

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Uma Proposta**  
**para o Gerenciamento de Aplicações em Rede**

**Luiz Fernando Tavares Meirelles**

Florianópolis  
1997

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Uma Proposta**  
**para o Gerenciamento de Aplicações em Rede**

por

**Luiz Fernando Tavares Meirelles**

Dissertação apresentada ao Curso de Mestrado em  
Ciência da Computação da Universidade Federal de  
Santa Catarina, como requisito parcial à obtenção do  
título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientadora: Prof<sup>a</sup> Liane M. R. Tarouco

Co-Orientadora: Prof<sup>a</sup> Elizabeth S. Specialski

Florianópolis  
1997

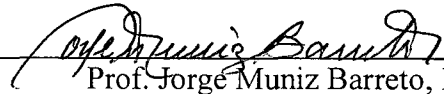
# Uma Proposta para o Gerenciamento de Aplicações em Rede

**Luiz Fernando Tavares Meirelles**

Esta dissertação foi julgada adequada para obtenção do Título de

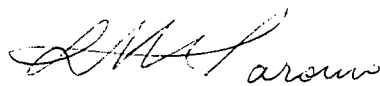
## MESTRE EM CIÊNCIA DA COMPUTAÇÃO

Especialidade Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

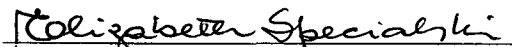


Prof. Jorge Muniz Barreto, Dr.  
Coordenador do Curso

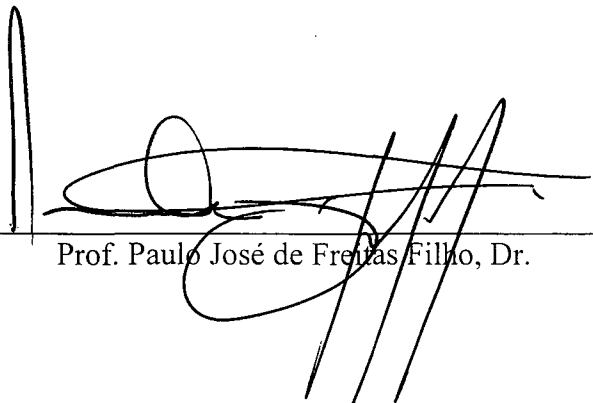
### BANCA EXAMINADORA:



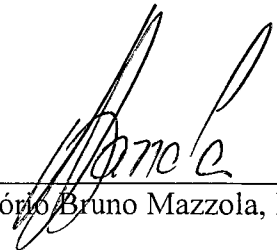
Prof<sup>a</sup> Liane Margarida Rockenbach Tarouco, Dr<sup>a</sup>.  
Orientadora



Prof<sup>a</sup> Elizabeth Sueli Specialski, M.Sc.  
Co-Orientadora



Prof. Paulo José de Freitas Filho, Dr.



Prof. Vitorio Bruno Mazzola, Dr.

**Aos amigos e parentes, por todo o apoio e incentivo.**

**Em especial, à minha Esposa Geovana  
e a meu Filho Lucas,  
pois conseguiram superar minha ausência,  
compreender minhas inquietudes e  
amenizar os períodos de dificuldades  
com amor e dedicação.**

## Agradecimentos

Inicialmente, quero agradecer aos Professores Jandir João Zonotelli, Aroldo Roberto Peduzzi e Circe Cunha, pelo apoio para que o projeto fosse desenvolvido nas instalações da Universidade Católica de Pelotas.

Aos Acadêmicos, agora Analistas de Sistemas, Alexandre, Ane e Márcia, cujos temas dos projetos de graduação ajudaram na fundamentação e amadurecimento desta dissertação, pela oportunidade de tê-los como orientandos.

Aos Bolsistas de Iniciação Científica, Davi, Diego, Iuri e Luciano, pelo companheirismo e pelo apoio prestado durante a realização deste trabalho.

Às Professoras Liane Tarouco e Elizabeth Specialski, pelos ensinamentos durante o desenvolvimento do projeto, os quais foram fundamentais no alcance dos objetivos propostos.

Finalmente, mas não em último lugar, agradeço a **Deus** por mais esta oportunidade de crescimento pessoal.

# Sumário

<b>1</b>	<b>Introdução</b> .....	<b>1</b>
1.1	Motivação .....	2
1.2	Organização do trabalho .....	4
<b>2</b>	<b>Gerenciamento <i>OSI</i></b> .....	<b>6</b>
2.1	Gerenciamento de Contabilização .....	6
2.1.1	Função de medida de uso .....	7
2.1.2	Atributos de controle de medida de uso .....	9
2.1.3	Atributos de dados de medida de uso .....	10
2.2	Gerenciamento de Segurança .....	11
2.2.1	Função de relatório de alarme de segurança .....	11
2.2.2	Função de registro para auditoria de segurança .....	13
<b>3</b>	<b>Gerenciamento <i>INTERNET</i></b> .....	<b>15</b>
3.1	O modelo de gerenciamento <i>SNMP</i> .....	15
3.2	Aplicação Gerente .....	18
3.2.1	Operações de gerenciamento .....	19
3.2.2	<i>Management Information Base - MIB</i> .....	20
3.2.3	Bando de Dados .....	21
3.2.4	Aplicações de gerenciamento .....	21
3.2.5	Interface de usuário .....	22
3.3	Aplicação Agente .....	23

3.3.1	O Agente .....	23
3.3.1.1	Pilha de transporte .....	25
3.3.1.2	<i>Engine</i> de protocolo .....	26
3.3.1.3	<i>Dispatch table</i> .....	26
3.3.2	Diagrama de um Agente .....	27
3.3.3	Estrutura de Agente .....	30
3.3.4	<i>MIB</i> do Agente .....	30
3.4	Operações e Mensagens <i>SNMP</i> .....	31
3.5	O Pacote de Gerenciamento <i>CMU-SNMP</i> .....	32
3.5.1	Os componentes de uma Aplicação Gerente .....	33
3.5.2	Os componentes de uma Aplicação Agente .....	33
<b>4</b>	<b>Gerenciamento Baseado em <i>Web</i></b> .....	<b>34</b>
4.1	A <i>World Wide Web</i> .....	35
4.2	Integrando o Gerenciamento de rede à <i>Web</i> .....	35
4.3	Aspectos a considerar para o gerenciamento de rede via <i>Web</i> .....	37
4.4	Migrando para o Gerenciamento Baseado em <i>Web</i> .....	38
4.4.1	Plataformas de Gerenciamento de Redes .....	39
4.4.2	<i>Web-Based Enterprise Management</i> .....	40
4.4.3	<i>SNMP over HTTP</i> .....	40
4.4.4	<i>Java Management API</i> .....	41
<b>5</b>	<b>Proposta para o gerenciamento de aplicações em rede</b> .....	<b>43</b>
5.1	Identificação dos requisitos de usuário (atividade 1) .....	44
5.2	Análise dos requisitos de usuário (atividade 2) .....	45
5.3	Análise das aplicações de rede (atividade 3) .....	46
5.4	Análise das <i>MIBs</i> existentes (atividades 4) .....	48
5.5	Especificação funcional para ferramentas de gerenciamento de aplicações (atividade 5) .....	49
5.5.1	Abordagens e arquiteturas de gerenciamento .....	49
5.5.2	Informações de gerenciamento .....	50

5.5.3	Funções de gerenciamento .....	51
5.6	Implementação de aplicações de gerenciamento (atividade 6) .....	54
5.6.1	Implementação de Aplicações Gerentes .....	54
5.6.2	Implementação de Aplicações Agentes .....	55
<b>6</b>	<b>Aplicação da proposta no gerenciamento do <i>HTTP</i></b> .....	<b>56</b>
6.1	Identificação dos requisitos de usuário .....	56
6.2	Análise dos requisitos de usuário .....	57
6.3	Análise das aplicações de rede .....	59
6.3.1	Protocolo <i>HTTP</i> .....	60
6.3.1.1	Mensagens do <i>HTTP</i> .....	62
6.3.1.2	Códigos de <i>status</i> .....	62
6.3.2	Capacidades de <i>logging</i> das implementações de servidores <i>HTTP</i> .....	64
6.3.3	Parâmetros de configuração das implementações de servidores <i>HTTP</i> .....	65
6.4	Análise das <i>MIBs</i> existentes .....	66
6.4.1	Propostas de objetos gerenciáveis para o <i>HTTP</i> .....	66
6.4.2	Aplicabilidade das <i>MIBs</i> padronizadas .....	70
6.4.3	Sobreposição de objetos .....	73
6.5	Especificação funcional para ferramentas de gerenciamento de aplicações .....	75
6.5.1	Abordagens e arquiteturas de gerenciamento .....	75
6.5.2	Informações de gerenciamento .....	76
6.5.3	Funções de gerenciamento .....	77
6.6	Implementação de aplicações de gerenciamento .....	77
6.6.1	Implementação de Aplicações Gerentes .....	79
6.6.2	Implementação de Aplicações Agentes .....	85
<b>7</b>	<b>Conclusões e trabalhos futuros</b> .....	<b>87</b>
7.1	Considerações e contribuições .....	87
7.2	Trabalhos futuros .....	89



<b>Referências Bibliográficas</b> .....	<b>91</b>
<b>Anexos</b> .....	<b>95</b>
<b>A - Descrição da <i>MIB</i> em <i>ASN.1</i></b> .....	<b>95</b>
<b>B - Descrição dos módulos do protótipo</b> .....	<b>107</b>

## Lista de Figuras

2.1	Relacionamento entre os processos que formam a atividade de contabilização .....	08
2.2	Modelo para Função de relatório de alarme de segurança .....	12
3.1	Modelo <i>SNMP</i> para o gerenciamento de rede .....	16
3.2	Tipos de aplicações de gerenciamento e forma de comunicação .....	17
3.3	Componentes de uma Aplicação Gerente .....	18
3.4	Adições de gerenciamento <i>SNMP</i> em um subsistema .....	25
3.5	Áreas funcionais de um Agente <i>SNMP</i> .....	28
3.6	Funções desempenhadas por um Agente <i>SNMP</i> .....	29
3.7	A <i>MIB</i> como um banco de dados .....	31
4.1	Visão simplificada do funcionamento da <i>Web</i> .....	35
4.2	Tecnologias em fase de migração para <i>Web</i> .....	38
4.3	Esquema para troca de mensagens entre o Agente <i>SNMP</i> e o <i>Web Browser</i> .....	41
4.4	Agentes de Gerenciamento de Rede suportados por <i>Java</i> .....	42
5.1	Ciclo de vida das atividades propostas .....	43
6.1	Dados utilizados nas demais atividades da proposta .....	57
6.2	Arquitetura genérica das aplicações de rede .....	59
6.3	Decomposição funcional para as aplicações de rede <i>Client-Server</i> .....	60
6.4	Troca de mensagens entre Cliente e Servidor <i>HTTP</i> .....	61
6.5	Componentes de uma mensagem <i>HTTP</i> .....	62
6.6	Sobreposição de objetos entre a <i>WWW-MIB</i> e <i>SYSAPPL-MIB</i> .....	73
6.7	Sobreposição de objetos entre a <i>WWW-MIB</i> e <i>NSM-MIB</i> .....	74
6.8	Sobreposição de objetos entre a <i>WWW-MIB</i> e <i>HR-MIB</i> .....	74

6.9 Especificação do Sistema em <i>SDL</i> (nível abstrato) .....	78
6.10 <i>Backbone</i> da Rede TCHÊ .....	79
6.11 Ferramenta utilizada para construção do Banco de Dados .....	80
6.12 Diagrama de <b>entidade-relacionamento</b> do Banco de Dados .....	83
6.13 Interface de abertura do protótipo de sistema .....	85
B.1 Interface de abertura do protótipo .....	107
B.2 Interface para o cadastramento de sistemas que serão gerenciados .....	108
B.3 Interface para confirmação e atualização dos dados dos sistemas no B D .....	109
B.4 Interface para o cadastramento das aplicações de rede .....	109
B.5 Interface para confirmação e atualização dos dados das aplicações .....	110
B.6 Interface para solicitação de informações de gerenciamento .....	111
B.7 Interface com a exibição dos relatórios de alarme de segurança .....	113
B.8 Interface com o indicador de confiabilidade da aplicação .....	114
B.9 Interface com a relação dos principais domínios que acessam a aplicação .....	115

## Lista de Tabelas

3.1 Documentos que definem a estrutura do modelo <i>SNMPv1</i> e <i>v2</i> .....	17
3.2 Operações do protocolo <i>SNMP</i> .....	32
3.3 Operações e aplicações de gerenciamento do <i>CMU-SNMP</i> .....	33
6.1 Compilação dos dados levantados junto aos usuários .....	57
6.2 Código de <i>status</i> especificados na versão 1.1 do <i>HTTP</i> .....	63
6.3 Classificação dos <i>status</i> do <i>HTTP</i> em relação aos requisitos do <i>OSI-RM</i> .....	64
6.4 Grupos e números de objetos da <i>HTTP MIB</i> e da <i>WWW MIB</i> .....	69
6.5 Requisitos de gerência para o Modelo Operacional e o de Serviços .....	71
6.6 Identificação de informações gerenciais presentes em <i>MIBs</i> padronizadas .....	72
6.7 Capacidades e limitações decorrentes da escolha da arquitetura .....	75

## Resumo

Este trabalho apresenta uma proposta para o gerenciamento de aplicações em rede. No escopo do projeto que foi desenvolvido, considerou-se como aplicações em rede, todo programa que desempenha o papel de *client/server*, ou ambos, e que é utilizado no oferecimento de serviços *Internet*. A proposta compreende, basicamente, o desenvolvimento de atividades relacionadas à identificação e análise dos requisitos de usuários; análise das aplicações de rede; análise das *MIBs*; especificação e implementação de um sistema de gerenciamento.

Para validar a proposta, foram considerados aspectos relacionados ao *OSI-RM* e à abordagem de gerenciamento *SNMP*. No *OSI-RM* buscaram-se os requisitos para gerência de contabilização e de segurança. O *SNMP* foi adotado como protocolo de gerenciamento, sendo a implementação dos elementos previstos no modelo de gerência da *IETF*, efetuada juntamente com uma abordagem de *Web-based Network Management*.

Como aplicação a ser gerenciada, escolheram-se implementações do *HTTP* em execução junto a *sites* com recursos computacionais e ambiente operacional heterogêneos. Uma *MIB* foi especificada, contendo grupos de objetos que mantêm dados sobre medidas de uso e eventos de segurança.

No desenvolvimento do protótipo de sistema para o gerenciamento de aplicações em rede, foram utilizados os sistemas operacionais *Linux*, *AIX* e *OSF/1*; as linguagens *C*, *Perl* e *HTML*; um servidor de banco de dados padrão *SQL*, além do pacote *CMU-SNMP*.

## *Abstract*

This work presents a proposal for the management of applications in network. In the outskirts of the project that was developed, the whole program that carries out the client/server role, or both, and that is used in the offer of Internet services was considered as application in network. The proposal comprises, basically, the development of activities related to the identification and analysis of the users' requirements; analysis of the network applications; analysis of MIBs; specification and implementation of a management system.

To validate the proposal, aspects related OSI-RM and the SNMP management approach were considered. In OSI-RM the requirements were looked for accountancy management and for security. SNMP was adopted as management protocol, being the implementation of the elements foreseen in the model of management of IETF made together with an approach of Web-based Network Management.

As application to be managed, was chosen implementation of the HTTP in execution sites were chosen with computational resources and heterogeneous operational environment. A MIB was specified, containing groups of objects that maintain data about use measures and security events.

In the development of the system prototype for the management of applications in network, the operating systems Linux, AIX and OSF/1; the languages C, Perl and HTML; a database server standard SQL, besides the CMU-SNMP package were used.

# Capítulo 1

## Introdução

A *Internet* tem proporcionado à sociedade inúmeros benefícios. O uso de suas aplicações em rede tem sido imprescindível na realização de tarefas em vários ambientes de trabalho. A utilização harmônica das tecnologias de telecomunicações e informática, de natureza e porte diferentes, vem atingido estágios muitas vezes inesperados. Atualmente, com o uso dessas tecnologias, é possível atender milhões de usuários, cujos interesses e necessidades de uso são suportados por aplicações de rede, principalmente as relacionadas ao fornecimento de serviços de informações.

As inúmeras redes interligadas ao que chamamos de *Internet*, receberam, desde o início de suas conexões, especial atenção para o controle e monitoramento dos dispositivos de *hardware* e *software*, dispostos nas camadas inferiores da arquitetura de rede utilizada. Atualmente, existe uma inversão de preocupações - inúmeras pesquisas e produtos comerciais estão sendo desenvolvidos para o gerenciamento das camadas superiores, principalmente a camada onde se encontram as aplicações de rede.

Dentre os esforços existentes, encontram-se os trabalhos desenvolvidos junto ao *Internet Engineering Task Force - IETF* pelo *Application MIB Working Group* [IETFa], o qual é responsável pela definição de um conjunto de objetos gerenciáveis para o monitoramento e controle de aplicações distribuídas. Esses objetos gerenciáveis deverão fornecer, especialmente, informações sobre a configuração (incluindo aplicações





dependentes e associações entre aplicações), falha (incluindo informações de *status*, tal como: *up*, *down* ou *degraded*) e desempenho (incluindo utilização de recursos) de aplicações distribuídas.

Durante a realização deste trabalho, o *Application MIB Working Group* havia produzido três *internet-drafts*: *Definitions of System-Level Managed Objects for Applications*, *Application Management MIB* e *Definitions of Management Objects for WWW Servers*.

Nas seções seguintes, apresenta-se a motivação deste trabalho, seguida da descrição da organização dos assuntos que são tratados nos demais capítulos.

## 1.1 Motivação

A definição de objetos, que representam abstrações dos diferentes dispositivos e serviços existentes em redes de computadores, compreende grande parte dos trabalhos desenvolvidos até o presente momento para o gerenciamento de aplicações. Embora possa ser considerada uma importante atividade, a especificação de módulos de informações de forma isolada, não permite o efetivo gerenciamento dos componentes existentes, independentemente da camada onde estão localizados. O gerenciamento (controle e monitoramento) somente é obtido com a adoção de ferramentas de *software*, desenvolvidas em conformidade com uma determinada abordagem/modelo de gerenciamento baseada em padrões internacionais.

Nos últimos anos, dois padrões destacaram-se no cenário de gerenciamento de rede. No final da década de 80 e início desta década [Pra95], iniciaram-se os trabalhos de padronização do chamado *Open System Interconnection - Reference Model (OSI - RM)*. Dentre os diferentes aspectos tratados pelos padrões de gerenciamento estabelecidos no modelo de referência *OSI*, destacam-se os documentos que dividem as atividades de gerenciamento em cinco grandes áreas funcionais e que definem os requisitos para as diferentes funções de gerenciamento. De outro lado, encontra-se a

abordagem de gerenciamento que é padronizada pela *IETF*, cujo início dos trabalhos foi efetuado em 1987. Diferentemente do *OSI-RM*, a abordagem desenvolvida pela comunidade da *Internet* tornou-se um padrão de fato, sendo utilizada em projetos de pesquisa e em ferramentas comerciais de gerenciamento.

Em decorrência do caráter mais pragmático da abordagem de gerenciamento da *IETF* e da possibilidade de aplicar parte dos conceitos estabelecidos no *OSI-RM*, este trabalho, a exemplo de outros com temática semelhante ([SiTa94], [CaMa94], [PiVe95], [SiMa95] e [PHH97]), baseou-se na utilização da abordagem conhecida como *Simple Network Management Protocol - SNMP* ou gerenciamento *TCP/IP*.

Definida a abordagem de gerenciamento e os aspectos do *OSI-RM* que deveriam ser tratados no desenvolvimento do trabalho, procurou-se identificar quais as lacunas existentes, que complementariam os esforços já realizados para o escopo de gerenciamento pretendido. Dentre as diferentes ações que poderiam ser conduzidas, buscou-se investigar a possibilidade de estabelecer um conjunto de atividades a serem utilizadas na especificação e desenvolvimento de ferramentas para o gerenciamento de aplicações em rede. Como aplicações em rede, entende-se todo programa que desempenha o papel de *client/server* ou ambos, e que são utilizados no oferecimento de serviços da *Internet*.

Visando delimitar um escopo para o trabalho, foram selecionados como temas para investigação, os requisitos de gerenciamento de contabilização e de segurança, levantados junto aos administradores e relacionados a implementações do *HyperText Transfer Protocol - HTTP*. Como forma de validar este trabalho, foram propostas e desenvolvidas seis atividades para o gerenciamento de aplicações em rede, de forma a complementar os modelos de gerenciamento padronizados, que tratam aspectos gerais, relacionados a todas as camadas, o que, de certa forma, caracteriza a necessidade de estabelecer-se um modelo específico para gerenciar, por exemplo, a camada de aplicação.

A impossibilidade de utilizar uma plataforma de gerenciamento proprietária, adicionada <sup>do</sup> interesse em desenvolver uma aplicação de gerenciamento com o uso de tecnologias abertas, proporcionou experimentos com o uso de um pacote de gerenciamento *SNMP* de domínio público, com a implementação de um banco de dados padrão *SQL (Structured Query Language)*, com a criação de aplicações de *CGI (Common Gateway Interface)* e com a formatação de interfaces com o uso de *HTML (HyperText Markup Language)* e suas extensões. Todo este conjunto de padrões possibilitou incorporar, ao escopo do trabalho, aspectos gerais de uma tendência atual para o gerenciamento de redes, denominada *Web-Based Network Management*.

Dessa forma, este trabalho, além de apresentar conceitos e terminologia de gerenciamento de redes, contém uma proposta para o gerenciamento de aplicações em rede, que tem como objetivo facilitar a análise e o desenvolvimento de ferramentas construídas com tecnologias abertas e destinadas ao controle e monitoramento de entidades *client/server* da *Internet*. Tais ferramentas poderão apoiar o gerenciamento da *Quality of Service - QoS* das aplicações disponibilizadas aos usuários da rede, pois, mesmo dispondo de soluções automatizadas, administradores de redes necessitam desenvolver aplicações que atendam às necessidades específicas de seus ambientes.

## **1.2 Organização do trabalho**

Além deste capítulo, o trabalho possui seis capítulos adicionais, contendo respectivamente o referencial teórico relativo ao Gerenciamento *OSI*, Gerenciamento *Internet*, Gerenciamento baseado em *Web*, proposta para o gerenciamento de aplicações em rede, aplicação da referida proposta, conclusões e trabalhos futuros.

No **capítulo 2**, são descritos os requisitos de gerenciamento definidos nos padrões da *International Organization for Standardization - ISO* para o gerenciamento de contabilização e de segurança, que foram considerados no escopo da proposta de gerenciamento efetuada.

No **capítulo 3**, é apresentada a abordagem de gerenciamento padronizada pela *IETF*, com a descrição detalhada dos elementos que constituem o modelo *SNMP*, como: Aplicação Gerente, Aplicação Agente e operações e mensagens do *SNMP*.

No **capítulo 4**, são tratados aspectos gerais do gerenciamento baseado em *Web*, e apresentadas algumas estratégias de migração, adotadas atualmente por desenvolvedores comerciais e organismos de padronização.

