

Alice Shimada Bacic

**Um Ambiente Colaborativo de Apoio ao Diagnóstico  
Médico Assistido por Computadores de Alto  
Desempenho**

Dissertação apresentada à Escola  
Politécnica da Universidade de  
São Paulo para obtenção do título  
de Mestre em Engenharia.

*São Paulo*  
*2001*

Alice Shimada Bacic

# Um Ambiente Colaborativo de Apoio ao Diagnóstico Médico Assistido por Computadores de Alto Desempenho

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia.

Área de Concentração:  
Sistemas Eletrônicos

*Orientador:*  
*Marcelo Knörich Zuffo*

*São Paulo*  
*200*



Dedico esta dissertação aos meus pais Jacy e Francisco que tanto me apoiaram em todos os momentos.

## **Agradecimentos**

Agradeço aos meus pais, Jacy Shimada Bacic e Francisco Bacic Fratric Neto, que muito me incentivaram e que sempre me apoiaram., ao meu orientador Marcelo Knörich Zuffo e esposa Roseli de Deus Lopez pela orientação, apoio e incentivos durante todo tempo em que trabalhamos juntos, aos amigos Fábio José Ayres e João Eduardo Kogler pela ajuda prestada e pelas sugestões que tanto enriqueceram este trabalho, aos amigos Marco Antônio Simon dal Poz, Matteo Nava, Jefferson Máximo da Silva, Alexei Znamensky, Luciano Pereira Soares e Ricardo Bittencourt Vidigal Leitão pelo suporte técnico e paciência nas horas mais atribuladas, aos amigos Edson Amaro Junior e Maria Fernanda de Souza Pereira Lopes pelas sugestões tão úteis para a escrita desta dissertação e enfim a todos que permaneceram ao meu lado desde o início deste trabalho.

## Sumário

<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1. OBJETIVOS .....	1
1.2. MOTIVAÇÃO .....	2
1.3. RELEVÂNCIA .....	2
1.4. ESTRUTURA DA DISSERTAÇÃO .....	3
<b>2. ESTADO-DA-ARTE E CONCEITUAÇÃO .....</b>	<b>4</b>
2.1. TELEMEDICINA .....	4
2.1.1. <i>Resumo da Seção</i> .....	7
2.2. VISUALIZAÇÃO DE IMAGENS MÉDICAS .....	8
2.2.1. <i>Modelo de Representação de Dados</i> .....	8
2.2.2. <i>Resumo da Seção</i> .....	9
2.3. APLICAÇÕES DISTRIBUÍDAS .....	10
2.4. CSCW (COMPUTER SUPPORTED COOPERATIVE WORK) E GROUPWARE .....	11
2.5. ARQUITETURAS CLIENTE SERVIDOR .....	13
2.6. OBJETOS DISTRIBUÍDOS .....	15
2.6.1. <i>Sistema Heterogêneos</i> .....	16
2.7. CORBA (COMMON OBJECT REQUEST BROKER ARCHITECTURE) .....	18
2.7.1. <i>OMA (Object Management Architecture)</i> .....	21
2.8. CONSIDERAÇÕES FINAIS .....	30
<b>3. CONCEPÇÃO E PROJETO DE SISTEMA COOPERATIVO PARA AUXÍLIO AO DIAGNÓSTICO MÉDICO POR IMAGENS.....</b>	<b>31</b>
3.1. INTRODUÇÃO .....	31
3.2. ANÁLISE DE REQUISITOS .....	31
3.2.1. <i>Descrição do Problema</i> .....	31
3.2.2. <i>Processo de Análise</i> .....	32
3.2.2.1 Situação Atual do Hospital .....	33
3.2.2.2 Infra-estrutura de Computação do Hospital .....	34
3.2.2.3 Rotina de Diagnóstico no Setor de Radiologia .....	34
3.2.2.4 Características do Dados .....	35
3.2.3. <i>Alguns Conceitos</i> .....	39
3.2.4. <i>Avaliação dos Requisitos</i> .....	40
3.2.5. <i>Proposta</i> .....	40
3.3. ESPECIFICAÇÃO FUNCIONAL .....	43
3.3.1. <i>Introdução</i> .....	43
3.3.2. <i>Visão Geral do Projeto</i> .....	44
3.3.3. <i>Serviços de CSCW</i> .....	46
3.3.3.1 Visualização e Manipulação de Dados .....	46
3.3.3.2 Gerenciamento de Usuários e Grupos de Trabalho .....	50
3.3.3.3 Organização do Grupo de Trabalho e Comunicação .....	54
3.3.3.4 Acesso a dados compartilhados .....	57
3.3.4. <i>Serviços de processamento de dados radiológicos</i> .....	58
3.4. ARQUITETURA DO SISTEMA .....	59
3.4.1. <i>Servidor de CSCW</i> .....	61
3.4.1.1 Projeto do Banco de Dados .....	62
3.4.1.2 Diagrama de objetos .....	64
3.4.2. <i>Servidor VV (Servidor de Visualização Volumétrica)</i> .....	65
3.4.2.1 Diagrama de Objetos .....	67
3.4.3. <i>Interface Cliente</i> .....	68
3.4.3.1 Diagrama de classes .....	70
3.4.4. <i>Ferramentas de Auxílio ao Trabalho Cooperativo</i> .....	72
3.4.5. <i>Ferramentas de Implementação</i> .....	72
3.5. RESUMO DO CAPÍTULO .....	74
<b>4. CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO DE UM SISTEMA COLABORATIVO PARA TELERADIOLOGIA.....</b>	<b>75</b>

4.1.	INTRODUÇÃO .....	75
4.2.	CONSIDERAÇÕES GERAIS .....	75
4.2.1.	<i>Metodologia de Projeto</i> .....	78
4.2.2.	<i>Expectativas em Telemedicina e o Trabalho Proposto</i> .....	79
4.3.	AMBIENTE DE DESENVOLVIMENTO .....	82
4.4.	INFRA-ESTRUTURA E PLATAFORMAS ADOTADAS .....	82
4.5.	SISTEMA DESENVOLVIDO .....	85
4.5.1.	<i>Implementação da Arquitetura</i> .....	85
4.5.2.	<i>Processo de Desenvolvimento do Ambiente Cooperativo</i> .....	88
4.5.2.1	Análise do Processo de Desenvolvimento .....	90
4.6.	RESUMO DA SEÇÃO .....	93
<b>5.</b>	<b>AVALIAÇÃO E ANÁLISE DA IMPLEMENTAÇÃO DE UM SISTEMA COLABORATIVO PARA TELERADIOLOGIA.....</b>	<b>94</b>
5.1.	INTRODUÇÃO .....	94
5.2.	DESCRIÇÃO DE TESTES REALIZADOS .....	94
5.2.1.	<i>Teste de Estabilidade do Servidor</i> .....	94
5.2.2.	<i>Teste transmissão de dados pela rede</i> .....	95
5.2.3.	<i>Teste de Número de Usuários em Sessão Cooperativa</i> .....	97
5.3.	AVALIAÇÃO GERAL DA DO SISTEMA SEGUNDO A CLASSIFICAÇÃO DE TEMPO-ESPAÇO .....	100
5.4.	AVALIAÇÃO SOBRE A PERSPECTIVA DA TELEMEDICINA .....	101
5.5.	CONCLUSÕES .....	102
<b>6.</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS.....</b>	<b>103</b>
6.1.	CONTRIBUIÇÕES DESTE TRABALHO .....	103
6.2.	TRABALHOS FUTUROS .....	105
6.3.	CONCLUSÕES .....	106
<b>7.</b>	<b>BIBLIOGRAFIA.....</b>	<b>108</b>
<b>8.</b>	<b>ENDEREÇOS ELETRÔNICOS CITADOS.....</b>	<b>114</b>
<b>9.</b>	<b>ANEXO 1 - DICIONÁRIO DE DADOS .....</b>	<b>115</b>

## Lista de Figuras

FIGURA 1: DEFINIÇÃO DE TELEMEDICINA.....	4
FIGURA 2: MODALIDADES E REQUISITOS.....	6
FIGURA 3: MODELO DE IMAGEM.....	9
FIGURA 4: MODELO DE VOLUME.....	9
FIGURA 5: CLASSIFICAÇÃO NO TEMPO-ESPAÇO DO GROUPWARE [ELLIS91].....	13
FIGURA 6: PREEXISTÊNCIA.....	18
FIGURA 7: COMPONENTES <i>PLUG-AND-PLAY</i> .....	18
FIGURA 8: PORTABILIDADE.....	19
FIGURA 9: TRANSPARÊNCIA.....	19
FIGURA 10: INTEROPERABILIDADE.....	19
FIGURA 11: ARQUITETURA OMA ( <i>OBJECT MANAGEMENT ARCHITECTURE</i> ) [ORFALI96].....	25
FIGURA 12: ARQUITETURA CORBA [OMG97-R2.1].....	26
FIGURA 13: FLUXO DOS DADOS MÉDICOS.....	39
FIGURA 14: SERVIÇOS OFERECIDOS PELO AMBIENTE.....	45
FIGURA 15: PROTÓTIPO DE INTERFACE GRÁFICA.....	48
FIGURA 16: ARQUITETURA DO SISTEMA.....	60
FIGURA 17: ESQUEMA DO BANCO DE DADOS DE USUÁRIOS.....	63
FIGURA 18: DIAGRAMA DE CLASSES DO SERVIDOR DE CSCW.....	65
FIGURA 19: ESTRUTURA DE ACESSO AO SERVIDOR VV.....	66
FIGURA 20: DIAGRAMA DE CLASSES DO SERVIDOR VV.....	67
FIGURA 21: TROCA DE DADOS ENTRE APLICAÇÕES.....	68
FIGURA 22: REPRESENTAÇÃO DE DADOS.....	69
FIGURA 23: DIAGRAMA DE CLASSES DA INTERFACE CLIENTE.....	71
FIGURA 24: ESQUEMA DE FUNCIONAMENTO DAS FERRAMENTAS AUXILIARES.....	72
FIGURA 25: ARQUITETURA DE CIRCUITARIA E AMBIENTE DE PROGRAMAÇÃO UTILIZADA NO DESENVOLVIMENTO E TESTES.....	84
FIGURA 26: TELA PRINCIPAL DA APLICAÇÃO GRÁFICA.....	86
FIGURA 27: TELA DE CONEXÃO COM OS SERVIÇOS DE CSCW.....	87
FIGURA 28: TELA DE VISUALIZAÇÃO DE EXAMES.....	87
FIGURA 29: CONFIGURAÇÃO FINAL DO AMBIENTE COOPERATIVO.....	88
FIGURA 30: COMPONENTES DA INTERFACE DE VISUALIZAÇÃO.....	89
FIGURA 31: ESTRUTURA DO SERVIDOR DE CSCW.....	90
FIGURA 32: REQUISITOS ATENDIDOS SEGUNDO A CLASSIFICAÇÃO NO TEMPO-ESPAÇO DO GROUPWARE [ELLIS91].....	101
FIGURA 33: DEFINIÇÃO DE TELEMEDICINA E TECNOLOGIAS APLICADAS.....	102



## Lista de Tabelas

TABELA 1: ATIVIDADES DE ANÁLISE DE REQUISITOS.....	33
TABELA 2: COMPARAÇÃO ENTRE FORMATOS DE DADOS.....	36
TABELA 3: LISTA DE CRITÉRIOS PARA VISUALIZAÇÃO DE IMAGENS MÉDICAS SEGUNDO A ACR/NEMA.....	38
TABELA 4: CARACTERÍSTICAS GERAIS DE SISTEMAS DE COLABORATIVOS.....	42
TABELA 5: CARACTERÍSTICAS GERAIS DE SISTEMAS DE COLABORATIVOS ORGANIZADAS SEGUNDO PRIORIDADE.....	43
TABELA 6: FUNCIONALIDADES DE MANUTENÇÃO DOS SERVIÇOS DE CSCW.....	52
TABELA 7: FUNÇÕES PARA VERIFICAÇÃO DO ESTADO DO SERVIDOR.....	53
TABELA 8: FUNCIONALIDADES DE GERENCIAMENTO DINÂMICO DE USUÁRIOS.....	53
TABELA 9: FUNCIONALIDADES PARA GERENCIAMENTO DINÂMICO DE GRUPOS DE TRABALHOS.....	54
TABELA 10: FUNCIONALIDADES DE ORGANIZAÇÃO DO GRUPO.....	55
TABELA 11: FUNÇÕES DA AGENDA ELETRÔNICA.....	55
TABELA 12: SERVIÇOS DE PROCESSAMENTO DE IMAGENS.....	59
TABELA 13: RESULTADO DA COMPILAÇÃO EM DIVERSAS ARQUITETURAS.....	91
TABELA 14: EXECUÇÃO DA APLICAÇÃO JAVA EM DIVERSAS ARQUITETURAS.....	92
TABELA 15: DESEMPENHO DA APLICAÇÃO COOPERATIVA EM DIVERSAS PLATAFORMAS.....	93
TABELA 16: RESULTADOS DO TESTE DE TRANSMISSÃO DE DADOS.....	96
TABELA 17: RESULTADOS DO TESTE DE DESEMPENHO DA APLICAÇÃO GRÁFICA COM (3 APLICAÇÕES).....	99
TABELA 18: RESULTADOS DO TESTE DE DESEMPENHO DA APLICAÇÃO GRÁFICA COM (4 APLICAÇÕES).....	99
TABELA 19: RESULTADOS DO TESTE DE DESEMPENHO DA APLICAÇÃO GRÁFICA COM (5 APLICAÇÕES).....	99

## Lista de Gráficos

GRÁFICO 1: RESULTADOS DO TESTE DE TRANSMISSÃO DE DADOS. ....	96
--	----

## Lista de Siglas

ACR	American College of Radiology
NEMA	National Electrical Manufacturers Association
DICOM	Digital Imaging and Communications in Medicine
CSCW	Computer Supported Cooperative Work
API	Application Program Interface
OMG	Object Management Group
CORBA	Common Object Request Broker
OMA	Object Management Architecture
ORB	Object Request Broker
IDL	Interface Definition Language
IOR	Interoperable Object Reference
SQL	Simple Query Language
ISO	International Standards Organisation
OQL	Object Query Language
ODMG	Object Database Management Group
POS	Persistent Object Service
DII	Dynamic Invocation Interface
DSI	Dynamic Skeleton Interface
BOA	Basic Object Adapter
POA	Portable Object Adapter
CDR	Common Data Representation
GIOP	General Inter-ORB Protocol
IIOP	Internet Inter-ORB Protocol
ESIOP	Environment Specific Inter-ORB Protocol
DCE-CIOP	DCE Common Inter-ORB Protocol
OSF	Open System Foundation
DCE	Distributed Computing Environment
RPC	Remote Procedure Call
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IP	Internet Protocol
SPMD	Simple Program Multiple Data
CT	Computer Tomography
MRI	Magnetical Resonance Imaging
RGB	Red/Blue/Green
GUI	Graphical User Interface
RMI	Remote Method Invocation
COM	Component Object Model

## Resumo

Atualmente o diagnóstico médico baseado em imagens é uma tarefa complexa envolvendo médicos que cuidam do atendimento local e especialistas dispersos geograficamente. Particularmente em radiologia, muitas aplicações de diagnóstico médico são baseadas em técnicas de processamento de imagens de grande complexidade computacional envolvendo a manipulação de grandes bases de dados. Neste trabalho propomos o desenvolvimento de um ambiente cooperativo de teleradiologia que permita o compartilhamento de recursos avançados entre especialistas fisicamente dispersos. Para atingir o nosso objetivo utilizaremos técnicas CSCW (*Computer Supported Cooperative Work*) que permitem a interação entre usuários numa rede WAN (*Wide Area Network*) e o compartilhamento dos recursos computacionais oferecidos através de servidores.

## **Abstract**

Nowadays the image based medical diagnosis is a complex task involving doctors that take care of the local attendance and specialists geographically dispersed. Particularly in radiology, many applications for medical diagnosis are based on techniques of image processing where great computation complexity and large databases are required. In this work we propose the development of a cooperative environment for teleradiology that allow advanced resources sharing among several users geographically dispersed. To reach our objective we will use CSCW (Computer Supported Cooperative Work) techniques that allow the interaction among users in a WAN (Wide Area Network) network and the computational resources sharing offered by special servers.

## 1. Introdução

A principal contribuição deste trabalho é a pesquisa, o projeto e a construção de um ambiente que permita aos médicos interagirem através de uma rede de computadores, tendo como ferramenta uma interface gráfica para visualização e análise em grupo de imagens radiológicas de diferentes modalidades, sendo o processamento das imagens médicas realizado transparentemente em máquinas de alto desempenho.

Neste trabalho, denominamos **ambiente cooperativo para medicina** a infra-estrutura que permite a um grupo de médicos a discussão de um caso clínico através do compartilhamento de dados, informações e experiências, com o objetivo de agilizar o processo de decisão e propiciar um diagnóstico final mais preciso.

O projeto e a implementação do ambiente de trabalho cooperativo proposto envolve estudos nas áreas de comunicação em rede, processamento distribuído, processamento de imagens médicas, interfaces gráficas e telemedicina.

O processamento das imagens médicas deve ser feito em servidores residentes em máquinas de alto desempenho, devido à necessidade de processamento intensivo e ao grande volume de dados intrínsecos à área de atuação. A interação entre as interfaces gráficas, os clientes, e os servidores envolve o estudo de protocolos de comunicação em rede e transferência de grandes quantidades de dados.

Como resultado final espera-se ter um ambiente que possa ser classificado como um ambiente de teleradiologia, permitindo aos médicos analisar as imagens radiológicas e interagirem na busca de um diagnóstico mais preciso e rápido. Tal ambiente é composto por interfaces gráficas cliente utilizadas na análise das imagens, servidores destinados ao processamento destas imagens, ambiente de comunicação entre clientes e servidores e infra-estrutura que proporcione a interação entre os usuários que buscam o diagnóstico compartilhado.

### 1.1. Objetivos

O objetivo principal da pesquisa proposta é o oferecimento de um ambiente e ferramentas que promovam o trabalho cooperativo entre médicos e especialistas

dispersos geograficamente, através de uma rede de computadores. Além de promover o trabalho cooperativo, o ambiente proposto deverá fornecer recursos para o processamento de imagens em máquinas de alto desempenho remotas. A combinação destes recursos objetiva o aumento da velocidade do diagnóstico bem como a melhoria de sua qualidade na área de radiologia.

## **1.2. Motivação**

Após diversas visitas ao Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo (HCFMUSP) foi possível levantar as características do processo de diagnóstico por imagens radiológicas. Da análise deste processo conclui-se que a introdução de novos métodos e novas tecnologias pode melhorar e agilizar este processo além de diminuir custos, uma vez que se passa a ter tratamentos mais adequados em virtude de melhorias no processo de diagnóstico.

A introdução dos computadores no processo de análise das imagens médicas traz muitos benefícios como o aumento da precisão do diagnóstico e a diminuição do tempo de decisão, alcançados através de interfaces gráficas de fácil manipulação, processamento digital de imagens e a cooperação entre múltiplos especialistas separados geograficamente, porém ligados por uma rede de computadores.

Diversos aspectos devem ser considerados na introdução deste métodos e tecnologias novas à rotina prática dos médicos. Devemos pensar em problemas desde a super lotação dos grandes centros médicos, como o Hospital das Clínicas, dotados de melhores recursos humanos e tecnológicos, até os riscos e custos envolvidos no transporte de pacientes e o tempo gasto no deslocamento de especialistas.

A utilização de redes de computadores dotados de ferramentas adequadas ao uso em medicina, pode facilitar e agilizar o diagnóstico, uma vez que torna-se possível compartilhar dados e experiências entre os diversos centros médicos e seus especialistas.

## **1.3. Relevância**

O Brasil é um país de dimensões continentais, havendo uma grande concentração de especialistas da área médica na região sul e sudeste, regiões de maior desenvolvimento e recursos financeiros.

Assim, do ponto de vista clínico podemos citar como benefícios da introdução das ferramentas de trabalho cooperativo a melhoria da qualidade de diagnóstico em locais afastados dos grandes centros médicos, uma vez que se facilita o acesso ao conhecimento dos especialistas pertencentes a estes centros e a diminuição de custos e riscos, visto que elimina-se a necessidade de deslocamento dos médicos envolvidos no diagnóstico ou mesmo do próprio paciente.

A área proposta para pesquisa possui uma natureza multidisciplinar, envolvendo diversos conceitos técnicos. Assim, do ponto de vista científico a relevância do projeto relaciona-se às pesquisas nas áreas de CSCW, teleradiologia, objetos distribuídos, interfaces gráficas e a integração das áreas mencionadas em um ambiente que proverá ferramentas para auxílio ao diagnóstico médico. Um outro aspecto interessante a ser considerado é a concepção e implementação do projeto desenvolvido sobre padrões de sistemas abertos como o CORBA (*Common Object Request Broker Architecture*).

#### **1.4. Estrutura da Dissertação**

O texto encontra-se organizado em 6 capítulos. O capítulo inicial apresenta uma introdução aos temas abordados nos demais capítulos. O segundo capítulo introduz e descreve os conceitos envolvidos no trabalho. O capítulo três descreve a especificação e o projeto do sistema de telemedicina. O quarto capítulo descreve a fase de implementação do sistema, os testes e resultados finais obtidos com o trabalho. O capítulo cinco expõe uma avaliação geral do trabalho. O último capítulo contém as conclusões finais do trabalho e os possíveis trabalhos futuros.



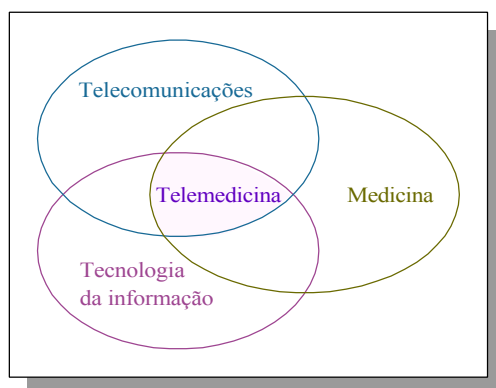
## 2. Estado-da-Arte e Conceituação

Este capítulo contém uma introdução de cada área envolvida no tema escolhido para o desenvolvimento do projeto e dissertação de mestrado. O tema escolhido abrange diversas disciplinas, sendo necessário descrever os conceitos básicos de cada uma a fim de estabelecer a ligação entre as mesmas de forma a compor o tema.

### 2.1. Telemedicina

A Telemedicina pode ser definida de muitas formas, sendo que neste trabalho adotamos a seguinte definição:

**Intercâmbio de informações para o suporte a prática médica à distância, por meio eletrônico, quando há necessidade por uma das partes. (Adaptado de [Field97])**



A Telemedicina encontra-se na interseção de áreas como telecomunicações, Tecnologia da Informação e medicina. Através das tecnologias da informação e telecomunicações a prática médica à distância torna-se viável.

Figura 1: Definição de Telemedicina.

O objetivo da telemedicina é levar às localidades remotas (i.é. dispersas geograficamente) informações de natureza médica, quando estas informações são necessárias.

Entre os benefícios oferecidos pela telemedicina podemos citar [Field97]:

- \* **O incremento do acesso a serviços de medicina:**

Ampliando o alcance dos serviços médicos, através da distribuição destes serviços em áreas antes desprovidas de tais recursos. O acesso ao tratamento especializado é expandido;

- \* **A melhora na qualidade de serviços:**

Facilitando a interação cooperativa entre especialistas situados remotamente

no atendimento de pacientes, oferecendo a possibilidade de encontros simultâneos entre o paciente, seu médico e um especialista remoto;

\* **A disseminação de conhecimento:**

Obtida através do contato entre médicos e especialistas em áreas determinadas para a troca de experiências e conhecimento;

\* **A redução de custos:**

Alcançada através da diminuição da quantidade de viagens necessárias no tratamento de um paciente e à melhora da qualidade do diagnóstico, agora assistido por diversos especialistas se necessário. Como consequência há a melhora no tratamento médico e a diminuição da quantidade de consultas e tratamentos.

Podemos classificar as aplicações de telemedicina segundo três níveis:

\* **Aplicações internas aos hospitais:**

Relacionadas à transferência de informação entre departamentos;

\* **Aplicações entre hospitais:**

Relacionadas à comunicação entre especialistas de hospitais diferentes como por exemplo a teleconferência;

\* **Aplicações entre pacientes externos e o hospital:**

Como por exemplo monitoração de pacientes em suas próprias residências, eliminando a necessidade de deslocamento até o hospital.

A telemedicina engloba diversas aplicações como por exemplo aplicações educacionais, de pesquisa, atendimento à pacientes ou administrativas. Conforme a modalidade da aplicação podemos subdividi-la em outras áreas específicas, como teleradiologia, telepsiquiatria, telecirurgia, telemonitoramento ou teleconsulta [Bai97]. A teleradiologia trata de aplicações ligadas ao diagnóstico em radiologia, a telepsiquiatria do atendimento psiquiátrico, a telecirurgia de procedimentos cirúrgicos assistidos à distância, a teleconsulta de consultas médicas remotas e o telemonitoramento do monitoramento remoto de pacientes.

Neste trabalho há um interesse particular na **teleradiologia**. Podemos defini-la como a transmissão eletrônica de imagens radiológicas de um local para outro com o propósito de interpretação ou consulta [ACR Standards 99].

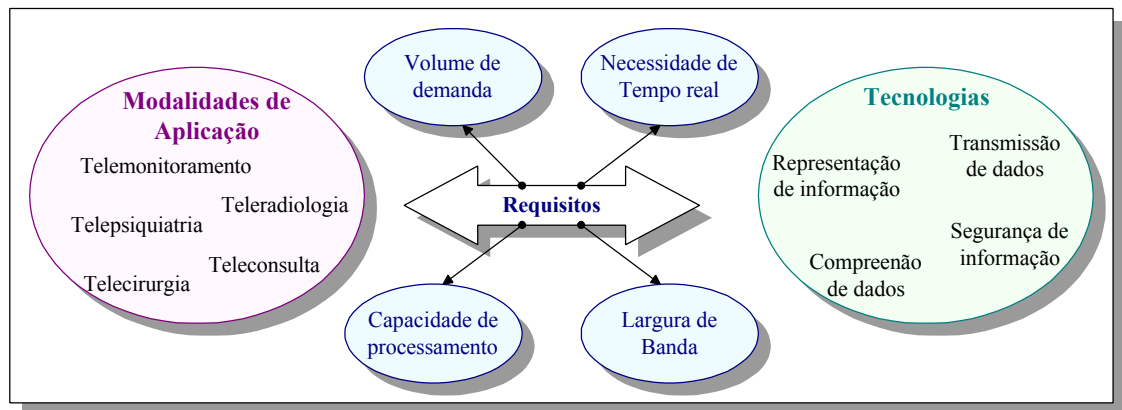


Figura 2: Modalidades e requisitos.

Cada tipo de aplicação possui requisitos diferentes que dizem respeito ao formato da informação transportada, largura de banda em rede, quantidade de processamento para refinamento da informação, necessidade de interação em tempo real ou número de usuários envolvidos. Assim, do ponto de vista técnico, a telemedicina acaba por envolver diversas áreas, como por exemplo: transmissão e compressão de dados, armazenamento e recuperação dos dados, segurança de dados médicos e pessoais e apresentação destes dados em um formato adequado.

Particularmente a telemedicina, com ênfase em radiologia, apresenta alguns requisitos especiais no que diz respeito à qualidade das imagens manipuladas. Quando usada na produção de interpretação oficial, não pode haver perdas que comprometam a interpretação clínica da imagem, desde a aquisição até a transmissão e armazenamento da mesma. Para fins apenas de exposição a qualidade deve atender a um valor mínimo de acordo com as circunstâncias. (adaptado de [ACR Standards 99]).

A apresentação adequada dos dados médicos envolve uma área denominada visualização de dados médicos (ver capítulo 2.2), onde os dados são processados em estações de trabalho e visualizados na forma de imagens 3D, 2D ou gráficos, o que constitui uma apresentação mais simples se comparada ao formato gerado originalmente, que não permite uma análise imediata.

Algumas das dificuldades encontradas ao trabalharmos com telemedicina dizem respeito à pouca disponibilidade e utilização de padrões que possibilitem o trabalho conjunto de diferentes aplicações, a difícil adaptação da infra-estrutura já existente de

tratamento médico para suporte à telemedicina e o treinamento e aceitação de novas tecnologias na rotina médica.

Outros desafios a serem vencidos pela telemedicina referem-se ao rápido ciclo de evolução das tecnologias de programação e equipamentos, que acabam por se tornar obsoletas rapidamente. Este fator acaba por forçar uma constante renovação objetivando a incorporação de novas tecnologias e aprimoramento das técnicas usadas. A utilização de soluções voltadas à reutilização e as tecnologias como orientação a objetos e componentes pode constituir uma solução para o problema de programação.

Por fim, a telemedicina requer um nível de cooperação à distância dependente de fatores humanos, como as instituições e indivíduos envolvidos no processo. A análise de seus resultados ainda depende de informações ainda não suficientes para comparação de seus efeitos no tratamento médico. É necessário realizar uma boa análise das implicações e requisitos envolvidos na área, para superarmos as dificuldades e podermos aproveitar os benefícios trazidos pela área.

### **2.1.1. Resumo da Seção**

Nesta sessão foram descritos os conceitos básicos que a telemedicina envolve, definindo-se a telemedicina como o suporte à prática médica à distância, através do uso de recursos de telecomunicações e de informática.

Além da definição de telemedicina foram discutidos os benefícios e complicações envolvidos em sua implementação e implantação. Entre os benefícios temos: redução de custos, acesso a serviços especializados em locais afastados dos grandes centros, distribuição de conhecimento e melhoria na qualidade dos serviços médicos.

Existem muitas dificuldades na implantação da telemedicina, sendo que as principais referem-se a aspectos culturais, que dificultam a aceitação de novas técnicas e tecnologias à rotina de atendimento médico e o difícil acompanhamento da evolução tecnológica, que faz com que equipamentos tornem-se obsoletos rapidamente.

Uma vez superadas as dificuldades relacionadas à área em questão, creio que os custos envolvidos compensem, se comparados aos benefícios alcançados.

## **2.2. Visualização de Imagens Médicas**

A visualização de imagens médicas insere-se no contexto da visualização científica, em particular fazendo grande uso da visualização volumétrica.

A visualização científica baseia-se na representação de dados abstratos gerados por algum processo ou simulação e transformados em informação visual significativa, de forma que os usuários possam entender estes dados mais facilmente (adaptado de [Chen96]). A apresentação da informação é obtida através da combinação dos dados abstratos com técnicas de computação gráfica, processamento de imagens e interfaces gráficas.

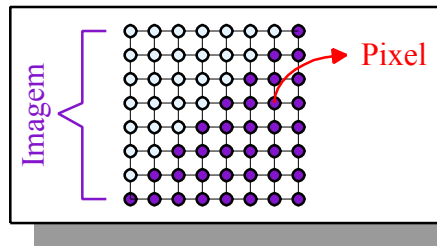
Muitas vezes os dados a serem trabalhados se referem a características tridimensionais. A visualização destes dados é denominada visualização volumétrica, um sub-campo da visualização científica, que é definido como o processo de projeção de um conjunto de dados multidimensionais em um plano bidimensional na forma de uma imagem, com o objetivo de entender as estruturas contidas dentro do dado volumétrico [Elvins92]. Aqui o conjunto de dados tridimensional pode assumir diversas características, como por exemplo múltiplos valores para cada ponto da grade de dados.

A visualização de imagens médicas é uma área da visualização científica, onde os dados a serem estudados provém de fontes médicas, como exames radiológicos ou bases de dados hospitalares. Em particular neste trabalho há um interesse na visualização de dados radiológicos de Tomografia Computadorizada (CT - *Computerised Tomography*) e Ressonância Magnética (MRI - *Magnetical Ressonance Imaging*). Os dados gerados a partir destas fontes são volumétricos, sendo necessário aplicar as técnicas de visualização volumétrica para gerar descrições de compreensão mais simples.

### **2.2.1. Modelo de Representação de Dados**

A visualização científica envolve alguns conceitos básicos descritos neste item. Entre estes conceitos temos a imagem, o *pixel*, o volume e o *voxel*.

Uma imagem discreta é uma descrição de dados mapeados em uma grade bidimensional, feita através de uma matriz de pontos denominados *pixels*.

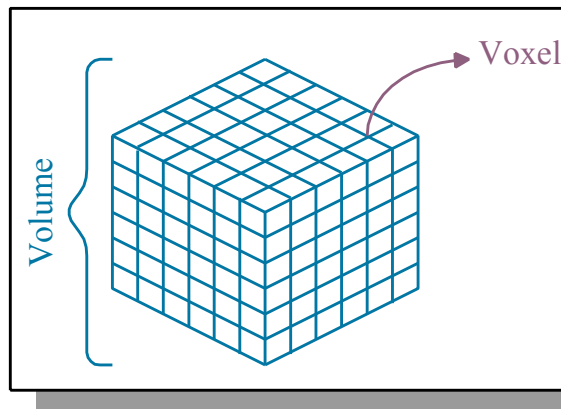


Cada posição da matriz corresponde a um ponto denominado *pixel*.

Uma imagem é uma matriz de *pixels*, onde o valor de cada *pixel* pode corresponder a um índice em uma tabela de cores do sistema, ou aos valores RGB (*Red Green Blue*) da cor.

Figura 3: Modelo de imagem.

Neste trabalho, um volume discreto é uma coleção de dados com características tridimensionais, mapeados sobre uma grade onde cada ponto desta grade constitui um *voxel*. Desta forma um *voxel* caracteriza-se por uma coordenada espacial e um valor. A representação espacial usualmente utilizada em volumes é a cartesiana homogênea.



O *voxel* é a unidade básica do volume, sendo uma amostra no espaço de um determinado ponto do volume.

Figura 4: Modelo de volume

O processamento requerido para visualização científica geralmente é elevado, devido a características como grande volume de dados ou necessidade de processamento intensivo. Para que a visualização científica se torne mais acessível é necessário combina-la com técnicas que possibilitem a obtenção de resultados de boa qualidade em um tempo aceitável. Uma possibilidade é a utilização de processamento paralelo e distribuído, para obtenção de um melhor desempenho da aplicação de visualização.

