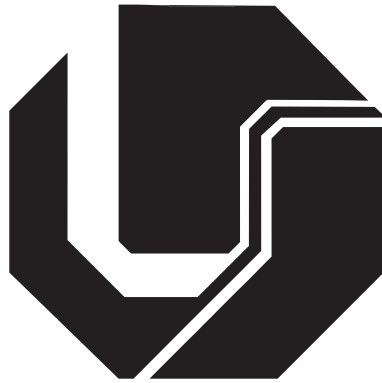


**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**Realidade Aumentada Distribuída na Web com
Independência de Servidor para Manipulação de
Objetos Virtuais na Camada Cliente**

Marlus Dias Silva

Abril

2011

Realidade Aumentada Distribuída na Web com Independência de Servidor para Manipulação de Objetos Virtuais na Camada Cliente

Marlus Dias Silva

Texto apresentado à Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Mestre em Ciências.

Prof. Alexandre Cardoso, Dr.

Orientador

Prof. Edgard A.Lamounier Jr, PhD.

Co-orientador

Prof. Alexandre Cardoso, Dr.

Coordenador do curso de Pós-Graduação

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

**Realidade Aumentada Distribuída na Web com
Independência de Servidor para Manipulação de
Objetos Virtuais na Camada Cliente**

Marlus Dias Silva

Texto apresentado à Universidade Federal de Uberlândia, perante a banca de examinadores abaixo, como parte dos requisitos necessários para a obtenção do título de Mestre em Ciências.

Banca Examinadora:

Prof. Alexandre Cardoso, Dr - Orientador (UFU)

Prof. Edgard A.Lamounier Jr, PhD - Co-orientador (UFU)

Prof. Keiji Yamanaka, PhD (UFU)

Prof. Claudio Kirner, PhD (UNIFEI)

Agradecimentos

Agradeço a minha família, meu pai Martinho Romualdo da Silva, minha mãe Marli Hozoria Dias Silva, minha irmã Marliane Dias Silva. Todos foram muito importante em todos os momentos da minha vida. E não seria diferente para a elaboração deste trabalho.

Agradeço a minha noiva Alina Tais Dário, pelo carinho e afeto durante toda essa luta.

Agradeço a meu amigos e colegas de Laboratório Keynes, Eduardo, Fernando, Fábio, Webert, Mônica, Ederson, Saulo, Gerson, pelos incentivos e companheirismo.

Agradeço aos meus colegas de trabalho, Aladir, Danilo, Sérgio, Renato, Leizer, Eliane, Heverton, pelo apoio prestado nessa luta.

Agradeço ao Senhor Igor vulgo (BOQUETA), pelas piadas e cafezinhos tomados durante todo o mestrado.

Aos meus orientadores Alexandre Cardoso e Edgard A.Lamounier Jr, pelos ensinamentos.

A todas as pessoas que participaram direto e indiretamente do trabalho.

Abstract

This dissertation presents an architecture to the Argumented Reality environment distribution , using the Web as a mechanism of distribution and execution. Considering the works related with the referred topic, realizing the works use the web as a mechanism of distribution, and the environment is installed and configured in its works station. Facing this, there is the aim of a development in the distribution of the Argumented Reality environment which executes directly on the web, throughout the browser. To prove the architecture, it was created a software prototype. The Argumented Reality interface from this software is based on a case study which represents a specific area of knowledge, in this case, a continuous current motor. This way the Argumented Reality interface makes the interaction by the menu and marker. The architecture uses the model client/server , and it was divided in three layers in the server and four layers on the customers. The prototype was submitted by shocked tests and scalability to prove his performance with multiple users. The tests were conducted by 35 students from the high school (technological) and teachers from the information technology area. All of them used the system and answered a questionnaire based on the ISONORM. The results obtained permits to conclude the software is effective and applied.

Key-Words: Argumented Reality, Distributed System, Middleware, Continuous Current.

Resumo

Esta dissertação apresenta uma arquitetura para a distribuição de ambiente de Realidade Aumentada, utilizando como mecanismo de distribuição e execução a Web. Considerando os trabalhos relacionados com o referido tópico, percebe-se que os trabalhos utilizam a web como mecanismo de distribuição, e o ambiente é instalado e configurado em suas estações de trabalho. Diante disto, tem-se como objetivo o desenvolvimento de um ambiente de Realidade Aumentada distribuído que execute diretamente na web, através de browser. Para validar a arquitetura, foi desenvolvido um protótipo de software. A interface de Realidade Aumentada deste software está baseada em um estudo de caso que representa uma área específica do conhecimento, neste caso, motores de corrente contínua. Desta forma a interface de Realidade Aumentada possibilita a interação por meio de menu e marcador. A arquitetura utiliza o modelo cliente/servidor, e foi dividida em três camadas no servidor e quatro camadas nos clientes. O protótipo foi submetido a testes de latência e escalabilidade para validar seu desempenho com múltiplos usuários. Os testes foram conduzidos por 35 alunos do ensino médio (tecnológico) e 5 professores da área de informática. Todos utilizaram o sistema e responderam um questionário baseado na ISONORM. Os resultados obtidos permitem concluir que o software é eficaz e aplicável.

Palavras-Chaves Realidade Aumentada, Sistemas Distribuídos, *Middleware*, Corrente Contínua.

PUBLICAÇÕES

A seguir são apresentadas as publicações resultantes desse trabalho:

SILVA, M. D. ; SANTOS, E. S. ; CARDOSO, A. ; LAMOUNIER JUNIOR, E. A. . DESENVOLVIMENTO DE UMA ARQUITETURA PARA DISTRIBUIÇÃO DE REALIDADE AUMENTADA NA WEB APLICADA AO ENSINO DE MOTORES DE CORRENTE CONTÍNUA. In: 40th IGIP International Symposium on Engineering Education, Santos, 2011.

SILVA, M. D. ; SANTOS, E. S. ; ABREU, R. O. ; LAMOUNIER JUNIOR, E. ; CARDOSO, A. Desenvolvimento de uma Arquitetura para Distribuição de Realidade Aumentada na Web Aplicada ao Ensino de Motores de Corrente Contínua. In: 7º Workshop de Realidade Virtual e Aumentada, São Paulo, 2010.

SILVA, M. D. ; SANTOS, E. S. ; ARANTES, W. V. ; LIMA, L. ; CARDOSO, A. ; LAMOUNIER JUNIOR, E. A. Uma Proposta de Middleware para distribuição de Realidade Aumentada pela Web como Ferramenta de Apoio para Educação a Distância. In: 6º. Workshop de Realidade Virtual e Aumentada, Santos, 2009.

Conteúdo

1	Introdução	1
1.1	Introdução	1
1.2	Objetivo	2
1.3	Organização da Dissertação	3
2	Fundamentos Tecnológicos	5
2.1	Introdução	5
2.2	Realidade Aumentada	5
2.2.1	Tipos de sistemas de Realidade Aumentada	7
2.2.2	Realidade Aumentada e Marcadores	10
2.2.3	Realidade Aumentada sem Marcadores	11
2.3	Ferramentas de Realidade Aumentada	12
2.3.1	ARToolKit	12
2.3.2	ArtoolkitPlus	13
2.3.3	ARTag	14
2.3.4	ARStudio	15
2.3.5	JARToolkit	16
2.3.6	DART	17
2.3.7	FLARToolkit	18

2.4	Sistemas Distribuídos	19
2.4.1	Características de um Sistema Distribuído	20
2.4.1.1	Tolerância a Falhas	22
2.4.2	Sistemas Distribuídos de Realidade Virtual e Aumentada	22
2.4.3	Middleware	23
2.4.4	Chamada de Procedimento Remoto	24
2.5	Modelo Cliente/Servidor	25
2.5.1	Modelo Cliente/Servidor de 2 Camadas	26
2.5.2	Modelo Cliente/Servidor de 3 Camadas	27
2.5.3	Modelo Cliente/Servidor de 4 Camadas	28
2.6	Conclusão	29
3	Trabalhos Relacionados	30
3.1	Introdução	30
3.1.1	Metodologia Para a Análise	30
3.2	Ambientes Colaborativos com Realidade Aumentada	31
3.3	Aplicações Educacionais em Ambientes Colaborativos com Realidade Aumentada	32
3.4	Uma Arquitetura para Distribuição de Ambientes Virtuais de Realidade Aumentada	34
3.5	Uma Arquitetura para a Distribuição de Realidade Virtual e Aumentada	35
3.6	Sistema de Autoria em Ambiente Colaborativo com Realidade Aumentada	37
3.7	Virtual Reality Tools for Internet-Based Robotic Teleoperation	38
3.8	Sumário e Conclusões	40
4	Arquitetura do Sistema Proposto	42

4.1	Introdução	42
4.2	Engenharia de Requisitos	42
4.3	Diagrama de Caso de Uso	43
4.4	Detalhes da Arquitetura	47
4.4.1	Módulo Servidor	49
4.4.2	Módulo Cliente	50
4.5	Conclusão	52
5	Detalhes de Implementação e Estudo de Caso	53
5.1	Introdução	53
5.2	Tecnologias Correlatas	53
5.2.1	Linguagem de Programação JAVA	54
5.2.2	Action Script 3	56
5.2.3	Glassfish	57
5.2.4	BlazeDS™	57
5.2.4.1	BlazeDS™ Arquitetura no Cliente	58
5.2.4.2	BlazeDS™ Arquitetura no Servidor	59
5.3	Detalhes de Implementação	60
5.3.1	Definição das Regras para Animação dos Modelos	61
5.3.2	Protocolo de Comunicação	61
5.3.3	Biblioteca de Objetos Virtuais	63
5.4	Estudo de Caso	63
5.4.1	Educação	64
5.4.2	Motor de Corrente Contínua	65
5.4.3	Modelagem do Motor	66

5.5	Funcionamento do Sistema	67
5.5.1	Funcionamento da Camada de Comunicação	67
5.5.2	A GUI	69
5.6	Considerações Finais	76
6	Avaliação e Resultados	79
6.1	Introdução	79
6.2	Avaliação do Sistema RANET	79
6.3	Latência	80
6.3.1	Escalabilidade	81
6.4	Testes de Usabilidade	82
6.4.1	Quanto à Finalidade do Uso da Ferramenta	82
6.4.2	Quanto à Interface com o Usuário	83
6.4.3	Quanto à Colaboração entre os Usuários é Perceptível	83
6.4.4	Quanto à Aprendizagem (Houve Aquisição de Conhecimento?)	84
6.4.5	Quanto ao Número de Comandos Existentes no Sistema ?	85
6.4.6	Quanto a Confiança Que o Sistema Executou o Que Foi Requisitado	85
6.4.7	Recursos não Implementados	86
6.5	Considerações Finais	86
7	Conclusões e Trabalhos Futuros	87
7.1	Introdução	87
7.2	Conclusões	87
7.3	Trabalhos Futuros	88

<i>CONTEÚDO</i>	vi
Referências bibliográficas	89
A Avaliação do Sistema - RANET	97
B Manual de Utilização do software - <i>RANET</i>	98
B.1 Menus Interativos	99
B.2 Biblioteca de Objetos Virtuais	100

Lista de Figuras

2.1	Diagrama de Realidade/Virtualidade Contínua (MILGRAM et al., 1994).	6
2.2	Diagrama adaptado do sistema de visão direta por vídeo (SUTHAU et al., 2002).	8
2.3	Diagrama adaptado do sistema de visão direta por vídeo e um modelo de dispositivo (ZORZAL, 2009).	8
2.4	Diagrama adaptado e dispositivo do sistema de visão por vídeo baseado em monitor (ZORZAL, 2009).	9
2.5	Ambiente de projeção (ZORZAL, 2009).	9
2.6	Marcadores de Realidade Aumentada.	10
2.7	Realidade Aumentada sem Marcador (FRAHM et al., 2005).	11
2.8	ARToolKit Aplicado em Jogos Educacionais (ZORZAL; KIRNER, 2005).	13
2.9	Funcionamento do ARToolKit (BASTOS, 2005).	14
2.10	ArtoolkitPlus Executando em um Dispositivo Móvel (WAGNER; SCHMALSTIEG, 2007).	15
2.11	Objeto 3D Carregados no Marcador ARTag (BASTOS, 2005).	16
2.12	Marcadores ARTag (BASTOS, 2005).	17
2.13	Marcadores ARStudio (MALIK; ROTH; MCDONALD, 2002).	17
2.14	<i>FLARToolkit</i> Exemplo de Aplicação.	19
2.15	Camadas de Software.	23
2.16	Modelo Cliente Servidor Tradicional (BATISTI, 2010).	26

2.17	Modelo Cliente Servidor de Duas Camadas (BATISTI, 2010).	26
2.18	Modelo Cliente Servidor de Três Camadas (BATISTI, 2010).	27
2.19	Modelo Cliente Servidor de Quatro Camadas (BATISTI, 2010).	28
3.1	Visualização dos Objetos Virtuais em Realidade Aumentada em um Ambiente Colaborativo (GERALDI; KIRNER, 2005).	32
3.2	Ambiente Colaborativo com Realidade Aumentada em Rede (KIRNER; ZORZAL, 2005).	33
3.3	Arquitetura Proposta para a Distribuição do Sistema (SILVA, 2008).	35
3.4	Arquitetura do Sistema (NOGUEIRA et al., 2009).	36
3.5	ESTRUTURA DO SACRA (SANTIN, 2008).	38
3.6	Arquitetura Desenvolvida (YANG; CHEN, 2004).	39
3.7	Ambiente Virtual em Execução (YANG; CHEN, 2004).	40
4.1	Diagrama de Caso de Uso do Sistema de Realidade Aumentada para Web.	43
4.2	Arquitetura do Sistema Proposto.	48
4.3	Arquitetura do Módulo Servidor.	49
4.4	Arquitetura do Módulo Cliente.	51
5.1	Representação do Sistema Proposto.	54
5.2	Representação da Java Virtual Machine.	55
5.3	Flash Player, Quantidade de Computadores (ADOBE, 2010).	57
5.4	Arquitetura do Cliente BlazeDS™ (BLAZEDS, 2010).	58
5.5	Arquitetura do Servidor BlazeDS (BLAZEDS, 2010).	59
5.6	Estrutura Lógica das Tecnologias Utilizadas.	60
5.7	Esboço do Motor Corrente Contínua (TÉCNICA, 2006).	65
5.8	Sistema de Comutação (TÉCNICA, 2006).	66

5.9	Motor de Corrente Contínua Modelado no Blender.	67
5.10	Interface de Distribuição do Software.	69
5.11	<i>Pop-up</i> , Lista dos Usuários Conectados no Sistema.	70
5.12	Marcador da Aplicação.	71
5.13	Interface do Sistema de Realidade Aumentada.	72
5.14	Animações Cadastradas no Sistema.	73
5.15	Demais Animações Cadastradas.	74
5.16	Motor Posicionado Sobre Marcador.	75
5.17	Motor Virtual Projetado Sobre Motor Real.	76
5.18	Usuários Utilizando o Sistema.	77
6.1	Latência de Comunicação.	81
6.2	Escalabilidade.	82
6.3	Gráfico: Finalidade da Ferramenta.	82
6.4	Gráfico: Interface com o Usuário.	83
6.5	Gráfico: Colaboração entre Usuários.	84
6.6	Gráfico: Aquisição de Conhecimento.	84
6.7	Gráfico: Comandos Existentes no Sistema.	85
6.8	Gráfico: Confiança no Sistema.	85
B.1	Interface gráfica do software RANET.	98
B.2	Interface de Distribuição.	99
B.3	<i>Pop-up</i> , Lista dos Usuários Conectados no Sistema.	100
B.4	Menus Interativos.	100

Lista de Tabelas

2.1	Camadas de Software do JARToolkit.	18
2.2	Comparação Entre as Tecnologias de Realidade Aumentada	20
3.1	Características Analisadas nos Trabalhos Relacionados.	41
4.1	Descrição do Caso de Uso Biblioteca Virtual.	44
4.2	Descrição do Caso de Uso Interface de Distribuição.	45
4.3	Descrição do Caso de Uso Executar Animações.	46
4.4	Descrição do Caso de Uso Realizar Transformadas Geométricas.	46
4.5	Descrição do Caso de Uso Imprimir Marcador.	47
5.1	Regras Definidas no Time Line de Animação.	61
5.2	Mensagens Trafegadas pela Rede para Animações	62
5.3	Comparações e Contribuições desta Dissertação.	78

Lista de Códigos

5.1 Método para Manipular Animação	62
----------------------------------------------	----

Capítulo 1

Introdução

1.1 Introdução

A Realidade Virtual é, antes de tudo, uma “interface avançada do usuário” para acessar aplicações executadas no computador, tendo como características, a visualização de, e movimentação em, ambientes tridimensionais em tempo real e a interação com elementos do mesmo (KIRNER; TORI, 2004). Nesse aspecto, pode-se considerar que a Realidade Virtual é considerada uma tecnologia revolucionária, pois possibilita a simulação de mundos reais e imaginários na tela do computador ou em outros dispositivos, criando no usuário a sensação de presença em um “mundo” virtual (SILVA, 2008).

A Realidade Aumentada é a inserção de objetos virtuais no mundo real por meio de um dispositivo computacional. Desta forma, a interface do usuário é aquela, que ele usa no ambiente real, adaptada para visualizar e manipular os objetos virtuais colocados no seu espaço (SILVA, 2008).

A Realidade Aumentada também pode ser definida como a inserção de objetos virtuais no ambiente real, 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; SISCOOTTO, 2007). A Realidade Aumentada aplica-se em todos os sentidos humanos (AZUMA et al., 2001) e proporciona ao usuário uma interação segura, uma vez que ele pode trazer para o seu

ambiente real objetos virtuais, incrementando e aumentando a visão que ele tem do mundo real (KIRNER; ZORZAL, 2005). Isto é obtido por meio de técnicas de Visão Computacional e de Computação Gráfica/Realidade Virtual, o que resulta na sobreposição de objetos virtuais com o ambiente real (BILLINGHURST; KATO; POUPYREV, 2005). Considerando o sentido da visão, além de permitir que objetos virtuais possam ser introduzidos em ambientes reais, a Realidade Aumentada também proporciona ao usuário o manuseio desses objetos com as próprias mãos, possibilitando uma interação natural e atrativa com o ambiente (ZHOU et al., 2004; BILLINGHURST; KATO; POUPYREV, 2005).

A Realidade Aumentada Distribuída, permite que uma comunidade de usuários compartilhe e interaja com um conjunto de informações associadas a objetos virtuais, em tempo real (KIRNER; TORI, 2004), permitindo a troca de experiências entre os participantes. Tradicionalmente, a Realidade Aumentada Distribuída tem sido concebida para arquiteturas desktop, possuindo limitações relacionadas com a dependência de sistema operacional, dificuldade para instalar/configurar novas versões na máquina do cliente etc. Recentemente, alguns trabalhos apresentam suporte à distribuição de Realidade Aumentada por meio da Web (GERALDI; KIRNER, 2005; KIRNER; ZORZAL, 2005; SILVA, 2008; NOGUEIRA et al., 2009; SANTIN, 2008; YANG; CHEN, 2004).

Entretanto, exigem do usuário o domínio de conhecimento de técnicas de programação e adequação de interface para a devida utilização, demandando equivalente tempo para configuração do ambiente. Com isto, tais sistemas não podem ser avaliados como sistemas transparentes ao usuário (cliente). Acresce-se a isto que os usuários não têm a liberdade de alterar as propriedades dos objetos virtuais na camada cliente e a não podem inserir novos objetos virtuais durante a experimentação. Tal motivo impede, ainda, a equivalente manipulação (síncrona) dos objetos virtuais distribuídos em ambientes aumentados por diferentes clientes.

Portanto, seria interessante que essas funcionalidades fossem providas porque os usuários podem ganhar facilidade de executar o Ambiente Aumentado Distribuído, e a possibilidade de adicionar novos objetos virtuais, e manipula-los.

1.2 Objetivo

Esta dissertação propõe uma arquitetura capaz de suportar ambientes de Realidade Aumentada Distribuída, com suporte à interação de diferentes usuários com objetos virtuais que compõem tais ambientes. Tal interação será suportada de forma assíncrona, com resultados imediatos sobre os objetos virtuais distribuídos nos ambientes aumentados, propiciando a cada usuário a visualização e a verificação das ações próprias e coletivas. Porém, não é objetivo neste trabalho propor uma estratégia de interação simultânea e síncrona com os objetos virtuais em ambientes aumentados. De forma a verificar a estratégia desenvolvida, elaborou-se estudo de caso, relacionado com uma ferramenta de ensino. Para alcançar esses objetivos, foram definidas as seguintes metas:

1. Escolher uma ferramenta para a programação de Realidade Aumentada na Web.
2. Escolher um *middleware* para a distribuição dos objetos virtuais.
3. Projetar uma arquitetura em camadas, com o intuito de facilitar a sua extensão, e possibilitar sua execução em diversos sistemas operacionais tais como Windows, Linux e Mac.
4. Implementar um protótipo que possibilite a distribuição e interação entre os usuários conectados por meio da internet.
5. Escolher um estudo de caso (*motores de corrente contínua*).
6. Realizar testes de escalabilidade e latência, e avaliar o *software* junto a usuários.

As contribuições deste trabalho encontram-se mediante a interseção de duas áreas de pesquisa e conhecimento: Realidade Aumentada e Sistema Distribuídos.

1.3 Organização da Dissertação

A presente dissertação consiste de 7 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 apresenta uma visão de Realidade Virtual e Aumentada, e sistemas distribuídos. O Capítulo 3 aborda o estado da arte dos sistemas distribuídos de Realidade Aumentada, suas abrangências, suas estruturas, potencialidades e limitações.

O capítulo 4 aborda a arquitetura desenvolvida. O capítulo 5 aborda os detalhes de implementação e o estudo de caso proposto (motor de corrente contínua). Capítulo 6 discorre sobre os resultados obtidos. O Capítulo 7 aborda sobre as conclusões e trabalhos futuros.

Capítulo 2

Fundamentos Tecnológicos

2.1 Introdução

Neste capítulo serão apresentados os conceitos de Realidade Aumentada, suas características e definições. Também serão abordados sistemas distribuídos e distribuição de ambientes de Realidade Aumentada, juntamente com as técnicas de distribuição utilizadas neste trabalho.

2.2 Realidade Aumentada

A Realidade Aumentada é a sobreposição de objetos virtuais gerados por computador num ambiente real, utilizando para isso dispositivos tecnológicos (MILGRAM et al., 1994) e viabilizando uma interface mais natural com dados e imagens geradas por computador (CARDOSO; KIRNER; LAMOUNIER, 2007). A conceituação de Realidade Aumentada fica clara com sua inserção em um contexto mais amplo: o da Realidade Misturada.

De acordo com (KIRNER; TORI, 2004) “Realidade Misturada, mistura o real com o virtual, abrange duas possibilidades: a Realidade Aumentada, cujo interface predominante é o mundo real, e a Virtualidade Aumentada, cujo interface predominante é o mundo virtual”. Desta forma a Realidade Aumentada é a particularização da Realidade

dade Misturada.

A Figura 2.1 apresenta o diagrama adaptado de realidade/virtualidade contínua, onde são mostradas as possibilidades gradativas de sobreposição do ambiente real com o virtual e vice-versa.

Observando a escala da Figura 2.1 nota-se que a Realidade Aumentada é uma particularização da Realidade Misturada, definida como a fusão entre o mundo real (Realidade) e objetos virtuais. Quando os objetos virtuais são trazidos para o mundo real (Realidade), tem-se a Realidade Aumentada e quando objetos reais são colocados no mundo virtual, tem-se a Virtualidade Aumentada (AZUMA et al., 2001).



Figura 2.1: Diagrama de Realidade/Virtualidade Contínua (MILGRAM et al., 1994).

A Realidade Aumentada é utilizada em várias áreas do conhecimento, como na visualização médica, marketing, simulação, games, engenharia, educação e é um forte campo de pesquisa nos laboratórios de computação gráfica.

Neste contexto, entende-se que as soluções de Realidade Aumentada envolvem a geração de elementos tridimensionais gerados por computador, que são inseridos no ambiente real de tal forma que o usuário crê que os mesmos são parte do meio no qual está inserido (SILVA, 2008).

Segundo (CARDOSO et al., 2006), as características básicas de sistemas de Realidade Misturada são:

- Processamento em tempo real.
- Combinação de elementos virtuais com o ambiente real.
- Uso de elementos virtuais concebidos em 3D.

A concepção de soluções de Realidade Aumentada necessita de componentes que permitam avaliar a posição do usuário, o ponto de vista, e gerar os elementos virtuais para finalmente combiná-los com o mundo real por meio de sistemas de projeção, sendo eles óculos especiais ou monitores. Para tanto, os elementos reais e virtuais necessitam ser alinhados corretamente, um em relação ao outro, o que pode demandar em alguns casos, um grande poder de processamento (CARDOSO et al., 2006; AZUMA et al., 2001).

2.2.1 Tipos de sistemas de Realidade Aumentada

Os sistemas de Realidade Aumentada podem ser classificados conforme o tipo de display utilizado (AZUMA et al., 2001), envolvendo visão ótica ou visão por vídeo, dando origem a quatro tipos de sistemas (KIRNER; ZORZAL, 2005).

- Sistema de visão ótica direta;
- Sistema de visão direta por vídeo;
- Sistema de visão por vídeo baseado em monitor;
- Sistema de visão ótica por projeção.

O sistema de visão ótica direta utiliza óculos ou capacetes com lentes que permitem o recebimento direto da imagem real, ao mesmo tempo em que possibilitam a projeção de imagens virtuais devidamente ajustadas com a cena real. Uma maneira comum de se conseguir essa característica é usar uma lente inclinada que permita a visão direta e que reflita a projeção de imagens geradas por computador diretamente

nos olhos do usuário (ZORZAL, 2009). A Figura 2.2 mostra o diagrama desse tipo de sistema.

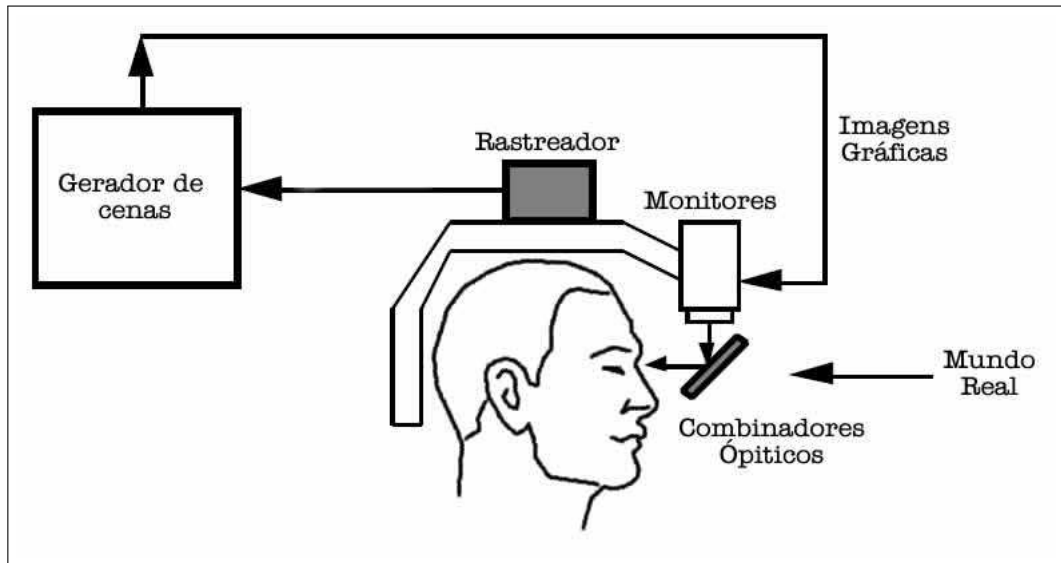


Figura 2.2: Diagrama adaptado do sistema de visão direta por vídeo (SUTHAU et al., 2002).

O sistema de visão direta, por vídeo, utiliza capacetes com microcâmeras de vídeo acopladas. A cena real, capturada pela microcâmera, é misturada com os elementos virtuais gerados por computador e apresentadas diretamente nos olhos do usuário, através de pequenos monitores montados no capacete (ZORZAL, 2009). A Figura 2.3 mostra o diagrama e apresenta um dispositivo de visão direta por vídeo.