No **capítulo 5**, é apresentada a proposta para o gerenciamento de aplicações em rede, que compreende o desenvolvimento de seis atividades, as quais tratam da identificação e análise dos requisitos de usuário, análise das aplicações de redes e *Management Information Base - MIBs* existentes, e da especificação e implementação de ferramentas de gerenciamento.

No **capítulo 6**, um estudo de caso exemplifica a utilização da proposta para o gerenciamento de aplicações em rede que implementam o *HTTP*. A proposta é aplicada no desenvolvimento de um protótipo de sistema contendo os componentes da abordagem de gerenciamento da *IETF*.

Finalizando, no **capítulo 7**, apresentam-se algumas considerações gerais sobre o trabalho, as contribuições efetuadas e sugestões para trabalhos futuros. Como anexo, foram incluídas a *MIB* especificada em nível experimental e a descrição dos principais módulos do protótipo desenvolvido.

## Capítulo 2

### Gerenciamento *OSI*

A *ISO* [BRI93] dividiu as atividades de gerenciamento em cinco áreas funcionais: Gerenciamento de Configuração; gerenciamento de Falhas; gerenciamento de Desempenho; gerenciamento de Segurança e gerenciamento de Contabilização. Dentro de cada área funcional, foram desenvolvidos padrões denominados Funções de Gerenciamento.

Como neste trabalho apresenta-se uma proposta para o gerenciamento de aplicações em rede, escolheu-se como áreas funcionais, visando validar a proposta, o gerenciamento de contabilização e de segurança de implementações do tipo servidor, baseadas no *HTTP*. Dessa forma, faz-se necessário descrever os principais requisitos definidos nos padrões da *ISO*, destinados ao gerenciamento de contabilização e gerenciamento de segurança.

#### 2.1 Gerenciamento de Contabilização

O Gerenciamento de Contabilização [BRI93] preocupa-se com a manutenção e monitoração de quais recursos e de quanto desses recursos estão sendo utilizados. Essas informações fornecem dados estatísticos ou permitem a “bilhetagem” (faturamento).

### 2.1.1 Função de medida de uso

A Função de medida de uso estabelece os critérios para criação de mecanismos destinados à coleta, controle e registro de informações relacionadas à utilização de recursos e serviços, e à identificação dos usuários e/ou assinantes. As medidas de uso efetuadas poderão ser utilizadas para redimensionar os recursos, incentivar a utilização dos serviços em horários de menor pico, estabelecer e controlar cotas e associar tarifas para efeito de faturamento e cobrança.

Em redes cujos recursos e serviços são disponibilizados gratuitamente ou possuem critérios de arrecadação indiretos, a gerência de contabilização pode ser aplicada no monitoramento do uso da rede, em atividades como teletrabalho e ensino à distância. Dessa forma, ao invés de se associarem tarifas para efeito de faturamento e cobrança, podem-se associar níveis de pontuações visando determinar, por exemplo, o perfil das ações e a produtividade dos usuários.

O padrão *ISO10164-10* sugere a implementação de três classes de objetos gerenciados, destinados à coleta, ao controle e registro das medições sobre a utilização dos recursos:

*accounting meter control object:* os objetos desta classe são responsáveis por permitir o controle sobre o início, suspensão e retomada das medidas de contabilização, mantendo, ainda, informações sobre como devem ser realizadas estas medidas;

*accounting meter data object:* seus objetos realizam efetivamente a coleta de dados sobre a utilização de um recurso por um usuário ou assinante, fornecendo informações, como: o número de unidades

utilizadas; a unidade empregada e o tipo de tarifa associada;

*accounting record*: os objetos desta classe são necessários para compor um histórico dos dados coletados pelos objetos de dados de medida.

A atual recomendação para Função de medida de uso, distribui a atividade de contabilização em três processos distintos [ISO94]: processo de medida de uso, processo de tarifação e processo de fatura. A FIG. 2.1 apresenta o relacionamento entre os processos que formam a atividade de contabilização.

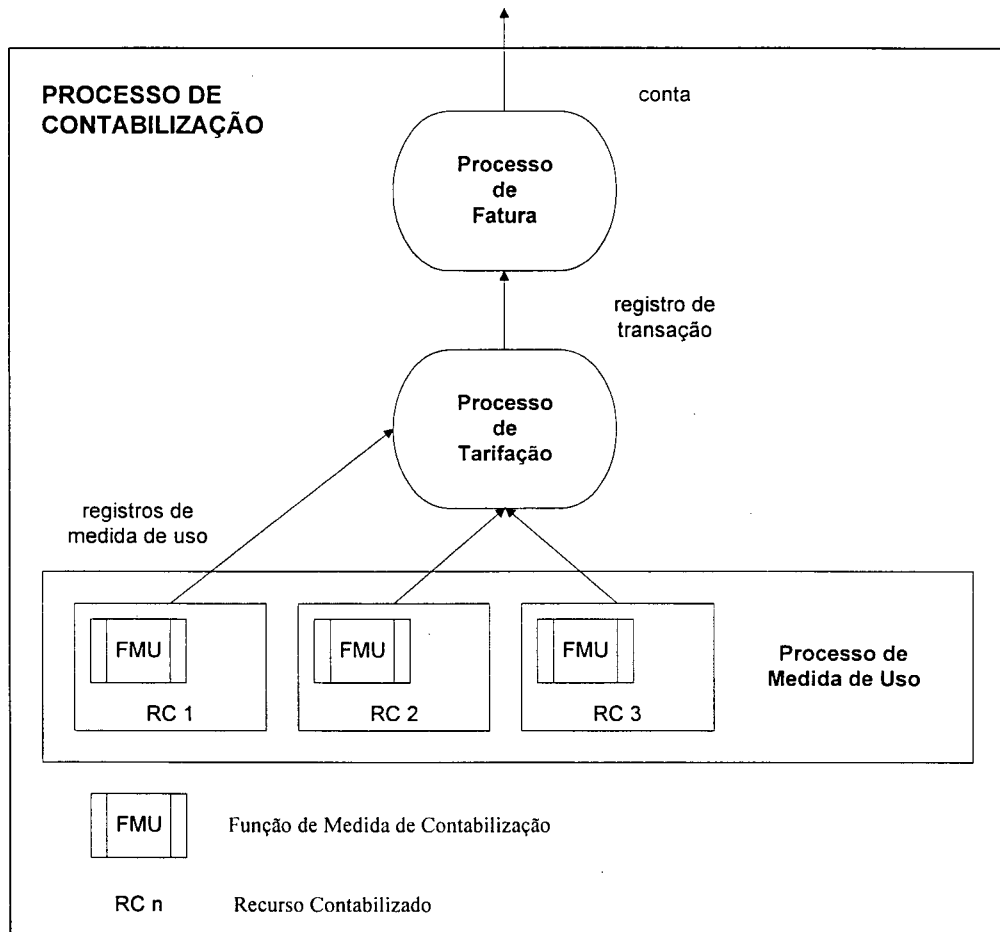


FIGURA 2.1: Relacionamento entre os processos que formam a atividade de contabilização  
[Fonte: ISO10164-10]

Os processos de Fatura e Tarifação fogem ao escopo do padrão internacional *ISO10164-10*, como também não estão sendo tratados neste trabalho. Por outro lado, informações de interesse foram extraídas da especificação, relativas ao processo de Medida de Uso, o qual é descrito em detalhes no referido documento.

O processo de Medida de Uso consiste na coleta de dados referente à utilização específica de cada recurso. Os dados coletados deverão ser armazenados em registros de medida de uso, contendo atributos que identifiquem o usuário, o tipo de serviço fornecido, a quantidade consumida e demais informações consideradas relevantes para os propósitos de gerenciamento de contabilização.

A especificação de mecanismos para a gerência de contabilização poderá utilizar dois grupos de atributos:

- atributos de controle de medida de uso e
- atributos de dados de medida de uso.

### **2.1.2 Atributos de controle de medida de uso**

Os atributos de controle de medida de uso compreendem:

unidades de utilização: unidade de medida da utilização de um recurso em particular;

disparo de registro: especifica os eventos que causam a atualização dos dados de medida de uso;

disparo de relatório: especifica os eventos que causam a emissão de notificação sobre informações de uso;



nome do recurso: especifica de maneira única a identificação de cada recurso, cuja utilização pode ser medida.

### **2.1.3 Atributos de dados de medida de uso**

Dentre os atributos de dados de medida de uso, encontram-se:

identificador do solicitador: identifica o usuário do serviço prestado;

identificador do assinante: identifica o assinante que, em nome do usuário, solicitador, possui um contrato com o prestador do recurso sujeito à medição de uso;

informação de medida: fornece dados de contabilização relativos à utilização específica de um recurso, como: unidade de medida, taxa de utilização e tarifa de utilização;

instante de início de utilização: indica o instante de tempo no qual foi iniciada a medição de uso;

instante de medida de utilização: indica o instante de tempo corrente em que existe uma medição de uso em curso, ou o instante de tempo em que a medição de uso da utilização do recurso correspondente foi suspensa;

estado da medição de uso: indica o estado da medição de uso, que pode ser: executando (medição em curso) ou suspenso (medição suspensa);

nome do recurso: identifica o recurso cuja utilização está sendo medida.

## 2.2 Gerenciamento de Segurança

O Gerenciamento de Segurança [BRI93] aborda os aspectos de segurança essenciais para operar uma rede corretamente e proteger os objetos gerenciados. O sistema de gerenciamento deve providenciar alarmes para uma Aplicação Gerente da rede quando eventos de segurança forem detectados.

Em virtude do escopo definido neste trabalho para o gerenciamento de segurança, o qual é caracterizado pelo monitoramento do acesso a repositórios de informações em formato *Web*, somente serão abordados aspectos relacionados a duas das três funções básicas que servem de suporte às atividades de gerenciamento de segurança:

- função de relatório de alarme de segurança e
- função de registro para auditoria de segurança.

### 2.2.1 Função de relatório de alarme de segurança

Esta função prevê o envio de notificações de eventos relativos à segurança. As notificações deverão conter atributos, como: identificação de operações errôneas, informações sobre ataques à segurança e gravidade percebida em relação às operações errôneas. O modelo proposto para esta função segue o modelo genérico definido para a função de relatório de evento. A FIG. 2.2 apresenta o modelo para Função de relatório de alarme de segurança.

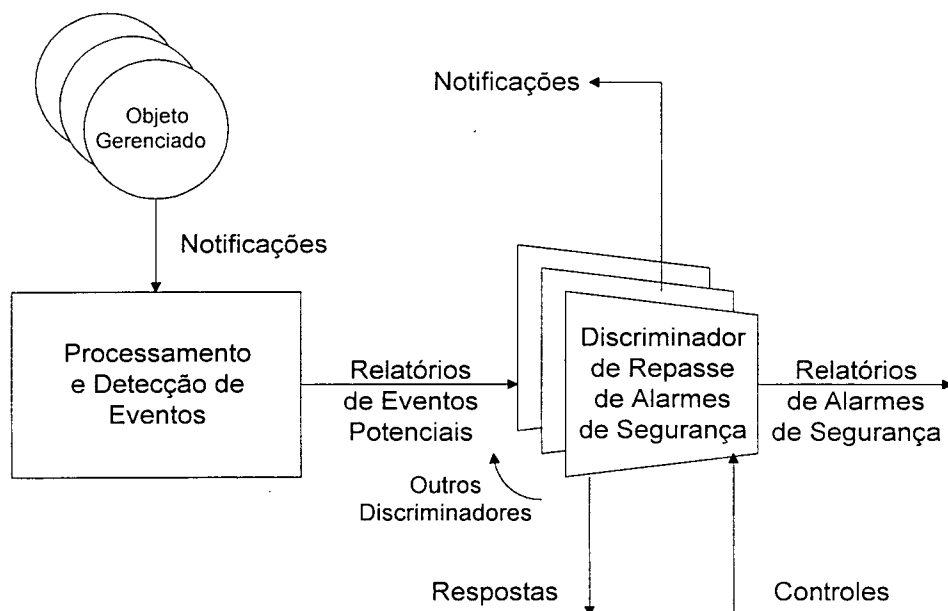


FIGURA 2.2: Modelo para Função de relatório de alarme de segurança  
 [Fonte: BRI93]

No modelo proposto, notificações locais provenientes dos objetos gerenciados são utilizadas na preparação de relatórios de eventos potenciais, que são fornecidos a todos os discriminadores de repasse de eventos presentes no sistema local. Um discriminador de repasse seleciona, em períodos determinados, os eventos relativos à segurança, para que sejam enviados na forma de alarmes de segurança a um usuário de gerenciamento.

Os relatórios de alarmes de segurança contêm um conjunto padrão de informações que permite definir o tipo, a causa, a severidade dos alarmes e o gerador do alarme, além de uma descrição textual do alarme e quaisquer outros dados não-padronizados relativos ao alarme.

No padrão *ISO10164-7* [BRI93], foram definidos alguns tipos de alarmes de segurança e causas prováveis dos alarmes. Dentre os tipos e causas de alarmes relacionadas, consideram-se relevantes para o escopo deste trabalho:

## Tipos de Alarmes

- violação operacional: indicação de que não foi possível fornecer o serviço requerido, devido à invocação incorreta do serviço;
- violação de serviço: indicação de que um serviço de segurança detectou um atentado contra a segurança do sistema;

## Causas de Alarme

- erro de procedimento: um procedimento incorreto foi usado para invocar um serviço;
- tentativa de acesso não-autorizado: um mecanismo de controle de acesso detectou uma tentativa ilegal de acesso a um objeto;

### **2.2.2 Função de registro para auditoria de segurança**

O armazenamento de todos os eventos potenciais relativos à segurança, constitui o principal objetivo desta função. Os eventos selecionados são dispostos em um *Log* de auditoria, para que, por meio de análise, sejam detectados desvios com relação às normas de política de segurança e descobertos pontos vulneráveis.

O *Log* de auditoria de segurança poderá possuir dois tipos de registro. O primeiro, denominado relatório de serviço, representa a indicação de que o registro é associado a um evento relacionado ao fornecimento, negação ou recuperação de um determinado serviço. O segundo tipo de registro é utilizado na indicação do registro que contém informações de natureza estatística.

O modelo previsto para esta função permite que a coleção de registros de auditoria de segurança seja armazenada no sistema local ou num outro sistema remoto. Um construtor de discriminador deverá ser utilizado na transferência dos relatórios de eventos para os *Logs* de auditoria de segurança. O modelo definido para a **Função de controle de Log** deverá ser utilizado para controlar os *Logs* de auditoria.

Conforme definido na Função de controle de *Log* [BRI93], o *Log* é um repositório de registros que contém informações as quais devem ser preservadas. Um *Log* armazena os registros por ordem de chegada e os identificadores de registros são atribuídos em seqüência numérica.

# Capítulo 3

## Gerenciamento *INTERNET*

O Gerenciamento *Internet*, também chamado de abordagem *SNMP* ou modelo de gerenciamento *SNMP*, é padronizado pela *IETF*, uma organização aberta que define os padrões para a *Internet*. O modelo *SNMP* possui uma abordagem genérica, podendo ser utilizado para gerenciar diferentes tipos de sistemas.

### 3.1 O modelo de gerenciamento *SNMP*

No gerenciamento de uma rede, o modelo *SNMP* consiste dos seguintes elementos [PeMc97] (FIG. 3.1):

- No mínimo uma estação de gerenciamento, contendo uma ou mais entidades *SNMP* chamadas de Gerente;
- Um ou mais *nodos* gerenciados, cada um contendo uma entidade *SNMP* chamada de Agente;
- Opcionalmente, entidades *SNMP* chamadas de entidades com dupla função, capazes de desempenhar o papel de Agente-Gerente;

- Informações de gerenciamento em cada *nodo* gerenciado, as quais descrevem a configuração, o estado, as estatísticas e as ações que controlam os *nodos* gerenciados;
- Um protocolo para comunicação de gerenciamento utilizado pelos Gerentes e Agentes durante a troca de mensagens de gerenciamento.

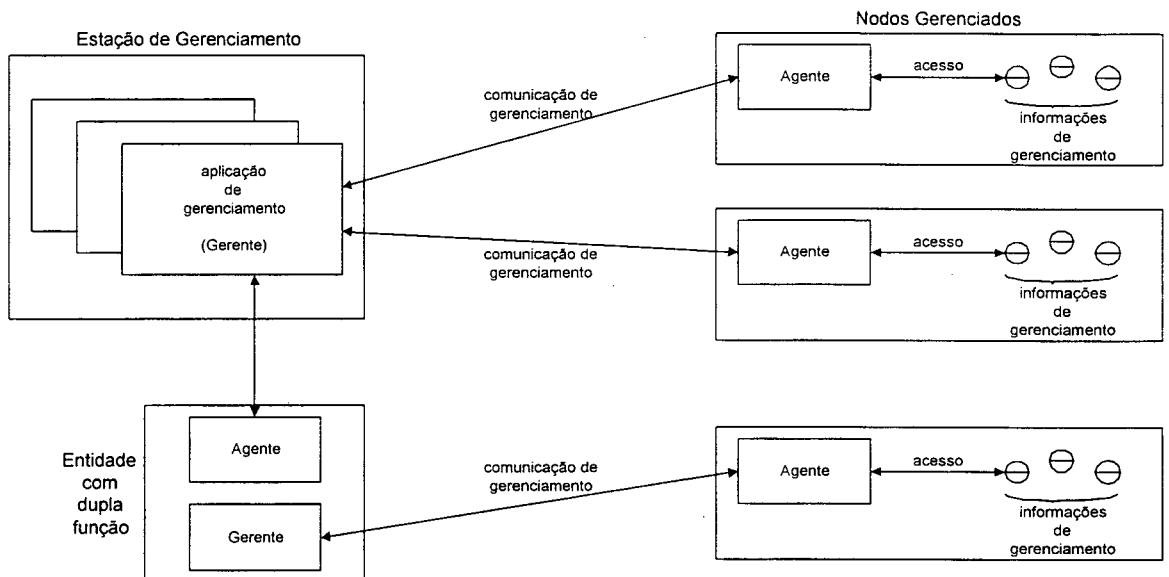


FIGURA 3.1: Modelo *SNMP* para o gerenciamento de rede  
Fonte: [PeMc97]

O modelo *SNMP* é definido em uma coleção de documentos, denominados *Request For Comments – RFC's*. As *RFC's* definem uma estrutura de gerenciamento que compreende: um protocolo de gerenciamento, a definição de informações de gerenciamento e eventos, um núcleo de informações de gerenciamento e eventos, e um mecanismo para gerenciar o uso do protocolo, incluindo aspectos de segurança. A TAB. 3.1 relaciona os documentos que definem a estrutura do modelo *SNMP* nas versões 1 e 2. A próxima geração do protocolo (versão 3) está sendo definida pelo *SNMPv3 Working Group* do *IETF* [IETFb], com previsão de término dos trabalhos para o mês de abril de 1998.

TABELA 3.1: Documentos que definem a estrutura do modelo *SNMPv1* e *v2*

ESCOPO	DESCRIÇÃO	PUBLICAÇÃO	RFC
Protocolo de Gerenciamento	<i>SNMPv1 Protocol</i>	Maio 1990	1157
	<i>SNMPv2 Protocol Operations</i>	Janeiro 1996	1905
	<i>SNMPv2 Transport Mappings</i>	Janeiro 1996	1906
Estrutura de Informações de Gerenciamento	<i>SMIv1 with typos fixed</i>	Maio 1990	1155
	<i>SMIv1 Concise MIB format</i>	Março 1991	1212
	<i>SMIv1 Traps formats</i>	Março 1991	1215
	<i>SMIv2</i>	Janeiro 1996	1902
	<i>SMIv2 Textual Conventions</i>	Janeiro 1996	1903
Núcleo de Informações de Gerenciamento	<i>MIB II</i>	Março 1991	1213
	<i>SNMPv2 Core</i>	Janeiro 1996	1907

Para o desenvolvimento de aplicações de gerenciamento [Har97], o modelo *SNMP* poderá ser visto como composto por dois tipos de aplicações denominadas Aplicação Gerente e Aplicação Agente, as quais se comunicam via interconexão e protocolos de rede, conforme é apresentado na FIG. 3.2.

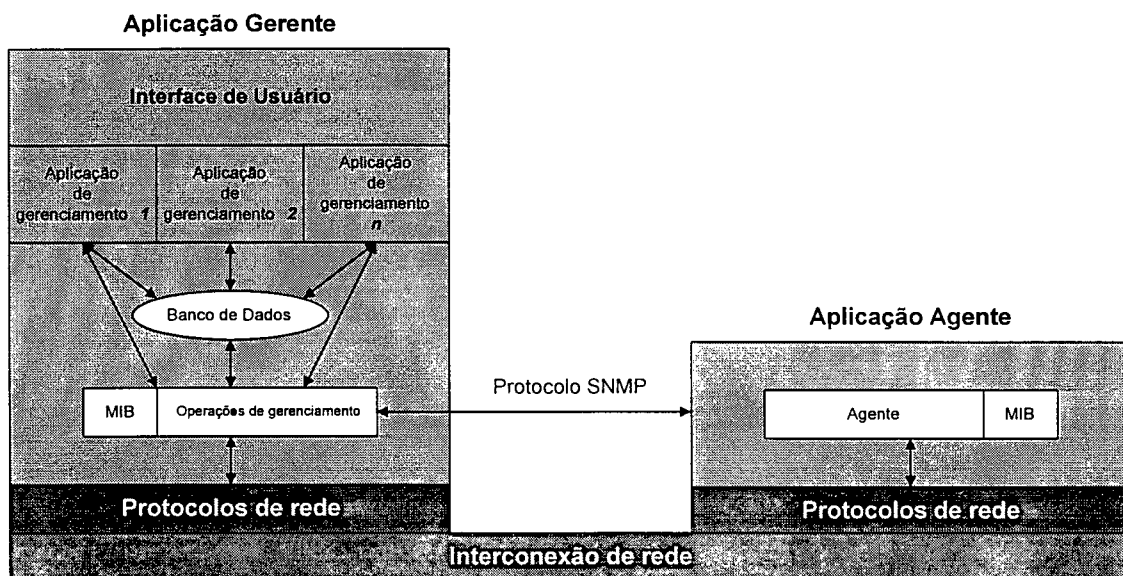


FIGURA 3.2: Tipos de aplicações de gerenciamento e forma de comunicação

No modelo *SNMP*, a interconexão de rede consiste da coleção de uma ou mais redes que usam protocolos comuns e são conectadas por *gateways*. Qualquer dois pontos finais podem se comunicar através da implementação de um esquema de endereçamento global e do uso de protocolos padronizados como os que formam a pilha



*TCP/IP* (*Transmission Control Protocol/Internet Protocol*). Nos protocolos existem regras, as quais permitem que a comunicação entre redes seja possível.

A comunicação de redes que utilizam a pilha *TCP/IP* é realizada por diferentes protocolos, os quais operam ao longo das camadas que formam a arquitetura. Um *host*, por exemplo, deverá possuir, a implementação de ao menos um protocolo em cada uma das camadas (camada de acesso à rede, camada de rede, camada de transporte e camada de aplicação (superior)).