### 2.2.2. Resumo da Seção

A visualização científica é a representação de dados abstratos gerados por algum processo computacional em representação visual de mais simples compreensão, representação esta obtida através da combinação de dados abstratos com técnicas de computação gráfica, processamento de imagens e interfaces gráficas.

Neste trabalho há um interesse particular em visualização de imagens médicas que é

uma sub-área da visualização científica, onde os dados a serem visualizados são obtidos de fontes médicas, como exames de radiografia, CT (Tomografia Computadorizada), MRI (Ressonância Magnética), etc.

A visualização de imagens médicas faz grande uso da visualização volumétrica, uma sub-área da visualização científica. Nesta subterra, o conjunto dos dados apresentam características multidimensionais, sendo estes projetados em um plano bidimensional na forma de uma imagem.

### **2.3. Aplicações Distribuídas**

Uma aplicação distribuída é um sistema cujos módulos encontram-se distribuídos entre um ou mais nós de uma rede. O conjunto dos nós realiza uma tarefa paralelamente ou não, sendo necessária a comunicação entre eles.

Enquanto que aplicações seqüenciais executam as ações em uma determinada ordem, as aplicações distribuídas, por terem sido particionadas entre os diversos nós da rede, necessitam de um esforço extra para sincronização das tarefas de cada nó, controle de dados compartilhados ou mesmo localização de cada pedaço da aplicação.

A manutenção da localização das partes da aplicação é importante, devido a necessidade de comunicação entre módulos distribuídos. Muitas vezes, por motivos de desempenho, os dados são replicados. A troca dos dados pode ser custosa, uma vez que depende de características como velocidade, carga e tempo de resposta da rede. Assim, surge a necessidade de gerenciar a coerência dos dados replicados, pois se em um determinado momento um dos módulos altera um dado replicado, os demais módulos deverão ser notificados para que atualizem suas respectivas réplicas.

Por último, é necessário implementar algum mecanismo de sincronização, pois algumas tarefas podem ser dependentes de outras, sendo necessária a imposição de uma determinada ordem de execução.

A comunicação em aplicações distribuídas pode ser feita de diversas formas: por meio de um trecho de memória compartilhada por todos os pedaços da aplicação, via troca de

mensagens, mecanismos de acesso como RPC<sup>1</sup> (*Remote Procedure Call*), ou chamada de métodos remotos (no caso de objetos distribuídos). A escolha entre os mecanismos para implementação da aplicação depende das necessidades envolvidas e das ferramentas disponíveis para a construção da mesma.

#### **2.4. CSCW (*Computer Supported Cooperative Work*) e Groupware**

Com a intenção de atingir objetivos em comum profissionais procuram interagir e produzir em conjunto. A evolução dos computadores e sua integração em ambientes profissionais, proporcionou o surgimento de uma nova tecnologia que poderia ser aplicada como solução de muitos problemas encontrados no trabalho em equipe. Os serviços, antes voltadas a automação de tarefas manuais, ganharam novas abordagens onde se cria todo um ambiente baseado em informações armazenadas e processadas em computadores, sendo essas informações compartilhadas entre diversas pessoas. Dentro deste contexto surgiu a pesquisa na área de CSCW (*Computer Supported Cooperative Work*).

Existem muitas definições para o termo CSCW e segundo [Ellis91], o CSCW trata-se de um sistema organizacional que integra o processamento de informações e atividades de comunicação, buscando-se analisar o trabalho em grupo do ponto de vista de como as tecnologias, em especial os computadores, possam auxiliar este trabalho. A pesquisa de CSCW é essencialmente multidisciplinar, envolvendo disciplinas de educação, psicologia, sociologia e ciência da computação.

Um segundo termo, *groupware*, é muitas vezes utilizado como sinônimo de CSCW, mas segundo [Ellis91] o *groupware* é definido como o sistema baseado em computadores, com suporte ao trabalho em grupo entre pessoas que desempenham uma tarefa em comum ou buscam um mesmo objetivo, através de uma interface para o compartilhamento de um ambiente.

O *Groupware* possui muitas definições e segundo *David Coleman*: “Se 20 especialistas em *Groupware* forem reunidos em uma sala teremos 20 definições diferentes”

---

<sup>1</sup>RPC: Mecanismo, especificado pelo DCE (Distributed Computing Environment), onde procedimentos em um programa podem ser executados em máquinas remotas. O DCE é uma camada de serviços padronizada que permite que aplicações distribuída comuniquem-se com uma coleção de computadores, sistemas operacionais e arquiteturas de redes diferentes. [capítulo 8 referência 7]



[Orfali99]. Apesar das diversas definições atribuídas ao CSCW e ao *Groupware*, geralmente encontramos o termo CSCW referenciando a pesquisa e o termo *groupware* referenciando as tecnologias usados no trabalho cooperativo.

Quando utilizados apenas para automatização de um sistema ineficiente, o *Groupware* e o *CSCW* apenas adicionam mais um nível de complexidade ao sistema. Como sabemos a implantação de qualquer nova tecnologia e sistema requer a reestruturação de processos já existentes e da forma como o trabalho é organizado e realizado. Apesar do aparente custo elevado de uma reestruturação, o CSCW e o *Groupware* trazem diversos benefícios como:

- \* Transformação de uma instituição através da mudança na forma de comunicação entre pessoas e como resultado obtém-se uma mudança no processo de negócios [Orfali99];
- \* Possibilita aos participantes trocar idéias e sincronizar seu trabalho, mantendo “registro” da “memória coletiva” do grupo;
- \* Automatização de tarefas;
- \* Facilita a comunicação;
- \* Facilita o gerenciamento do trabalho em grupo.

Há uma grande variedade de sistemas de CSCW, sendo que eles podem ser classificados segundo duas categorias básicas [Rodden92]:

\* **Forma de cooperação:**

A natureza da cooperação é distinguida segundo a forma com que as pessoas interagem. Quando a interação requer a presença de todas as pessoas envolvidas em um grupo ela é denominada síncrona, enquanto a cooperação assíncrona não necessita da presença simultânea de todos os membros do grupo, podendo ocorrer durante um longo período de tempo;

\* **Natureza geográfica:**

Considera a distribuição geográfica dos participantes de um grupo de trabalho.

	Simultâneo	Não simultâneo
Mesmo local	Interação síncrona (face-a-face)	Interação assíncrona
Locais diferentes	Interação síncrona distribuída	Interação assíncrona distribuída

[Ellis91] sugere uma taxinomia de tempo-espaço para o estudo do *Groupware*. Para ele um bom sistema *Groupware* deve atender a todos os quadrantes do gráfico ao lado.

Figura 5: Classificação no tempo-espaço do Groupware [Ellis91].

Finalizando a introdução sobre CSCW e *Groupware* vale a pena lembrar que a pesquisa nesta área acaba envolvendo outras áreas como sistemas distribuídos, redes de computadores, segurança de dados, interação homem-máquina e teoria social. A área de redes está ligada à necessidade de interação entre usuários remotos, a área de sistemas distribuídos esta ligada à necessidade de distribuição de tarefas e interação entre as diversas partes de um grupo. A área de segurança é necessária para manter o sigilo dos dados manipulados no trabalho em grupo, enquanto a teoria social diz respeito ao estudo do comportamento dos grupos, para melhor projeto do sistema de *Groupware*.

## 2.5. Arquiteturas Cliente Servidor

No modelo cliente/servidor, os serviços são colecionados e instanciados na forma de servidores distribuídos pela rede de computadores. Os clientes são programas que solicitam estes serviços em um dado momento da execução.

Um servidor basicamente espera por chamadas em um laço infinito. O cliente executa suas tarefas e quando necessita de algum serviço chama o servidor que recebe a requisição, realiza a tarefa e devolve os resultados ao cliente.

Um servidor pode ser iterativo, isto é, atende a apenas uma requisição por vez, ou concorrente, atendendo múltiplos clientes ao mesmo tempo [Comer93]. A implementação de um servidor iterativo é mais simples, contudo os servidores concorrentes apresentam um desempenho melhor, visto que as requisições não precisam esperar que o servidor termine uma tarefa já em andamento.

Outra forma de classificar os servidores é segundo o protocolo de transporte utilizado. Alguns servidores são orientados a conexão, outros não [Comer93].

Os servidores orientados a conexão usam protocolos como o TCP<sup>2</sup>, por exemplo. Neste caso, quando o cliente faz uma requisição, é aberta uma conexão com o servidor, e então os dados podem ser trocados entre os parceiros. O servidor não precisa preocupar-se com tratamento de erros, perdas de pacotes, segmentação da mensagem ou ordenação dos pacotes de rede. Tudo isto é tratado no nível da camada de transporte do protocolo de rede utilizado. A desvantagem deste tipo de servidor é a existência de um tempo extra, associado ao estabelecimento da conexão, além de ser necessário uma porta (um *socket* por exemplo) para cada conexão.

Os servidores não orientados a conexão usam protocolos como UDP<sup>3</sup>, por exemplo. Neste tipo de servidor, o cliente não precisa abrir uma conexão com o servidor, basta mandar as mensagens pela rede e endereça-las ao servidor. A responsabilidade do gerenciamento das mensagens é da própria aplicação, que deve verificar os pacotes perdidos, a segmentação das mensagens e a ordenação dos pacotes. No caso de perdas, o cliente deve retransmitir a mensagem. A comunicação entre cliente e servidor necessita de um protocolo que assegure a integridade das mensagens, pois no caso não orientado a conexão a probabilidade de perdas é maior, já que a camada de transporte do protocolo utilizado não cuida destes aspectos. Uma vantagem deste tipo de abordagem é que torna-se mais fácil implementar o serviço de distribuição (*broadcast*).

A última forma de classificar um servidor é segundo a manutenção do estado de suas conexões [Comer93]. Assim, um servidor pode ser *statefull*, isto é, mantém informações a respeito de suas conexões, ou *stateless*, que não mantém informações sobre suas conexões. O uso de protocolos não orientados a conexão aumenta a necessidade de criação de servidores *statefull*, pois o gerenciamento da recepção dos pacotes está totalmente ao cargo do servidor.

---

<sup>2</sup> **TCP (Transmission Control Protocol):** Protocolo de comunicação, orientado a conexão. Garante a entrega dos dados pela rede, ordenados e sem erros, possuindo mecanismos de detecção e correção de erros.

<sup>3</sup> **UDP (User Datagram Protocol):** Protocolo alternativo ao TCP, não orientado a conexão, responsabilidade da aplicação o detecção e tratamento de erros e a entrega dos dados.

## 2.6. *Objetos Distribuídos*

Um sistema baseado em objetos distribuídos é essencialmente um sistema cliente/servidor. Objetos solicitam serviços uns aos outros através de chamadas aos métodos. A execução de um método pode ser local ou remota, sendo a localização do objeto chamado transparente ao objeto que realiza a chamada. As chamadas aos métodos são tipicamente síncronas, isto é, o fluxo de execução do programa é interrompido até o retorno da chamada.

Alguns conceitos que devem ser mantidos em mente quando se trabalha com objetos distribuídos são:

- \* **Os objetos podem residir em qualquer nó da rede:**

Os objetos que interagem entre si podem estar situados no mesmo nó ou em nós distintos da rede, sem que sua interatividade seja afetada.

- \* **Cada objeto é identificado de forma única:**

O conceito de objetos pressupõe a existência de um mecanismo de identificação. No caso de objetos distribuídos, este mecanismo tem importância duplicada, pois através dele que se torna possível referenciar os objetos, sejam eles remotos ou não. A identificação do objeto é feita através de identificadores (IDs) únicos em todo o sistema distribuído;

- \* **Transparência de localidade:**

Ao referenciar um objeto, a aplicação não precisa saber se este objeto é local ou remoto, sendo sua localização transparente ao mecanismo de referência;

- \* **Chamadas aos métodos remotos:**

Um objeto pode invocar os métodos de outro independentemente de localização. O mecanismo de chamada aos métodos é semelhante às chamada do tipo RPC (*Remote Procedure Call*), porém estendida para os conceitos de orientação a objetos.

O mecanismo de identificação acrescenta um nível indireto de acesso, pois toda referência aos objetos é feita por intermédio dos IDs únicos, que passam por um sistema capaz de descobrir se a referência é remota ou não. Através deste esquema consegue-se manter a transparência de localidade dos objetos.

Quando um objeto deseja referenciar a outro, deve ser feita a negociação entre eles. A negociação (*binding*) é um processo de estabelecimento de comunicação entre os parceiros. Esse é feito por um sistema de tempo de execução (*runtime*) ou *kernel*<sup>4</sup>, que deve verificar a localização dos objetos envolvidos, o protocolo a ser utilizado e os parâmetros passados. Este processo é feito dinamicamente, permitindo que os objetos possam migrar de um nó para outro na rede. Ao migrar, um objeto precisa apenas refazer a negociação para continuar a trabalhar.

Em sistemas não distribuídos a negociação é feita na compilação ou na ligação (*link*). É o caso estático, onde os parceiros já conhecem a localização um do outro, e é denominado negociação prévia (*early binding*). A negociação prévia permite a otimização e o aumento do desempenho da aplicação.

Quando os parceiros não conhecem a localização um do outro, a negociação deve ser feita dinamicamente. Este processo é denominado negociação tardia (*late binding*). A negociação tardia é mais flexível, porém requer tempo adicional na execução relacionado ao estabelecimento de comunicação entre parceiros.

Os servidores disponíveis na rede devem ser registrados em um servidor de nomes. Quando um objeto desejar utilizar um serviço, fará consulta ao servidor de nomes fornecendo o nome do serviço desejado. O servidor então fornece o endereço do serviço ao cliente que então pode fazer a negociação. O registro no servidor de nomes é feito através de uma chamada do tipo exportação (*export*), enquanto que a requisição por nome de um servidor é feita através de uma chamada do tipo importação (*import*).

O mecanismo descrito acima funcionaria muito bem se considerássemos um sistema homogêneo, em que todos os computadores da rede utilizassem mesmas plataforma de máquina e programas. Porém, a realidade das redes costuma estar um tanto distante deste cenário. No item a seguir, abordaremos as principais características de um sistema heterogêneo.

### **2.6.1. Sistema Heterogêneos**

A comunicação entre objetos em um sistema distribuído e heterogêneo depende

---

<sup>4</sup>sistema de tempo de execução ou *Kernel*: sistema que faz a mediação entre os parceiros, em tempo de execução, para estabelecimento de conexão e manutenção da mesma.

do estabelecimento de padrões que permitam a comunicação entre estes objetos de forma a possibilitar o acesso irrestrito de uns aos outros, em qualquer combinação de plataformas de sistemas operacionais ou máquina.

O cenário envolvido em um sistema heterogêneo de objetos distribuídos é complicado. Há uma variedade muito grande de ferramentas e protocolos incompatíveis. A comunicação entre estas aplicações depende de algum mecanismo que seja capaz de eliminar as diferenças entre as aplicações (por exemplo APIs-*Application Programming Interface* e/ou protocolos padronizados).

Neste cenário, devemos nos preocupar não só com a localidade, sincronização e coerência dos dados entre os objetos; há outros fatores, como tipos de dados, diferença entre o desempenho de cada plataforma e até mesmo a ordem dos *bytes* de uma palavra transmitidos de um nó para outro.

As diferenças de desempenho entre plataformas pode ser causada por otimizações geradas por programas diferentes ou por diferenças de circuitaria. Essa diferença de desempenho pode prejudicar a aplicação global, pois enquanto um dos nós finaliza sua parte da tarefa rapidamente, a aplicação deve esperar um outro nó mais lento. Neste caso é necessário a adoção de um esquema de balanceamento de carga entre os objetos.

Como cada objeto pode residir em uma plataforma diferente, a representação dos dados pode não coincidir. Por exemplo, o tipo inteiro pode ser representados por 4 bytes em um nó e por 8 em outro.

Outro problema é a representação das palavras em cada nó. Enquanto um dos nós pode representar os dados na memória como *Little endian*<sup>5</sup> outro pode representar os dados como *Big endian*. Uma alternativa para este problema, deve existir um formato comum de representação. Quando cada nó recebe dados via rede, estes dados deve ser convertido para o formato usado pelo nó.

Na tentativa de contornar estes problemas e muitos outros, um grupo de mais de 500 empresas foi constituído para estabelecimento de um padrão de objetos distribuídos. Este grupo foi batizado como OMG (*Object Management Group*) e seu objetivo é a

---

<sup>5</sup> *Little endian* e *Big endian* refere-se a representação de um dado na forma de bytes, onde podemos ter o byte mais significativo à direita ou esquerda.

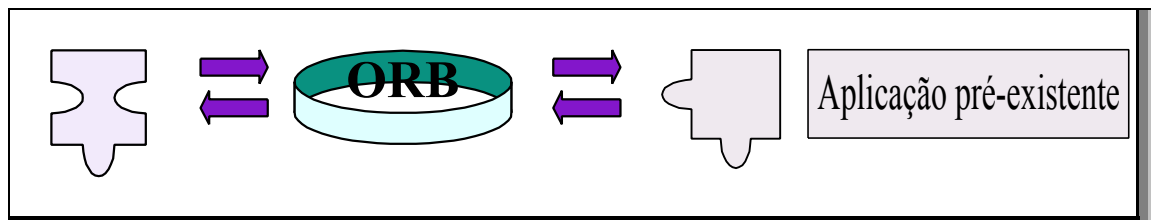
criação de uma infra-estrutura aberta para objetos distribuídos. Como resultado desta reunião entre empresas, surgiu a especificação do CORBA (*Common Object Request Broker Architecture*), que será descrito no próximo item.

## 2.7. CORBA (*Common Object Request Broker Architecture*)

Criado pela OMG (*Object Management Group*), um consórcio de empresas da área de informática, o CORBA é a especificação de uma arquitetura aberta para integração de sistemas de objetos distribuídos. O CORBA faz parte da OMA (*Object Management Architecture*), um conjunto de padronizações que oferece um ambiente para construção de aplicações distribuídas.

Os membros da OMG acreditam que a orientação a objetos é a melhor solução para seus objetivos, o desenvolvimento de programas de forma modular promovendo o reúso de código, a integração de linhas de desenvolvimento diferentes, integração de plataformas de máquinas e programação diferentes e facilidade de manutenção do código. A especificação do CORBA não abrange detalhes de codificação, mas apenas as interfaces e serviços a serem oferecidos pela implementação.

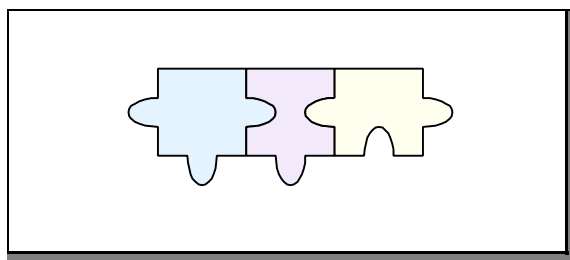
Entre os objetivos da arquitetura CORBA podemos citar:



### **Preexistência de aplicações:**

Aplicações já existentes (aplicações legadas) podem ser encapsuladas de forma a continuarem a trabalhar em conjunto com o novo sistema.

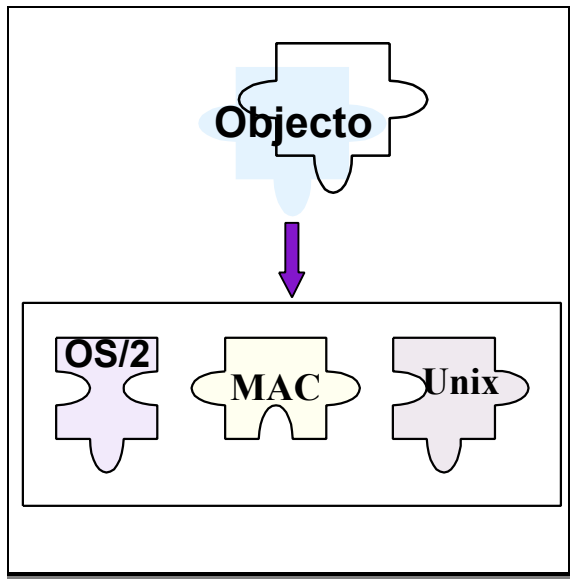
Figura 6: Preexistência



### **Criação de componentes *plug-and-play*:**

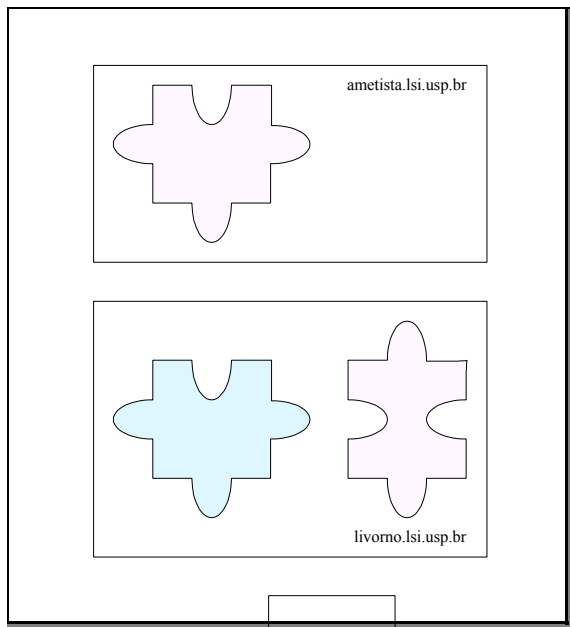
Componentes são criados de forma a serem incorporados às aplicações como *plugins*;

Figura 7: Componentes *plug-and-play*

**Portabilidade:**

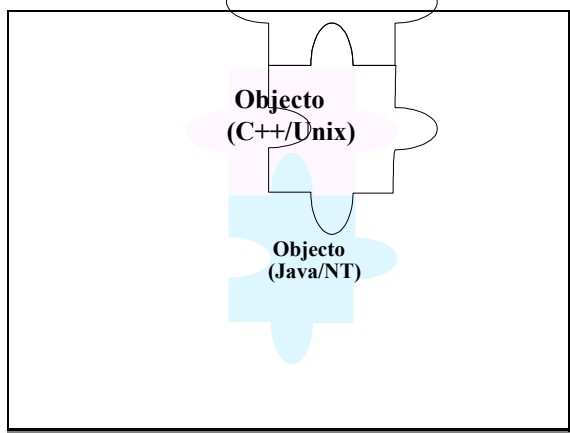
Os objetos podem trabalhar sobre qualquer plataforma de máquina ou sistema operacional, sendo os detalhes de nível mais baixo escondidos da aplicação;

Figura 8: Portabilidade

**Transparência de localidade:**

Objetos interagem independentemente de localização no sistema distribuído.

Figura 9: Transparência

**Interoperabilidade:**

Objetos podem interagir independentemente da plataforma de circuitaria, sistema operacional e linguagem de programação, sendo a interação entre eles transparente.

Figura 10: Interoperabilidade

A arquitetura de objetos do CORBA é uma arquitetura cliente/servidor, onde um cliente



é uma entidade que realiza uma operação em uma implementação de objeto que comporta-se como o servidor. Uma implementação de objeto (*object implementation*) é o conjunto de código e dados que implementam o objeto.

O termo componente (*Component*) é muito utilizado quando trabalhamos com a arquitetura CORBA. Podemos definir componente como um conjunto de objetos que colaboram entre si, para a realização de tarefas do sistema. Um componente não constitui uma aplicação por si só, contudo podem ser agrupados de forma a constituir uma aplicação ou agregados a aplicações já existentes de forma a complementá-las. Os componentes não constituem necessariamente um objeto, porém podem apresentar características de objetos suportando herança, polimorfismo e encapsulação. As características mínimas que definem um componente são:

- \* Um componente é um pedaço de código binário independente que pode ser agregado à aplicação;
- \* Não constitui uma aplicação completa, mas podem ser combinados de forma a constituir uma aplicação, sendo que cada componente realiza uma determinada tarefa.
- \* Podem ser usados em qualquer combinação. Tipicamente são usados em conjunto com outros componentes da mesma família, segundo uma abordagem *plug-and-play*.
- \* Possui uma interface bem definida que constitui a forma pela qual o componente apresenta suas funcionalidades ao mundo exterior. O componente só pode ser manipulado através desta interface.
- \* Comporta-se como um objeto que pode ser chamado através de espaços de endereçamento, redes, linguagens, sistemas operacionais ou ferramentas. É uma entidade de programação independente.

Os componentes, construídos sobre a estrutura do CORBA, interagem através de suas interfaces. A interface de um objeto deve ser descrita em uma linguagem denominada IDL (*Interface Definition Language*). Esta linguagem fornece uma forma padronizada de descrição das interfaces dos componentes para seus clientes. É uma linguagem declarativa não especificando detalhes de implementação e apenas oferecendo uma forma de definição de APIs (*Application Program Interfaces*).

Um método definido em IDL pode ser implementado e utilizado em qualquer linguagem que possui estrutura de negociação do CORBA, sendo que a descrição das classes em IDL permite a interação entre objetos implementados em qualquer linguagem que possua o mapeamento IDL.

O mapeamento entre linguagens (*language mapping*) especifica como uma definição em IDL deve ser traduzida para a linguagem de implementação dos objetos. O mapeamento da IDL para as linguagens usadas na implementação de objetos é padronizado pela OMG, que já definiu os mapeamentos para as linguagens JAVA, C, C++, COBOL, smalltalk e ADA. Existem mapeamentos para outras linguagens, porém ainda não são padronizados pela OMG.

Outro ponto importante que garante a interoperabilidade entre objetos são as referências a objetos. Uma referência a objeto (*Object Reference*) é um identificador único usado para distinguir o objeto dentro do sistema distribuído. A referência a objeto contém as informações necessárias à ORB para o estabelecimento da comunicação entre objetos.

A interoperabilidade entre ORBs (*Object Request Broker*) diferentes deve ser garantida pela IOR (*Interoperable Object References*), a forma padronizada de uma referência a objeto, usada para passar as referências através de diferentes implementações de ORB. A IOR independe de protocolo da camada de comunicação, uma vez que contém informações específicas aos protocolos suportados, permitindo a fácil adição de novos protocolos de comunicação sem a necessidade de reconstrução da aplicação. No caso do TCP/IP, por exemplo, a IOR contém o endereço da máquina, o número da porta e a chave do objeto (*object key*).

O CORBA faz parte da OMA (*Object Management Architecture*), uma arquitetura de objetos distribuídos proposta pela OMG. A seguir veremos uma descrição mais detalhada desta arquitetura.

### **2.7.1. OMA (*Object Management Architecture*)**

Esta sessão introduz os conceitos da arquitetura de objetos distribuídos proposto pela OMG. Muitos termos serão mantidos em inglês por referenciar nomes de pedaços do sistema.

A arquitetura CORBA proposta pela OMG possui os seguintes elementos:

\* **ORB (*Object Request Broker*):**

Implementa o núcleo do CORBA, sendo responsável pelo encaminhamento das requisições entre objetos de forma transparente, além de oferecer uma variedade de serviços para manipulação de objetos distribuídos;

\* **Common Object Services, ou CORBAservices:**

Coleção de serviços de nível de sistema acessados via interface definida em IDL. Os serviços CORBA ampliam a funcionalidade da ORB, permitindo a criação e a disponibilização dos objetos no sistema distribuído;

\* **Common Facilities, ou CORBAfacilities:**

Coleção de objetos que oferecem serviços diretamente aos objetos da aplicações;

\* **Application Objects:**

São os objetos de nível mais alto dentro da estrutura do CORBA (*Business objects*), sendo os consumidores finais dos serviços oferecidos pela infraestrutura do CORBA.

Os Objetos de Negócios (*Business Objects*) descrevem conceitos independentes da aplicação, tentando imitar o mundo real. Exemplos de Objetos de Negócios são: paciente, imagem, volume, entre outros.

Os Objetos de Serviço (*Object Services*) constituem um pacote de objetos, na forma de componentes, que desempenham serviços no nível de sistema e que possuem interfaces específicas em IDL. Eles são um complemento dos serviços da ORB, possibilitando a criação, a nomeação e a introdução de um objeto ao ambiente.

O CORBA oferece 15 tipos diferentes de serviços básicos:

\* **Naming Service:**

Permite a localização de objetos através de nomes associados aos mesmos. Os nomes formam uma hierarquia que pode ser comparada a uma estrutura de diretórios em um sistema de arquivos. Dentro desta estrutura são criados contextos de nomes (*Naming Context*), que podem ser comparados aos diretórios, sendo os nomes finais associados ao objeto. Um objeto é identificado através do caminho completo, desde a raiz até o nó final;

\* **Externalization Service:**

Através do *Externalization Service* é possível fazer a leitura e escrita de dados em um componente através de um mecanismo tipo *stream*. Define protocolos e convenções para gravação do estado de objetos em arquivos de dados (em memória, disco ou através de redes), para que então possam ser absorvido por um novo objeto no mesmo processo ou em um processo diferente. O objeto salvo pode existir por tempo indeterminado, podendo ser transportado através de ORBs diferentes.

\* **Persistent Object Service (POS):**

Oferece uma interface para armazenamento dos objetos em servidores, que podem ser servidores de banco de dados ou simples arquivos;

\* **Events Service:**

Oferece um canal (*event channel*) de captura e distribuição de eventos entre componentes. Os componentes interessados em eventos específicos devem se registrar dinamicamente para recepção do evento. O serviço de eventos é distribuído, não sendo necessária a existência de um servidor centralizado.;

\* **Life Cycle Service:**

Define operações para criação, cópia, movimentação e remoção de componentes. Por se tratar de um ambiente distribuído, o serviço de *Life Cycle* define serviços e convenções que permitem ao cliente realizar operações de ciclo de vida em objetos remotos;

\* **Transactions Service:**

Oferece o protocolo de duas fases para controle de transações do tipo aninhadas (*nested*) ou simples (*flat*). Permite a comunicação entre sistemas diferentes para que um serviço de transações possa interagir com outros sistemas de transações cooperativos usando ORBs diferentes;

\* **Properties Service:**

Oferece operações para associação dinâmica de valores (*named values*) ou propriedades a qualquer objeto, como por exemplo um título ou uma data;

\* **Query Service:**

Permite que consultas sejam realizadas em coleções de objetos. Uma consulta (*query*) é um texto declarativo composto por predicados e valores de atributos.

O serviço de consultas (*query service*) permite a criação de mecanismo de índices, semelhante aos encontrados nos sistemas de bancos de dados e outros sistemas usados no armazenamento de grandes coleções de objetos. É baseado em padrões já existentes como SQL3 (*Simple Query Language*) e OQL (*Object Query Language*), este último especificado pela ODMG (*Object Database Management Group*);

\* **Concurrency Control Service:**

Permite a múltiplos clientes gerenciar o acesso a recursos compartilhados, de forma a manter um estado coerente.

\* **Relationship Service:**

Através deste serviço componentes podem ser associados dinamicamente, através de representação de entidades e relacionamentos. Para o CORBA, uma entidade é representada por um objeto. Relacionamentos e papéis (*relationships* e *roles*) são representados como objetos, sendo um papel (*role*) um objeto CORBA pertencente a um relacionamento. Restrições de tipo e cardinalidade podem ser expressadas e checadas, sendo geradas exceções no caso de violação de alguma restrição;

\* **Object Collections Service:**

Uma coleção é um grupo de objetos cujo relacionamento está ligado ao comportamento dos objetos e das operações suportadas, ao invés de seus tipos. O objetivo deste serviço é prover uma forma uniforme de criação e manipulação das coleções mais comuns. Exemplo de coleções são filas, pilhas, listas ou árvores.

\* **Time Service:**

Este serviço oferece recursos para sincronização dentro do sistema de objetos distribuídos, oferecendo também operações para definição e gerenciamento de gatilhos de eventos (*time-triggered events*). Permite a verificação de ordem de ocorrência de eventos, geração de eventos baseados em tempo através de *timers* e alarmes e o cálculo do intervalo entre dois eventos;

\* **Security Service:**

Oferece mecanismos para a segurança em um sistema de objetos distribuídos, suportando autenticação, confidência, controle de acesso, aceitação (*no*

*repudiation*) e distribuição de credenciais entre objetos;

\* **Object Trader Service:**

Oferece um mecanismo de verificação de serviços para os objetos. Um Provedor de Serviços registra os serviços disponíveis através de invocação e exportação de operações no corretor (*trader*). Informações sobre o serviço são passadas como parâmetros. Através de uma referência a objeto, que é passada na exportação do serviço, o cliente pode invocar operações no serviço anunciado.

\* **Licensing Service:**

Oferece mecanismos para medição de uso de componentes, permitindo a tarifação sobre seu uso. Suporta qualquer modelo de controle de uso, como tarifação por sessão, por nó, por instância ou *site*.

A Figura 11 representa a arquitetura OMA proposta pela OMG:

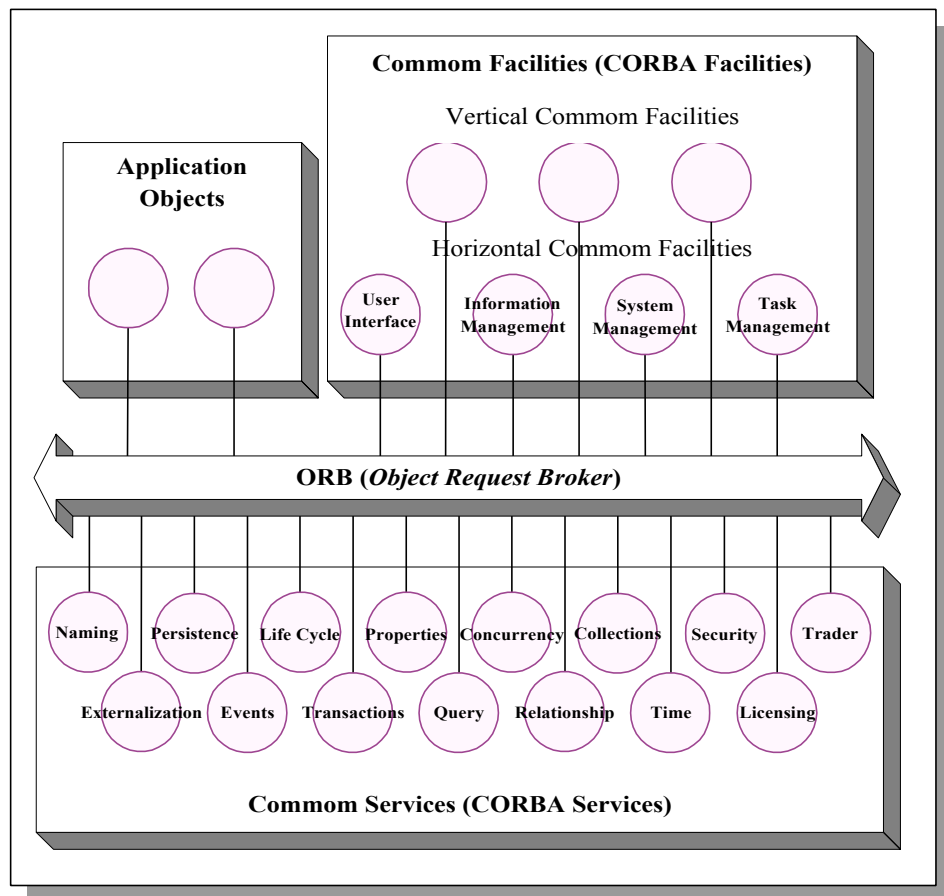


Figura 11: Arquitetura OMA (*Object Management Architecture*) [Orfali96].