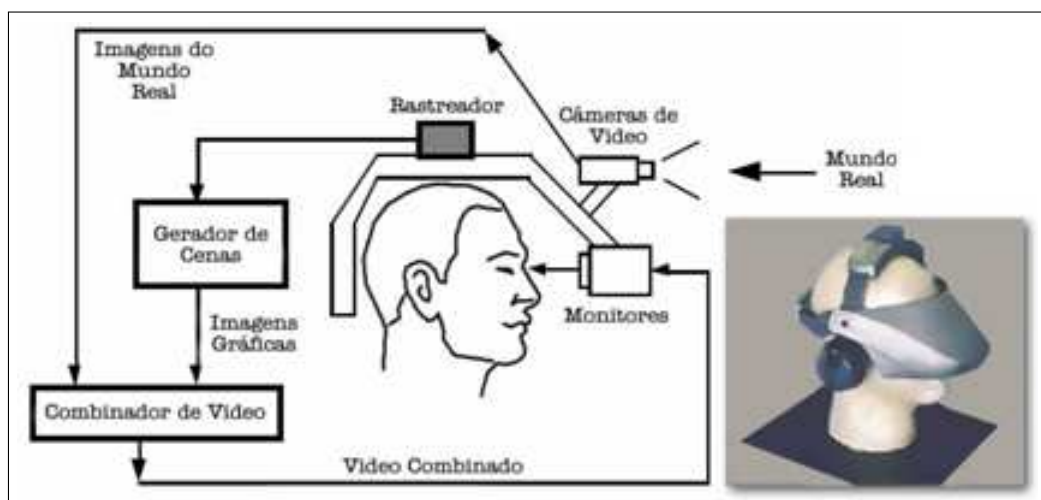


Figura 2.3: Diagrama adaptado do sistema de visão direta por vídeo e um modelo de dispositivo (ZORZAL, 2009).

O sistema de visão por vídeo, baseado em monitor, utiliza uma webcam para capturar a cena real. Depois de capturada, a cena real é misturada com os objetos virtuais gerados por computador e apresentada no monitor. O ponto de vista do usuário normalmente é fixo e depende do posicionamento da webcam. A Figura 2.4 mostra o diagrama e os equipamentos utilizados nesse caso.

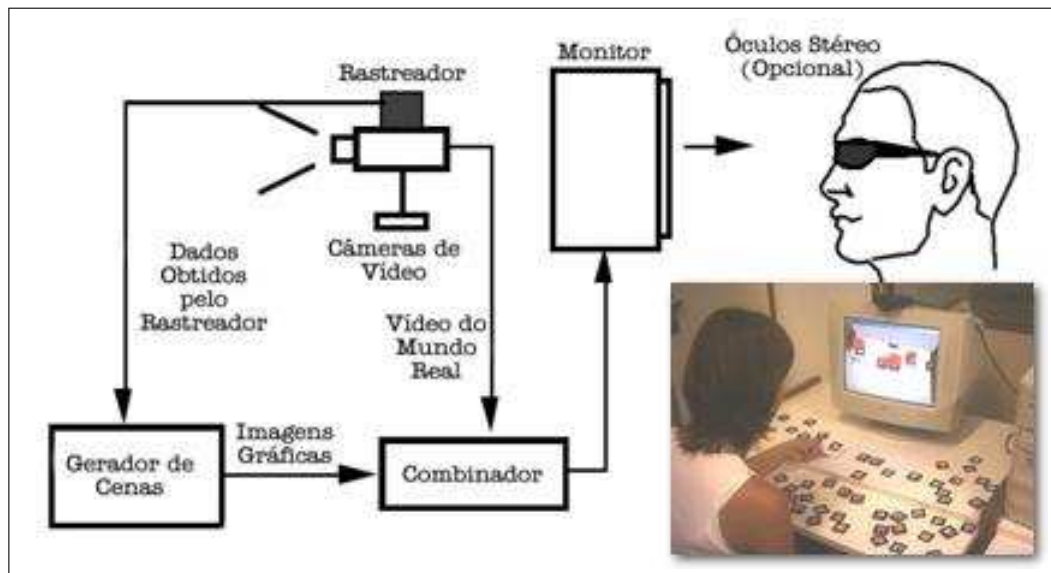


Figura 2.4: Diagrama adaptado e dispositivo do sistema de visão por vídeo baseado em monitor (ZORZAL, 2009).

O sistema de visão ótica, por projeção, utiliza superfícies do ambiente real, onde são projetadas imagens dos objetos virtuais, cujo conjunto é apresentado ao usuário que o visualiza sem a necessidade de nenhum equipamento auxiliar. Embora interessante, esse sistema é muito restrito às condições do espaço real, em função da necessidade de superfícies de projeção. A Figura 2.5 apresenta um ambiente que utiliza o sistema de visão ótica por projeção.

Os sistemas de visão direta são apropriados para situações, nas quais a perda da imagem pode ser perigosa, como é o caso de uma pessoa andando pela rua, dirigindo um carro ou pilotando um avião. Em locais fechados, nas quais o usuário tem controle da situação, o uso da visão por vídeo é adequado e não oferece perigo, pois em caso de perda da imagem pode-se retirar o capacete com segurança, se for o caso. O sistema com visão por vídeo é mais barato e mais fácil de ser ajustado (ZORZAL, 2009).



Figura 2.5: Ambiente de projeção (ZORZAL, 2009).

Este trabalho utilizará o sistema de visão por vídeo, baseado em monitor.

2.2.2 Realidade Aumentada e Marcadores

Os marcadores são placas de referência, onde software como *ARToolKit*, *FLARToolKit* e outros utilizam de visão computacional para calcular a posição destes marcadores em relação à uma câmera, possibilitando a inserção de objetos virtuais sobre os marcadores, o que justifica a expressão “Realidade Aumentada”. Esses marcadores são objetos físicos com bordas quadradas ou retangulares na cor preta. No centro, pode-se inserir um desenho preto sobre um fundo branco ou o contrário. A Figura 2.6 demonstra um marcador (BUCIOLI, 2008).

Pelo fato dos sistemas de Realidade Aumentada serem baseados em visão computacional, os objetos virtuais só podem ser exibidos quando as marcas estiverem sendo rastreadas. Se o usuário cobrir parte do marcador com as mãos, o objeto virtual desaparecerá. Outro problema é com a iluminação. Dependendo da iluminação, o software não consegue reconhecer o marcador e até mesmo a inclinação do mesmo pode tornar irreconhecível pelo programa (CONSULARO et al., 2004).

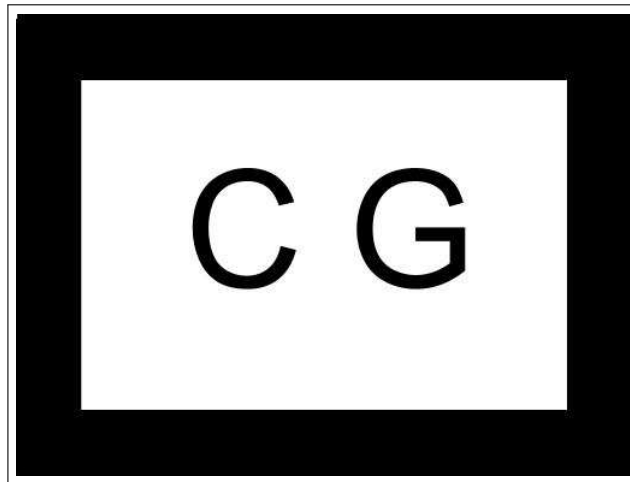


Figura 2.6: Marcadores de Realidade Aumentada.

2.2.3 Realidade Aumentada sem Marcadores

A Realidade Aumentada sem marcadores (*Markerless Augmented Reality*), executa o rastreamento (*tracking*) da câmera, e possibilita a inserção dos elementos virtuais na cena real, utilizando informações naturalmente presentes como arestas (MARCHAND et al., 2007), texturas (VACCHETTI; LEPETIT; FUA, 2004) ou a própria estrutura da cena (POLLEFEY, 1999) sem a inserção de marcadores no ambiente real.



Figura 2.7: Realidade Aumentada sem Marcador (FRAHM et al., 2005).

No trabalho em (FRAHM et al., 2005), como demonstra a Figura 2.7, um carro é

inserido na cena utilizando Realidade Aumentada sem marcadores em uma sala real, com o objetivo de desenvolver uma ferramenta para utilização em programas de TV.

2.3 Ferramentas de Realidade Aumentada

Diversas ferramentas possibilitam o desenvolvimento de Realidade Aumentada, em linguagens de programação distintas, como C/C++, Flash, JAVA e outras. A seguir, um resumo destas ferramentas é apresentado.

2.3.1 ARToolkit

O *ARToolkit* é uma biblioteca de programação multi-plataforma considerada um kit de ferramenta para Realidade Aumentada. É uma ferramenta de código livre para modificações e uso no desenvolvimento de aplicações não comerciais sob licença GLP (LICENSE, 2010).

Foi desenvolvido originalmente para servir de apoio na concepção de interfaces colaborativas (KATO; BILLINGHURST, 1999) por Hirokazu Kato, na Universidade de Osaka.

A biblioteca é implementada em linguagem C/C++, utiliza rastreamento óptico e implementa técnicas de visão computacional, para identificar e estimar em tempo real a posição e a orientação de um marcador em relação ao dispositivo de vídeo (SANTIN; KIRNER, 2007). Assim, o cálculo da correlação entre os dados estimados do marcador e sua imagem, possibilita posicionar objetos virtuais alinhados à imagem do marcador como demonstra a Figura 2.8 (KATO; BILLINGHURST, 1999) .

O funcionamento do *ARToolkit* consiste das seguintes etapas: a câmera é posicionada no *display* do usuário e captura as imagens do mundo real que são enviadas ao computador, onde o mesmo processará os *frames*, verificando se existe algum marcador. Se o marcador for encontrado, é efetuada uma série de cálculos matemáticos para definir a posição da câmera ao marcador visualizado. Uma vez que a posição da câmera é conhecida, um modelo computacional virtual será posicionado nessa posi-



Figura 2.8: ARToolKit Aplicado em Jogos Educacionais (ZORZAL; KIRNER, 2005).

ção, onde o resultado final é uma imagem virtual sobreposta em cima do marcador. Esses passos podem ser visualizados através da Figura 2.9.

2.3.2 ArtoolkitPlus

Uma biblioteca foi desenvolvida baseada no *Artoolkit*, que possibilita a implementação em dispositivos móveis. O *ArtoolkitPlus* foi desenvolvido na *Graz University Technology* dentro do projeto *Studierstube* (WAGNER; BARAKONYI, 2003), e é uma biblioteca *open source*. Seu código-fonte conta com otimizações, tais como, a utilização de matemática de ponto fixo com o intuito de gerar aplicações eficientes para dispositivos com o poder de processamento limitado. Os marcadores utilizados por essa biblioteca são semelhantes aos do *Artoolkit*, essa biblioteca trabalha com a técnica de limiar adaptativo, que consiste em perceber alterações de iluminação no ambiente, capturado pelas câmeras, de forma que a detecção dos marcadores não seja prejudicada, a Figura 2.10 demonstra uma aplicação desenvolvida utilizando o *ArtoolkitPlus*

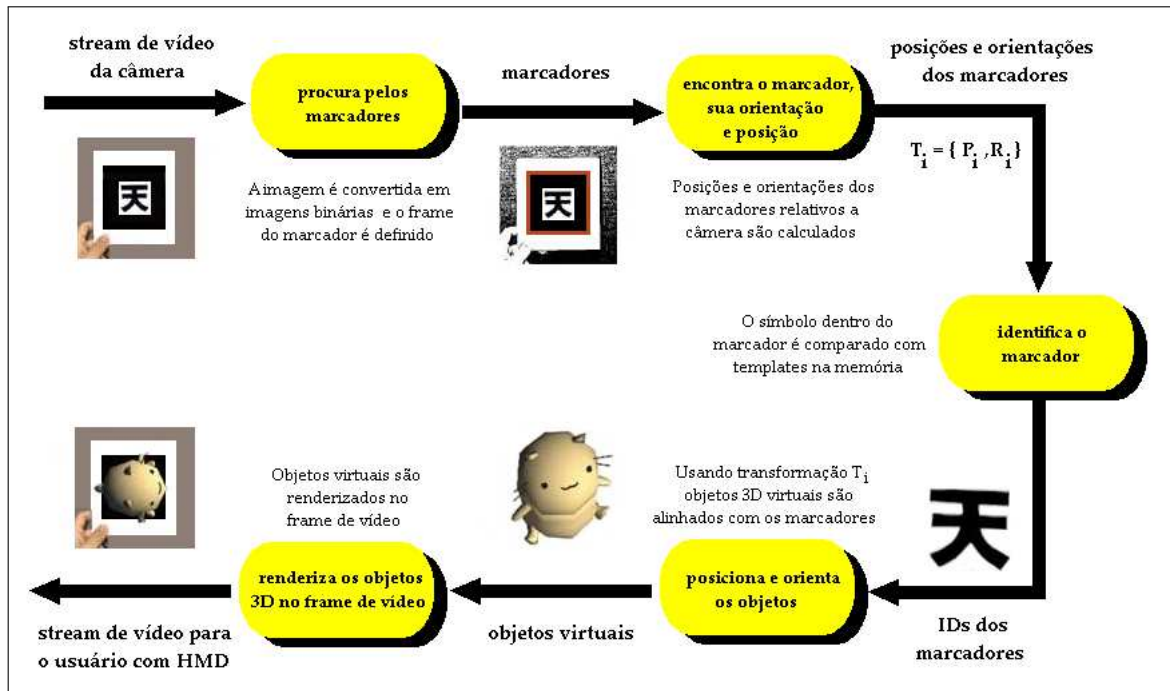


Figura 2.9: Funcionamento do ARToolKit (BASTOS, 2005).

(WAGNER; BARAKONYI, 2003; WAGNER; SCHMALSTIEG, 2003).

2.3.3 ARTag

O *ARTag* foi desenvolvido pelo *National Research Council of Canada* (FIALA, 2005), tendo como base o *ARToolkit*. O *ARTag* consiste em uma biblioteca de padrões, diferentemente do *ARToolkit* que compara imagens. Ele compara códigos digitais compostos de 0 e 1. O interior de seu marcador é preenchido com uma malha 6x6 de quadrados pretos e brancos que representam os códigos digitais como demonstra a Figura 2.11.

O *ARTag* foi desenvolvido com o propósito de solucionar alguns problemas existentes do *ARToolkit*. Entre esses problemas pode-se destacar três:

- *false positive* \mapsto quando a presença de um marcador é reportada, mas não existe nenhum no ambiente.
- *inter-marker confusion* \mapsto quando a presença de um marcador é reportada, mas com o ID errado.

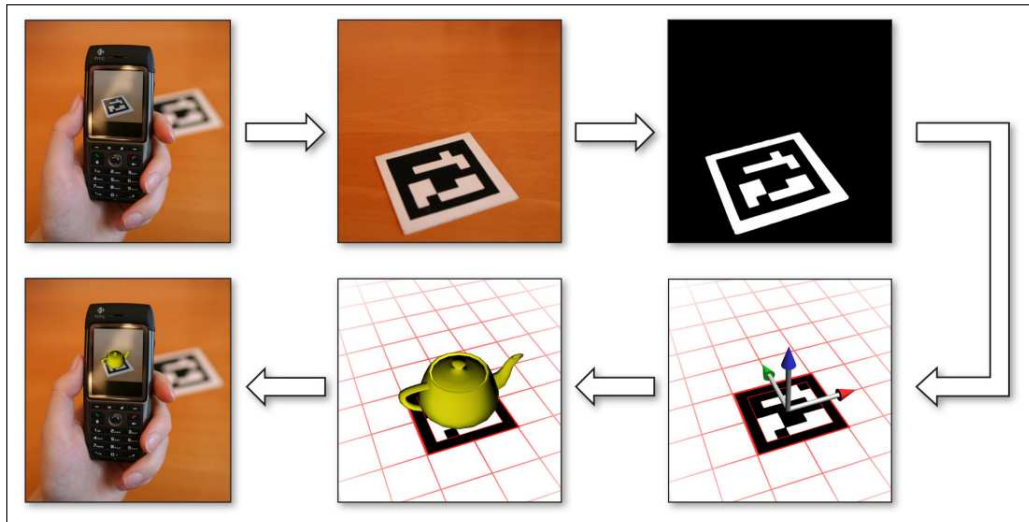


Figura 2.10: ArtoolkitPlus Executando em um Dispositivo Móvel (WAGNER; SCHMALSTIEG, 2007).

- *inter-marker confusion* \mapsto quando existe um marcador no ambiente, mas sua presença não é reportada.

O *ARTag* pode possuir uma vasta quantidade de marcadores, 2002 no total, sendo que 1001 possui o mesmo padrão do *Artoolkit*. Isto é, um quadrado de bordas pretas e interior branco, o restante têm um padrão inverso, um quadrado de bordas brancas e interior preto como demonstra a Figura 2.12 (FIALA, 2005).

O processo de reconhecimento do marcador do *ARTag* Segundo (BASTOS, 2005) é: “Primeiramente deve ser localizado o contorno da borda do quadrado. Após isso, a região interna que contém a malha é retirada e são determinados os códigos de 0’s e 1’s contidos. Todo o processamento subsequente de identificar e verificar o marcador são feitos digitalmente. Quatro sequências binárias de 36 bits são obtidas do *array* de códigos binários originários da malha. Uma para cada uma das quatro possíveis posições rotacionadas. Dentre essas, apenas uma será validada pelo processo de decodificação. A sequência binária codificada no marcador encapsula um ID de 10 bits usando métodos digitais. Os 26 bits extras proveem redundância para reduzir as chances de detecção e identificação falsas e também para prover unicamente uma das quatro possíveis rotações”.

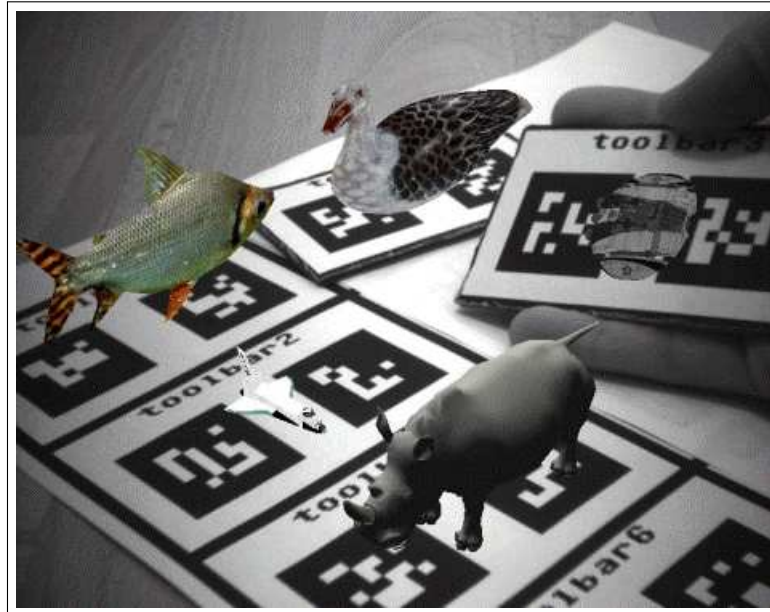


Figura 2.11: Objeto 3D Carregados no Marcador ARTag (BASTOS, 2005).

2.3.4 ARStudio

Outro conjunto de biblioteca desenvolvido pela *National Research Council of Canada*, foi o *ARStudio* que possui os mesmos princípios do *ARTag*. Os marcadores no *ARStudio* contém um quadrado com bordas pretas, e no seu interior se encontram retângulos brancos igualmente distribuídos (MALIK; ROTH; MCDONALD, 2002), como demonstra a Figura 2.13.

No reconhecimento do marcador, os quatro cantos de cada retângulo é armazenado em um arquivo de dados, que é associado a um padrão. Cada padrão gerado é unicamente identificável com respeito aos padrões já existentes (MALIK; ROTH; MCDONALD, 2002).

2.3.5 JARToolkit

O *JARToolkit* consiste de um conjunto de classes de código aberto desenvolvido em linguagem de programação JAVA, com todos os recursos existentes no *ARToolkit* e utilizados em conjunto com bibliotecas gráficas como OpenGL e DirectX, permitindo assim, o desenvolvimento de Realidade Aumentada (BREGA et al., 2008).

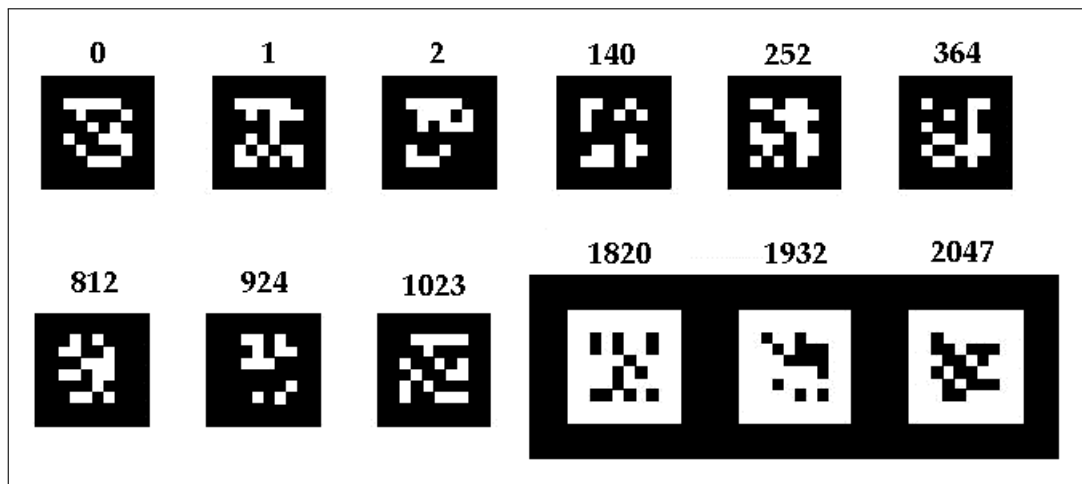


Figura 2.12: Marcadores ARTag (BASTOS, 2005).

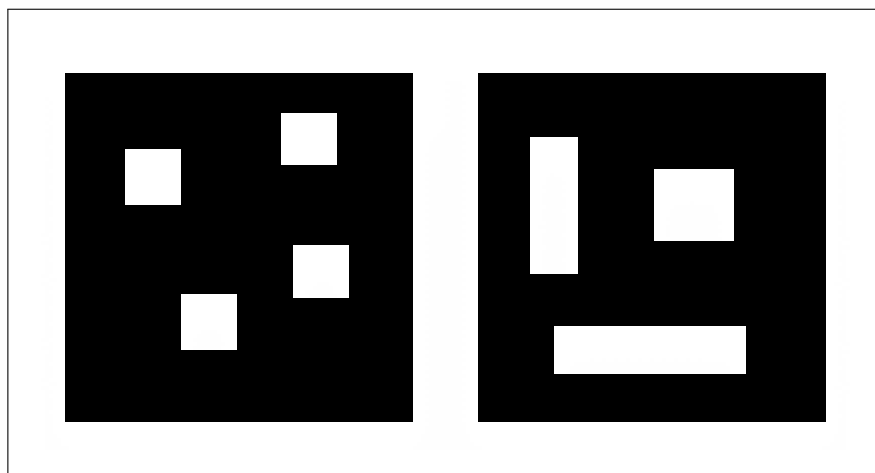


Figura 2.13: Marcadores ARStudio (MALIK; ROTH; MCDONALD, 2002).

Para tanto, é utilizado o JNI, uma característica do JAVA, que oferece a possibilidade de reaproveitamento de código de outras linguagens, dentro da linguagem de programação JAVA. A estrutura geral de classes do *JARToolkit* pode ser observada pela Tabela 2.1.

O uso *JARToolkit* possibilita a utilização em conjunto com outros pacotes do JAVA, como o JAVA3D¹ (SOWIZRAL; RUSHFORTH; SOWIZRAL, 1997).

¹API 3D para criação de mundos e objetos tridimensionais em JAVA.

Tabela 2.1: Camadas de Software do JARToolkit.

Classe do JARToolkit	
Classe	Descrição
JARtoolkit	Interface com o código nativo(ARToolkit)
ARPattern	Encapsula um padrão a ser reconhecido pelo ARToolkit
JARFrameGrabber	Captura o vídeo usando Direct X
JARPictureSource	Detecta diferentes entradas de vídeo
JARVideo	Pacote de suporte à captura de diferentes tipos de vídeo

2.3.6 DART

O GVV Center Georgia Institute of Technology desenvolve o *Designers Augmented Reality Toolkit* DART, que é um conjunto de ferramentas que provêem um desenvolvimento rápido em Realidade Aumentada (BASTOS, 2005).

O DART tem como objetivo permitir que *designers*, artistas e pesquisadores de outras áreas tenham experiências com aplicações em Realidade Aumentada. Por essa razão, o DART foi desenvolvido como uma coleção de extensões do ambiente de programação multimídia Macromedia Director, consistindo de *behaviors* (extensões do Director escritas na linguagem LINGO) e *Xtras* (*plugins* para o Director escritos na linguagem C++) (MACINTYRE et al., 2003).

Essas bibliotecas foram desenvolvidas para uma classe específica de experiências de Realidade Aumentada, onde uma mídia gerada por computador é diretamente integrada com a percepção dos participantes. Neste tipo de experiência, o usuário utiliza de óculos ou capacetes que permitem uma integração entre computação gráfica e som no mundo físico real. O DART é utilizado no suporte a reparos de equipamentos, treinamento baseado em situações de turismo (DOCUMENTATION, 2010).

2.3.7 FLARToolkit

O *FLARToolkit* é um conjunto de classes desenvolvidas em *Action Script 3*², que juntamente com o *Papervision3D*³ possibilita o desenvolvimento de aplicações de Re-

²Linguagem de programação.

³Conjunto de bibliotecas 3D para Action Script 3.

alidade Aumentada, como demonstra a Figura 2.14. Para executar aplicações utilizando o *FLARToolkit* o cliente deve instalar no navegador o *Flash Player versão 9* ou superior.

O *FLARToolkit* pode trabalhar paralelamente com as *engines 3D*, o que torna as aplicações desenvolvidas com ele mais robustas, o mesmo pode trabalhar com formatos de arquivos 3D exportados por modeladores como Blender e 3D Studio Max, nos formatos Collada (*COLLABorative Designer Activity*), MD2(*Quake 2*) e 3DS(*3D Studio*).

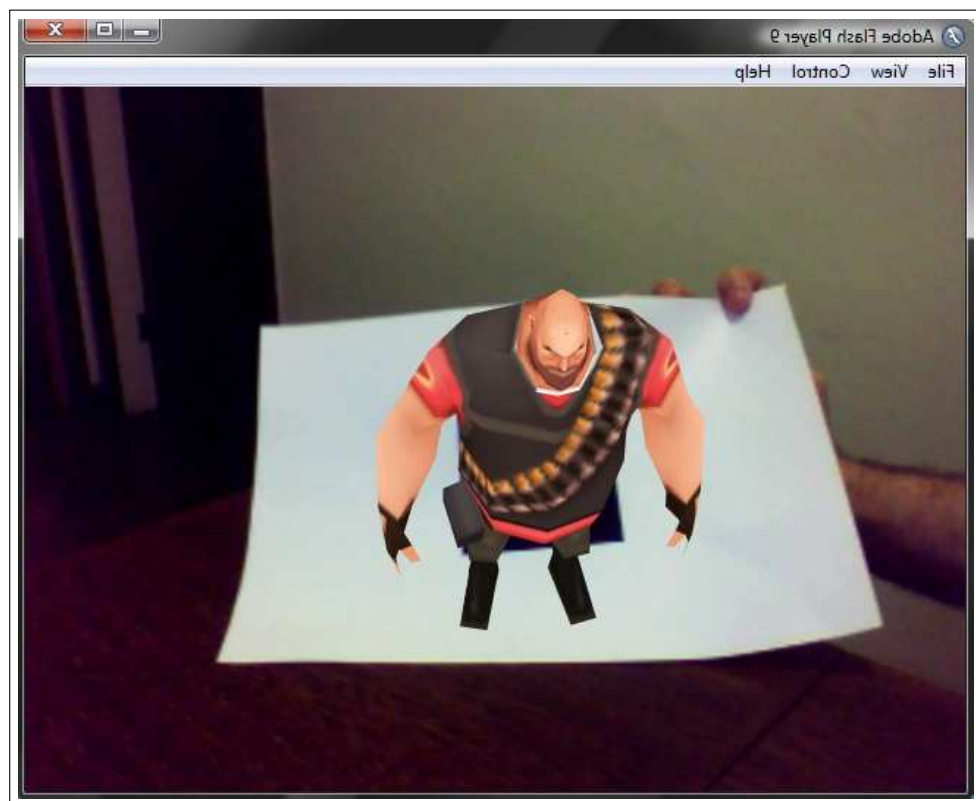


Figura 2.14: *FLARToolkit* Exemplo de Aplicação.

