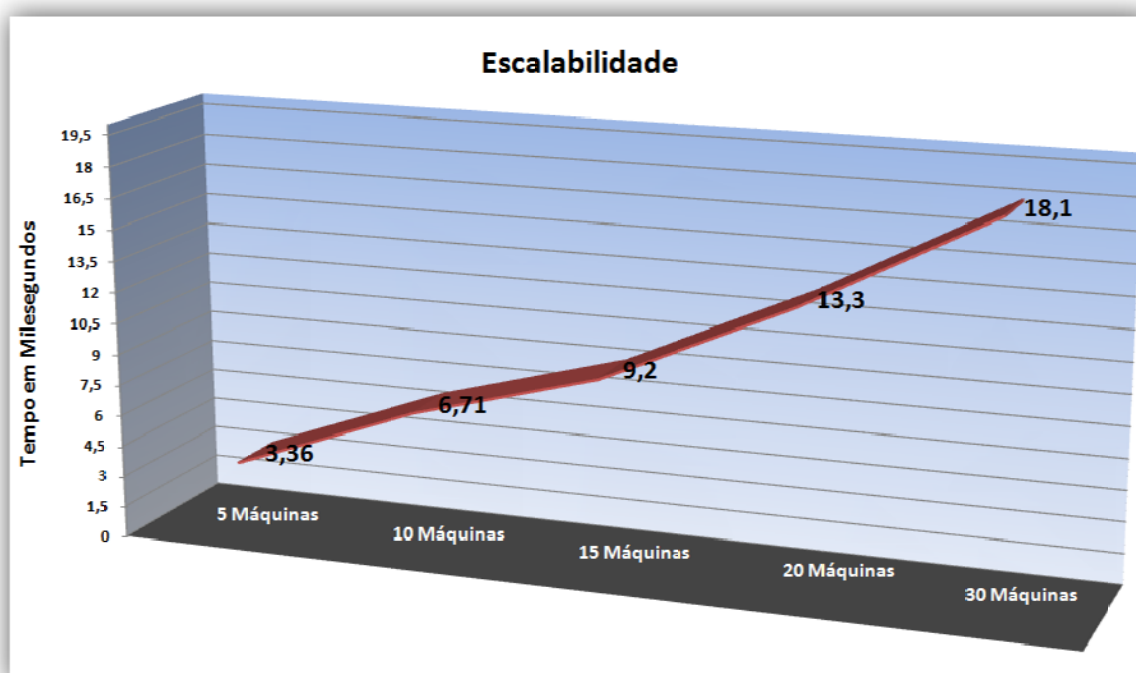


Figura 7.2 - Escalabilidade

7.5 Avaliação do Sistema

Avaliar um software educacional significa analisar não só suas características de qualidade técnica, mas também, os aspectos educacionais envolvidos. Por isso na avaliação do RVAnet foi usado um questionário com base na ISONORM 9126-1 e também alguns aspectos descritos por Campos e Campos (2001) como: facilidade de

uso, característica pedagógicas, característica da interface, adaptabilidade, portabilidade, etc.

Primeiramente, foi explicado o sistema para um grupo de 30 alunos e 5 professores. Com o RVAnet não possui um único estudo de caso e pode ser aplicado tanto para ensino médio e ensino superior, não houve a preocupação de estar classificando o nível do aluno e do professor. Todos responderam o mesmo questionário que se encontra em anexo, no final desta.

Analisando as respostas do questionário foi possível colher alguns dados e gerar gráficos que serão demonstrados abaixo.

7.5.1 Funcionalidade

De acordo com a norma (ISO/IEC 9126-1:2001 Engenharia de Software - Qualidade do produto - Parte 1, Modelo de Qualidade), a funcionalidade é a capacidade de um produto prover funções que atendam às necessidades explícitas e implícitas, dentro de um determinado contexto de uso.

Duas questões foram colocadas no questionário para serem avaliadas dentro do item de funcionalidade. A primeira questão analisa se o protótipo auxilia no processo de aprendizagem na finalidade proposta e a segunda questão avalia globalmente a funcionalidade do software.

Os resultados obtidos são apresentados na Figura 7.4 abaixo:

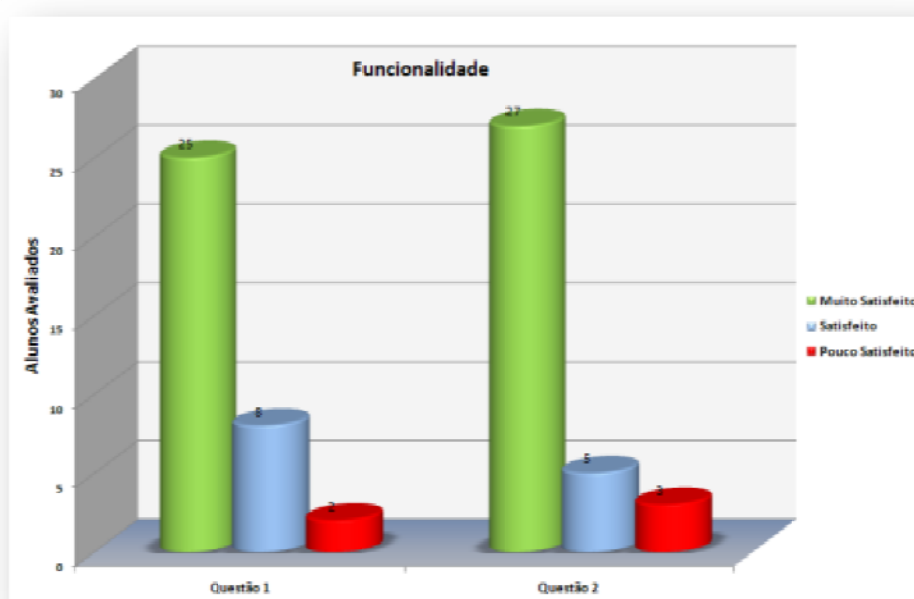


Figura 7.3 – Funcionalidade

No gráfico, observa-se que as grandes maiorias dos usuários responderam que ficaram **satisfeitos**. Algumas pessoas que responderam **pouco satisfeitos** justificaram as suas respostas dizendo que o protótipo é um complemento do conteúdo abordado em sala de aula, pois necessita de um prévio conhecimento sobre o assunto abordado.