As *Common Facilities* formam um conjunto de componentes definidos em IDL que oferecem serviços direto aos objetos de aplicação, sendo divididas segundo duas categorias: *Vertical CORBAfacilities* e *Horizontal CORBAfacilities*. A primeira trata de funcionalidades ligadas a domínios de aplicações específicos, como medicina (CORBAMED) ou telecomunicações, enquanto a segunda trata de serviços de uso geral, como interface de usuário (*User Interface*), gerenciamento de tarefas (*Task Management*), Gerenciamento de informações (*Information Management*) e Gerenciamento de Sistemas (*System Management*).

O terceiro elemento básico da arquitetura OMA é a ORB, que constitui o coração desta arquitetura. Quando é feita uma referência a um objeto, a ORB é responsável por encontrar o objeto referenciado, preparar a implementação do objeto para receber a requisição e fazer a comunicação de dados entre os objetos. Este mecanismo deve funcionar independentemente da localização, de linguagem de implementação e de interface de acesso ao objeto.

No topo da infra-estrutura do CORBA estão os objetos de negócios (*Business objects*). Os objetos de negócio descrevem conceitos que independem de aplicação, como por exemplo: cliente, carro, paciente, médico, etc.

A Figura 12 mostra a estrutura da arquitetura CORBA segundo a especificação 2.2:

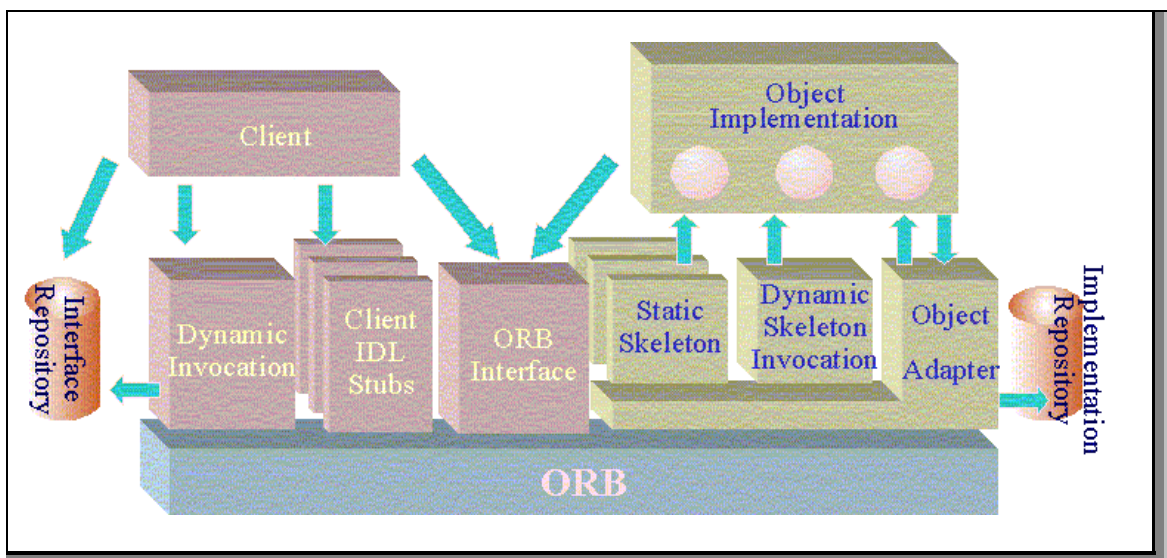


Figura 12: Arquitetura CORBA [OMG97-R2.1]

Podemos observar na figura os seguintes componentes:

\* **ORB Interface:**

A *ORB Interface* define as operações implementadas diretamente pela ORB e que podem ser aplicadas a qualquer referência a objeto. Ela provê operações padronizadas para inicialização e finalização da ORB além de operações para manipulação de objetos, como *object\_to\_string*, que converte uma referência a objeto para cadeia de caracteres e *string\_to\_object* que converte uma cadeia de caracteres para referência a objeto;

\* **Client IDL Stubs:**

Ao fazer uma requisição a uma implementação de objetos, o cliente pode utilizar dois tipos de mecanismos diferentes: o mecanismo estático e o mecanismo dinâmico. Os *stubs* fazem parte do mecanismo estático de requisição, sendo sua função descrever a interface da implementação de objetos. Os *stubs* são gerados por um compilador IDL que fornece pedaços de código, na linguagem desejada, que devem ser compilados com a aplicação. O código gerado pelo compilador IDL contém uma implementação intermediária (*dummy*) de cada método oferecido pela interface, sendo esta implementação intermediária responsável por comunicar-se com a ORB e codificar e decodificar as invocações e seus parâmetros em mensagens, que são enviadas à implementação do objeto;

\* **Static Skeletons:**

Os *Static Skeletons* constituem o lado do servidor para o mecanismo estático de interação, sendo sua função exportar a interface da implementação de objeto para o mundo externo. Assim como os *stubs* do cliente, os *Skeletons* são gerados pelo compilador IDL, que cria métodos vazios para serem preenchidos com a implementação do objeto. Após gerar os *Skeletons*, o programador pode adotar duas abordagens para preencher os métodos com o código: a primeira é por derivação da classe esqueleto gerada pelo compilador e a segunda por delegação da mesma classe;

\* **DII (Dynamic Invocation Interface):**

Quando um cliente não é compilado com os *stubs* para acesso ao servidor, ele pode utilizar o DII, que constitui o mecanismo de acesso dinâmico aos servidores. O DII permite ao cliente obter a descrição da interface de uma implementação de objetos em tempo de execução. Ao descobrir a interface do



objeto destino, o cliente deve construir uma lista de argumentos a serem passados ao método alvo, criar a requisição e então realizar a invocação do método. O mecanismo dinâmico de interação fornece máxima flexibilidade, contudo apresenta maior complexidade de implementação e maior carga de trabalho para descoberta da interface;

\* **Interface Repository:**

Trata-se de um repositório distribuído e dinâmico de interfaces IDL acessado através de uma API para manutenção (acesso, inserção, remoção e atualização) de seu conteúdo. Possibilita aos clientes encontrar objetos cuja interface é desconhecida em tempo de compilação. É usado para determinação das operações válidas em um objeto permitindo que requisições sejam feitas ao mesmo. O DII deve consultar o *Interface Repository* para obtenção das interfaces dos servidores;

\* **DSI (Dynamic Skeleton Interface):**

O DSI implementa o mecanismo de interação dinâmico no servidor. Sua função é realizar a negociação (*binding*) em tempo de execução para servidores que abrigam objetos que não possuem um esqueleto (*skeleton*) compilado. Através do DSI são verificados os parâmetros passados ao servidor e é identificado o método e o objeto alvo. Um cliente não é capaz de identificar qual o mecanismo, DSI ou *Static Skeleton*, utilizado na conexão da implementação de objetos com a ORB;

\* **Implementation Repository**

Armazena informações associadas aos servidores, como as classes suportadas por este, objetos instanciados e seus IDs, além de informações adicionais pertinentes à implementação da ORB, como informações de controle administrativo, alocação de recursos e segurança. Oferece um local comum para este armazenamento, permitindo à ORB localizar e ativar objetos em um servidor;

\* **Object adapter:**

O *Object Adapter* faz a interface entre o servidor e a ORB, sendo responsável por demultiplexar e repassar as requisições ao servidor adequado. Entre as responsabilidades do *Object Adapter* temos: gerar e interpretar referências a

objetos, invocação de métodos, segurança das interações, ativação e desativação de implementações de objetos, mapeamento das referências a objetos para a implementação correspondente e registro das implementações. Podem existir diversos tipos de *Object Adapters*, sendo eles proprietários ou não, porém para que uma implementação CORBA esteja de acordo com as especificações da OMG, ela precisa implementar pelo menos um *Object Adapter*, o BOA (*Basic Object Adapter*).

Até a versão 2.1 do CORBA o BOA era o único *Object Adapter* padronizado pela OMG, porém ele apresentava algumas ambigüidades e deficiências em sua especificação, levando os desenvolvedores a produzirem *Object Adapters* proprietários, o que resulta em pouca portabilidade. A partir da versão 2.2 do CORBA surge o POA (*Portable Object Adapter*), que permite que aplicações sejam portáveis através de ORBs de diferentes fabricantes.

Além da estrutura descrita, existem protocolos que garantem o funcionamento da mesma sobre a infra-estrutura de rede. A interoperabilidade entre ORBs diferentes é garantida por um protocolo denominado GIOP (*General Inter-ORB Protocol*). Este protocolo pode ser mapeado sobre qualquer protocolo de transporte orientado a conexão, sendo sua arquitetura constituída por 3 elementos:

\* **CDR (Common Data Representation):**

O CDR define uma sintaxe de mapeamento dos tipos de dados definidos na IDL para uma representação de baixo nível usada na transferência de dados entre ORBs e *Inter-ORB bridges* (agentes). Trata de aspecto como ordem dos *bytes* ou estabelecimento de fronteiras entre tipos de dados primitivos;

\* **GIOP Message Formats:**

Mensagens trocadas entre agentes de forma a facilitar as requisições, localizar implementações de objetos e gerenciar canais de comunicação;

\* **GIOP Transport Assumptions:**

Descreve suposições gerais a respeito de qualquer protocolo de transporte que pode ser usado na troca de mensagens GIOP, descrevendo o gerenciamento das conexões e como as mensagens GIOP são requisitadas.

O funcionamento do GIOP sobre TCP/IP é especificado pelo IIOP (*Internet Inter-ORB*

*Protocol*). O IIOP não constitui uma especificação separada do GIOP, mas sim uma especialização (ou mapeamento) do GIOP para trabalho sobre o TCP/IP. O IIOP adiciona um elemento ao GIOP, o *Internet IOP Message Transport*, que descreve como agentes abrem conexões TCP/IP e transferem mensagens através destas conexões.

A OMG especifica ainda o ESIOP (*Environment Specific Inter-ORB Protocol*), usado para operação em redes específicas. O primeiro ESIOP especificado pelo CORBA 2.0 usa o ambiente OSF/DCE e é denominado DCE-CIOP (*DCE Common Inter-ORB Protocol*). O transporte de mensagens é possível através do mecanismo DCE-RPC, porém deixando ao cargo da ORB a formatação das mensagens, o despacho das operações e a ordenação dos dados. Entre os objetivos do DCE-CIOP podemos citar:

- \* Suporte a ORBs de fornecedores diferentes de missão crítica e corporativas;
- \* Propiciar o uso de serviços DCE quando apropriado;
- \* Possibilita o uso de APIs DCE públicas;
- \* Preservação da liberdade de implementação da ORB.

Neste capítulo foi apresentada uma descrição geral do ambiente CORBA, que será utilizado na implementação do projeto do ambiente cooperativo. A vantagem na utilização CORBA relaciona-se a sua portabilidade, facilidade de implementação, se comparada com outras tecnologias, e facilidade de atualização da aplicação.

## **2.8. Considerações Finais**

Os itens discutidos neste capítulo procuram oferecer uma idéia geral da teoria e das tecnologias que servem de base para a pesquisa proposta, oferecem um subsídio para o entendimento do sistema.

Tendo definido o contexto e a base teórica para a pesquisa, os capítulos subsequentes procuram expor as características ligadas ao projeto e implementação do sistema de CSCW proposto inicialmente.

## **3. Concepção e Projeto de Sistema Cooperativo para Auxílio ao Diagnóstico Médico por Imagens**

### **3.1. Introdução**

Este capítulo é dedicado á análise e concepção do sistema cooperativo para medicina sendo que o projeto do sistema cooperativo compreende as seguintes fases, que serão descritas mais detalhadamente nas próximas sessões:

\* **Análise de requisitos:**

Constitui a primeira fase do projeto onde são levantadas as características do problema a ser solucionado.

\* **Especificação Funcional:**

Após estudar o problema a ser resolvido passamos para uma fase onde será desenvolvido um modelo geral do sistema. Este modelo descreve o comportamento e as funcionalidades a serem oferecidas.

\* **Projeto da Arquitetura:**

Tendo descrito o modelo geral do sistema a última fase de projeto envolve uma descrição mais detalhada dos módulos e modelos de objeto que servirão de base para a implementação do mesmo.

### **3.2. Análise de Requisitos**

Este item objetiva apresentar as principais características e possíveis soluções para o problema proposto como pesquisa de pós-graduação.

O tema focaliza a criação de um ambiente cooperativo para auxílio ao diagnóstico médico por imagens. O trabalho cooperativo promove a interação entre usuários que buscam um mesmo objetivo. Neste caso temos usuários remotos interagindo através de uma rede de computadores em busca da opinião de especialistas, objetivando um diagnóstico mais preciso através do compartilhamento de dados, informações e experiências.

#### **3.2.1. Descrição do Problema**

A análise inicial do problema levou a uma classificação em três categorias de

problemas a serem solucionados:

\* **Ferramentas de análise de imagens médicas:**

O processo de diagnóstico necessita de ferramentas adequadas para a análise das imagens. Uma ferramenta de análise em radiologia utiliza técnicas para a visualização das imagens bidimensionais e tridimensionais e a extração de informações e medidas a partir dos dados analisados;

\* **Estrutura e método para troca de informações e dados entre usuários em rede de forma interativa:**

A opinião de especialistas muitas vezes é essencial para os casos mais complicados, porém nem sempre estes especialistas encontram-se disponíveis nos locais onde são necessários. É interessante do ponto de vista financeiro, de disseminação de conhecimento e qualidade de diagnóstico local dispor de uma estrutura de comunicação adequada, onde os dados possam ser analisadas em conjunto entre especialistas distribuídos em locais geograficamente distintos

\* **Técnicas mais avançadas para processamento de imagens:**

A prática radiológica gera diariamente grandes quantidades de dados, que se agrupam em unidades caracterizadas por: volume extenso e custo elevado de processamento. Muitos algoritmos utilizados no processamento das imagens médicas demandam máquinas de alto desempenho para manter a viabilidade de seu uso.

### **3.2.2. Processo de Análise**

O problema foi analisado com base em discussões com médicos, visitas ao Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo, leitura de artigos da área de CSCW e Telemedicina e participação de palestras e listas de discussão. A análise dos artigos é interessante para conhecimento das aplicações já existentes na área, porém as visitas ao Hospital e as conversas com médicos serviram para definir o contexto brasileiro na área de saúde e o comportamento dos usuários do sistema.

<b>Atividade</b>	<b>Descrição</b>
Visita ao Hospital	Visita aos Departamentos de Radiologia, Instituto da Criança e Instituto do Coração do Hospital das Clínicas de São Paulo.
Discussões	Reuniões com médicos do Hospital das Clínicas de São Paulo.
Leitura de artigos	Leitura de artigos nas áreas de CSCW, <i>Groupware</i> , Telemedicina e outras tecnologias.
Palestras	Participação em palestras sobre informática médica.
Listas de discussões	Participação em listas de discussão sobre informática médica.

Tabela 1: Atividades de análise de requisitos.

As atividades listadas na tabela acima forneceram os dados necessários para compreensão dos problemas e para a escolha das possíveis soluções. A seguir serão expostas as informações levantadas através da pesquisa realizada para especificação do ambiente de telemedicina.

### **3.2.2.1 Situação Atual do Hospital**

Atualmente a disseminação do conhecimento na área médica é feita a partir de centros principais que detêm o conhecimento, melhores recursos financeiros e de infraestrutura. Dessa forma, a população acaba se deslocando preferencialmente para estes centros em busca de especialistas ou de melhor atendimento.

A comunicação entre centros e os locais que necessitam da informação mais especializada é feita através de cartas, fax ou telefonemas. Estes meios de comunicação não são idealmente eficientes no que diz respeito a rapidez, qualidade da informação transportada e interatividade com a fonte. Assim, muitas vezes é necessário transportar os pacientes para os grandes centros ou levar o especialista para os locais desprovidos do conhecimento necessário para obtenção de um diagnóstico mais preciso.

O transporte dos pacientes pode envolver risco, conforme a gravidade da doença a ser tratada, custos extras, problemas com o excesso de pacientes nos grandes centros, falta de estrutura para acomodação do paciente e familiares, desestruturação de seu cotidiano

e total desambientação com a cidade que o abrigara durante o diagnóstico e tratamento.

O deslocamento de especialistas envolve custos adicionais, depende da disponibilidade do mesmo, desestrutura sua rotina de trabalho além de afastá-lo temporariamente de seus familiares.

### **3.2.2.2 Infra-estrutura de Computação do Hospital**

Atualmente o hospital conta com uma rede de computadores conectada à Internet, constituída por PC's e estações de trabalho. Além dos computadores há alguns tomógrafos DICOM<sup>6</sup> ou proprietários conectados a esta rede.

### **3.2.2.3 Rotina de Diagnóstico no Setor de Radiologia**

Para o desenvolvimento do projeto, escolheu-se trabalhar com dados radiológicos. Esta decisão baseou-se na intenção de selecionar problemas críticos, apresentando características como: volume extenso de dados, necessidade de processamento intenso e armazenamento em forma digital.

Ao visitar o departamento de radiologia do hospital, foi possível analisar o processo de envolvido em um exame radiológico. Este processo pode ser dividido em três etapas.

\* **Primeira etapa – Aquisição dos dados:**

A aquisição de dados radiológicos, em diversas modalidades de diagnóstico por imagens, envolve a submissão do paciente ao aparelho (tomógrafo ou Ressonância Magnética, especificamente neste trabalho). Nesta etapa são coletados os dados necessários para o diagnóstico, sendo a operação de aparelho feita por um técnico e idealmente com acompanhamento de um médico. Os dados são impressos em filme radiológico.

Muitas vezes os dados são adquiridos em formato analógico pelo aparelho, que os converte para formato digital, conservando ainda uma boa resolução. Contudo estes dados devem sofrer reconversão posterior para formato analógico, ao serem impressos em filme radiológico. Após a impressão, a

---

<sup>6</sup> **DICOM (Digital Imaging and Communications in Medicine):** Padrão desenvolvido pela ACR-Nema (*National Electrical Manufacturers Association*) que promove a interconexão de equipamentos de aquisição de imagens médicas em redes de computadores. Este padrão é desenvolvido segundo o padrão de arquitetura de redes OSI.

chapa passa a ser o único registro do exame, sendo que muitas vezes elas são cedidas ao paciente que as leva para casa.

\* **Segunda etapa – Análise dos dados:**

Esta etapa envolve a análise dos dados coletados pelo aparelho, onde um especialista analisa o filme radiológico proveniente do exame. A análise dos filmes radiológicos envolve a avaliação das estruturas de interesse (medidas de distância/área/volume, avaliação de formato e textura). Atualmente estas medidas são feitas de modo informal, comprometendo sua precisão.

\* **Terceira etapa – Emissão do laudo:**

A última etapa envolve a emissão do laudo baseada na análise das imagens. Nesta etapa o especialista determina as principais possibilidades referentes ao paciente. No caso de dúvidas, podem ser feitas consultas à base de dados de casos, procura por bibliografia na área em questão ou consulta a outros especialistas. Outra atividade comum, e muito importante neste processo, é a comparação entre exames de diferentes modalidades ou exames adquiridos dentro de um certo intervalo de tempo para acompanhamento da evolução do paciente.

Um aspecto importante a ser considerado é o volume de exames diários realizados no hospital. Este número é elevado, tornando o processo de diagnóstico mais lento, devido ao acúmulo de trabalho. A adoção de uma forma de automação deste serviço e novas técnicas de análise dos exames aumentaria a velocidade do processo de diagnóstico contribuindo para a melhora da qualidade do mesmo.

#### **3.2.2.4 Características do Dados**

Os dados a serem tratados são obtidos através de exames radiológicos como CT (Tomografia Computadorizada) ou MRI (Ressonância Magnética), apresentando características como:

\* **Formatos de arquivo:**

Os dados a serem trabalhados podem ser armazenados nos formatos *hdf*, *Raw Data*, *DICOM* ou algum formato proprietário. O formato *hdf* é um formato conhecido e amplamente utilizado. O *Raw Data* é um formato constituído por arquivos binários onde cada elemento do volume (*voxel*) ou da imagem (*pixel*)



é representado por números. O *DICOM* é uma padrão para medicina, especificado pela *ACR/Nema* (*American College of Radiology / National Electrical Manufacturers Association*). Os formatos proprietários dependem de cada fabricante, que podem negar documentação referentes a sua formatação, devendo ser evitados.

	Características	Grau de utilização	Grau de padronização
<b>Hdf</b>	Formato genérico, com formatação simples, usado em uma grande quantidade de aplicações.	Alto	Médio
<b>DICOM</b>	Formato específico para medicina, permitindo a gravação de dados dos pacientes juntamente com os exames.	Médio	Alto
<b>Raw Data</b>	Formato genérico sem formatação alguma.	Alto	Baixo

Tabela 2: Comparação entre formatos de dados

\* **Aquisição e Digitalização dos dados**

Existem duas categorias básicas de dados em radiologia: *small matrix size* (CT, MRI, ultra-som, medicina nuclear, angiografia digital e fluorografia digital) e *large matrix size* (radiografia computadorizada e filmes radiológicos digitalizados). Os dados do tipo *small matrix* devem oferecer uma resolução de 512x512 *pixels* com profundidade mínima de 8 bits por *pixel*. Os dados *large matrix* devem oferecer uma resolução espacial mínima de 2,5 lp/mm com profundidade de 10 bits por *pixel* [ACR Standards 99].

A aquisição e digitalização dos dados pode ser feita segundo dois processos [ACR Standards 99]:

**Captura Direta da Imagem:** os dados produzidos pela modalidade radiológica digital podem ser transferidos diretamente para o sistema de teleradiologia usando padrões como o DICOM, sendo o modo preferível

de aquisição de dados digitais para o diagnóstico primário;

**Captura Secundária da Imagem:** neste caso os dados podem ser tanto *large matrix* como *small matrix*. No caso de *small matrix*, o dado deve ser digitalizado para um tamanho de matriz maior ou igual ao original com uma profundidade mínima de 8 bits/pixel. No caso de *large matrix*, os dados devem ser digitalizados para um tamanho de matriz correspondente a 2,5 lp/mm ou maior, com profundidade mínima de 10 bits/pixel.

No momento da aquisição dos dados devem ser incluídas informações referentes ao paciente como nome, número de identificação do paciente, data e hora do exame, instituição, e dados referentes ao exame, como tipo de exame, orientação, parte anatômica do paciente e método de compressão.

\* **Escala Hounsfield:**

A modalidade de CT trabalha com uma escala de tons de cinza para visualização das imagens. Esta escala é denominada *Hounsfield*, onde cada tom de cinza é associado a uma escala que varia de -1000 a 1000, sendo o valor zero associado a água. A escala é baseada na densidade dos tecidos, sendo conhecidos os valores relacionados a cada um deles.

No console do aparelho são usados dois parâmetros para seleção do tecido a ser visualizado, a largura da janela (*Window Wide* ou *WW*) e o nível da janela (*Window Level* ou *WL*). A utilização destes dois parâmetros é semelhante ao conceito de brilho e contraste de uma imagem, porém dentro do intervalo já mencionado.

\* **Visualização das imagens:**

No cotidiano, as imagens analisadas por médicos, são impressas em filmes radiológicos e visualizadas no negatoscópio, tela acrílica translúcida iluminada por trás por uma lâmpada fluorescente, cuja a luz se dispersa de forma homogênea através da tela onde são colocados os filmes radiológicos.

Os tomógrafos possuem 12 ou 16 bits para codificação da escala de tons de cinza utilizada nas imagens (dependendo do fabricante do aparelho), sendo possível obter uma variação de 4096 ou 65535 tons de cinza. A resolução das imagens (cada fatia da tomografia) geralmente alcançam a resolução de 512x512 pixels, podendo chegar até 1024x1024 pixels em alguns casos.

De forma geral as características do ambiente e das estações de trabalho usadas para interpretação das imagens devem seguir os seguintes critérios [ACR Standards 99]:

1	Sugere-se a luminosidade de 550 lumen/m <sup>2</sup> para monitores monocromáticos.
2	Deve haver cuidados especiais com a iluminação do ambiente, evitando a ocorrência de reflexos no monitor.
3	Possibilidade de seleção de seqüência de imagens.
4	Capacidade de associação do paciente e estudos demográficos com a imagem em estudo.
5	Capacidade de ajuste de largura e nível de janela se necessário.
6	Oferecimento de funções de <i>Pan</i> e <i>Zoom</i> .
7	Possibilidade de rotação e inversão da imagem, porém preservando os dados referentes a orientação original do paciente.
8	Capacidade de calcular e mostrar medidas lineares precisas e valores dos <i>pixels</i> em valores apropriados para cada modalidade.
9	Capacidade de mostrar a taxa de compressão das imagens, processamento ou regiões cortadas.
10	Mostrar elementos como tamanho da matriz, profundidade dos bits ( <i>bit depth</i> ) e total de imagens adquiridas no estudo.

Tabela 3: Lista de critérios para visualização de imagens médicas segundo a ACR/Nema

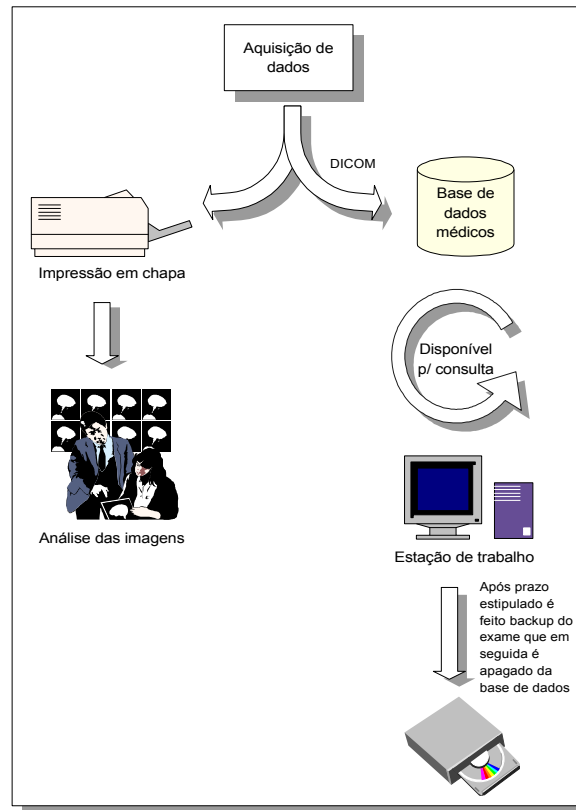
\* **Processamento dos exames:**

Os arquivos de dados de CT e MRI costumam ser extensos, apresentando características 3D. Um exame, contendo dados volumétricos, de resolução 256X256X64 o tipo inteiro curto (*short integer*), tem um tamanho da ordem de 8M bytes.

Além de consumirem muito espaço para armazenamento, acabam por exigir uma alta taxa de E/S (entrada/saída) e boa capacidade de processamento na máquina que os hospeda.

\* **Fluxo dos dados:**

O fluxo dos dados médicos envolvem as seguintes etapas:



1. Aquisição dos dados;
2. Impressão em filme radiológico;
3. Armazenamento em banco de dados;
4. Os dados são colocados a disposição para consulta durante período pré determinado;
5. Criação de cópia de segurança e remoção do banco de dados, por motivo de economia de espaço na máquina servidora. Caso se deseje consultar um dado antigo, é necessário fazer uma requisição para que os dados sejam recuperados através da cópia de segurança.

Figura 13: fluxo dos dados médicos

### 3.2.3. Alguns Conceitos

Alguns conceitos básicos serão apresentados a seguir com o intuito de fixar alguns termos empregados em diagnóstico por imagens.

\* **Multitemporalidade:**

Possibilidade de comparação entre dois exames realizados em diferentes datas para o acompanhamento da evolução de um paciente.

\* **Multimodalidade:**

Possibilidade de observar diferentes modalidades de exames de um mesmo paciente.

\* **Multiparâmetros:**

Possibilidade de analisar exames de uma mesma modalidade porém realizados com diferentes ajustes de parâmetros.

\* **Multidimensionalidade:**

Um exame pode ser bidimensional (raio X), tridimensional (MRI ou CT) ou ainda serem efetuadas aquisições em um intervalo de tempo, adicionando mais

uma dimensão.

\* **Multicaso:**

Pode ser necessário a comparar casos de diferentes pacientes relacionados a uma mesma patologia.

### **3.2.4. Avaliação dos Requisitos**

É interessante que os pacientes possam ser tratados em seus locais de origem, evitando problemas como superlotação dos hospitais nos grandes centros. Assim sendo, deseja-se a disseminação do conhecimento para os locais desprovidos de tais recursos, de forma a alcançar uma melhor qualidade de atendimento local e diminuição de custos.

Quando trabalhamos com exames radiológicos, há uma grande facilidade na obtenção dos dados em forma digital, o que facilita a aplicação de algum tipo de computação a estes dados e sua transmissão por algum meio de telecomunicação. Neste contexto novas tecnologias podem ser introduzidas às metodologias de diagnóstico, oferecendo como benefício uma velocidade maior de diagnóstico e um aumento da precisão dos resultados do mesmo. Entre as tecnologias que podem ser utilizadas temos a computação de alto desempenho, que permite uma análise mais rápida dos dados com uma precisão maior e tecnologias de telecomunicações, que permitem a comunicação entre especialistas fisicamente dispersos.

Outro fator a ser considerado é o custo de implementação, implantação e manutenção das novas tecnologias adotadas. Assim é interessante ao novo sistema de informática aproveitar, na medida do possível, a infra-estrutura já existente, bem como aceitar a incorporação de novas técnicas e tecnologias que possam surgir, evitando a obsolescência.

### **3.2.5. Proposta**

Este trabalho propõe o modelamento e o desenvolvimento de um protótipo de ambiente cooperativo, que aproveite a rede de computadores já existente no hospital, e que possibilite a comunicação entre especialistas localizados em pontos remotos. Além dos recursos de CSCW (*Computer Supported Cooperative Work*), o modelo de ambiente prevê a utilização recursos para processamento remoto dos dados de exames

como CT (Tomografia Computadorizada) e MRI (Ressonância Magnética).

A radiologia constitui uma área muito extensa, possuindo diversas modalidades diferentes de exames, como por exemplo: radiografia, ultra-sonografia, CT, MRI, etc. Desta forma escolheu-se trabalhar com exames do tipo CT e MRI, visto que é muito trabalhoso desenvolver um sistema que atenda a todas as particularidades de cada modalidade. Porém graças ao ambiente e à estratégia de modelagem do sistema adotados, é possível a extensão posterior para novas modalidades sobre a infra-estrutura de comunicação implementada, isto é, a infra-estrutura de gerenciamento do ambiente CSCW pode ser aproveitada por novas ferramentas que atendam outras modalidades radiológicas.

Outro motivo da escolha por modalidades como CT e MRI se deve a características dos dados como: volume extenso e processamento intensivo, por se tratarem de duas modalidades que geram um grande volume de dados diários no hospital e por necessitarem de técnicas precisas para obtenção de medidas.

O processamento remoto possibilita a disponibilização de recursos como computadores de alto desempenho, existentes só nas universidades no caso do Brasil. Já o CSCW fornece base teórica para a implementação de aplicações para trabalho em grupo.

A partir de uma pesquisa sobre sistemas de CSCW e telemedicina já existentes, foram levantadas algumas características gerais existentes nestes sistemas. Estas características são listadas na Tabela 4:

<b>Característica</b>	<b>Descrição</b>	<b>Impacto</b>
Colaboração entre múltiplos usuários	Os usuários interagem em tempo real, compartilhando informações e dados.	Possibilidade de estabelecimento de reuniões mesmo que os usuários não encontrem-se alocados no mesmo local físico.
Lousa	Área (tela) compartilhada entre usuários em sessão cooperativa.	Possibilita a troca de informações, de forma síncrona, entre pessoas distribuídas em locais distintos.
Área de dados compartilhada	Área usada para compartilhamento de arquivos, dados de pacientes, ou qualquer tipo de informação relevante.	Facilita o acesso aos dados, de forma assíncrona, entre locais distintos.
<i>Chat</i>	Usado para conversas entre especialistas, com baixo custo para a rede.	Permite o contato fácil e rápido entre dois especialistas.
Segurança	Deve ser evitado o acesso indevido aos dados médicos.	Proteção da informação médica que pode ser usada indevidamente, o que pode constituir fator de discriminação com relação ao paciente.
Disponibilidade	O sistema de computação deve estar sempre disponível, mesmo na ocorrência de falhas ou erros.	A disponibilidade do sistema, mesmo em situações crítica não interrompe a rotina hospitalar.
Serviço de correio eletrônico	Troca de mensagens fácil e rápida de forma assíncrona.	Usado para correspondência entre usuários, agendamento de encontros, discussão de casos, etc.
Pastas de casos	Manutenção pastas com informações sobre casos, como histórico, exames, relatórios de progresso do paciente	Acesso fácil e organizado às informações sobre pacientes e casos, auxiliando na tomada de decisões.
Acesso a conhecimento instantaneamente ( <i>on line</i> )	Acesso fácil à informações bibliográficas.	É interessante ao médico poder consultar bases de dados de casos, para referência, facilitando a tomada de decisão
<i>Bulleting boards</i> e listas de discussão	Forma de comunicação assíncrona.	Usado para discussão de casos e troca de informações.
Serviço transferência de arquivos	Arquivos podem ser transferidos livremente pela rede, desde que respeitada as restrições de segurança.	Facilidade de obtenção de arquivos médicos.
Calendário de eventos	Auxílio na programação de eventos e distribuição de datas e horários.	O calendário de eventos é usado, por exemplo, para agendamento de reuniões para discussão de caso.
Vídeo conferência e áudio.	Transmissão de vídeo e voz nas reuniões cooperativas.	Formal fácil de interação entre pessoas distantes, oferecendo identificação visual e facilidade de comunicação..
<i>Webcasting</i>	Canal de distribuição de informações em tempo real.	Distribuição constante de informações e notícias.
Configurações de usuários	Cada usuário pode configurar seu <i>desktop</i> da forma que lhe for mais agradável.	Possibilidade de configurar o sistema conforme as necessidades e preferências pessoais. Adiciona conforto e praticidade ao ambiente.
Esquema de <i>Log</i>	Manutenção de <i>logs</i> das sessões para posterior consulta.	Possibilita a consulta do conteúdo das sessões de trabalho em grupo para fins diversos.