### 3.2 Aplicação Gerente

O modelo *SNMP* apresenta a Aplicação Gerente como um dispositivo de rede que usa determinados protocolos das camadas de transporte, de rede, de acesso à rede e de aplicação, para a comunicação com a entidade de rede gerenciada. Conforme apresentado na FIG. 3.3, a Aplicação Gerente poderá ser constituída de cinco componentes principais:

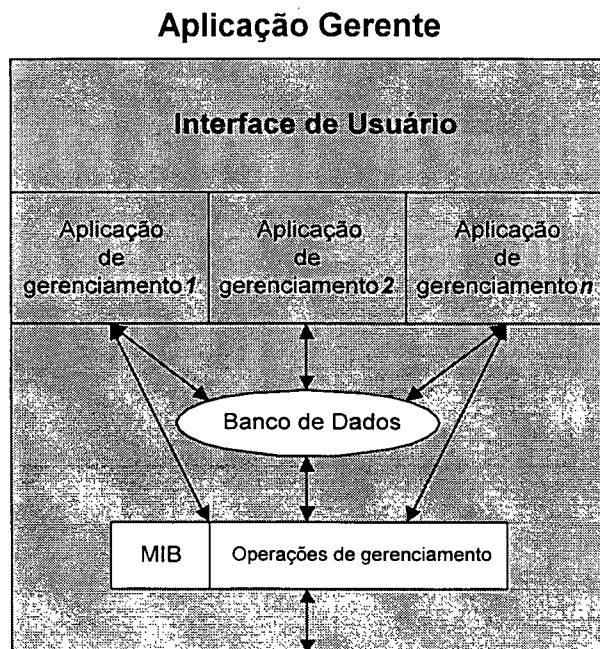


FIGURA 3.3: Componentes de uma Aplicação Gerente

- Operações de gerenciamento
- *Management Information Base – MIB*
- Banco de Dados
- Aplicações de gerenciamento
- Interface de usuário

### 3.2.1 Operações de gerenciamento

As operações de gerenciamento são componentes da Aplicação Gerente que controlam e monitoram os Agentes pertencentes à comunidade de um determinado domínio de gerenciamento. As operações de gerenciamento podem ler e escrever em variáveis da *MIB* de cada Agente para gerenciar o dispositivo de rede. As operações de gerenciamento podem, também, armazenar informações de gerenciamento recuperadas junto aos Agentes em uma *MIB* própria e/ou em um Banco de Dados.

Uma operação de gerenciamento é uma entidade de *software* que implementa as regras estabelecidas no protocolo *SNMP*. A implementação deverá codificar e decodificar mensagens construídas em unidades de dados de protocolo, baseadas no subconjunto da *Abstract Syntax Notation One – ASN.1*, para utilização em formatos internos. Para atender as regras da versão 1 do *SNMP*, as operações de gerenciamento deverão implementar processos para execução das requisições: *GetRequest*, *SetRequest* e *GetNextRequest – PDUs (Protocol Data Units)*. Para atender as regras da versão 2 do *SNMP*, as operações de gerenciamento deverão implementar processos adicionais para a requisição *GetBulk-PDU*, para o tratamento das alterações da mensagem *Response-PDU*, de um novo formato de *Trap*, e para a possibilidade de mensagens do tipo *InformRequest-PDU*.

A comunicação com a *MIB*, visando reconhecer o próprio conjunto de objetos gerenciados, que podem ser controlados e monitorados, e uma interface para as funções de envio e recebimento de mensagens *SNMP* via camada de transporte (protocolo *UDP* e outros utilizados), são procedimentos que deverão ser incluídos nas operações de

gerenciamento. As operações de gerenciamento também deverão fornecer os serviços necessários para qualquer requisição de aplicações de gerenciamento que podem compor um Sistema de Gerenciamento de Rede (chamados de Plataformas de Gerenciamento, quando são utilizados produtos proprietários). A *Application Programming Interface-API* poderá ser proprietária, implementada por um fabricante de operações de gerenciamento, ou aberta, como, por exemplo, a que foi desenvolvida para Aplicações Gerente baseadas em ambiente Windows, denominada *WinSNMP* [Nat95], ou ainda a *API*, desenvolvida por [RoMc95] para *X Window System*.

As operações de gerenciamento podem ser encaradas como mecanismos de comunicação *SNMP*, utilizados na troca de mensagens entre Aplicações Gerente e Agente. Esses mecanismos estão disponíveis em uma implementação do *SNMP*, denominada *CMU-SNMP* [Dav96], e portada para o sistema operacional *Linux* [ScSc97]. As operações de gerenciamento estão implementadas no pacote de *software* desenvolvido na *Carnegie Mellon University-CMU*, na forma de utilitários (operações de gerenciamento) que processam as requisições *Get*, *GetNext*, *Walk*, *Set* e *Trapd*.

### **3.2.2 Management Information Base – MIB**

A *MIB* da Aplicação Gerente, quando existir, conterà a “relação mestre” para as *MIBs* de todos os Agentes pertencentes à comunidade de gerenciamento. Se uma operação de gerenciamento deseja controlar as variáveis da *MIB* do Agente, ela deverá conhecer ou ser habilitada para descobrir as variáveis existentes. Muitas vezes as operações de gerenciamento mantêm informações adicionais de gerenciamento em um Banco de Dados.

### **3.2.3 Banco de Dados**

O Banco de Dados é constituído por um conjunto de variáveis de interesse, que compreendem todos os nomes, configurações, topologia e dados de auditoria dos

dispositivos gerenciados. Sua implementação poderá ser efetuada com o uso de um programa servidor que implemente um banco de dados do tipo relacional e que adote como mecanismo de iteração o padrão *SQL*.

### 3.2.4 Aplicações de gerenciamento

As aplicações de gerenciamento são programas que transformam os dados *SNMP* em informações usadas pelos usuários da Aplicação Gerente (administradores de rede, usuários dos serviços de rede, etc). As aplicações de gerenciamento compreendem uma variedade de programas que *poll* os Agentes, via operações de gerenciamento (*Get / Set*), manipulam e formatam as mensagens de *trap* ou acessam o Banco de Dados. As aplicações de gerenciamento auxiliam no processamento e análise dos dados obtidos (recuperados) junto às Aplicações Agentes, via operações de gerenciamento, ou junto ao Banco de Dados.

As aplicações de gerenciamento constituem uma recente área no gerenciamento de rede, com a abordagem do modelo *SNMP*. As aplicações de gerenciamento obtêm dados destinados a apoiarem os administradores e usuários dos serviços de rede na tomada de decisões inteligentes, fornecendo gerenciamento pró-ativo e estendendo as funcionalidades das plataformas de gerenciamento em muitas direções.

A primeira geração de aplicações de gerenciamento incluiu a geração de relatório, *trouble tickets* e alarmes audíveis. Uma segunda geração de aplicações de gerenciamento, muitas vezes, inclui sistemas especialistas (*knowledge-based*) que podem alertar o pessoal de rede quando sérios problemas ocorrem, antecipando possíveis falhas em várias partes da rede gerenciada e usando eurísticas para apoiar o planejamento, a operação, e fornecer dados valiosos que podem ser usados para o conjunto de informações que será apresentado na Interface de Usuário.

Um componente chave das aplicações de gerenciamento é o uso de uma *API* padronizada. Isso permite que aplicações de múltiplos fornecedores sejam incluídas em

uma única plataforma de gerenciamento, visando oferecer uma *mix* de aplicações para uma determinada organização. A interface entre as aplicações de gerenciamento e as operações de gerenciamento deverá ser bem definida, aberta e baseada em padrões, sempre que possível.

### 3.2.5 Interface de usuário

A intervenção do usuário com a Aplicação Gerente é realizada via uma Interface de Usuário-IU. A IU possibilita ao operador entrar com comandos de gerenciamento e visualizar as respostas solicitadas ou não solicitadas (*traps*), enviadas pelas aplicações de gerenciamento, processadas junto à Base de Dados ou junto à Aplicação Agente, via operações de gerenciamento.

Muitas Aplicações Gerente oferecem uma *Graphic User Interface—GUI*, para apresentar estatísticas de desempenho, resumos de contabilização, relatório de falha, inventários de configuração, formulários para criação de *queries* e mapas da topologia da rede gerenciada. Interfaces gráficas de usuário atualmente são baseadas nos sistemas operacionais da *Microsoft (W95 e NT)*, *IBM OS/2* e uma variedade de padrões disponíveis com os sistemas *Xwindows*, tais como *MOTIF* e *Open Look*. A interface de usuário não é especificada no modelo *SNMP*, devendo ser considerada em nível de implementação.

O uso de *Web Browser* como mecanismo de suporte à interface de usuário, começa a ser empregado tanto em produtos comerciais como em ferramentas de domínio público. Os atuais recursos de *Web* poderão proporcionar a construção de interfaces ricas em componentes e de fácil utilização.

Devido à natureza de intensos recursos dos componentes, a Aplicação Gerente é usualmente implementada em uma *Workstation* rodando sistema operacional multitarefa, tal como *Unix* ou *Windows NT*. Muitas vezes o dispositivo de rede (*Workstation*) destinado a abrigar a Aplicação Gerente deverá disponibilizar uma grande quantidade

de memória *RAM*, um considerável espaço em disco, outros mecanismos de armazenamento secundário e dispositivos de *backup*. Esses recursos poderão ser escaláveis.

### 3.3 Aplicação Agente

O modelo *SNMP* apresenta a Aplicação Agente como uma entidade processando junto a um dispositivo de rede. A Aplicação Agente também utiliza os protocolos de rede e o protocolo *SNMP* para se comunicar via interconexão de rede com uma Aplicação Gerente. A Aplicação Agente consiste de dois componentes chaves:

- O Agente
- A *MIB* do Agente

#### 3.3.1 O Agente

Todo sistema diretamente gerenciado por *SNMP* deverá conter uma entidade chamada Agente, processando em modo *background*. O Agente é um tipo de programa que recebe requisições da Aplicação Gerente, pertencente a uma determinada comunidade de gerenciamento, verifica se as requisições são válidas e envia as respostas apropriadas. O Agente pode também ser configurado para enviar mensagens de *trap* na forma de relatórios assíncronos para eventos pré-definidos. O Agente utiliza rotinas de instrumentação que manipulam estruturas de dados locais para recuperar e configurar os vários objetos (variáveis) da *MIB*.

Um Agente necessita de um mecanismo de transporte baseado em datagramas para a troca de mensagens com entidades do tipo Gerente. Quando se utiliza o mesmo mecanismo e/ou rede para operações de gerenciamento e outras operações de um sistema, o gerenciamento é chamado de *in-band*. O caminho utilizado para a troca de mensagens é chamado de canal de gerenciamento *in-band*. Utilizando um mecanismo

e/ou rede separadas, somente para operações de gerenciamento, o gerenciamento é chamado de *out-of-band*. Neste caso, o caminho utilizado para a troca de mensagens de gerenciamento é chamado de canal de gerenciamento *out-of-band*.

O modelo *SNMP* geralmente impõe uma leve carga no mecanismo de transporte, e, desse modo, uma abordagem de gerenciamento *in-band* mostra-se eficiente e fornece alto desempenho. Entretanto, se o caminho de transporte entre uma Aplicação Agente e uma Aplicação Gerente não está operacional, será impossível diagnosticar e recuperar um problema sem a intervenção externa (tal como a de um administrador de rede). Com o uso de um canal de gerenciamento do tipo *out-of-band*, uma Aplicação Gerente deverá comunicar-se com uma Aplicação Agente para diagnosticar e, possivelmente, resolver o problema sem a necessidade de intervenção externa.

O principal fator que dificulta a adoção de uma abordagem de gerenciamento do tipo *out-of-band* é o custo de implementação. Entretanto, em ambientes onde existem necessidades críticas de gerenciamento, um canal de comunicação de baixa velocidade (*dial-up*, por exemplo) poderá ser utilizado.

Um Agente poderá suportar um ou mais protocolos de transporte. O *User Datagram Protocol - UDP* é o protocolo da pilha *TCP/IP* escolhido. O protocolo de transporte *UDP* foi selecionado, por aumentar a possibilidade de ser um protocolo compatível e por possibilitar a interoperação entre qualquer Gerente e Agente.

Um Agente *SNMP* deverá ser habilitado a acessar informações de gerenciamento em um subsistema, para que este seja gerenciado. Um Agente (via rotinas e métodos) deverá ser capaz de acessar informações de gerenciamento do sistema e ser habilitado para receber notificações de eventos que ocorreram no sistema. Cada subsistema gerenciado deverá incluir um mecanismo de acesso para instrumentação de gerenciamento para o Agente contido no subsistema (FIG. 3.4).

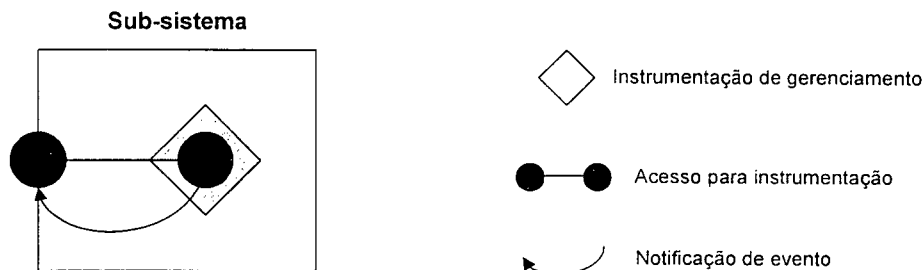


FIGURA 3.4: Adições de gerenciamento *SNMP* em um subsistema

A instrumentação é efetuada por meio da utilização de dispositivos extras de *hardware* ou de *software* que deverão ser adicionados ao subsistema para medir e reter o valor ou o procedimento correspondente a uma alteração de valor. Muitos subsistemas podem ser capazes de executar estas funções ou fornecer serviços externos destas instrumentações. Também, um subsistema poderá já possuir alguma instrumentação, mas isso poderia não atender as definições encontradas em um módulo de *MIB*. A interface entre o Agente e as informações de gerenciamento também foge ao escopo do modelo *SNMP*.

Existem diferentes abordagens que poderão ser utilizadas na construção de um Agente. Contudo, em todas as abordagens, um Agente poderá consistir das seguintes áreas funcionais [PeMc97]:

- Acesso para uma ou mais pilhas de transporte.
- Uma máquina de protocolo (que inclui mecanismos de segurança).
- Uma tabela de mapeamento para rotinas e métodos.

### 3.3.1.1 Pilha de transporte

Um Agente *SNMP* utiliza uma pilha de transporte para receber e transmitir respostas. Uma entidade do tipo Agente utiliza, geralmente, um conhecido seletor de transporte para receber mensagens. Por exemplo, no conjunto de protocolos da *Internet*, a porta *UDP* 161 é usada como seletor de transporte, juntamente com o endereço *IP* do



sistema que contém o Agente para formar o endereço de transporte para as mensagens de gerenciamento. A interface entre o Agente e a pilha de transporte está fora do escopo do modelo *SNMP*.

Existem interfaces bem definidas, tal como do tipo *sockets*, que podem ser disponíveis para serem usadas pelo Agente. Por outro lado, o uso de um Agente para transmitir e receber mensagens, transforma a pilha de transporte em uma das áreas primárias de interesse de gerenciamento de rede em um sistema gerenciado. A pilha de transporte tem que incluir instrumentação e interfaces de acesso para rotinas e métodos a todas as camadas, para que informações de gerenciamento sejam acessadas pelo Agente. Isto nem sempre é possível, principalmente quando os fornecedores de pilhas de transporte não implementam instrumentação ou não fornecem acesso ao código fonte.

### **3.3.1.2 Máquina de protocolo**

O componente de um Agente que recebe uma mensagem *SNMP*, decodifica, implementa a verificação dos mecanismos de segurança, efetua o mapeamento das informações de gerenciamento, chama rotinas e métodos, codifica e envia mensagens de respostas, é chamado de máquina de protocolo. A máquina de protocolo também recebe notificações de eventos de dentro do sistema gerenciado e gera relatórios de eventos. Esses mecanismos estão presentes em produtos comerciais e em códigos disponíveis para a livre utilização, não existindo a necessidade de desenvolver esse tipo de código a partir de um ponto zero.

### **3.3.1.3 Tabela de mapeamento**

A tabela de mapeamento contém o relacionamento para as rotinas e métodos baseadas na identificação das informações de gerenciamento e um seletor. As mensagens *SNMP* recebidas ou geradas por um Agente, contêm a identificação da

informação de gerenciamento (ou aproximação da identificação da informação de gerenciamento, quando se tratar de operações do tipo *GetNext* e *GetBulk*).

A identificação de instâncias de informações de gerenciamento consiste de duas partes. A primeira é o tipo de informação de gerenciamento e a segunda especifica uma instância do tipo. As mensagens *SNMP* também possuem um seletor que é usado para qualificar a identificação na mensagem. Na versão 1 do *SNMP*, o seletor é o campo *string* que contém o nome da comunidade<sup>1</sup>. Na versão 2 do *SNMP*, o seletor é o campo *context* da mensagem. Para cada identificação de informação de gerenciamento na mensagem, um Agente deverá usar o seletor da mensagem em ambas as partes da identificação, para acessar as informações de gerenciamento. Uma identificação e um seletor também devem ser especificados na requisição via interface local, se isso for suportado pelo Agente.

Os detalhes desta tabela (e interface local) estão fora do escopo da estrutura de gerenciamento do modelo *SNMP*. Entretanto, é importante entender este aspecto relacionado a um Agente, pois poderá afetar a definição de módulos da *MIB*. Algumas implementações antigas de Agentes requerem que esta tabela seja projetada sem o uso de mecanismos automatizados (de forma manual). Implementações antigas têm estas tabelas construídas nos processos de compilação e *link*-edição do Agente. Algumas implementações contemporâneas de Agentes permitem que novas entradas possam ser criadas ou removidas durante a execução do Agente. A adição de novos mapeamentos é chamada de *registration*. A remoção de mapeamentos é chamado de *deregistration*.

### 3.3.2 Diagrama de um Agente

Diferentes abordagens podem ser usadas para decompor as áreas funcionais de um Agente. Algumas abordagens podem dimensioná-lo na forma de um pacote com a adição de outros componentes, tais como rotinas, métodos e mecanismos de

---

<sup>1</sup> A *string* de comunidade é usada atualmente como seletor e autenticador.

instrumentação. A FIG. 3.5 apresenta um diagrama contendo uma possível decomposição de um Agente, onde cada área funcional é indicada separadamente.

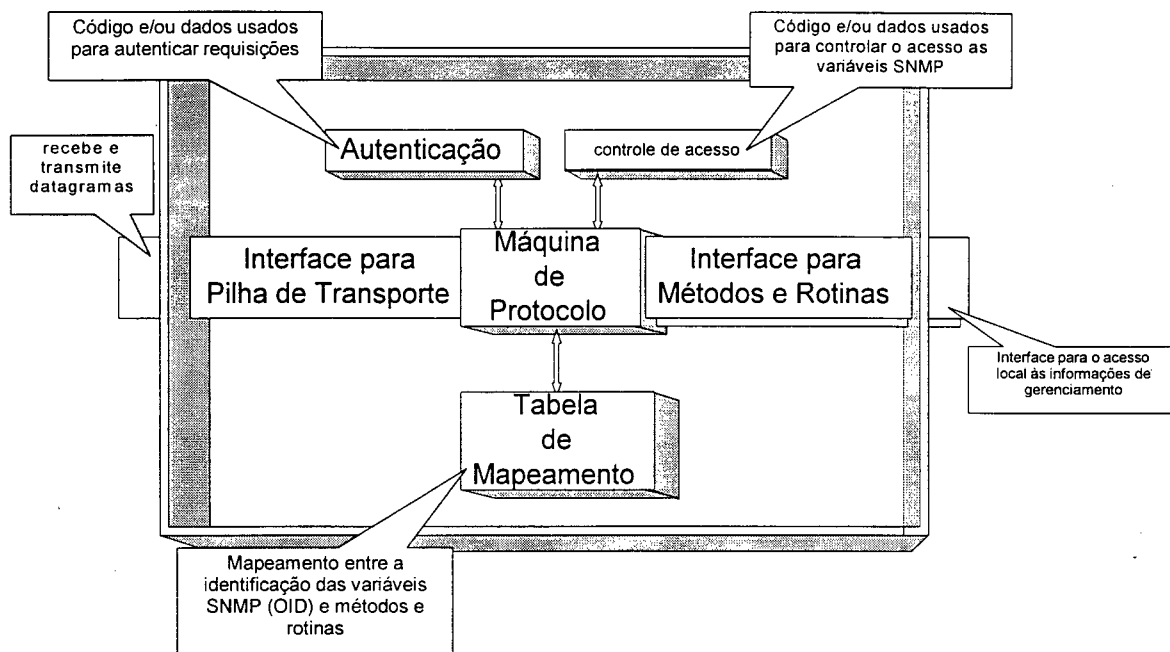


FIGURA 3.5: Áreas funcionais de um Agente *SNMP*

Fonte: [PeMc97]

Com base no diagrama apresentado, pode-se notar que o Agente contém somente uma *engine* de protocolo (incluindo aspectos de segurança), uma *dispatch table*, acesso para uma interface de transporte, acesso para métodos e rotinas (incluindo *registration* e *deregistration*) e uma interface local para acessar informações de gerenciamento via Agente.

Nas implementações atuais, as áreas funcionais de um Agente e as adições para atendimento dos requisitos necessários ao gerenciamento baseado no modelo *SNMP* são claramente separadas. Os fatores que afetam a escolha são os seguintes: o responsável por adicionar gerenciamento; o total de software no sistema; o uso de um simples processador ou de múltiplos processadores no sistema e configuração do sistema de forma estática ou dinâmica.

Os dois primeiros fatores relacionam-se ao modo como o *software* para o gerenciamento do sistema é desenvolvido. Em um sistema pequeno, dedicado e fechado, uma única pessoa ou grupo poderá ser responsável por adicionar capacidade de gerenciamento *SNMP*. Em grandes projetos (incluindo a segunda ou terceira geração de um pequeno Agente), em sistemas multiprocessados e sistemas configurados dinamicamente, as rotinas e métodos tendem a ser implementados separadamente. Isso simplifica o desenvolvimento de um Agente, permitindo sua modificação independente do subsistema gerenciado e possibilitando que novas informações de gerenciamento sejam adicionadas em um subsistema já existente ou novo, sem a necessidade de alterar o Agente. O projeto de um Agente genérico, juntamente com a descrição da arquitetura e interfaces dos serviços implementados são encontrados em [PHL95]. Outra abordagem, apresentada em [Har97], define que um Agente pode ser representado como um conjunto de 6 funções (FIG. 3.6).