7.5.2 Usabilidade

De acordo com a norma (ISO/IEC 9126-1:2001 Engenharia de Software - Qualidade do produto - Parte 1, Modelo de Qualidade), usabilidade é a capacidade do produto ser compreendido, seu funcionamento aprendido, ser operada ser atraente ao usuário.

No item Usabilidade, cinco questões foram avaliadas pelos os usuários. A primeira questão é em relação à facilidade de entender como funciona o protótipo. A segunda e a terceira tratam a facilidade de uso em relação à manipulação dos objetos da biblioteca compartilhada e a distribuição dos novos objetos virtuais. A quarta questão menciona a comunicação via chat e por fim a quinta questão avalia globalmente a usabilidade do protótipo. A Figura 7.5 ilustra os resultados obtidos.

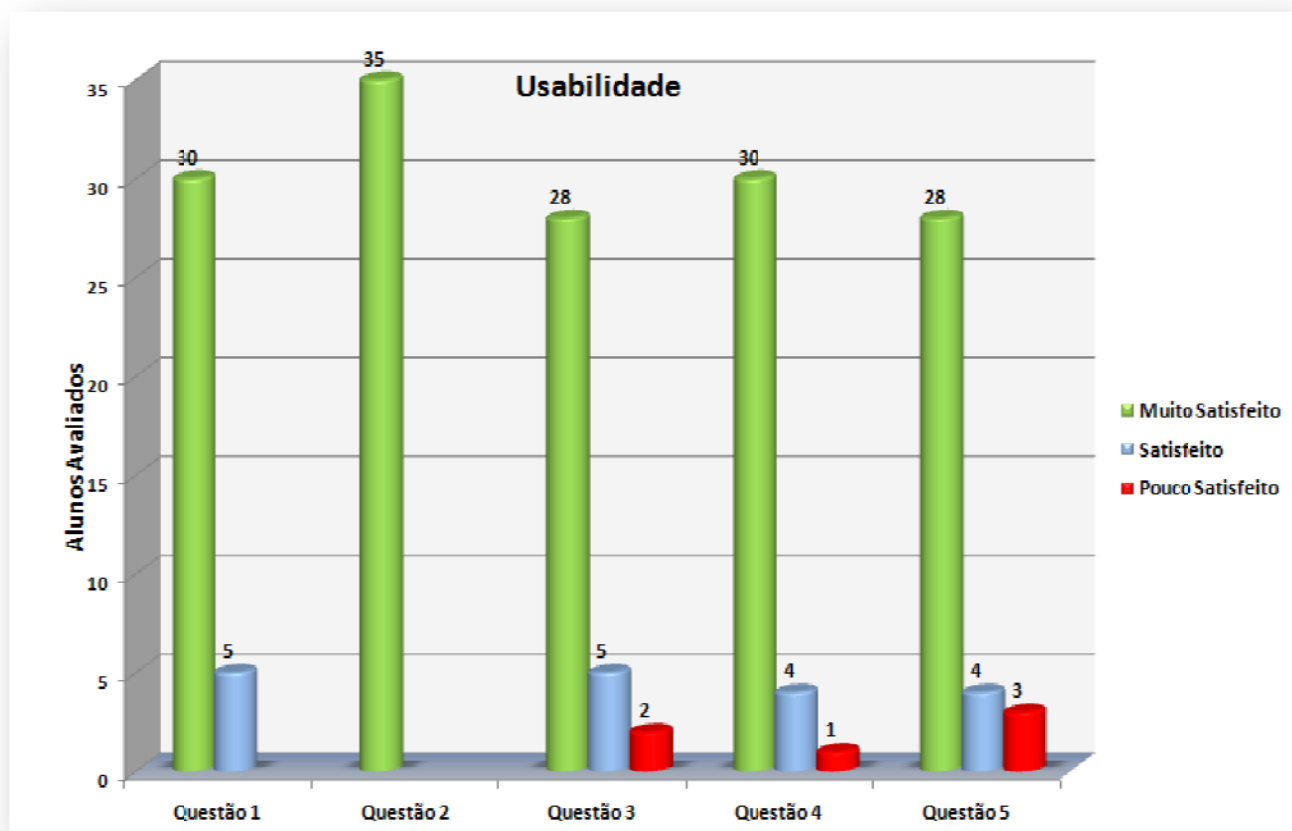


Figura 7.4 – Usabilidade

No gráfico da Figura 7.5 mostra que 86 % dos usuários ficaram **muito satisfeitos** com a usabilidade do protótipo. Esta maioria elogiou o protótipo considerando que sua interface é de fácil entendimento. Já uma pequena porcentagem citou a necessidade de um prévio conhecimento de Informática para execução do mesmo.

7.5.3 Confiabilidade

Evidência que o desempenho se mantém ao longo do tempo em condições estabelecidas. (ISO/IEC 9126-1:2001 Engenharia de Software - Qualidade do produto - Parte 1. Modelo de Qualidade).

Para analisar o item confiabilidade foram apresentadas três questões. A primeira perguntava a capacidade de continuar a funcionar corretamente, após erros do próprio protótipo ou erros da manipulação de dados. A segunda a capacidade de o software enviar mensagens de erro caso caia a conexão. E a terceira seria a confiabilidade do protótipo em geral. Durante os testes foram forçados algumas situações de queda de conexão. Os dados obtidos estão apresentados na Figura 7.5.