Tabela 4: Características gerais de sistemas de colaborativos.

Nota-se que as características levantadas podem atender a uma vasta gama de aplicações de trabalho cooperativo. Contudo, dentre estas características gerais devemos escolher as mais adequadas aos sistemas de teleradiologia, em particular voltados à modalidades como CT e MRI e trabalho cooperativo em medicina, o que será feito nas próximas etapas.

A Tabela 5 procura classificar as características descritas na tabela anterior segundo sua prioridade no contexto desta pesquisa.

Característica	Prioridade	Viabilidade
Colaboração entre múltiplos usuários	Alta prioridade por ser o foco principal de estudo em CSCW.	Viável e essencial. Apresenta certa complexidade de implementação, contudo constitui a base em um sistema de CSCW.
Lousa	Alta prioridade por ser uma metáfora essencial para a colaboração síncrona entre usuários distantes.	Viável, apresentando forma simples de interação entre usuários remotos e de complexidade de implementação média.
Área de dados compartilhada	Prioridade média., afetando apenas a colaboração assíncrona. Oferece uma forma de troca informações entre usuários.	Dentro do contexto desta pesquisa a viabilidade é baixa por requerer segurança de acesso e integração com a base de dados hospitalar.
Chat	Alta prioridade. Além do lousa é necessário uma segunda forma de comunicação e troca de informações entre usuários. Funciona como complemento na forma de interação do lousa.	Viável por apresentar fácil implementação e baixo custo para rede.
Segurança	Alta prioridade. A segurança dos dados médicos é muito importante, tentando-se evitar que estes dados sejam acessados e publicados indevidamente.	Inviável neste trabalho. A segurança de dados é complexa, sendo tratada por outras áreas de pesquisa.
Disponibilidade	Alta prioridade. Mesmo em situações especiais, o sistema deve estar disponível para uso, mesmo que o desempenho seja degradado até níveis aceitáveis..	Viável, porém em um nível básico para este trabalho. Envolve outras áreas de estudo, como gerenciamento de redes por exemplo, não abordados neste trabalho.
Serviço de correio eletrônico	Alta prioridade, por oferecer um meio assíncrono de comunicação fácil, rápido e bastante utilizado por muitas pessoas.	Viável, porém por existirem muitas ferramentas para este fim, é interessante trabalhar com a integração de ferramentas apenas.
Pastas de casos	Baixa prioridade neste trabalho, pois não influencia fortemente a interação entre usuários, apesar de ser importante no processo de diagnóstico médico.	Não é viável nesta pesquisa por requerer integração com a base de dados hospitalar e segurança de acesso.
Acesso a conhecimento <i>On line</i>	Baixa prioridade neste trabalho, por também não apresentar características de trabalho em grupo assistido por computador. Porém importante no processo de diagnóstico médico.	Viável pois apresenta fácil implementação em ambiente <i>web</i> .
<i>Bulleting boards</i> e listas de discussão	Alta prioridade, pois influencia diretamente o trabalho em grupo, oferecendo formas assíncronas de comunicação.	Viável. Implementação simples porém o uso depende do interesse dos usuários.
Serviço transferência de arquivos	Prioridade alta. Usada para troca de dados entre usuários, oferecendo forma de interação assíncrona.	Viável, podendo ser feito por FTP.
Calendário de eventos	Prioridade média. Oferece um a forma de comunicar a ocorrência de reuniões.	Viável, com implementação simples através de <i>Web</i> .
Vídeo conferência e áudio.	Prioridade alta. Forma rápida e fácil de comunicação	Neste trabalho inviável. A vídeo conferência apesar de ser atraente para os usuários requer equipamentos e programas específicos. Seria interessante tentar integrar alguma tecnologia já existente futuramente.
<i>Webcasting</i>	Prioridade média. A distribuição de informações na rede de forma automática, é interessante, porém não essencial. Não afeta fortemente a interação entre usuários.	Não é viável neste trabalho, por depender da alimentação do sistema com as informações a serem distribuídas.
Configurações de usuários	Prioridade média. Oferece conforto aos usuários.	Viável. Depende apenas de projeto e implementação dentro do sistema.
Esquema de <i>Log</i>	Prioridade média. No caso de dúvidas é interessante poder consultar o <i>log</i> da sessão de trabalho cooperativo.	Viável. Pode ser implementado facilmente.

Tabela 5: Características gerais de sistemas de colaborativos organizadas segundo prioridade.

### 3.3. Especificação Funcional

#### 3.3.1. Introdução

Este item tem como objetivo delimitar as características desejáveis em um sistema de telemedicina. Existem diversas modalidades abrangidas pela telemedicina, como o telemonitoramento, a teleconsulta ou a teleradiologia. Entre estas modalidades,



escolheu-se trabalhar com a teleradiologia, que cuida de aplicações voltadas à radiologia em telemedicina.

Neste caso o problema a ser resolvido é a interação entre médicos radiologistas, permitindo-se a troca de informações, dados e experiência de forma mais rápida e fácil entre especialistas situados em localidades remotas.

Como solução para este problema, propõe-se um ambiente onde médicos possam interagir remotamente, através do uso de computadores em rede. Este ambiente é um ambiente de CSCW (*Computer Supported Cooperative Work*).

Dentro desse ambiente os médicos devem ser capazes de se comunicar através de interfaces gráficas simples, trocando informações e dados. Todo esse processo objetiva o diagnóstico, no qual diversos especialistas opinam para chegar a um consenso.

### **3.3.2. Visão Geral do Projeto**

O ambiente a ser modelado deve proporcionar a integração entre: (i) comunicação entre especialistas que colaboram em um diagnóstico, (ii) ferramentas para gerenciamento dos encontros cooperativos, (iii) integração com a base de dados hospitalar e (iv) processamento dos dados médicos em máquinas de alto desempenho.

O acesso ao ambiente deve ser transparente, isto é, um usuário não precisa ter conhecimento dos detalhes de acesso, localização ou processamento dos dados. É oferecida uma interface gráfica para esse acesso e que atenda de forma simples os objetivos dos médicos ocultando o aspecto computacional. A interface para suporte ao diagnóstico deve conter pelo menos as ferramentas indispensáveis a este fim, porém extensíveis para o trabalho em conjunto.

Resumidamente, o ambiente seria constituído por serviços que viabilizariam o trabalho cooperativo e o processamento remoto dos dados médicos, e interfaces destinadas ao acesso a tais serviços. Dessa forma a colaboração entre usuários residiria nas interfaces de acesso ao ambiente e em serviços que encapsulariam o gerenciamento de usuários e dos grupos de trabalho. Os serviços voltados ao processamento remoto de dados médicos seriam encapsulados em um segundo servidor destinado apenas para este fim.

Em linhas gerais podemos agrupar os serviços oferecidos pelo ambiente cooperativo segundo as seguintes categorias:

\* **Serviços de CSCW:**

1. Gerenciamento de usuários e grupos de trabalho;
2. Organização do grupo de trabalho e comunicação;
3. Acesso a dados compartilhados;
4. Visualização e colaboração:

\* **Interface gráfica para visualização de imagens;**

1. Ferramentas para análise das imagens em grupo;

\* **Serviços de processamento de dados radiológicos:**

1. Processamento remoto de dados;

A Figura 14 representa de forma simplificada o modelo do sistema cooperativo de auxílio ao diagnóstico médico.

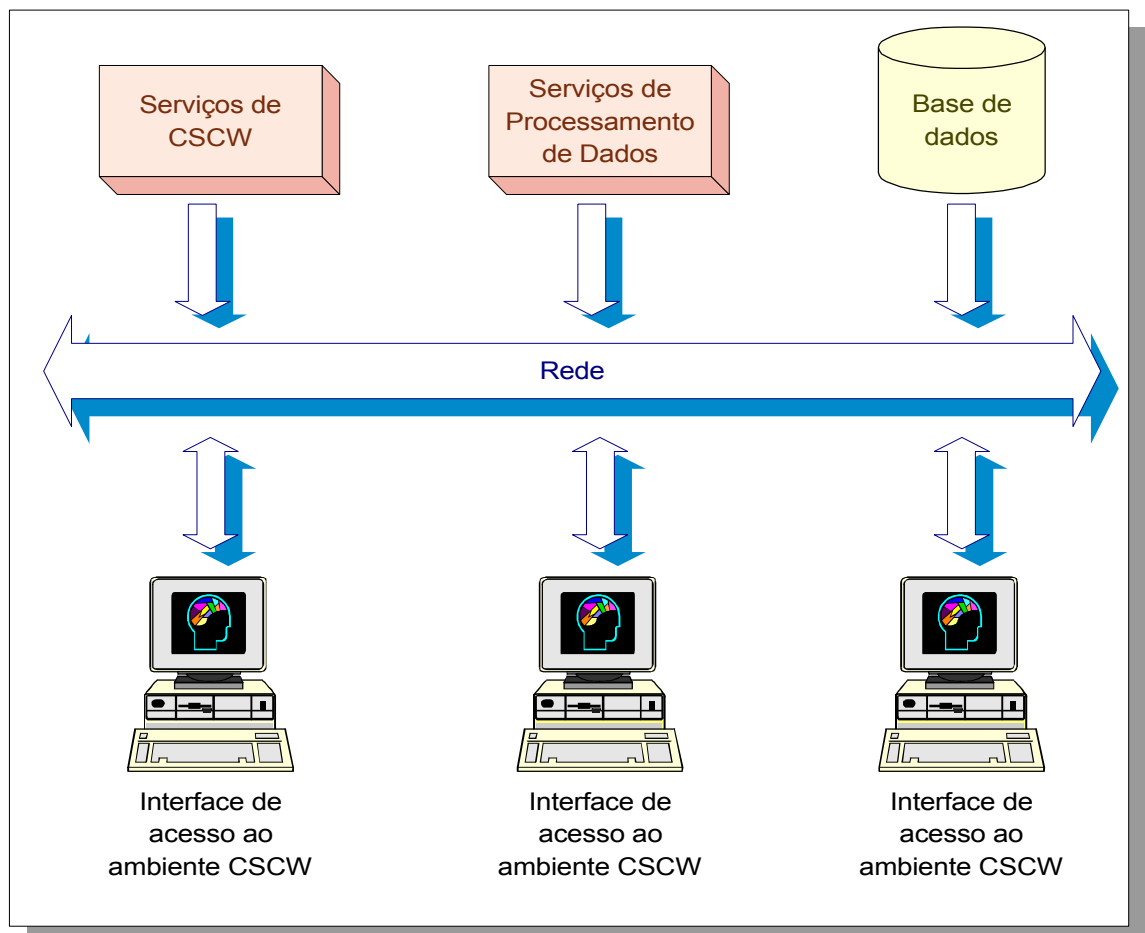


Figura 14: Serviços oferecidos pelo ambiente

Os serviços de CSCW compreendem toda a parte de suporte ao trabalho em grupo, enquanto os serviços de processamento de dados englobam o processamento remoto de imagens e volumes provenientes de exames radiológicos.

Os itens subsequentes descrevem com mais detalhes os grupo de serviços identificados no ambiente.

### **3.3.3. Serviços de CSCW**

Os serviços de CSCW, como dito anteriormente, estão relacionados ao trabalho em grupo, oferecendo recursos para identificação de usuários, gerenciamento de grupos de trabalho em tempo de execução, gerenciamento de tarefas, agendamento de reuniões cooperativas e acesso a dados compartilhados dos pacientes e banco de dados de casos.

Esses serviços são subdivididos nos seguintes grupos conforme o tipo de tarefa desempenhada: visualização e manipulação de dados, gerenciamento de usuários e grupos de trabalho, gerenciamento de tarefas e encontros e acesso a dados compartilhados.

#### **3.3.3.1 Visualização e Manipulação de Dados**

A visualização e manipulação dos dados médicos é feita através de uma interface gráfica projetada especificamente para este fim. Tal interface deverá fornecer recursos para visualização e avaliação de exames de CT e MRI, permitindo a análise dos dados de forma individual ou em grupo.

Quando utilizada em modo de trabalho cooperativo, a interface deve permitir trabalho conjunto entre um grupo de usuários que compartilham não apenas as imagens visualizadas mas também as ações realizadas sobre estas imagens, de forma a manter o sincronismo entre as interfaces dos diferentes participantes. Por exemplo, se um dos usuários abrir uma imagem, todos visualizarão a mesma imagem. O compartilhamento de recursos e ações é feito através de uma área da interface marcada como compartilhada. Esta área compartilhada servirá como lousa.

A qualquer momento da execução da interface o usuário pode optar por trabalhar sozinho ou em sessões cooperativas, onde uma sessão cooperativa é caracterizada pelo trabalho em grupo de dois ou mais usuários. Toda sessão de trabalho em grupo deve ter

um líder que a coordena, sendo este responsável por aceitar ou não a adesão de novos integrantes ao grupo, selecionar os dados a serem trabalhadas ou solicitar o processamento remoto destes dados. Algumas ações podem ser executadas por qualquer um dos integrantes do grupo, como por exemplo, selecionar uma área de uma imagem, obter uma medida de distância, etc. A qualquer momento da sessão, o líder pode passar o controle a outro participante. As mesmas funcionalidades oferecidas para o usuário que trabalha isoladamente devem ser fornecida ao grupo, isto é obtenção de medidas de áreas, distâncias ou volume, entre outros.

#### ***3.3.3.1.1 Diretivas de Projeto da Interface***

A interface é direcionada a usuários com pouca experiência em computação. Entretanto, são especialistas em sua área de formação (radiologia). A ferramenta prevê dois níveis de diálogo: um diálogo direcionado aos principiantes feito por menus, e um diálogo feito por teclas de atalho e linhas de comando, direcionado aos usuários mais avançados. As opções de menu devem sempre indicar o caminho por teclas de atalho, estimulando o usuário a utilizar este método mais rápido de acesso às funções da aplicação. Esses requisitos objetivam tornar a ferramenta atraente, convidando ao seu uso freqüente em todos os estágios de habilitação do profissional médico no uso do computador.

As telas devem seguir a aparência mostrado na Figura 15. Aqui existem três janelas para visualização de imagens nos eixos ortogonais (X, Y e Z ou axial, coronal e sagital) e uma quarta janela para visualização tridimensional, sendo as vistas configuráveis. Assim, cada janela possui uma barra para seleção de uma fatia do volume, e botões para seleção do eixo de visualização ou imagem 3D. Qualquer uma das janelas pode ser maximizada para visualização da imagem ampliada. Um menu de acesso rápido disponibiliza as funcionalidades da interface, como medidas de área, distância e volume.

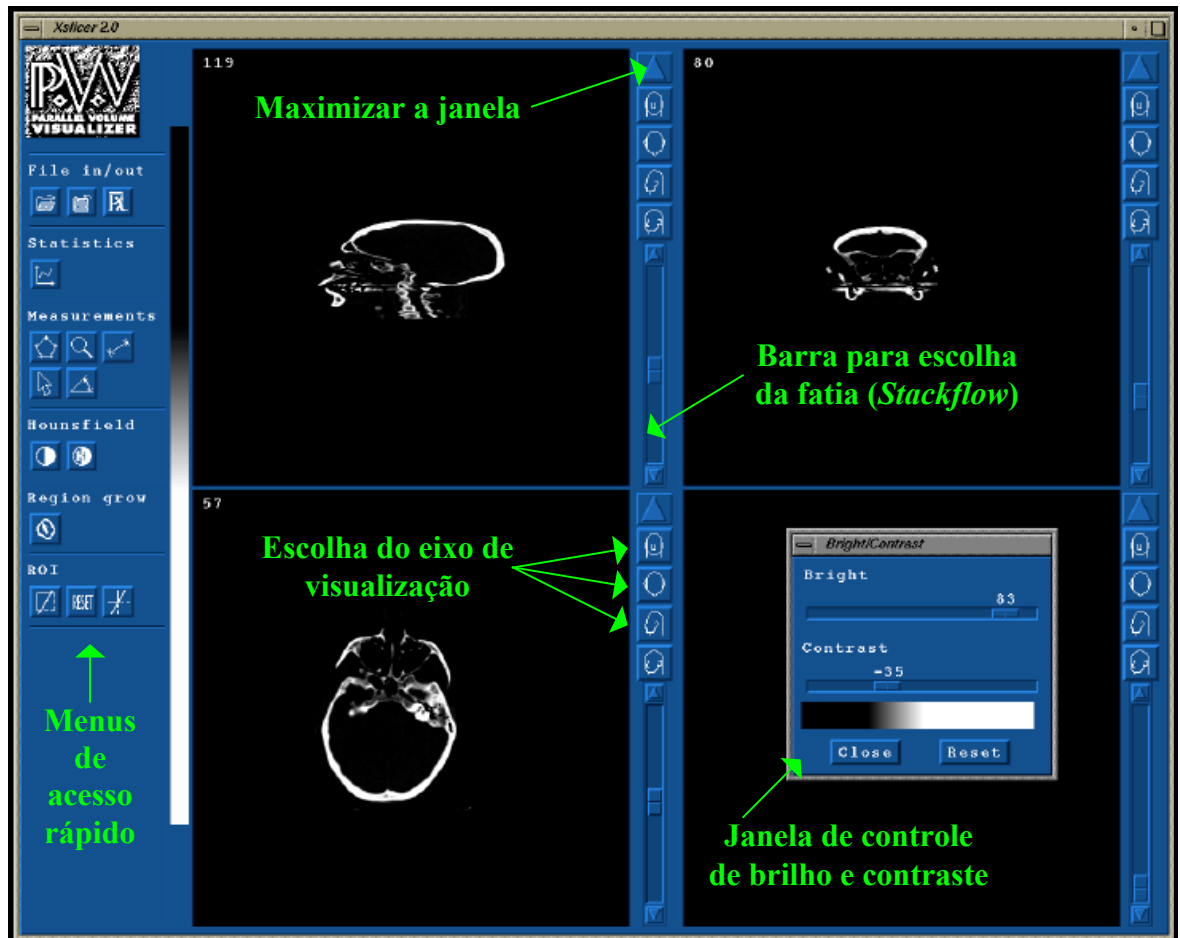


Figura 15: Protótipo de Interface Gráfica

Resumidamente, devem ser seguidas as seguintes regras para construção da aparência e do diálogo da interface em questão :

\* **Foco de atenção:**

O foco de atenção na interface está nas imagens. Deve-se evitar janelas sobrepondo-se as imagens, cores muito fortes ou qualquer característica dispersiva, que possa desviar a atenção do médico de seu objetivo. Geralmente a resolução das imagens varia entre 64X64 a 512X512, sendo interessante evitar a interpolação para resoluções menores para adequação aos tamanhos de janela.

\* **Interface limpa:**

Deve-se evitar exageros, como excesso de janelas e menus, que prejudiquem a navegação pela interface, cores muito fortes, ícones piscando, entre outros;

\* **Transparência:**

Tanto processamento dos dados como sua localização na rede, devem ser

transparente. Não é de interesse do usuário conhecer detalhes como localização dos dados ou onde é realizado o processamento de uma imagem, mas sim se a interface oferece um tempo de resposta adequado e imagens de boa qualidade.

As características listadas acima fornecem uma linha geral para o projeto de uma interface de auxílio ao diagnóstico em radiologia, procurando atender aos objetivos específicos da área estudada, teleradiologia.

### 3.3.3.1.2 *Funcionalidades Oferecidas*

A interface de visualização dos dados médicos deve oferecer uma série de funcionalidades que permitam, além da visualização de imagens 2D e 3D, a obtenção de medidas a partir dos dados de forma individual ou em grupo. Alguns recursos de visualização e de trabalho em grupo são obtidos através de acesso aos serviços do ambiente, outros recursos são processados na própria máquina que hospeda a interface.

Especifica-se a seguinte lista desses serviços, genericamente descritos:

<b>Abrir arquivo:</b>	Abre um arquivo de exame (CT ou MRI) local ou remoto.
<b>Fechar arquivo:</b>	Fecha um arquivo (CT ou MRI) local ou remoto.
<b>Salvar arquivo:</b>	Salva arquivo (CT ou MRI) em disco.
<b>Stackflow:</b>	Visualização de um volume através um mecanismo de deslizamento na pilha de fatias do mesmo. Deve ser possível visualizar as fatias nos três eixos ortogonais (Axial, Coronal e Sagital).
<b>Interpolação:</b>	Ampliação ou redução de uma imagem ou volume.
<b>Lente de aumento:</b>	Ampliação de uma área da imagem determinada pelo cursor, seguindo o passeio do mesmo como se fosse uma lente de aumento.
<b>Área:</b>	Cálculo da área de uma região selecionada em uma imagem.

<b>Distância:</b>	Cálculo da distância entre dois pontos selecionados em uma imagem.
<b>Ângulo:</b>	Cálculo do ângulo a partir de três pontos (duas retas) selecionados na imagem.
<b>Hounsfield</b>	Escala a tons de cinza utilizada na visualização de exames de tomografia.
<b>Brilho e contraste:</b>	Tratamento de brilho e contraste de uma imagem.
<b>Crescimento de região:</b>	Crescimento de uma região a partir de uma semente e de uma tolerância selecionados.
<b>Reconstrução 3D:</b>	Criação de uma imagem 3D a partir dos dados de CT ou MRI, através de parâmetros selecionados pelo usuário.
ROI (Region of Interest):	Seleção de uma região de interesse, isto é, delimitação de um sub-volume ou sub-imagem de interesse.
<b>Login/logout:</b>	Identificação de usuários no ambiente de CSCW.
Sessão Cooperativa:	Habilitar ou desabilitar o recebimento de chamadas para sessões cooperativas.
<b>Compartilhamento</b>	Habilitar compartilhamento de uma janela de imagem e das ferramentas disponíveis na janela (stackflow, área, distância, ângulo, lente de aumento e interpolação).
<b>Passê especial:</b>	Cada sessão cooperativa tem um líder, que deve coordenar as ações do grupo. O usuário dono do passê é o líder da sessão, podendo manter o passê por quanto tempo desejar, ou repassa-lo a outro usuário na sessão.

### 3.3.3.2 Gerenciamento de Usuários e Grupos de Trabalho

Esta categoria de serviços do ambiente cooperativo é responsável por identificar os usuários conectados ao sistema e agrupá-los para trabalho cooperativo. Para identificação, os usuários devem ser cadastrados em um sistema de banco de dados, sendo oferecida uma senha e um nome *login* aos usuários cadastrados.

Os usuários conectados ao sistema podem ser agrupados para o trabalho cooperativo. Assim, o sistema deve listar todos os usuários conectados, bem como os grupos existentes em um determinado momento. A adesão ou saída de um usuário em um grupo é permitida, sendo que o grupo pode vetar a adesão de um novo usuário.

No caso de trabalho em grupo, um usuário é escolhido como líder do grupo. Este usuário é identificado como líder recebendo um passe especial. O passe especial pode ser passado adiante a qualquer momento, sendo que usuário que retém o passe especial se torna o novo líder.

O sistema é dividido em domínios. Um domínio consiste na divisão da rede em localidades independentes que interagem com menor frequência. Cada domínio é responsável por administrar seus próprios usuários e dados, porém é possível obter informações ou conectar usuários externos, através de consultas a outros domínios.

Como vantagens da divisão em domínios temos a divisão do trabalho de gerenciamento dinâmico dos usuários e grupos de trabalho, a possibilidade de manutenção de bases de dados independentes para cada domínio, divisão do serviço de manutenção de dados, e a maior tolerância a falhas, pois no caso de queda de um dos domínios nem todo o sistema é afetado.

Além de controlar os usuários, grupos e domínios, também é importante controlar as aplicações disponíveis no ambiente. Este controle é importante do ponto de vista de segurança, constituindo uma forma de autenticar as aplicações disponíveis no sistema e gerenciar a disponibilidade das mesmas. Dessa forma, toda aplicação que possa oferecer algum tipo de serviço ao sistema é cadastrada, sendo o estado da mesma monitorado periodicamente. Este processo torna possível o reconhecimento das aplicações realmente pertencentes ao ambiente cooperativo e se estas se encontram disponíveis em um dado momento.

Os usuários cadastrados no sistema são classificados segundo duas categorias:

\* **Administrador:**

Este tipo de usuário não precisa ter formação em medicina, sendo ele responsável apenas por cuidar da parte do sistema relacionada a informática. Deve gerenciar o ambiente de trabalho cooperativo, podendo realizar as seguintes tarefas: cadastro e remoção dos usuários, gerenciamento do banco de dados, além das demais tarefas permitidas aos usuários comuns;

\* **Usuário comum:**

Usuário que utiliza os recursos do sistema, porém não realiza nenhuma tarefa



administrativa. Dentro desta categoria de usuários encontram-se os profissionais ligados a área de saúde, para o qual o sistema é destinado.

O usuários recebem permissões de acesso segundo sua categoria e restrições impostas pelo administrados do sistema na hora do cadastro. As permissões referem-se ao acesso aos dados do sistema (leitura e escrita), a participação em reuniões cooperativas e acesso aos serviços.

Por último, deve haver uma política de acesso com relação a usuários de diferentes domínios, definida pelo administrador do sistema local.

#### 3.3.3.2.1 *Funcionalidades Oferecidas*

<b>Funcionalidade</b>	<b>Descrição</b>
Manutenção de usuários	Cadastro, remoção e atualização de registros de usuários no sistema
Manutenção de Domínios	Cadastro, remoção e atualização de Domínios
Manutenção de Aplicações	Cadastro, remoção e atualização das aplicações disponíveis.

Tabela 6: Funcionalidades de manutenção dos serviços de CSCW.

Os serviços de manutenção de usuários são usados para cadastro de usuários no sistema, sendo este cadastro mantido em um banco de dados. Os serviços de manutenção de domínios são utilizados para localização e identificação dos domínios pertencentes ao ambiente cooperativo, viabilizando a comunicação entre os mesmos.

A manutenção de aplicações é utilizada para cadastro dos servidores e aplicações do sistema. Este cadastro é feito para garantir aos clientes que estão se conectando a servidores ou aplicações autênticos, isto é, que realmente pertencem ao sistema.

O servidor de CSCW deve possuir alguma funcionalidades para gerenciamento do mesmo por parte do administrador do sistema e para verificação do estado do servidor por parte das aplicações conectadas a este.

<b>Funcionalidade</b>	<b>Descrição</b>
Versão	Consulta a versão do servidor.
Versão da ORB	Consulta a versão de ORB utilizada na construção do servidor.
Estado do servidor	Consulta o estado do servidor em um dado momento.
Nº de clientes	Número de clientes conectados ao servidor.
Nº max. de clientes	Número máximo de clientes que podem se conectar ao servidor.

Tabela 7: Funções para verificação do estado do servidor.

Além das funcionalidades de manutenção do ambiente cooperativo, é necessário gerenciar usuários conectados ao ambiente e os grupos de trabalho existentes neste ambiente. A Tabela 8 mostra as funcionalidades referentes ao gerenciamento dinâmico de usuários no ambiente cooperativo.

<b>Funcionalidade</b>	<b>Descrição</b>
Login	Realiza a conexão do cliente com o servidor.
Logout	Fecha a conexão do cliente com o servidor.
Lista de usuários	Consulta a lista de usuários conectados ao servidor.
Chave de acesso de usuário	Cada usuário possui uma chave de acesso aos serviços do servidor. A chave é cedida ao cliente no momento do <i>Login</i> .
Estado do cliente	Consulta o estado de um cliente conectado.
Valor de <i>timeout</i> do cliente	Valor de em segundos entre o <i>polling</i> .
<i>Polling</i> do cliente	Cliente informa de tempos em tempos o próprio estado para o servidor.
Altera visibilidade do cliente	O cliente decide se deseja ou não aparecer na lista de clientes conectados.

Tabela 8: Funcionalidades de gerenciamento dinâmico de usuários.

Além de gerenciar os usuários existentes também é necessário gerenciar dinamicamente os grupos existentes no ambiente cooperativo. A Tabela 9 mostra as funcionalidades para controle de grupos de trabalhos.

<b>Funcionalidade</b>	<b>Descrição</b>
Nº máx. Grupos de trabalho	Número máximo de grupos de trabalhos suportado pelo servidor.
Nº de grupos de trabalho	Número de grupos de trabalho em um dado momento no servidor.
Cria Grupo de trabalho	Cria um novo grupo de trabalho.
Destroi grupo de trabalho	Destroi um grupo de trabalho.
Lista de grupos de trabalho	Lista os grupos de trabalho existentes no servidor em um dado momento.
Lista de usuários em um grupo de	Lista de usuários pertencentes a um grupo de trabalho.
Estado do grupo de trabalho	Estado do grupo de trabalho em um dado momento.
Líder do grupo de trabalho	Cada grupo de trabalho possui um líder.
Chave de acesso ao grupo de trabalho	Cada grupo de trabalho possui uma cópia da chave de acesso ao grupo. A chave permite aos usuários executarem tarefas no grupo.
Visibilidade do grupo de trabalho	O líder do grupo decide se o grupo aparecerá na lista de grupos de trabalho existentes no servidor.
Troca líder do grupo de trabalho	O líder do grupo de trabalho pode passar a responsabilidade para outro usuário do grupo.
Abandona Grupo de trabalho	Usado por um usuário para abandonar um grupo de trabalho.
Requisita entrada no grupo de	Usado por um usuário para requisitar a entrada em um grupo de trabalho.
Adiciona usuário ao grupo de trabalho	Se um usuário é aceito no grupo, o líder o adiciona ao grupo.
Recusa entrada do usuário no grupo de	O líder pode recusar a entrada de um usuário ao grupo.
Remove usuário do grupo de trabalho	Remove um usuário do grupo de trabalho.

Tabela 9: Funcionalidades para gerenciamento dinâmico de grupos de trabalhos.

### **3.3.3.3 Organização do Grupo de Trabalho e Comunicação**

Além de oferecer recursos para a comunicação e troca de dados e informações interativamente entre usuários, um sistema de trabalho cooperativo deve também oferecer recursos para gerenciamento de tarefas e encontros.

Estas ferramentas não objetivam promover uma interatividade síncrona entre usuários, mas sim criam um ambiente para organização pessoal e do grupo. Podem oferecer muitas vantagens aos seus usuários, facilitando o agendamento de encontros cooperativos e a distribuição de informações. Entretanto, por exigirem algum trabalho extra por parte dos usuários, que precisam preencher o sistema com seus dados e

verificarem periodicamente os eventos do sistema, o sucesso dessas ferramentas dependem da disciplina e do interesse dos usuários.

Para que o uso deste grupo de ferramentas seja viável, é necessário projetá-las para que sejam de simples utilização, a entrada de dados no sistema deve ser mínima e principalmente devem atestar sua real utilidade prática. Um aspecto importante no projeto desta parte do sistema é que ele deve atender não apenas às necessidades de alguns indivíduos mas sim de todo grupo.

A Tabela 10 mostra os serviços que auxiliam na organização e comunicação entre integrantes de um grupo de trabalho:

<b>Função</b>	<b>Descrição</b>
Agenda	Agenda eletrônica pessoal e do grupo.
Tabela de horários	Tabela de horários mantida pelo usuário.
Quadro de avisos	Quadro de avisos eletrônico.
Serviço de mensagens	Serviço de entrega de mensagens.
Agendamento eletrônico	Sistema de agendamento eletrônico de encontros.
<i>Chat</i>	Troca de mensagens interativamente
Vídeo conferência	Comunicação usando sons e vídeo.

Tabela 10: Funcionalidades de organização do grupo.

A maior parte deste módulo pode ser apresentada em formato que permita acesso através de *web (World Wide Web)*, permitindo o fácil acesso de qualquer ponto onde haja *Internet* e fácil manipulação por parte do usuário, muitas vezes já habituado a este ambiente.

A agenda eletrônica é usada para controle pessoal de cada usuário de seus horários e compromissos. A agenda do grupo é utilizada para verificação das reuniões envolvendo toda a equipe, ou parte dela. A Tabela 11, apresenta as funções de agenda, sendo estas oferecidas através de *web (World Wide Web)*.

<b>Função</b>	<b>Descrição</b>
Alarme	Alarme para aviso de data especificada.
Agendamento de compromissos	Agendamento de datas e horários.
Lista de tarefas	Lista de tarefas pendentes.
Calendário	Consulta a calendário.
Caderno de endereços	Armazenamento de endereços.

Tabela 11: Funções da agenda eletrônica

A tabela de horários é utilizada para informar os horários de cada usuário a fim de facilitar o agendamento de encontros. O usuário deve preencher um formulário informando sua disponibilidade, para que os demais possam consultar uma tabela de horários via *web*.

O quadro de avisos é utilizado para divulgação de recados e avisos de caráter geral para todos os usuários. A utilização do quadro de avisos consiste no preenchimento de um formulário, onde são informados o assunto, a data, a data de vencimento do aviso e um recado. O aviso será divulgado em uma página *web*, sendo removido automaticamente após a data de vencimento estabelecida.

Para realizar a distribuição de mensagens entre usuários do sistema é necessário um serviço de mensagens, que pode ser por exemplo um serviço de correio eletrônico. Já existem diversas ferramentas que implementam o serviço de correio eletrônico, sendo possível aproveitá-las.