O conjunto de classe *FLARToolkit* dá uma gama de possibilidades para o usuário no sentido de interação. O mesmo pode interagir com os objetos virtuais através do teclado, mouse ou utilizando outros marcadores.

Como pode ser observado na Tabela 2.2, o *FLARToolkit* é único que possibilita o desenvolvimento de aplicações de Realidade Aumentada em diversas plataformas de hardware. Devido à essa característica ele foi escolhido para o desenvolvimento da arquitetura proposta nesta dissertação.

Tabela 2.2: Comparação Entre as Tecnologias de Realidade Aumentada

Tecnologias de RA	Desktop	WEB	Dispositivos Móveis
ARToolkit	✓	✗	✗
ArtoolkitPlus	✗	✗	✓
ARTag	✓	✗	✗
ARStudio	✓	✗	✗
JARToolkit	✓	✗	✗
DART	✓	✗	✗
FLARToolkit	✓	✓	✓

2.4 Sistemas Distribuídos

Segundo (TANENBAUM; STEEN, 2002) “Um Sistema Distribuído é uma coleção de computadores independentes que se apresentam aos seus usuários como um único sistema coerente”.

De acordo com (COULOURIS, 2005) “Um Sistema Distribuído é um sistema no qual os componentes (*hardware* e *software*) estão localizados em uma rede de computadores e se comunicam e coordenam suas ações somente por troca de mensagens”.

Outra definição dada por (MULLENDER, 1993) diz que: “Definir exatamente um Sistema Distribuído é **perigoso**, pois uma característica importante (ou que pode ser importante no futuro) pode ser esquecida.

Segundo (LAMPART, 2006) “Um Sistema Distribuído é aquele que “roda” em um conjunto de máquinas sem memória compartilhada, aparecendo como um único computador para seus usuários”.

Um Sistema Distribuído, segundo (ECKHOUSE; STANKOVIC; DAM, 1978) “é formado por um conjunto de módulos, compostos pelo menos por processador-memória, interligados frouxamente por meio de um subsistema de comunicação de topologia

arbitrária. Esse *hardware* deve oferecer facilidades de comunicação entre processadores e entre processos, os quais, cooperando sob um controle descentralizado, possibilitam a execução de programas de aplicação”.

Neste trabalho será adotada a definição de (COULOURIS, 2005). Devido ao fato de concordamos com tal definição e utilizarmos um *middleware* para a comunicação através de mensagens.

2.4.1 Características de um Sistema Distribuído

As características de um Sistema Distribuído segundo (KIRNER; MENDES, 1988; SPECTOR, 1982) são:

1. Multiplicidade de nós de processamento para permitir aumento no desempenho, tolerância a falhas (presença de defeito interferências indevidas, que aparecem tanto em nível de *hardware* quanto de *software*), disponibilidade do sistema ou uso de dados geograficamente dispersos.
2. Mecanismos de comunicação para suportar comunicações entre os componentes do sistema distribuído.
3. Isolamento entre componentes para suportar controle de diferentes entidades administrativas, tolerância a falhas e disponibilidade.
4. Possibilidade de expansão para permitir crescimento incremental.
5. Mecanismos de detecção de erros para obter disponibilidade e tolerância a falhas.
6. Redundância para obter disponibilidade e tolerância a falhas, pois, uma vez os erros tenham sido detectados, a redundância é que permitirá a recuperação.
7. Dispersão geográfica para permitir que os recursos sejam separados geograficamente.

Sistemas Distribuídos devem possuir obrigatoriamente, as características 1 e 2 mencionadas acima, e, excepcionalmente, algumas das outras características (KIRNER; MENDES, 1988; SPECTOR, 1982).

Outra forma de caracterizar os Sistemas Distribuídos são (PARDO, 1979; KIRNER; MENDES, 1988):

1. O *hardware* básico de um sistema distribuído consistirá de uma rede de computadores.
2. A arquitetura física do sistema de comunicação apresenta pouca importância do ponto de vista lógico, podendo impor maior impacto em questões como desempenho.
3. *Software* é a palavra-chave em sistemas distribuídos. Em um sistema distribuído, a cooperação entre os sistemas remotos é bastante elaborada, uma vez que envolve a implementação de compartilhamento implícito de múltiplos recursos remotos.

2.4.1.1 Tolerância a Falhas

Os sistemas distribuídos possuem múltiplas partes de *hardware* e de *software* funcionando em conjunto. A chance de alguma dessas partes falhar é maior do que em um sistema local (KIRNER; MENDES, 1988). Em sistemas distribuídos é desejável que os aplicativos continuem a funcionar corretamente na presença de falhas no *hardware*, *software*, ou na comunicação (COULOURIS, 2005).

Sistemas distribuídos podem falhar por diversos fatores devido a sua complexidade. Para propiciar confiabilidade a um sistema distribuído é necessário que apresente tolerância a falhas (RIBEIRO, 2006).

Segundo (COULOURIS, 2005), “a confiabilidade é obtida por intermédio da redundância - o provisionamento de vários recursos para que o sistema e o *software* aplicativo possam se reconfigurar e continuar a executar suas tarefas na presença de falhas”.

2.4.2 Sistemas Distribuídos de Realidade Virtual e Aumentada

As aplicações de Realidade Virtual e Aumentada podem ser vistas sob um aspecto bastante amplo, variando de uma única pessoa, usando um único computador, até muitos usuários, usando um sistema distribuído (RINALDI et al., 2006). Neste contexto os Ambientes Virtuais Distribuídos vêm crescendo e apresentando um elevado potencial de aplicação. Eles são caracterizados como Ambiente Virtual Interativo em que os usuários dispersos geograficamente têm como objetivos a cooperação e o compartilhamento dos recursos computacionais em tempo real usando um suporte de rede de computadores para melhorar o desempenho coletivo por meio de troca de informações (BENFORD et al., 1994; SINGHAL; ZYDA, 1999).

Segundo (RINALDI et al., 2006), “Em Ambientes Virtuais Distribuídos, 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*)”.

2.4.3 Middleware

Middleware é uma camada de software que proporciona uma abstração e assim esconde a heterogeneidade entre as redes, hardware e sistema operacional, ficando localizado na camada entre o sistema operacional e as aplicações como demonstra a Figura 2.15 (RIBEIRO, 2006). Segundo (COULOURIS, 2005) “O software que fornece um modelo de programação acima dos blocos de construção básicos de processos e de passagem de mensagens é chamado *middleware*. A camada de *middleware* usa protocolos baseados em mensagens entre processos para fornecer suas abstrações de mais alto nível, como as invocações e eventos remotos”.

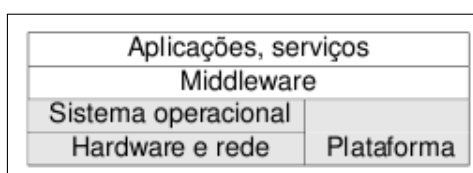


Figura 2.15: Camadas de Software.

Um aspecto importante do *middleware* segundo (COULOURIS, 2005) “é a transparência de localização e a independência dos detalhes dos protocolos de comunicação, sistemas operacionais e hardware de computador. Algumas formas de *middleware* permitem que componentes distintos sejam escritos em diferentes linguagens de programação”.

O *middleware* ultrapassa o nível de comunicação e permite abstrações como Invoção Remota, comunicação de grupos, notificação de eventos, replicação de dados, transmissão de multimídia em tempo real (COULOURIS, 2005).

A transparência de localização é uma característica do *middleware*: um cliente chama um processo, e não sabe se ele é executado local ou remotamente. Também em arquiteturas cliente/servidor o cliente não precisa saber onde está localizado o servidor, pois o mesmo é chamado localmente.

Programadores utilizam *middleware* pela facilidade de implementação onde os mesmos se preocupam com as lógicas de alto nível do software a ser desenvolvido. Por exemplo, detalhes como a comunicação por socket⁴ é solucionado pelo *middleware*.

2.4.4 Chamada de Procedimento Remoto

Uma Chamada de Procedimento Remoto (*Remote Procedure Call*) é uma tecnologia de comunicação entre processos, estando esses processos sendo executados localmente, ou distribuídos por uma rede de computador. É um modelo semelhante ao de chamada de procedimentos locais, pois um processo cliente chama um procedimento que está sendo executado em um processo servidor (COULOURIS, 2005).

Uma diferença importante entre chamadas de procedimento remotas e chamadas de procedimento locais é que, no primeiro caso, a chamada pode falhar por problemas da rede. Nesse caso, não há nem mesmo garantia de que o procedimento foi invocado.

O processo de invocação remota é controlado por uma Thread que é responsável por gerenciar o processo invocador e servidor. O processo invocador primeiro envia

⁴É a identificação única ou a partir da qual a informação é transmitida pela rede, o leitor interessado em socket pode ler o RFC 147 (WINETT, 1971).

uma mensagem para o processo servidor e aguarda uma mensagem de resposta. A mensagem de invocação contém os parâmetros do procedimento e a mensagem de resposta contém o resultado da execução do procedimento. Uma vez que a mensagem de resposta é recebida, os resultados da execução do procedimento são coletados e a execução do invocador prossegue.

Do lado do servidor, um processo permanece em espera até a chegada de uma mensagem de invocação. Quando uma mensagem de invocação é recebida, o servidor extrai os parâmetros, processa-os e produz os resultados, que são enviados na mensagem de resposta. O servidor, então, volta a esperar por uma nova mensagem de invocação.

Interface

Uma interface fornece a definição das assinaturas de métodos, sem especificar sua implementação. Em Java⁵, pode-se implementar uma interface em qualquer classe. Uma interface define os tipos que podem ser usados para declarar o tipo de variáveis, parâmetros e valores de retorno dos métodos (COULOURIS, 2005).

Grande parte das linguagens de programação modernas traz uma forma de organizar um programa, em vários módulos que podem se comunicar. A comunicação entre os módulos pode ser feita por meio de chamadas de procedimentos entre eles ou por acesso direto às variáveis de outro módulo. Para que essa interação entre os módulos seja possível, a interface especifica os procedimentos e as variáveis que podem ser acessadas a partir de outros módulos. Os módulos são implementados de modo a ocultar todas as informações sobre eles, exceto as que estiverem disponíveis por intermédio de uma interface (COULOURIS, 2005).

A comunicação entre os módulos acontece em processos separados onde, as variáveis de um módulo que está sendo executado em um processo não podem acessar as variáveis de outro módulo que está sendo executado em outro processo. A comunicação entre os processos é feita através das RPC⁶, que através da troca de mensagens possibilita a comunicação entre os módulos (COULOURIS, 2005).

⁵Linguagem de programação orientada a objetos.

⁶Chamada de Procedimento Remota

Interface de serviço é utilizado em modelos cliente-servidor. Cada servidor fornece um conjunto de procedimentos (funções) para o uso dos clientes. Interface de serviço segundo (COULOURIS, 2005), “é usado para se referir à especificação dos procedimentos oferecidos por um servidor, definindo os tipos dos argumentos de entrada e saída de cada um dos procedimentos”.

Interfaces remotas especificam os métodos de um objeto que está disponível em outro processo. Segundo (COULOURIS, 2005) “uma interface remota especifica os métodos de um objeto que estão disponíveis para invocação por parte dos objetos de outros processos, definindo os tipos dos argumentos de entrada e saída de cada um deles”. As interfaces remotas possibilitam passar objetos como argumentos para outro processo.

2.5 Modelo Cliente/Servidor

O modelo Cliente/Servidor foi criado tendo como base a descentralização dos dados e recursos de processamento, em oposição ao modelo centralizado utilizado em computadores como Mainframe (BATISTI, 2010). No modelo Cliente/Servidor, conforme indicado pela Figura 2.16, em uma rede de computadores, existem uma ou mais máquinas que atuam como Servidores, disponibilizando recursos para as demais máquinas, as quais atuam como Clientes.

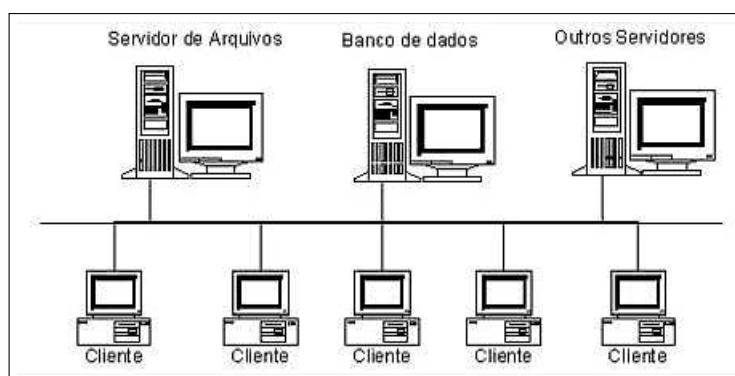


Figura 2.16: Modelo Cliente Servidor Tradicional (BATISTI, 2010).

Como demonstra Figura 2.16, temos Servidores para Arquivos, Banco de dados e outros por exemplo: Servidores de impressão, Servidores Web, etc.

2.5.1 Modelo Cliente/Servidor de 2 Camadas

No Modelo Cliente/Servidor de duas camadas (Figura 2.17), o programa é instalado na máquina do cliente, e faz requisições e consultas em um máquina servidora que hospeda um banco de dados, esta máquina é um Servidor de Banco de Dados (BATISTI, 2010). No modelo de duas camadas, o cliente é responsável pelas seguintes funcionalidades:

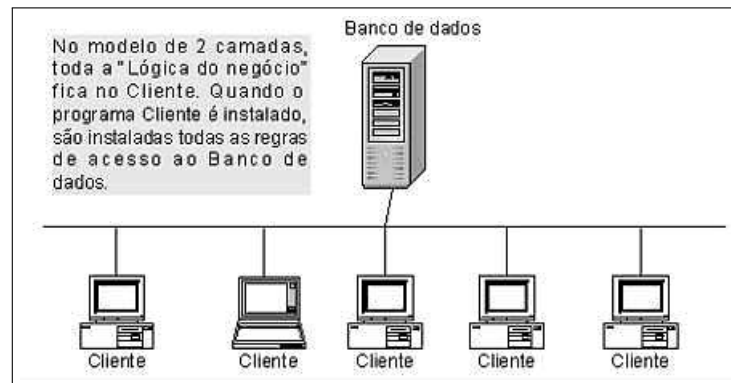


Figura 2.17: Modelo Cliente Servidor de Duas Camadas (BATISTI, 2010).

- Apresentação: camada responsável pela parte visual da aplicação, é a camada onde o usuário interage com o software.
- Lógica de Negócio: definem a maneira como os dados serão acessados e processados.
- Persistência: neste modelo esta camada é responsável pelos dados que nesta arquitetura é centralizada em um servidor.

2.5.2 Modelo Cliente/Servidor de 3 Camadas

Como a evolução do modelo de 2 camadas, e o advento da Internet, surgiu o modelo em três camadas (Figura 2.18). A conceito do modelo em três camadas é retirar a regra de negócio do cliente, e centraliza-la em um ponto, o qual é chamado de Servidor de Aplicações (BATISTI, 2010). Com isso as três camadas são as seguintes:

- Apresentação: camada responsável pela parte visual da aplicação, é a camada onde o usuário interage com o software.

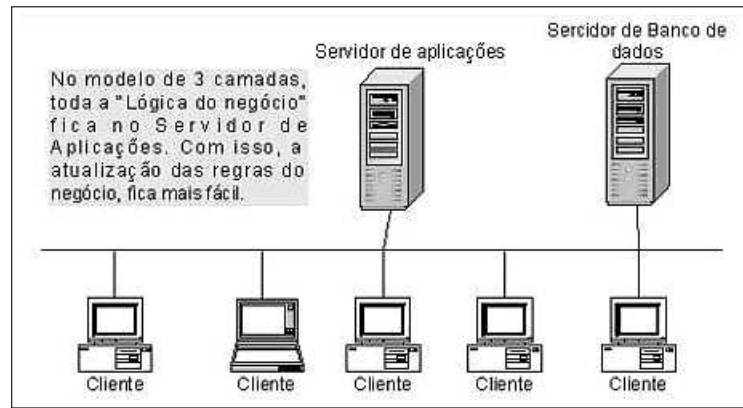


Figura 2.18: Modelo Cliente Servidor de Três Camadas (BATISTI, 2010).

- Lógica de Negócio: definem a maneira como os dados serão acessados e processados e são alocados e centralizados em um servidor de Aplicação.
- Persistência: neste modelo esta camada é responsável pelos dados que na esta arquitetura é centralizada em um servidor.

2.5.3 Modelo Cliente/Servidor de 4 Camadas

O conceito do modelo de 4 camadas (Figura 2.19) é retirar a apresentação do cliente e centralizá-las em um determinado ponto, o qual na maioria dos casos é um servidor Web (BATISTI, 2010). Com isso, o próprio Cliente deixa de existir como um programa que precisa ser instalado em cada computador da rede.

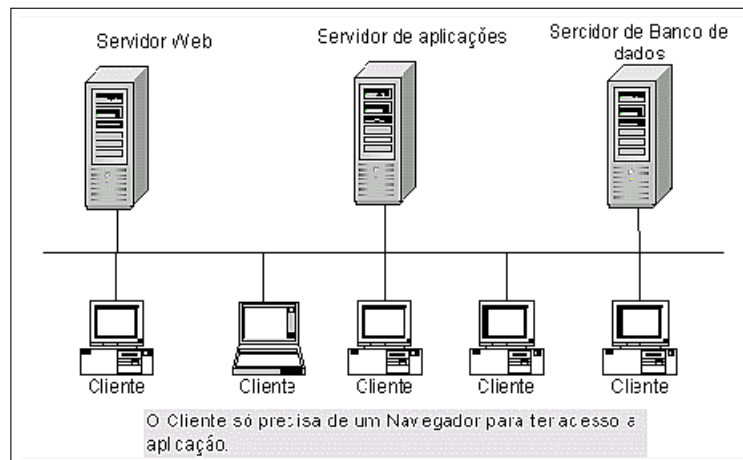


Figura 2.19: Modelo Cliente Servidor de Quatro Camadas (BATISTI, 2010).

- Cliente: para visualizar as aplicações desenvolvidas, o cliente deve utilizar um Navegador Web (*browser*), desta forma não há a necessidade do cliente instalar nenhum aplicativo em sua máquina.
- Apresentação: passa para um Servidor Web. A interface pode ser composta de páginas HTML, ASP, Flex, JSP ou qualquer outra tecnologia capaz de gerar conteúdo para o Navegador. Com isso as alterações na interface da aplicação, são feitas diretamente no servidor Web, sendo que estas alterações estarão, automaticamente, disponíveis para todos os Clientes.
- Lógica de Negócio: definem a maneira como os dados serão acessados e processados e são alocados e centralizados em um servidor de Aplicação.
- Persistência: neste modelo esta camada é responsável pelos dados que na esta arquitetura é centralizada em um servidor.

Para o desenvolvimento da arquitetura a ser apresentada nessa dissertação será utilizado o modelo cliente servidor de 4 camadas.

2.6 Conclusão

Os conceitos apresentados neste capítulo são essenciais para o entendimento de Realidade Virtual e Aumentada. O objetivo aqui não é ensinar Ambientes Distribuídos de Realidade Virtual e Aumentada, mas sim dar, ao leitor, uma visão dos conceitos de Realidade Virtual e Aumentada, e ambientes distribuídos de Realidade Virtual ou não. Dessa forma, o próximo capítulo apresenta o estado da arte dessa pesquisa, através dos trabalhos relacionados.

Capítulo 3

Trabalhos Relacionados

3.1 Introdução

Com o intuito de encontrar novas tecnologias que auxiliem na criação do sistema proposto, foi realizado um estudo do estado da arte em relação a distribuição de Realidade Aumentada, em ambientes que possibilitem a interação em tempo real. Este capítulo tem como objetivo, apresentar uma visão dos principais trabalhos encontrados na literatura e suas principais características.

3.1.1 Metodologia Para a Análise

Dentre os trabalhos analisados, verificou-se as diversas soluções de software, associadas a Realidade Aumentada e Distribuição, que permitem a interação em tempo real e que proporcionem a distribuição das interações realizadas por meio de rede de computadores (local ou não). Adicionalmente, buscou-se softwares que pudessem ser utilizados como ferramentas de apoio pedagógico (softwares educacionais). Para a construção deste capítulo verificou-se trabalhos em duas vertentes de software: **Interfaces de Realidade Aumentada e Interface Distribuída de Realidade Aumentada**. Neste aspecto, avaliaram-se os seguintes tópicos:

- Arquitetura de Distribuição! Trabalhos que possuam uma variedade de softwares, que trabalhando em conjuntos possibilitam a distribuição através de uma

rede de computadores.

- Distribuição dos Modelos! Trabalhos que viabilizem a distribuição de modelos virtuais ou suas respectivas posições através de uma rede de computadores.
- Distribuição na Internet! Trabalhos que possibilitem a distribuição por meio da Internet.
- Distribuição em Lan! Trabalhos que possibilitem a distribuição por uma rede local.
- Execução na Web! Trabalhos que possibilitam a execução do ambiente na web, e a distribuição das interações entre todos os participantes.
- Sistemas multiplataforma! Software que sejam capazes de executar em diversas plataformas de hardware independente do sistema operacional, sem na necessidade de recompilar o código.

3.2 Ambientes Colaborativos com Realidade Aumentada

(GERALDI; KIRNER, 2005), propõe uma ferramenta que possibilita a colaboração entre usuários por meio de uma rede de computador, utilizando o software *ARToolkit* e técnica de sockets. Para chegar ao resultado esperado os autores modificaram o *ARToolkit* permitindo a importação e exportação de posições dos objetos virtuais, cadastrados através da rede.

Os mesmos são cadastrados em todas as máquinas, de forma que, quando um dado usuário adiciona a placa marcadora na área de visão da webcam, o objeto virtual associado à placa marcadora aparece na cena, de forma que, ao mover a placa, sua posição é repassada para a rede e as máquinas remotas recebem tais informações, reposicionando o objeto virtual na mesma localização e permitindo que os usuários visualizem e manipulem seus objetos no mesmo ambiente (Figura 3.1).

Os autores frisam que, para evitar sobrecarga de tráfego na rede, o envio das

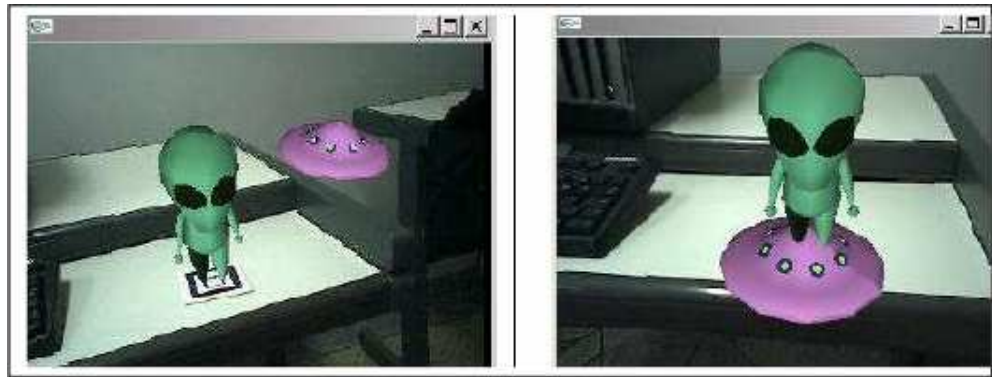


Figura 3.1: Visualização dos Objetos Virtuais em Realidade Aumentada em um Ambiente Colaborativo (GERALDI; KIRNER, 2005).

atualizações de posições é feito apenas quando uma das três coordenadas(x,y,z), varia além de uma dada referência - meio centímetro.

Os autores concluíram que a utilização da Realidade Aumentada beneficia de maneira muito significativa a percepção dos usuários e sua interação, principalmente em ambientes colaborativos.

O software desenvolvido não possibilita ao usuário adicionar novos modelos virtuais de forma dinâmica, e não foi projeto pensando em um ambiente suportado pela web.

3.3 Aplicações Educacionais em Ambientes Colaborativos com Realidade Aumentada

A Colaboração remota com Realidade Aumentada, segundo (KIRNER; ZORZAL, 2005) “baseia-se em interfaces computacionais que compartilham informações e sobrepõem os espaços físicos dos vários usuários (mesa, por exemplo), utilizando uma rede de computadores. Assim, cada usuário pode colocar objetos virtuais sobre a mesa, de forma a visualizar todo o conjunto de objetos e manipulá-los.”.

Nesse trabalho, o usuário por meio do software *ARToolkit* posiciona as placas marcadoras no campo de visão da webcam, desta forma visualizando os objetos virtuais na cena. Esses objetos são manipulados e visualizados pelos participantes.

Com o objetivo de convalidar o trabalho foi desenvolvido um jogo da velha utilizando dois usuários conectados por uma rede de computadores como demonstra a Figura 3.2.

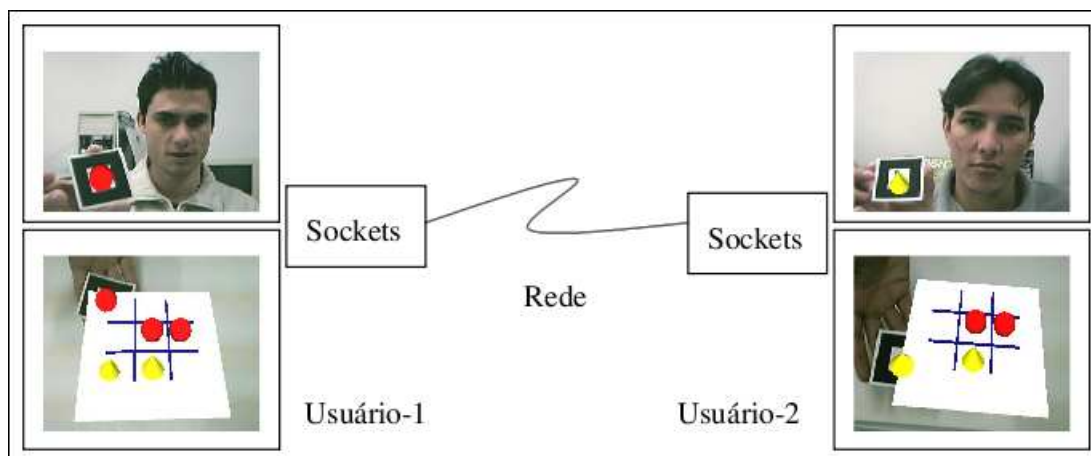


Figura 3.2: Ambiente Colaborativo com Realidade Aumentada em Rede (KIRNER; ZORZAL, 2005).

Para o desenvolvimento do jogo em questão foi utilizado o software *ARToolkit* para o compartilhamento do jogo da velha nos ambientes físicos de cada usuário, constituindo-se em um único ambiente sobreposto. A técnica de comunicação entre os sistemas através da rede foi utilizando a técnica de sockets.

Como pode ser observado na Figura 3.2, ao posicionar um marcador em qualquer dos ambientes remotos, um tabuleiro virtual é gerado para cada usuário. Alguns marcadores ficam escondidos sobre o tabuleiro virtual, e são responsáveis pelo posicionamento das peças do jogo.

A estratégia utilizada pelo autores para diminuir o tráfego na rede foi verificar se houve uma mudança de posicionamento dos marcadores nos sistemas, dando uma tolerância de 5 centímetros. Caso as posições (X,Y,Z) sejam alteradas ultrapassando a tolerância, o novo posicionamento é enviado através rede, assim não é encaminhado um tráfego de informações desnecessária pela rede, o que viabiliza a utilização do software em questão em redes de baixa velocidade (KIRNER; ZORZAL, 2005).