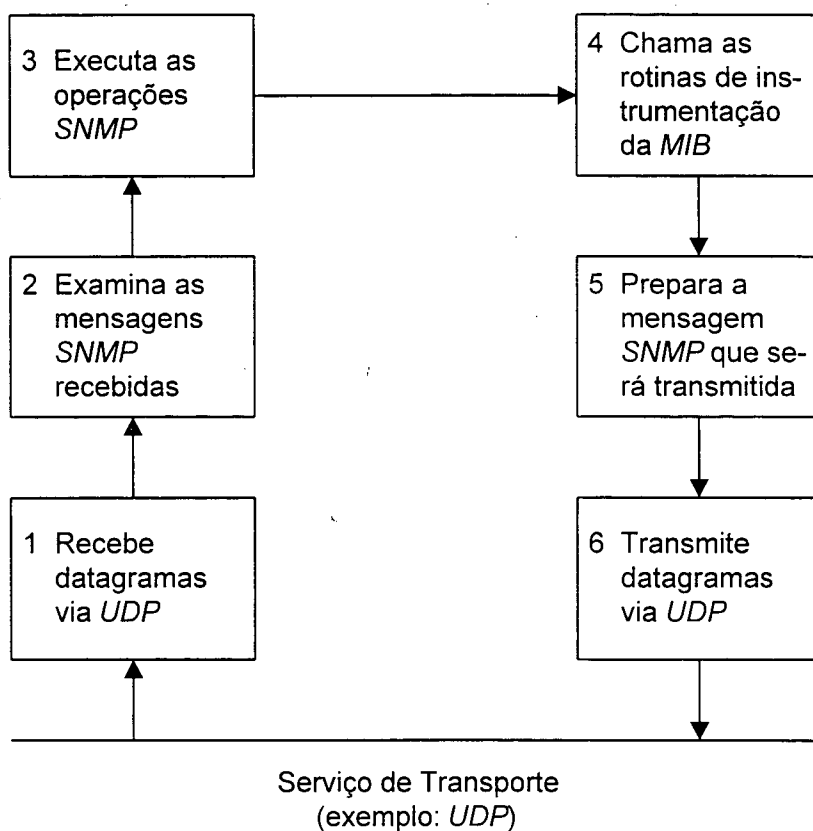


FIGURA 3.6: Funções desempenhadas por um Agente *SNMP*

### 3.3.3 Estrutura do Agente

Existem duas abordagens predominantes que têm sido usadas para implementar Agentes. Em sistemas pequenos, dedicados e fechados, a abordagem monolítica é tipicamente utilizada. Nesta abordagem, o Agente, rotinas e métodos, e grande parte das instrumentações são combinadas em conjunto. Esta é uma boa escolha quando todo o subsistema de *software* é conhecido e integrado ao Agente em tempo de compilação<sup>2</sup> e/ou *link*-edição<sup>2</sup>. Em sistemas de propósitos gerais e abertos, a abordagem extensível é tipicamente utilizada. Esta abordagem sustenta a separação em Agente, rotinas e métodos, até o tempo de carga<sup>2</sup>, tempo de inicialização<sup>2</sup> ou tempo de execução<sup>2</sup>.

### 3.3.4 MIB do Agente

A *Management Information Base* do Agente é uma coleção de variáveis de interesse. Os grupos da *MIB* que compreendem um particular módulo de informações de gerenciamento controlado pelo Agente, são dependentes das funcionalidades do dispositivo e de quais recursos ou serviços o Agente deverá gerenciar.

Muitas pessoas, que estudam pela primeira vez o modelo *SNMP*, criam a convicção errada de que um sistema gerenciado contém um banco de dados com informações de gerenciamento, e elas acreditam que este banco de dados é a *MIB* do sistema gerenciado. A verdade (FIG. 3.7) é que nenhum banco de dados é necessário ou existe. Muitas informações de gerenciamento são abstrações, somente geradas quando requisitadas por uma aplicação de gerenciamento. Outras informações de gerenciamento são utilizadas como um argumento para iniciar ações em um subsistema gerenciado.

---

<sup>2</sup> Uma definição para os termos [PeMc97]: **Tempo de compilação** refere-se ao tempo durante o qual um programa ou módulo é compilado; **Tempo de link-edição** refere-se ao tempo durante o qual todos os módulos são combinados na forma de um código executável; **Tempo de carga** refere-se ao tempo durante o qual um código é carregado pelo sistema operacional para ser executado; **Tempo de inicialização** refere-se ao início da fase de execução de um programa, em que ocorrem alocação de recursos ou processamentos; **Tempo de execução** é a fase de execução de um programa, após ter ocorrido sua inicialização e antes de seu término.

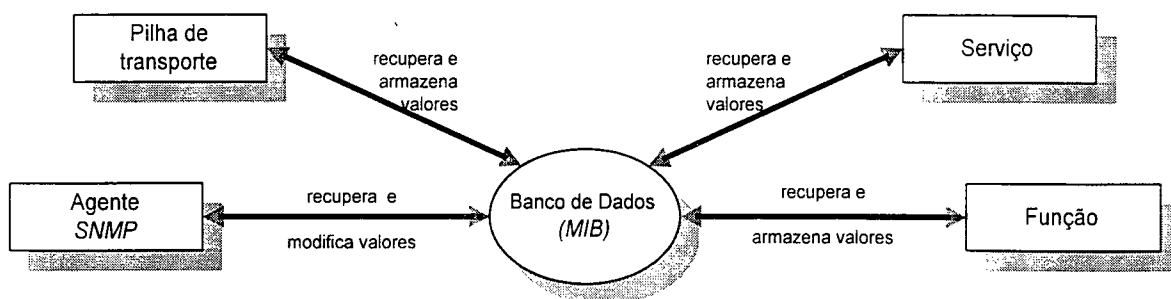


FIGURA 3.7: A *MIB* como um banco de dados

Se as informações de gerenciamento estão contidas em um banco de dados, então as *MIBs* projetadas e a implementação dos Agentes poderão ser consideradas uma tarefa fácil. Entretanto, o sistema gerenciado poderá sofrer sobrecarga pelo consumo de mais recursos de memória e de processamento. Todas as informações que possuem troca de valores, tais como *counters* e *gauges*, poderão causar atualizações do banco de dados.

### 3.4 Operações e Mensagens *SNMP*

O modelo *SNMP* possui um conjunto de operações utilizadas via troca de mensagens entre Aplicações Gerentes e Aplicações Agentes. Um Gerente pode recuperar ou modificar o valor de uma informação de gerenciamento acessada por um Agente. Um Agente pode enviar um relatório de evento para um Gerente, que pode informar a outro Gerente o valor de uma informação de gerenciamento obtida de um Agente. Usando operações de recuperação e modificação, um Gerente pode solicitar a um Agente que execute uma ação ou um comando, visando, por exemplo, à criação e à exclusão de instâncias de informações de gerenciamento.

As operações previstas no *SNMP* (TAB. 3.2) ocorrem por meio da troca de mensagens sobre um mecanismo de transporte. A recuperação e modificação de informações requerem mensagens do tipo *request*, *response* e *event report (trap)*.

TABELA 3.2: Operações do protocolo *SNMP*

OPERAÇÃO E DESCRIÇÃO	DE	MENSAGEM ENVIADA	PARA	MENSAGEM RETORNADA
<b>GET</b> recupera o valor de informações de gerenciamento.	Gerente	<i>get-request</i>	Agente	<i>get-response(v1)</i> ou <i>response(v2)</i>
<b>GETNEXT</b> recupera o valor de informações de gerenciamento existentes após um determinado identificador.	Gerente	<i>get-next-request</i>	Agente	<i>get-response(v1)</i> ou <i>response(v2)</i>
<b>GETBULK</b> estende a funcionalidade da operação	Gerente	<i>get-bulk-request(v2)</i>	Agente	<i>response(v2)</i>
<b>SET</b> modifica o valor de informações de gerenciamento.	Gerente	<i>set-request</i>	Agente	<i>get-response(v1)</i> ou <i>response(v2)</i>
<b>TRAP</b> informa um evento ocorrido no sistema gerenciado.	Agente	<i>trap(v1)</i> ou <i>snmpV2-trap(v2)</i>	Gerente	não ocorre
<b>INFORM</b> fornece uma informação de gerenciamento não-solicitada.	Agente- Gerente	<i>informe-request(v2)</i>	Gerente	<i>response(v2)</i>

### 3.5 O Pacote de Gerenciamento *CMU-SNMP*

O Pacote *CMU-SNMP*, é uma plataforma portátil para *SNMP v1* e *v2*, que fornece uma coleção de ferramentas de gerenciamento de rede. O Pacote é livremente disponível e tem sido portado para diversas implementações do Sistema Operacional *Unix*.

### 3.5.1 Os componentes de uma Aplicação Gerente

Dentre os componentes previstos para uma Aplicação Gerente, o pacote de *software CMU-SNMP* possui a implementação de operações e aplicações de gerenciamento, executadas por meio de utilitários, conforme apresentado na TAB. 3.3.

TABELA 3.3: Operações e aplicações de gerenciamento do *CMU-SNMP*

Operações de gerenciamento	Aplicações de gerenciamento
<i>Snmget</i>	<i>Walk</i>
<i>Snmgetnext</i>	<i>Netstat</i>
<i>Snmset</i>	<i>Status</i>
<i>Snmtrapd</i>	<i>Test</i>

Dessa forma, o desenvolvimento de uma Aplicação Gerente deverá ser efetuado adicionando-se os demais componentes: *MIB* (quando necessária), Banco de Dados, Aplicações de Gerenciamento adicionais e Interface de Usuário.

### 3.5.2 Os componentes de uma Aplicação Agente

No Pacote *CMU-SNMP*, portado para *Linux*, é distribuído um Agente *SNMP* com suporte às *MIBs* definidas nas *RFCs* 1213, 1414 e 1514. As notificações (*Traps*) são enviadas para a Aplicação Gerente com o uso do utilitário *snmptrapd*. Como protocolo de transporte, foi escolhido o *UDP*.

Dentre as áreas funcionais previstas para um Agente, já se encontram implementados, no utilitário *snmpd*, a Pilha de transporte, a *Engine* de protocolo e a *Dispatch Table*. O acréscimo de novos módulos na *MIB* deverá ser efetuado de forma manual. A instrumentação das variáveis e as interfaces para métodos e rotinas, deverão ser adicionadas ao *daemon* via compilação, o que caracteriza a adoção de uma abordagem de implementação monolítica.



## Capítulo 4

### Gerenciamento Baseado em *Web*

A crescente popularidade da *World Wide Web* oferece uma nova forma de fornecer acesso amplo a aplicações de *software* complexas. Quase toda plataforma suporta *Web Browsers*, e muitas pessoas estão usando a *Web* como uma ferramenta simples para acessar diferentes serviços. Dessa forma, a *Web* está rapidamente tornando-se um recipiente para fins gerais, servindo de interface e ambiente de desenvolvimento para todo tipo de informação distribuída em rede ou contida num único banco de dados amplo.

A utilização das tecnologias da *WWW*, no complexo mundo do gerenciamento de rede, tem contribuído para integrar ferramentas poderosas à *Web*, as quais permitem gerenciar recursos de rede. Trata-se de uma mudança de paradigma, onde pessoas são capacitadas ao uso da *Web* no desempenho de operações ativas, tal como mudar a topologia da rede, ao invés de usá-la simplesmente para recuperar registros de banco de dados ou para seguir vínculos de hipertexto.

As atividades de gerenciamento de rede podem ser apoiadas com a combinação da tecnologia *SNMP* ao poder da *Web*, evitando, muitas vezes, a necessidade de aquisição de ferramentas de custo elevado e de uso satisfatório em tarefas específicas.

#### 4.1 A World Wide Web

A *World Wide Web* permite que as pessoas trabalhem juntas, combinando seu conhecimento em documentos interligados com o uso de hipertexto. A *Web* é considerada um meio de comunicação que usa computadores como uma parte largamente invisível da infra-estrutura. A *Web* foi desbravada visando constituir um sistema capaz de lidar com vários protocolos da *Internet*, bem como diferentes formatos de dados, usando uma única e consistente interface de usuário. Em adição, este sistema possui como principais padrões o protocolo *HTTP* e a linguagem de formatação *HTML*. Uma visão simplificada do funcionamento da *Web* é apresentada na FIG. 4.1. Na seção 6.3, são apresentados detalhes do *HTTP*.

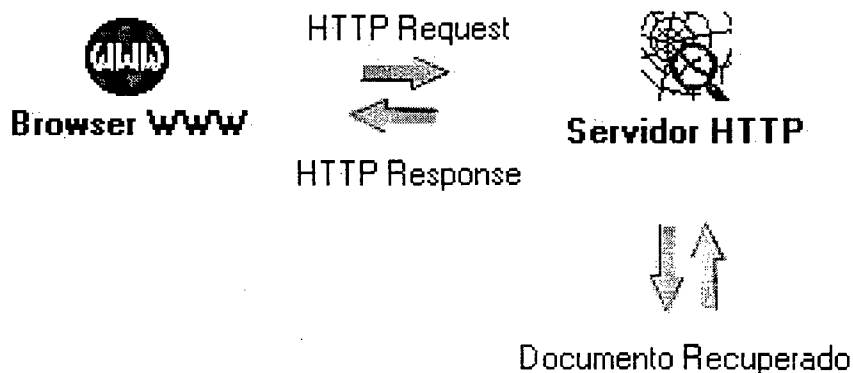


FIGURA 4.1: Visão simplificada do funcionamento da *Web*

#### 4.2 Integrando o Gerenciamento de Rede à *Web*

Estudos feitos na área de Gerenciamento de Redes [Der96] constataram que, embora muitas ferramentas estejam no mercado, elas freqüentemente não trabalham juntas de modo consistente. Além disso, elas são alvo para grandes corporações com redes imensas para gerenciar. O tempo necessário para instalar e configurar tais

ferramentas, juntamente com a dificuldade de usá-las, é considerado demasiado para um usuário de nível médio.

A idéia de integrar o gerenciamento de redes à *Web*, visa diminuir a frustração do usuário com ferramentas convencionais de gerenciamento de rede, substituindo-as, em muitos casos, por ferramentas simples e suportadas por quase todas as plataformas. Embora os computadores estejam tornando-se mais rápidos e potentes, ferramentas de *software* não mudaram muito. Há poucos anos, era aceitável ter ferramentas simples para atividades, tais como processamento de palavras e ferramentas complicadas, porém poderosas para tarefas mais complexas. Hoje os usuários demandam interfaces mais simples para tarefas fáceis e complexas.

A grande popularidade da *Web*, combinada com sua força e aceitação, tornou-a uma forte candidata a dar suporte a uma nova classe de ferramentas de gerenciamento de rede. Essas ferramentas estão acessíveis para cada plataforma, capaz de rodar um *Web Browser*, e possuem uma interface de usuário bastante simples. Frequentemente, interfaces gráficas não são muito fáceis de usar, embora pareçam atraentes e convidativas.

Esta nova classe de aplicações de gerenciamento de rede baseada em *Web* deverá ter as seguintes propriedades [Der96]:

- ferramentas diferentes têm o mesmo “visual” e “toque”;
- configuração limitada: o sistema localiza recursos de rede e limita o número de escolhas que o usuário tem que fazer, fornecendo valores ausentes;
- auxílio *on-line*: documentação e outras facilidades estão diretamente disponíveis *on-line* e exploram as facilidades de hipertexto da *Web*;
- apoio ativo: em caso de erro, o sistema identifica a causa do problema e indica possíveis soluções;
- alternativas limitadas: o sistema mostra apenas operações válidas, prevenindo, assim, o usuário de utilizar as inválidas.

A escolha da *Web* pode ser justificada em função de:

- ser uma interface bastante conhecida pelo usuário;
- estar disponível virtualmente para cada plataforma;
- ser de propósito geral, capaz de acomodar os serviços de hoje e estar aberta a extensões futuras;
- ter uma capacidade de hipertexto integrada, ideal para navegar diferentes tipos de informações;
- possuir facilidades de segurança fornecidas pelos protocolos de gerenciamento de rede que podem ser combinadas com protocolos emergentes de segurança da *Web*;
- ser um programa independente, que pode ser rodado em qualquer plataforma.

### 4.3 Aspectos a considerar para o gerenciamento de rede via *Web*

A tendência dos últimos anos é de substituir computadores centralizados e poderosos por uma rede de computadores médios e baratos. O gerenciamento de rede foi afetado pelo *downsizing*, e a tendência atual é usar grupos médios para rodar Aplicações Agentes e reservar máquinas de poder limitado para Aplicações Gerentes. Estas aplicações gerenciadoras apresentam:

- facilidade de uso: usuários com máquinas médias estão acostumados à metáfora *Windows Input Mouse Pointers* e não gostam de aplicações com uma interface de usuário pobre;
- *plug and play*: usuários médios querem instalar uma aplicação e rodá-la imediatamente, sem necessidade de tornarem-se *experts* em configuração;
- utilização mínima de recursos: aplicações de gerência devem causar o menor impacto possível em outras aplicações correntes.

A mobilidade é um outro fator importante, pois as pessoas tendem a mudar-se mais freqüentemente do que no passado, e um conjunto completo de novas tecnologias tem de ser introduzido, a fim de acessar recursos de rede transparente e independentemente da localização e do tipo de computador. Este requisito obriga os desenvolvedores de *software* a criarem aplicações que rodem eficientemente em diferentes plataformas, independentemente da versão e tipo do sistema operante. Uma solução é usar estruturas que escondam as diferenças entre os sistemas de operação, usando agentes *proxy* que possam ser acessados por um conjunto de protocolos, bem estabelecido como *TCP/IP*, ou pelas linguagens interpretadas. Neste cenário complexo, a *Web* desempenha um papel chave, devido à sua popularidade crescente e sua habilidade em acessar qualquer tipo de informação, usando uma única interface estabelecida.

#### 4.4 Migrando para o Gerenciamento Baseado em *Web*

Devido ao grande sucesso da *Web* como interface padrão para diferentes aplicações [QDS97], desenvolvedores comerciais e organismos de padronização estão elaborando diversas estratégias de migração para o gerenciamento baseado em *Web*. Na FIG. 4.2, são apresentadas as tecnologias de gerenciamento que estão sendo migradas para *Web*.

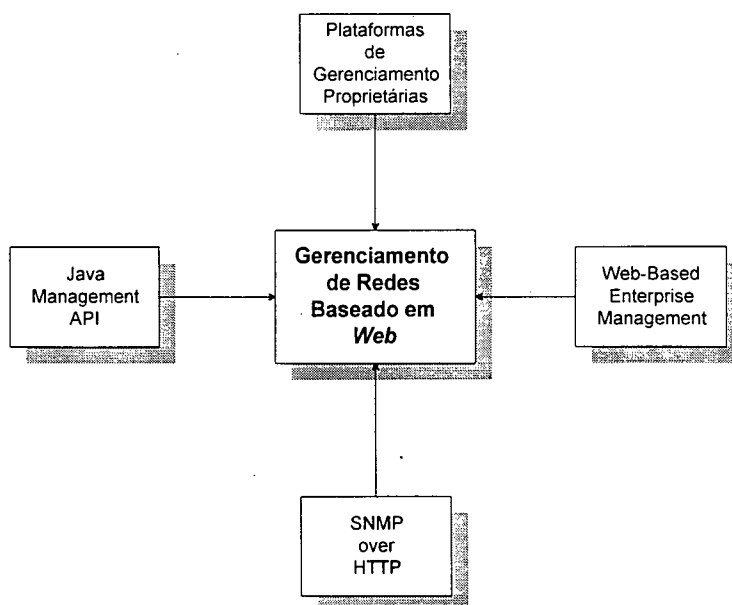


FIGURA 4.2: Tecnologias em fase de migração para *Web*

#### 4.4.1 Plataformas de Gerenciamento de Redes

As Plataformas de Gerenciamento de Rede são oferecidas em produtos comerciais. Dentre inúmeras plataformas existentes, destacam-se [Har97]: *OpenView Network Node Manager (HP)*, *Solstice Domain Manager (SunSoft)*, *Spectrum (Cabletron)* e *NetView (IBM)*.

As plataformas de gerenciamento proprietárias utilizam, em sua maioria, o protocolo de gerenciamento *SNMP*, e atuam intrinsecamente sobre as redes que fornecem a pilha de protocolos *TCP/IP*. Essas plataformas oferecem recursos para atuarem junto às áreas funcionais, permitindo, assim, o gerenciamento de todas ou algumas dessas áreas.

Em algumas plataformas, os serviços de manipulação de objetos comportam os padrões da *MIB I* e *MIB II*, propostas pelo *IETF*, e apresentam ferramentas para a extensão das *MIBs* existentes, e, em acréscimo, a inclusão de módulos com especificações não-padronizadas.

Interfaces gráficas são fornecidas para facilitar a consulta e alteração sobre os objetos de uma *MIB*. Algumas das plataformas oferecem interfaces gráficas aos usuários, contendo símbolos gráficos que apóiam a representação da topologia de rede, contexto em que, os componentes são reproduzidos por meio de *ícones*. Essas interfaces consistem em ferramentas poderosas para a manipulação de informações: dados podem ser coletados e, posteriormente, apresentados em gráficos bi ou tridimensionais. Quanto à observação de anomalias na rede, algumas plataformas notificam a ocorrência por meio de avisos sonoros, alterações de símbolos gráficos e mensagens de correio eletrônico.

#### 4.4.2 *Web-Based Enterprise Management*

O propósito inicial do *Web-Based Enterprise Management - WBEM* [FMI97] é consolidar e unificar dados fornecidos por tecnologias de gerenciamento existentes. As atenções desta abordagem estão voltadas a empresas, visando ao rastreamento de áreas problemáticas que iniciam em nível de usuário ou aplicação, evoluindo até os sistemas e camadas de rede para instâncias de serviços ou servidores remotos.

No enfoque desta proposta, podem ser identificadas duas iniciativas:

- Definição de uma extensibilidade, que corresponde à implementação independente da descrição/esquema de dados em comuns, permitindo que dados de uma variedade de origens sejam descritos e acessados, independentemente da fonte original daqueles dados.
- Definição de um protocolo padronizado sobre o qual estes dados podem ser publicados e administrados, para serem independentes de plataformas e fisicamente distribuídos em uma empresa.

A proposta do *WBEM* não tenta substituir padrões de gerenciamento existentes como *SNMP*, *DMI* ou *CMIP*, ou impedir plataformas proprietárias ou estruturadas específicas. De fato, o *WBEM* visa complementar essas iniciativas, incluindo um ponto de integração em que dados de todas as fontes possam ser acessadas. Isso faz com que qualquer aplicação de gerenciamento, independentemente de *APIs* específicas ou padronizadas, manipule as entidades de gerenciamento, permitindo correlação de dados e eventos de múltiplas fontes sobre uma base local ou distribuída.

#### 4.4.3 *SNMP over HTTP*

Um servidor *HTTP*, apoiando a função de gerenciamento, pode produzir uma variedade de conteúdos, incluindo páginas formatadas com *HTML*, gráficos, códigos

executáveis e dados binários. As mensagens (*Protocol Data Unit - PDU*) do *SNMP* podem ser transmitidas sobre o *HTTP* como dados do tipo *application/octet-stream*. Esta proposta foi encaminhada ao *IETF* por *Harrison* [HMP96].

Como descrito na *MIB* de gerenciamento *HTTP*, o objeto da *MIB* *httpMgSNMPEnabled* determina se um agente *SNMP/HTTP* pode suportar *SNMP* sobre *HTTP*. A *URL* usada é definida no objeto da *MIB* *httpMgDefaultURL*. Esta *URL* vai incluir uma referência à porta 280 quando for atribuído a um servidor *HTTP* suporte para *SNMP* sobre *HTTP*. Requisições *SNMP* devem ser feitas, encapsulando a codificação da *PDU* em uma requisição *HTTP* (FIG. 4.3).