O agendamento de consultas remotas pode ser feito através de um sistema de agendamento eletrônico, em que se faz a solicitação de uma consulta para um determinado dia e horário. O Sistema se encarrega de verificar a viabilidade do horário sugerido (através da tabela de horários e das consultas já agendadas), comunicando o usuário destino (quem desejamos consultar) que deverá confirmar a data.

O *chat* oferece uma forma interativa de comunicação entre usuários, facilitando a tomada de decisões, por ser muito mais rápida se comparada ao correio eletrônico por exemplo.

A vídeo conferência permite a comunicação entre participantes através de conversa e visualização da imagem de cada um dos participantes na tela de seus computadores. Esta ferramenta é interessante segundo os seguintes aspectos: identificação visual dos participantes e visualização dos participantes permitindo a observação de seus gestos.

A identificação visual dos participantes proporciona alguma segurança com relação a identificação das pessoas além de fornecer informação visual adicional. A transmissão apenas da voz, imagens isoladas ou texto resulta em alguma perda de informação (gestos, expressões e comportamento), que não ocorre quando se utiliza a vídeo conferência, permitindo ao especialista remoto orientar mais facilmente o médico local

na análise de seus dados ou mesmo no exame de um paciente.

Apesar de importante, a vídeo conferência não vai ser implementada nesta pesquisa, por necessitar de equipamentos e programas específicos e por existirem diversos produtos comerciais que a implementam. É razoável pensar na futura integração de algum produto já existente ao ambiente cooperativo, evitando esforços redundantes. Também vale a pena lembrar que um dos objetivos deste trabalho é auxiliar o diagnóstico por imagens com o uso de computadores em rede, sendo que a falta do serviço de vídeo conferência não afeta esta pesquisa.

Todas as funcionalidades deste grupo devem ser protegidas ao acesso de pessoas não autorizadas, sendo necessária a utilização de senhas de acesso, com diferentes categorias de permissão conforme a função a ser executada. Os dados transmitidos pela rede também devem ser protegidos contra acessos indevidos.

#### **3.3.3.4 Acesso a dados compartilhados**

O sistema cooperativo de auxílio ao diagnóstico médico trabalha sobre dados e informações armazenados em algum ponto da rede. Estes dados são coletados através de exames ou consultas médicas e armazenados em uma base de dados do hospital.

É importante que o sistema cooperativo possa interagir com esta base de dados, pois ao integrar o sistema com a base de dados hospitalar torna-se possível ao especialista consultar não apenas os exames de um determinado paciente, mas todo o seu histórico médico. Além da possibilidade de consultar a ficha médica de um paciente, torna-se possível obter estatísticas a respeito de doenças e tratamentos e também consultar casos semelhantes.

Os dados dos pacientes devem ser armazenados em um servidor de banco de dados, sendo que nesse trabalho será utilizado um protótipo de base de dados centralizada por questão de simplicidade.

Uma característica importante em um sistema de telemedicina é o acesso ao conhecimento. Muitas vezes o médico necessita consultar bases de dados de casos e bibliografia específica, que servem de referência para o diagnóstico de um paciente. Ao tornarmos fácil o acesso a estas informações, o processo de diagnóstico torna-se mais

rápido e eficiente.

A criação de aplicações para acesso ao conhecimento é relativamente simples, se comparada ao trabalho de alimentação da base de dados com o conhecimento, que requer um vasto período de tempo. A alimentação da base de conhecimento requer um processo de triagem, digitalização e interpretação da informação, para que então seja oferecida em um formato de mais fácil entendimento por parte dos usuários.

#### **3.3.4. Serviços de processamento de dados radiológicos**

Esta categoria de serviços é responsável pela aplicação de algoritmos para processamento dos exames com a intenção de refinar os dados, de forma a obter informações que geralmente não são visíveis com a representação na forma de imagem ou gráficos.

Tais serviços devem ser oferecidos na forma de um servidor localizado em uma máquinas de alto desempenho. A utilização de máquinas de alto desempenho prende-se à necessidade de processamento intensivo por parte das aplicações de processamento de dados médicos e ao grande volume destes dados.

Essa categoria de serviços gera imagens usando parâmetros selecionados pelo usuário, sendo estes parâmetros enviados em requisições ao servidor, que retorna um resultado ao usuário e seu grupo de trabalho. O processamento funciona apenas em volumes ou imagens carregados na memória do servidor, devendo ser transparente ao usuário o fato de seus dados estarem sendo processados remotamente.

Os dados a serem processados são obtidos da base de dados hospitalar ou fornecidos pelo próprio usuário. Quando consultada a base de dados hospitalar, devem ser fornecido o nome do paciente e a identificação do exame escolhido, para localização no banco de dados.

A Tabela 12, descreve os serviços oferecidos dentro desta categoria:

<b>Tipo</b>	<b>Funcionalidade</b>	<b>Descrição</b>
Imagem	Crescimento de região	Cresce uma região de uma imagem, segundo parâmetros estabelecidos.
Imagem	Interpolação	Ampliação de uma imagem segundo um fator fornecido
Volume	Crescimento de região	Cresce uma região de um volume, segundo parâmetros estabelecidos.
Volume	<i>Rendering</i>	Gera uma imagem 3D a partir de um volume

Tabela 12: Serviços de Processamento de Imagens

O acesso a essa categoria de serviços deve ser totalmente transparente ao usuário, que não deve conhecer detalhes de localização ou de como seus dados são processados. Seu único interesse é obter de forma mais rápido e precisa os resultados.

### **3.4. Arquitetura do Sistema**

Foi escolhido trabalhar com uma arquitetura cliente/servidor e objetos distribuídos. A arquitetura do sistema prevê a existência de um servidor responsável por gerenciar o trabalho cooperativo, um servidor de alto desempenho responsável por processar os dados médicos e uma interface cliente, onde as imagens resultantes são visualizadas. A opção por objetos distribuídos considera o crescente uso da tecnologia de objetos, que proporciona modularidade e estruturação ao código, sendo escolhido trabalhar com a padronização CORBA de objetos distribuídos.

Em linhas gerais os serviços inicialmente oferecidos pelo ambiente cooperativo dividem-se entre dois servidores:

\* **O servidor de CSCW:**

Controla o acesso ao sistema. É dividido em três módulos: um responsável pelo controle de acesso de usuários ao sistema, um módulo responsável pelo gerenciamento de grupos de trabalho e um último módulo responsável por consultas ao banco de dados com informações sobre usuários;

\* **Interface de trabalho cooperativo:**

Interface gráfica para a análise de dados e trabalhos cooperativo. Estende os recursos do Servidor de CSCW, através de componentes que gerenciam localmente o trabalho em grupo e dados compartilhados.



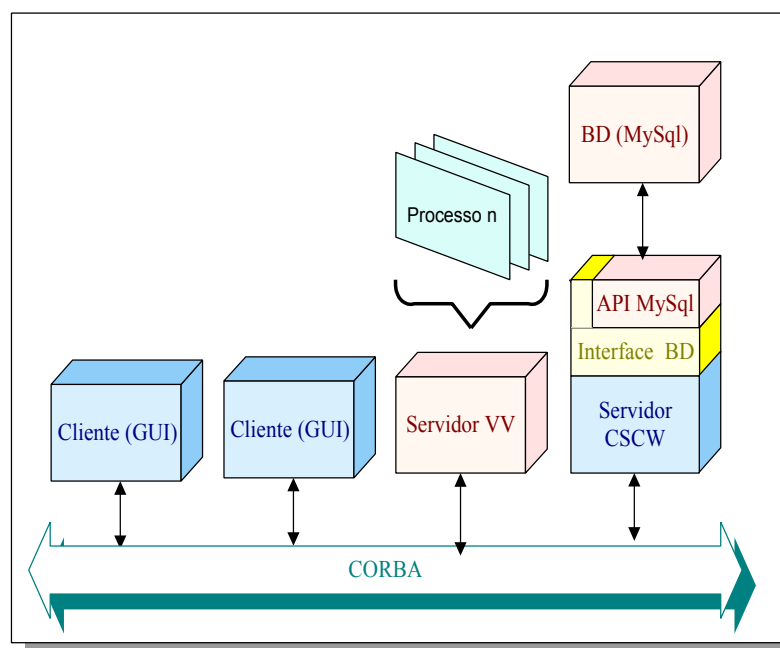
\* **Servidor VV (Servidor de Visualização Volumétrica):**

Oferece os serviços de alto desempenho, devendo gerenciar arquivos compartilhados em memória e fornecer serviços de processamento de imagens e volumes. Este servidor é construído sobre uma biblioteca de visualização volumétrica, o PVV (*Parallel Volume Visualisation Toolkit*), que torna transparente a programação em máquinas com arquitetura paralela e/ou distribuída. Assim, este servidor deve rodar em máquinas de alto desempenho.

O ambiente proposto funciona sobre a plataforma CORBA, onde cada elemento, cliente ou servidor é constituído por componentes. A ideia do sistema é criar diversos componentes básicos de programação que possam ser encaixados de forma a montar uma aplicação completa.

A escolha por uma arquitetura cliente/servidor e objetos distribuídos, permite o acréscimo fácil de novos serviços ao ambiente encapsulados em novos servidores, ou como expansão dos servidores já existentes. Outra vantagem dos objetos distribuídos, é a fácil manipulação dos mesmos, de forma que se possa atualizar versões de qualquer componente do ambiente, sem parar todo o sistema.

A Figura 16 mostra a arquitetura do sistema de forma mais detalhada.



**Servidor VV:** Servidor de processamento de dados;

**BD:** Banco de dados de pacientes, dados médicos e usuários;

**Servidor CSCW:** Fornece os recursos para localização de usuários e grupos de trabalho;

**Cliente/Servidor:** Uma interface para o trabalho cooperativo comporta-se como cliente para os servidores CSCW e VV e como servidora para outras interfaces em um grupo de trabalho;

**Cliente:** Interface gráfica para análise de dados médicos e trabalho cooperativo.

Figura 16: Arquitetura do sistema

A parte do sistema responsável pelo trabalho cooperativo, será implementada sobre o CORBA, enquanto as demais partes utilizam uma arquitetura mista, que utilizam o CORBA para encapsular aplicações já existentes.

A implementação do Servidor VV diretamente sobre CORBA não é viável, devido a arquitetura utilizada pelo PVV, que duplica a aplicação em cada nó onde ela deverá ser executada, além de haver grandes possibilidades de conflito entre a estrutura CORBA e do PVV, no que diz respeito ao gerenciamento de *threads* e mensagens. Como solução a este problema optou-se por montar um servidor que trabalha sobre a estrutura do CORBA, porém responsável apenas por iniciar tarefas em processos separados. Estas tarefas realizam algum processamento sobre os dados.

O acesso ao banco de dados é feito através de uma API específica para este fim encapsulada de forma a permitir chamadas de métodos remotos em CORBA, permitindo assim maior flexibilidade na aplicação. Tendo encapsulado a API original torna-se mais fácil trocar a camada mais baixa, que cuida da comunicação com o banco de dados, com o mínimo de alterações na aplicação.

#### **3.4.1. Servidor de CSCW**

Podemos identificar duas categorias básicas de serviços a serem oferecidas pelo servidor de CSCW:

- \* **Serviços de identificação e controle de acesso**

Para utilizar os serviços disponíveis no ambiente o usuário deverá identificar-se perante o sistema, que permitirá ou não o acesso aos serviços, e verificará qual a política de acesso adotada para cada usuário;

- \* **Serviços de gerenciamento de grupos de trabalho**

Estando identificado no sistema, o usuário pode usar os serviços do sistema, segundo suas permissões de acesso, e participar de sessões de trabalho cooperativo. Nas sessões de trabalho cooperativo, deve haver um controle dos participantes de cada grupo de trabalho, e dos dados acessados em cada grupo.

Os serviços são disponibilizados através de uma interface de acesso, isto é, o ambiente enxerga estes serviços através de métodos e objetos definidos no servidor.

Para facilitar o gerenciamento dos usuários, o sistema é dividido em domínios, onde cada domínio possui o seu próprio servidor de CSCW. A divisão entre domínios considera fatores como local físico, concentração dos usuários e acessos a determinados serviços. Por exemplo: se temos um hospital A e um hospital B, é natural criar um domínio para cada hospital e cadastrarmos os usuários de cada hospital nos respectivos domínios.

Os servidores de CSCW podem interagir, para oferecimento de serviços não disponíveis em um determinado domínio e para estabelecimento de sessões cooperativas entre domínios diferentes. Esta interação deve considerar as permissões do usuário que requisita o serviço e permissões do próprio domínio.

O gerenciamento dos grupos de trabalho é realizado dinamicamente. Cada vez que um usuário se identifica no sistema ele fica disponível para trabalho cooperativo. Qualquer usuário identificado, pode verificar os demais usuários e solicitar sessões cooperativas. Um usuário pode iniciar um novo grupo, ao solicitar o trabalho conjunto com um outro usuário, ou entrar em um grupo já existente. O grupo ou usuário solicitado para trabalho em conjunto, pode vetar a entrada de um novo participante, sendo que no caso de grupos, o veto é responsabilidade do líder de sessão.

Um líder de sessão, é responsável por coordenar as atividades do grupo. Todas as ações realizadas pelo líder são refletidas para os demais participantes. Assim se o líder pede visualiza uma fatia de uma tomografia, todos os demais participantes do grupo verão a mesma imagem. A qualquer momento, o líder do grupo pode passar o cargo para um outro participante. O líder é responsável pela distribuição dos dados, porém todos os participantes pode interagir sobre estes dados. É tarefa do líder permitir a adesão de novos participantes ao grupo.

Após definir em linhas gerais o modelo de interação do servidor de CSCW, os próximos itens descreverão a arquitetura deste servidor. A arquitetura será descrita usando o padrão para modelamento UML e a parte referente a bancos de dados será modelada como banco de dados relacional.

#### **3.4.1.1 Projeto do Banco de Dados**

Para controle de usuários, domínios e dados é necessário criar um banco de dados.

Neste banco de dados devem ser armazenados os dados referentes a nome, senha e permissões de usuários, e nome e endereços dos domínios.

O banco de dados proposto deve solucionar apenas os problemas de identificação de usuários, organização dos dados e localização de domínios, visto que pretende apenas atender à exigências de uma aplicação protótipo, porém o ideal seria trabalhar diretamente com a base de dados hospitalar.

Por simplicidade é considerado que os dados de pacientes serão apenas referentes a exames e textos descritivos dos exames. A Figura 17 mostra o modelo do banco de dados a ser implementado.

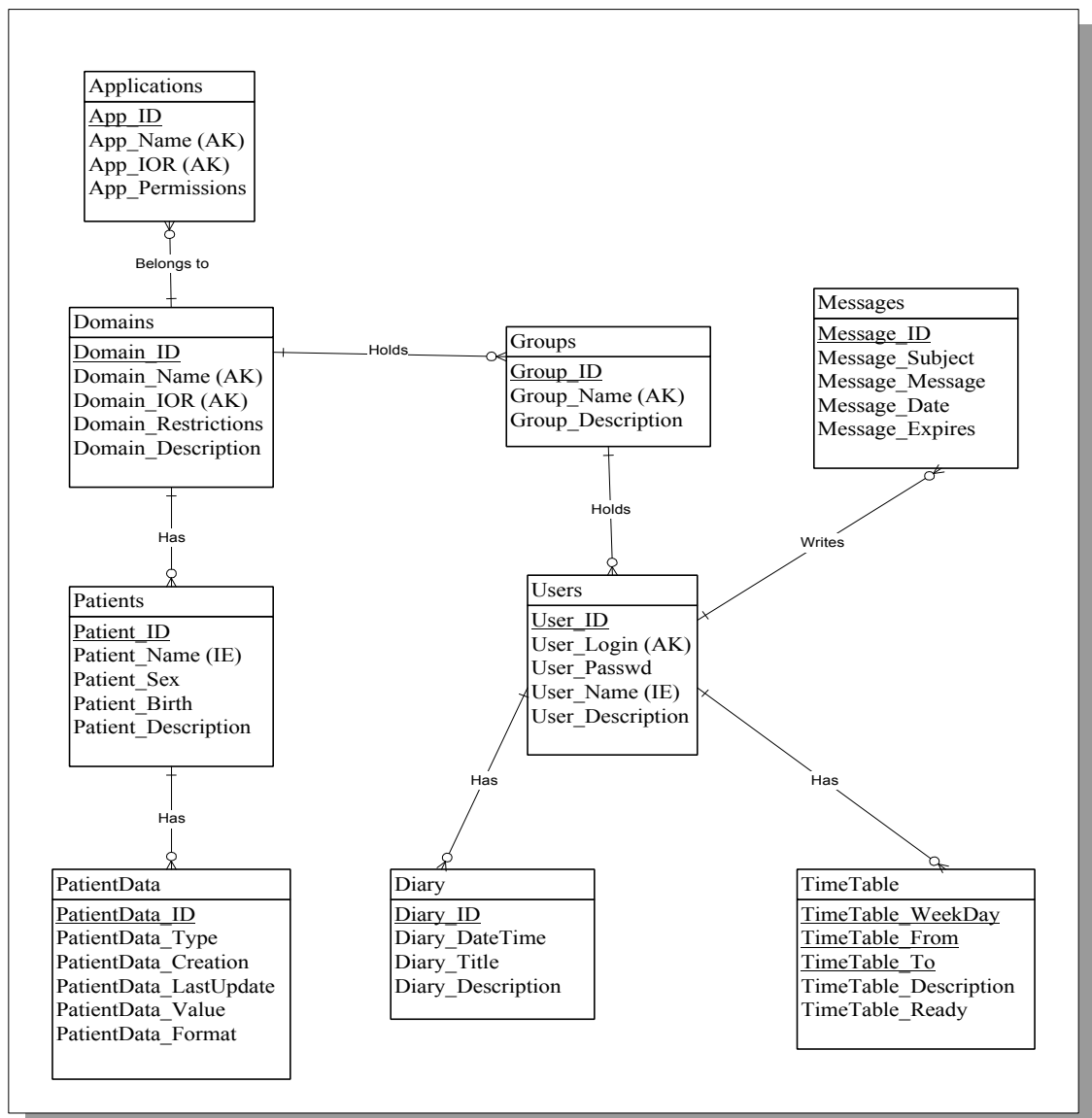


Figura 17: Esquema do Banco de Dados de Usuários

Cada usuário tem um nome para *login*, uma senha, permissões de acesso e informações adicionais. Além das informações descritas acima, o usuário deve pertencer a um grupo, que pode referir-se a um projeto, departamento ou categoria de usuários. Este conceito é semelhante ao usado nas contas de usuários *Unix*, e servem para definir as pessoas que trabalham juntas, e os níveis de acesso ao sistema. Os usuários e grupos pertencem a um domínio, sendo que cada domínio é responsável por gerenciar dados apenas de seus usuários. É feita esta restrição para evitar réplicas de dados e para manter a integridade e consistência dos dados. Isto não impede que usuários de um domínio acesse a outro, o que apenas depende da política adotada de interação entre os domínios.

Por fim, os arquivos de pacientes (arquivos de volume, imagem ou texto), também são armazenados no banco de dados, facilitando a manipulação e localização dos mesmo por parte de clientes remotos. Este mecanismo vai permitir localizar os arquivos de exames no servidor através dos nomes dos pacientes, e não por um sistema de arquivos e diretórios. O cadastro de usuários domínios e pacientes é feito através de uma interface gráfica específica para este fim.

#### **3.4.1.2 Diagrama de objetos**

O servidor de CSCW é constituído por três módulos básicos: um módulo responsável por gerenciar os usuários conectados ao ambiente cooperativo, um módulo responsável pelos grupos de trabalho e outro por gerenciar a conexão com o banco de dados para verificação de usuários.

A Figura 18 mostra as classes principais do Servidor CSCW. As classes em verde representam classes de uso geral. As classes em azul pertencem ao módulo de gerenciamento de grupos. As classes em salmão pertencem ao módulo de gerenciamento de usuários. A classe em lilás pertence ao módulo de comunicação com o banco de dados e por fim as classes em amarelo constituem a aplicação servidora, englobando os módulos e suas classes.

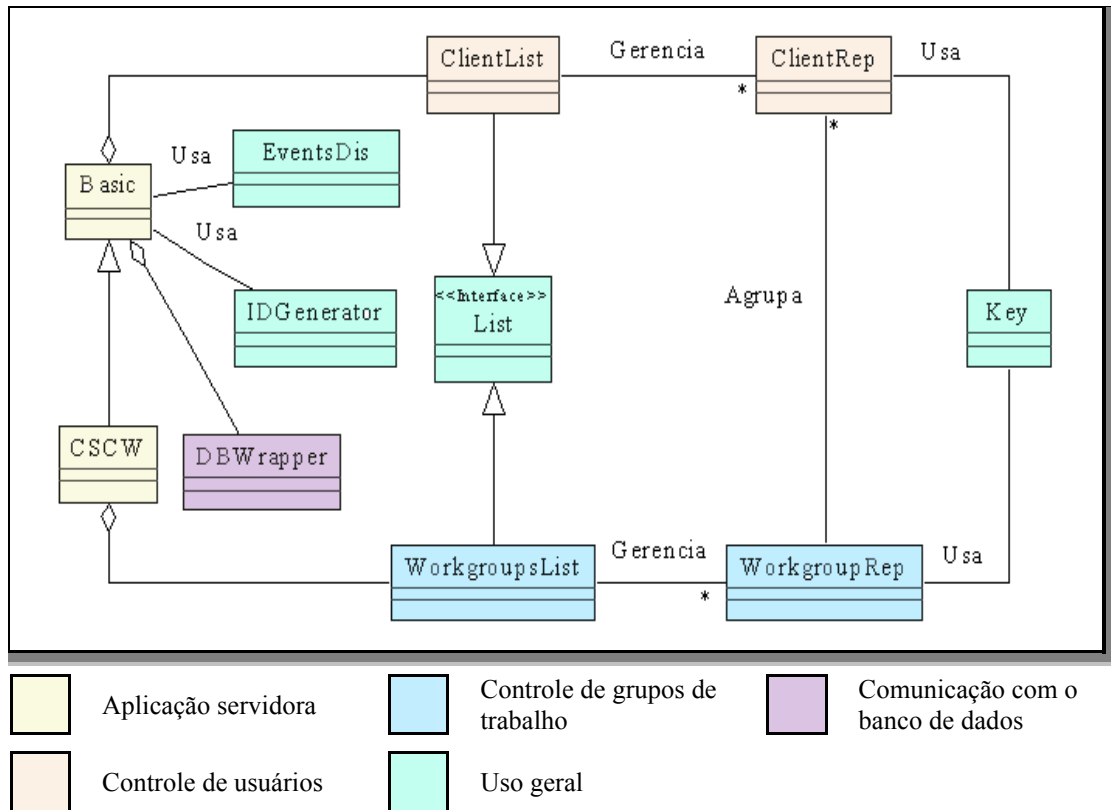


Figura 18: Diagrama de classes do Servidor de CSCW.

### 3.4.2. Servidor VV (Servidor de Visualização Volumétrica)

O Servidor VV é dividido em dois grandes módulos: Um módulo responsável por entrada/saída e gerenciamento dados outro módulo responsável por processar estes dados. Cada módulo constitui um gerenciador (*manager*).

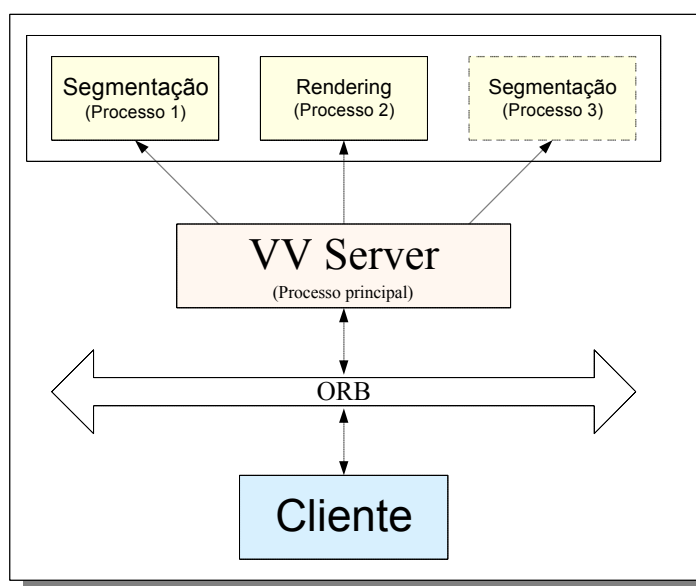
O módulo de gerenciamento dos dados (*Data Manager*) trata apenas arquivos de exames (volumes e imagens), devendo gerenciar o acesso aos dados e o ciclo de vida dos mesmos. Acoplado a este módulo há um componente responsável por intermediar a obtenção dos dados, que podem ser negociados com uma aplicação remota ou podem ser obtidos de um arquivo em disco. A aplicação remota pode ser um banco de dados, ou qualquer outro tipo de aplicação.

O módulo de processamento de imagens e volumes (*VV Manager*) é responsável pelos serviços de processamento das imagens e volumes. Este módulo encapsula os algoritmos de visualização construídos sobre a biblioteca PVV (*Parallel Volume Visualisation Toolkit*). O PVV tem como objetivos oferecer um conjunto de ferramentas específicas para o desenvolvimento de técnicas de visualização volumétrica, explorar

recursos de processamento paralelo e garantir a portabilidade do código para diferentes ambientes de execução sequenciais e paralelos.

Devido a particularidades da arquitetura da biblioteca PVV não é viável sua aplicação direta junto a estrutura CORBA. A principal característica que inviabiliza o uso das aplicações PVV sob o CORBA é o modelo de programação utilizado, o SPMD (*Simple Program Multiple Data*), onde o mesmo programa é executado em cada unidade de processamento. Cada unidade de processamento conhece seu próprio contexto, podendo decidir quais tarefas executar. Se pensarmos em adicionar mais uma camada de comunicação, o CORBA acima da camada PVV, seriam necessárias alterações na estrutura interna da biblioteca, além de que toda a estrutura CORBA e os objetos contidos no servidor seriam replicados para cada unidade de processamento, porém utilizada em apenas uma das unidades.

A estrutura escolhida para o servidor de visualização divide o servidor em vários processos dentro da máquina que hospeda o servidor. O processo principal, que implementa o servidor, é executado sobre a arquitetura CORBA. Este processo é responsável por disparar novos processo que implementam os algoritmos de processamento volumétrico ou de imagens. Os algoritmos são implementados em pequenos programas, sendo que o servidor apenas os executa e resgata os resultados do processamento.



O Servidor VV oferece uma interface IDL que é acessada pelos clientes que requisitam um serviço. A requisição executa um comando na máquina servidora iniciando um programa que realiza o processamento necessário nos dados.

Figura 19: Estrutura de acesso ao Servidor VV.

A passagem dos dados para o programa responsável pelo processamento pode ser feita por uma *pipe* ou um *socket*. Ao terminar o processamento o resultado é devolvido para o processo principal então repassado ao cliente.

### 3.4.2.1 Diagrama de Objetos

O Servidor VV é dividido em dois módulos principais: O *Data Manager*, responsável por gerenciar os arquivos de dados bem como compartilhamento dos mesmos em memória e o *VV Manager*, responsável pelos algoritmos de processamento de imagens ou volumes.

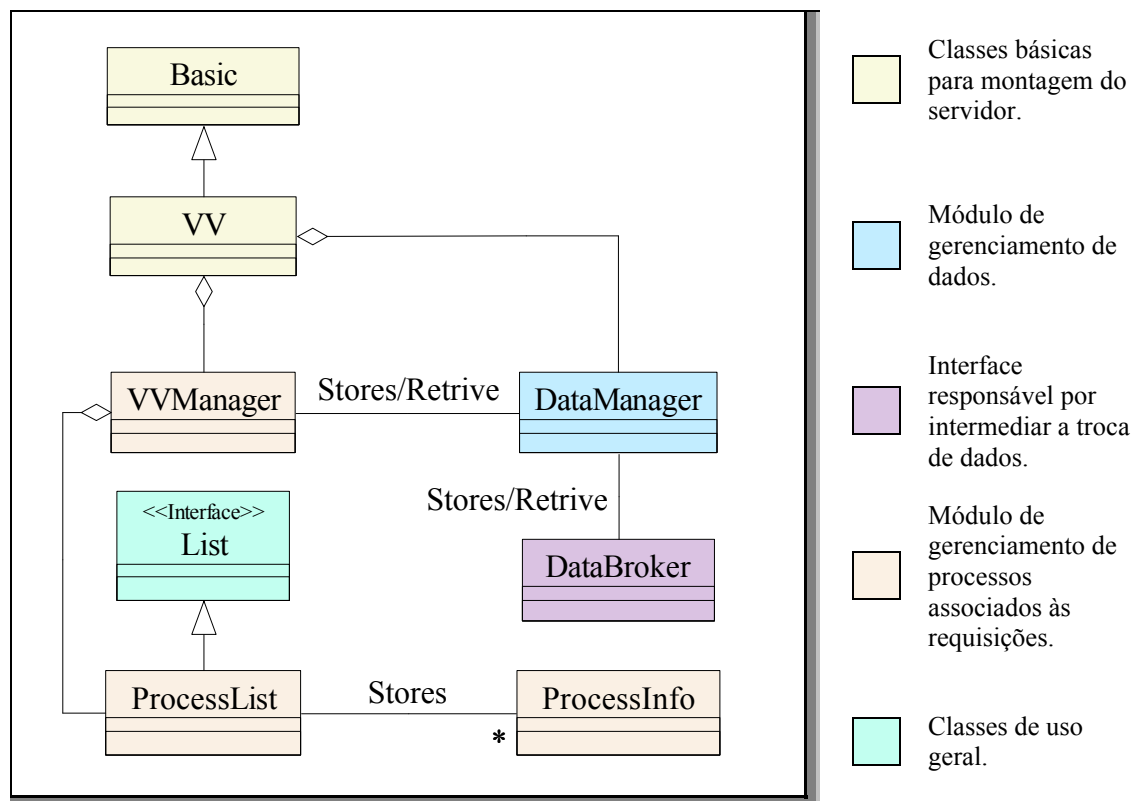


Figura 20: Diagrama de classes do Servidor VV.

O gerenciador de dados referencia uma classe (*DataBroker*) que intermedia a obtenção dos dados, que podem ser obtidos de um objeto remoto, de um banco de dados ou de um arquivo na própria máquina que hospeda o servidor.

O *VVManager* gerencia os processos iniciados para realização de uma tarefa requisita por um cliente. A informação referente a cada processo é armazenada em uma lista dentro do *VVManager*.



### 3.4.3. Interface Cliente

O acesso aos serviços do ambiente é feito através de uma interface gráfica que deve permitir o trabalho conjunto de análise das imagens geradas localmente ou em um servidor de alto desempenho.

A interface cliente é constituída por um módulo de interface gráfica, responsável pelos *Widgets* e telas (GUI), um módulo de CSCW (*CSCW Manager*), responsável pelo trabalho cooperativo no nível da interface e pelo acionamento dos serviços de CSCW, um módulo de visualização (*VV Manager*), responsável por tratar as imagens a serem visualizadas e fazer a comunicação com o servidor de alto desempenho e um módulo de gerenciamento dos dados compartilhados (*Data Manager*).

O compartilhamento de dados é feito da mesma forma que o Servidor VV, onde uma classe de comunicação é encapsulada em um classe de formatação dos dados de forma compreensível à aplicação. A Figura 21 exemplifica a troca de dados entre duas aplicações dentro de uma sessão cooperativa.

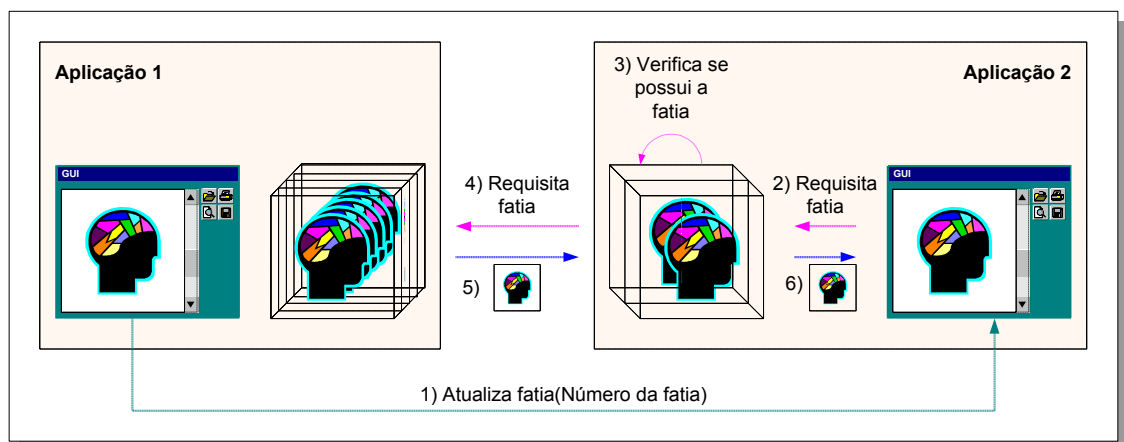


Figura 21: Troca de dados entre aplicações.

Na Figura 21 temos duas interfaces gráficas com um volume aberto em memória. A *Aplicação 1* possui os dados a serem compartilhados, enquanto a *Aplicação 2* aos poucos preenche seu *buffer* até obter uma cópia completa dos dados da *Aplicação 1*. A cópia dos dados é feita segundo a demanda na *Aplicação 2*.

A Figura 21 descreve a seguinte seqüência de eventos:

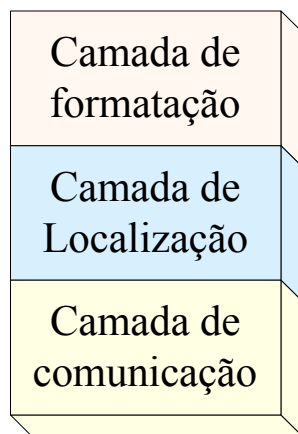
1. A *Aplicação 1* envia um comando de sincronização para a *Aplicação 2*. A

*Aplicação 2* deve executar um *Stackflow* e mostrar a mesma fatia da tomografia;

2. A *Aplicação 2* requisita a fatia para o objeto que representa o volume;
3. O objeto que representa o volume verifica que não possui a fatia do volume requisitada;
4. O objeto que representa o volume na *Aplicação 2* requisita a fatia para o objeto que representa o volume da *Aplicação 1*;
5. A *Aplicação 1* transfere a fatia para a *Aplicação 2*;
6. A *Aplicação 2* atualiza a tela na interface gráfica.