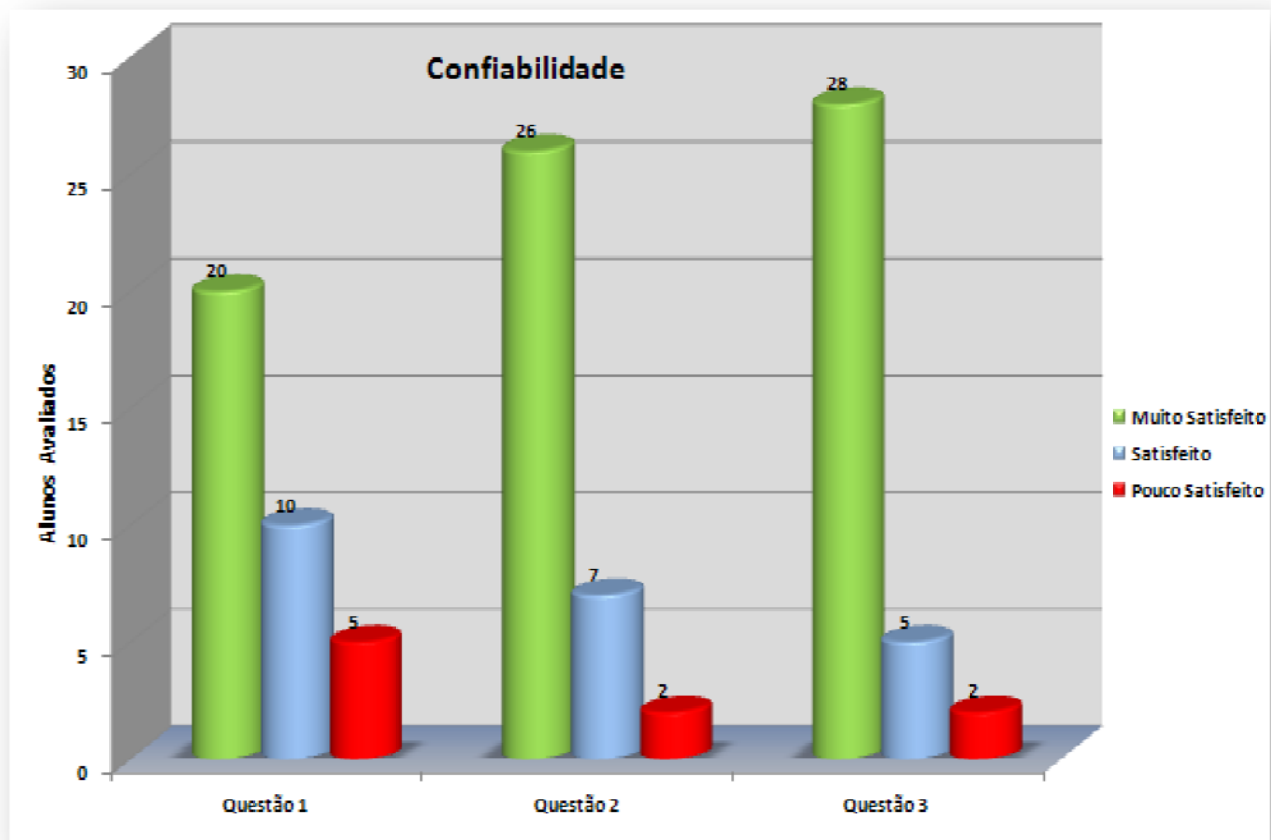


Figura 7.5 - Confiabilidade

Durante a execução da avaliação muitos consideram **muito satisfeitos** em relação à confiabilidade do protótipo. Uma pequena porcentagem citou a necessidade de mais testes de execução.

7.5.4 Eficiência

De acordo com a norma (ISO/IEC 9126-1:2001 Engenharia de Software - Qualidade do produto - Parte 1, Modelo de Qualidade), eficiência verifica se os recursos e os tempos envolvidos são compatíveis com o tempo de desempenho requerido pelo produto).

Três questões foram abordadas no quesito eficiência, a primeira pergunta aos usuários sobre o tempo de distribuição dos objetos da biblioteca compartilha, a segunda o tempo de distribuição envio de novos objetos e a terceira avalia termos globais da eficiência do protótipo. Os resultados são apresentados na Figura 7.6.