O jogo desenvolvido não executa em ambientes Web, pelo fato de ter sido desenvolvido utilizando a ferramenta *Artoolkit*, que foi projetada para executar em Desktop ou seja apenas local na máquina do usuário, neste caso o usuário tem que instalar e

configurar o *software* em seu computador para que ele possa utilizar a ferramenta.

3.4 Uma Arquitetura para Distribuição de Ambientes Virtuais de Realidade Aumentada

(SILVA, 2008), apresentou uma arquitetura para distribuição de ambientes virtuais de Realidade Aumentada como ferramenta de apoio a projetos de ensino. Foi criada uma interface cognitiva de Realidade Aumentada que possibilita a interação do usuário com o ambiente virtual por meio de menus de interação e também por meio de marcadores reais, viabilizando ao usuário o estudo da **fisiologia vegetal**¹.

Para o desenvolvimento da arquitetura proposta por (SILVA, 2008), foram utilizadas as seguintes tecnologias:

- O *ARToolkit* que possibilita o desenvolvimento de aplicações de Realidade Aumentada em Desktop.
- O Framework CORBA², que viabiliza a comunicação entre os clientes e o servidor.
- Objetos virtuais modelados em VRML³, são carregados no *ARToolkit* onde são transformados em matriz de pontos e vértices OpenGL⁴.

A Figura 3.3 demonstra a arquitetura proposta e suas camadas.

Nessa arquitetura, o usuário interage com um ambiente de Realidade Aumentada de tal forma que as informações capturadas são repassadas para a Camada de Comunicação. Desta forma, a aplicação-cliente lê os dados da Camada de Comunicação e verifica o que está sendo posicionado na cena. Em seguida, envia as informações

¹A fisiologia vegetal estuda os fenômenos vitais que concerne às plantas.

²*Common Object Request Broker Architecture*.

³*Virtual Reality Modeling Language* é um padrão de formato de arquivo para realidade virtual, o leitor interessado sobre VRML deve ler (GORALSKI; POLI; VOGEL, 1996; AMES; NADEAU; MORELAND, 1997).

⁴API Gráfica free utilizada no desenvolvimento de aplicativos gráficos, ambientes 3D, jogos, entre outros.

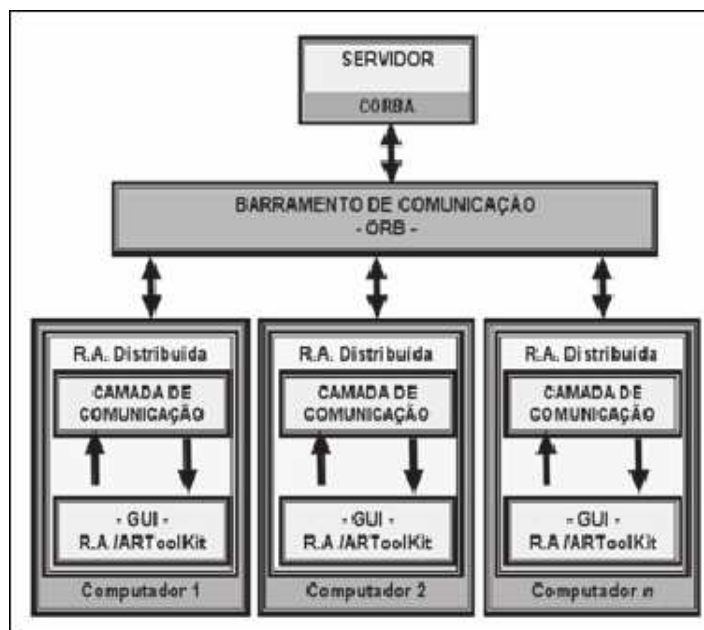


Figura 3.3: Arquitetura Proposta para a Distribuição do Sistema (SILVA, 2008).

para a aplicação-servidora que, por sua vez disponibiliza as informações para todas as outras aplicações-clientes na rede (SILVA, 2008).

Todos os objetos virtuais são carregados localmente e a posição dos mesmos é distribuída na rede local, através do Framework CORBA.

O software desenvolvido não possui a flexibilidade de executar em diversos ambientes operacionais, sem que haja a necessidade de recompilar o código, pelo fato de ter sido desenvolvido utilizando o *ARToolkit*. Outro fator limitante é que a arquitetura desenvolvida não possibilita adicionar objetos virtuais de forma dinâmica. A arquitetura em questão não foi projetada para executar na Web.

3.5 Uma Arquitetura para a Distribuição de Realidade Virtual e Aumentada

No trabalho de Distribuição de Realidade Virtual e Aumentada (NOGUEIRA et al., 2009), os autores desenvolveram uma arquitetura que viabiliza a distribuição de objetos virtuais no formato VRML, para todos os participantes através de uma rede de computador, utilizando *sockets* para a transferência de arquivos. A arquitetura pode

ser visualizada na Figura 3.4.

- A Camada GUI (Interface Gráfica com o Usuário), possui comunicação via texto, uma biblioteca comum de objetos virtuais no formato VRML e um ambientes de visualização de Realidade Virtual e outro de Realidade Aumentada.
- A Camada de Ambiente Virtual é responsável por conter os objetos virtuais VRML cadastrados, podendo inserir novos objetos. Essa camada é responsável pela manipulação dos objetos virtuais, dando ao usuário a possibilidade de alterar o código em tempo real.
- A Camada do Distribuidor de Eventos para rede, possibilita através do mecanismo de *multi-thread*⁵ o envio dos eventos e dos arquivos virtuais, para os clientes conectados na rede.

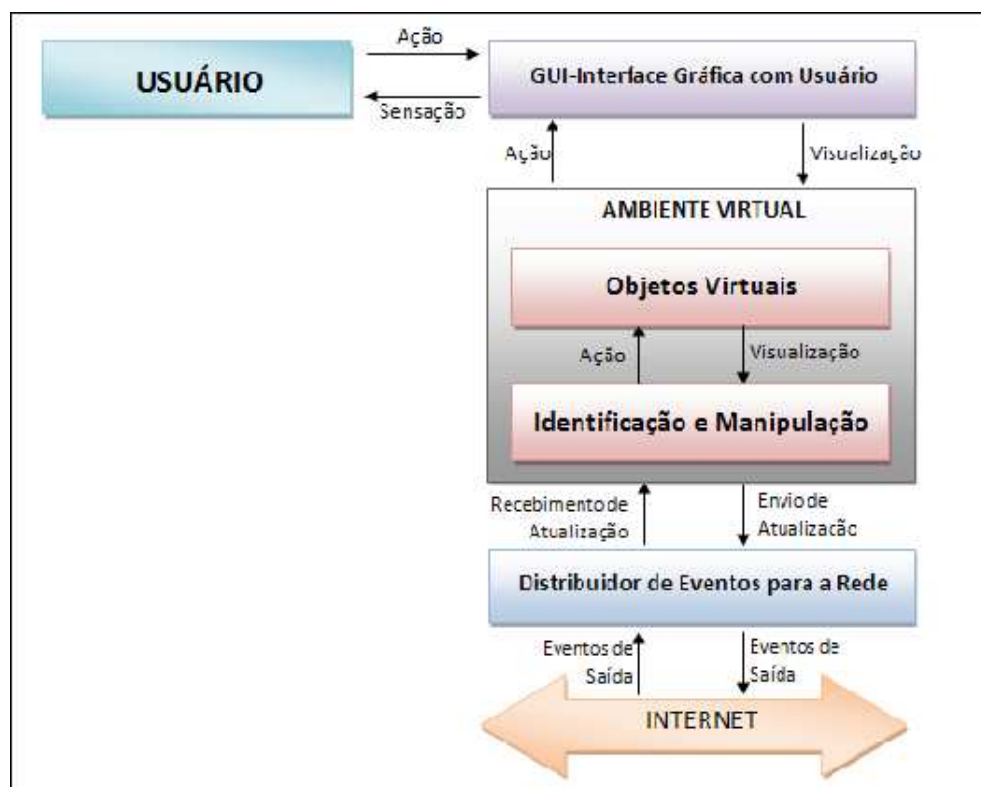


Figura 3.4: Arquitetura do Sistema (NOGUEIRA et al., 2009).

O software desenvolvido viabiliza a visualização dos objetivos virtuais em Realidade Aumentada, utilizando o conjunto de biblioteca *ARToolkit*. Um fator relevante

⁵Uma *Thread* é um fluxo em execução, *multi-thread* são vários fluxos em execução em um programa. O leitor interessado em *Thread* pode ler (COULOURIS, 2005; TANENBAUM; STEEN, 2002).

neste trabalho é que o mesmo possibilita a interação com os objetos virtuais como escala, rotação e translação com todos os participantes em seu módulo de Realidade Virtual, essas características porém, não são abordadas no módulo de Realidade Aumentada do sistema.

Outro fator relevante no trabalho é que o mesmo utiliza a API *winsock*⁶ do Sistema Operacional Windows, o que limita a execução da aplicação em questão, em outros Sistemas Operacionais como o Linux, MAC/OS e outros. O trabalho não utiliza a web como mecanismo de execução.

3.6 Sistema de Autoria em Ambiente Colaborativo com Realidade Aumentada

(SANTIN, 2008), desenvolveu um sistema de autoria em ambiente colaborativo com 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 foi desenvolvido utilizando a linguagem de programação C com base na biblioteca *ARToolKit*, com suporte a objetos virtuais no formato VRML. Oferecendo um ambiente de Realidade Aumentada tangível, para a autoria de ambientes virtuais, possibilitando a colaboração face a face e remota.

A interação face a face no SACRA, 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 as informações foram desenvolvidas. Na interação síncrona distribuída o usuário remotos realizam o trabalho colaborativo em tempo real.

O SACRA possui um mecanismo de interação por meio de marcadores de ações, que implementam funcionalidades específicas para a manipulação dos objetos virtuais. Os marcadores de ações possibilitam que usuários adicionem objetos virtuais e

⁶É uma camada intermediária do Sistema Operacional Windows, e a rede, que possibilita a comunicação de aplicações cliente-servidor (QUINN, 1998).

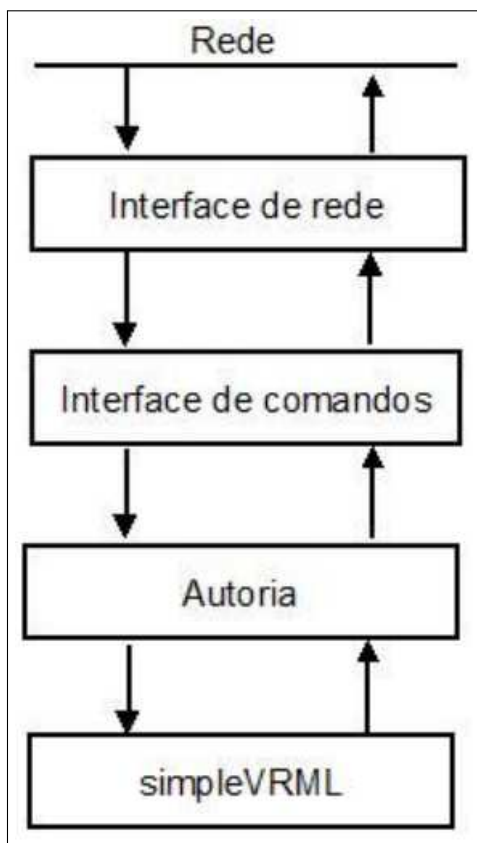


Figura 3.5: ESTRUTURA DO SACRA (SANTIN, 2008).

manipulem os objetos através dos marcadores.

A estrutura do sacra se baseia em quatro camadas Figura 3.5, sem a utilização de um servidor. O Sacra utiliza como base para a comunicação entre os clientes o NETARTOOLKIT (OLIVEIRA, 2008), que utiliza o modelo de comunicação peer to peer.

Caso o usuário necessite atualizar o software, terá que realizar o procedimento de atualização, em todas as máquinas. Tal procedimento não seria necessário se a aplicação fosse hospedada em um servidor WEB, e utiliza-se a topologia Cliente/Servidor.

3.7 Virtual Reality Tools for Internet-Based Robotic Teleoperation

(YANG; CHEN, 2004), possibilita controlar um robô remotamente por intermédio da internet. Os autores frisam que a instabilidade da internet dificulta as operações remotas, utilizando a técnica da vídeo conferência.

Para simular o ambiente real, foi desenvolvido um ambiente virtual, e projetada a seguinte arquitetura, como demonstra a Figura 3.6.

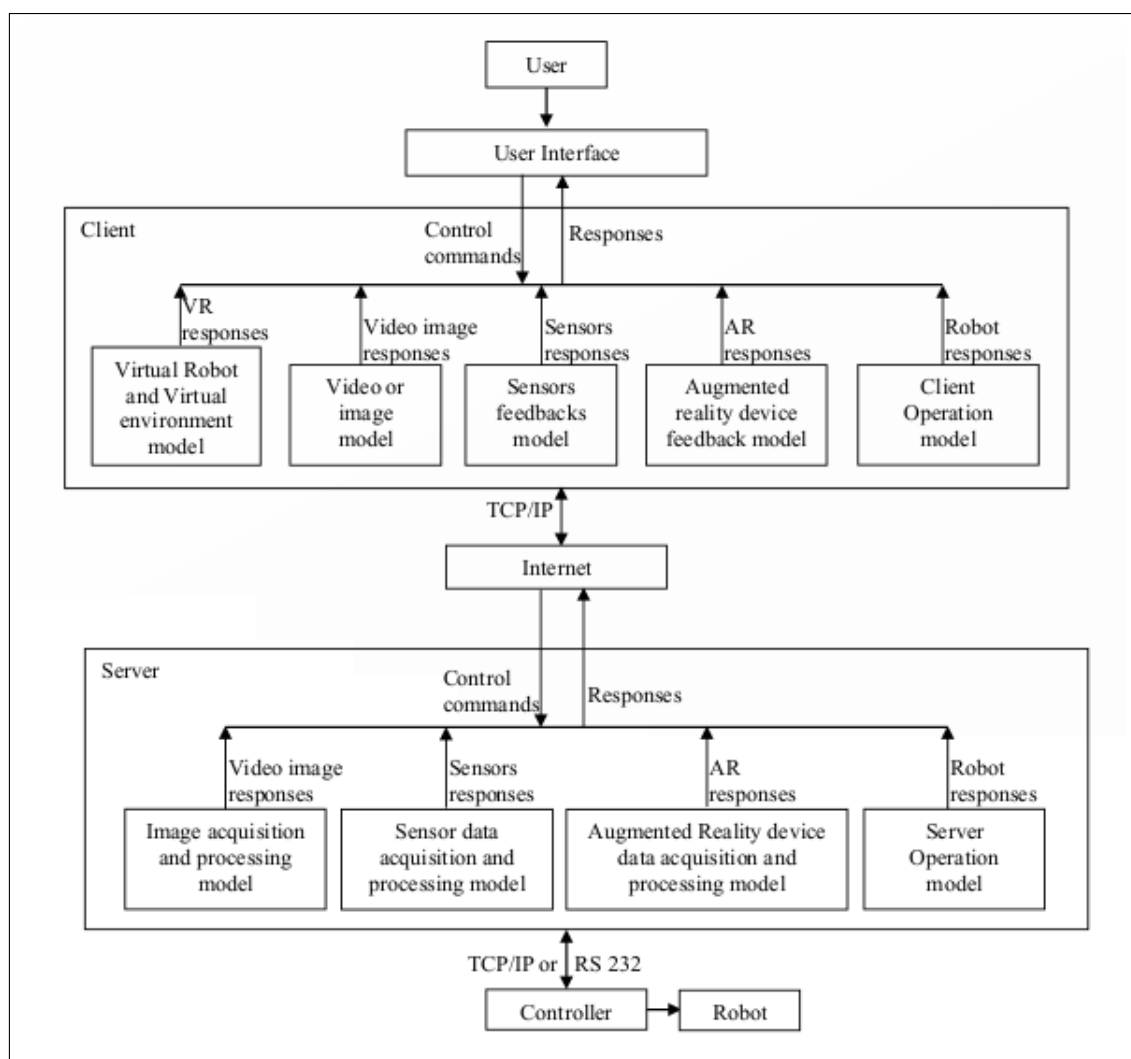


Figura 3.6: Arquitetura Desenvolvida (YANG; CHEN, 2004).

Os clientes através da JVM (*Java Virtual Machine*) executam em seus computadores o Ambiente Virtual (Figura 3.7). Todas as interações são realizadas no ambiente virtual, tem efeito direto no robô real.

A arquitetura foi projetada utilizando três módulos: cliente, servidor e robô.

O módulo servidor recebe todas as requisições enviadas pelos clientes por intermédio da internet, tais mensagens são processadas pelo computador servidor, e as ações são realizadas no robô. As operações realizadas pelo robô são enviadas para os clientes e tais alterações ocorrem no Ambiente Virtual. Segundos os autores o Robô foi controlado remotamente em diversas regiões geográficas.

A Realidade Aumentada pode ser utilizada no sistema para obter uma melhor visualização e ajudar o operador à obter uma maior imersão do Ambiente Virtual.

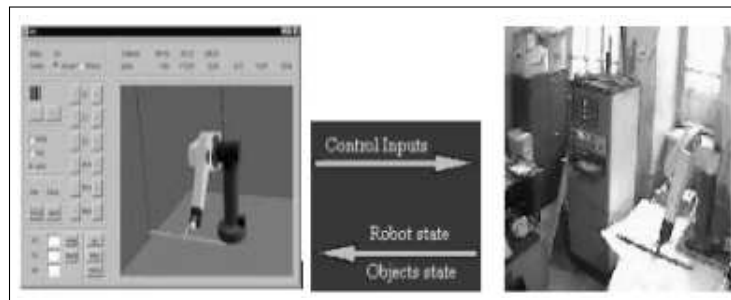


Figura 3.7: Ambiente Virtual em Execução (YANG; CHEN, 2004).

O trabalho apresenta uma arquitetura completa e eficaz, que possibilita controlar um robô remotamente através da internet. Porém o ambiente de execução é desktop, ficando sua limitação em sua execução, onde a mesma não foi projetada para executar em um ambiente web.

3.8 Sumário e Conclusões

Baseado no estudo feito neste capítulo, é possível visualizar o estado da arte sobre distribuição de ambientes de Realidade Aumentada, e as principais tecnologias utilizadas em cada um destes trabalhos.

A Tabela 3.1 demonstra um comparativo entre os trabalhos analisados. Através desta análise, pode-se perceber que em nenhum destes trabalhos foi desenvolvido um software que viabilize a distribuição de Realidade Aumentada na Web e que execute em sistemas operacionais distintos. E até mesmo em plataforma de hardware distintas como smartphones (Android, Windows Mobile).

Um fator limitante na maioria dos trabalhos analisados é que utilizam o *ARToolkit*, que não possibilita o desenvolvimento de aplicações voltadas para Web. Por esse fator, neste trabalho foi utilizado o *FARToolkit*, que viabiliza o desenvolvimento de aplicações de Realidade Aumentada na Web.

Portanto, um sistema de distribuição de Realidade Aumentada que apresente estas características seria muito importante porque:

Possibilita que novas ferramentas e aplicações sejam desenvolvidas e aplicadas nas diversas áreas de conhecimento, tais como: Educação à Distância, treinamento, Engenharia, Medicina e áreas afins.

Assim, o próximo capítulo apresenta a proposta de arquitetura de um sistema de Realidade Aumentada que contemple estas funcionalidades.

Tabela 3.1: Características Analisadas nos Trabalhos Relacionados.

Trabalhos Relacionados	Distribuição RA	Distribuição LAN	Distribuição Internet	Execução na Web	Multiplataforma	Distribuir Novos Modelos
Ambientes Colaborativos com Realidade Aumentada	✓	✓	✓	✗	✗	✗
Aplicações Educacionais em Ambientes Colaborativos com Realidade Aumentada	✓	✓	✗	✗	✗	✗
Uma Arquitetura para Distribuição de Ambientes Virtuais de Realidade Aumentada	✓	✓	✓	✗	✗	✗
Uma Arquitetura para a Distribuição de Realidade Virtual e Aumentada	✓	✓	✓	✗	✗	✓
Sistema de Autoria em Ambiente Colaborativo com Realidade Aumentada	✓	✓	✓	✗	✗	✓
Virtual Reality Tools for Internet-Based Robotic Teleoperation	✓	✓	✓	✗	✗	✗

Capítulo 4

Arquitetura do Sistema Proposto

4.1 Introdução

A partir dos estudos realizados, foi possível conceder uma arquitetura para a Distribuição de Realidade Aumentada pela WEB onde possibilite o usuário interagir e distribuir objetos virtuais através da Internet¹. Este capítulo apresenta esta arquitetura.

4.2 Engenharia de Requisitos

A Engenharia de Requisitos, auxilia os projetistas à entender quais as necessidades do software a ser desenvolvido. Para que tenha sucesso, é preciso antes de iniciar a codificação, entender qual é a necessidade do cliente (PRESSMAN, 2006).

No caso da Arquitetura proposta nesta dissertação, buscou-se modelar utilizando a notação da UML, prevendo quais os requisitos exigidos para o funcionamento da arquitetura em questão, com base no diagrama de caso de uso.

¹As redes de computadores estão por toda parte. A Internet é uma delas, assim como as muitas redes das quais ela é composta (COULOURIS, 2005).

4.3 Diagrama de Caso de Uso

O diagrama de caso de uso procura, pro meio de uma linguagem simples, possibilitar a compreensão do comportamento externo do sistema e as funcionalidades oferecidas por ele, possibilitando apresentar o software por intermédio de uma perspectiva do usuário (GILLEANES, 2009).

Através desse diagrama, o usuário tem uma visão externa geral das funcionalidades que o sistema oferecerá aos usuários, sem se preocupar com a questão de como tais funcionalidades serão implementadas, assim esse diagrama é de grande importância para a compreensão dos requisitos do sistema.

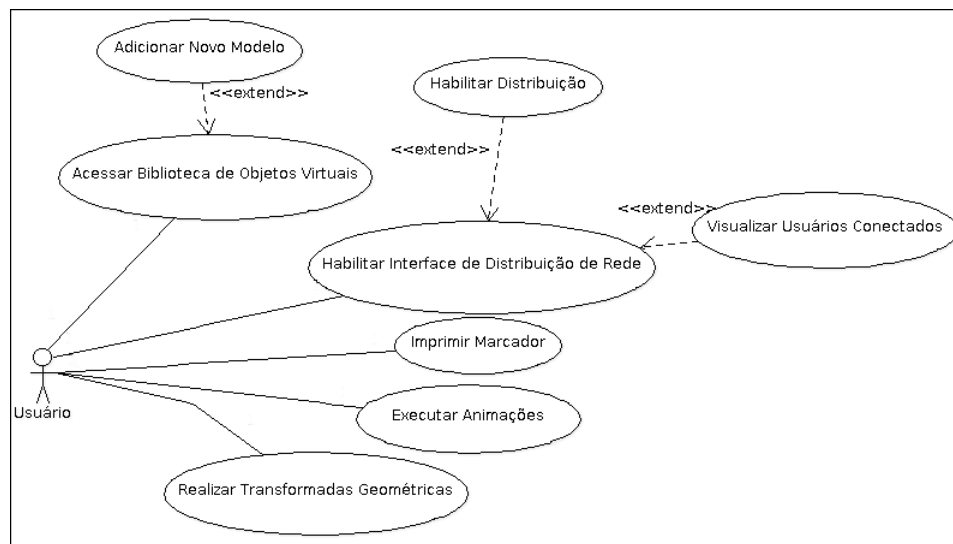


Figura 4.1: Diagrama de Caso de Uso do Sistema de Realidade Aumentada para Web.

Esse diagrama concentra-se em dois itens principais: atores e caso de uso. Os atores representam os papéis desempenhados pelos diversos usuários que poderão utilizar, de alguma maneira, as funcionalidades do software em questão. Um ator pode ser qualquer elemento externo que interaja com o software. Os casos de uso referem-se aos serviços, tarefas, ou funcionalidades que podem ser utilizados de alguma maneira pelos atores que interagem com o sistema (GILLEANES, 2009).

Com o objetivo de auxiliar no entendimento e desenvolvimento do *software* proposto nesta dissertação, foi desenvolvido um diagrama de caso de uso, com o intuito de demonstrar as funcionalidades do sistema, na visão do usuário como demonstra a

Figura 4.1.

Nas tabelas subsequentes estão os cenários de caso de uso de cada funcionalidade do *software*.

Tabela 4.1: Descrição do Caso de Uso Biblioteca Virtual.

Caso de Uso: Biblioteca de Objetos Virtuais	Resumo: O usuário pode adicionar e atualizar a lista de objetos virtuais cadastrados no sistema.
Atores: (professores e alunos)	
Precondições: O usuário acessar a página onde o <i>software</i> está hospedado.	
Descrição: O usuário clica no botão.	
Exceções: Caso o cheque box na interface de distribuição esteja ativado. Ao clicar sobre um objeto virtual na biblioteca todos os usuários visualizam aquele objeto virtual na tela.	
Pós Condições: Se o usuário clicar no botão Adicionar Novo Modelo, poderá adicionar um novo objeto virtual na biblioteca compartilhada. Se o usuário clicar no botão atualizar o sistema verificará se há alguma atualização na biblioteca.	

Tabela 4.2: Descrição do Caso de Uso Interface de Distribuição.

Caso de Uso: Interface de Distribuição de Rede	Resumo: O sistema habilita a distribuição dos eventos para todos os usuários.
Atores: (professores e alunos)	
Precondições: O usuário acessar a página onde o <i>software</i> está hospedado.	
Descrição: O usuário clica no cheque box. Todas as ações realizadas nos menus ou teclado serão repassados pela Web, para todos os usuários. Usuário pode clicar no botão Visualizar Usuários Conectados. Abre uma lista com o nome de todos os usuários conectados no sistema.	
Exceções: Para visualizar quais usuários estão conectados, o cheque box deve estar selecionado.	
Pós Condições: –	

Tabela 4.3: Descrição do Caso de Uso Executar Animações.

Caso de Uso: Menu Motor	Resumo: O sistema abre submenus que possibilita a interação com os objetos virtuais.
Atores: (professores e alunos)	
Precondições: O usuário acessar a página onde o <i>software</i> está hospedado.	
Descrição: O usuário clica no Menu Motor, que abrirá outros dois submenus, possibilitando ao usuário iniciar algumas animações com o motor.	
Exceções: Caso o cheque box de distribuição esteja desmarcado. A simulação será realizada localmente, caso esteja marcado o cheque box. Todos os usuários visualizarão a simulação.	
Pós Condições: Animação é executada.	

Tabela 4.4: Descrição do Caso de Uso Realizar Transformadas Geométricas.

Caso de Uso: Transformações	Resumo: Possibilita ao usuário realizar transformadas geométricas no objeto virtual que está atuando na cena.
Atores: (professores e alunos)	
Precondições: O usuário acessar a página onde o <i>software</i> está hospedado.	
Descrição: O usuário clica no Menu Transformações.	
Exceções: –	
Pós Condições: Todas as transformadas geométricas selecionadas pelo usuário, são aplicadas no objeto virtual que está atuando na cena.	

Tabela 4.5: Descrição do Caso de Uso Imprimir Marcador.

Caso de Uso: Imprimir Marcador	Resumo: O sistema abre um arquivo PDF para impressão do marcador após o usuário clicar no botão “Imprimir Marcador”.
Atores: (professores e alunos)	
Precondições: O usuário acessar a página onde o <i>software</i> está hospedado, e o usuário tenha instalado um leitor de PDF.	
Descrição: O Usuário clica no botão “Imprimir Marcador”. O sistema carrega e exibe o arquivo PDF. O usuário imprime o marcador pelo arquivo PDF aberto.	
Exceções: Fala leitor de PDF: Se o usuário não tiver algum leitor de PDF instalado ele será direcionado para fazer o <i>download</i> , após instalado o usuário deverá clicar novamente no botão para abrir o arquivo e imprimi-lo.	
Pós Condições:	

Com base no diagrama de caso de uso e nos casos de usos apresentados nessa seção, foi desenvolvida uma arquitetura/software que atenda aos requisitos levantados anteriormente.

4.4 Detalhes da Arquitetura

O propósito dessa seção é a apresentação de uma arquitetura de Realidade Aumentada que viabilize a interconexão com uma gama de computadores por meio de uma rede, possibilitando a interação e a distribuição de objetos virtuais em tempo real, com os computadores conectados no sistema. Para tal efeito, a arquitetura foi montada baseada no modelo Cliente/Servidor como demonstra a Figura 4.2.