Segundo este modelo, um dado pode ser remoto ou local, uma vez que um exame é compartilhado por diversas pessoas em uma sessão cooperativa. O acesso ao conteúdo do dado é transparente para aplicação, que não sabe diferenciar um dado remoto de um local. Assim, o compartilhamento é feito através de uma estrutura onde há um objeto principal, que representa os dados completos, e objetos que são *proxies* para o principal.

A Figura 22 mostra estrutura do componente *proxy* para dados:



A camada de formatação representa os dados em um formato conhecido da aplicação, como por exemplo um volume ou uma imagem. A camada de localização se encarrega de verificar se o dado é local ou remoto e no caso de dado remoto de localiza-lo no sistema distribuído. A camada de comunicação faz a troca dos dados pela rede.

Figura 22: Representação de dados.

Na aplicação cooperativa deste projeto, dois tipos de dados são utilizados na interface: imagens e volumes. Seguindo a estrutura descrita anteriormente, teremos uma classe para representação de volumes e outra para as imagens. Segundo a estrutura proposta na Figura 22, a representação de volume ou imagem situa-se na camada mais alta (camada de formatação) sendo mantida as duas camadas inferiores para ambas representações.

A representação do dado ou o *proxy* para o dado original residem no *Data Manager*, responsável por gerenciar o ciclo de vida do dado.

O *CSCW Manager* deve gerenciar a conexão com o Servidor de CSCW, e a sessão cooperativa. Ao abrir a interface, deve ser executado o *login* no servidor, permitindo assim aos demais usuários do ambiente cooperativo localizar uns aos outros. Caso se deseje tomar parte de uma sessão, é feito o pedido de entrada em um grupo. Estabelecida a sessão cooperativa, os usuários interagem e compartilham seus dados.

A sincronização entre aplicações pertencentes à uma sessão é feita através de eventos distribuídos via serviço de eventos do CORBA (*CosEvent Service*). Desta forma, pretende-se trocar o mínimo possível de dados pela rede, evitando sua sobrecarga. Somente quando necessário imagens e volumes são enviados de uma aplicação para outra, segundo o modelo de *proxy* para dados descrito anteriormente.

A sessão cooperativa pode ser encerrada a qualquer momento, dependendo apenas do interesse dos participantes.

Quando é realizada uma requisição ao servidor de processamento de imagens e volumes, o objeto que representa o dado conecta-se a um *proxy*, de estrutura idêntica ao *proxy* utilizado na interface cooperativa, residente no servidor de visualização. Os dados são totalmente transferidos para o servidor que realiza o processamento e retorna o resultado ao cliente.

#### **3.4.3.1 Diagrama de classes**

A interface gráfica cliente é composta por quatro gerenciadores: *VV Manager*, *GUI Manager*, *Data Manager* e *CSCW Manager*. O *VV Manager* é responsável por gerenciar as requisições ao Servidor VV, o *GUI Manager* é responsável por gerenciar os objetos da interface gráfica e seus eventos, o *Data Manager* é responsável por gerenciar o ciclo de vida dos dados utilizados na interface e o *CSCW Manager* é responsável por gerenciar o trabalho em grupo.

A Figura 23 mostra as classes principais da interface cliente. Neste diagramas são omitidas as classes referentes a objetos de interface, objetivando apenas mostrar a estrutura da aplicação.

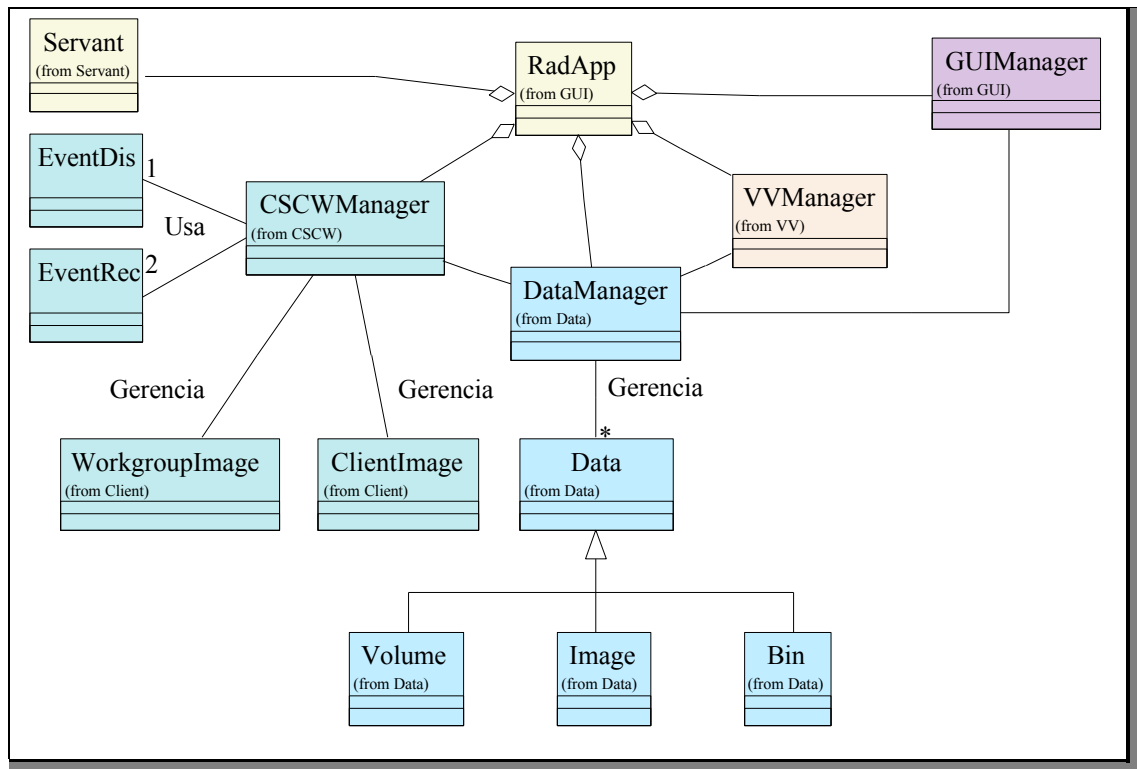


Figura 23: Diagrama de classes da interface Cliente.

A classe *Servant* inicializa a estrutura do CORBA, encarregando-se de iniciar a ORB, o BOA e conectar-se ao serviço de nomes. Deve haver apenas uma instância deste objeto em uma aplicação. A classe *RadApp* constitui a aplicação principal, que engloba as demais classes.

Os dados (volumes, imagens ou dados binários) derivam de uma classe básica (*Data*) que implementa os níveis de localização e comunicação de um componente que representa os dados.

Por fim, o *CSCW Manager* possui uma referência para um objeto do tipo *ClientImage* e outra para um *WorkgroupImage*. Estas duas classes constituem uma imagem da representação do cliente e do grupo de trabalho no servidor de CSCW. As imagens são utilizadas para sincronizar o servidor e os clientes, para auxiliar na localização de objetos entre aplicações e para registrar o interesse em eventos entre aplicações dentro de uma mesma sessão cooperativa.

### 3.4.4. Ferramentas de Auxílio ao Trabalho Cooperativo

Além da interface para análise de dados radiológicos, é necessário criar algumas ferramentas que auxiliam na organização do ambiente cooperativo, dos grupos de trabalho e dos usuários do sistema. Basicamente constituem interfaces (formulários) para utilização do banco de dados.

Segundo a especificação funcional, devemos ter ferramentas para agendamento de eventos, tabela de horários dos usuários, agenda do usuário, quadro de avisos, serviço de mensagens, *chat* e videoconferência.

A maior parte destas ferramentas podem ser construídas para funcionarem em ambiente *web* na forma de páginas. A construção sobre *web*, facilita o acesso das ferramentas, não sendo necessário instalar novos programas na máquina de cada usuário além de possibilitar o acesso de qualquer local.

Os serviços de quadro de avisos, agendamento de eventos, agenda de usuário e tabela de horários serão implementados segundo a Figura 24.

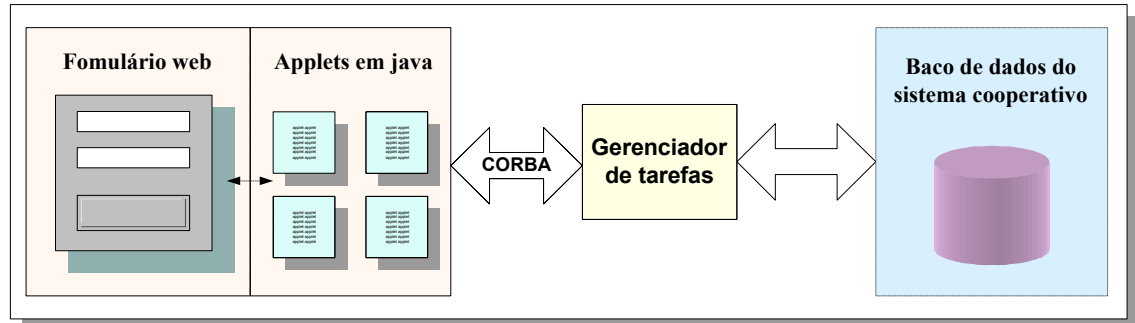


Figura 24: Esquema de funcionamento das ferramentas auxiliares

Na Figura 24 verificamos a existência de um formulário *web* que serve de interface para o serviço desejado, *applets* que fazem a comunicação com Gerenciador de Tarefas que por sua vez comunica-se com o banco dados. O gerenciador de tarefas é responsável por processar e filtrar os dados antes de enviá-los ao formulário.

### 3.4.5. Ferramentas de Implementação

Para implementação é necessário especificar algumas ferramentas e plataforma de trabalho. Como plataforma de trabalho foi escolhido Pc/Linux, por se tratar de uma plataforma barata, estável e de uso bastante comum.

Como linguagens de programação foram escolhidas C, C++ e JAVA. O C e C++ são utilizados na confecção dos servidores, que precisam de um desempenho melhor. A linguagem JAVA foi escolhida para implementação dos clientes, por ser facilmente portátil para diversas plataformas e por possuir uma biblioteca para criação de interfaces, além de facilidade de trabalho com objetos remotos e web.

Com relação às ferramentas, os problemas a serem atacados envolvem:

\* **Gerenciamento de Banco de Dados:**

O Cadastro de usuários e domínios envolve a utilização de um Banco de dados. Foi escolhido trabalhar com o MySQL, um banco de dados relacional de uso livre, que oferece APIs para programação em C, C++ e JAVA. Possui implementação para diversas plataformas Unix e para *Windows 95/NT*, além de utilizar o padrão SQL92<sup>7</sup>.

\* **Padronização de objetos distribuídos;**

Para trabalho com objetos distribuídos foi escolhido o CORBA. A implementação CORBA escolhida foi a *ORBacus 3.x*, também conhecida como *OminiBroker*. Esta implementação suporta os sistemas operacionais Windows 95 e NT, além de suportar diversas versões de Unix (Solaris, Linux, Irix, entre outros). Trata-se de uma implementação CORBA de uso livre para fins não comerciais, possuindo mapeamento IDL para C++ e JAVA. Possui os serviços básicos de *Naming*, *Event* e *Property*, *Trading* e *Time* especificados pelo CORBA. Aceita concorrência através do modelo de *Threads* permitindo a construção de servidores iterativos, além de possuir ferramentas para documentação automática de código semelhantes ao JavaDoc.

\* **Implementações de algoritmos de visualização volumétrica:**

A implementação dos algoritmos de processamento de imagens e volumes utiliza uma biblioteca de visualização volumétrica denominada PVV. Esta biblioteca oferece recursos para programação de algoritmos de visualização volumétrica em máquinas de alto desempenho (máquina paralelas). Será utilizada na confecção do servidor de processamento de imagens. Trata-se de uma biblioteca para linguagem C. Detalhes a respeito da biblioteca PVV são

---

<sup>7</sup>**SQL (Simple Query Language):** Linguagem utilizada para definição de consultas em nacos de dados relacionais. O SQL92 é uma padronização ISO para a linguagem SQL.

encontrados em [Zuffo97]. Muitos algoritmos necessários já se encontram implementados sobre esta biblioteca.

\* **Conjunto de ferramentas para desenvolvimento de interfaces gráficas:**

A interface cliente para acesso ao ambiente deverá ser construída em linguagem JAVA e a biblioteca *Swing*. O JAVA foi escolhido por se tratar de uma plataforma bastante difundida além de apresentar as funcionalidades interessantes ao projeto, como orientação a objetos, portabilidade e baixo custo.

### **3.5. Resumo do Capítulo**

Foi apresentado neste capítulo a descrição do sistema cooperativo para medicina de forma a detalhar as fase de concepção e projeto do mesmo. Cada seção procurou estabelecer as características esperadas em um sistema de tal natureza e as soluções propostas.

Os próximos capítulos descreverão a implementação e resultados de um protótipo baseado nas especificações deste capítulo.

## **4. Considerações sobre a Implementação de um Sistema Colaborativo para Teleradiologia**

### **4.1. Introdução**

Nos capítulos anteriores foi realizada uma descrição conceitual de uma proposta de ambiente cooperativo adequado para a teleradiologia. Neste capítulo apresentamos as considerações sobre a implementação do sistema conceitual proposto.

Devido a diversos fatores como tempo e esforço de programação decidiu-se implementar um protótipo que atenda os aspectos necessários para validação científica desta proposta.

Procurou-se modelar o sistema implementado de forma a facilitar o acréscimo das demais partes restantes e possíveis atualizações e melhoramentos. Esta abordagem é importante devido rápido ciclo tecnológico vivido atualmente, onde a circuitaria e os ambientes de programação tornam-se obsoletos em um período de tempo relativamente pequeno.

Durante a pesquisa e o desenvolvimento deste projeto, muitos avanços em pesquisa e inovações tecnológicas surgiram, sendo necessário sempre manter o estado-da-arte atualizado, dentro do foco principal de investigação e análise científica.

Independentemente da tecnologia e recursos utilizados, o objetivo do trabalho cooperativo se mantém: alcançar melhores condições de trabalho e resultados através do compartilhamento de tarefas e conhecimento entre diversos indivíduos e organizações.

### **4.2. Considerações Gerais**

Segundo a proposta inicial de pesquisa, considerou-se importante a integração de ferramentas heterogêneas para a construção de um ambiente cooperativo para teleradiologia. Seguindo esta proposta, decidiu-se construir toda a infra-estrutura do sistema cooperativo sobre padrões de comunicação entre computadores em redes e por construir ferramentas portáteis e que possam interagir independentemente do ambiente e programação e da circuitaria utilizada.



A arquitetura proposta neste trabalho objetiva não apenas inserir novas tecnologias e processos ao ambiente de trabalho médico, mas também utilizar os recursos de tecnologia da informação já existentes neste ambiente, seja através de redes de comunicação de dados locais disponíveis, seja através da infra-estrutura nacional de telecomunicações existente.

A principal razão desta escolha é que uma instituição de saúde constrói e estrutura seu ambiente de informática através de investimentos realizados ao longo de muitos anos, entretanto o avanço tecnológico é muito rápido, tornando-se complexo o acompanhamento destes avanços, tanto financeiramente como em termos de trabalho e tempo gastos. Lembrando que em muitos casos, uma estrutura antiga ainda atende aos requisitos e objetivos do grupo de trabalho.

É claro que não podemos nos desfazer de todo um trabalho já realizado em nome das inovações, mas também não podemos deixá-las de lado, correndo o risco de ficarmos estagnados no tempo. Toda nova tecnologia, se bem utilizada, trás muitos benefícios à qualidade do atendimento médico. Neste caso, torna-se necessário muito bom senso, de forma a não permitir a simples automatização de processos ineficientes, mas aperfeiçoar as técnicas já existentes com o objetivo de melhorar o atendimento médico.

Dentro deste contexto, surge uma primeira grande questão. Como conciliar o avanço tecnológico com os sistemas legados? Uma proposta para este tipo de trabalho envolve a utilização de plataformas de desenvolvimento abertas como o CORBA para a integração das diversas tecnologias existentes. A utilização do CORBA permite a criação de novas ferramentas, já projetadas objetivando a operação em sistemas heterogêneos e a fácil atualização das mesmas, além de permitir o encapsulamento de ferramentas legadas de forma a maquiá-las as particularidades destas ferramentas. Idealmente, busca-se montar uma estrutura onde velho e novo interagem de forma a fornecer serviços de boa qualidade ao menor custo possível.

No início desta pesquisa, os trabalhos existentes nesta área, apesar de ainda estarem sendo desenvolvidos sobre outros tipos de plataformas de desenvolvimento, apontavam para o futuro promissor da plataforma CORBA, sendo que no presente momento a

---

<sup>8</sup> Ao longo do texto denominaremos a Plataforma de Desenvolvimento CORBA também por “Tecnologia CORBA”

utilização tal tecnologia está se expandindo rapidamente na área de medicina, inclusive com a criação do CORBAmed, comitê técnico responsável por definir os serviços CORBA horizontais na área de medicina. Grandes projetos, como o Telemed [capítulo 8, item 10] (LANL – *Los Alamos National Laboratory*) já usufruem dos benefícios trazidos pelo CORBA.

Devido aos diversos benefícios trazidos pela utilização de padrões de plataformas de desenvolvimento como o CORBA, optou-se pela utilização desta tecnologia em fase inicial da pesquisa. Dentre as inúmeras vantagens da tecnologia CORBA, podemos destacar:

Padrão aberto para construção de sistemas distribuídos, projetado segundo as necessidades de um consórcio de mais de 500 empresas, a OMG;

- \* Permite a encapsulação de sistemas legados;
- \* Integração de ferramentas dentro de um ambiente heterogêneo;
- \* Trabalho com componentes de programas facilitando abordagens do tipo plugin;
- \* Arquitetura voltada a fácil manutenção e atualização dos programas desenvolvidos;
- \* Desenvolvimento mais fácil e rápido se comparado com as demais tecnologias, de nível de abstração mais baixos;
- \* Diversas opções de implementações do CORBA, facilitando a escolha segundo as necessidades particulares de cada projeto. Existem implementações gratuitas ou de baixo custo, outras voltadas a nichos de aplicação como o de alto desempenho ou de tempo real.

Outras alternativas ao CORBA existentes para a criação de sistemas como o proposto são o JAVA/RMI e o COM+, entretanto estas alternativas não incorporam todos os benefícios integralmente apresentados pelo CORBA.

A primeira opção JAVA/RMI, permite alcançarmos diversos objetivos como integração de sistemas heterogêneos, construção de componentes de programação, facilidade de desenvolvimento, devido à diversidade de ferramentas e APIs já existentes, contudo apresenta algumas restrições de desempenho e integração com demais aplicações

legadas. Podemos dizer que o padrão CORBA é mais abrangente que o conjunto JAVA/RMI. Na realidade o RMI (*Remote Method Invocation*) é muito semelhante ao CORBA, porém restrito à linguagem JAVA.

A segunda opção, o COM+ (*Component Object Model*), trata-se de uma solução proprietária da *Microsoft*. Apesar do COM já ser parte integrante do próprio sistema operacional nos produtos *Microsoft*, é uma solução restrita ao ambiente PC/*Windows*. Trata-se de um dos grandes concorrentes do CORBA, sendo ambas tecnologias incompatíveis, a princípio. A tecnologia CORBA, por ser muito abrangente permite a construção de pontes de comunicação CORBA/COM, o que facilita a integração de sistemas, se olharmos pelo pontos de vista do CORBA.

Analisando estas características e muitas outras, decidiu-se construir este trabalho sobre a padronização CORBA.

#### **4.2.1. Metodologia de Projeto**

Inicialmente foi realizado um estudo e descrição do problema a ser resolvido para melhor entendimento do mesmo. Neste estudo foram levantadas as características e implicações do processo de diagnóstico médico por imagens, sendo possível descobrir como um exame é realizado, quais as formas de armazenamento de imagens e como é feita a interação entre especialistas médicos distantes. Através desta análise foi possível identificar as particularidades do processo e suas deficiências, o que permitiu também a formulação de propostas para minimização de tais deficiências.

Conhecendo mais detalhadamente o processo de diagnóstico médico baseado em imagens foi possível formular uma proposta para resolução do problema de organização do trabalho em grupo assistido por computador.

Considerando a forma de concepção do projeto, podemos classifica-lo **como Projeto Racional** (Rational Design) [Olson91], onde um sistema é construído para mudar a forma como as pessoas comportam-se, objetivando possíveis melhorias na qualidade do trabalho. Entre os fatores que levaram a esta abordagem citamos:

- \* O processo atual de diagnóstico por imagens, apesar de apresentar facilidades para informatização, hoje é feito de forma tradicional, onde chapas são

impressas e analisadas no negatoscópio<sup>9</sup>;

- \* Ferramentas de análise de imagens podem trazer muitos benefícios como maior precisão na tomada de medidas e aplicação de técnicas de visualização;
- \* A comunicação com auxílio de computadores em rede pode agilizar o processo de diagnóstico, promovendo a interação entre especialista para segunda opinião e automatização.

As características listadas acima, levam-nos a crer que muitas melhorias podem ser alcançadas através da informatização do processo de diagnóstico por imagens e através do trabalho em grupo assistido por computador. A comprovação das expectativas geradas pela proposta deste trabalho depende da implantação e utilização do sistema cooperativo, o que envolve aceitação por parte dos usuários, treinamento e algum tempo para análise e validação dos resultados.

#### **4.2.2. Expectativas em Telemedicina e o Trabalho Proposto**

Do ponto de vista médico, a implantação de novas tecnologias e sistemas na rotina médica pode trazer diversos benefícios já discutidos anteriormente (seção 3.1), contudo a aceitação destes sistemas apresenta certas resistências por parte dos usuários alvo. Esta resistência muitas vezes está ligada a aspectos culturais e à falta de informação nas instituições médicas.

Quando falamos em aceitação, este é um grande problema em CSCW e sistemas de informação, pois é necessário incorporar as novas técnicas e metodologias ao cotidiano dos usuários finais, o que muitas vezes é dificultado por questões culturais. Apenas após algum tempo de implantação e comprovação da utilidade e facilidade de um sistema, consegue alcançar boa aceitação.

Como em qualquer área, não apenas a de saúde, a adaptação a novos processos demanda algum esforço. Se as pessoas não estiverem preparadas e não entenderem os benefícios que podem ser alcançados com o novo sistema de trabalho, a adaptação a este novo sistema pode parecer mais custosa do que realmente é. Assim sendo, a disseminação da telemedicina depende muito da distribuição de informação à respeito da área e da

---

<sup>9</sup> Tela acrílica translúcida iluminada por trás por uma lâmpada fluorescente, cuja a luz se dispersa de forma homogênea através da tela onde são colocados os filmes radiológicos (ver seção 3.1).

educação tanto de profissionais como dos pacientes a serem beneficiados.

Considerando este aspecto, é muito importante que os novos profissionais, ainda em treinamento, estejam em constantes contato com as novidades para que a próxima geração de profissionais esteja pronta para aceitar mais facilmente mudanças que possam alterar seu cotidiano em nome da melhor qualidade de serviço.

Outro aspecto a ser considerado é que no trabalho cooperativo aplicado à medicina não se pretende substituir o contato humano no atendimento médico, mesmo por que nenhum sistema de computação tem a mesma capacidade de decisão de um ser humano, e sim oferecer suporte a decisão, automatização de tarefas desempenhadas por grupos e distribuição de informação e conhecimento na área de saúde.

Dentro deste trabalho não será possível realizar uma avaliação da aceitação do sistema cooperativo por parte dos profissionais da área de saúde e do impacto causado pela nova tecnologia na área de diagnóstico por imagens. Tal avaliação requer a implantação do sistema proposto em diversas instituições de saúde, o estudo estatístico para que se possa quantificar a alteração da qualidade do atendimento e por fim o um estudo de como o trabalho cooperativo auxiliou ou não no processo de diagnóstico por imagens, tentando-se identificar os reais benefícios alcançados e o que pode ainda ser melhorado no processo. Este tipo de estudo requer tempo e colaboração das instituições envolvidas e espera-se que em trabalhos futuros seja possível realizar tal tipo de avaliação.

Do ponto de vista tecnológico, a implementação e implantação de um ambiente de trabalho cooperativo em medicina, como o proposto neste trabalho, é perfeitamente viável não apenas para a área de diagnóstico por imagens, mas também para outros setores da telemedicina, como por exemplo para o telemonitoramento ou teleconsulta.

Podemos identificar as seguintes classes de problemas tecnológicos quando trabalhamos com telemedicina:

\* **Eficiência e segurança de aplicações em rede:**

A transmissão eficiente de dados médicos envolve a manutenção da integridade do dados, controle de qualidade do mesmo, podendo haver perdas ou não segundo a classe e os objetivos da aplicação final, segurança da informação transportada e manutenção da qualidade de serviço de transporte

de dados por rede de computadores;

\* **Desempenho das aplicações:**

As aplicações de processamento de dados ou interação síncrona entre integrantes de um grupo de trabalho requerem bom desempenho tanto do ambiente de computação, do ambiente de comunicação de dados e dos aplicativos;

\* **Qualidade de serviço prestado:**

O serviço prestado deve ser confiável e estar disponível mesmo que em situações adversas.

\* **Integração:**

Integração entre sistemas legados e novas ferramentas e tecnologias com o mínimo custo financeiro e de trabalho;

Atualmente a eficiência das aplicações em rede pode ser alcançada através das tecnologias de redes de alta velocidade ou mesmo da *Internet 2*, se considerarmos um nível mais amplo de interação. Já a segurança de dados médicos é alcançada através de diversos mecanismos e algoritmos (codificação de dados, mecanismos de chaves de acesso em sistemas em rede, conscientização dos usuários, dentre outros) destinados a este fim, existindo diversos produtos comerciais e pesquisas na área de segurança de redes e dados.

O desempenho no processamento de dados médicos pode ser alcançado através da utilização de computadores de alto desempenho, processamento paralelo e distribuído aliados às tecnologias de redes de alta velocidade para transmissão dos dados. Desta forma pode-se conseguir o processamento rápido e preciso de dados.

A qualidade do serviço prestado é garantida pela utilização de arquiteturas redundantes, aplicação de gerenciamento de redes ao sistema de informação, estruturação do sistema através de técnicas de projeto, além da utilização de padronizações de equipamentos e aplicativos.

A integração de sistemas heterogêneos e sistemas legados é alcançada através da utilização de padronizações como o CORBA e da utilização de técnicas de engenharia de *software* para melhor projeto das aplicações.

Dentro do contexto deste trabalho pretendeu-se estudar a viabilidade de projeto e implementação de aplicações para trabalho cooperativo em medicina, tentando-se provar que atualmente já dispomos de tecnologias suficientes para a realização de tal nicho de aplicação. Além de estudar a viabilidade de implementação de tais sistemas, procurou-se estudar e usar tecnologias as discutidas na construção de ferramentas para telemedicina identificando-se o benefícios e deficiências das mesmas.

### **4.3. Ambiente de Desenvolvimento**

Após a pesquisa bibliográfica a respeito do tema e de todo o processo de análise e concepção do trabalho passamos à fase de desenvolvimento. Durante esta fase, tentou-se simular um ambiente próximo da realidade encontrada em hospitais, tentando-se abordar os seguintes aspectos:

- \* **Integração de sistemas heterogêneos:**  
O sistema foi concebido e construído sobre uma ambiente de redes heterogêneo na tentativa de simular o ambiente existente nos hospitais;
- \* **Introdução de novas tecnologias e conceitos no processo de diagnóstico:**  
O processo de automatização do trabalho envolve a adoção de novas formas de interação, sendo abordado neste trabalho o trabalho em grupo suportado por computadores;
- \* **Desenvolvimento baseado em arquiteturas do tipo PC:**  
Estas plataformas são mais comuns e baratas;
- \* **Utilização de padrões abertos:**  
Para desenvolvimento de um sistema cooperativo de auxílio ao diagnóstico médico por imagens, neste caso escolhemos a plataforma de desenvolvimento aberta CORBA;
- \* **Aplicação de técnicas de engenharia de programação:**  
Para melhor organização e concepção do trabalho, objetivando a continuação da pesquisa além de facilitar sua extensão.

### **4.4. Infra-estrutura e Plataformas Adotadas**

Segundo as considerações descritas a pesquisa concretizou-se sobre a seguinte infra estrutura:

\* **Computadores PC/*Windows* para desenvolvimento e teste da aplicação de visualização cooperativa:**

Circuitaria: Gateway GP7-450, Intel Pentium III 450Mhz, 256Mb RAM, placa de rede SMC EZ Card 10/100 (SMC1211TX), placa de vídeo NVIDIA Riva TNT 16Mb;

Ambiente de Programação: JAVA/SUN JDK 1.2.2, Borland JBuilder Professional 3.0, ORBacus/JAVA 3.3.2 (implementação CORBA) e API ImageJ para leitura de imagens no formato DICOM e sistema operacional Windows 2000 Professional.

\* **Computadores para teste da aplicação cooperativa:**

Plataforma 1:

Circuitaria: Silicon Graphics O<sub>2</sub>, CPU MIPS R10000 Processor Chip Revision 3.4 250 MHz IP32, FPU MIPS R10010 Floating Point Chip Revision 0.0, 128Mb RAM, placa de rede ec0, placa de vídeo MVP unit 0 version 1.4;

Ambiente de Programação: sistema operacional IRIX Release 6.5 IP32, JAVA/SUN JDK 1.2.2

Plataforma 2:

Circuitaria: Silicon Graphics Indigo<sup>2</sup>, MIPS R10000 Processor Chip Revision 2.5 195 MHz IP28, MIPS R10.000 Floating Point Chip Revision 0.0, 512 Mbytes RAM, placa de rede ec0 version 1, placa de vídeo Maximum Impact/TRAM option card;

Ambiente de Programação: sistema operacional IRIX Release 6.5 IP28, JAVA/SUN JDK 1.2.2

Plataforma 3:

Circuitaria: Intel Pentium III 700MHz, 384Mb RAM, placa de rede Ethernet 10/100, placa de vídeo Diamond Viper V770 Ultra;

Ambiente de Programação: Sistema operacional Windows 98, JAVA/SUN JDK 1.2.2

Plataforma 4:

Circuitaria: Intel Pentium III 1GHz, 256Mb RAM, placa de rede Ethernet 10/100, placa de vídeo NVIDIA GE Force 256 AGP ;

Ambiente de Programação: Sistema operacional Windows 98, JAVA/SUN JDK 1.2.2



- \* **Servidor PC/Linux para manutenção dos serviços CORBA (nomes e eventos) e para desenvolvimento do serviço de CSCW:**

Circuitaria: Pentium PRO 4 processadores 200MHz, 256M RAM, 18Gb disco rígido, placa de rede Ethernet 10/100 Mbps;

Ambiente de Programação: Linux redhat 6.2, gcc 2.95, ORBacus/C++ 3.3.2;

- \* **Rede local Ethernet (TCP/IP), 10Mbps com conexão para *internet*;**

A rede utilizada para desenvolvimento e testes, por não ser restrita ao projeto, apresenta características semelhantes às encontradas no ambiente alvo, apresentando alto tráfego de informações, horários de pico e ambiente heterogêneo, isto é, diversas combinações de arquitetura/sistema operacional conectados à mesma rede.

A Figura 25 representa a circuitaria e programas utilizados no desenvolvimento do ambiente cooperativo.

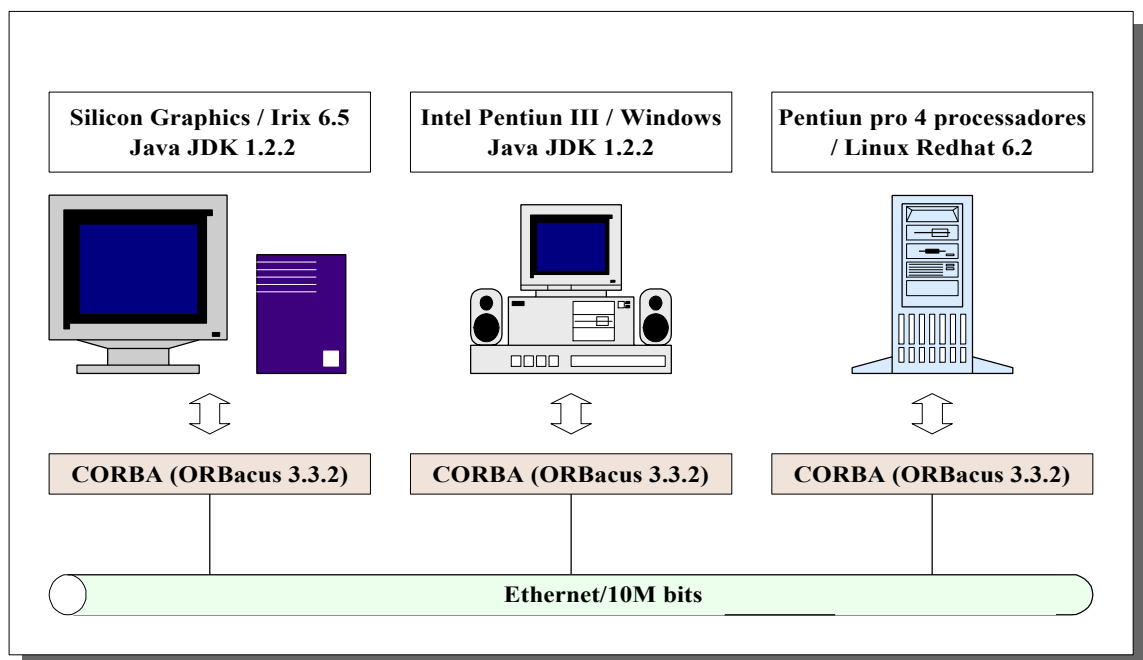


Figura 25: Arquitetura de circuitaria e ambiente de programação utilizada no desenvolvimento e testes.