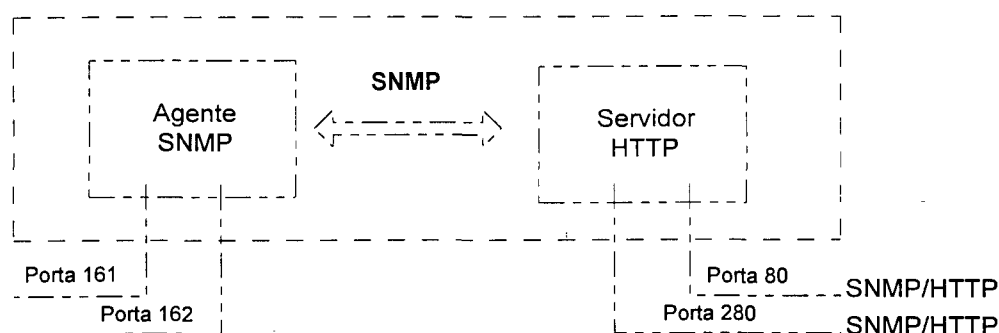


FIGURA 4.3: Esquema para troca de mensagens entre o Agente *SNMP* e o *Web Server*

#### 4.4.4 Java Management API

A *Application Program Interface* de gerenciamento *Java* [Sun97] é composta por um conjunto de objetos extensíveis e de métodos para o desenvolvimento de sistemas completos, destinados ao gerenciamento de redes e serviços em plataformas heterogêneas.

Utilizando a *API* de gerenciamento *Java*, desenvolvedores criam aplicações consistentes com uma curva de aprendizagem decrescente e um ciclo de desenvolvimento reduzido [JaSo96]. A *API* de Gerenciamento *Java* inclui o *Java Management API User Interfaces Style Guide*, documento que fornece um guia para o



desenvolvimento de interfaces de usuários baseadas em *Java/Web*. Esse modelo de guia de estilo foi construído para endereçar a configuração e localização de problemas, apoiando os administradores de pequenos e grandes ambientes de rede.

A *API* de gerenciamento Java inclui também o *Admin View Modulo*, que é uma extensão do *Java Abstract Window Toolkit* especialmente designada para o desenvolvimento de interfaces de usuários utilizadas em aplicações de gerenciamento distribuídas. Tais aplicações compreendem o desenvolvimento de gerentes e os agentes com o uso de classes *Java* (FIG. 4.4).

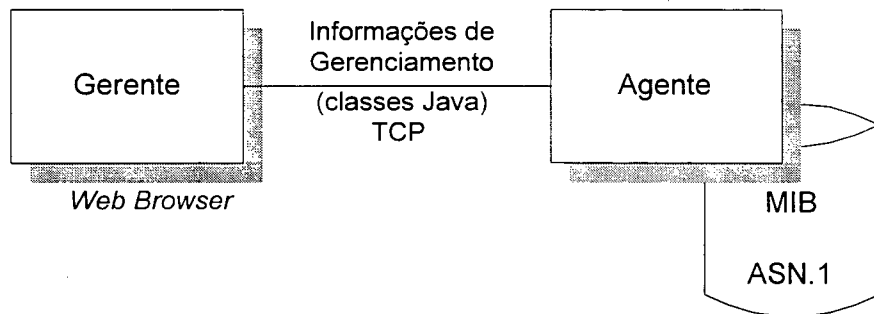


FIGURA 4.4: Agentes de Gerenciamento de Rede suportados por *Java*

O *Browser* Gerente consiste de uma interface gráfica de usuário composta de painéis, campos e botões, construída com o uso do Gerente de *Layout* existente na *Abstract Window Toolkit Class Java*. Os processos executados por intermédio da *GUI* do sistema de gerenciamento de rede utilizam classe *Java*.

O Agente é uma aplicação *Java* que executa funções de gerenciamento sobre elementos de rede. O código da Aplicação Agente é executado na forma de *background process* em diferentes sistemas operacionais.

## Capítulo 5

### Proposta para o gerenciamento de aplicações em rede

Neste capítulo é apresentada uma proposta para o gerenciamento de aplicações em rede, a qual compreende o desenvolvimento de seis atividades. As etapas propostas são apresentadas de forma seqüencial por uma questão meramente didática, pois em realidade, quando aplicadas, poderão seguir uma abordagem circular, ou seja, o acúmulo de conhecimento e definições, com o desenvolvimento das atividades, poderá implicar a reformulação de atividades anteriores. Uma representação gráfica do ciclo de vida da proposta é apresentada na FIG. 5.1.

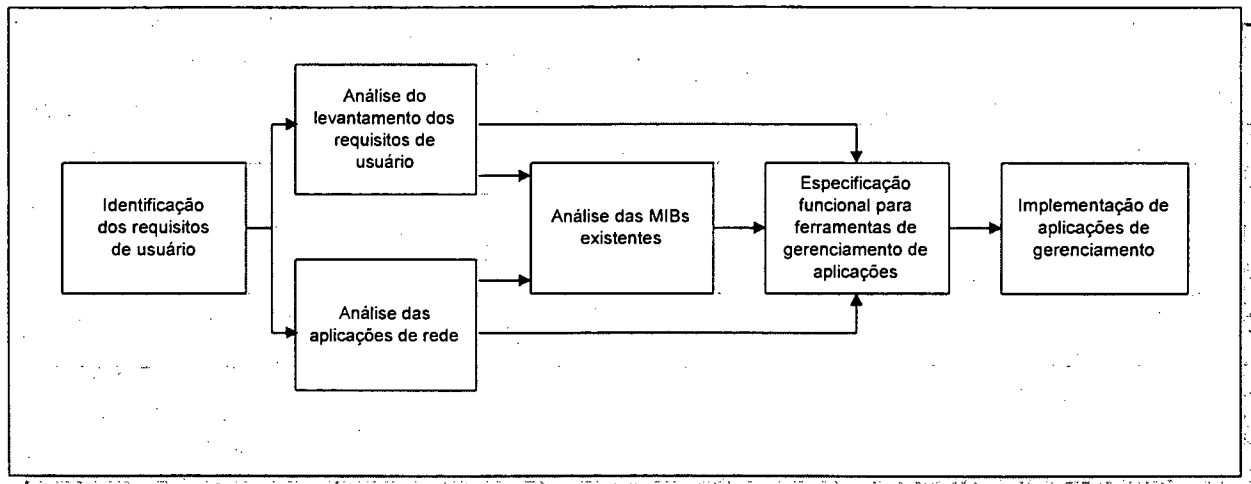


FIGURA 5.1: Ciclo de vida das atividades propostas

## 5.1 Identificação dos requisitos de usuário (atividade 1)

Esta atividade compreende o levantamento de dados junto a indivíduos relacionados com o desenvolvimento, a administração e/ou utilização de aplicações de rede, visando identificar diferentes papéis e necessidades de gerenciamento de aplicações distintas, porém comuns a cada categoria de usuários. O enfoque do levantamento poderá ser direcionado a determinado grupo de aplicações<sup>1</sup>, de usuário ou tipo de aplicação, visando a definição de um escopo para a pesquisa, tanto em nível de dados (tipo e quantidade) como de população que será pesquisada.

Os dados levantados junto aos desenvolvedores de aplicações deverão possibilitar a identificação da existência ou do interesse destes, em relação a funcionalidades de gerenciamento incorporadas nos produtos. O levantamento junto aos administradores de aplicações deverá ser capaz de indicar o nível de apoio que as ferramentas de gerenciamento poderão proporcionar às suas atividades operacionais. Junto aos usuários finais, os dados levantados deverão indicar os requisitos para o fornecimento de informações em tempo real, relativas (que reportem) a índices (métricas e indicadores) associados à qualidade dos serviços oferecidos pelas aplicações.

O número de usuários pesquisados deverá permitir a extração de requisitos e tendências para as aplicações de gerenciamento, de forma representativa e coerente. A população pesquisada deverá pertencer a grupos de usuários que utilizam: diferentes velocidades de conexão (*link*) com a *Internet*; diferentes plataformas de *hardware* e *software*; diferentes níveis de conhecimento em relação às aplicações de rede utilizadas e diferentes locais de atuação - uma vez que cada usuário interage com as aplicações e serviços com diferentes ponto de vista.

O levantamento de dados poderá ser efetuado pessoalmente junto aos grupos de usuários, ou de forma *on-line* com o uso de formulários de *Web*. Cabe observar que,

---

<sup>1</sup> As aplicações de rede podem ser agrupadas em diferentes categorias, como, por exemplo: aplicações clientes, aplicações servidoras, aplicações utilizadas em serviços de informações via WWW.

para chegar a este ponto, faz-se necessário todo o trabalho de planejamento e execução das etapas anteriores ao levantamento, em conformidade com os princípios da estatística [Mat96]. Para preenchimento dos formulários, poderão ser efetuadas chamadas junto a *newsgroups* e listas de discussão relacionadas. Os dados enviados via formulários eletrônicos poderão ser armazenados em um Sistema de Gerenciamento de Banco de Dados-SGBD e processados posteriormente mediante análises estatísticas. Os resultados do levantamento deverão ser utilizados como entrada na atividade **2 - Análise dos requisitos de usuário** e apoiar a definição de escopo para atividade **3 - Análise das aplicações de rede**.

## **5.2 Análise dos requisitos de usuário (atividade 2)**

Compreende a formação de subsídios, por meio de análises estatísticas dos dados levantados, visando à definição de módulos para a *MIB* e do Banco de Dados, isto é, a identificação de um conjunto de informações de gerenciamento de interesse dos usuários. O propósito da análise é a obtenção de significados nos dados coletados. Esta atividade pressupõe a execução de cálculos estatísticos visando estabelecer cenários que indiquem os requisitos e necessidades de gerenciamento de aplicações.

Os usuários finais poderão estar interessados em ter acesso a estatísticas de desempenho e em ser notificados sobre as causas que provocaram falhas nas aplicações. Por outro lado, desenvolvedores e administradores podem estar interessados em aplicações de gerenciamento que facilitem a detecção e correção de problemas junto às aplicações executadas em ambiente distribuído.

A identificação da abrangência (heterogeneidade, distribuição geográfica e aplicações utilizadas) dos sistemas onde estão inseridos os usuários e de tendências e requisitos comuns para as aplicações de gerenciamento, é outro resultado que deverá ser produzido nessa atividade.

A análise dos dados levantados deverá fornecer informações quantitativas e qualitativas, necessárias para a especificação funcional das ferramentas de gerenciamento. As informações resultantes da análise deverão permitir a identificação: dos requisitos de gerenciamento impostos por usuários finais; das dificuldades enfrentadas pelos administradores, que deverão ser apoiadas por ferramentas de gerenciamento, e das funcionalidades de gerenciamento oferecidas ou previstas pelos desenvolvedores de aplicações.

As informações resultantes da análise do levantamento efetuado deverão ser utilizadas como entrada nas atividades: **4 - Análise das MIBs existentes** e **5 - Especificação funcional para ferramentas de gerenciamento de aplicações.**

### **5.3 Análise das aplicações de rede (atividade 3)**

Esta atividade compreende o estudo da configuração e do comportamento dos protocolos e de suas respectivas implementações em nível da camada de aplicações. Deverá ser executada, nesta atividade, a análise detalhada das funcionalidades e arquitetura das aplicações, visando à identificação e parametrização de componentes que podem refletir o comportamento de um determinado sistema e que podem ser importantes para a computação das funções de gerenciamento que serão implementadas.

Esta atividade tem como objetivo fornecer alguns indicativos para o mapeamento de um conjunto comum de parâmetros de configuração e objetos de monitoramento, que poderão ser incluídos na *MIB*. A identificação do relacionamento operacional e dependências das aplicações em relação a outras aplicações, poderá, também, indicar a necessidade de objetos na *MIB* e apoiar a instrumentação de agentes usados em qualquer cenário de gerenciamento, habilitando-os a fazerem uso de procedimentos comuns que estão implementados nas *engines* das aplicações.

A identificação dos requisitos de gerenciamento com base na análise das aplicações de rede, poderá ser apoiada por meio da observação de uma arquitetura

genérica e da respectiva decomposição de seus componentes, capaz de representar as funcionalidades das aplicações como um todo ou de uma categoria de aplicações. Dentre os aspectos a serem observados, encontram-se:

- Configuração e administração da aplicação: esta análise visa apoiar a definição de objetos que possibilitem a configuração e a administração remota da aplicação.
- *Logging* suportados: a análise do mecanismo e dos dados da capacidade de *logging* poderá contribuir para a implementação de monitoramento remoto.
- Desempenho: a identificação e o estudo de parâmetros de desempenho poderão apoiar o projeto da *MIB* e a definição de indicadores e métricas para o delineamento dos cenários de gerenciamento.

Inicialmente, todas as aplicações que serão gerenciadas deverão ser tratadas com base em um determinado modelo genérico, visando a uma abordagem uniforme para a definição da *MIB*. À medida que as informações resultantes da análise forem tomando significados específicos e dependentes de aplicação ou de funcionalidades de gerenciamento, grupos específicos de objetos podem ser definidos. A principal vantagem na utilização de grupos de objetos genéricos reside na possibilidade de gerenciar novas aplicações sem precisar trocar as semânticas da *MIB*.

Nessa atividade, o levantamento das seguintes informações poderá ser considerado no processo de análise:

- Identificação dos arquivos de configuração e diretivas de execução.
- Método de execução da aplicação (*standalone / inetd-started*).
- Variáveis de configuração e diretivas de execução para diferentes implementações de um mesmo protocolo de aplicação.
- Tipos de arquivos de *Log* atualizados pela aplicação.

- Formato dos arquivos de *Log* atualizados pela aplicação.
- Ferramentas de software para o processamento dos registros de *Log*.
- Existência de funcionalidades de gerenciamento.

Os resultados dessa atividade deverão apoiar as atividades: **4 - Análise das *MIBs* existentes** e **5 - Especificação funcional para ferramentas de gerenciamento de aplicações**.

#### **5.4 Análise das *MIBs* existentes** (atividade 4)

O estudo dos relevantes esforços internacionais destinados à definição de *MIBs* para o gerenciamento de aplicações de rede, compreende o principal objetivo desta atividade, cujos resultados deverão ser utilizados na atividade **5 - Especificação funcional para ferramentas de gerenciamento de aplicações**.

Essa atividade deverá ser capaz de relatar a aplicabilidade das linhas de *MIBs* que estão sendo propostas e padronizadas para o gerenciamento de aplicações, identificar a possibilidade de utilização das diferentes *MIBs* em conjunto, relacionar suas vantagens e desvantagens, evitar a sobreposição de objetos e propor objetos adicionais necessários ao atendimento dos requisitos de usuários e do gerenciamento efetivo das aplicações.

Uma análise completa de módulos da *MIB* poderá ser efetuada, considerando-se a seguinte lista de aspectos a observar [PeMc97]:

- *Background* sobre a tecnologia que foi modelada.
- Descrição dos recursos que foram modelados.
- Estruturação em alto nível das definições existentes no módulo da *MIB*.
- Forma de relacionamento do módulo, analisado com outros módulos da *MIB*.
- Definição do modo como as especificações de objetos são mapeadas para os recursos.

- Esquema de indexação para cada tabela.
- Forma pela qual é feito o relacionamento entre tabelas.
- Requisitos para a criação e deleção de linhas via operações *SNMP*.
- Uma classificação dos objetos e eventos com relação a: configuração, monitoramento, controle, ações, estatísticas, estado, componentes e atributos.
- Uma lista das principais aplicações de gerenciamento que poderão ser efetuadas com a utilização das definições presentes no módulo da *MIB*.

Dentre as fontes (materiais bibliográficos) que poderão ser utilizadas para subsidiar a lista de aspectos a serem analisados, encontram-se: artigos sobre definição de *MIBs*, *RFCs* informacionais e *RFCs* contendo propostas de *MIBs*, e especificações técnicas dos recursos modelados (manuais de desenvolvedores e especificações de protocolos para camada de aplicações).

## **5.5 Especificação funcional para ferramentas de gerenciamento de aplicações** (atividade 5)

Com base nos resultados produzidos nas atividades de análise (2, 3 e 4), uma abordagem e arquitetura de gerenciamento baseada no paradigma Gerente-Agente deverá ser escolhida. Um conjunto genérico de informações e de funções de gerenciamento também deverá ser definido.

### **5.5.1 Abordagens e arquiteturas de gerenciamento**

Inicialmente, diferentes abordagens e arquiteturas de sistemas de gerenciamento que estão sendo utilizadas ou propostas ([SCM95] e [StMa97]) - como, por exemplo, gerenciamento centralizado, gerenciamento distribuído, gerenciamento hierárquico ou outra que combine aspectos de duas ou mais abordagens - deverão ser avaliadas no sentido de escolher a que melhor atende os requisitos de gerenciamento. Os resultados da avaliação deverão indicar as vantagens e desvantagens da abordagem escolhida.



A identificação das capacidades e limitações das abordagens e arquiteturas existentes para o gerenciamento de redes poderá ser efetuada, considerando-se:

- Componentes suportados (Gerentes, Agentes-Gerentes, Agentes e *MIBs*).
- Desenvolvimento (fácil / difícil).
- Manutenção (fácil / difícil).
- Escalabilidade (sim / não).
- Gerenciamento por domínio (sim / não).

A natureza do funcionamento das aplicações de rede, que poderá compreender processamento centralizado ou distribuído, independente ou dependente e relacionado com outras aplicações, deverá também ser considerada quando da escolha do tipo de abordagem e arquitetura que serão utilizadas para implementação do sistema de gerenciamento.

Aplicações de rede utilizadas na montagem e oferecimento de serviços distribuídos, como, por exemplo, sistemas de informações da *WWW*, poderão ser melhor gerenciadas com base em uma abordagem e arquitetura que permita a execução de operações de gerenciamento (controle e monitoramento) de forma eficiente e em diferentes níveis de abstração e áreas geográficas, o que poderá indicar a adoção de uma arquitetura combinada, do tipo gerenciamento hierárquico-distribuído.

Outro fator que deverá ser considerado na escolha da arquitetura, diz respeito à necessidade de minimizar o tráfego de informações de gerenciamento em ambientes de *LAN* e, principalmente, de *WAN*.

### **5.5.2 Informações de gerenciamento**

Deverá ser um dos processos executados nesta atividade, a avaliação da possibilidade de agrupamento de aplicações por meio da definição de objetos de gerenciamento comuns, visando criar uma interface e representação uniforme, onde

diferentes implementações e novas aplicações poderão ser adicionadas, sem a necessidade de recompilar a *MIB*.

Somente parâmetros genéricos, suportados pelas aplicações que serão gerenciadas, deverão ser considerados no projeto da *MIB* e conseqüente instrumentação. Objetos associados a características e parâmetros de protocolos da camada de aplicação deverão ter um enfoque genérico, e não específico a uma determinada versão de protocolo. Se certas características específicas para uma determinada versão de protocolo necessitam ser gerenciadas, os objetos deverão ser organizados em grupos relacionados às específicas versões.

O escopo das informações de gerenciamento deverá compreender mecanismos para:

- Existência de informações administrativas genéricas sobre as aplicações gerenciadas.
- Gerenciamento remoto da configuração das aplicações.
- Apuração de dados estatísticos, destinados, por exemplo, à estimação da qualidade dos serviços de rede oferecidos pelas aplicações de rede.
- Um conjunto de alarmes gerados por Aplicações Gerente intermediárias (Agente-Gerente) e direcionados a aplicações de nível superior, sempre que eventos anormais ocorrerem.

### **5.5.3 Funções de gerenciamento**

As funções de gerenciamento deverão ser definidas com base nos requisitos dos usuários, desempenho, implementação das entidades do modelo de gerenciamento e na

perspectiva de arquitetura das aplicações gerenciadas, a qual fornece o ponto de partida para o projeto do sistema de gerenciamento.

Como requisitos de usuários, poderão surgir:

- Identificação automatizada das aplicações e dos respectivos serviços de rede implementados, relacionados e dependentes.
- Condições de erros e falhas classificados em função de parâmetros temporais, por tipos de serviços ou domínios de aplicações clientes ou servidoras, objetivando identificar erros padrões, suas origens e causas.
- Processamento de dados estatísticos com base no monitoramento de *Logs*, com o uso de entidades de gerenciamento do tipo Agente.
- Configuração remota dos parâmetros e diretivas de execução das aplicações gerenciadas.

Como requisito de desempenho para as funções de gerenciamento, podem surgir:

- A entidade de gerenciamento do tipo agente deverá causar o menor impacto possível em relação ao conjunto de recursos de processamento (*CPU*, memória, ...) e armazenamento junto ao recurso gerenciado. Este requisito impõe restrições ao método de implementação que deverá ser utilizado. A forma, como as instâncias de objetos da *MIB* são atualizadas e mantidas é também um importante fator de desempenho. A grande maioria dos objetos definidos para a *MIB*, deverá ter seus valores calculados sob demanda e, se possível, derivados de outros recursos do sistema, evitando-se o armazenamento dos dados da *MIB* junto à memória da entidade Agente ou em dispositivos de armazenamento permanente.

- Minimização do tráfego total gerado por entidades do protocolo de gerenciamento nas interações Gerente-Agente, principalmente em ambientes de *WAN*.
- Utilização de recursos computacionais parcialmente dedicados à implementação de entidades de gerenciamento do tipo Gerente.

Como requisitos de implementação das funções de gerenciamento, podem surgir:

- Necessidade de utilização de compiladores de *MIB* para verificação da sintaxe dos módulos de informações de gerenciamento definidos.
- Utilização de mecanismos para descrever o conjunto de capacidades suportadas por entidades do tipo Agente, possibilitando aos desenvolvedores indicarem o nível preciso de suporte que um Agente sustenta em relação ao módulo da *MIB*.
- Inserção do módulo de gerenciamento no código da aplicação que será gerenciada, tornando possível implementar atividades de gerenciamento de controle.
- Utilização de uma plataforma de gerenciamento proprietária ou construção de uma plataforma baseada em tecnologias abertas. Em ambas as situações, deverão ser previstos os componentes mínimos estipulados para as Aplicações Gerente, conforme foi abordado no Capítulo 3.

As especificações resultantes dessa atividade deverão apoiar os processos que serão desenvolvidos na atividade **6 - Implementação de aplicações de gerenciamento**.

## 5.6 Implementação de aplicações de gerenciamento

As aplicações de gerenciamento podem ser agrupadas em duas categorias: Aplicações Gerentes e Aplicações Agentes. Dessa forma, inicialmente deve-se verificar que categorias de aplicações de gerenciamento serão implementadas e o nível de implementação necessário, já que existe a possibilidade de serem utilizadas e/ou adaptadas implementações existentes.

### 5.6.1 Implementação de Aplicações Gerentes

Aplicações Gerentes estão disponíveis nas chamadas plataformas proprietárias de gerenciamento de rede, comercializadas pelos principais fabricantes de *hardware* e *software*. Como alternativa, existem pacotes de *software* de domínio público que implementam os padrões abertos de gerenciamento definidos, por exemplo, no *OSI - RM* e/ou no modelo *SNMP*.

Independentemente da alternativa escolhida, a implementação de Aplicações Gerentes deverá, no mínimo, compreender a disponibilização dos componentes descritos no Capítulo 3, que são:

- Operações de gerenciamento;
- *Management Information Base - MIB*;
- Banco de Dados;
- Aplicações de gerenciamento;
- Interface de usuário.

Dos componentes relacionados, a *MIB* de uma Aplicação Gerente somente será necessária quando esta aplicação desempenhar um duplo papel (Agente-Gerente). Este tipo de implementação geralmente ocorre quando se opta por uma abordagem e arquitetura de gerenciamento hierárquico-distribuída.