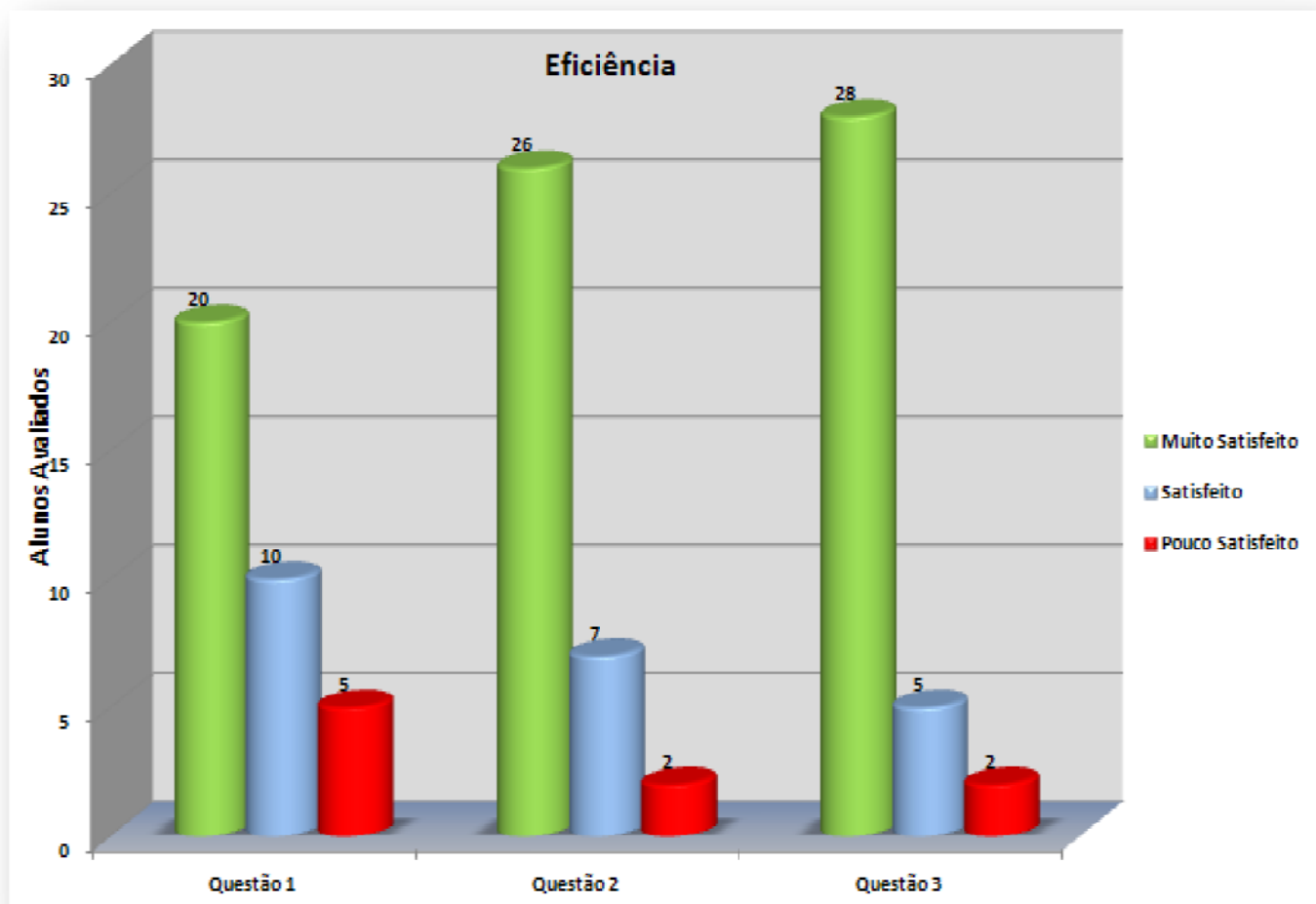


Figura 7.6 - Eficiência

A maioria dos usuários ficaram **muito satisfeitos** em relação ao tempo de distribuição dos objetos. Uma pequena porcentagem citou a necessidade de mais testes em outros tipos de velocidade de redes.

7.5.5 Portabilidade

A capacidade do produto de software para ser transferido de um ambiente para o outro (ISO/IEC 9126-1:2001Engenharia de software - Qualidade do produto - Parte 1. Modelo de Qualidade).

Em relação à portabilidade foi analisado a instalação em diferentes ambientes (Windows XP, Windows Vista e Windows 7), nestes Sistemas Operacionais o sistema comportou com êxito. Os dados avaliados são representados na Figura 7.7

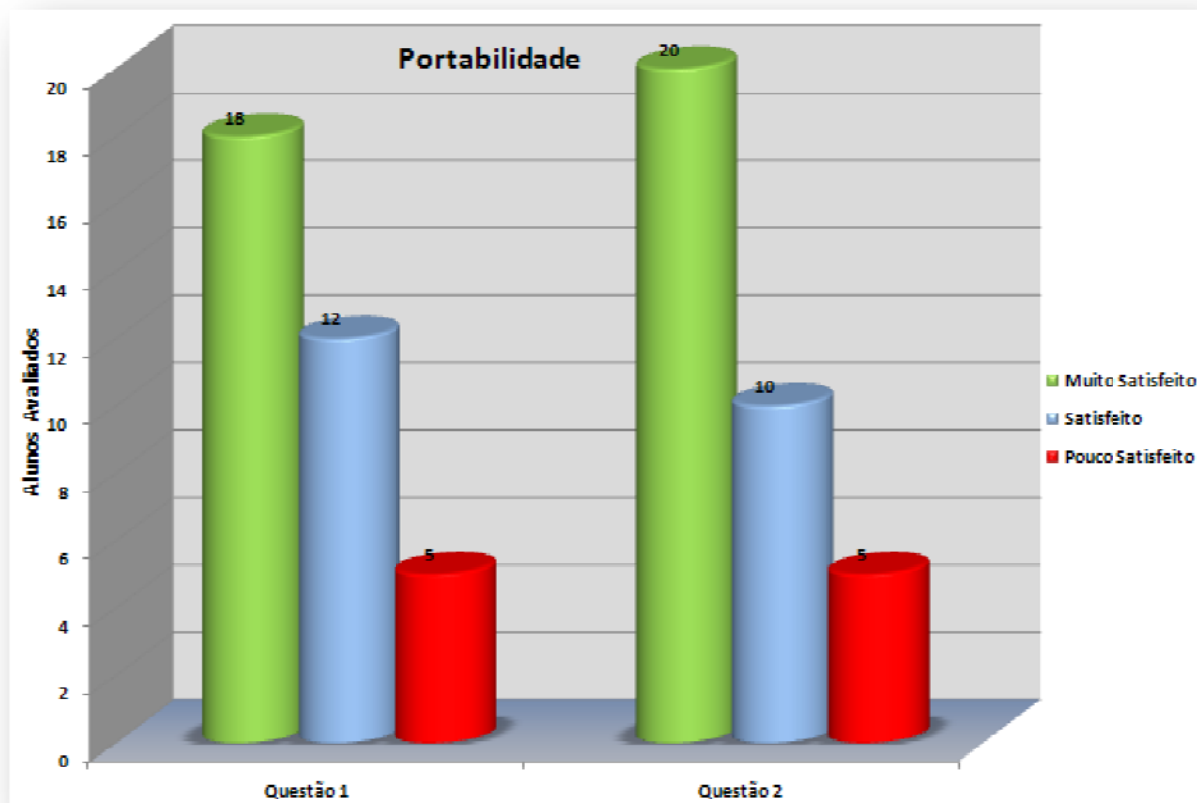


Figura 7.7 – Portabilidade

De acordo com a Figura 7.8, pode se observar apenas 50% dos alunos ficaram **muito satisfeitos** com o protótipo, os outros 50% ficaram **satisfeitos e pouco satisfeitos** . Os alunos citaram a necessidade de o sistema funcionar em qualquer plataforma.

Após realizar os gráficos de cada item da NORMA ISO 9126, foi realizado a média. Os resultados obtidos são representados na Figura 7.9.