A arquitetura foi desenvolvida utilizando várias camadas de software. No lado do cliente foi desenvolvida as seguintes camadas:

- GUI: responsável pela visão e interação do usuário com o software.
- Misturador de Ambiente: responsável pelo processamento, das interações do

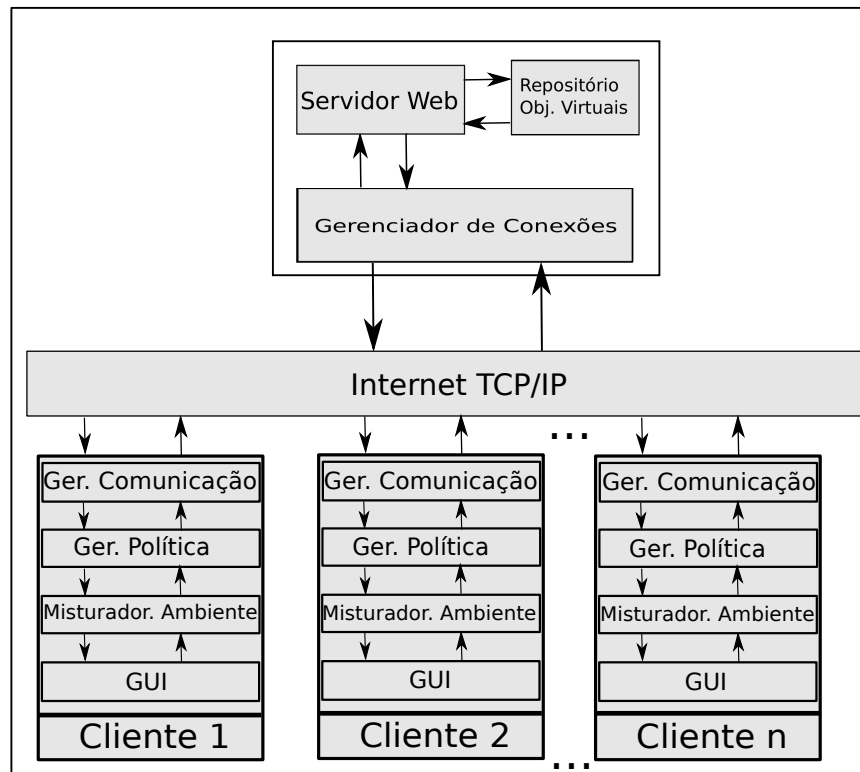


Figura 4.2: Arquitetura do Sistema Proposto.

usuário e visualização no ambiente gráfico de Realidade Aumentada.

- Gerador de Política: é uma camada responsável pela interligação das camadas do Misturador de Ambiente e a camada do Gerador de Comunicação, as regras de negócio estão presentes nesta camada.
- Gerador de Comunicação: Camada responsável pela comunicação entre os módulos cliente e servidor.

No lado do servidor a arquitetura foi dividida em três camadas:

- Gerenciador de Conexões: Camada responsável por gerenciar as conexões dos clientes.
- Servidor Web: Camada responsável pelo processamento das requisições dos clientes.
- Repositório de Objetos Virtuais: Diretório no servidor, onde são armazenados os modelos virtuais.

O propósito da arquitetura ter sido dividida em diversas camadas de software, é de facilitar a manutenibilidade do software, possibilitando ao desenvolvedor no caso de uma manutenção atualizar apenas a camada de seu interesse. Desta forma, o mesmo não corre o risco de inserir códigos defeituosos nas demais camadas da arquitetura.

Para a comunicação entre os módulos a arquitetura utiliza a Internet e seus protocolos. No caso da arquitetura proposta nesta dissertação será utilizando o protocolo TCP/IP para a comunicação entre os módulos cliente e servidor, e seus respectivos usuários conectados na aplicação.

4.4.1 Módulo Servidor

O Módulo Servidor como demonstra a Figura 4.3, tem como finalidade gerenciar as conexões dos clientes e possibilitar a distribuição dos objetos virtuais no Ambiente de Realidade Aumentada dos clientes. Outra característica do servidor é possibilitar a sincronização das animações executadas pelos clientes através de um protocolo definido na seção 5.3.1.

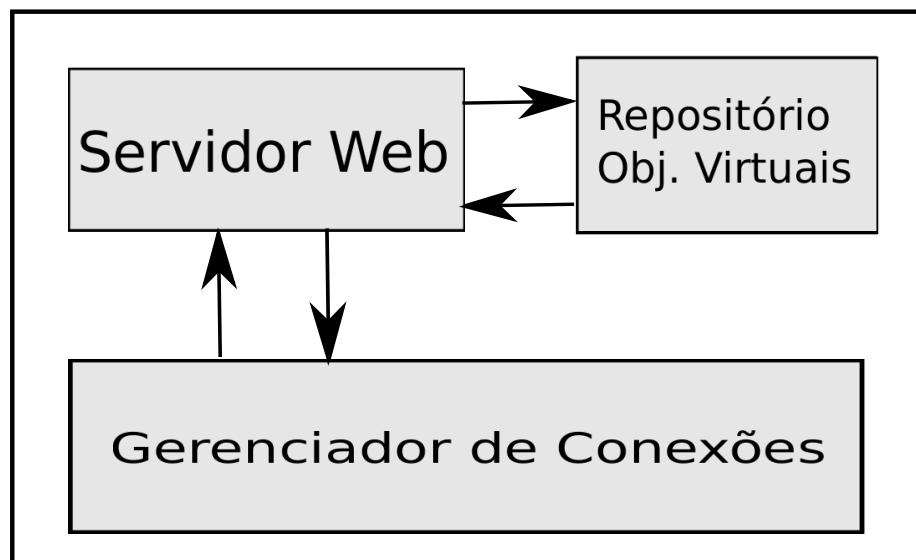


Figura 4.3: Arquitetura do Módulo Servidor.

Para que o Módulo Servidor consiga enviar, processar e receber informações do cliente o mesmo é dividido em três camadas: a camada do gerenciador de conexões, servidor Web e o repositório de objetos virtuais.

A camada onde está localizado Gerenciador de Conexões (Figura 4.3) é responsável por administrar as conexões dos usuários conectados ao módulo servidor. As mensagens enviadas pelos clientes são filtradas e encaminhadas para a camada do Servidor Web, onde serão processadas.

Para que as funcionalidades das camadas fiquem claras, vamos imaginar o seguinte cenário: um determinado usuário conectado no sistema desenvolveu um modelo tridimensional em um modelador 3D de sua preferência, onde o mesmo contém suporte para exportar o objeto virtual tridimensional no formato Collada. Após exportar o objeto virtual, o usuário deseja inserir este objeto na arquitetura proposta para demonstrar seu trabalho com todos os usuários conectados no sistema. Ao enviar o arquivo para o servidor, o mesmo é transformado em um fluxo de bytes e transportado pela Internet utilizando o protocolo AMF (*Action Message Format*). Chegando na camada Gerenciador de Conexões que encapsula o middleware BlazeDS™ (seção 5.2.4), esse fluxo de bytes é convertido para ByteArray e enviado a camada do Servidor Web, que hospeda o servidor GlassFish, o servidor irá salvar esse ByteArray no formato de um arquivo no Repositório de Objetos Virtuais. Após salvar o arquivo, o servidor verificará quais os usuários estão conectados no sistema e solicitará a camada do Gerenciador de Conexões que envie o objeto a todos os clientes conectados no sistema.

Quando o que se deseja compartilhar é a animação de um objeto virtual a todos os usuários do sistema, um cliente específico envia uma mensagem utilizando o protocolo HTTP. Através da camada de Comunicação o cliente envia uma requisição. A requisição ao chegar no servidor é processada pela camada Gerenciador de Conexões, que é responsável por encaminhar a requisição ao serviço correto que a processa, e envia a nova solicitação de animação a todos os usuários conectados no sistema.

4.4.2 Módulo Cliente

O Módulo cliente possui várias camadas de software na Arquitetura Proposta como demonstra a Figura 4.4, tais camadas são descritas abaixo:

A camada da GUI (*graphical user interface*), é a camada de visão da arquitetura do lado do cliente. Essa camada possibilita ao usuário visualizar o ambiente de Realidade Aumentada. Toda a interface gráfica foi desenvolvida utilizando a Linguagem de programação Flex, que possibilita o desenvolvimento de aplicações RIA (*Rich Internet Application*), ou seja aplicativos mais elegantes para Web. Todas as ações e interações do usuário com o Ambiente de Realidade Aumentada são repassadas para a camada abaixo do *Misturador de Ambiente*.

O Misturador de Ambiente encapsula o *FLARToolkit* que é responsável por possibilitar o desenvolvimento de aplicações de Realidade Aumentada. Algumas partes do código original da biblioteca foram modificados para viabilizar a comunicação dessa camada com as outras do sistema.

O Gerador de Políticas é responsável por receber as interações realizadas pela camada de Realidade Aumentada, processá-las e encaminha-las para o Gerador de Comunicação. Ela também é responsável por fazer o sentido inverso: ao receber uma mensagem da camada Comunicação, processar a mensagem recebida e aplicar as alterações cabíveis e repassar para a camada de Realidade Aumentada.



Figura 4.4: Arquitetura do Módulo Cliente.

O Gerador de Comunicação é uma camada de software responsável pela troca de

mensagens entre os módulos cliente e servidor. Os eventos recebidos pela camada do Gerador de Políticas são enviados para o servidor através da Internet por intermédio do *middleware* BlazeDS, onde as mesmas serão processadas no lado do servidor e reenviadas para os clientes. As mensagens recebidas são repassadas para o Gerador de Políticas que aplica as alterações cabíveis e as repassam para o Misturador de Ambiente que é responsável por posicionar/adicionar objetos virtuais na cena.

4.5 Conclusão

A arquitetura apresentada nesta seção, viabiliza e possibilita a distribuição de Realidade Aumentada pela Web, utilizando de algumas tecnologias. Para chegar a este feito, a arquitetura foi desenvolvida utilizando a programação em camadas o que viabiliza a manutenção e a possível extensão da mesma com novos módulos e funcionalidades.

No próximo capítulo, será abordado os detalhes da implementação da arquitetura proposta e um estudo de caso que convalida que o *software* contém carácter educativo.

Capítulo 5

Detalhes de Implementação e Estudo de Caso

5.1 Introdução

Este capítulo descreve as tecnologias correlatas e os detalhes de implementação do *software* (RANET) desenvolvido com o intuito de convalidar a arquitetura proposta nesta dissertação. Também demonstra o funcionamento do sistema proposto baseado em um estudo de caso (*motor de corrente contínua*). Assim, descreve-se o que é necessário pra que o sistema funcione corretamente. Entende-se que um estudo de caso é importante para deixar clara a ideia da utilização do sistema desenvolvido na área educacional; em seguida, é apresentado o funcionamento do sistema implementado.

5.2 Tecnologias Correlatas

Novas tecnologias vêm sendo desenvolvidas e utilizadas para dar suporte a aplicações de Realidade Virtual e Aumentada (RIBEIRO, 2006). O sistema proposto, como demonstra a Figura 5.1, tem como intuito possibilitar que clientes utilizando um navegador Web, execute aplicações de Realidade Aumentada, viabilizando a distribuição e a interação de objetos virtuais por meio da internet. Para alcançar esse objetivo, foi utilizado uma arquitetura cliente-servidor e algumas tecnologias descritas nesse

capítulo.

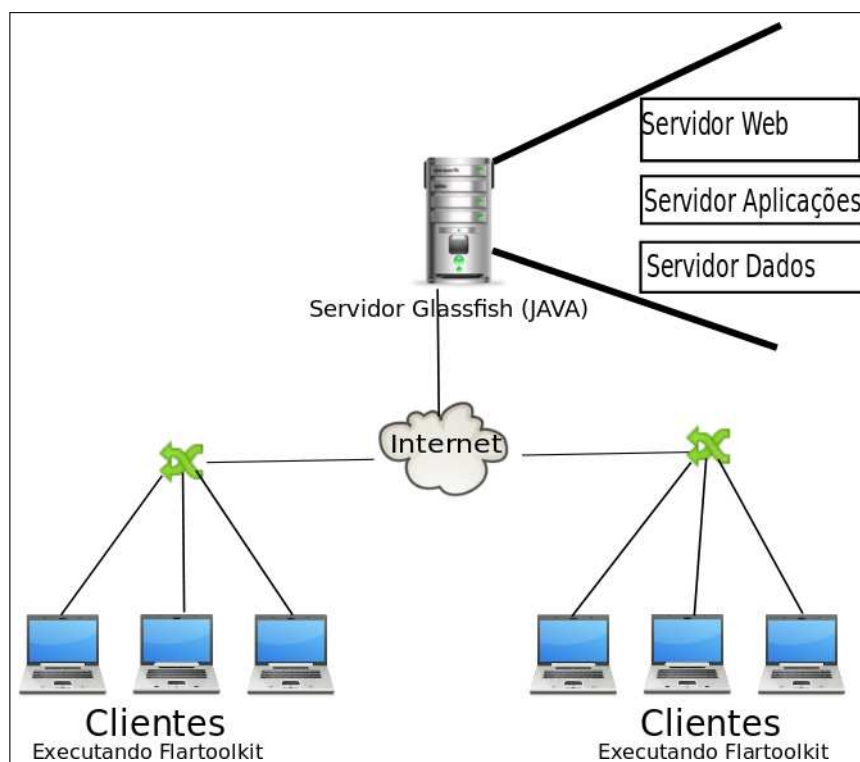


Figura 5.1: Representação do Sistema Proposto.

5.2.1 Linguagem de Programação JAVA

A Linguagem de programação JAVA teve início nos anos 90, desenvolvida pela empresa Sun Microsystem, liderada pelo engenheiro James Gosling. A proposta inicial da equipe de desenvolvimento era criar uma linguagem de programação que executasse em diversos tipos de hardware, sem a necessidade de recompilar o código. A mesma deveria ser robusta, segura, e implicar baixos custos de desenvolvimento de aplicações (GREANIER, 2004).

Com o surgimento da World Wide Web¹, houve a necessidade de uma linguagem executasse em plataformas de hardware distintas. Isso levou a linguagem a ser utilizada por uma grande gama de desenvolvedores.

Os códigos desenvolvidos na linguagem de programação Java possuem a exten-

¹A World Wide Web, também conhecida como WWW, o leitor interessando sobre o assunto deve ler (BERNERS-LEE; CAILLIAU, 1994).

são .java, e quando compilados não são traduzidos diretamente para o código nativo da plataforma de hardware (ao contrário do que acontece com outras linguagens como C e C++) (GOSLING et al., 2005). O fonte em java é compilado para um código intermediário designado *bytecode*. Esse *bytecode* é interpretado pela JVM (*Java Virtual Machine*), onde a JVM é responsável por gerar o código nativo da plataforma de hardware como demonstra a Figura 5.2.

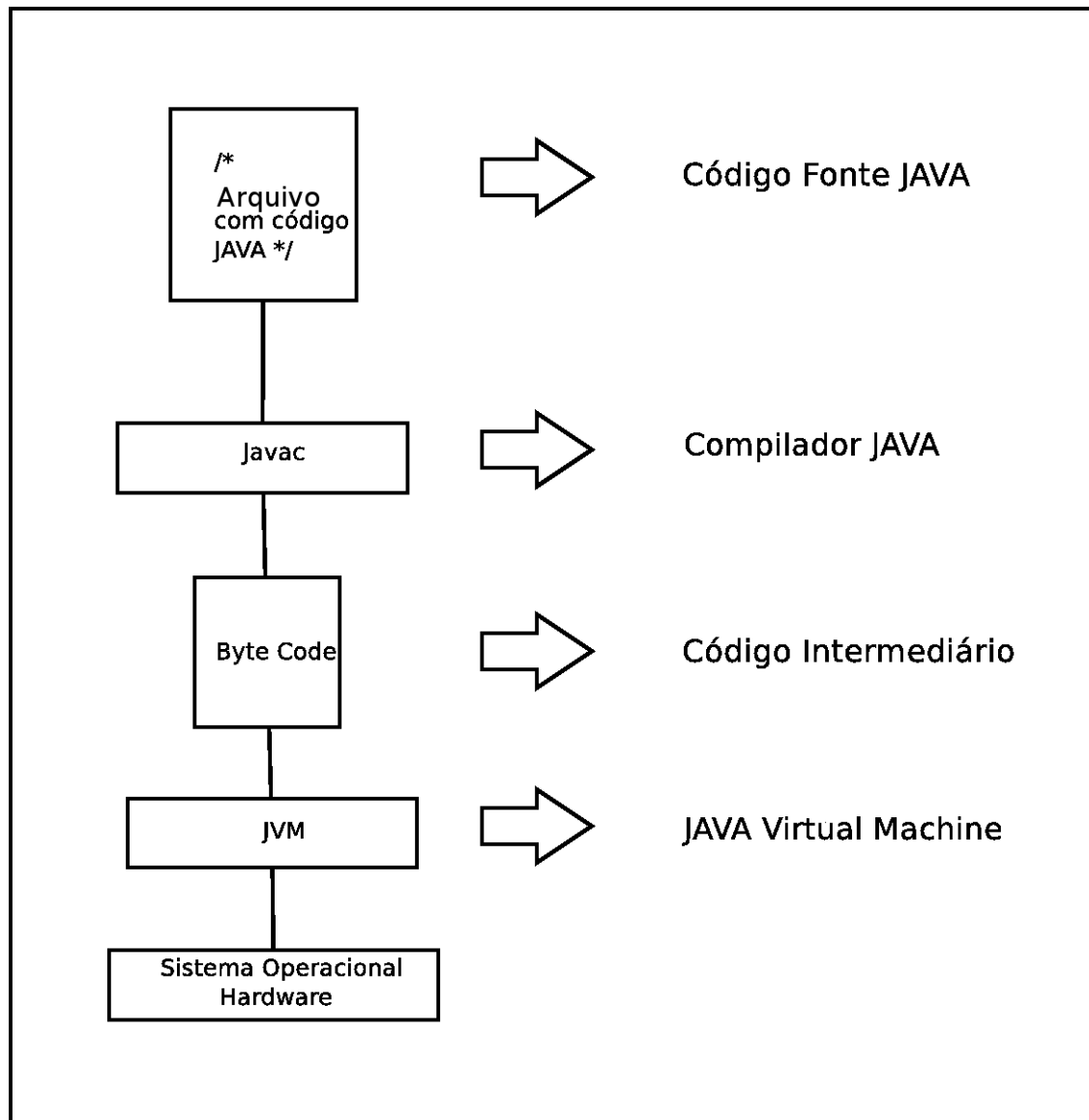


Figura 5.2: Representação da Java Virtual Machine.

A empresa Sun Microsystem, definiu um conjunto de arquiteturas para o desenvolvimento de aplicações utilizando a linguagem de programação Java:

- J2SE(*Java 2 Standard Edition*), é uma plataforma genética utilizando para de-

envolver aplicações que executam localmente.

- J2EE(*Java 2 Enterprise Edition*), é uma plataforma destinada ao desenvolvimento de aplicações WEB, no lado do servidor.
- J2ME(*Java 2 Micro Edition*), é uma plataforma para o desenvolvimento de software em dispositivos com um poder de processamento reduzido.

Neste trabalho foi utilizado a arquitetura de desenvolvimento da linguagem de programação Java J2EE, pelo fato da mesma possibilitar o desenvolvimento de aplicações voltadas para a Web.

5.2.2 Action Script 3

A linguagem de script Action Script foi desenvolvida baseada na ECMAScript². A linguagem é utilizada principalmente para o desenvolvimento de sites e softwares Web utilizando a plataforma Adobe Flash Player. Os arquivos carregados pela plataforma estão no formato SWF(*ShockWave Flash*). A linguagem veio evoluindo com o passar do tempo agregando novas funcionalidades, dentre as quais já é possível trabalhar com vídeos em tempo real, até mesmo aplicações web com finalidade comercial com acesso a base de dados (FRANKLIN; MAKAR, 2003; TEAM, 2008).

Originalmente, a linguagem foi desenvolvida pela empresa Macromedia, mas a empresa foi comprada pela Adobe em 2005, que hoje é quem desenvolve a linguagem. O Action Script foi desenvolvido inicialmente com a finalidade de controlar o vetor de animações 2D do Flash, atualmente na versão 3 a linguagem de script possibilita o desenvolvimento de ambientes virtuais e de realidade aumentada, com conjuntos de classes específicos conhecidos como Papervision e Flartoolkit (TONDEUR; WINDER, 2009).

Pelo fato de vários navegadores web como o Internet Explorer, Mozilla Firefox, Safari e outros, possibilitam a inserção da plataforma Flash Player, e depois da plataforma instalada nos navegadores, os mesmos executam a mesma aplicação sem a

²O ECMAScript é um linguagem de script utilizada em larga escala na WEB, também conhecida como JavaScript o leitor interessado em ECMAScript pode ler (ECMAScript,).

necessidade de recompilar o código em navegadores diferentes. A linguagem Action Script 3 foi escolhida para o desenvolvimento da arquitetura proposta nessa dissertação pelo fato de 99 % dos computadores atuais possuírem o Flash Player instalados em seus navegadores como demonstra a Figura 5.3.

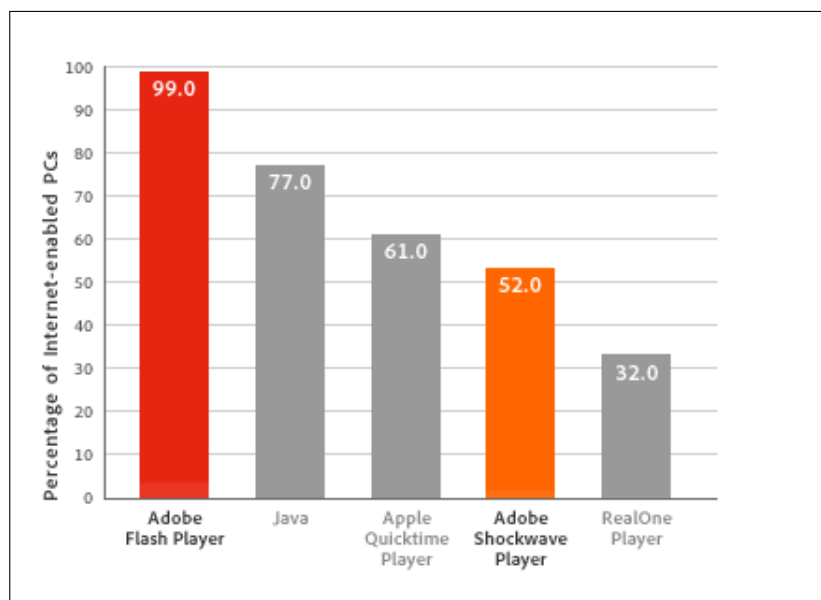


Figura 5.3: Flash Player, Quantidade de Computadores (ADOBE, 2010).

5.2.3 Glassfish

O servidor Web Glassfish™, construído pela comunidade Glassfish, é uma implementação compatível com a especificação Java J2EE.

Este servidor é de código fonte aberto, possibilita aos desenvolvedores inserirem novas funcionalidades e rotinas no mesmo. Através desse servidor, o desenvolvedor pode hospedar páginas e projetos Web desenvolvidos na linguagem de programação Java.

Por ser compatível com o BlazeDS™ esse servidor foi escolhido como tecnologia para o desenvolvimento da arquitetura proposta.

5.2.4 BlazeDS™

O BlazeDS™ é um *middleware* que se localiza entre o cliente e o servidor, ele é responsável por estabelecer a comunicação entre ambos, utilizando para isso objetos remotos. A aplicação cliente pode ser desenvolvida em linguagem de programação Flex/Flash(*Action Script 3*) ou Ajax e o servidor obrigatoriamente tem que ser desenvolvido na linguagem de programação Java (BLAZEDS, 2010).

Através das chamadas de procedimento remota, o desenvolvedor fica responsável em codificar as interfaces remotas, e o BlazeDS™ realiza a comunicação entre o cliente e o servidor utilizando seus protocolos, que serão descritos nessa seção.

5.2.4.1 BlazeDS™ Arquitetura no Cliente

Os clientes BlazeDS™ utilizam um conjunto de mensagens pré estabelecidas pela arquitetura como demonstra Figura 5.4, para viabilizar a comunicação do cliente com o servidor, utilizando para estabelecer essa comunicação entre ambos os protocolos HTTP³ e AMF⁴ (BLAZEDS, 2010).

- RemoteObject component: possibilita a definição de interfaces remotas para que possa ocorrer a comunicação entre o cliente e o servidor.
- HTTPService/WebService components: possibilita a comunicação com o servidor através do protocolo HTTP ou AMF.
- Producer/Consumer components: no modo produtor o cliente envia mensagens para o servidor, e no modo consumidor o cliente aguarda que o servidor o envie uma mensagem.

Os clientes Flex utilizam um canal de comunicação para estabelecer uma seção com o servidor BlazeDS™. Os canais tem como sua principal característica fornecer conectividade entre o cliente e o servidor.

³O protocolo *HyperText Transfer Protocol*, define as maneiras pelas quais os navegadores e outros tipos de clientes interagem com os servidores Web (COULOURIS, 2005; FIELDING et al., 1999).

⁴O protocolo *Action Message Format*, é um formato compacto binário utilizado para serializar objetos em ActionScript, o leitor interessado em AMF pode ler (AMF... , 2010).

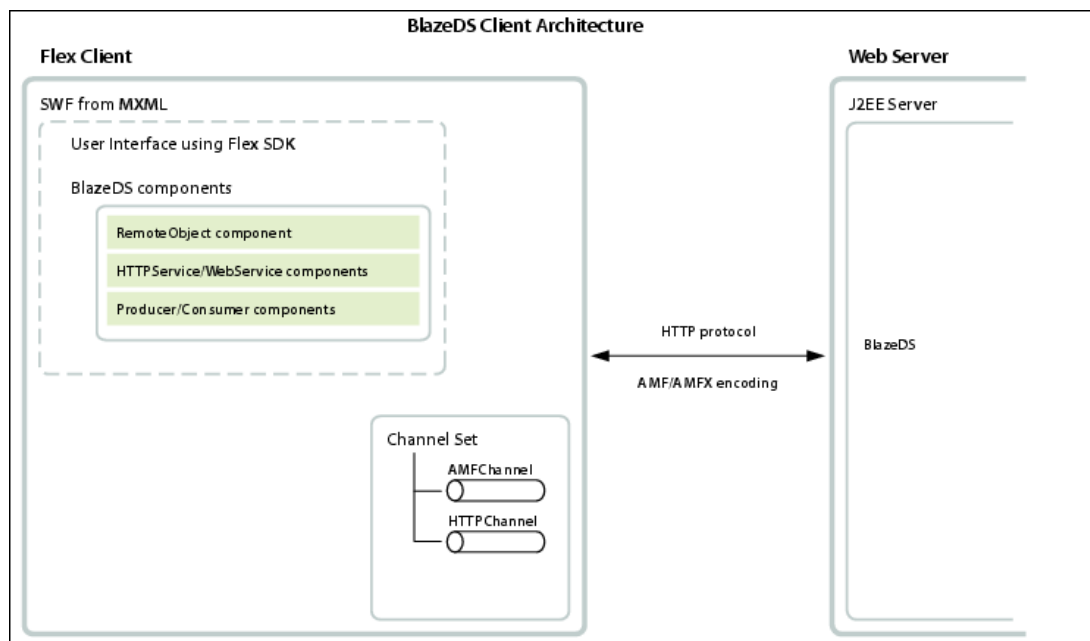


Figura 5.4: Arquitetura do Cliente BlazeDS™ (BLAZEDS, 2010).

Os clientes utilizam diferentes tipos de canais como o AMFChannel e HttpChannel. A seleção de canais depende de uma série de fatores, se estiver utilizando transferência de dados binário é utilizado o protocolo AMFChannel, caso a comunicação for utilizando mensagens não binárias é utilizando o protocolo HttpChannel.

5.2.4.2 BlazeDS™ Arquitetura no Servidor

O servidor BlazeDS™ está contido em uma aplicação Web J2EE. Em sua arquitetura um cliente solicita um serviço através de uma canal de comunicação para o servidor BlazeDS™. O pedido é processado e a resposta é enviada para o cliente que solicitou o serviço. A Figura 5.5 demonstra a arquitetura do servidor BlazeDS™ (BLAZEDS, 2010).