### 5.6.2 Implementação de Aplicações Agentes

De forma semelhante ao que ocorre com Aplicações Gerentes, existem Aplicações Agentes disponibilizadas em produtos comerciais ou em pacotes de *software* de domínio público. Uma outra possibilidade é fazer uso de funcionalidades de gerenciamento, inerentes ao sistema que será gerenciado.

Esta categoria de aplicação de gerenciamento poderá ser implementada na forma de agente ou subagente, compreendendo as áreas funcionais descritas no Capítulo 3, que são:

- Pilha de transporte
- *Engine* de protocolo
- *Dispatch table*

Outro fator que deverá ser considerado na implementação de Aplicações Agentes, é a estrutura da aplicação. Uma estrutura monolítica deverá ser adotada no gerenciamento de sistemas pequenos, dedicados e fechados. Por outro lado, em sistemas de propósitos gerais e abertos, deverá ser utilizada uma estrutura do tipo extensível.

Por último, na implementação de Aplicações Agentes deverá ser definida a forma de instrumentação das variáveis presentes na *MIB* que foi especificada. Independentemente dos requisitos de gerenciamento que deverão ser atendidos, a instrumentação deverá causar o menor impacto possível junto ao sistema gerenciado. Sempre que for viável, os processos de instrumentação deverão utilizar mecanismos já existentes no sistema, gerando informações somente quando requisitadas (sob demanda) por Aplicações Gerentes.

# Capítulo 6

## Aplicação da proposta no gerenciamento do *HTTP*

Este capítulo trata da aplicação da proposta apresentada no capítulo anterior. Particularmente, apresenta-se o desenvolvimento das atividades previstas, aplicadas no desenvolvimento de um protótipo de sistema de gerenciamento, destinado ao monitoramento de implementações do *HTTP*.

### 6.1 Identificação dos requisitos de usuário

Por tratar-se da experimentação da proposta efetuada e de sua utilização em primeira mão, optou-se por conduzir o levantamento dos requisitos de usuário via contato direto e restrito à categoria de administradores de aplicações.

Inicialmente foi disponibilizado via *Web* na *URL* <http://redes.ucpel.tche.br/~lftm> o conjunto de dados a serem pesquisados. Em seqüência, foi agendada uma entrevista com os administradores de aplicações de instituições locais (Pelotas - RS), na qual foram coletadas as informações. Cabe salientar que, embora tenham sido bastante reduzidos os procedimentos e o escopo para a identificação dos requisitos de usuário, tais restrições foram compensadas com a experiência profissional adquirida em relação ao sistema que será gerenciado. As questões relacionadas a seguir (FIG. 6.1)

compreendem os itens que, juntamente com aspectos de ordem prática, foram considerados para a execução das demais atividades.

- |   |
|---|
| <p>1 - <i>Bandwidth</i> da conexão com a Internet</p> <p>2 - Servidores <i>Web</i> utilizados</p> <p>3 - <i>Hardware</i> e sistemas operacionais onde estão funcionando os servidores <i>Web</i></p> <p>4 - Existência de conteúdo com acesso restrito</p> <p>5 - Posição dos servidores <i>HTTP</i> em relação à estrutura de rede da organização</p> <p>6 - Utilização ou aprovação para o uso de ferramentas no monitoramento de segurança e de contabilização</p> |
|---|

FIGURA 6.1: Dados utilizados nas demais atividades da proposta

## 6.2 Análise do levantamento dos requisitos de usuário

A análise dos dados levantados, conforme descrito anteriormente, resultou na identificação das questões apresentadas na FIG. 6.1, compiladas na TAB. 6.1, e na formação de subsídios para a análise das aplicações de rede que serão gerenciadas, análise das *MIBs* existentes e especificação funcional para ferramentas de gerenciamento.

TABELA 6.1: Compilação dos dados levantados junto aos usuários

DADOS LEVANTADOS		UFPEL	ETPEL	UCPEL
<i>Bandwidth</i> com a Internet		64 Kbps	28Kbps	2Mbps
Ambiente Dos Web Servers	<i>Hardware</i>	<i>Workstation</i>	PC	PC <i>Workstation</i>
	<i>Sist. Operacional</i>	<i>DEC OSF/1</i>	<i>Linux</i>	<i>Linux</i> <i>AIX</i>



DADOS LEVANTADOS		UFPEL	ETFPEL	UCPEL	
<i>Web Servers Utilizados</i>	Nome	<i>CERN</i>	<i>Apache</i>	<i>Apache</i> <i>NCSA</i>	
	Versão	3.0	1.2.4	1.2.4	
				1.5.1	
Conteúdo com acesso restrito		NÃO	NÃO	SIM	
Posição dos <i>Web Servers</i> na estrutura de rede	U-LAN	X	X		
	S-LAN			X	
	WAN				
Ferramentas para monitoramento de:	Segurança	usa	NÃO	NÃO	NÃO
		aprova qual ?	SIM	SIM	SIM
	Contabilização	usa	SIM	NÃO	SIM
		aprova qual ?		SIM	
			<i>www-stat</i>		<i>Http-analyser</i>

Legenda: U-LAN-Único segmento de LAN S-LAN-Segmentos de LAN

Os seguintes itens constituem os subsídios para as demais atividades, segundo as informações levantadas junto aos administradores de aplicações:

- A *bandwidth* utilizada por duas instituições que terão seus sistemas gerenciados, não apresenta restrições quanto ao caminho utilizado para troca de mensagens de gerenciamento, podendo ser utilizado um canal do tipo *in-band*. Por outro lado, a terceira instituição (ETFPEL) considerada no levantamento, cuja velocidade de conexão é de 28Kbps, deverá ter seu tráfego de rede monitorado, antes e após a implantação das funcionalidades de gerenciamento, para que o impacto causado pelas mensagens de gerenciamento seja mantido em níveis aceitáveis.
- O ambiente computacional (*hardware* e sistema operacional) mostrou-se heterogêneo, o que, por certo, motiva a necessidade de considerar diferentes formas de implementação de entidades *SNMP* do tipo agente.

- A adoção de diferentes implementações do *HTTP* torna necessário um estudo detalhado quanto à conformidade com a versão do protocolo que será analisada e quanto à compatibilidade para os arquivos de configuração e capacidades de *logging*.
- A existência de conteúdo com acesso restrito permitirá a definição de mecanismos para o gerenciamento de contabilização e de segurança. Para contabilização, será possível identificar consumidores/clientes e, para segurança, será efetuado o monitoramento dos processos de autenticação.
- A distribuição dos servidores em relação à estrutura da rede e o número reduzido dos sistemas que serão gerenciados, permitirão a adoção de uma arquitetura de gerenciamento centralizada.
- A inexistência de ferramentas para o monitoramento de segurança e a utilização de ferramentas para o monitoramento de contabilização, independentes e aplicadas, principalmente na apuração de dados estatísticos, justificam a realização das demais atividades previstas na proposta.

### 6.3 Análise das aplicações de rede

Para auxiliar a identificação dos requisitos e a definição dos objetos de gerenciamento, uma arquitetura genérica das aplicações de rede é apresentada na FIG. 6.2. As aplicações de rede geralmente consistem de múltiplos componentes agrupados em duas entidades de rede denominadas **cliente** e **servidor**.

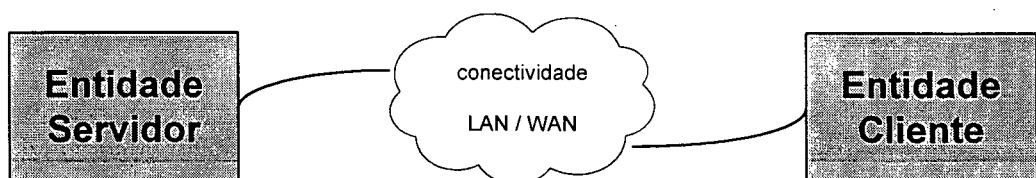


FIGURA 6.2: Arquitetura genérica das aplicações de rede

Embora este trabalho tenha sido direcionado para o gerenciamento de contabilização e de segurança de aplicações do tipo servidor, na FIG. 6.3 apresenta-se uma decomposição funcional para as duas categorias de aplicação.

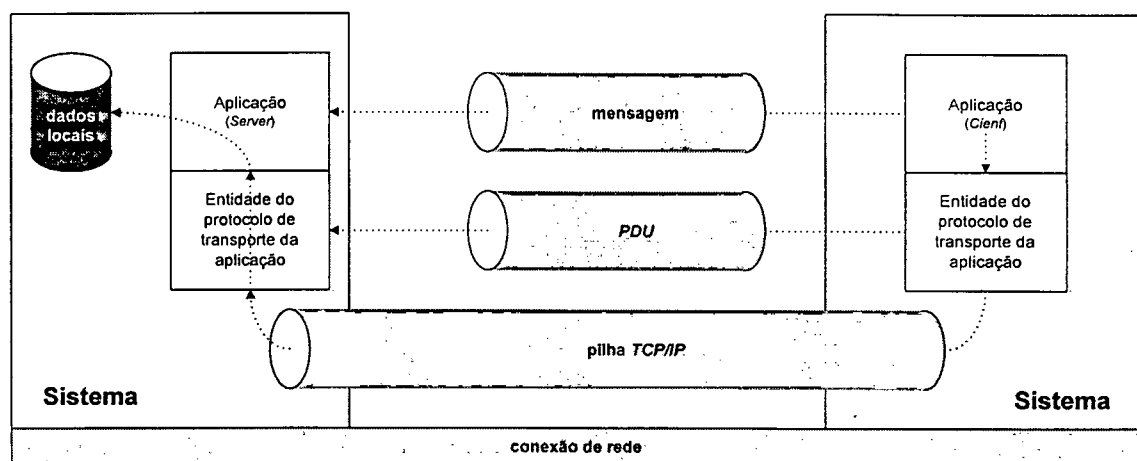


FIGURA 6.3: Decomposição funcional para as aplicações de rede *Client-Server*

No contexto desta arquitetura e decomposição funcional, podem-se enquadrar diferentes protocolos e implementações, que constituem as chamadas aplicações de rede. Para efeito de aplicação da proposta, foi escolhido o protocolo *HTTP* e considerados os aspectos comuns presentes em diferentes implementações de *HTTP Servers*.

### 6.3.1 Protocolo *HTTP*

O *HTTP* é um protocolo da camada de aplicação para distribuição e troca de informações em formato hipermídia, utilizado desde 1990. A atual especificação define o protocolo referenciado como "*HTTP/1.1*" [FGM97]. O *HTTP* é também usado como um protocolo genérico por *user agents*<sup>1</sup> e *proxies*<sup>2</sup>/*gateways*<sup>3</sup> para comunicação com

<sup>1</sup> São clientes que iniciam uma requisição. Estes são muitas vezes *browsers*, editores, aranhas (*robots*) ou outras ferramentas de usuários finais.

<sup>2</sup> Programas intermediários que atuam como servidores e clientes, com o propósito de envio de requisições em favor de outros clientes.

<sup>3</sup> Um servidor que atua como intermediário de outros servidores.

outros sistemas da *Internet*, incluindo aqueles suportados por protocolos, como: *SMTP*, *NNTP*, *FTP*, *Gopher* e *WAIS*.

O *HTTP* é um protocolo do tipo *request/response*. Um cliente envia uma requisição para o servidor na forma de um *request method*, *URI* e *protocol version*, seguida por uma mensagem *MIME-like*, contendo diversas informações. O servidor responde com uma linha de *status*, incluindo a versão do protocolo, o código de sucesso ou de erro e outras informações. A FIG. 6.4 apresenta as mensagens trocadas entre um cliente (comando *telnet* para a porta 80 do *host* 200.17.82.55) e um servidor (*Apache* versão 1.2b7).

baronesa:/home/lftm>telnet 200.17.82.55 80	⇒ conexão com o servidor
Trying...	
Connected to 200.17.82.55.	
Escape character is '^]'.	
<b>HEAD / HTTP/1.0</b>	⇒requisição do <i>header</i> do HTTP
<b>HTTP/1.1 200 OK</b>	⇒resposta do servidor com Informações do <i>header</i>
<b>Date: Thu, 10 Apr 1997 09:43:42 GMT</b>	<i>header</i>
<b>Server: Apache/1.2b7</b>	
<b>Connection: close</b>	
<b>Content-Type: text/html</b>	
Connection closed.	⇒ final da conexão

FIGURA 6.4: Troca de mensagens entre Cliente e Servidor *HTTP*

A troca de informações entre entidades do *HTTP*, na forma de envio ou recebimento de mensagens, geralmente é efetuada sobre conexões *TCP/IP*. O endereço 80 é utilizado como porta *TCP default*, o que não impossibilita a utilização de outras portas. Muitas implementações baseadas na versão anterior do protocolo (*HTTP 1.0*), utilizam uma nova conexão para cada troca de *request/response*. Na especificação do *HTTP/1.1*, uma conexão poderá ser usada para uma ou mais *request/response*.

### 6.3.1.1 Mensagens do *HTTP*

As mensagens do *HTTP* contêm requisições enviadas por clientes e respostas fornecidas por servidores. Mensagens de requisições e de respostas usam um formato genérico definido na *RFC822*. Os dois tipos de mensagens consistem de uma linha de início, um ou mais campos de cabeçalhos, uma linha em branco, dois caracteres especiais “*CRLF*”, indicando o final do cabeçalho e o corpo da mensagem. A FIG. 6.5, relaciona os componentes de uma mensagem *HTTP*.

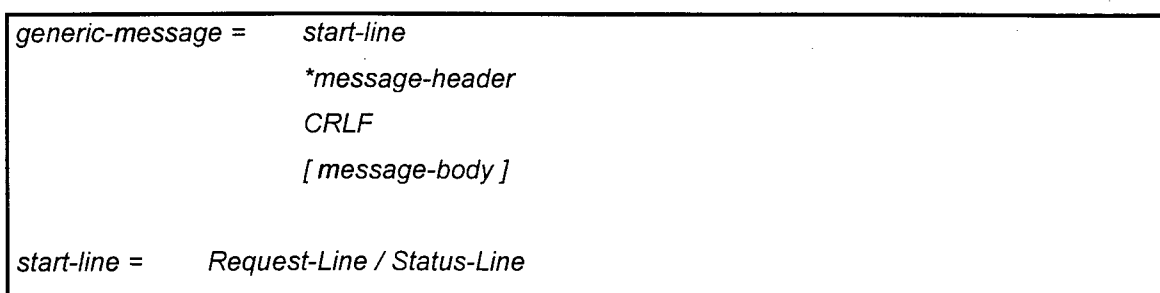


FIGURA 6.5: Componentes de uma mensagem *HTTP*

Dentre os componentes das mensagens do *HTTP*, os *headers* podem ser considerados, para efeitos de gerenciamento, uma das principais fontes de informação, principalmente pela capacidade de transferência dos campos para arquivos de *Log*. Na atual versão do *HTTP*, estão definidos os seguintes tipos de *header*: *general-header*, *request-header*, *response-header* e *entity-header*. Os valores contidos nos campos dos *headers* poderão ser considerados para atualização de uma determinada *MIB*.

### 6.3.1.2 Códigos de *status*

Outra informação que poderá ser utilizada para efeitos de gerenciamento é o código de *status*, presente na primeira linha de uma mensagem *HTTP* do tipo resposta. A TAB. 6.2, apresenta os *status* especificados na atual versão do protocolo.

Tabela 6.2: Códigos de *status* especificados na versão 1.1 do *HTTP*

<b>Código de <i>status</i></b>	<b>Significado</b>
100	<i>Continue</i>
101	<i>Switching Protocols</i>
200	<i>OK</i>
201	<i>Created</i>
202	<i>Accepted</i>
203	<i>Non-Authoritative Information</i>
204	<i>No Content</i>
205	<i>Reset Content</i>
206	<i>Partial Content</i>
301	<i>Moved Permanently</i>
302	<i>Moved Temporarily</i>
303	<i>See Ohter</i>
304	<i>Not Modified</i>
305	<i>Use Proxy</i>
400	<i>Bad Request</i>
401	<i>Unauthorized</i>
402	<i>Payment Required</i>
403	<i>Forbidden</i>
404	<i>Not Found</i>
405	<i>Method Not Allowed</i>
406	<i>Not Acceptable</i>
407	<i>Proxy Authentication Required</i>
408	<i>Request Timeout</i>
409	<i>Conflict</i>
410	<i>Gone</i>
411	<i>Length Required</i>
412	<i>Precondition Failed</i>
413	<i>Request Entity Too Large</i>
414	<i>Precondition Failed</i>
415	<i>Unsupported Media "Type"</i>
500	<i>Internal Server Erros</i>
501	<i>Not Implemented</i>
502	<i>Bad Gateway</i>
503	<i>Service Unavailable</i>
504	<i>Gateway Timeout</i>
505	<i>HTTP Version not supported</i>

O gerenciamento de contabilização poderá utilizar determinados códigos como atributo de disparo de registro para eventos que causarão a atualização dos dados de medida de uso.

Como visto anteriormente, no padrão *ISO10164-7* foram definidos tipos e causas prováveis para os alarmes de segurança. Na TAB. 6.3 apresenta-se uma possível classificação dos códigos de *status* em relação aos tipos e causas para alarmes.

TABELA 6.3: Classificação dos *status* do *HTTP* em relação aos requisitos do *OSI-RM*

Definições da <i>ISO</i> para Alarmes de Segurança		Códigos de <i>status</i> do <i>HTTP</i>										
		400	401	403	405	407	411	413	414	415	501	505
TIPOS	Violação operacional	✓			✓	✓	✓	✓		✓		✓
	Violação de serviço		✓	✓					✓		✓	
CAUSAS	Erro de procedimento	✓				✓	✓	✓		✓		✓
	Tentativa de acesso ã autorizado		✓	✓	✓				✓		✓	

No gerenciamento de segurança, os códigos de *status* podem ser vistos como notificações de um objeto gerenciando (*HTTP*), sujeitas a um processamento para detecção de eventos potenciais a serem submetidos a um discriminador de repasse de alarmes de segurança, ocasionando, em determinadas circunstâncias, o envio de Relatórios de alarme de segurança.

### 6.3.2 Capacidades de *logging* das implementações de servidores *HTTP*

A análise dos *Logs* gerados por servidores do protocolo *HTTP*, poderá fornecer subsídios para o monitoramento remoto com o uso de uma Aplicação Agente. Geralmente é implementada nos servidores a criação de um arquivo de *Log* com base no *Common Logfile Format - CLF*. Cada entrada neste tipo de arquivo de *Log* obedece ao seguinte formato:

```

remothost  user  authuser  [date]  "request"  status  bytes
remothost  -      nome do host ou número IP da origem da requisição.
user       -      nome de login remoto do usuário.
authuser   -      nome com o qual o usuário é autenticado.
date       -      data e tempo da requisição.
request    -      tipo de requisição que foi feita pela aplicação cliente.
status     -      código de status retornado para a aplicação cliente.
Bytes      -      número de bytes do conteúdo transferido para a aplicação
               cliente.
    
```

Os dados de *Log* dos servidores *HTTP* podem ser processados de forma *off-* ou *on-line* visando fornecer estatísticas de acesso aos conteúdos, permitindo a identificação de aplicações clientes utilizadas, dos domínios que originaram as requisições e dos usuários com acesso controlado pelo *HTTP Server*. Tais estatísticas poderão ser incluídas na *MIB* e geradas por demanda via Aplicação Agente. Se os requisitos de processamento permitirem, por outro lado, as estatísticas poderão ser mantidas em memória e atualizadas de forma incremental.

O monitoramento do *Log* de erros poderá apoiar o gerenciamento de segurança, haja vista que a segurança de um sistema hospedeiro (*Host*) poderá ser afetada [Web97] sempre que ocorrer um ataque que explore os *bugs* de aplicações acessadas via *Web*, como, por exemplo, as aplicações de *CGI*.

### 6.3.3 Parâmetros de configuração das implementações de servidores *HTTP*

Os parâmetros de configuração comuns a diferentes implementações do *HTTP*, poderão ser considerados na implementação de funções de gerenciamento. Programas servidores, como *CERN* e *Apache*, possuem grande compatibilidade entre as variáveis e opções de configuração.

Como forma de exemplificar a utilização destes mecanismos na implementação de funcionalidades de gerenciamento, utilizou-se o parâmetro *ErrorDocument*, presente no arquivo *srm.conf* do servidor *Apache*. Este parâmetro, também chamado de diretiva, possibilita, por exemplo, a execução de um programa, sempre que um determinado tipo de erro (código de *status*) for detectado pelo servidor. Desta forma, foi possível executar a ferramenta *snmptrap*, do pacote *CMU SNMP*, responsável pelo envio de uma notificação (*trap*) ao Gerente, sempre que ocorrer uma tentativa de acesso não-autorizado, cujo código de *status* da operação é igual a 401.



## 6.4 Análise das MIBs existentes

Uma síntese da análise das principais MIBs existentes, relevantes para um futuro padrão de MIB, destinado ao gerenciamento de servidores HTTP, é descrita nesta seção. Como critérios de análise, procurou-se verificar a aplicabilidade das linhas de MIBs existentes, identificar as vantagens e desvantagens oferecidas nas propostas já estabelecidas, como também evitar a sobreposição de objetos gerenciados, garantindo, dessa forma, que o módulo de MIB definido contenha objetos necessários para o efetivo gerenciamento deste tipo de aplicação.

### 6.4.1 Propostas de objetos gerenciáveis para o HTTP

Pesquisadores da *University of Lisboa* [PiVe95] desenvolveram uma MIB com o propósito de possibilitar o gerenciamento remoto de servidores WWW, usando o protocolo SNMP. A MIB proposta consiste de três grupos de objetos: *general group*, *client group* e *document group*.

O *General group* contém variáveis que fornecem informações das últimas requisições ao servidor WWW e dados de contabilização. O *Client group* possui variáveis do tipo *counters*, resultantes do monitoramento da origem das requisições enviadas por clientes WWW. Finalmente, o *Document group* contém variáveis que indicam estatísticas de acesso aos documentos controlados pelo servidor WWW.

Outra instituição que dedicou esforços para a definição de uma MIB para o HTTP, foi o *Centre of Earth Observation - CEO* da Europa, que escolheu a WWW como mecanismo preferido para disponibilizar os dados das observações realizadas. Durante o projeto três módulos de MIB foram definidos para o gerenciamento dos serviços da WWW. Dentre os módulos definidos, encontra-se o *HTTP MIB* [HHP96], que pode ser quase totalmente implementado via acesso aos registros do arquivo de log gerado pelo servidor. O *CEO HTTP MIB* é um módulo que contém detalhadas informações de

gerenciamento, estruturadas nos seguintes grupos de objetos: *httpSystem*, *httpStatistics* e *httpTimeOuts*.