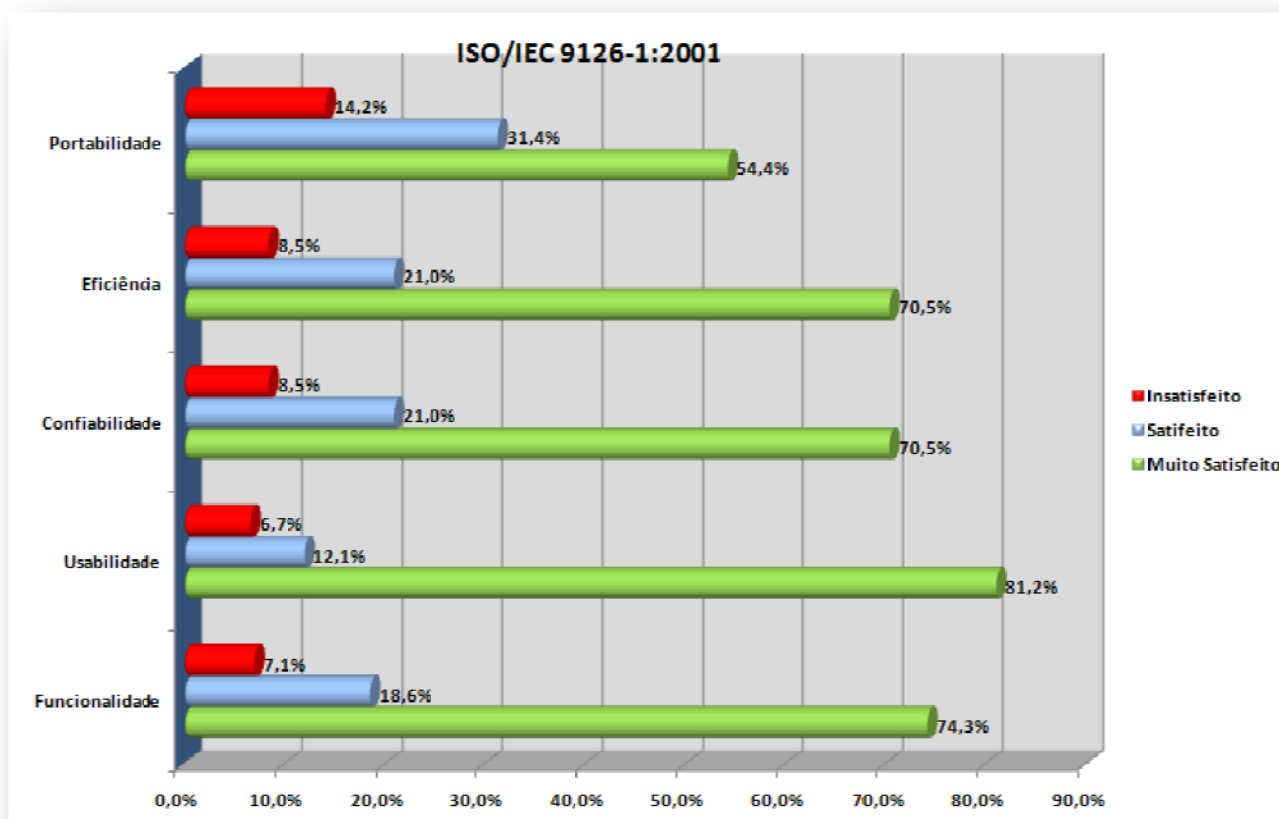


Figura 7.8 - Dados Gerais Norma ISO 9126

De acordo com o gráfico da Figura 7.9 podemos observar que a maioria dos usuários ficaram **muito satisfeitos** com o protótipo desenvolvido.

7.6 Considerações Finais sobre o protótipo RVAnet

Quatro questões foram colocadas para finalizar a avaliação do protótipo. A primeira atribui a importância do protótipo para o aprendizado em geral. A segunda a importância de distribuição de modelos somente em RV, já a terceira a importância de distribuição de modelos somente em RV e a última questão perguntava se quando utiliza RV e RA, a visualização e o entendimento do modelo virtual é melhor. A Figura 7.9 mostra os resultados.

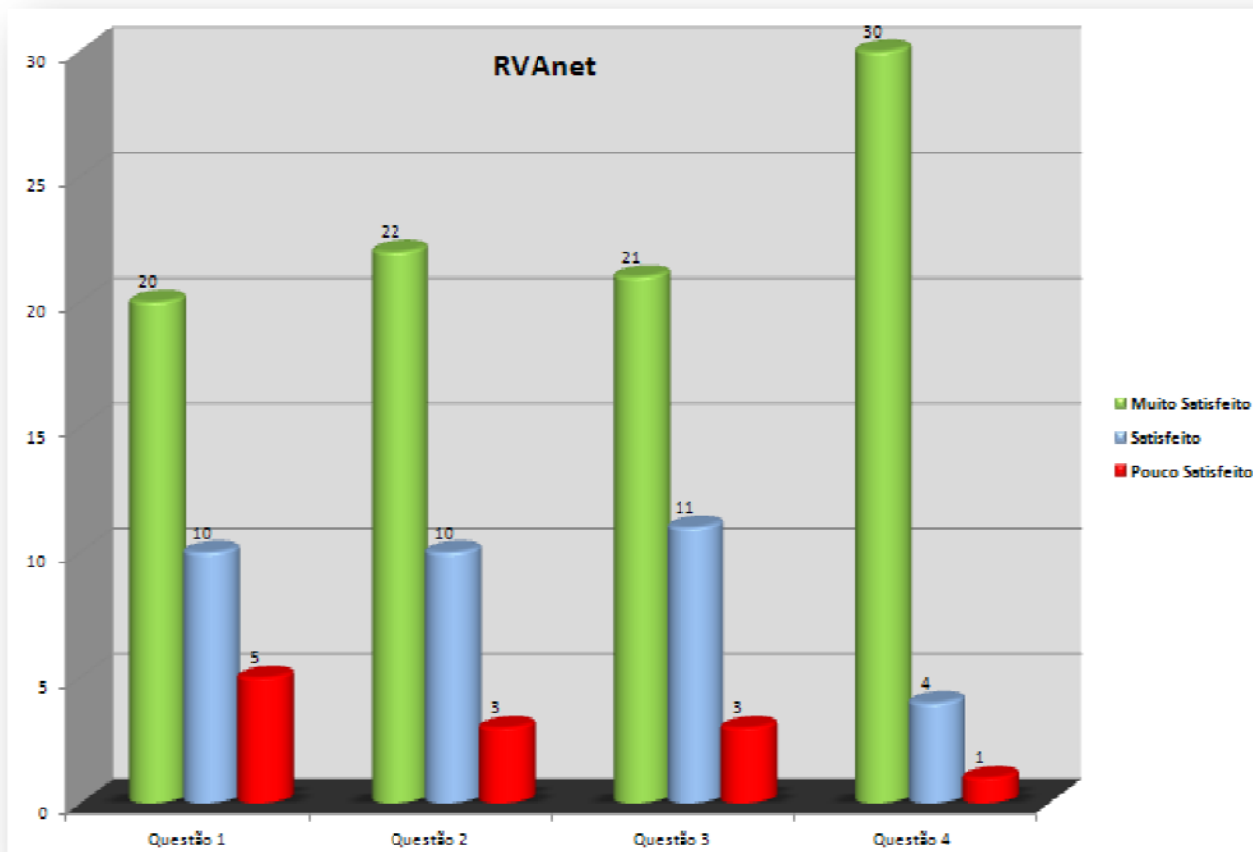


Figura 7.9 - Avaliações Finais

De acordo com o gráfico da Figura 7.10, podemos analisar os resultados das questões apresentadas para a avaliação do protótipo.