O MessageBroker é responsável pelo roteamento das mensagens para os serviços do BlazeDS. O servidor recebe uma mensagem de algum cliente e a processa, logo após a mesma é enviada para o MessageBroker onde essa camada verifica qual o destino da mensagem e a encaminha para o serviço apropriado dentro de sua arquitetura.

O servidor BlazeDS™ vem com um rico conjunto de adaptadores, que possibilita

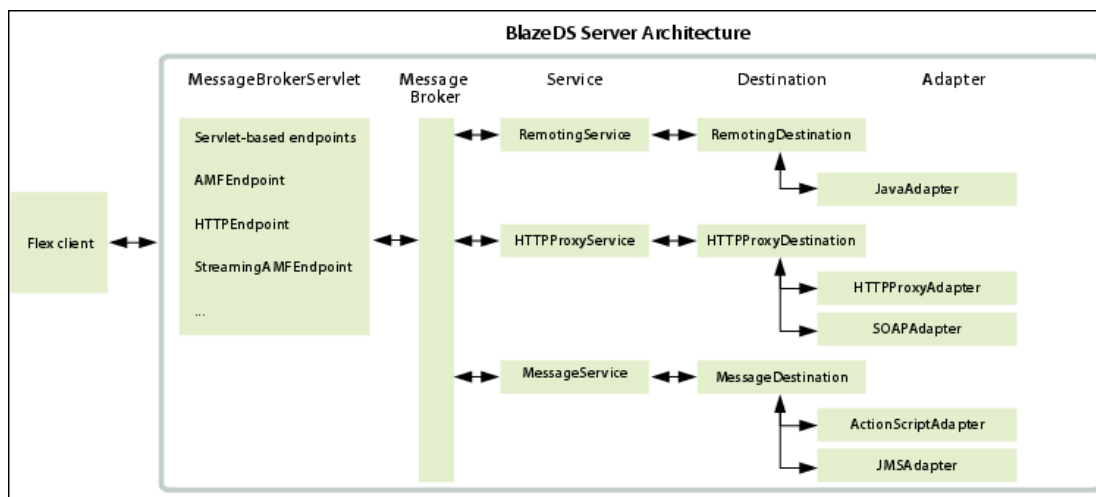


Figura 5.5: Arquitetura do Servidor BlazeDS (BLAZEDS, 2010).

ao servidor comunicar com diversas aplicações hospedadas no mesmo servidor Web (BLAZEDS, 2010).

5.3 Detalhes de Implementação

Para o desenvolvimento do *software* (RANET) que convalida a arquitetura proposta nesta dissertação. Foram utilizadas as seguintes tecnologias:

O *Flartoolkit*: responsável pela interface de Realidade Aumentada, e quando ocorre uma interação através de funções as mensagens são enviadas para a tecnologia BlazeDS.

O BlazeDs: *middleware* responsável pela comunicação entre a camada de Realidade Aumentada e o Servidor Web, através de uma chamada de procedimento remota.

O Servidor Web *Glassfish*: responsável por receber as solicitações através do *middleware* BlazeDs, processar essas solicitações e enviar à todos os clientes conectados no sistema.

Como pode ser observado através da Figura 5.6, essas tecnologias são separadas logicamente da seguinte forma:

Apesar do modelo lógico demonstrado pela Figura 5.6, todas essas camadas são

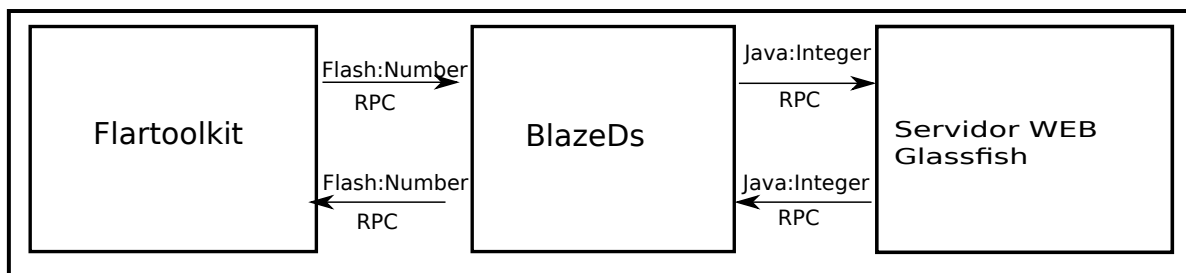


Figura 5.6: Estrutura Lógica das Tecnologias Utilizadas.

hospedadas em um servidor Web. A grande vantagem em colocar toda a aplicação em um servidor Web é que livra o usuário de ter que instalar e configurar qualquer serviço em sua máquina, já que a aplicação está toda hospedada na Web.

5.3.1 Definição das Regras para Animação dos Modelos

Uma estratégia foi adotada para que qualquer objeto virtual gerado nos modeladores, blender, 3DS MAX e Maya, fossem adicionado no *software* desenvolvido. Foi necessário definir algumas regras na criação das animações, obedecendo o seguinte tempo definido no time line dos modeladores, como demonstra a Tabela 5.1. Após a animação ser criada obedecendo as regras apresentada na Tabela 5.1. O objeto virtual pode ser exportado para o formato *Collada*⁵ (*dae*) e adicionado no *software* (RANET).

Tabela 5.1: Regras Definidas no Time Line de Animação.

Time Line	Descrição
80	Faixa destinada a primeira animação
160	Faixa destinada a segunda animação
240	Faixa destinada a terceira animação
320	Faixa destinada a quarta animação
400	Faixa destinada a quinta animação

Como pode ser observado na Tabela 5.1, foi definida uma faixa de tempo que deve ser obedecida dentro da time line quando o objeto virtual está sendo animado

⁵COLLADA (*COLLABorative Design Activity*) é um padrão de exportação e importação de arquivos criado pela Sony e usado como padrão para o console Playstation. É suportado pelos melhores editores 3D, como Blender, 3D Studio Max e Maya. O formato utiliza XML como padrão e suporta além de modelos detalhados, animações e iluminação. O leitor interessado deve ler (COLLADA, 2010)

nos modeladores 3Ds. Desta forma, a animação é executada e controlada dentro do *software* desenvolvido.

5.3.2 Protocolo de Comunicação

A comunicação entre os módulos cliente e servidor é feita através do mecanismo de troca de mensagens.

Para que essas mensagens sejam compreendidas pelos módulos, houve a necessidade de definir um protocolo. O protocolo de animação define qual mensagem deve ser enviada na rede.

A Tabela 5.2 demonstra quais mensagens podem ser transmitidas para o servidor. Quando a mensagem chega até o servidor ele é responsável por processá-la e encaminhá-la para os demais usuários conectados no sistema.

Tabela 5.2: Mensagens Trafegadas pela Rede para Animações

Mensagem na Rede	Ação no Modelo 3D
1	Executa a primeira animação
2	Executa a segunda animação
3	Executa a terceira animação
4	Executa a quarta animação
5	Executa a quinta animação

Quando os usuários receberem a mensagem através de uma chamada de procedimento remota, os mesmos executam a animação referida de acordo com a Tabela 5.2. Apesar de teoricamente simples o protocolo se mostrou de extrema eficiência, pois não sobrecarrega a rede com dados desnecessários, e devido essa ser uma aplicação voltada para web quanto menor o tráfego de dados na rede, melhor será a comunicação entre os módulos.

O Código 5.1 demonstra a implementação do protocolo de animação.

Código 5.1: Método para Manipular Animação

```
1 private void protocolo(int msg) {  
2  
3     switch(msg) {
```



```
4         case 1:
5             anim1 ();
6         break;
7         case 2:
8             anim2 ();
9         break;
10        case 3:
11            anim3 ();
12        break;
13        case 4:
14            anim4 ();
15        break;
16        case 5:
17            anim5 ();
18        break;
19    }
20
21 }
```

Outro protocolo foi desenvolvido para possibilitar a troca do objeto virtual que está atuando na cena. Este protocolo envia através da *internet* uma mensagem contendo o código do objeto virtual, que está presente na biblioteca de objetos virtuais.

5.3.3 Biblioteca de Objetos Virtuais

Foi implementado no *software* RANET uma biblioteca de objetos virtuais. Essa biblioteca possibilita que os usuários adicionem modelos virtuais, e distribuam esses modelos para os demais usuários conectados no sistema. O que possibilita o *software* ser utilizado em várias áreas do conhecimento.

É importante ressaltar que para adicionar um novo modelo virtual na biblioteca de objetos virtuais, este modelo virtual deve ser desenvolvido obedecendo as regras apresentadas na seção 5.3.1.

5.4 Estudo de Caso

O presente estudo de caso aborda a apresentação de componentes de um **motor de corrente contínua**, e a possibilidade de acionamento do mesmo, sem a representação de campos magnéticos envolvidos. O protótipo está focado nas questões tecnológicas, ou seja, uma das contribuições está na construção de uma aplicação de Realidade Aumentada distribuída pela Web com forte apelo pedagógico. O estudo de caso utiliza Realidade Aumentada para o ensino teórico de motores de corrente contínua, usando distribuição, que possibilita que vários usuários interajam com a aplicação em tempo real mesmo não estando presente no mesmo espaço geográfico. Esta possibilidade permite ao docente utilizar um artefato tecnológico para auxiliar no processo de ensino/aprendizagem dos discentes. Desta forma, o estudo de caso contribui para ilustrar a possibilidade de utilização da aplicação desenvolvida no processo de ensino/aprendizagem e, também, possibilitando o entendimento geral do sistema sob o ponto de vista educativo.

5.4.1 Educação

A Educação está sofrendo uma transformação com o acesso à tecnologias que estão auxiliando o professor, na missão de ensinar (SANTOS; ANDRADE, 2009).

A maioria das pessoas que vivem no mundo tecnologicamente desenvolvido tem um acesso sem precedentes à informação. Isto não significa que disponha de habilidades e do saber necessários para convertê-los em conhecimento, esse é o papel da escola (SANTOS; ANDRADE, 2009).

Um grande desafio para o uso intensivo de tecnologias de informação e comunicação em educação está diretamente relacionada com a implantação de uma infraestrutura adequada em escolas e outras instituições de ensino.

Com o apoio do governo federal são poucos os centros educacionais que não possuem recursos computacionais (PROINFO, 2010). Segundo (NOGUEIRA, 2010) as tecnologias computacionais aplicáveis à Educação são normalmente enquadradas em quatro categorias:

- 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 nova categoria que se pode destacar é o uso de Realidade Virtual e Aumentada (NOGUEIRA, 2010). Para o desenvolvimento do protótipo e estudo de caso proposto nesta dissertação, observa-se que a mesma é desenvolvida em uma faculdade de engenharia elétrica, desta forma o estudo de caso proposto é aplicado nessa área, porém poderia ser desenvolvido em diversas áreas do conhecimento.

5.4.2 Motor de Corrente Contínua

Um motor de corrente contínua converte a energia elétrica em energia mecânica, o que o diferencia dos demais motores é que ele deve ser alimentado com corrente de tensão contínua. Essa tensão contínua pode ser adquirida através de pilhas e baterias, no caso de motores pequenos, ou de uma rede alternada após retificação, no caso de motores maiores. Os principais componentes de um motor de corrente contínua são o estator e rotor, como demonstra a Figura 5.7.

Estador, contém um enrolamento, que é alimentado diretamente por uma fonte de tensão contínua, no caso de pequenos motores, o estator pode ser um ímã permanente.

O rotor é um eletroímã constituído de um núcleo de ferro com enrolamentos em sua superfície que são alimentados por um sistema mecânico de comutação Figura

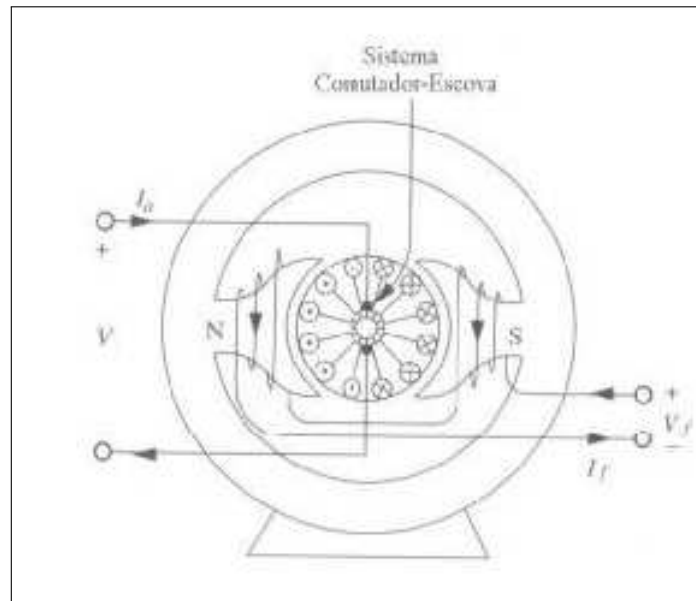


Figura 5.7: Esboço do Motor Corrente Contínua (TÉCNICA, 2006).

5.8. Esse sistema é formado por um comutador, solidário ao eixo do rotor, que possui uma superfície cilíndrica com diversas lâminas às quais são conectados os enrolamentos do rotor; e por escovas fixas, que exercem pressão sobre o comutador e que são ligadas aos terminais de alimentação. O propósito do comutador é o de inverter a corrente na fase de rotação apropriada de forma a que o conjugado desenvolvido seja sempre na mesma direção.



Figura 5.8: Sistema de Comutação (TÉCNICA, 2006).

5.4.3 Modelagem do Motor

O objeto virtual visualizado no sistema de Realidade Aumentada, é baseado no conceito de motores de corrente contínua. Desta forma, com o objeto virtual baseado em uma área do conhecimento específica, pode ilustrar o potencial que este protótipo possui para aplicação em áreas educacionais.

Para o desenvolvimento do modelo a ser utilizado pelo sistema de Realidade Aumentada para Web, no estudo de caso proposto, foi utilizado o software Blender. Blender é um software de modelagem e animação 3D de código aberto, está sobre licença GNU-GPL (*General Public License*), que permite a qualquer pessoa ter acesso ao código-fonte do programa para que possa fazer melhorias, contanto que as disponibilize à comunidade (BRITO, 2008).

Utilizando o software Blender, foi modelado um motor de corrente contínua e adicionado a esse modelo, algumas animações. Essas animações possibilitam simular o funcionamento do motor.

A Figura 5.9 demonstra o motor de corrente contínua, modelado na ferramenta Blender.

5.5 Funcionamento do Sistema

5.5.1 Funcionamento da Camada de Comunicação

Para que o sistema desenvolvido funcione corretamente, é necessários que os clientes tenham em suas máquinas um navegador web com o flash player configurado. Caso o cliente não tenha o flash player instado em seu navegador, o mesmo pode ser adquirido no site da Adobe⁶.

Uma das características cruciais da camada de comunicação, através do *framework* blazeDS, é que não há a necessidade de iniciar e instalar nenhum serviço na máquina do cliente. Quando o cliente acessa a url onde a aplicação está hospedada, um código

⁶O plugin do flash pode ser adquirido na url: <http://get.adobe.com/br/flashplayer/>.

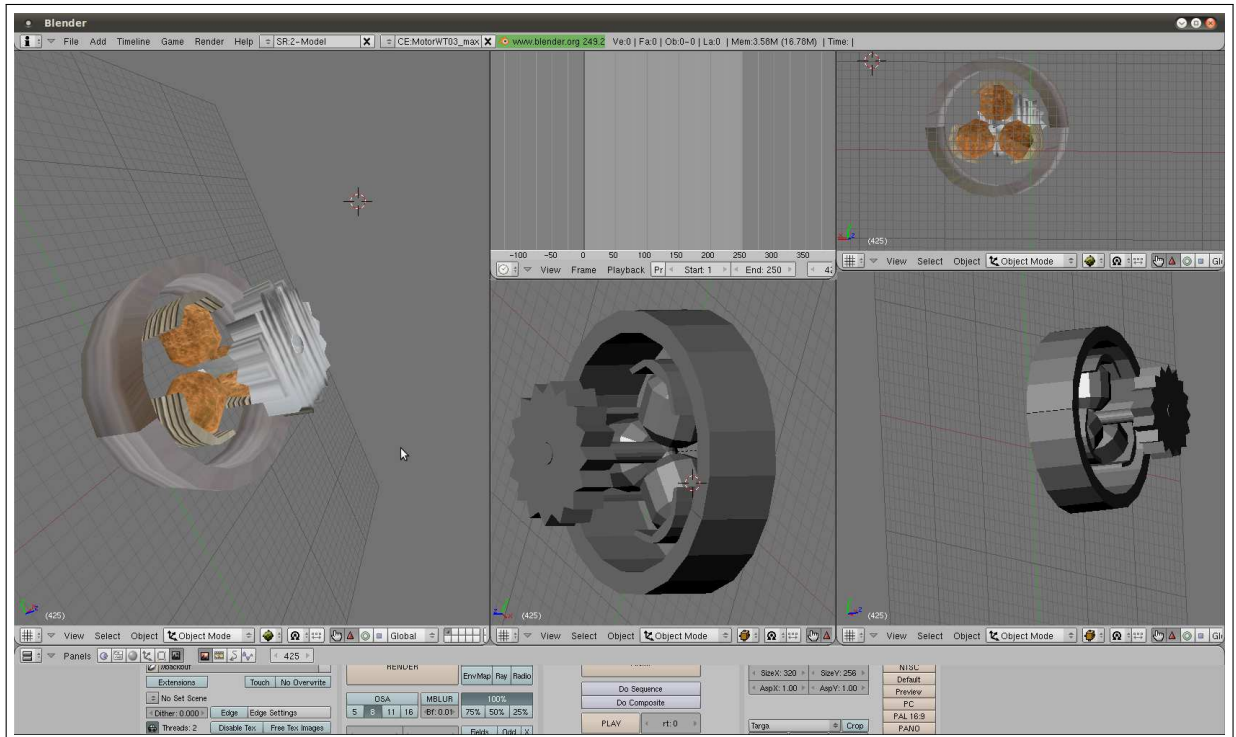


Figura 5.9: Motor de Corrente Contínua Modelado no Blender.

objeto do blazeDs é enviado para o cliente de forma transparente.

Na aplicação servidor não há interface gráfica, pois o mesmo tem a função de gerenciar, distribuir e enviar as informações e objetos virtuais solicitados pelos clientes, para todos os usuários conectados.

Já à aplicação cliente hospedada em um Web Service, recebe as informações enviadas pelo servidor e as armazena na camada de gestão para que a camada do misturador de ambientes leia essas informações e adicione os objetos virtuais na cena, e se for o caso executa a animação solicitada. A aplicação cliente também verifica quais as informações foram armazenadas pela GUI, na camada de gestão e as encaminha para a camada de comunicação.

O *software* **Realidade Aumentada Distribuída pela Web (RANET)** é composto por quatro camadas: GUI, Misturador de Ambientes, gestão e uma camada de comunicação.

A GUI é responsável por enviar um código objeto para o cliente contendo toda a aplicação, para que a mesma seja executada no computador do usuário através de

um *browser*.

O Misturador de Ambiente é responsável pela visualização dos objetos virtuais na cena por meio do *Flartoolkit*. Já a camada de gestão faz a conexão entre o misturador de ambiente e a camada de comunicação, desta forma, a camada de comunicação funciona como uma arquitetura Cliente/Servidor que faz a distribuição dos objetos virtuais.

Entretanto, a camada de gestão possibilita a execução de animações utilizando o protocolo definido na seção 5.3.2, que é interpretado pela camada de comunicação e repassado para o misturador de ambientes.

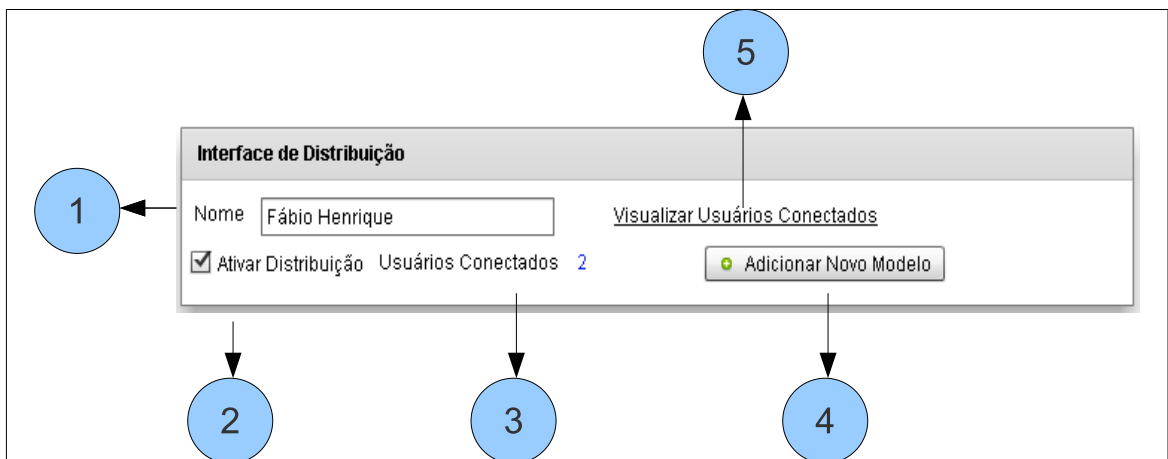


Figura 5.10: Interface de Distribuição do Software.

Como pode ser observado na Figura 5.10, pode-se notar as seguintes interfaces:

1. Usuário entra com o nome, que posteriormente será visualizado pelos demais usuários.
2. Demonstra para o usuário, através da caixa de texto Ativar Distribuição, se a aplicação está enviando as alterações que ocorrem na GUI, para os demais usuários conectados na rede.
3. Mostra, para o usuário, quantos participantes estão conectados no sistema.
4. Através desse botão, o usuário pode adicionar um novo objeto virtual na biblioteca compartilhada. Esse objeto pode ser visualizado por todos os usuários conectados no sistema.

5. Abre um *pop-up*, contendo uma lista com os usuários conectados no sistema naquele momento como demonstra a Figura 5.11.



Figura 5.11: *Pop-up*, Lista dos Usuários Conectados no Sistema.

5.5.2 A GUI

Para o reconhecimento de que local o objeto virtual deve ser projetado na cena, foi utilizado o marcador desenvolvido pelo grupo de pesquisa em Realidade Virtual e Aumentada da Universidade Federal de Uberlândia, como demonstra a Figura 5.12.

A aplicação de Realidade Aumentada para Web (RANET) possui três menus, como observados na Figura 5.13. Cada menu nessa aplicação é responsável por realizar uma ação distinta em cada objeto virtual que esteja atuando na cena. Esta interface também possui outras funcionalidades descritas abaixo:

1. Primeiro Menu: possibilita ao usuário realizar algumas operações em cima do objeto virtual, presente na cena. No caso do estudo de caso proposto nesta dissertação, possibilita através do sub menu Ligar: simular o funcionamento do motor com uma tensão de 3 volts Figura 5.14(a), 6 volts Figura 5.14(b) ou volts 9 Figura 5.14(c). Já através do sub menu Motor: é possível visualizar o motor

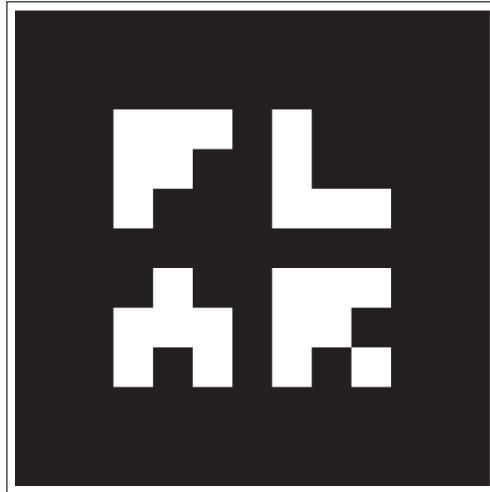


Figura 5.12: Marcador da Aplicação.

fechado com a sua execução pausada Figura 5.15(a), visualizar seus componentes Figura 5.15(b), também é possível visualizar os componentes do motor aberto em funcionamento Figura 5.15(c).

2. Segundo Menu: possibilita ao usuário adicionar novos objetos virtuais na biblioteca através do sub menu Biblioteca, ou atualizar a lista de objetos virtuais existentes na biblioteca de objetos virtuais.
3. Terceiro Menu: possibilita aplicar transformadas geométricas nos objetos virtuais visualizados na cena. As operações geométricas são: escala, rotação (X,Y,Z) e translação (X,Y,Z).
4. A Biblioteca de objetos virtuais: possibilita que o usuário altere o objeto virtual que está atuando na cena. Selecionando o nome do objeto virtual que deseja inserir na cena, desta forma esse objeto vai aparecer no lugar do objeto que estava atuando anteriormente.
5. Botão atualizar: utilizado para recarregar a lista de objetos virtuais na biblioteca.
6. Botão imprimir marcador: possibilita ao usuário imprimir o marcador para utilizar o sistema.
7. A descrição do objeto virtual: demonstra para o usuário um pequeno texto informativo do objeto virtual que esteja atuando na cena.

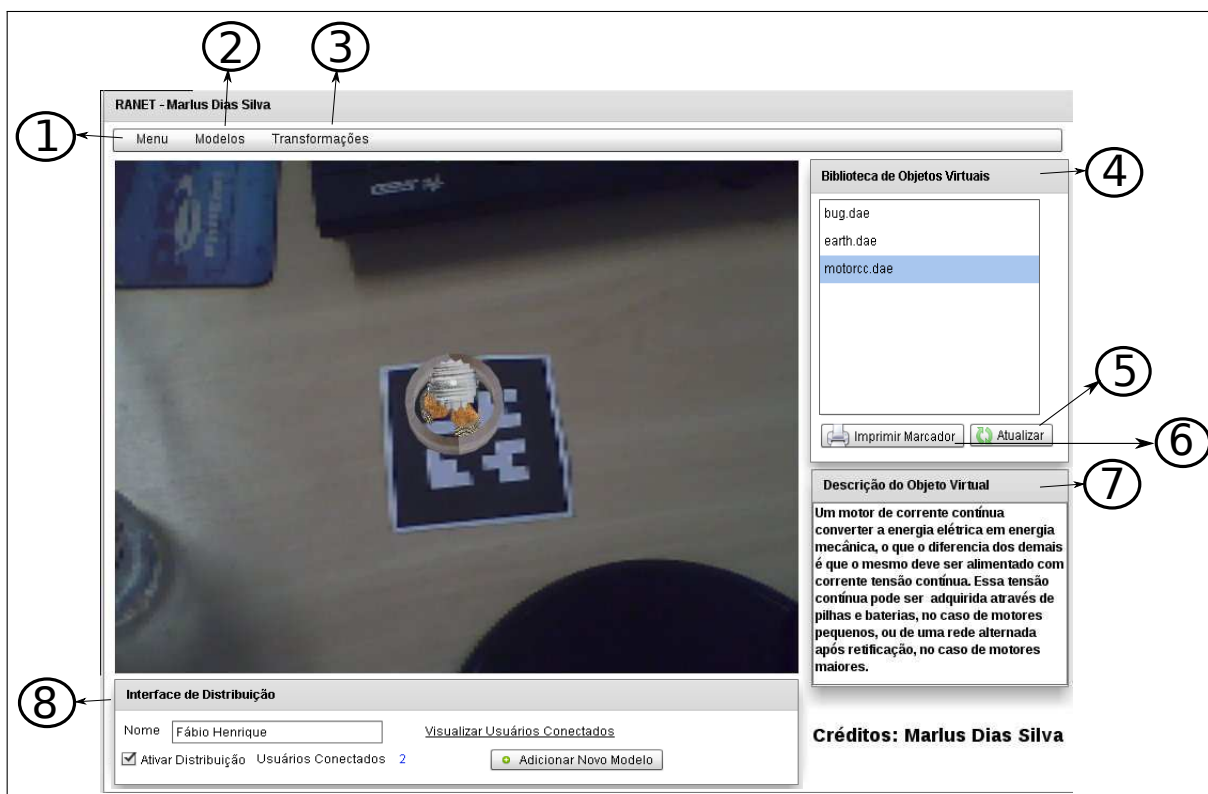
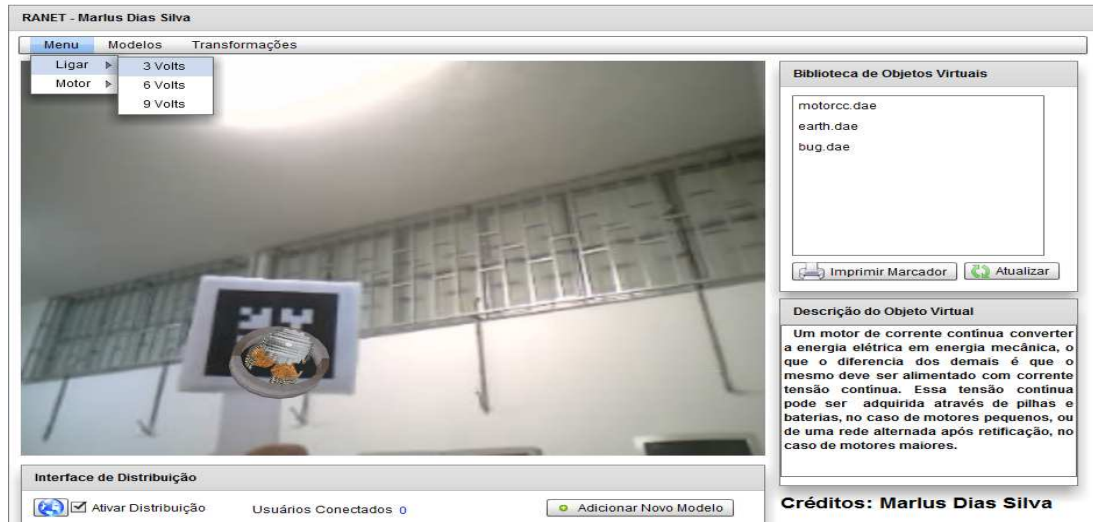


Figura 5.13: Interface do Sistema de Realidade Aumentada.