Uma outra proposta [KHJ96], desenvolvida em nível da *IETF*, denominada *Definitions of Managed Objects for WWW Servers*, estabeleceu um amplo conjunto de objetos gerenciáveis para servidores *WWW*, ampliando o número de grupos definidos no módulo *CEO HTTP MIB*. O enfoque da base de informações definida é para o gerenciamento de entidades *WWW*. Estas entidades são formadas por servidores, clientes e procuradores que se comunicam usando o *HTTP* e o *Internet Cache Protocol - ICP*. A comunicação entre as entidades é efetuada por meio de mensagens, divididas para efeitos de estatísticas em *bytes* de cabeçalho e *bytes* de dados. A *MIB* proposta possui as seguintes partes:

- Extensões para *Network Services Management*
- Extensões para *sysAppl*
- Relatório de Erro
- Estatísticas e armazenamento de documentos
- Relacionamento para outras *MIB's*

Os objetos (variáveis) da *MIB* estão organizados nos seguintes grupos:

*Grupo System*            O grupo *System* possui uma tabela (*wwwEntityTable*) com objetos que descrevem todas as entidades, cujos processos de execução estão ativados no *host*. O conjunto de variáveis definidas permite gerenciar entidades do tipo servidor, com domínios virtuais.

*Grupo Statistics*        O grupo *Statistics* está organizado em três tabelas que contêm informações de gerenciamento de rede sobre o tráfego recebido ou transmitido pela entidade presente no sistema. As tabelas do grupo *Statistics* são:

*wwwSummaryTable* contém um conjunto de variáveis que representam a totalização do tráfego de rede recebido ou enviado pela entidade.

*wwwRequestTable* tabela contendo estatísticas detalhadas sobre as requisições enviadas/recebidas pelas entidades presentes no sistema.

*wwwResponseTable* tabela contendo estatísticas detalhadas para as respostas enviadas/recebidas pelas entidades presentes no sistema.

*Grupo Document* Contém informações sobre os conteúdos acessados, organizadas nas seguintes tabelas:

*wwwDocNameTable* fornece um meio para a rápida indexação da tabela de documentos.

*wwwDocTable* tabela contendo os atributos de um determinado documento.

*wwwDocFilterTable* utilizada para que o gerente possa configurar quais documentos são armazenados nas outras duas tabelas.

*Grupo Error* - NÃO DEFINIDO.

*Grupo Application* - Define os atributos que estende o *sysAppl/applMIB* para *WWW*.

*Grupo Relation* - Define como a *MIB* se relaciona com outras *MIBs*.

Relacionadas as principais características das *MIBs* propostas para o gerenciamento de aplicações pertencentes a *World Wide Web*, pode-se notar que a *MIB* proposta para o gerenciamento do *HTTP*, é especialmente destinada para gerenciar entidades do tipo servidor *Web*, com possibilidades de ser utilizada no gerenciamento de clientes *Web*. Por outro lado, o conjunto de objetos gerenciáveis definido para servidores *WWW*, possibilitará o gerenciamento de todas as entidades *WWW*, atualmente implementadas, formadas por: *server*, *client*, *proxy* e *caching proxy*. A TAB. 6.4 apresenta os grupos e o número de objetos definidos em cada *MIB*.

TABELA 6.4: Grupos e números de objetos da *HTTP MIB* e da *WWW MIB*

GRUPOS	Número de Objetos	
	<i>HTTP MIB</i>	<i>WWW MIB</i>
<i>System</i>	13	9
<i>Statistics</i>	23	32
<i>TimeOuts</i>	4	ñ definido
<i>Document</i>	ñ definido	19
<i>Error</i>	ñ definido	0
<i>Application</i>	ñ definido	6
<i>Relation</i>	ñ definido	7

Os objetos incluídos na *WWW MIB* possibilitam, principalmente, o gerenciamento de aspectos relacionados ao funcionamento das entidades *WWW*, caracterizando a necessidade de complementação com objetos que possibilitem a identificação dos usuários com a devida contabilização de uso e o monitoramento das tentativas de acessos não-autorizadas, visando o gerenciamento de segurança.

Fabricantes de *Web servers*, como a *Netscape* e a *Microsoft*, oferecem funcionalidades de gerenciamento nos produtos *Netscape Enterprise Server 2.0* e *Microsoft Internet Information Server*, respectivamente.

O produto da *Netscape* possui funcionalidades de gerenciamento, em conformidade com as versões 1 e 2 do *SNMP*, fornecendo capacidades de

monitoramento remoto. A *MIB* definida possui um prefixo de identificador de objeto com valor 1.3.6.1.1.1450.

Acessando os objetos da *MIB*, é possível obter informações administrativas e monitorar o servidor da *Netscape* em tempo real. As informações de gerenciamento estão em três tabelas: *httpEntityTable*, *httpStatisticsTable* e uma outra tabela, contendo estatísticas específicas relacionadas ao funcionamento do protocolo da *Netscape* no sistema operacional *Windows NT*.

A *MIB* definida pela *Netscape* fornece diversas informações estáticas e estatísticas sobre o *Web server*. Todas as variáveis da *MIB* possuem como *max-access* o tipo *read-only* e a Aplicação Gerente não pode configurar o servidor via *SNMP*. As variáveis estáticas são semelhantes às que foram propostas e descritas anteriormente.

A *Microsoft*, por sua vez, definiu na *MIB* do seu servidor, um rico conjunto de métricas para o monitoramento de desempenho via *SNMP*. A *MIB* implementada no *Microsoft Internet Information Server* inclui objetos gerenciados para os serviços de informações como *FTP*, *Gopher* e *HTTP*, e para o monitoramento de *cache* (*Cache Hits*, *Cache Hits %*, *Cache Misses*, *Cache Size*, *Cache Used*, *Cache File Handles*, etc.).

No caso específico de servidores *HTTP*, são mantidas informações de gerenciamento contendo a taxa de *bytes* enviados e recebidos, o número total de requisições executadas via aplicações de *CGI*, desde que o serviço foi inicializado, o corrente número de conexões atendidas pelo servidor, o número de requisições não atendidas em função de o servidor não ter encontrado o documento e o número de requisições para cada método definido no *HTTP*.

#### **6.4.2 Aplicabilidade das *MIBs* padronizadas**

Na *RFC2039* [Kal96], é definida a aplicabilidade das *MIBs* padronizadas para o gerenciamento do *HTTP*. Segundo *Kalbfleisch*, servidores *Web* podem ser vistos com

diferentes perspectivas, denominadas de Modelo Operacional e Modelo de Serviços, quando considerados aspectos de gerenciamento.

No Modelo Operacional, o servidor *Web* é visto como um dispositivo de rede com *hardware*, disco, sistema operacional e o software do *daemon Web*. Este modelo representa os atuais recursos que compõem o equipamento utilizado, podendo ser gerenciado com base na perspectiva de utilização de recursos. No Modelo de Serviços, o servidor *Web* é visto como uma “caixa preta” que simplesmente manuseia e responde às requisições de clientes localizados na *Web*. A TAB. 6.5 relaciona os requisitos de cada modelo.

TABELA 6.5: Requisitos de gerência para os Modelos Operacional e de Serviços

Modelo Operacional	Modelo de Serviços
<ul style="list-style-type: none"> <li>• Monitoramento de <i>Host</i></li> <li>• Dependência entre aplicações</li> <li>• Relatórios de erro</li> <li>• Planejamento</li> <li>• Excesso de <i>logs</i></li> </ul>	<ul style="list-style-type: none"> <li>• Recuperação de serviços</li> <li>• Gerenciamento de documentos</li> <li>• Configuração do servidor</li> <li>• Controle do servidor</li> <li>• Qualidade dos serviços</li> </ul>

Os dois modelos se complementam, fornecendo distintas informações sobre o servidor *Web* que, em geral, é formado por componentes, como:

- estações de propósito geral, executando um sistema operacional;
- *software* servidor de *HTTP*, respondendo a requisições via *Web*;
- suporte a várias rotinas, como programas *CGI* ou aplicações (com *SGBD*), usadas para acessar informações;
- armazenamento de documentos (*HTML*, *GIF*<sup>4</sup>, *JPEG*<sup>5</sup>, *CLASS*<sup>6</sup> e outros) em um ou mais dispositivos.

<sup>4</sup> *Graphics Interchange Format* - é o mais comum formato de imagem encontrado na Internet.

<sup>5</sup> *Joint Photographic Experts Group* - padrão desenvolvido para a compressão de imagens.

<sup>6</sup> Extensão para arquivos, contendo aplicações desenvolvidas com a linguagem *JAVA*.

O controle e monitoramento dos componentes citados poderão ser parcialmente efetuados com o uso de *MIBs* padronizadas, como: *RFC1213*, 1514 e 1565. A TAB. 6.6 apresenta de forma resumida a aplicação e os grupos e/ou atributos relacionados em cada *RFC*. Além dos padrões já estabelecidos, propostas de novas *MIBs* vêm sendo efetuadas, como, por exemplo: *Application Management MIB* e *Definitions of Managed Objetos for Applications*.

TABELA 6.6: Identificação de informações gerenciais presentes em *MIBs* padronizadas

<b>RFC</b>	<b>APLICAÇÃO</b>	<b>GRUPOS/ATRIBUTOS</b>
1213 <i>MIB II</i>	Objetos gerenciais contidos em dispositivos que suportam o conjunto de protocolos <i>TCP/IP</i> .	Atributos dos grupos: <i>System, Interfaces, ICMP, UDP, TCP e SNMP.</i>
1514 <i>Host Resources MIB</i>	Informações gerais sobre o <i>status</i> do <i>Host</i> (dispositivos e aplicações).	Atributos dos grupos: <i>System, Storage, Device e Installed Software.</i>
1565 <i>Network Services Monitoring MIB</i>	Estatísticas e <i>status</i> de processos ativos que fornecem informações sobre os serviços de rede.	Todos os atributos são relevantes.

A existência de inúmeros atributos, capazes de atender parcialmente os requisitos do Modelo Operacional (*RFC1213*, 1514 e proposta para *Application Management MIB*), bem como os requisitos do Modelo de Serviços (*RFC1565*), não são suficientes, sendo necessário definir um conjunto adicional de variáveis, visando contemplar aspectos, como [Kal96]:

- Definição de relacionamento entre as *MIBs Network Services Monitoring e Application*;
- Controle e monitoramento de falhas das diferentes operações de servidores *Web*;
- Monitoramento de acessos com: estatísticas de acesso, permissões de acesso e relatórios de violações de acesso.

### 6.4.3 Sobreposição de objetos

Um levantamento efetuado por *Harrie Hazewinkel* [Haz97], tomando como base diversas *MIBs* relacionadas com o gerenciamento da camada de aplicação, identificou a sobreposição ou duplicação de objetos e apontou problemas com o relacionamento das *MIBs*. O cruzamento dos objetos definidos na *WWW-MIB* [CHJ97], com os objetos das demais *MIBs* analisadas, resultou na identificação das seguintes sobreposições (FIG. 6.6, 6.7 e 6.8):

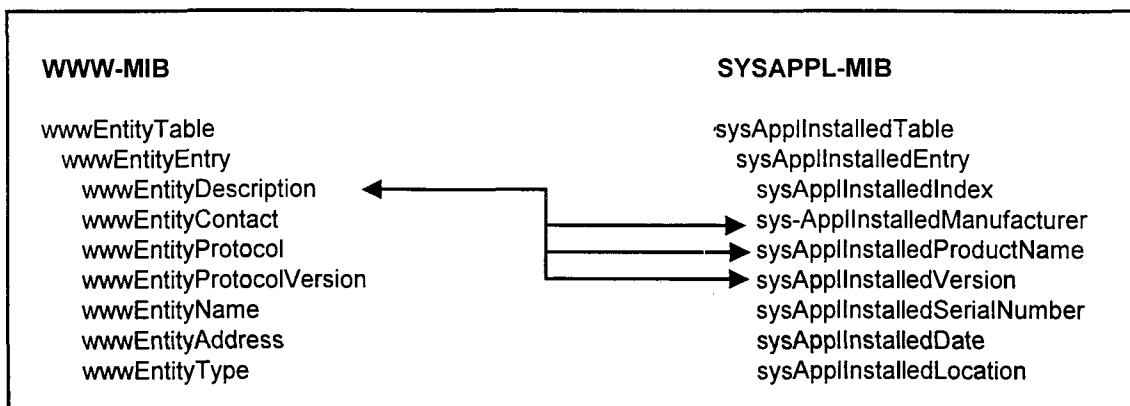


FIGURA 6.6: Sobreposição de objetos entre a *WWW-MIB* e *SYSAPPL-MIB*  
Fonte: [Haz97]



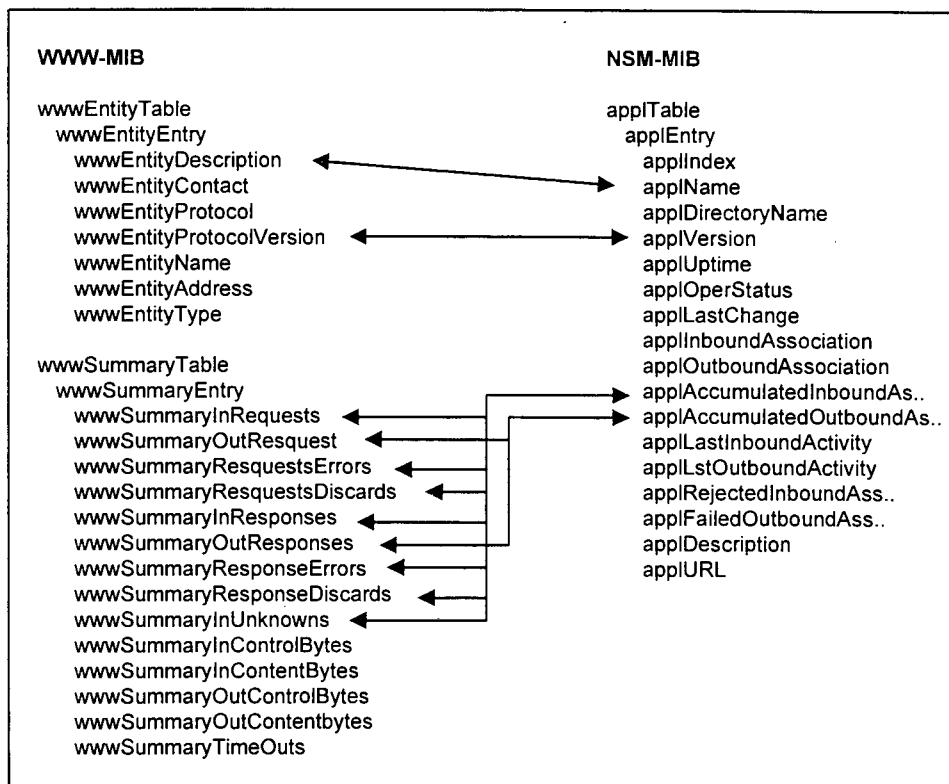


FIGURA 6.7: Sobreposição de objetos entre *WWW-MIB* e *NSM-MIB*  
 Fonte: [Haz97]

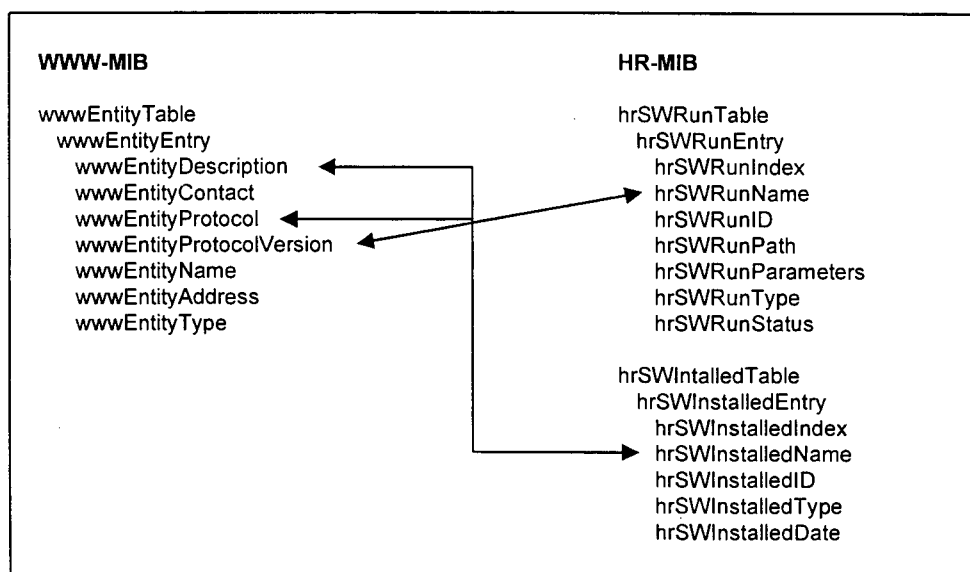


FIGURA 6.8: Sobreposição de objetos entre *WWW-MIB* e *HR-MIB*  
 Fonte: [Haz97]

## 6.5 Especificação funcional para ferramentas de gerenciamento de aplicações

Os resultados produzidos nas atividades de análise (2, 3 e 4) serviram de base para a especificação da abordagem de informações e de funções de gerenciamento descritas nos próximos itens.

### 6.5.1 Abordagens e arquitetura de gerenciamento

Considerando-se o escopo definido para o Sistema e a abrangência do domínio de gerenciamento, optou-se em implementar uma arquitetura de gerenciamento centralizada. Entende-se como Sistema, entidades *SNMP* implementadas na forma de Aplicações Gerente e Agente. O domínio de gerenciamento, por sua vez, compreende a distribuição geográfica e os tipos de aplicações que serão gerenciadas. A TAB. 6.7 apresenta as capacidades e limitações decorrentes da arquitetura de gerenciamento selecionada.

TABELA: 6.7: Capacidades e limitações decorrentes da escolha da arquitetura

Itens identificados	Capacidades / limitações
Componentes suportados	Gerente, Agentes e <i>MIB</i>
Desenvolvimento	Fácil
Manutenção	Fácil
Escalabilidade	Não permite
Gerenciamento por domínio	Não permite

Os aspectos relacionados à escalabilidade e ao gerenciamento por domínio, embora não suportados em decorrência da arquitetura que foi escolhida, deverão ser previstos na especificação de sistemas reais, visando atender os requisitos de gerenciamento, impostos pela característica distribuída e os relacionamentos, presentes em boa parte das aplicações de rede.

## 6.5.2 Informações de gerenciamento

Na especificação das informações de gerenciamento (ANEXO A), procurou-se atender em parte os requisitos definidos no item 5.5.2, em conformidade com os propósitos estabelecidos para o protótipo de sistema de gerenciamento implementado. Dessa forma, foi especificado um conjunto de informações relacionadas ao gerenciamento parcial de contabilização e de segurança.

Foi necessária a definição de um novo módulo de *MIB*, em contraposição ao aproveitamento de um dos módulos já propostos e analisados no item 6.4, haja vista que o enfoque dos objetos gerenciados, que foram definidos nos documentos analisados, destina-se principalmente ao gerenciamento de configuração (incluindo aplicações dependentes e associações entre aplicações), de falhas (incluindo informações de *status*, tal como: *up*, *down*) e de desempenho (incluindo indicadores e métricas de utilização de recursos) de aplicações distribuídas.

A exemplo das *MIBs* que foram analisadas e obedecendo às regras estabelecidas pelo *IETF* para o gerenciamento de redes, a *MIB* definida no protótipo está organizada em três grupos: *naSystem*, *naStatistics* e *naMonitor*.

O grupo *naSystem* mantém informações sobre as aplicações gerenciadas na tabela *naEntityTable* e sobre os recursos associados como a identificação do *Host* e *bandwidth* da conexão de *WAN*, nos objetos escalares *naSysHost* e *naSysWanBandwidth*, respectivamente.

Informações estatísticas sobre as requisições atendidas e autenticações efetuadas pela aplicação gerenciada, foram definidas no grupo *naStatistics*, cujos objetos estão estruturados na tabela *naStatTable*.

Por último, foi definido o grupo *naMonitor*, contendo objetos de monitoramento para as tabelas *naMonDocumTable* e *naMonRecentUserTable*. A tabela *naMonDocumTable* contém informações referentes aos documentos acessados. Na

tabela *naMonRecentUserTable* são mantidas informações dos usuários que efetuaram conexão recente com a aplicação gerenciada.

Como se trata de uma *MIB* em nível experimental, futuramente, parte do conjunto de objetos definidos poderá ser substituída pela adoção da *WWW Service MIB* [HKS97], em fase de padronização junto ao *IETF*.

### 6.5.3 Funções de gerenciamento

Após análise dos dados levantados, ficou constatado que os administradores de aplicações necessitam, dentre outras, de funções de gerenciamento destinadas à contabilização e segurança. No protótipo desenvolvido, o atendimento de tais requisitos foi efetuado com a implantação de processos destinados a:

- Identificação das autenticações de usuários com ou sem sucesso
- Processamento de dados estatísticos com base no monitoramento de *Logs*

Durante a implementação, foi necessário utilizar *MIB Compilers* e optar pela construção de um protótipo de sistema baseado em tecnologias abertas, como, por exemplo, pacotes de gerenciamento *SNMP* de domínio público, sistemas gerenciadores de banco de dados e outros utilitários descritos com mais detalhes no item seguinte.

## 6.6 Implementação de aplicações de gerenciamento

A inexistência de um sistema de gerenciamento no ambiente escolhido para teste, motivou a implementação de um protótipo contendo as duas categorias de aplicações de gerenciamento, denominadas Aplicação Gerente e Aplicação Agente.

A FIG. 6.9 apresenta o nível abstrato do Sistema, com o uso da *Specification and Description Language* (sintaxe gráfica). Na especificação, além da declaração dos sinais, aparecem os blocos *Web Browser*, *Web Server* e *HTTP Server*. O *Web Browser* serve como interface de usuário, que juntamente com o *Web Server* desempenha o papel de estação de gerenciamento e o *HTTP Server* representa o dispositivo gerenciado.