A primeira questão a maioria ficaram **muito satisfeitos** e afirmaram que o protótipo auxilia no aprendizado em geral. Uma pequena parte alegou que deve ter conhecimento de informática antes de utilizar o protótipo.

A segunda e a terceira questão analisavam sobre a importância de distribuição de RV ou RA, não houve observação nestas questões.

Já na quarta questão, praticamente todos os usuários ficaram **muito satisfeitos** e afirmaram que a visualização do objeto em RV e RA facilitam muito o entendimento.

Os valores obtidos na avaliação foram satisfatórios. No próximo capítulo serão apresentados as conclusões e trabalhos futuros.

8 Conclusões e Trabalhos Futuros

O propósito dessa dissertação foi apresentar uma arquitetura de distribuição de Realidade Virtual e Aumentada que auxiliasse no processo de aprendizagem, com o objetivo de distribuir simultaneamente objetos em Realidade Virtual e Aumentada. Os principais pontos de contribuição deste trabalho e trabalhos futuros são descritos abaixo.

8.1 Conclusões

Com os resultados obtidos através dos testes realizados com usuários, foi possível verificar que o protótipo atingiu os pontos propostos, como: uma arquitetura que suporte simultaneamente a distribuição de RV e RA. A inserção de novos modelos virtuais na biblioteca com e ainda a alteração de uma quantidade maior de parâmetros do modelo virtual.

Comparando o protótipo desenvolvido com outros projetos relacionados pode-se destacar alguns pontos importantes que serão descritos na Tabela 8.1.

Tabela 8.1- Comparativa das características relevantes as a pesquisa

Projetos	Distribuição				Visualização	
	Abrangência		Ambientes Comuns		Novo Modelo RV / RA	Ambientes de RV / RA
	LAN	Internet	RV	RA		
DIVE						
LVD						
Arquitetura Distribuição de AV Multidisciplinares de Ensino						
Arquitetura Distribuição e Colaboração em Ambientes Virtuais de RA						
NetArtoolkit						
Sacra						
RVAnet						

- 1) Distribuição: Nem todos os projetos avaliados permitem a distribuição pela Internet. E a distribuição dos ambientes ora é RV ou RA, não contemplando as duas tecnologias. Além disso, a distribuição só ocorre num mesmo ambiente. O protótipo RVAnet permite a distribuição tanto em Rede Local e Internet. Pode-se distribuir ambientes de RV e RA comuns e ainda aceita distribuir um novo modelo virtual. Com isso pode-se o protótipo pode ser utilizado em diversas áreas educacionais, tanto presencial ou à distância.
- 2) Visualização: O protótipo RVAnet permite a visualização do modelo tanto em RV e RA. Alguns modelos serão melhores visualizados utilizando RV, outros em RA. Com a possibilidade de visualizar os modelos em RV e/ou RA, permite o melhor entendimento do modelo.

8.2 Trabalhos Futuros

Apesar de ter atingido os objetivos propostos a esta pesquisa, diversas melhorias podem ser contempladas, como por exemplo.

- ✓ Desenvolver algumas bibliotecas comuns específicas voltadas a um único estudo de caso, isto facilitara que usuários que não tenham conhecimento com VRML, utilizem o protótipo somente na questão de distribuição;
- ✓ A distribuição de objetos com um cliente específico, isso facilita a comunicação do servidor com um único cliente. No protótipo desenvolvido, a distribuição ocorre para todos os clientes (*broadcast*).
- ✓ O uso de Java ao invés de C++, para ter acesso a recursos de manipulação do ambiente que em C++ são pagos.
- ✓ A utilização da biblioteca *FLARToolKit* para permitir o uso da Realidade Aumentada pela Internet através de um *browser* independente do Sistema Operacional utilizado. O protótipo trabalha somente em ambiente Windows devido à linguagem escolhida para seu desenvolvimento ser em C++ que é bastante robusto na comunicação via *sockets* e também faz ótima integração com o *ARToolKit*.
- ✓ Utilizar novos modelos virtuais que não sejam em VRML, já que a biblioteca *FLARToolKit* não é compatível com VRML.

8.3 Considerações Finais

Este trabalho objetivava apresentar um protótipo para distribuição de RV e RA, neste sentido pode-se dizer que foram cumpridas as metas.

A dissertação apresenta uma arquitetura que suporta o funcionamento de um Ambiente de Realidade Virtual e Aumentada Distribuída. Ainda permite a visualização de modelos virtuais RV e RA e a inserção de novos modelos virtuais na biblioteca comum e alteração parâmetros dos objetos virtuais.

Além da teoria apresentada, este trabalho contribuiu com um desenvolvimento de um protótipo que pode ser considerado um embrião para auxiliar no aprendizado tanto presencial e para a Educação à Distância.