Conforme o ambiente descrito, duas plataformas foram usadas para desenvolvimento e testes do sistema cooperativo, enquanto as demais foram utilizadas para testes de funcionamento e desenvolvimento. Foi realizado um trabalho extensivo de configuração e teste em diversas opções de programas e máquina até o ambiente descrito ser alcançado. Mais adiante o trabalho de configuração e testes será descrito.

#### **4.5. Sistema Desenvolvido**

Para esta pesquisa foi realizado um trabalho de levantamento das características desejáveis em um sistema cooperativo para medicina e implementado um protótipo que atenda aos requisitos mínimos necessários para testes e validação do trabalho.

##### **4.5.1. Implementação da Arquitetura**

Segundo a arquitetura proposta o sistema é composto por 2 servidores, uma base de dados, uma interface gráfica para análise de imagens em grupo e algumas ferramentas de auxílio ao trabalho em grupo. Como o objetivo principal do trabalho é estudo de CSCW em medicina implementado sobre a estrutura de objetos distribuídos, enumera-se a seguir em ordem de prioridade os subsistemas do ambiente cooperativo:

- \* Servidor de CSCW;
- \* Interface de análise de imagens cooperativa;
- \* Protótipo do banco de dados;
- \* Serviços extra de auxílio ao trabalho em grupo;
- \* Servidor de processamento de dados médicos.

O protótipo de banco de dados encontra-se implementado em um servidor de banco de dados MySQL rodando em uma máquina IBM/AIX, contudo os componentes de acesso ao banco de dados não foram implementados devido a falta de tempo, como já mencionado anteriormente e a problemas na compilação da biblioteca CORBA em ambiente IBM/AIX.

Os serviços de auxílio ao trabalho em grupo apesar de importantes em um sistema cooperativo, requerem a configuração de um servidor web, a confecção de paginas e comunicação com o banco de dados. Como o banco de dados não encontra-se inteiramente funcional, devido a falta dos componentes de acesso, a implementação deste serviço tornou-se inviável na atual implementação.

Outras ferramentas que também são consideradas como auxílio ao trabalho em grupo são: ferramenta para leitura e escrita de correio eletrônico e ferramentas de videoconferência. A implementação de tais ferramentas é desnecessária, pois já existem diversos produtos no mercado que já desempenham tal função. A preocupação

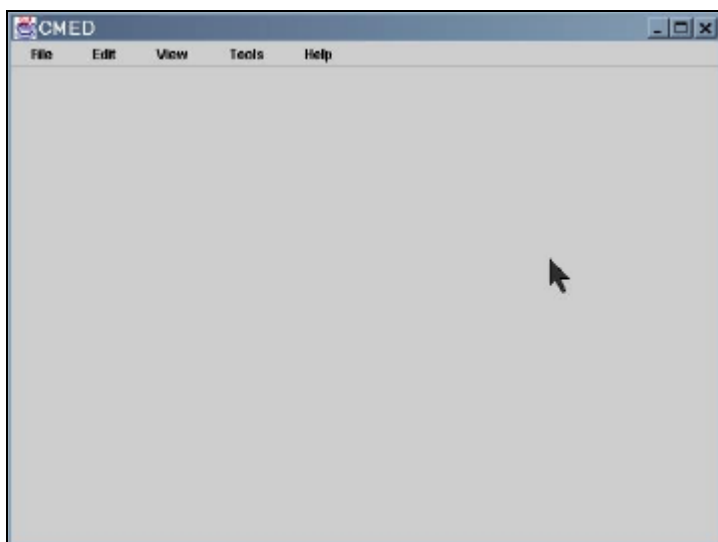
relacionada às ferramentas de correio eletrônico e videoconferência é apenas em como integra-las ao ambiente cooperativo de forma simples e natural para o usuário.

A implementação do servidor de processamento de dados não é essencial para o funcionamento dos sistema cooperativo, sendo assim sua implementação será realizada futuramente por questão de tempo disponível.

O servidor de CSCW é essencial para o gerenciamento dinâmico de usuários e de grupo de trabalho, constituindo o principal subsistema do ambiente cooperativo. Sendo assim, o Servidor de CSCW deve possuir uma implementação completa por ser vital ao ambiente cooperativo.

A interface gráfica para análise de imagens é importante pois é através dela que os componentes que promovem o trabalho cooperativo são testados. Esta interface gráfica para análise de imagens em grupo foi simplificada para atender a um tempo máximo de implementação.

Apesar da redução da aplicação gráfica implementada, ela ainda atende aos requisitos básicos para a interação entre especialistas. As figuras Figura 26, Figura 27 e Figura 28 representam as telas da interface para o trabalho cooperativo.



A tela principal da aplicação gráfica permite o acesso aos diversos recursos e subsistemas da interface. Ao executar a aplicação esta é a primeira tela exibida pelo sistema.

Figura 26: Tela principal da aplicação gráfica.

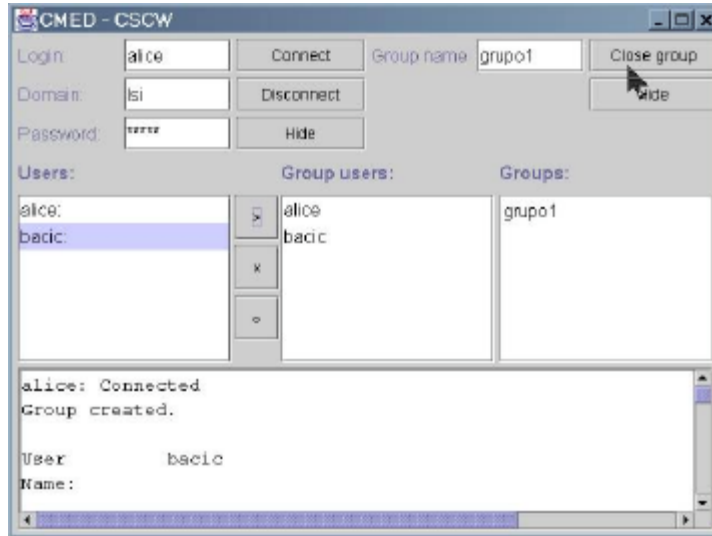


Figura 27: Tela de conexão com os serviços de CSCW.

A conexão com os serviços de CSCW é feita em uma tela, onde o usuário fornece um nome e senha de identificação. A tela permite a visualização dos grupos de trabalhos existentes, a criação de novos grupos e administração dos usuários em uma sessão cooperativa.



Figura 28: Tela de visualização de exames.

Os exames são visualizados em uma tela específica, onde é possível realizar atividades tanto em grupo como individualmente. No exemplo ao lado temos a extração de distâncias de forma cooperativa. As marcações em laranja pertencem ao usuário local enquanto as marcações em azul ao usuário remoto. É possível realizar um *stackflow* entre as fatias do exame, sendo que todas as interfaces estão sincronizadas com a interface mestre (que possui a liderança do grupo).

Atualmente, utiliza-se muito a videoconferência como meio de comunicação entre as partes de um grupo, não havendo formas mais precisas, eficientes e interativas para troca de dados radiológicos. Considerando a forma simples como as reuniões são realizadas no modelo vigente, o sistema implementado mostra-se muito útil para troca de informação no formato digital.

A Figura 29 representa a configuração final do ambiente cooperativo de auxílio ao diagnóstico médico por imagens.

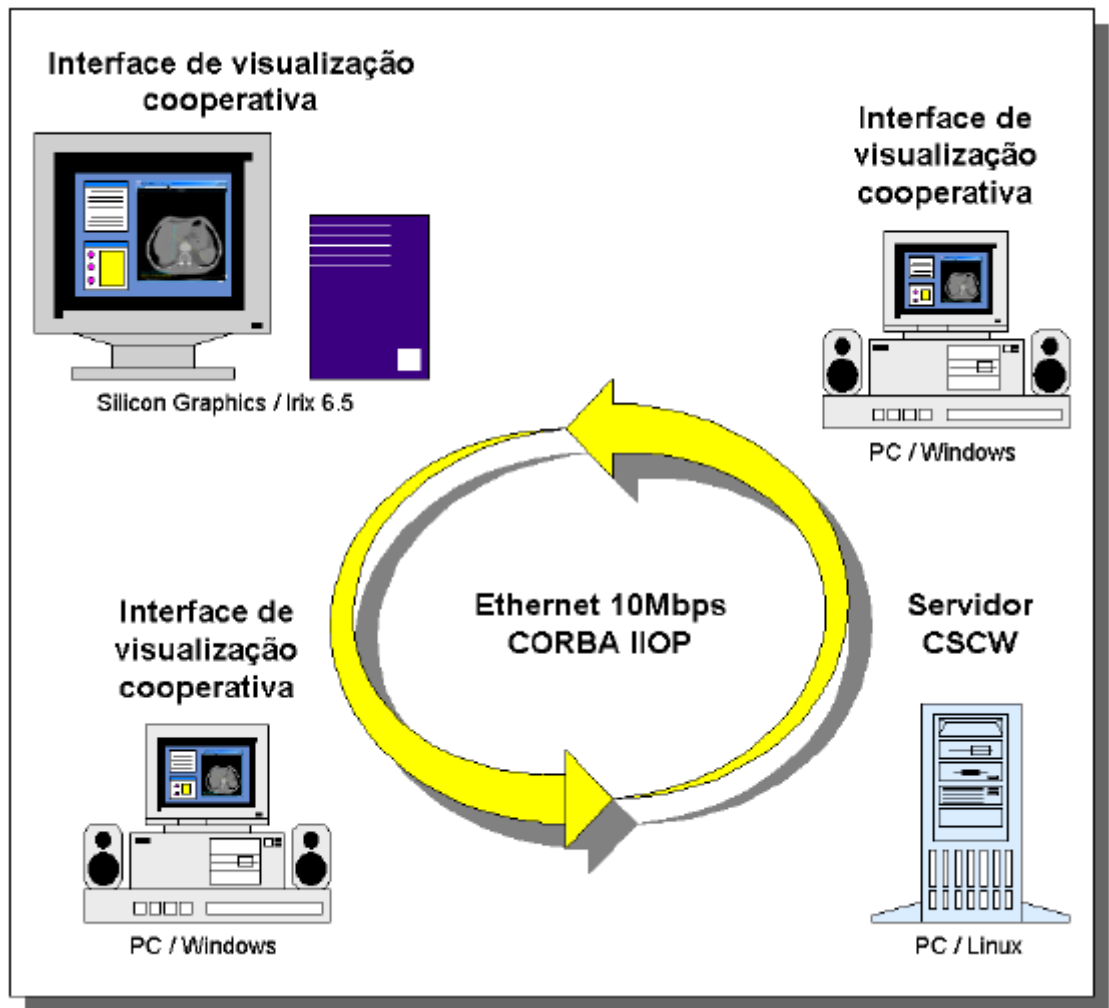


Figura 29: Configuração final do ambiente cooperativo

#### **4.5.2. Processo de Desenvolvimento do Ambiente Cooperativo**

O desenvolvimento dos subsistemas do ambiente cooperativo seguiu uma ordem segundo a importância e dependência dos módulos.

Inicialmente tivemos o desenvolvimento de parte do Servidor de CSCW, que constitui o principal elemento do sistema. Paralelamente ao desenvolvimento do servidor, foi criada uma interface de gerenciamento por linha de comando. Esta interface serviu para iniciar os testes no servidor.

A partir de um certo momento foi iniciado o desenvolvimento da interface para análise de imagens em grupo, pois através dela foi possível testar mais precisamente as

funcionalidades do Servidor de CSCW já implementadas.

A aplicação de visualização foi construída através de módulos de programação que ao serem agrupados permitiram a comunicação entre diversas aplicações e o servidor de CSCW. O desenvolvimento desta aplicação foi feito em JAVA sobre CORBA, o que facilita o seu transporte para qualquer plataforma que possua uma máquina virtual JAVA.

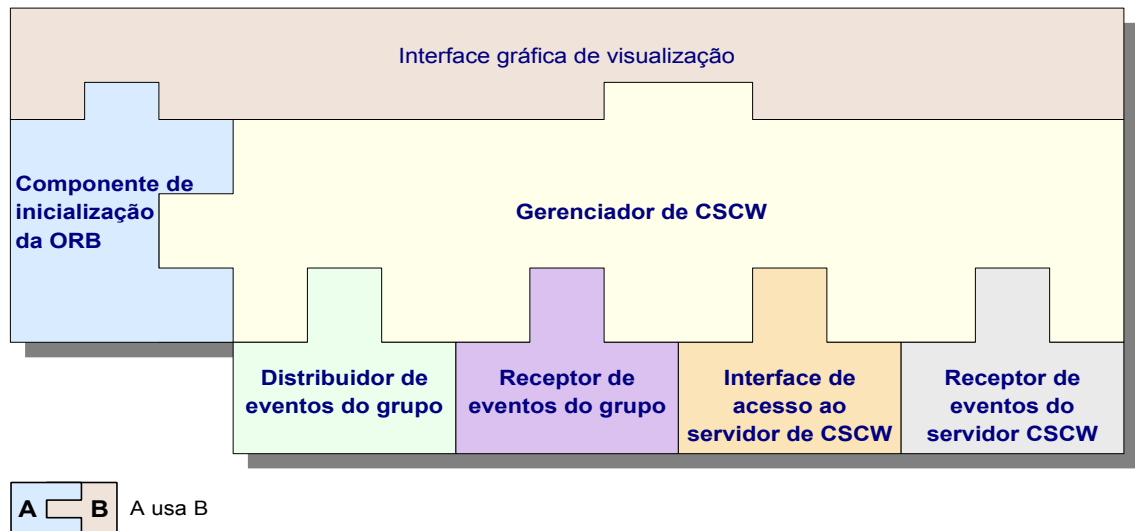


Figura 30: Componentes da interface de visualização.

Os módulos de programação da aplicação de visualização podem ser aproveitados em outras aplicações, facilitando a construção de novas interfaces gráficas para a visualização de dados de forma cooperativa. Este conceito facilita a renovação apenas da interface do usuário, evitando o trabalho de reconstrução de todo aparato para comunicação com o ambiente cooperativo.

O Servidor de CSCW foi construído em C++ sobre CORBA em uma plataforma Linux. Apresenta uma interface definida em IDL, que oferece acesso aos serviços do mesmo e permitindo o acesso por parte de qualquer aplicação que se beneficie da arquitetura fornecida pelo CORBA, não importando qual a linguagem em que a aplicação foi escrita ou qual implementação de ORB foi utilizada.

A Figura 31 esquematiza a estrutura do Servidor de CSCW:

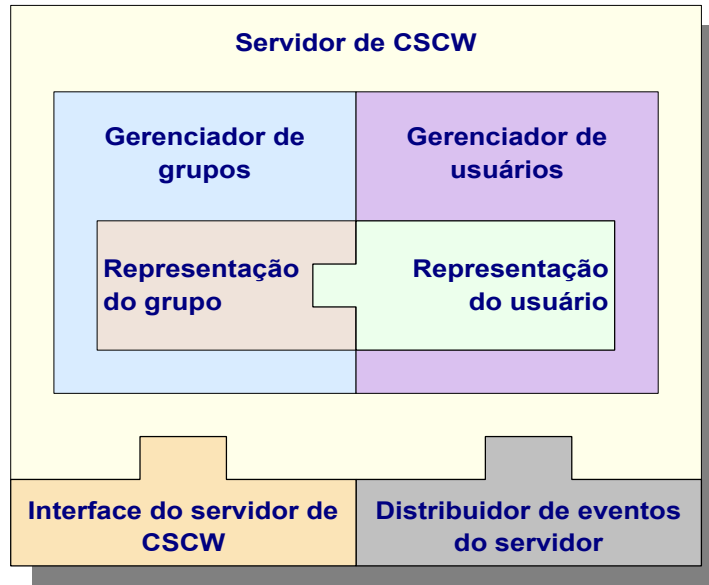


Figura 31: Estrutura do Servidor de CSCW

A abordagem usada na construção do Servidor de CSCW foi a de derivação de classes, de forma a estender uma interface básica até a obtenção do servidor final. A ampliação dos serviços oferecidos pelo servidor pode

ser feita por mais derivações da classe original.

#### 4.5.2.1 Análise do Processo de Desenvolvimento.

Toda a arquitetura do sistema cooperativo foi elaborada sobre padrões e tecnologias abertas, objetivando atender a usuários de sistemas heterogêneos. Tais padrões e tecnologias mostraram-se muito úteis no projeto, desenvolvimento e manutenção da aplicação tornando-os mais simples e rápidos se comparados com outras ferramentas onde se trabalha em um nível de abstração mais baixos, sendo o projeto e a programação muito dependentes de arquitetura.

Apesar da grande promessa trazida por padrões abertos, nem tudo é tão simples quanto parece. Ao optar por uma solução como o CORBA, todo trabalho voltado à interação entre objetos distribuídos é poupado, contudo muitas vezes acontece que um componente construído em uma determinada combinação de máquina e sistema operacional, pode apresentar variações de desempenho e comportamento quando transportado para outra plataforma. Durante o desenvolvimento do ambiente cooperativo, foram encontrados diversas dificuldades, entre elas citamos:

- \* **Diferenças de comportamento da biblioteca CORBA (C++) em plataformas diferentes.**

A biblioteca CORBA utilizada, é portátil para diversas plataforma de sistema

operacional e máquina, devendo ser compilada para cada uma das plataformas desejadas. Durante a implementação do ambiente, verificou-se que a biblioteca apresenta alguns problemas dependentes da plataforma. Um exemplo refere-se aos threads. O funcionamento de aplicações *multithread* depende da qualidade da biblioteca de threads nativa da plataforma, o que causa algumas falhas esporádicas no sistema.

\* **Diferenças entre compiladores C++ nas diferentes plataformas.**

No caso da parte C++ da biblioteca CORBA, dependendo do compilador usado, há diferenças de otimização no código da biblioteca ou mesmo falhas de compilação devido a defeitos no compilador. Como a biblioteca utilizada não é fornecida já compilada e sim na forma de código fonte, este problema dificultou a montagem de uma plataforma de desenvolvimento para os servidores. A tabela a seguir mostra as combinações de arquitetura onde foi testada a compilação da biblioteca ORBacus 3.3.2 para C++:

Plataforma	Compilador	Resultado
Intel Pentium Pro 4 processadores / Linux Redhat 6.2	GCC 2.95	Bibliotecas geradas e funcionais.
Intel Pentium III / Linux Redhat 6.2	GCC 2.95	Bibliotecas geradas e funcionais.
Intel Pentium III / Windows NT4	Visual C++ 6.0	Bibliotecas geradas e funcionais tanto para NT4 como para Windows 98 e Windows 2000.
Intel Pentium III / Windows 2000	Visual C++ 6.0	Bibliotecas geradas e funcionais tanto para NT4 como para Windows 98 e Windows 2000.
AMD K6 III / Windows 98	Visual C++ 6.0	Dificuldades em gerar bibliotecas. A compilação deve ser totalmente manual devido à diferenças na interface de linha de comando em relação ao <i>Windows NT</i> . O código gerado funciona perfeitamente.
AMD K6 III / Linux Redhat 6.2	GCC 2.95	O compilador <i>gcc</i> gera código não compatível com a arquitetura. Aplicações geradas em arquitetura <i>Intel</i> , com o mesmo compilador também apresentam o mesmo problema, quando executadas na arquitetura K6.
Silicon Indigo / Irix 6.5	GCC 2.95	Compilação da biblioteca completa, porém comportamento das aplicações anômalo.
IBM/AIX	GCC 2.95	Não foi possível compilar a biblioteca.
Alpha / Digital Unix	GCC 2.95	Não foi possível compilar a biblioteca.

Tabela 13: Resultado da compilação em diversas arquiteturas

As dificuldades em gerar a biblioteca CORBA nas diferentes plataformas relacionam-se a diversos tipos de problemas como defeitos no compilador ou nas bibliotecas nativas, até problemas de configuração da máquina. A identificação precisa da origem dos problemas é difícil e demanda muito trabalho por parte da equipe de desenvolvimento. Para uma equipe pequena,



tais problemas podem assumir uma proporção muito grande, inviabilizando o trabalho, o que não aconteceria em uma equipe de trabalho maior;

\* **Diferenças entre máquinas virtuais JAVA nas diferentes plataformas.**

No que se refere ao JAVA, houve alguns problemas de porte da aplicação gráfica para o Linux. Este problema só foi solucionado no momento que a SUN lançou uma Máquina Virtual JAVA 1.2 estável para Linux.

Apesar de alguns problemas de compilação da porção JAVA da biblioteca, que depende do sucesso da compilação da porção C++, é interessante frisar que teoricamente a aplicação gerada em uma arquitetura pode ser transportada para qualquer outra máquina que possua uma máquina virtual JAVA, sem a necessidade de alterações ou adaptações. Ao testar a aplicação JAVA em diversas plataformas, constatou-se que esse transporte nem sempre é bem sucedido, pois algumas plataformas apresentam defeitos em sua máquina virtual ou no próprio sistema operacional.

A tabela a seguir mostra os resultados de testes realizados em uma aplicação JAVA gerada em uma máquina Pentium III / Windows 2000 / Borland JBuilder 3.0 e executada em diversas plataformas.

Plataforma	Resultado
Pentium III Windows 2000 Borland JBuilder 3.0	Execução bem sucedida. Não foi constatado nenhum problema tanto de comunicação com os serviços alocados no servidor <i>Linux</i> quanto entre aplicações gráficas cooperativas. No entanto foram identificados alguns defeitos na máquina virtual da Borland que não chegam a afetar o desempenho da aplicação.
Pentium III Windows 2000 JDK 1.2.2	Execução bem sucedida. Não foi constatado nenhum problema tanto de comunicação com os serviços alocados no servidor <i>Linux</i> quanto entre aplicações gráficas cooperativas.
Silicon Graphics Irix 6.5 JDK 1.2.2	Execução aparentemente bem sucedida. Não foi constatado nenhum problema tanto de comunicação com os serviços alocados no servidor <i>Linux</i> quanto entre aplicações gráficas cooperativas. Após algum tempo de execução, com a manipulação intensa de dados, ocorre um problema de memória. Aparentemente existe algum problema no <i>Garbage collector</i> nesta arquitetura.
Pentium III Windows 98 JDK 1.2.2	Execução bem sucedida. Não foi constatado nenhum problema tanto de comunicação com os serviços alocados no servidor <i>Linux</i> quanto entre aplicações gráficas cooperativas.
Pentium III NT4 SP6 JDK 1.2.2	Não foi possível executar a aplicação devido a um problemas de alocação de recursos no sistema operacional, que devido a um defeito, não permite a usuários que não possuam permissões de administrador alocar <i>sockets</i> .

Tabela 14: Execução da aplicação JAVA em diversas arquiteturas.

\* **Diferenças de desempenho**

A aplicação JAVA apresenta variação de desempenho muito grande em

plataformas diferentes. A própria máquina virtual do JAVA requer bom desempenho da máquina hospedeira. A Tabela 15 mostra os resultados da execução da aplicação de visualização de imagens cooperativa em diferentes plataformas:

<b>Plataforma</b>	<b>Resultado</b>
Pentium III 450MHz, 256Mb RAM / Windows 2000	Desempenho bom. Foi possível manter a interface interativa
Silicon Graphics, Indigo <sup>2</sup> 195MHz, 512Mb RAM / Irix 6.5	Desempenho bom, sendo possível manter a interface interativa.
Silicon Graphics, Indy 100MHz, 64Mb RAM / Irix 6.5	Desempenho ruim. A aplicação torna-se muito lenta, não sendo possível interagir com as demais aplicações em uma sessão cooperativa.
Silicon Graphics, O <sub>2</sub> 250MHz, 128Mb RAM / Irix 6.5	Desempenho bom, sendo possível manter a interface interativa.
Pentium III 700MHz, 384Mb RAM / Windows 98	Desempenho bom. Foi possível manter a interface interativa

Tabela 15: Desempenho da aplicação cooperativa em diversas plataformas.

\* **Instabilidade do sistema operacional Windows 98:**

O Windows 98 se revelou uma péssima plataforma de desenvolvimento, apesar de muito utilizada. A máquina perde desempenho gradativamente e trava com muita frequência. Como plataforma de execução da aplicação o Windows 98 não apresenta grandes problemas.

Problemas como os mencionados acima, apesar de parecem irrelevantes, são fatores significativos que podem atrasar o desenvolvimento, pois suas soluções estão fora do alcance da equipe de implementação do sistema. Para evitar tais problemas, é necessária uma avaliação prévia detalhada das ferramentas de desenvolvimento.

#### **4.6. Resumo da Seção**

O protótipo implementado para atende os requisitos básicos para validação da pesquisa, tendo sido empregadas as técnicas de engenharia e *Software* e objetos distribuídos, o que facilita a complementação do protótipo.

## **5. Avaliação e Análise da Implementação de um Sistema Colaborativo para Teleradiologia**

### **5.1. Introdução**

Neste capítulo é apresentada uma avaliação e análise do sistema implementado face ao sistema proposto no capítulo 3.

A validação do trabalho pode ser encarada sobre duas perspectivas: perspectiva técnica e perspectiva funcional.

Do ponto de vista técnico, procurou-se avaliar a viabilidade experimental da proposta inicial de trabalho através da implementação de um protótipo construído sobre as diversas tecnologias investigadas e discutidas neste trabalho.

Do ponto de vista funcional, objetivou-se comprovar a utilidade do trabalho através da implantação e uso, esta avaliação foi realizada em um ambiente controlado de laboratório,

Futuramente pretende-se avaliar funcionalmente o sistema num ambiente médico institucional como um hospital ou em clínicas de pequeno e grande porte. Este trabalho deverá ser baseado no aprimoramento dos recursos funcionais não implementados ainda (capítulo 4).

### **5.2. Descrição de Testes Realizados**

Para avaliação técnica do ambiente cooperativo foram executados alguns testes, objetivando verificar a estabilidade e o desempenho do sistema.

#### **5.2.1. Teste de Estabilidade do Servidor**

O testes de estabilidade do servidor objetiva verificar o funcionamento do mesmo ao gerenciar diversas conexões com clientes. Pretende-se estabelecer a capacidade do servidor em gerenciar as conexões sem que o serviço se deteriore.

O servidor foi capaz de atender diversas conexões com clientes em diversos tipos de plataformas. Teoricamente o servidor apresenta capacidade ilimitada, contudo sabemos

que esta capacidade é limitada por características da máquina hospedeira como memória e CPU. A aplicação servidora é relativamente leve, sendo que não foi possível estabelecer o número máximo de clientes dentro do ambiente de testes, pois o servidor jamais apresentou degradação de funcionamento com as máquinas usadas no teste.

O teste consistiu em conectar diversas aplicações clientes ao servidor e verificar se todo o sistema continuou funcional, sendo o teste bem sucedido.

### **5.2.2. Teste transmissão de dados pela rede**

O primeiro teste de transmissão de dados realizado objetiva estabelecer o tempo de transmissão de dados pela rede verificando-se a viabilidade da aplicação cooperativa, que deve manter taxas que forneçam ao usuário a sensação de interatividade.

O procedimento para este teste foi estabelecer a comunicação entre duas interface de visualização e medir o tempo de transmissão de uma tomografia pela rede. A configuração do teste foi a seguinte:

\* **Tipo de rede:**

Rede local, Ethernet 10Mbps

\* **Tipo de dados:**

Formato:	Dicom 3
Bytes/pixel:	2 (inteiro curto sem sinal)
Resolução da fatia:	512 x 512
Tamanho em bytes:	524288 por fatia

\* **Número de máquinas envolvidas: 2**

Plataforma 1 (Origem):

Modelo:	Gateway GP7-450
Processador:	Intel Pentium III 450Mhz
Memória:	256Mb RAM
Placa de rede:	SMC EZ Card 10/100 (SMC1211TX)
Placa de vídeo:	NVIDIA Riva TNT 16Mb
Sistema Operacional:	Windows 2000 Professional

Plataforma 2 (Destino):

Modelo:	Silicon Graphics O <sub>2</sub>
Processador:	CPU MIPS R10000 Processor Chip, Revision 3.4, 250 MHz IP32

Coprocessador:	FPU MIPS R10010 Floating Point Chip, Revision 0.0
Memória:	128Mb RAM
Placa de rede:	Ec0
Placa de vídeo:	MVP unit 0 version 1.4
Sistema Operacional:	IRIX Release 6.5 IP32

Foram executadas diversas medidas de tempo segundo a configuração descrita acima, não havendo isolamento da rede onde foram executados os testes. As medidas foram tomadas em horário normal de uso da rede. Obteve-se os seguintes resultados:

Nº de fatias	Tamanho (bytes)	Tempo médio (ms)
20	10.485.760	6.221,8
40	20.971.520	12.374,6
60	31.457.280	18.655,6
80	41.943.040	24.456,4
100	52.428.800	30.761,0

Tabela 16: Resultados do teste de transmissão de dados.

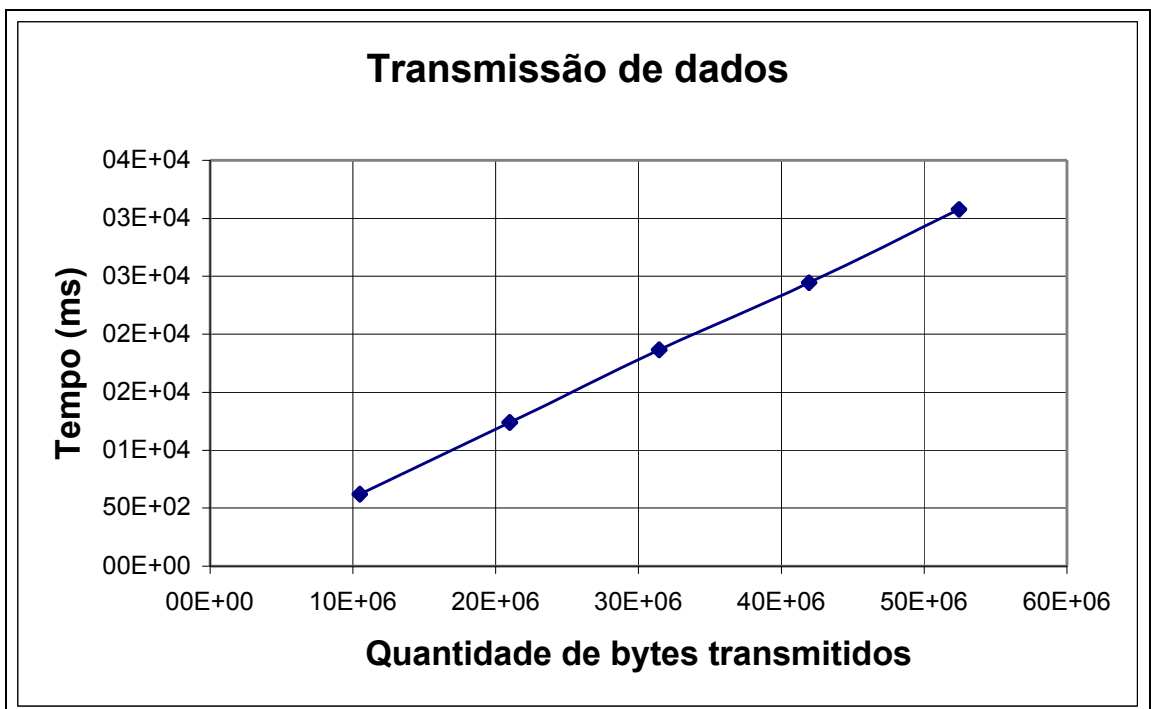


Gráfico 1: Resultados do teste de transmissão de dados.

Segundo os resultados observados no **Erro! A origem da referência não foi encontrada.**, verificamos que a relação bytes transmitidos/tempo é linear, apresentando valores satisfatórios para a aplicação em questão. A aplicação estudada deve trabalhar

com dados em resolução 512x512 pixels não ultrapassando muito mais que 100 fatias. Exames tomográficos não ultrapassam muito essa resolução para que o paciente não receba uma dose muito alta de radiação.

Os tempos obtidos são aceitáveis para manutenção da interatividade, sendo este tempo equivalente ao de abertura de arquivos no disco local.

### **5.2.3. Teste de Número de Usuários em Sessão Cooperativa**

Este teste pretendeu verificar o funcionamento das aplicações gráficas dentro de uma sessão cooperativa. Tentou-se classificar se o desempenho do sistema permite a interação em síncrona entre diversos usuários.

Segue a configuração do teste realizado:

\* **Tipo de rede:**

Rede local, Ethernet 10Mbps

\* **Tipo de dados:**

Formato:	Dicom 3
Bytes/pixel:	2 (inteiro curto sem sinal)
Resolução da fatia:	512 x 512
Tamanho em bytes:	524288 por fatia
Nº de fatias:	20

\* **Número máximo de máquinas envolvidas: 5**

Plataforma 1 (líder):

Modelo:	Gateway GP7-450
Processador:	Intel Pentium III 450Mhz
Memória:	256Mb RAM
Placa de rede:	SMC EZ Card 10/100 (SMC1211TX)
Placa de vídeo:	NVIDIA Riva TNT 16Mb
Sistema Operacional:	Windows 2000 Professional
JAVA:	SUN/JDK 1.2.2

Plataforma 2:

Modelo:	-
Processador:	Intel Pentium III 733Mhz
Memória:	384Mb RAM
Placa de rede:	3COM EtherLink 10/100 PCI For Complete PC Management NIC (3C905C-TX)

Placa de vídeo: Diamond Viper V770 Ultra  
 Sistema Operacional: Windows 98  
 JAVA: SUN/JDK 1.2.2

Plataforma 3:

Modelo: Silicon Graphics O<sub>2</sub>  
 Processador: MIPS R10000 Processor Chip, Revision 3.4, 250 MHz  
 IP32  
 Coprocessador: FPU MIPS R10010 Floating Point Chip, Revision 0.0  
 Memória: 128Mb RAM  
 Placa de rede: Ec0  
 Placa de vídeo: MVP unit 0 version 1.4  
 Sistema Operacional: IRIX Release 6.5 IP32  
 JAVA: SUN/JDK 1.2.2

Plataforma 4:

Modelo: Silicon Graphics Indigo<sup>2</sup>  
 Processador: MIPS R10000 Processor Chip, Revision 2.5, 195 MHz  
 IP28  
 Coprocessador: MIPS R99999 Floating Point Chip, Revision 0.0  
 Memória: 512 Mbytes  
 Placa de rede: ec0, version 1  
 Placa de vídeo: Maximum Impact/TRAM option card  
 Sistema Operacional: IRIX Release 6.5 IP28  
 JAVA: SUN/JDK 1.2.2

Plataforma 5:

Modelo: Dell Precision 4100  
 Processador: Intel Pentium III 1GHz  
 Memória: 256Mb RAM  
 Placa de rede: 3COM EtherLink 10/100 PCI For Complete PC  
 Management NIC (3C905C-TX)  
 Placa de vídeo: NVIDIA GE Force 256 AGP  
 Sistema Operacional: Windows 98  
 JAVA: SUN/JDK 1.2.2

As medidas foram tomadas em uma rede não isolada em seu horário normal de funcionamento. Optou-se por não isolar a rede para se obter resultados mais próximos da realidade e avaliar a viabilidade do sistema. Inicialmente foram tomadas medidas em sessões com dois integrantes, sendo aumentado o número de integrantes a cada teste até atingirmos o total de cinco integrantes.