8. Interface de Comunicação: já comentado na seção 5.5.1.

Utilizando o marcador, pode-se instanciar qualquer objeto que esteja na biblioteca. Desta forma quando um usuário seleciona um objeto virtual, é disparado uma mensagem para a camada de gestão, onde essa grava qual foi o objeto virtual está associado ao marcador, e repassa para a camada de comunicação, que envia a mensagem para o servidor. O servidor por sua vez, faz a distribuição da mensagem para todos os clientes conectados no sistema, esses clientes fazem a leitura ao receber as informações enviadas pelo servidor e grava na camada de gestão. Posteriormente, a camada do misturador de ambientes faz a leitura dos dados da camada de gestão e renderiza o objetivo virtual na cena.

Como ilustrado na Figura 5.16, foi adicionado um motor de corrente contínua em cima do marcador cadastro no sistema. Desta forma, quando a webcam do usuário visualiza o marcador. Um motor virtual Figura 5.17 com as características preservadas é projetado sobre o motor real, possibilitando ao usuário visualizar os componentes internos do motor.



(a) Selecionando Animação 3 Volts.



(b) Selecionando Animação 6 Volts.



(c) Selecionando Animação 9 Volts.

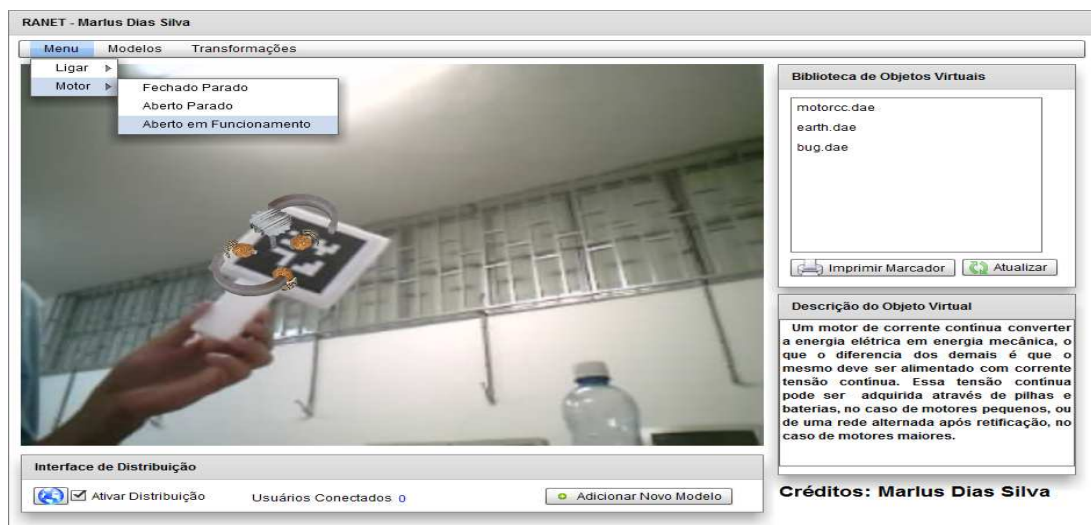
Figura 5.14: Animações Cadastradas no Sistema.



(a) Selecionando Animação Motor Fechado Pause



(b) Selecionando Animação Motor Aberto Pause



(c) Selecionando Animação Motor Aberto Funcionando

Figura 5.15: Demais Animações Cadastradas.



Figura 5.16: Motor Posicionado Sobre Marcador.

Quando um usuário conectado no sistema chamado de (**user1**) executa uma animação através dos menus interativos na GUI Figura 5.18(a) uma mensagem é disparada para a camada de gestão, essa camada irá repassar as informações para a camada de comunicação que é responsável por encaminha a mensagem para o servidor web. O servidor por sua vez enviará a mensagem para todos os clientes conectados no sistema. A Figura 5.18(b) demonstra outro usuário conectado ao sistema chamado (**user2**), onde a camada de comunicação de sua aplicação recebe a mensagem enviado pelo servidor Web, por meio da camada comunicação, essa camada encaminha a mensagem para a camada de distribuição. A GUI processa a mensagem contida na camada de gestão e exibe a animação solicitada pelo o usuário (**user1**), a todos os participantes conectados no sistema.

A Figura 5.18(c) demonstra um usuário (**user3**) que não possui um motor real de corrente contínua. Neste a caso o motor é projeto em cima do marcador, e as funcionalidades do sistema continuam inalteradas.

O *software* desenvolvido possibilita que qualquer usuário, adicione outro modelo virtual e execute as animações contidas neste modelo. Desde que obedeça as regras definidas na seção 5.3.1.

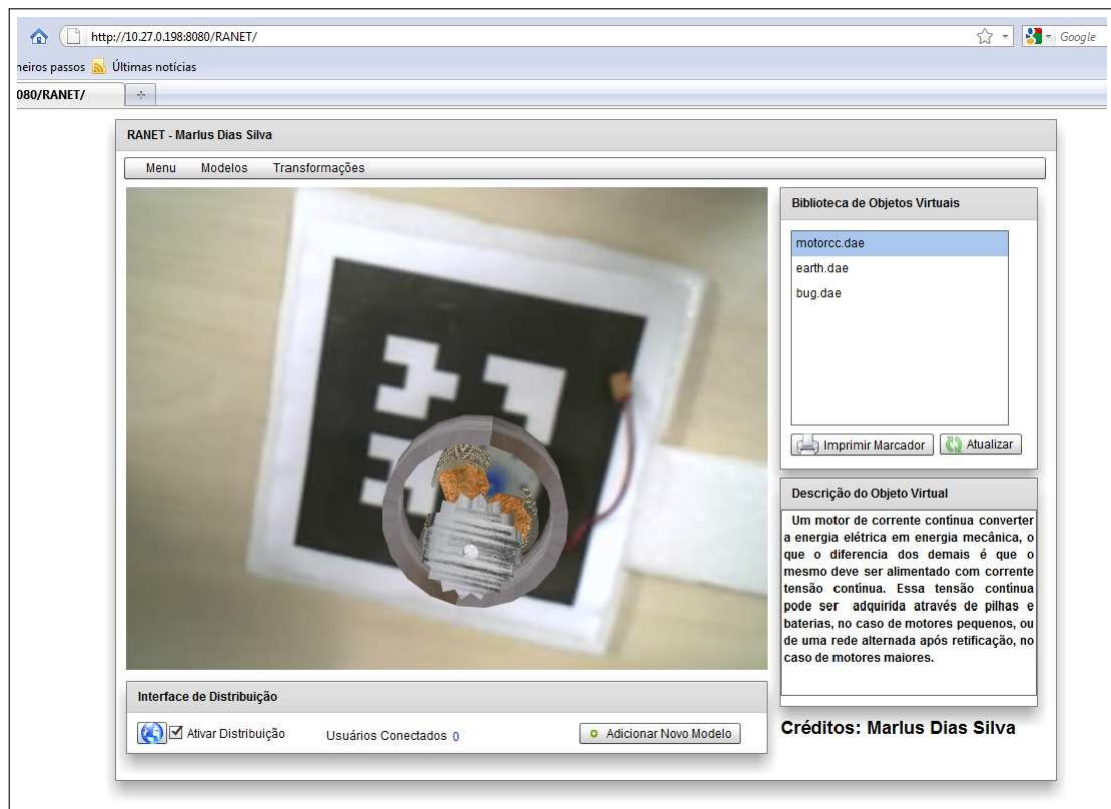


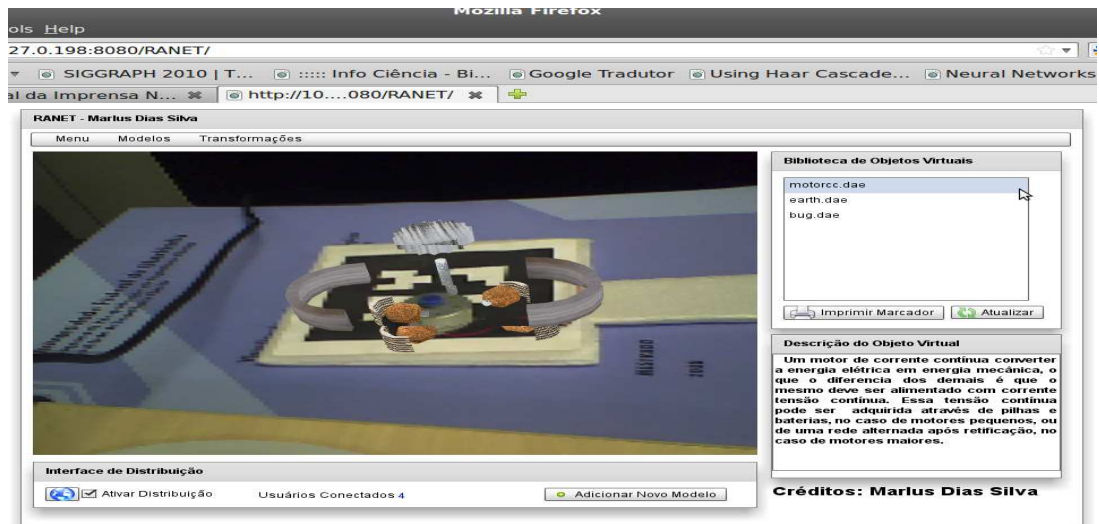
Figura 5.17: Motor Virtual Projetado Sobre Motor Real.

5.6 Considerações Finais

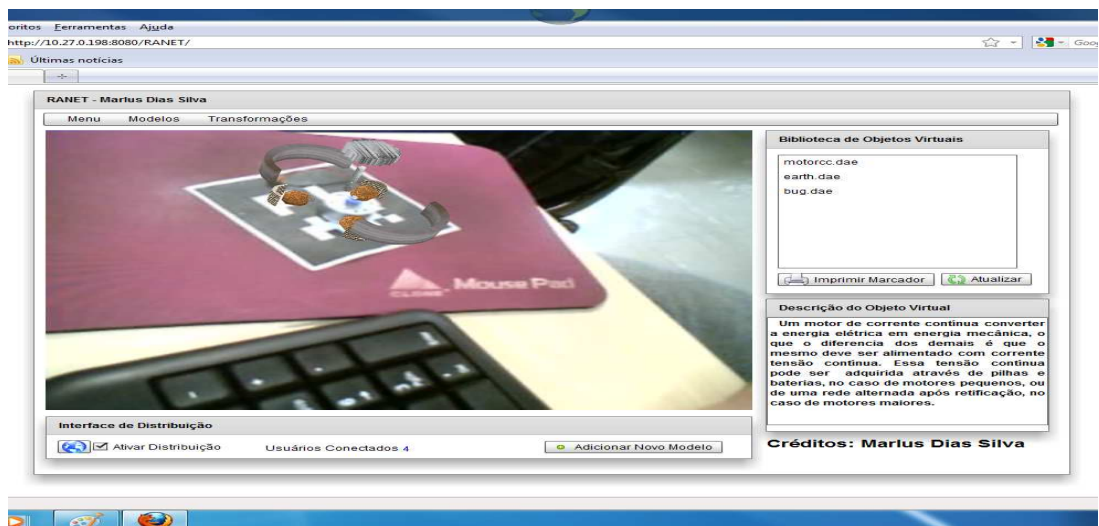
Este capítulo apresentou os detalhes de implementação juntamente com o estudo de caso do *software Realidade Aumentada para Web*, detalhando o seu funcionamento, no contexto do estudo de caso do motor de corrente contínua. Abordando também a forma com que as mensagens são distribuídas na rede.

É importante ressaltar que o ambiente de *Realidade Aumentada para Web*, pode suportar inúmeros clientes conectados na web, isso é possível pela forma que a arquitetura foi projetada.

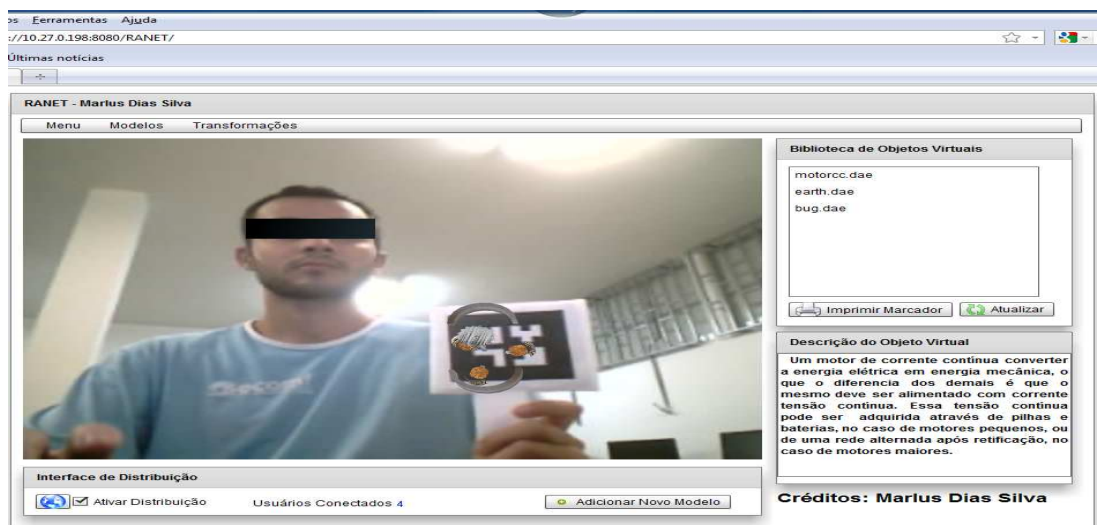
Baseado no estudo comparativo realizado nos trabalhos relacionados, percebe-se a contribuição deste trabalho, através da Tabela 5.3.



(a) User 1 Utilizando o Software



(b) User 2 Utilizando o Software.



(c) User 3 Utilizando o Software.

Figura 5.18: Usuários Utilizando o Sistema.

Tabela 5.3: Comparações e Contribuições desta Dissertação.

Trabalhos Relacionados	Distribuição RA	Distribuição LAN	Distribuição Internet	Execução na Web	Multiplataforma	Distribuir Novos Modelos
Ambientes Colaborativos com Realidade Aumentada	✓	✓	✓	✗	✗	✗
Aplicações Educacionais em Ambientes Colaborativos com Realidade Aumentada	✓	✓	✗	✗	✗	✗
Uma Arquitetura para Distribuição de Ambientes Virtuais de Realidade Aumentada	✓	✓	✓	✗	✗	✗
Uma Arquitetura para a Distribuição de Realidade Virtual e Aumentada	✓	✓	✓	✗	✗	✓
Sistema de Autoria em Ambiente Colaborativo com Realidade Aumentada	✓	✓	✓	✗	✗	✓
Virtual Reality Tools for Internet-Based Robotic Teleoperation	✓	✓	✓	✗	✗	✗
Realidade Aumentada Distribuída na Web com Independência de Servidor para Manipulação de Objetos Virtuais na Camada Cliente	✓	✓	✓	✓	✓	✓

Capítulo 6

Avaliação e Resultados

6.1 Introdução

Neste capítulo, é apresentada uma análise do protótipo descrito no capítulo 5. Esta análise teve como objetivo demonstrar a eficiência do protótipo. Através de testes de escalabilidade e usabilidade. Para o teste de usabilidade foi elaborado um questionário para a avaliação do sistema.

Para realizar a análise de usabilidade do *software*, foi necessário necessário a construção de uma metodologia específica baseado na ISONORM para fazer á avaliação.

6.2 Avaliação do Sistema RANET

Para a verificação e validação do *software* RANET foi elaborado, um questionário baseado nos preceitos da ISONORM 9241/10 que trata da usabilidade de *software* (PRÜMPER, 1999). Este questionário foi elaborado no formato de um *checklist* onde os usuários puderam informar suas opiniões.

O *software* RANET foi executado nos sistemas operacionais Ubuntu e Windows, e foram utilizados os navegadores Internet Explorer (Windows), Mozilla Firefox (Windows e Ubuntu), Google Chrome (Windows e Ubuntu), Opera (Windows e Ubuntu) e

Safari (Windows). O software executou normalmente nos navegadores citados.

Também foram realizados testes de latência e escalabilidade do software desenvolvido.

6.3 Latência

Segundo (COULOURIS, 2005) latência é o tempo decorrido após uma operação de envio ser executada e antes que os dados comecem a chegar em seu destino. Ela pode ser medida como o tempo necessário para transferir uma mensagem. No caso do experimento foi utilizados pacotes de 4 *bytes*.

Segundo (RIBEIRO, 2006) se o ambiente distribuído existe para emular o mundo real, deve operar em termos da percepção humana. 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. A latência será o suporte para analisar a escalabilidade (capacidade de aumento na quantidade de usuários). A análise foi realizada em três etapas:

Na primeira etapa: foi feito um teste para medir o tempo que o pacote enviado pelo cliente demora para chegar até o servidor.

Na segunda etapa: usou-se servidor e 3 clientes. Neste caso mede o tempo de envio do pacote do cliente para o servidor e do servidor para o outro cliente.

Na terceira etapa: usou-se servidor e 6 clientes. Neste caso mede o tempo de envio do pacote do cliente para o servidor e do servidor para o outro cliente.

A Figura 6.1 demonstra o resultado obtido. Como pode ser observado o protótipo se comportou satisfatoriamente com a quantidade de usuários a que foi submetido.

Para o teste de escalabilidade, será utilizado os valores obtidos no teste de latência.

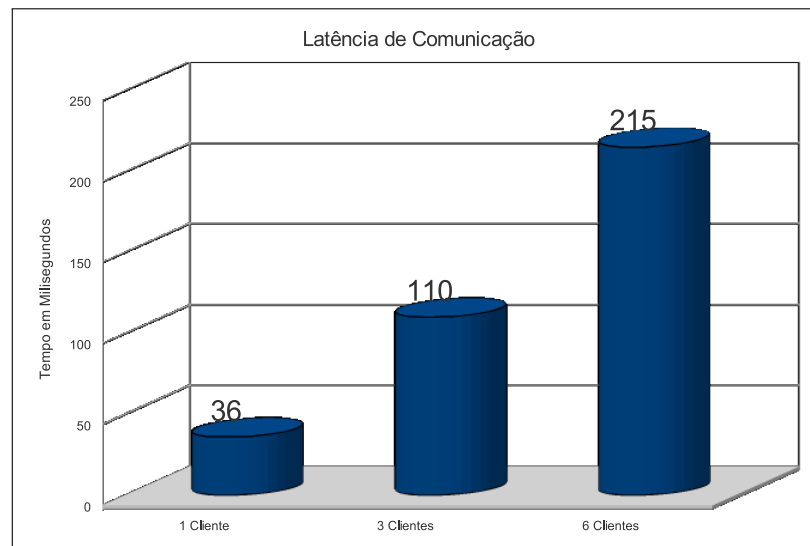


Figura 6.1: Latência de Comunicação.

6.3.1 Escalabilidade

A escalabilidade foi interpretada como sendo a capacidade do sistema em aceitar novos clientes, numa mesma máquina ou em máquinas diferentes (RIBEIRO, 2006). Segundo (RIBEIRO, 2006) sistemas que envolvem operadores humanos devem entregar pacotes com a mínima latência e gerar gráficos texturizados 3D a uma frequência de 30 a 60 Hz, para garantir a ilusão de realidade. Este desafio torna-se maior em sistemas que utilizam redes a longa distância, conforme mencionado anteriormente. A latência da rede pode ser reduzida, até certo ponto, por meio da utilização de enlaces (links) dedicados. Todavia, uma maior largura de banda não é necessariamente a melhor solução.

A escalabilidade foi medida em função do resultado da análise de latência na seção 6.3, 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 seis clientes. A escalabilidade foi testada, aumentando esta quantidade para 12, 18, 24 e 30 clientes, respectivamente. O Resultado obtido pode ser observado através da Figura 6.2.

Como pode ser observado, a escalabilidade do sistema com 30 usuários ficou na casa dos milissegundos o que não afetou o desempenho do *software*.

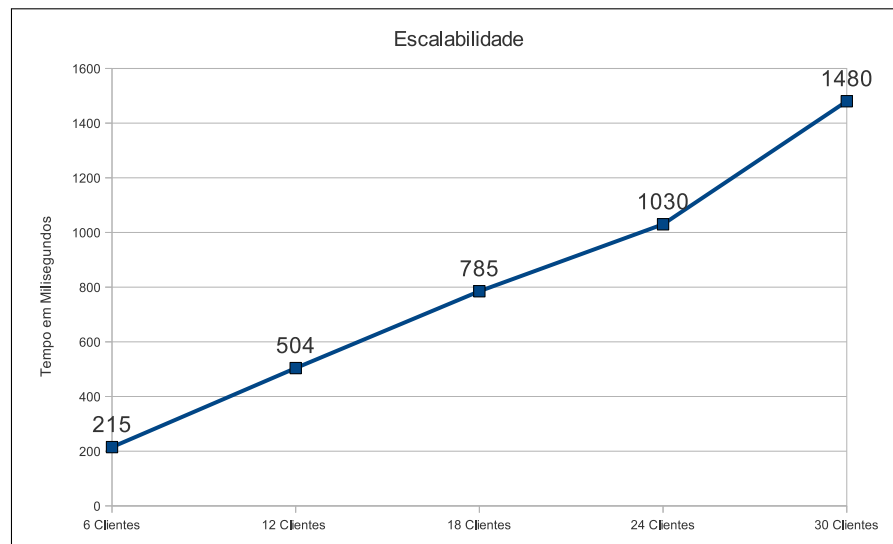


Figura 6.2: Escalabilidade.

6.4 Testes de Usabilidade

O sistema foi apresentado a 35 alunos do ensino médio (tecnológico) e a 5 professores ambos da área de informática.

Explicou-se a tais usuários qual a finalidade do sistema. Logo após foram separados 4 grupos. Cada grupo era composto por 10 usuários, sendo um tutor e 9 alunos. Após a realização do experimento todos responderam a um questionário de avaliação do sistema.

6.4.1 Quanto à Finalidade do Uso da Ferramenta

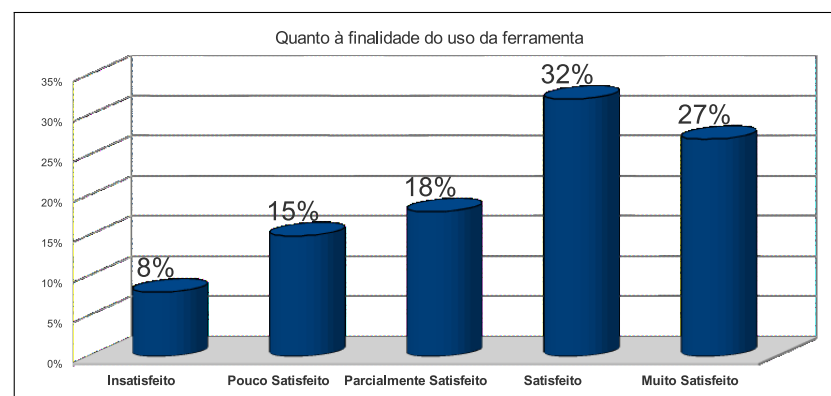


Figura 6.3: Gráfico: Finalidade da Ferramenta.

Como pode ser notado no gráfico da Figura 6.3, observa-se que a maioria das pessoas consideraram-se satisfeita com o sistema. Porém algumas pessoas ressaltaram que para operar o *software*, as pessoas necessitam de certo conhecimento de informática.

6.4.2 Quanto à Interface com o Usuário

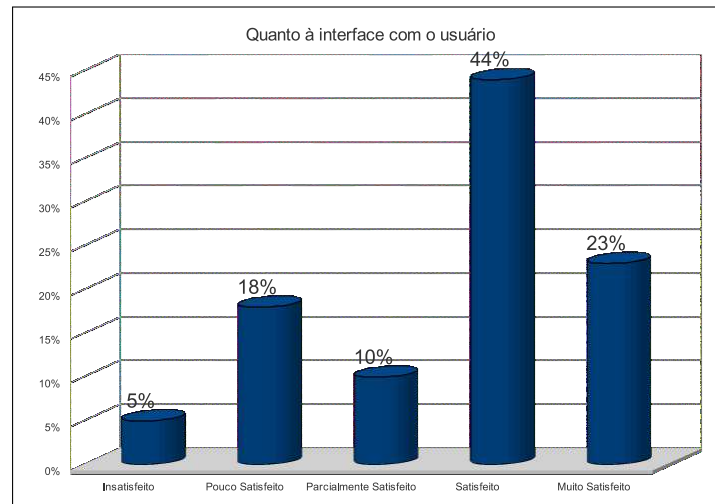


Figura 6.4: Gráfico: Interface com o Usuário.

No gráfico da Figura 6.4, observa-se que uma gama de usuários se sentiram *insatisfeitos*. E um justificou sua resposta dizendo que a interface do *software* não é intuitiva o suficiente. Mas a maioria dos usuários consideraram o *software* com uma interface agradável e bem semelhante as interfaces web tradicionais.

6.4.3 Quando à Colaboração entre os Usuários é Perceptível

No gráfico da Figura 6.5, observa-se que a maioria dos usuários avaliaram a colaboração entre os usuários perceptível, ficando muito satisfeito com o protótipo. Porém alguns usuários comentaram que o *software* poderia suportar mais objetos virtuais na cena. Tal procedimento será sugerido como trabalho futuro. Outra sugestão foi a possibilidade de implementar uma trava com o objetivo de forçar aos usuários que não estão com a trava, apenas assistirem a distribuição, o que possibilitaria o uso do protótipo como uma ferramenta interessante no caso de educação à distância.

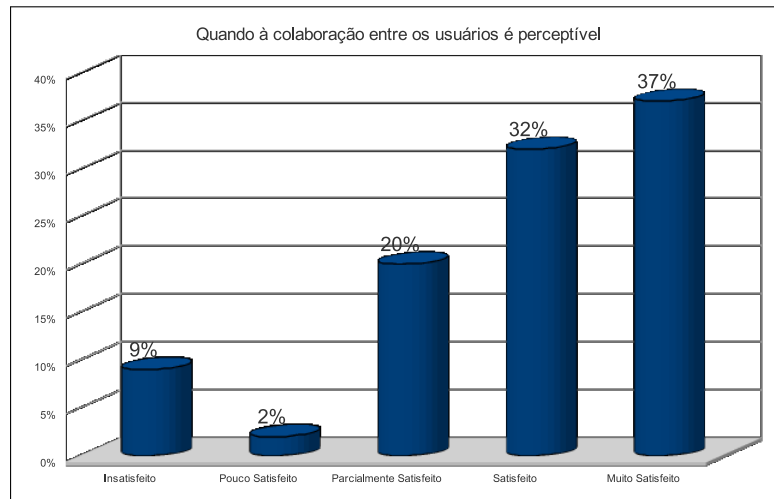


Figura 6.5: Gráfico: Colaboração entre Usuários.