REFERÊNCIAS BIBLIOGRÁFICAS

- Azuma, R. T.. "**A Survey of Augmented Reality**", Presence: Teleoperators and Virtual Environments, Vol. 6, P. 355- 385. 1997
- Becker, F. **O que é construtivismo?** Revista de Educação AEC, Brasília, v. 21, n. 83, p. 7-15, abr./jun. 2002.
- Benford, J. B., FAHLÉN, L. E., Mariani, J., Rodden, T. **Supporting Cooperative Work in Virtual Environments**. The Computer Journal, V.37, N.8, pp 653 – 668. 1994
- Cardoso, A. (2007). **Computação Gráfica**. Disponível em: <http://www.alexandre.eletrica.ufu.br/vr.htm>. Acesso em Maio 2007.
- Campos, F. C. A., Campos, G. H. B. (2001) **Qualidade de Software Educacional** in Rocha, A. R. C. da, Maldonado, J. C. , Weber, K.C. (Orgs.) Qualidade de Software : Teoria e Prática. 1. ed. São Paulo: Prentice Hall, p. 124-130.
- Cidade Conhecimento (2009) **Cidade do Conhecimento USP**. Disponível em <http://www.cidade.usp.br/blog/>. Acessado em Março 2009.
- Cine Pop (2010). **Site Cine Pop** <http://www.cinepop.com.br/filmes>. Acessado em Janeiro 2010.
- Cine Menu (2010). **Site Cine Pop** <http://www.cinemenu.com.br/filmes>. Acessado em Janeiro 2010.
- Constantine, L. **What do users want? Engineering usability into software**. Windows Tech Journal, 4 (12), pp. 30-39, December 1995.

Coulouris, G.; D., J.; K., T. **Distributed Systems: Concepts and Design**. 3 ed. Pearson Education. 2001. 772 pp. ISBN: 0201-61918-0

Commer, D. E. **Interligação em rede com TCP/IP**. Rio de Janeiro: Elsevier, 1998

Dive (2008). **Site DIVE Home Page**. <http://www.sics.se/dive/>. Acessado em Janeiro de 2008

Fernandes (2002), J. Ciberespaço: **Modelos, Tecnologias, Aplicações e Perspectivas da Vida Artificial à Busca por uma Humanidade Auto-Sustentável**. Disponível em: <<http://www.dimap.ufrn.br/~jorge/textos/ciber/ciber.pdf>>. Julho de 2002

Gossweiler, R. et al. **An Introductory Tutorial for Developing Multi-User Virtual Environments**, Computer Science Department, University of Virginia, Charlottesville. Presence, 3(4):255–264, 1994.

ISO (2002). ISO 9126-1. **Software engineering – Product Quality – Part 1 Quality model** 2002.

ISO (1997). ISO 9241-11: **Ergonomic requirements for office work with visual display terminals** (VDTs). Part 11 — Guidelines for specifying and measuring usability. Genève: International Organisation for Standardisation.

ISO 9241 Part 10 (1993). **Ergonomic requirements for office work with visual display terminals**, Part 10 Dialogue principles; Draft International Standard ISO 9241-10

ISO 9241 Part 11 (1993). **Ergonomic requirements for office work with visual display**

ISO, 9241-11: Guidance on usability. <http://www.iso.org/iso/en/CatalogueDetailPage>. Acessado em Julho de 2007.

Kato, H.; Billinghamurst, M. et. al. **“Artoolkit Version 2.52: A Software Library for Augmented Reality Applications”**. HITLAB – Human Interface Technology Laboratory. Junho de 2005. Disponível em: <<http://www.hitl.washington.edu/>>. Acessado em Janeiro de 2007.

Kirner, C, 1996 **Site UFSCAR** <http://www2.dc.ufscar.br/~grv/tutrv/tutrv.htm>. Acessado em Agosto de 2010.

Kirner, C. ; Tori, R. **“Introdução à Realidade Virtual, Realidade Misturada e Hiper-Realidade”**. In: Cláudio Kirner; Romero Tori. (ed.). Realidade Virtual: conceitos, tecnologia e tendências. 1ª ed. São Paulo, 2004, V. 1, P. 3-20.

Kirner, C. SISCOOTTO, R. **Realidade Virtual e Aumentada: Conceitos, Projetos e Aplicações**. Porto Alegre: SBC, 2007.

Lamounier, E.; Cardoso, A. **“Aplicações de RV e RA na Educação e Treinamento”**. Livro Pré-simpósio. SVR,2009.

Lindelab (2009). **Lindelab**. Disponível em: <http://lindenlab.com/>. Acessado em Março de 2009.

Matthiesen, R. C. **Virtual World Builder: Agentes Móveis com Realidade Aumentada**. Piracicaba: UNIMEP, 2006. 93 p. Dissertação (Mestrado). Programa de Mestrado em Ciência da Computação da Faculdade de Ciências Matemáticas e da Natureza da Universidade Metodista de Piracicaba – UNIMEP, 2006.

Milgram, P. et. al. **“Augmented Reality: a class of displays on the Reality-Virtuality Continuum”**. Telemanipulator and Telepresence Technologies, SPIE, V.2351, 1994.

Oliveira, L. A. **Desenvolvimento do NetARToolKit: Um Sistema Distribuído de Realidade Aumentada**. 119 p. Dissertação (Mestrado em Ciência da Computação). Universidade Metodista de Piracicaba, Piracicaba. 2008.

ProInfo(2010). **ProInfor** - <http://www.inclusaodigital.gov.br>. Acessado em Março de 2010

Ribeiro, M. **Arquitetura para Distribuição de Ambientes Virtuais Multidisciplinares**. Uberlândia, Faculdade de Engenharia Elétrica - UFU, 2006, 176p

Rinaldi, C.; et al. **“Ambientes Virtuais Distribuídos e Compartilhados”**. In Tori, Romero; Kirner, Cláudio; Siscoutto, Robson. Fundamentos e Tecnologia de Realidade Virtual e Aumentada. Porto Alegre. Editora SBC, 2006.

Rogado, J (2002). **Aplicação de Realidade Virtual e Aumentada na Educação Química: O Ensino de Estrutura Atômica**. Disponível em: <http://www.abq.org.br/cbq/2008/trabalhos/13/13-487-4843.htm>. Julho de 2010

Rodello, I.; et.al. (2002). **Um Ambiente Virtual para Auxiliar o Ensino de Química em Escolas de Ensino Fundamental**. In: XXII Congresso da Sociedade Brasileira de Computação. Anais do VIII WIE. Florianópolis, SC.

Santin, R. **Sistema de Autoria em Ambiente Colaborativo com Realidade Aumentada**. 125 p. Dissertação (Mestrado em Ciência da Computação). Universidade Metodista de Piracicaba, Piracicaba. 2008.