Os resultados obtidos foram:

Plataforma	Tempo médio de recepção de dados	Desempenho da interface
2	5876,67 ms	Bom
3	6959,33 ms	Razoável

Tabela 17: Resultados do teste de desempenho da aplicação gráfica com (3 aplicações)

Plataforma	Tempo médio de recepção de dados	Desempenho da interface
2	5676,67 ms	Bom
3	6388,67 ms	Razoável
4	6733,67 ms	Razoável

Tabela 18: Resultados do teste de desempenho da aplicação gráfica com (4 aplicações)

Plataforma	Tempo médio de recepção de dados	Desempenho da interface
2	5803,33 ms	Bom
3	6369,33 ms	Razoável
4	6988,67 ms	Razoável
5	5623,33 ms	Bom

Tabela 19: Resultados do teste de desempenho da aplicação gráfica com (5 aplicações)

O primeiro resultado visível é que o número de interfaces dentro da sessão cooperativa não influenciou no desempenho da aplicação em grupo. Os tempos de transmissão de dados se mantiveram coerentes conforme a plataforma, sendo que há um tempo de espera na fila de requisição de dados, pois a interface líder é capaz de atender a apenas uma requisição de dados por vez. O tempo de espera não foi considerado na coluna de "Tempo médio de transmissão", mas foi considerado para avaliar se as interfaces podem ser consideradas interativas.

A coluna de classificação de desempenho da interface procurou avaliar se a resposta da aplicação manteve um resultado compatível com aplicações interativas. As plataformas 3 e 4 apresentaram resultados razoáveis, o que significa que a interação síncrona é possível nestas máquinas. Máquinas com configuração muito inferior não são compatíveis com os níveis de interação desejados para a aplicação, pois além da máquina virtual JAVA necessitar boa capacidade de processamento, a aplicação necessita de quantidade razoável de memória além de necessitar certo tempo de CPU para processamento dos dados.



A funcionalidade que apresentou maior tempo de sincronização foi o *stackflow*<sup>10</sup>, porém este tempo de resposta é influenciado por fatores como desempenho da parte gráfica do JAVA. Aqui além do tempo de distribuição de eventos temos que considerar o tempo que o JAVA requer para atualizar a tela da interface. Nas plataformas 3 e 4, de desempenho mais baixo, tivemos a interatividade diminuída.

A arquitetura implementada atualmente, por não possuir garantia de tempo de entrega dos eventos entre as aplicações integrantes de uma sessão, pode se mostrar inadequada em situações críticas. Futuramente pretende-se implementar algum tipo de mecanismo que compense esta deficiência.

Um último aspecto a ser considerado é que nas plataformas mais modernas (2 e 5), o tempo gasto com a pilha de protocolos parecem não influenciar fortemente a transmissão de dados por rede. Se considerarmos a carga da rede estável (com pouca variação) durante a tomada das medidas, a diferença de tempo de transmissão entre as duplas de plataformas 2/5 e 3/4 parece diminuir conforme a velocidade de processamento de cada plataforma. Um fato acontecido durante a tomada de medidas foi que o mesmo valor de tempo foi obtido repetidas vezes nas plataformas 2 e 5.

Este resultado prova que as plataforma CORBA é compatível com aplicações que requerem alto nível de interação, pois a tecnologia de circuitaria encontra-se avançada o suficiente para manutenção de aplicações interativas sobre o padrão já mencionado.

### **5.3. Avaliação Geral da do Sistema Segundo a Classificação de Tempo-Espaço**

Uma primeira avaliação que podemos fazer do sistema implementado segue os critérios de avaliação introduzidos no capítulo Figura 5 Sabemos que um sistema de CSCW completo deve preencher todos os quadrantes do gráfico. No caso do sistema proposto e implementado obtivemos a seguinte configuração:

---

<sup>10</sup> *Stackflow*: Apresentação de um dado volumétrico na forma de fatias, sendo possível deslizar através das fatias fornecendo a impressão de profundidade.

	Simultâneo	Não simultâneo
Mesmo local	Interação síncrona (face-a-face)	Interação assíncrona
Locais diferentes	Interação síncrona distribuída	Interação assíncrona distribuída

Requisito atendido
  Requisito não atendido

A interface gráfica fornecida permite tanto o trabalho em grupo de forma síncrona, isto é, diversos usuários trabalhando em locais diferentes através da rede de computadores, como o trabalho individual ou em grupo no mesmo local. Assim temos dois quadrantes do gráfico da Figura 32 atendidos.

Figura 32: Requisitos atendidos segundo a classificação no tempo-espaço do Groupware [Ellis91].

Uma forma de obter a interação assíncrona dentro do sistema proposto é através da utilização de uma ferramenta de correio eletrônico mais as ferramentas adicionais em *web*. Uma ferramenta de videoconferência, apesar de muito utilizada, ainda manteria o sistema nos quadrantes de aplicações síncronas.

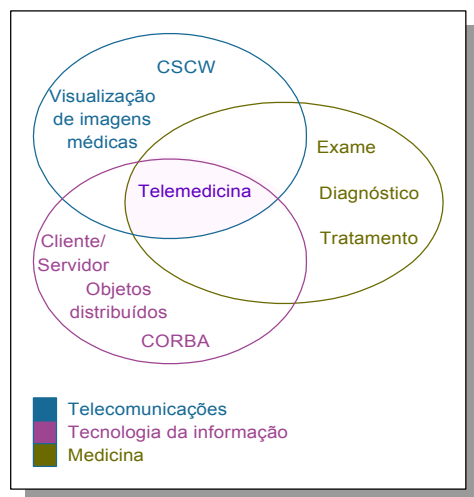
No caso estudado neste trabalho, a maior dificuldade de trabalho em grupo encontra-se no quadrante das aplicações síncronas/distribuídas, sob este ângulo podemos dizer que os objetivos foram alcançados, uma vez que o sistema implementado preenche este quadrante.

#### 5.4. Avaliação sobre a Perspectiva da Telemedicina

Por definição a telemedicina é o fornecimento de serviços médicos a distância, assim sendo a ferramenta implementada atende a tal requisito, pois permite a interação entre médicos através de uma rede de computadores. Analisando as características descritas no capítulo 2.1 sobre telemedicina, a aplicação gráfica pode ser classificada segundo o nível de aplicações entre hospitais, onde dados radiológicos são transportados e visualizados por um grupo de usuários localizados em lugares diferentes em busca de um diagnóstico.

Considerando o conjunto formado por CSCW/Telemedicina/CORBA, as três áreas de pesquisa podem ser perfeitamente integradas para o fornecimento de serviços de qualidade. No contexto deste projeto podemos dizer que a telemedicina usa recursos de

CSCW para o fornecimento de serviços médicos à distância e o CSCW é implementado sobre a estrutura de objetos distribuídos do CORBA, fazendo com que as áreas estudadas sejam complementares no contexto estudado.



Considerando a definição de telemedicina é necessário abordar diversas áreas técnicas para alcançarmos os objetivos esperados. Conforme a Figura 33 temos o CORBA sobre rede TCP/IP desempenhando o papel dos recursos de telecomunicações, o CSCW mais a visualização de imagens médicas desempenhando o papel da tecnologia da informação e os médicos fornecendo o conhecimento na área de saúde.

Figura 33: Definição de telemedicina e tecnologias aplicadas.

## 5.5. Conclusões

O protótipo desenvolvido apresentou resultados satisfatórios do ponto de vista técnico. A abordagem para estruturação de programação facilitou muito o desenvolvimento, pois uma vez implementados os componentes utilizados para a interação dentro do ambiente cooperativo, a adição de funcionalidades à aplicação gráfica foi simplificada, facilitando a manutenção e extensão da ferramenta. O tempo gasto com implementação decaiu conforme a estrutura básica foi sendo completada, o que comprova a real utilidade de abordagens adequadas de projeto e adoção de ferramentas baseadas em tecnologias de componentes e objetos distribuídos.

O funcionamento do ambiente comprovou a adequação de uma implementação CORBA na construção de aplicações CSCW e teleradiologia, mostrando que a tecnologia já se encontra evoluída o suficiente para a construção de tais aplicações, mesmo estando ainda em constante processo de especificação e renovação.

## 6. Conclusão e Trabalhos Futuros

Neste capítulo apresentamos as considerações finais e perspectivas de trabalhos futuros.

### 6.1. Contribuições deste Trabalho

O maior contribuição deste trabalho foi a pesquisa, projeto, desenvolvimento e avaliação de um sistema CSCW para teleradiologia incorporando tecnologia avançada de objetos distribuídos com vantagens intrínsecas como: portabilidade, expansibilidade, e execução sobre ambientes heterogêneos.

O estudo inicial proporcionou a pesquisa do estado da arte das áreas envolvidas neste trabalho. Aplicações CSCW reúnem diversas áreas técnicas de atuação, sendo que estas áreas devem ser desenvolvidas de forma adequada para aplicações médicas. Entre as áreas estudadas temos:

- \* CSCW;
- \* Telemedicina;
- \* Objetos distribuídos e CORBA;
- \* Arquiteturas cliente servidor;
- \* Visualização de dados médicos;
- \* Interfaces gráficas;
- \* Metodologias de projeto.

A reunião das diversas áreas estudadas possibilitou a criação de um protótipo de sistema cooperativo para radiologia sobre uma configuração de ambiente heterogêneo, porém perfeitamente integrado graças à aplicação de padrões de objetos distribuídos como o CORBA e metodologias adequadas de projeto.

A utilização do CORBA trouxe diversos benefícios já discutidos anteriormente neste texto, porém mesmo sendo aprovado como um padrão ISO, ainda existem algumas desvantagens em sua utilização. O CORBA é um padrão em constante aperfeiçoamento,

havendo sempre a perspectiva de uma versão mais completa.

Ao iniciar este trabalho, optou-se por usar a implementação ORBacus 3 compatível com a especificação CORBA 2, contudo surgiram atualizações seguidas de novas implementações, sendo liberada a implementação ORBacus 4, compatível com a especificação CORBA 2.3. Hoje a OMG está pronta para lançar a especificação CORBA 3, que cobre áreas como integração com *internet*, qualidade de serviço e arquitetura de componentes CORBA (*CORBAcomponent Architecture*).

Apesar dos visíveis benefícios da evolução de diversas versões do CORBA lançadas, muitas vezes é difícil manter uma aplicação compatível com a especificação mais recente. Fatores como tempo e tamanho da equipe de projeto são cruciais para a atualização do projeto. Apesar da especificação mais antiga atender aos requisitos básicos de projeto, as novidades sempre acabam sendo interessantes em algum momento.

No caso desta pesquisa, seria interessante já possuir a parte referente à qualidade de serviço especificada desde o início, contudo devido à tardia especificação este constitui um dos aspectos deixado para ser abordado em trabalhos futuros. A adaptação para a mais nova especificação CORBA não deve ser muito trabalhosa, porém requer tempo de treinamento e implementação não disponíveis no momento. Ao adaptar a aplicação atual para a especificação mais recente, primeiramente é necessário porta-la para a nova implementação para então poder incorporar novas funcionalidades referentes à novidades. Procurou-se modelar o sistema de forma que a mudança seja o mais suave possível.

No que se refere a telemedicina, o protótipo implementado apresenta características suficientes para início do processo de adaptação dos usuários alvo para novas rotinas de trabalho. Além de treinamento a implantação de um protótipo ajuda na avaliação das funcionalidades oferecidas, permitindo ao usuário expor sua impressão a respeito da ferramenta e fornecendo mais parâmetros para uma implementação final do ambiente cooperativo para medicina.

É importante ressaltar que para a criação do protótipo desta pesquisa foi realizado um

levantamento das características do processo atual de diagnóstico por imagens e das características necessárias em uma aplicação médica de trabalho em grupo.

Espera-se que a ferramenta implementada segundo as características levantadas facilite a interação síncrona entre médicos, permitindo a visualização de exames tomográficos e de ressonância magnética para avaliação de casos onde seja necessária uma segunda opinião. A avaliação do ambiente cooperativo como ferramenta diária de trabalho constitui mais um linha de trabalhos futuros a serem realizados.

Do ponto de vista do CSCW o desenvolvimento do ambiente cooperativo para medicina ajudou na compreensão de como estruturar a implementação de ferramentas destinadas ao trabalho em grupo em forma de componentes que possam ser reaproveitados na construção de diversas outras aplicações. Um novo trabalho interessante que poderia ser desenvolvido é a criação de um conjunto de ferramentas (*toolkits*) e a padronização de componentes específicos para tal nicho de aplicações. Seria muito interessante a criação de uma força de trabalho junto à OMG para CSCW, da mesma forma que hoje existem forças de trabalho para aplicações médicas (CORBAmed) ou para telecomunicações.

## **6.2. Trabalhos Futuros**

O ambiente implementado, por ser um protótipo, ofereceu recursos suficientes para o desenvolvimento e avaliação desta pesquisa, todavia com a colaboração de entidades de saúde ainda é possível realizar um estudo mais completo do impacto de tecnologias como o CSCW no uso clínico cotidiano.

Um importante trabalho ainda a ser realizado é a integração das ferramentas desenvolvidas com uma base de dados médicos, com o propósito de fornecer os dados necessários para o trabalho dos grupos médicos e ainda fornecer uma base de conhecimento para auxílio na tomada de decisões. Esta integração é importante como suporte ao trabalho e distribuição de conhecimento.

A integração com a base de dados deve ser feita em ambiente *web*, o que amplia o acesso e facilita o treinamento, uma vez que os usuários já se encontram habituados com este ambiente de trabalho. A adaptação do ambiente implementado para a

integração com a base de dados médicas e a implantação em uma instituição de saúde, envolve trabalhos com a especificação CORBA 3, que prevê a utilização em Internet. Outro ponto a ser considerado é a adição de recursos para segurança, fator muito importante para a manutenção do sigilo dos dados médicos.

Após expandir o protótipo para implantação pode-se iniciar o trabalho de avaliação do ambiente de CSCW proposto dentro do ambiente hospitalar, possibilitando o estudo mais detalhado das vantagens e desvantagens do trabalho cooperativo em medicina.

Finalmente pretende-se incorporar as metodologias e o conhecimento deste trabalho no projeto “Telemedicina Contra o Câncer Infantil”, cujo objetivo é interligar instituições de atendimento oncológico pediátrico dispersas pelo país numa rede de Telemedicina incorporando tecnologia avançada de comunicação digital de informações para o apoio ao atendimento desta moléstia em regiões remotas do país.

### **6.3. Conclusões**

A pesquisa proposta neste trabalho não se resumiu apenas a implementação de um ambiente cooperativo para medicina, mas também ao levantamento das características necessárias a este ambiente considerando a situação brasileira no que se refere a telemedicina.

Definida a situação atual da telemedicina no Brasil, procurou-se propor uma solução considerando aspectos como custo, facilidade de adaptação do usuário e facilidade de manutenção e expansão. A utilização de padrões e metodologias de projeto foi muito importante para que tais objetivos fossem alcançados.

Este trabalho permitiu a pesquisa do estado da arte das técnicas e tecnologias envolvidas na criação de ferramentas para telemedicina, o desenvolvimento de um protótipo para avaliação da utilização do CSCW em medicina e o desenvolvimento e integração de aplicações em ambientes de redes heterogêneas.

As tecnologias utilizadas no trabalho preencheram as expectativas iniciais, mostrado-se adequadas à aplicação proposta levando a resultados satisfatórios. Algumas adaptações ainda são necessárias, porém com a rápida evolução tecnológica e a abordagem de

componentes assumida, tais adaptações não requerem um esforço muito grande.



## 7. Bibliografia

- [Agha93] Agha, Gul; Frolund, Svend; Kim, Woo Young; "Abstraction and Modularity Mechanisms for Concurrent Computing"; IEEE Parallel & Distributed Tecnology; May 1993;
- [Alhir98] Alhir, Sinan Si; "UML in a Nutshell - A desktop Quick Reference"; 1998; 1 edição
- [Anupam94] Anupam, Vinod; Bajaj, Chandrajit; "Shastra: Multimedia Collaborative Desing Environment"; IEEE Multimedia; Summer 1994; Vol. 1; N 2
- [ACR Standards 99] ACR Standards; "ACR Standard for Teleradiology"; 1994 (Res. 21) Effective 1999
- [Bai97] Bai, Jing; Hu, Bingyi; Zhang, Yonghong; Ye, Datian; "A Communication Server for Telemedicine Applications"; IEEE Transactions on Information Technology In Biomedicine; September 1997; Vol. 1; N 3
- [Blattner94] Blattner, Meera M.; Livermore, Lawrence; "In Our Image: Iterface Design in the 1990s"; IEEE Multimedia; Spring 1994; Vol. 1; N 1
- [Brooks93] Brooks, Ruven; "The Case For The Specialized Interface"; IEEE Software; March 1993; Vol. 10; N 2
- [Bucci94] Bucci, Giacomo; Detti, Ricardo; Pasqui, Valdo; Nativi, Stefano; "Sharing Multimidia Data Over a Client/Server Network"; IEEE Multimedia; Fall 1994; Vol. 1; N 3
- [Burleigh93] Burleigh, Scott; "ROME: Distributing C++ Object System"; IEEE Parallel & Distributed Tecnology; May 1993
- [Caribom92] Caribom, Ingrid; Hsu, Willian M.; Klinker, Gudrum; "Modeling and Analysis of Empirical Data in Collaborative Environments"; Communications of the ACM; June 1992; Vol. 35; N 6
- [Chen96] Chen, Jim X.; Rine, David; Simon, Horst D.; "Advancing Interactive Visualization and Computational Steering"; Computational Science & Engineering; Winter 1996; Vol. 3; N 4
- [Chronaki97] Chronaki, Catherine E.; Katehakis, Dimitrios G.; Zabulis, Xenophon C.; Tsiknakis, Manolis; Orphanoudakis, Stelios C.; "WebOnCOLL: Medical Collaboration in Regional

- Healthcare Networks"; IEEE Transactions on Information Technology In Biomedicine; December 1997; Vol.1 N 4
- [Cleynenbreugel96] Cleynenbreugel, J. Van; Bellon, E.; Marchal, G.; "Annotating Radiological Images for Computer Assisted Communication and Teaching"; Computer Communications; 1996; N 19
- [Comer93] Comer, Douglas E.; Stevens, David L.; "Internetworking With TCP/IP - Client/Server Programming and Applications/Volume 3"; 1993; 1 edição
- [Cooper96] Cooper, Alan; "Goal-Directed Software Design"; Dr. Dobb's Journal; 1996; N 251
- [Cosic97] Cosic, Domagoj; "An Open Medical Imaging Workstation Architecture for Platform-Independent 3-D Medical Image Processing and Visualization"; IEEE Transactions on Information Technology In Biomedicine; December 1997; Vol. 1; N 4
- [Cypser92] Cypser, R. J.; "Communications for Cooperating Systems OSI, SNA and TCP/IP"; 1992; 2 edição
- [Detmer97] Detmer, William M.; "Using the Internet to Improve Knowledge Diffusion in Medicine"; Communications of the ACM; August 1997; Vol. 40; N 8
- [Disz95] Disz, Terrence L.; Evard, Remy; Henderson, Mark W.; Nickless, William; Olson, Robert; Papka, Michael E.; Stevens, Rick; "Designing the Future of Collaborative Science: Argonne's Futures Laboratory"; IEEE Parallel & Distributed Technology; Summer 1995; Vol.3; N 2
- [Ellis91] Ellis, C. A.; Gibbs S. J.; Rein, G. L.; "Groupware: Some Issues and Experiences"; Communications of the ACM; January 1991; Vol. 34; N. 1;
- [Elvins92] Elvins, T. Todd; "A Survey of Algorithms for Volume Visualization"; Computer Graphics; August 1992; Vol. 26; N 2
- [Field97] Field, Marilyn J.; "Telemedicine: A Guide to Accessing Telecommunications in Healthcare"; 1997; 2 edição
- [Friesen95] Friesen, Jerrold A.; Yang, Christine L.; Cline, Jr, Raymond E.; "DAVE: A Plug-and-Play Model for Distributed Multimedia Application Development"; IEEE Parallel & Distributed Technology; Summer 1995; Vol. 3; N 2

- [Gennari] Gennari, H. John; Stein, Adan R.; Musen, Mark A.; "Reuse For Knowledge-Based Systems and CORBA Components"; Section on Medical Informatics - Stanford University School Of Medicine
- [Gidney94] Gidney, Eric; Chandler, Annmarie; McFarlane, Greg; "CSCW for Film and TV Preproduction"; IEEE Multimedia; Spring 1994; Vol. 1; N 1
- [Gorton95] Gorton, Ian; Gray, Jonathan P.; Jelly, Innes; "Object-Based Modelig of Parallel Programs"; IEEE Parallel & Distributed Tecnology; Summer 1995; Vol. 3; N 2
- [Grimshaw93] Grimshaw, Andrew S.; Strayer, W. Timothy; Narayan, Padmini; "Dynamic, Object-Oriented Parallel Processing"; IEEE Parallel & Distributed Tecnology; May 1993
- [Grudin88] Grudin, Jonathan; "Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces"; Conference on Computer-Supported Cooperative Work; September 1988
- [Henning98] Henning, Michi; "Binding, Migration, and Scalability in CORBA"; Communications of the ACM; October 1998; Vol. 41; N 10
- [Henning99] Henning, Michi; Vinoski, Steve; "Advanced CORBA Programming With C++"; 1999; 3 edição
- [Horii96] Horii, Steven C.; Bidgood, Jr, W. Dean; "Network and ACR-NEMA Protocols"; IEEE Computational Science & Enineering; Winter 1996; Vol. 3; N 4
- [Jadav95] Jadav, Divyesh; Choudhary, Alok; "Designing and Implementing High-Performance Media-on-Demand-Servers"; IEEE Parallel & Distributed Tecnology; Summer 1995; Vol. 3; N 2
- [Kleinholz94] Kleinholz, Lutz; Virchow, Rudolf; Ohly, Martin; "Supporting Cooperative Medicine: The Bermed Project"; IEEE Multimedia; Winter 1994; Vol. 1; N 4
- [Kobara91] Kobara, Shiz; "Visual Design With OSF/Motif"; 1991; 1 edição
- [Korth95] Korth, Henry F.; Silberschatz, Abraham; "Sistema de Banco de Dados"; 1995; 2 edição
- [Levy91] Levy, Henry M.; Tempero, Ewan D.; "Modules, Objects and Distributed Programming: Issues in RPC and Remote Object Invocation"; Software - Practice and Experience;

January 1991; Vol. 21; N 1

- [Little94] Little, Thomas D. C.; Venkatesh, Dinesh; "Prospects for Interactive Video-on-Demand"; IEEE Multimedia; Fall 1994; Vol. 1; N 3
- [Lou97] Lou, S. L.; Sickles, Edward A.; Huang, H. K.; Hoogstrate, David; Cao, Fei; Wang, Jun; Jahangiri, Mohammad; "Full-Field Direct Digital Telemammography: Technical Components, Study, Protocols, and Preliminary Results"; IEEE Transactions on Information Techn
- [Marca92] Marca, David; Bock, Geoffrey; "Groupware: Software for Computer-Supported Cooperative Work"; 1992; 2 edição
- [Mercurio92] Mercurio, Philip; Elvins, T. Todd; Young, Stephen J.; Cohen, Philip S.; Fall, Kevin R.; Ellisman, Mark H.; "The Distributed Laboratory : An Interactive Visualization Environment for Electron Microscopy and 3D Imaging"; Communications of the ACM; June 1992; Vol. 35; N6
- [Narasimhan99] Narasimhan, Priya; Moser, Louise E.; Smith, Melliar; "Using Interceptors to Enhance CORBA"; Computer; July 1999; Vol. 32; N 7
- [Olson91] Olson, Gary M.; Olson, Judith S.; "User Centered Design of Collaboration Technology"; Journal of Organizational Computing; 1991; Vol. 1; N 1
- [OMG97] OMG, Object Management Group; "A Discussion of the Object Management Architecture"; 1997
- [OMG97-R2.1] OMG, Object Management Group; "The Common Object Request Broker: Architecture and Specification - Revision 2.1"; 1997
- [Orfali95] Orfali, Robert; Harkey, Dan; "Client/Server with Distributed Objects"; Byte; April 1995
- [Orfali96] Orfali, Robert; Harkey, Dan; Edwards, Jeri; "The Essential Distributed Objects Survival Guide"; 1996; 1 edição
- [Orfali98] Orfali, Robert; Harkey, Dan; "Client/Server Programming with CORBA"; 1998; 2 edição
- [Orfali99] Orfali, Robert; Harkey, Dan; Edwards, Jeri; "Client/Server Survival Guide"; 1999; 3 edição
- [Patel] Patel, Vimla L.; Allen, Vanessa G.; Arocha, José F.; "Representing Clinical Guidelines in GLIF Individual and Collaborration Expertise"; Section on Medical Informatics

- Stanford University School Of Medicine

- [Pavlakos93] Pavlakos, Constantine J.; Schoof, Larry A.; Mareda, John F.; "A Visualization Model for Supercomputing Environments"; IEEE Parallel & Distributed Tecnology; October 1993; Vol. 1; N 4
- [Raghupathi97] Raghupathi, W.; "Health Care Information Systems"; Communications of the ACM; August 1997; Vol. 40; N 8
- [Rindfleisch97] Rindfleisch, Thomas C.; "Privacy, Information Tecnology, and Health Care"; Communications of the ACM; August 1997; Vol. 40; N 8
- [Rodden92] Rodden, Tom; Blair, Gordon S.; "Distributed Systems Support for Computer Supported Cooperative Work"; Computer Communications; October 1992; Vol. 15; N 8
- [Rumbaugh91] Rumbaugh, James; Blaha, Michael; Premerlani, William; Eddy, Frederick; Lorensen, Willian; "Object Oriented Modeling and Design"; 1991; 1 edição
- [Sakas95] Sakas, Georgios; Schreyer, Lars-Arne; Grimm, Marcus; "Preprocessing and Volume Rendering of 3D Ultrasonic Data"; IEEE Computer Graphics and Applications; July 1995; Vol. ; N 0
- [Santarelli97] Santarelli, Maria Filomena; Positano, Vincenzo; Landini, Luigi; "Real-Time Multimodal Medical Image Processing: A Dynamic Volume-Rendering Application"; IEEE Transactions on Information Technology In Biomedicine; September 1997; Vol. 1; N 3
- [Schmidt98] Schmidt, Douglas C.; "Evaluating Architectures for Multithreaded Object Request Brokers"; Communications of the ACM; October 1998; Vol. 41; N 10
- [Seetharaman98] Seetharaman, Krishnan; "The CORBA Connection"; Communications of the ACM; October 1998; Vol. 41; N 10
- [Shah97] Shah, Pinkesh J.; Martinez, Ralph; Zeigler, Bernard P.; "Design, Analysis, and Implementation of a Telemedicine Remote Consultation and Diagnosis Session Playback Usisng Discrete Event System Specification"; IEEE Transactions on Information Technology In
- [Shortliffeyy] Shortliffe, Edward H.; Patel, Vimla L.; Cimino, James J.; Barnett, Octo G.; Greenes, Robert A.; "A Study of Collaboration Among Medical Information Research Laboratory"; Section on Medical Informatics - Stanford

University School Of Medicine

- [Siegel98] Siegel, Jon; "OMG Overview: CORBA and OMA in Enterprise Computing"; Communications of the ACM; October 1998; Vol. 41; N 10
- [Siegel99] Siegel, Jon; "A Preview of CORBA 3.0"; Computer; May 1999; Vol. 32; N 5
- [Sinha92] Sinha, Alok; "Client - Server Computing"; Communications of the ACM; July 1992; Vol. 35; N 7
- [Stevens94] Stevens, W. Richard; "Advanced Programming in the Unix Environment"; 1994; 8 edição
- [Taylor95] Taylor, Herb; Chin, Danny; Knight, Stan; "The Magic Video-on-Demand Server and Real-Time Simulation System"; IEEE Parallel & Distributed Technology; Summer 1995; Vol. 3; N 2
- [Vinoski98] Vinoski, Steve; "New Features for CORBA 3.0"; Communications of the ACM; October 1998; Vol. 41; N 10
- [Zuffo97] Zuffo, Marcelo Knörich; "Um Ambiente de Programação de Alto Desempenho para Visualização Volumétrica"; 1997;

## 8. Endereços Eletrônicos Citados

1. CCS-SIS (Consócio de Componentes de Software para Sistemas de Informação em Saúde) <http://www.datasus.gov.br/ccsis/>
2. CORBA/ORBacus <http://www.ooc.com/products/orbacus.html>
3. CORBAMED Latin America <http://www.sucesusp.com.br/corbamed/>
4. DICOM Home page <http://medical.nema.org/dicom.html>
5. Grupo de usuários CORBA <http://www.sucesusp.com.br/html/grupos/corba/>
6. IBM BookManager® BookServer Library <http://www.raleigh.ibm.com:80/cgi-bin/bookmgr/library>
7. EZ311300 - Understanding OSF DCE 1.1 <http://www.raleigh.ibm.com:80/cgi-bin/bookmgr/BOOKS/EZ311300/CCONTENTS>
8. JAVA/SUN <http://www.java.sun.com/>
9. National Electrical Manufacturers Association <http://www.nema.org/>
10. OMG <http://www.omg.org>
11. SIG GROUP (Special Interest Group on Supporting Group Work) <http://www.acm.org/siggroup/>
12. Stanford Medical Informatics <http://smi-web.stanford.edu/>
13. TeleMed <http://www.acl.lanl.gov/TeleMed/>

## 9. Anexo 1 - Dicionário de Dados

### \* Tabela: Applications

Nome	Tipo	Descrição
» App_ID	Long Integer	Código da aplicação
App_Name	Char[8]	Nome da aplicação (No serviço de nomes)
App_IOR	Char[]	IOR da aplicação
App_Permissions	Integer	Permissões de acesso à aplicação
Domain_ID pertence	Long Integer	Código do domínio ao qual aplicação

### \* Tabela: Diary

Nome	Tipo	Descrição
» Diary_ID	Long Integer	Código do evento agendado
Diary_DateTime	Date/Time	Data e horário do evento
Diary_Title	Char[50]	Título do evento
Diary_Description	Char[256]	Descrição do evento
User_ID	Long Integer	Código do usuário dono do diário

### \* Tabela: Domains

Nome	Tipo	Descrição
» Domain_ID	Long Integer	Código do domínio
Domain_Name	Char[8]	Nome do domínio
Domain_IOR	Char[]	IOR do domínio
Domain_Description	Char[256]	Texto descritivo sobre o domínio
Domain_Restrictions	Long Integer	Restrições de acesso externo

### \* Tabela: Groups

Nome	Tipo	Descrição
» Domain_ID pertence	Long Integer	Código do domínio ao qual o grupo
Group_ID	Long Integer	Código do grupo
Group_Name	Char[8]	Nome do grupo
Group_Decription grupo	Char[256]	Texto com pequena descrição sobre o grupo

### \* Tabela: Messages

Nome	Tipo	Descrição
» Message_ID	Long Integer	Código da mensagem
Message_Subject	Char[50]	Assunto da mensagem
Message_Data	Date/Time	Data da mensagem



Message_Expires	Date/Time	Data final
Message_Text	Char[256]	Corpo da mensagem
User_ID	Long Integer	Código do usuário dono do mensagem

\* **Tabela: PatientData**

Nome	Tipo	Descrição
» PatientData_ID	Long Integer	Código do paciente
PatientData_Type	Byte	Tipo do exame
PatientData_Format	Byte	Formato do dado
PatientData_Creation	Date/Time	Data da criação do registro
PatientData_LastUpdate	Date/Time	Data da última alteração
PatientData_Value	BLOB	BLOB que representa o dado
Patient_ID	Long Integer	Código do paciente dono do dado

\* **Tabela: Patients**

Nome	Tipo	Descrição
» Patient_ID	Long Integer	Código do paciente
Patient_Name	Char[100]	Nome completo do paciente
Patient_Birth	Date/Time	Data de nascimento
Patient_Sex	Char[1]	Sexo
PatientDescription	Char[256]	Comentários
Patient_DocumentNo	Char[20]	Número do documento de identificação
Patient_DocType	Byte	Tipo do documento de identificação.
Domain_ID	Long Integer	Domínio ao qual o paciente pertence

\* **Tabela: TimeTable**

Nome	Tipo	Descrição
» User_ID	Long Integer	Código do usuário dono da tabela de horários
TimeTable_Weekday	Char[7]	Dia da semana
TimeTable_From	Date/Time	Horário de início
TimeTable_To	Date/Time	Horário de fim
TimeTable_ready	Boolean	Disponibilidade
TimeTable_Decription	Char[256]	Comentários

\* **Tabela: Users**

Nome	Tipo	Descrição
» User_ID	Long Integer	Código do usuário
User_Name	Char[256]	Nome completo do usuário
User_Login	Char[8]	Nome <i>login</i> do usuário
User_Passwd	Char[8]	Senha de acesso ao sistema
User_Description	Char[256]	Destinado a comentários
User_Type	Char[1]	Tipo de usuário
Group_ID	Long Integer	Código do grupo do usuário.