Alguns usuários demonstraram-se insatisfeitos porém não justificaram suas respostas.

6.4.4 Quanto à Aprendizagem (Houve Aquisição de Conhecimento?)

O gráfico da Figura 6.6 demonstra que a maior parte dos usuários ficaram muito satisfeitos com a questão da aprendizagem, que o estudo de caso através do *software* RANET proporcionou. Porém alguns professores da área de informática, comentaram que o *software* poderia além do controle da animação, e a possibilita de alterar o objeto virtual visualizado na cena o *software* poderia possibilitar a inserção de vários objetos virtuais na cena.

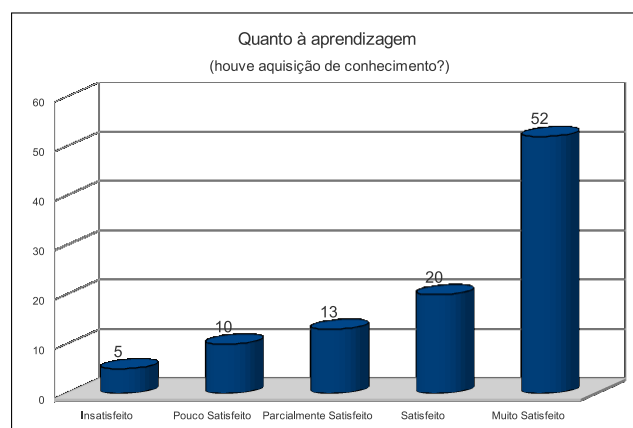


Figura 6.6: Gráfico: Aquisição de Conhecimento.

6.4.5 Quanto ao Número de Comandos Existentes no Sistema ?

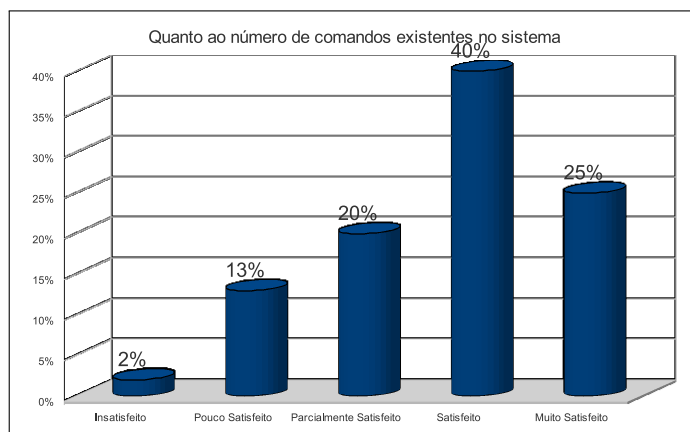


Figura 6.7: Gráfico: Comandos Existentes no Sistema.

Como pode ser observado no gráfico 6.7, a maior parte dos usuários se demonstraram satisfeitos com a quantidade de comandos existentes do *software* RANET. Uma pequena quantidade dos usuários se demonstraram insatisfeitos, porém não justificaram.

6.4.6 Quanto a Confiança Que o Sistema Executou o Que Foi Requisitado

O gráfico da Figura 6.8, demonstra que mais de 50% dos entrevistados demonstraram uma confiança que o software executa o que foi solicitado.

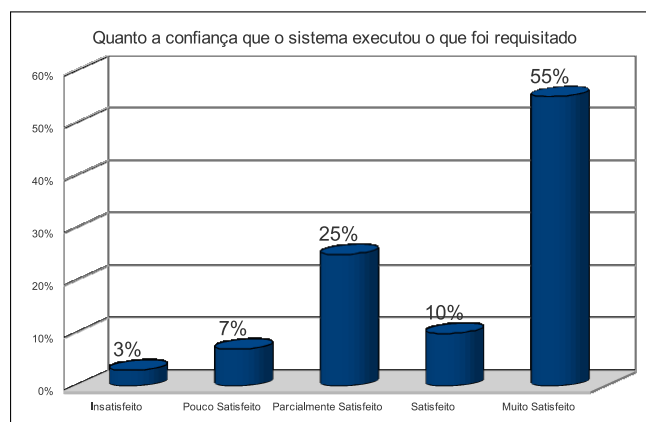


Figura 6.8: Gráfico: Confiança no Sistema.

6.4.7 Recursos não Implementados

Os tópicos a seguir mostram algumas limitações do *software* RANET.

- Não há a possibilidade de inserir múltiplos objetos virtuais na cena.
- A arquitetura não possibilita a comunicação cliente/cliente.
- A interface de Realidade Aumentada não está preparada para utilizar equipamentos HMD (*head mounted display*).

6.5 Considerações Finais

Este capítulo apresentou a avaliação do *software* tanto no ponto de vista tecnológico, quanto em relação aos testes com os usuários, demonstrando os resultados na forma de gráfico. Apesar de alguns recursos não terem sido implementados o *software* atendeu aos objetivos propostos.

Capítulo 7

Conclusões e Trabalhos Futuros

7.1 Introdução

Este capítulo tem como objetivo demonstrar os principais pontos estudados nesta dissertação, apresentar os possíveis trabalhos futuros e avaliar a principal contribuição deste trabalho para a área científica.

7.2 Conclusões

Durante a pesquisa, constatou-se que existem diversos ambientes distribuídos de Realidade Aumentada.

Este trabalho apresentou uma arquitetura para o sistema proposto e implementado um estudo de caso que apresentou o funcionamento de um motor de corrente contínua. Para tal, foram utilizadas as linguagens JAVA (J2EE), Flex (Action Script), o *middleware* BlazeDs, o conjunto de bibliotecas 3D *Papervision 3D* e o conjunto de biblioteca para o desenvolvimento de Realidade Aumentada para Web *Flartoolkit*.

Como conclusões da arquitetura, por meio do estudo de caso apresentado, pode-se citar que:

- O *middleware* BlazeDs apresentou um desempenho satisfatório na construção do ambiente distribuído voltado para web.

- No levantamento efetuado no decorrer deste trabalho, verificou-se que até o momento de desenvolvimento desta dissertação, nenhum ambiente distribuído de Realidade Aumentada utiliza o *middleware* BlazeDs para realizar a comunicação entre os módulos, o que representa uma contribuição deste trabalho.
- O protótipo apresentado nesta dissertação representa uma contribuição, pois o mesmo pôde ser utilizado como uma ferramenta de ensino, e devido ao fato de ser executado através da Web, representou uma ferramenta de ensino para a prática da Educação.
- Outra contribuição é a possibilidade de inserir novos objetos virtuais, dando a possibilidade do software desenvolvido abordar diversas áreas do conhecimento.
- Devido a arquitetura ter sido desenvolvida utilizando camadas, é possível reaproveitamento de código, pela expansão da arquitetura proposta, outra possibilidade é a adaptação das regras de negócio que estão presentes na camada de comunicação caso se deseje alterar algum comportamento do ambiente proposto é necessário alterar uma camada o que facilita a manutenção do software.

Quanto a interface de Realidade Aumentada o *Flartoolkit*, atendeu todas as expectativas possibilitando sua integração com as demais camadas de software da arquitetura proposta, sua integração e comunicação com a interface gráfica do Flex possibilitou um ambiente agradável para o usuário, baseado nos ambientes Webs atuais.

7.3 Trabalhos Futuros

De acordo com a evolução deste trabalho, alguns aspectos foram considerados importantes na consolidação de um produto final com todos os requisitos necessários para utilização do mesmo, tanto no meio acadêmico/científico quanto comercial. O objetivo deste trabalho é validar um aspecto específico, portanto não tem pretensão de resolver todos os problemas ou necessidades advindas desta pesquisa, possibilitando a este estudo ser referência para outros trabalhos futuros. Desta forma, destaca-se algumas contribuições futuras:

1. Adaptar a arquitetura, para possibilitar a utilização de múltiplos marcadores.
2. Implementar o reconhecimento de qualquer objeto como um marcador por exemplo uma caneta ser utilizada como marcador.
3. Implementar a arquitetura de comunicação ponto a ponto, para que as mensagens não precisem ser centralizadas por um servidor.
4. Implementar a utilização com o uso de dispositivos (*Head mounted display*)HMD.
5. Implementar um novo módulo para o *Flartoolkit* reconhecer objetos virtuais no formato VRML.
6. Implementar a comunicação entre os participantes através de um chat (que pode ser através de texto ou voz).
7. Adicionar marcadores de controle.
8. Desenvolver mecanismos para que os marcadores não fiquem visíveis.
9. Avaliar sua eficiência junto a dispositivos móveis.
10. Verificar sua aplicabilidade junto a programação nas nuvens.

Desta forma, criou-se nesta dissertação um Ambiente de Realidade Aumentada Distribuído pela Web, tendo como estudo de caso o motor de corrente contínua. O **RANET** possui fortes características Educacionais. O software é implementado utilizando o *Flartoolkit* e o *middleware* BalzeDs para realizar a distribuição dos objetos virtuais e através de um protocolo distribuir as animações de cada objeto. Entende-se que a partir deste estudo pode-se desenvolver novas interfaces distribuídas de Realidade Aumentada, e possibilitar sua disseminação através da rede mundial de computadores.

Referências bibliográficas

ADOBE. *Flash Player penetration*. 2010. Acessado em: abril/2010. Disponível em: <http://www.adobe.com/products/player_census/flashplayer/>.

AMES, A.; NADEAU, D.; MORELAND, J. *VRML 2.0 sourcebook*. [S.l.]: Wiley New York, 1997.

AMF 3 Specification. 2010. Disponível em: http://download.macromedia.com/pub/labs/amf/amf3_spec_121207.pdf Acesso em: 14 mar. 2010.

AZUMA, R. et al. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, IEEE Computer Society, p. 57–47, 2001.

BASTOS, N. C. *Arquitetura para dispositivos não convencionais de interação utilizando Realidade Aumentada: Um Estudo de Caso*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2005.

BATISTI, J. *Criando Aplicações em 3, 4 ou n Camadas*. 2010. Disponível em: <http://www.juliobattisti.com.br/artigos/ti/ncamadas.asp> Acesso em: 15 out. 2010.

BENFORD, S. et al. Supporting cooperative work in virtual environments. *The Computer Journal*, Br Computer Soc, v. 37, n. 8, p. 653, 1994.

BERNERS-LEE, T.; CAILLIAU, R. The world-wide web. *Communications of the ACM*, 1994.

BILLINGHURST, M.; KATO, H.; POUPYREV, I. The magicbook-moving seamlessly between reality and virtuality. *Computer Graphics and Applications, IEEE*, IEEE, v. 21, n. 3, p. 6–8, 2005. ISSN 0272-1716.

- BLAZEDS. *BlazeDS Developer Guide*. 2010. Disponível em:http://livedocs.adobe.com/blazeds/1/blazeds_devguide/ Acesso em: 15 mar. 2010.
- BREGA, J. et al. O EMPREGO DE REALIDADE AUMENTADA NA VIABILIZAÇÃO DA COMUNICAÇÃO EM LIBRAS. 2008.
- BRITO, A. *Blender 3D-Guia do Usuário*. [S.l.]: Novatec Editora, 2008.
- BUCIOLI, A. A. B. *Visualização e simulação de sinais cardiológicos utilizando técnicas de realidade aumentada*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, 2008.
- CARDOSO, A.; KIRNER, C.; LAMOUNIER, E. *Conceitos de Realidade Virtual e Aumentada*. [S.l.]: Tecnologias para o Desenvolvimento de Sistemas de Realidade Virtual e Aumentada, 2007.
- CARDOSO, A. et al. *Tecnologias para o desenvolvimento de sistemas de realidade virtual e aumentada*. [S.l.]: Universitária UFPE, 2006.
- COLLADA. *COLLABorative Design Activity*. 2010. Disponível em: <http://www.collada.com> Acesso em: 08 out. 2010.
- CONSULARO, L. et al. ARToolKit: Aspectos Técnicos e Aplicações Educacionais. *Realidade Virtual: Uma Abordagem Prática. Livro dos Minicursos do SVR2004, SBC, São Paulo*, p. 141–183, 2004.
- COULOURIS, G. *Distributed systems: concepts and design*. [S.l.]: Addison-Wesley Longman, 2005.
- DOCUMENTATION, D. Disponível em: <http://www.cc.gatech.edu/dart/>. Acesso em, v. 30, n. 01, 2010.
- ECKHOUSE, R.; STANKOVIC, J.; DAM, A. V. Issues in distributed processing-An overview of two workshops. *Computer*, v. 11, n. 1, p. 22–26, 1978.
- ECMAScript, E. *ECMAScript Language Specification*.

- FIALA, M. ARTag, a fiducial marker system using digital techniques. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*. [S.l.: s.n.], 2005. v. 2.
- FIELDING, R. et al. Hypertext Transfer Protocol-HTTP/1.1. 1999.
- FRAHM, J. et al. Markerless augmented reality with light source estimation for direct illumination. In: CITESEER. *Conference on Visual Media Production CVMP, London*. [S.l.], 2005.
- FRANKLIN, D.; MAKAR, J. *Macromedia Flash MX 2004 ActionScript: training from the source*. [S.l.]: Macromedia Press, 2003.
- GERALDI, T. H.; KIRNER, C. Ambientes colaborativos com realidade aumentada. *WRA 2005 - II Workshop de Realidade Aumentada*, p. 13 – 16, 2005.
- GILLEANES, T. UML–Uma Abordagem Prática 2. *Novatec Editora*, 2009.
- GORALSKI, W.; POLI, M.; VOGEL, P. *VRML: exploring virtual worlds on the Internet*. [S.l.]: Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1996.
- GOSLING, J. et al. *Java (TM) Language Specification, The (Java (Addison-Wesley))*. [S.l.]: Addison-Wesley Professional, 2005.
- GREANIER, T. *Java foundations*. [S.l.]: Sybex Inc, 2004.
- KATO, H.; BILLINGHURST, M. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: SAN FRANCISCO, CA. *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*. [S.l.], 1999. v. 99, p. 85–94.
- KIRNER, C.; MENDES, S. *Sistemas Operacionais Distribuídos-Aspectos Gerais e Análise de sua Estrutura*. *Editora Campus*, 1988.
- KIRNER, C.; SISCOOTTO, R. Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações. In: *Petrópolis–RJ, Livro do Pré-Simpósio, IX Symposium on Virtual Reality*. *Editora SBC–Sociedade Brasileira de Computação*. [S.l.: s.n.], 2007. v. 412, p. 85–7669.

- KIRNER, C.; TORI, R. Introdução à Realidade Virtual, Realidade Misturada e Hiper-realidade. *Realidade Virtual*, p. 3, 2004.
- KIRNER, C.; ZORZAL, E. Aplicações Educacionais em Ambientes Colaborativos com Realidade Aumentada. In: *Anais do Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2005. v. 1, n. 1, p. 114.
- LAMPORT, L. Lower bounds for asynchronous consensus. *Distributed Computing*, Springer, v. 19, n. 2, p. 104–125, 2006.
- LICENSE, G. Disponível em: <http://www.gnu.org/licenses/gpl.html>. Acesso em, v. 21, n. 01, 2010.
- MACINTYRE, B. et al. Dart: The designer's augmented reality toolkit. In: *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*. [S.l.: s.n.], 2003. p. 329–330.
- MALIK, S.; ROTH, G.; MCDONALD, C. Robust 2d tracking for real-time augmented reality. *configurations*, Citeseer, v. 1, n. 2, p. 12, 2002.
- MARCHAND, E. et al. Robust real-time visual tracking using a 2D-3D model-based approach. In: *IEEE Int. Conf. on Computer Vision, ICCV'99*. [S.l.: s.n.], 2007. v. 1, p. 262–268.
- MILGRAM, P. et al. Augmented reality: A class of displays on the reality-virtuality continuum. In: CITESEER. *Proceedings of Telem manipulator and Telepresence Technologies*. [S.l.], 1994. p. 282–292.
- MULLENDER, S. Introduction to distributed systems. *CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH-REPORTS-CERN*, CERN, p. 29–29, 1993.
- NOGUEIRA, K. et al. Uma arquitetura para a distribuição de realidade virtual e aumentada. *6º Workshop de Realidade Virtual e Aumentada*, 2009.
- NOGUEIRA, K. D. F. C. *Desenvolvimento de uma arquitetura de distribuição de realidade virtual e aumentada aplicada em ambientes educacionais*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, 2010.

- OLIVEIRA, L. de A. *Desenvolvimento do NetARToolkit: Um Sistema Distribuído de Realidade Aumentada*. Dissertação (Mestrado) — Universidade Metodista de Piracicaba, 2008.
- PARDO, R. *Interprocess communication and synchronization for distributed systems*. Tese (Doutorado) — The Ohio State University, 1979.
- POLLEFEY, M. Self Calibration and metric 3D reconstruction from uncalibrated image sequences. *status: published*, 1999.
- PRESSMAN, R. *Engenharia de Software-6a edição*. McGraw-Hill Interamericana do Brasil, São Paulo, Brasil., 2006.
- PROINFO. *Programa Nacional de Tecnologia Educacional*. 2010. Disponível em:http://portal.mec.gov.br/index.php?option=com_content&view=article&id=244&Itemid=823 Acesso em: 27 ago. 2010.
- PRÜMPER, J. Test it: ISONORM 9241/10. In: L. ERLBAUM ASSOCIATES INC. *Proceedings of HCI International (the 8th International Conference on Human-Computer Interaction) on Human-Computer Interaction: Ergonomics and User Interfaces-Volume I-Volume I*. [S.l.], 1999. p. 1032.
- QUINN, B. *Windows sockets network programming*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1998.
- RIBEIRO, M. W. de S. *Arquitetura para Distribuição de Ambientes Virtuais Multidisciplinares*. Tese (Doutorado) — Universidade Federal de Uberlândia, 2006.
- RINALDI, L. C. A. et al. Ambientes virtuais distribuídos e compartilhados. *Fundamentos e Tecnologia de Realidade Virtual e Aumentada*, v. 1, p. 60–78, 2006.
- SANTIN, R. *Sistema de Autoria Em Ambiente Colaborativo com Realidade Aumentada*. Dissertação (Mestrado) — Universidade Metodista de Piracicaba, 2008.
- SANTIN, R.; KIRNER, C. Artoolkit. *Tecnologias para o desenvolvimento de sistemas de realidade virtual e aumentada*, p. 91–109, 2007.

- SANTOS, J. A.; ANDRADE ângela christina S. As tecnologias: Auxílio ao processo de ensino/aprendizagem. *II seminário educação, comunicação, inclusão e interculturalidade*, 2009.
- SILVA, W. A. da. *Uma Arquitetura para Distribuição de Ambientes Virtuais de Realidade Aumentada*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, 2008.
- SINGHAL, S.; ZYDA, M. *Networked virtual environments: design and implementation*. [S.I.]: ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1999.
- SOWIZRAL, K.; RUSHFORTH, K.; SOWIZRAL, H. *The Java 3D API Specification*. [S.I.]: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1997.
- SPECTOR, A. Multiprocessing Architectures for Local Computer Networks. *Dissertation Abstracts International Part B: Science and Engineering*[DISS. ABST. INT. PT. B- SCI. & ENG.],, v. 42, n. 8, p. 1982, 1982.
- SUTHAU, T. et al. A concept work for Augmented Reality visualisation based on a medical application in liver surgery. *INTERNATIONAL ARCHIVES OF PHOTOGRAMMETRY REMOTE SENSING AND SPATIAL INFORMATION SCIENCES*, Citeseer, v. 34, n. 5, p. 274–280, 2002. ISSN 1682-1750.
- TANENBAUM, A.; STEEN, M. V. *Distributed systems*. [S.I.]: Citeseer, 2002.
- TEAM, A. *ActionScript 3.0 for Adobe Flash CS4 Professional Classroom in a Book*. Adobe Press, 2008.
- TÉCNICA, P. *MOTORES DE CORRENTE CONTÍNUA*. 2006. Disponível em: www.siemens.com.br/medias/FILES/2910_20060505141908.pdf Acesso em: 14 mar. 2010.
- TONDEUR, P.; WINDER, J. *Papervision3d Essentials*. [S.I.]: Packt Publishing, 2009.
- VACCHETTI, L.; LEPETIT, V.; FUA, P. Stable real-time 3 D tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Citeseer, v. 26, n. 10, p. 1385–1391, 2004.

WAGNER, D.; BARAKONYI, I. Augmented reality kanji learning. In: IEEE COMPUTER SOCIETY. *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*. [S.l.], 2003. p. 335.

WAGNER, D.; SCHMALSTIEG, D. Artoolkit on the pocketpc platform. In: CITESEER. *IEEE Augmented Reality Toolkit Workshop*. [S.l.], 2003. p. 14–15.

WAGNER, D.; SCHMALSTIEG, D. Artoolkitplus for pose tracking on mobile devices. In: *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)*. [S.l.: s.n.], 2007. p. 139–146.

WINETT, G. The Definition of a Socket. *Request for Comments (RFC)*, v. 147, 1971.

YANG, X.; CHEN, Q. Virtual reality tools for internet-based robotic teleoperation. In: IEEE COMPUTER SOCIETY. *Proceedings of the 8th IEEE International Symposium on Distributed Simulation and Real-Time Applications*. [S.l.], 2004. p. 236–239.

ZHOU, Z. et al. Interactive entertainment systems using tangible cubes. In: YUSUF PISAN. *Proceedings of the First Australian Workshop on Interactive Entertainment*. [S.l.], 2004. p. 19.

ZORZAL, E.; KIRNER, C. Jogos Educacionais em Ambiente de Realidade Aumentada. In: *II Workshop sobre Realidade Aumentada. Unimep. Piracicaba, SP*. [S.l.: s.n.], 2005.

ZORZAL, E. R. *Estratégia para o Desenvolvimento de Aplicações Adaptativas de Visualização de Informações com Realidade Aumentada*. Tese (Doutorado) — Universidade Federal de Uberlândia, 2009.

Apêndice A

Avaliação do Sistema - RANET

Avaliador: _____

Nível de Escolaridade: () Ensino Médio () Ensino Superior () Pós-Graduação

Nome do Curso: _____

Questionário	Insatisfeito	Pouco Satisfeito	Parcialmente Satisfeito	Satisfeito	Muito Satisfeito
Quanto à finalidade do uso da ferramenta?					
Quanto à interface com o usuário?					
Quanto à colaboração entre os usuários é perceptível?					
Quanto à aprendizagem (Houve aquisição de conhecimento)?					
Quanto ao número de comandos existentes no sistema?					
Quanto a confiança que o sistema executou o que foi requisitado					

Apêndice B

Manual de Utilização do software - *RANET*

O software RANET contém uma interface gráfica intuitiva, possibilitando ao usuário, o acesso à todas as funcionalidades do sistema. Através de sua tela principal, Figura B.1.

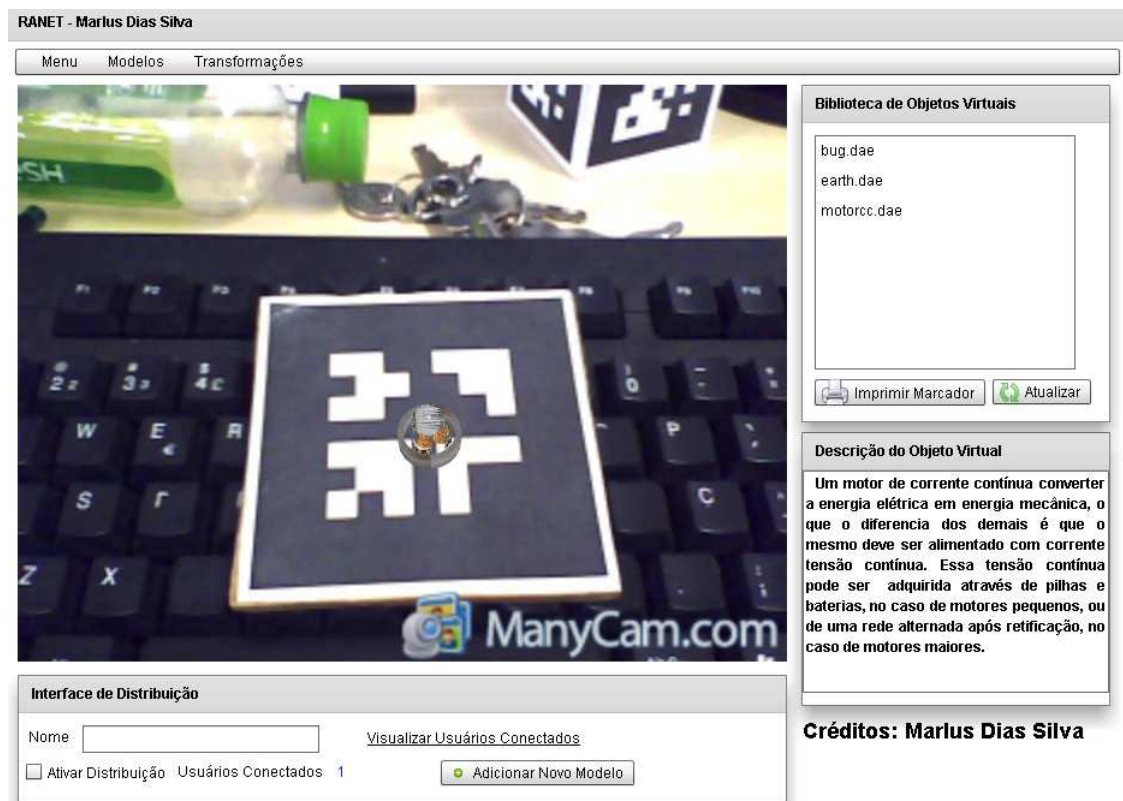


Figura B.1: Interface gráfica do software RANET.

Caso o usuário não possua um marcador, ele pode imprimir o marcador clicando no botão **Imprimir Marcador**.

O Botão atualizar, possibilita a atualização dos objetos virtuais adicionados nas biblioteca de objetos virtuais.



Figura B.2: Interface de Distribuição.

A interface gráfica responsável pela distribuição contém as seguintes possibilidades, como demonstra a Figura B.2:

- A caixa de seleção **Ativar Distribuição**: Ativa a distribuição das transformadas geométricas e objetos virtuais com todos os participantes do sistema.
- O label **Usuários Conectados**: Informa a quantidade de usuários que estão conectados no sistema no momento.
- O link **Visualizar Usuários Conectados**: Abre uma janela que demonstra o nome de todos os usuários conectados no sistema, como demonstra a Figura B.3.
- O botão **Adicionar Novo Modelo**: Possibilita adicionar novo modelos virtuais na biblioteca de objetos virtuais.

B.1 Menus Interativos

O software contém uma gama de menus Figura B.4, que possibilita o sistema executar diversas operações. Os menus estão localizados na parte superior do software. Através desse menus, o usuário pode realizar operações de escala, movimentações nas posições (x,y,z), executar animações pré definidas como aumentar a velocidade do motor, pausar, ver seu funcionamento com ele aberto e demais funcionalidade.



Figura B.3: *Pop-up*, Lista dos Usuários Conectados no Sistema.

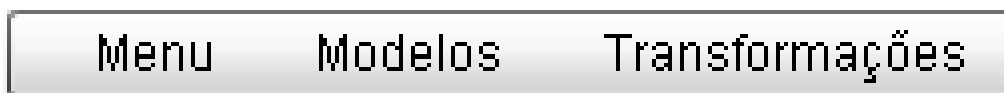


Figura B.4: Menus Interativos.

Apesar do software ter sido projeto pensando no estudo de caso descrito nesta dissertação, o usuário pode adicionar novos modelos virtuais e executá-los na sistema.

B.2 Biblioteca de Objetos Virtuais

A Biblioteca de objetos virtuais, possibilita que o usuário adicione novos modelos virtuais, é importante ressaltar que o usuário deve obedecer ao protocolo definido nessa dissertação para a criação dos novos modelos virtuais que serão adicionados no software em questão.

O RANET foi desenvolvido pensando na internet, onde o usuário para utilizar o software não precisa se preocupar em instalar nenhum serviço adicional em seu computador o único pré requisito do software é que tenha instalado no computador o flash player, que atualmente está presente em 99% dos navegadores web.