Second Life (2009). **Site Second Life** <http://secondlife.com>. Acessado em Março de 2009.

Sementille, A. C. **A utilização da Arquitetura CORBA na Construção de Ambientes Virtuais Distribuídos**. 1999. 186f; Tese (Doutorado em Física Computacional) – Instituto de Física de São Carlos, USP. São Carlos, 1999.

Shulman, S. **Virtual reality goes to school**. Computer Graphics World, Nashua, v.22, n.3, p.38-44, Mar. 1999.

Silberschatz, A.; GALVIN, P.; GREG, G. **Sistemas Operacionais Conceitos e Aplicações**. Rio de Janeiro: Editor Campus, 2000.

Silberschatz, A.; G., Peter B.;G., Greg **Fundamentos de Sistemas Operacionais**. 6.ed. Rio de Janeiro:LTC,2004.

Silva, L. F. **Ambientes distribuídos em Realidade Virtual como suporte a Aprendizagem Cooperativa para a Resolução de Problemas**, Uberlândia, Faculdade de Engenharia Elétrica - UFU, 2008.

Silva, W. A. **Uma Arquitetura para Distribuição e Colaboração em Ambientes Virtuais de Realidade Aumentada**, Uberlândia, Faculdade de Engenharia Elétrica - UFU, 2008, 215p.

Stallings, W.. **Data and Computer Communications**. 5ª Edição.2007.

Tanembaum, A.S. **Distributed Operating Systems**. 1. Ed. New Jersey: Prentice-Hall. 2003. 614 pp. ISBN: 0-13-219908-4.

Tilley,T.; P., S.; E., P. **Evaluating Distributed Graphical User Interface Communication**. In Fifth Australian World Wide Web Conference (AusWeb99), 1999.

Tramontina G. B, A. W. **Sockets, RPC, RMI e CORBA uma comparação dessas tecnologias para comunicação cliente-servidor**. Outer, 2002.

UNESCO (2010). **Site UNESCO** http://www.unesco.org/pt/brasil/single-view/news/2010_education_for_all_global_monitoring_report_is_being_launched_19_january_in_new_york/back/9669/cHash/c79019d132/ - Acessado Março 2010.

Valente C.; Mattar J. **SECOND LIFE e WEB 2.0 na EDUCAÇÃO**. Novatec Editora. ISBN 978-85-7522-147-1.2007

Yang,Z.; D., K. **CORBA: A Platform for Distributed Object Computing (A State-of-the-Art Report on OMG/CORBA)**. ACM Operating Systems Review, Volume 30, Number 2, Abril,1996, pp 4-31

WANG, J.; K., S. **Efficient and accurate Ethernet simulation**.Technical Report TR99-1749. Department of Computer Science, Cornell University, May 20, 1999.

Weidlich, D., Scherer, S. ,W, M.; **“Analyses Using VR/AR Visualization”**, IEEE Computer Graphics and Visualization, pp.84-86, September/October 2008.

Wloka, M. M. **Lag in Multiprocessor VR**. *Presence*, v.4, n.1, Spring 1995.

World of Warcraft (2010). **Site World of Warcraft** [http:// worldofwarcraftbrasil.com](http://worldofwarcraftbrasil.com)
Acessado em Janeiro 2010.

Anexo I

Avaliação do Sistema de Distribuição de RV e RA		
Avaliador:		Data Avaliação:
() professor () aluno	Escolaridade: () Ensino Médio () Ensino Superior	

Assinale, por favor, a opção que melhor traduz a sua opinião.

1. FUNCIONALIDADE- Evidência que o conjunto de funções atende às necessidades explícitas e implícitas para a finalidade que se destina o produto	Muito Satisfeito	Satisfeito	Insatisfeito
1.1. O protótipo auxilia no processo de aprendizagem na finalidade proposta			
1.2. Como avalia globalmente a funcionalidade do protótipo.			

2. USABILIDADE- Evidência a facilidade de utilização de protótipo	Muito Satisfeito	Satisfeito	Insatisfeito
2.1. Em relação à facilidade de entender como funciona o protótipo			
2.2. Quanto à manipulação dos objetos da biblioteca compartilhada (facilidade de uso)			
2.3. Quanto à distribuição dos novos objetos (facilidade de uso).			
2.4. Quanto à comunicação via chat.			
2.5. Como avalia globalmente a usabilidade do protótipo.			

3. CONFIABILIDADE- Evidência que o desempenho se mantém ao longo do tempo em condições estabelecidas.	Muito Satisfeito	Satisfeito	Insatisfeito
3.1. Capacidade de continuar a funcionar corretamente, após erros do próprio protótipo ou erros de manipulação de dados.			
3.2. Capacidade de o protótipo enviar mensagens de erro caso caia a conexão.			

3.3. Como avalia globalmente a confiabilidade do protótipo.			
---	--	--	--

4. EFICIÊNCIA - Evidência que os recursos e os tempos envolvidos são compatíveis com o tempo de desempenho requerido para o produto.	Muito Satisfeito	Satisfeito	Insatisfeito
4.1. Tempo de distribuição dos objetos da biblioteca compartilhada.			
4.2. Tempo de distribuição envio de novos objetos.			
4.3. Como avalia em termos globais a Eficiência do protótipo.			

5. Portabilidade - A capacidade do produto de software para ser transferido de um ambiente para o outro	Muito Satisfeito	Satisfeito	Insatisfeito
5.1. Facilidade de instalação em diferentes ambientes (Windows XP/ Windows Vista/ Windows 7).			
5.2. Como avalia em termos globais a Portabilidade do protótipo.			

6. Sobre o protótipo RVAnet	Muito Satisfeito	Satisfeito	Insatisfeito
6.1. Importância que atribui o protótipo para o aprendizado em geral.			
6.2. A importância de distribuição de modelos em Realidade Virtual.			
6.3. A importância de distribuição de modelos em Realidade Aumentada.			
6.4. Quando se utiliza RV e RA, a visualização e o entendimento do modelo virtual é melhor			
Comentários/ Observações:			

Este questionário foi elaborado com base na ISONORM 9126-1 e também considerou alguns aspectos de software educacional descritos por Campo e Campos (2001).