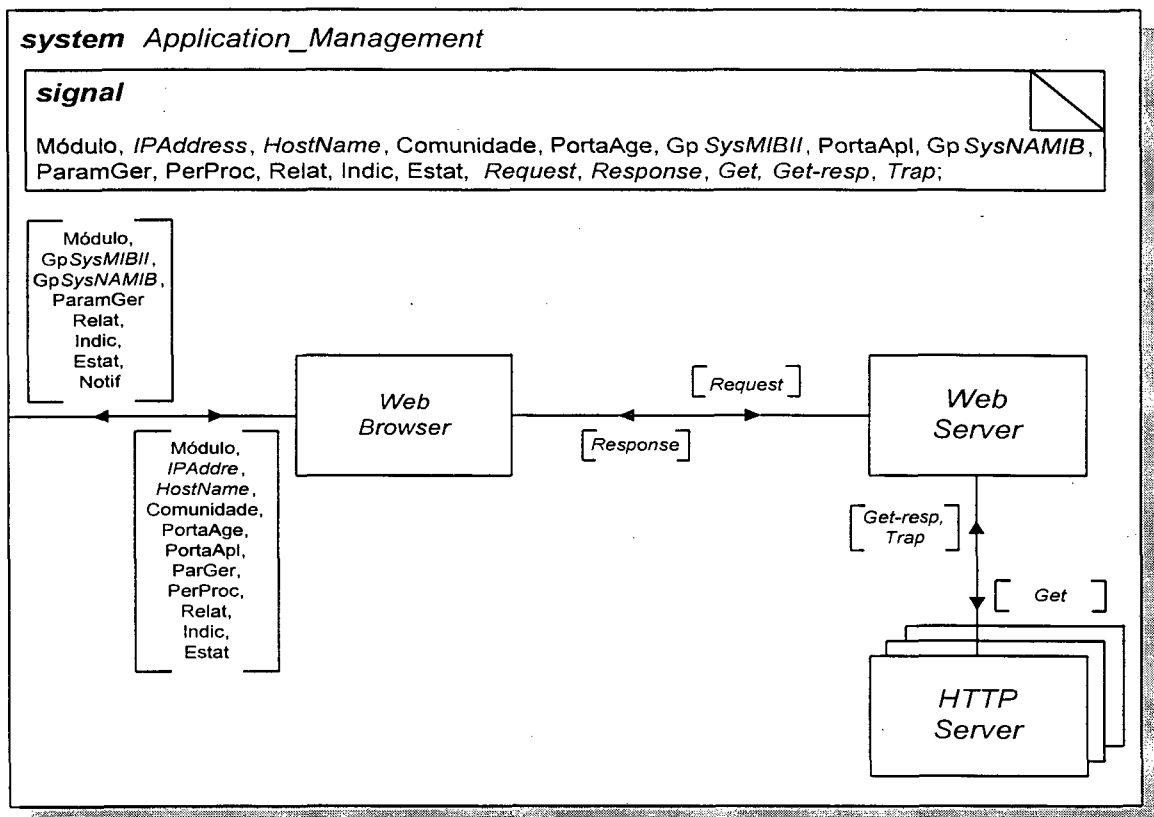


FIGURA 6.9: Especificação do Sistema em *SDL* (nível abstrato)

Como local para teste da aplicação desenvolvida, foi escolhido o ponto de presença da Rede Estadual de Informações em Ciência e Tecnologia (Rede TCHÊ), instalado na Universidade Católica de Pelotas - UCPel (FIG. 6.10), o qual, além de fornecer conectividade para a própria UCPel, é responsável pela interconexão de outras instituições de ensino. Dessa forma, a Universidade Federal de Pelotas, Escola Técnica Federal de Pelotas e a própria UCPel foram utilizadas como ambiente para validação da proposta.

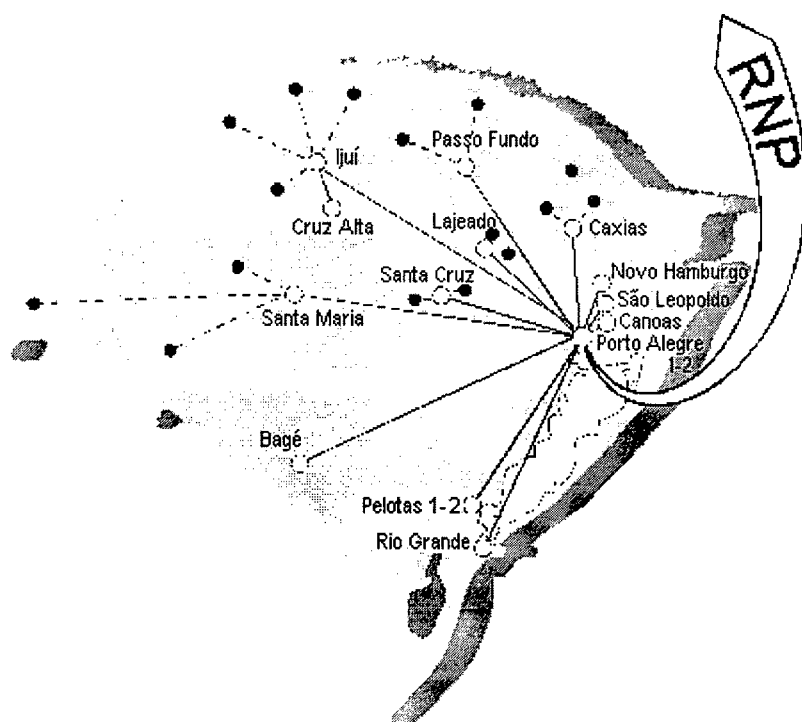


FIGURA 6.10: *Backbone* da Rede TCHÊ  
Fonte: [<http://www.sct.rs.gov.br/programa/tche.html>]

### 6.6.1 Implementação de Aplicações Gerentes

Por tratar-se de um protótipo, utilizou-se como estação de gerenciamento um microcomputador com processador *Intel* de 166 MHz e o Sistema Operacional *Linux*. Nesse *Host*, foram implementados todos os componentes (exceto a *MIB*) utilizados na construção de Aplicações Gerentes, conforme foi descrito no Capítulo 3.

O pacote de gerenciamento desenvolvido pela *Carnegie Mellon University*, denominado *CMU-SNMP* e portado para o sistema operacional *Linux* [ScSc97], foi utilizado na implementação da Aplicação Gerente.

Do pacote *CMU-SNMP*, foram utilizados na implementação desta categoria de aplicação os utilitários *snmpget*, *snmpwalk* e *snmptrapd*, os quais desempenham o papel de operações de gerenciamento *SNMP*.

Como na Aplicação Gerente foi implementada uma entidade *SNMP*, gerente de nível superior, não foi necessário especificar uma *MIB*. Tal procedimento seria necessário no caso de tratar-se de um gerente de nível médio (entidade *SNMP* gerente-agente).

Para implementar o Banco de Dados, foram utilizados o servidor *MySQL* [TcX97] e a ferramenta *XMySQLAdmin*, desenvolvida por *Gilbert Therrien* [Xmy97], conforme apresentado na FIG. 6.11.

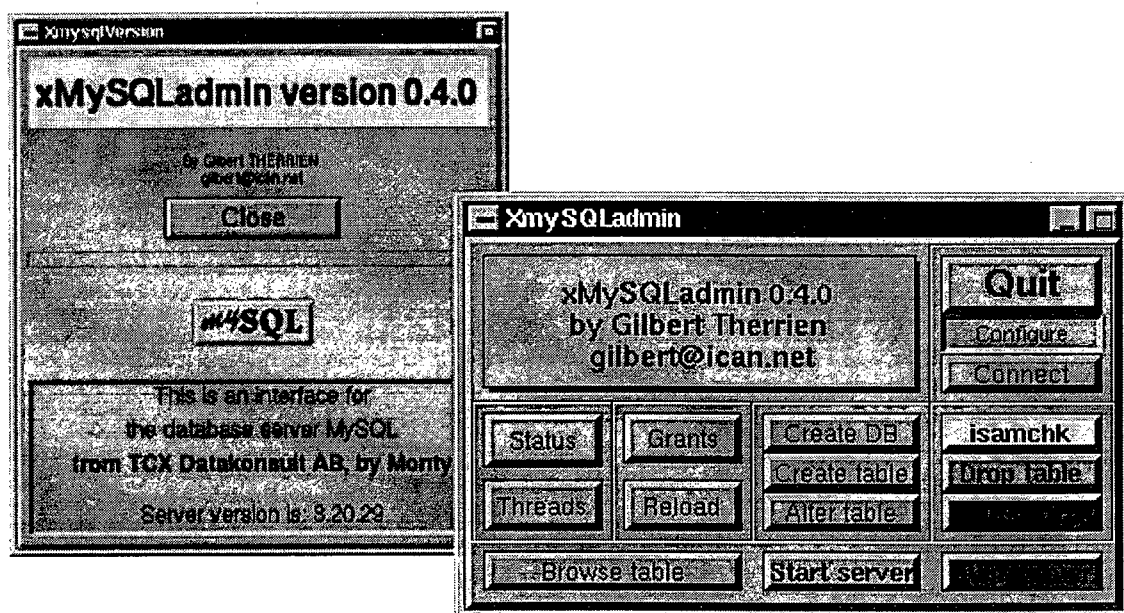


FIGURA 6.11: Ferramenta utilizada para construção do Banco de Dados

A modelagem de dados foi desenvolvida sob o enfoque da abordagem **entidade-relacionamento** [Cou97] e resultou na definição de um Banco de Dados (geraplic), contendo as seguintes entidades (tabelas):

- SISTEMA (sis\_ger) utilizada para o cadastramento dos sistemas (*hosts*) que hospedam aplicações que serão gerenciadas.

```
Database: geraplic Table: sis_ger Rows: 0
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null  | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ipaddress  | char(35)  |       | PRI  |         |       |
| email_adm  | char(35)  | YES   |      |         |       |
+-----+-----+-----+-----+-----+-----+
```

- APLICAÇÃO (apl\_ger) utilizada para o cadastramento das aplicações que serão gerenciadas.

```
Database: geraplic Table: apl_ger Rows: 0
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null  | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| porta_apl  | int(5)    |       | PRI  | 0       |       |
| nome_soft  | char(20)  | YES   |      |         |       |
| ver_prot   | char(10)  | YES   |      |         |       |
| ver_impl   | char(10)  | YES   |      |         |       |
| oid_tab    | char(20)  | YES   |      |         |       |
+-----+-----+-----+-----+-----+-----+
```

- AGENTE (age\_ger) utilizada para o cadastramento dos agentes que serão implementados no gerenciamento.

```
Database: geraplic Table: age_ger Rows: 0
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null  | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| porta_age  | int(5)    |       | PRI  | 0       |       |
| ipaddress  | char(35)  |       | PRI  |         |       |
| comunidade | char(15)  | YES   |      |         |       |
+-----+-----+-----+-----+-----+-----+
```

- USUÁRIO (ace\_usr) utilizada para armazenar os acessos efetuados por usuários para propósitos de contabilização. Os usuários serão identificados pelo *username* ou pelo domínio.



```
Database: geraplic Table: ace_usr Rows: 0
```

Field	Type	Null	Key	Default	Extra
nome_usr	char(15)		PRI		
acessos	int(10)	YES			
bytes	int(10)	YES			
ult_acesso	int(10)	YES			

- INDICADOR (ind\_conf) utilizada para armazenar o Indicador de Confiabilidade das aplicações gerenciadas.

```
Database: geraplic Table: ind_conf Rows: 0
```

Field	Type	Null	Key	Default	Extra
tempo	datetime		PRI		
porta_apl	int(5)				
ipaddress	char(35)				
requisicoes	int(10)	YES			
req_err_cli	int(10)	YES			
req_err_ser	int(10)	YES			

- RELATÓRIO DE ALARME (rel\_ala\_seg) utilizada para armazenar os alarmes de segurança enviados pelos agentes via utilitário *snmptrap*.

```
Database: geraplic Table: rel_ala_seg Rows: 0
```

Field	Type	Null	Key	Default	Extra
num_ordem	int(10)		PRI	0	
status_http	int(4)			0	
tempo_ocorr	datetime	YES			

- DESCRIÇÃO DE ALARME (des\_ala\_seg) utilizada para cadastrar a descrição dos alarmes de segurança, relacionados aos códigos de *status* do *HTTP*.

```
Database: geraplic Table: des_ala_seg Rows: 0
```

Field	Type	Null	Key	Default	Extra
status_http	int(4)		PRI	0	
tipo_alarme	varchar(20)	YES			
causa_alarme	varchar(30)	YES			
descr_alarme	varchar(50)	YES			

Um diagrama **entidade-relacionamento**, segundo a abordagem de *Peter P. Chen*, é apresentado na FIG. 6.12. Como pode ser observado, o modelo possui relacionamento entre objetos de diferentes tipos, com grau de associação **muitos para muitos (M:N)** e **um para muitos (1:N)**, além do recurso de agregação.

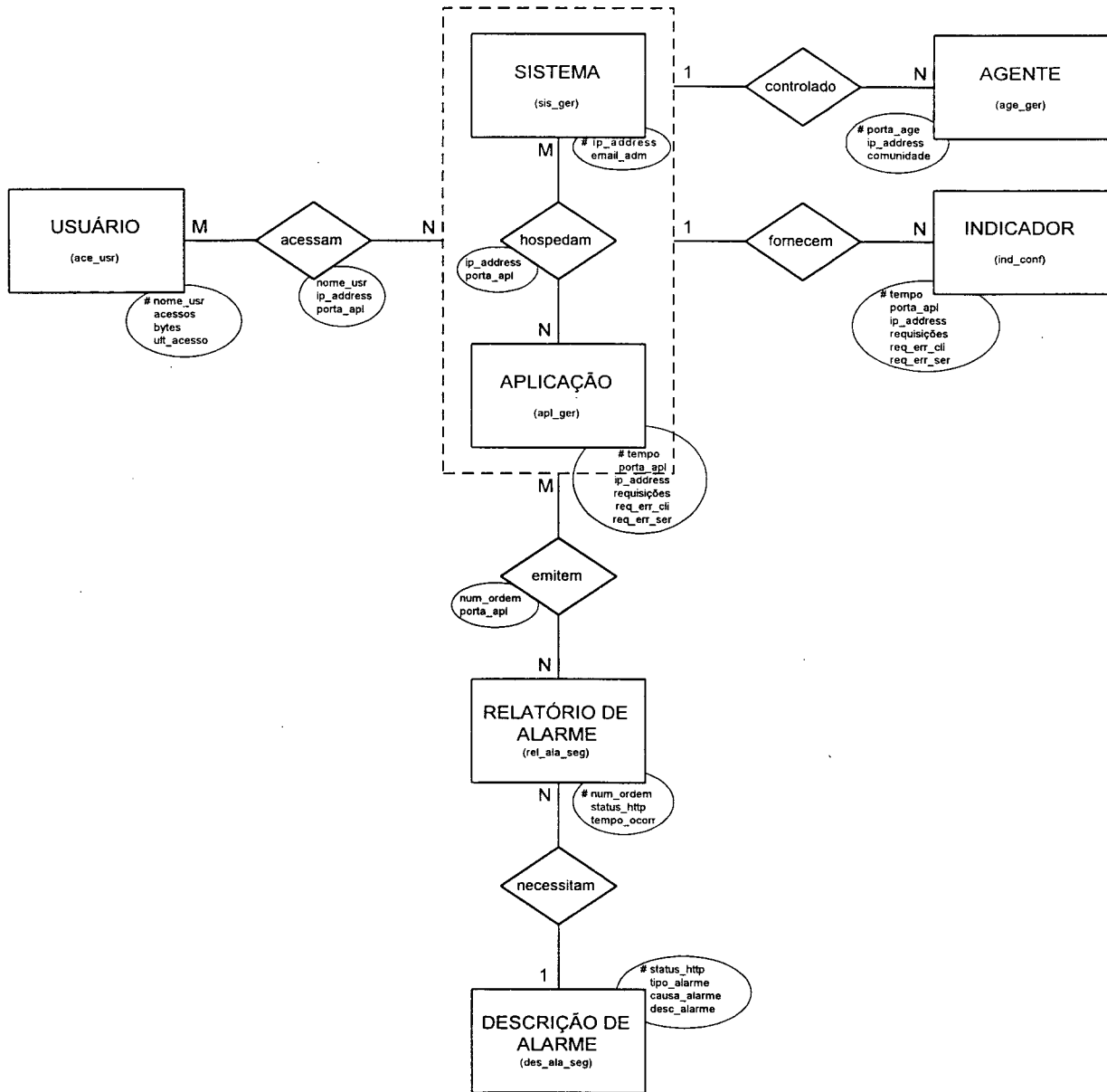


FIGURA 6.12: Diagrama de **entidade-relacionamento** do Banco de Dados

Na implementação das aplicações e operações de gerenciamento, foram utilizadas as linguagens *Perl* e *C*. A escolha dessas linguagens justifica-se pela existência de códigos fontes que serviram de base ou foram adaptados. Dentre as aplicações e operações de gerenciamento implementadas, foram desenvolvidos módulos de código para:

- Efetuar o tratamento dos dados enviados pelos usuários via formulário codificado em *HTML*, destinado à inserção ou extração de informações do banco de dados. Programa **sga\_mbd.pl**, desenvolvido em *Perl*.
- Efetuar o tratamento dos dados enviados pelos usuários via formulário codificado em *HTML*, destinado à recuperação de informações de gerenciamento diretamente da *MIB* da Aplicação Agente. Programa **sga\_aim.c**, desenvolvido em *C*.
- Formatar em modo textual e/ou gráfico e direcionadas à *Web Browsers*, as informações de gerenciamento solicitadas. Programas **sga\_fiw.pl** e **.c**, desenvolvidos em *Perl* e *C*, respectivamente.
- Armazenar no banco de dados as informações de gerenciamento não solicitadas, geradas por mensagens de notificações enviadas pelas Aplicações Agente. Programa **sga.abd.c**, desenvolvido em *C*.
- Formatar as informações de gerenciamento não solicitadas, geradas por mensagens de notificações enviadas pelas Aplicações Agente. Programa **sga\_fig.c**, desenvolvido em *C*.

Na interface de usuário que foi implementada, procurou-se explorar o uso dos recursos de *Web*, visando inserir no sistema as funcionalidades e recursividades dos *Web Servers* e *Web Browsers*. Dentre os recursos utilizados, encontram-se: *HTML* (*frames, forms, images, refresh, ...*), *JavaScript* e *CGI*. A FIG. 6.13 apresenta a interface de abertura do protótipo de sistema de gerenciamento desenvolvido.

No ANEXO B são apresentadas as principais interfaces de entrada e saída de dados, seguidas de uma breve descrição dos componentes e de suas funcionalidades.

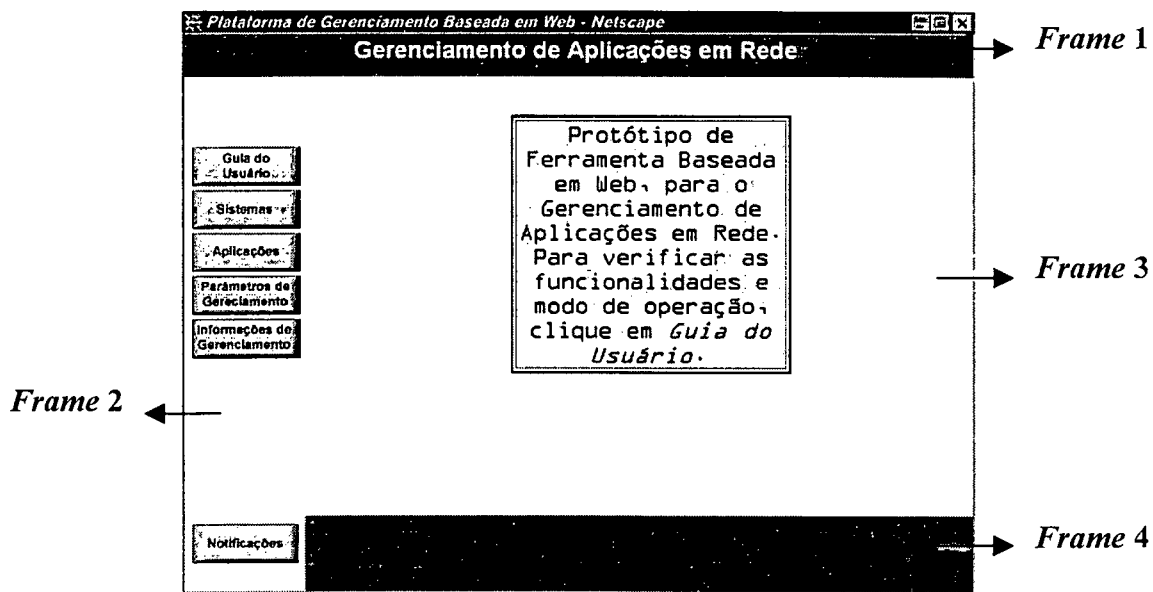


FIGURA 6.13: Interface de abertura do protótipo de sistema

A interface do sistema foi dividida em quatro *frames*. O **Frame 1** foi reservado para descrever o nome do protótipo, sendo preenchido com conteúdo estático. No **Frame 2**, foram incluídos os *links* com o uso de figuras em formato de botões, permitindo acesso aos módulos implementados. A principal área da interface do sistema foi construída no **Frame 3**, onde é efetuada a entrada e saída de dados. Por último, o **Frame 4** foi reservado para a exibição de notificações (*Traps*) enviadas pelas Aplicações Agentes.

### 6.6.2 Implementação de Aplicações Agentes

Em decorrência da heterogeneidade dos sistemas gerenciados (*hardware* e sistema operacional) foi implementado um agente em ambiente *PC - Linux* e *OSF/1*, com estrutura monolítica via *API* do *CMU-SNMP* [Gas96] e um subagente em ambiente *Workstation - ALX*, com estrutura extensível com o uso do protocolo *SMUX* [CLW96].

A inexistência de caminhos de rede exclusivos, somente para o tráfego das mensagens de gerenciamento e o baixo consumo de banda observado durante a utilização do protótipo, permitiram que um canal de gerenciamento do tipo *in-band* fosse utilizado.

Parte do código necessário à implementação das áreas funcionais de um agente, conforme foi descrito no capítulo 3, já estava disponível no pacote de *software CMU-SNMP* e no agente e subagente disponibilizado no sistema operacional *ALX*.

O código que foi desenvolvido para a Aplicação Agente, destinou-se principalmente à implementação das rotinas de instrumentação das variáveis do módulo de *MIB* especificado e apresentado no ANEXO A. Uma abordagem completa deverá prever, ainda, a instrumentação do módulo *APPLICATION-MIB*, especificado na *RFC1565* [KiFr94].

# Capítulo 7

## Conclusões e trabalhos futuros

Neste capítulo, são apresentadas algumas considerações gerais sobre o trabalho, destacada as contribuições e as dificuldades encontradas ao longo do desenvolvimento, além de serem sugeridos trabalhos futuros.

### 7.1 Considerações e contribuições

O gerenciamento de aplicações em rede é um tema relativamente novo para usuários de redes de computadores, e ainda em estudo pela comunidade científica, organismos de pesquisa e de padronização, e por desenvolvedores de *software*.

Com a crescente utilização de aplicações em rede, principalmente as relacionadas aos serviços oferecidos pela *Internet*, o gerenciamento de tais entidades tornou-se indispensável. A complexidade e a natureza distribuída das aplicações propiciam a ocorrência de falhas que degradam a qualidade dos serviços oferecidos aos usuários. Experiências demonstram que o mau funcionamento das aplicações pode ocasionar riscos e custos elevados. O desenvolvimento de ferramentas de gerenciamento poderá assegurar uma certa qualidade de serviço aos usuários, proporcionando-lhes maior produtividade e satisfação.

Embora, até o presente momento, poucos desenvolvedores estejam incluindo capacidades de gerenciamento em suas aplicações, a definição de um modelo que considere o aproveitamento de tais mecanismos, poderá incentivá-los para que adicionem, nos produtos, funcionalidades de uma Aplicação Agente, em conformidade com as abordagens/modelos de gerenciamento padronizados.

Neste trabalho, foram apresentados os requisitos estipulados no *OSI - RM* para o gerenciamento de contabilização e de segurança, de forma a complementar os trabalhos desenvolvidos pelo *Application MIB Working Group* da *IETF*, dedicados prioritariamente ao gerenciamento de configuração, falhas e desempenho.

A apresentação detalhada dos componentes previstos na abordagem de gerenciamento proposta pela *IETF*, efetuada no capítulo 3, estende os trabalhos já desenvolvidos, cujo tema se relaciona com gerenciamento de redes, onde geralmente foram somente conceituados os componentes como Gerente, Agente, *MIB* e objetos gerenciados, sem, no entanto, identificar subcomponentes e detalhes relativos à implementação.

Os diferentes esforços que estão sendo realizados por desenvolvedores comerciais e organismos de padronização em direção ao *Web-Based Network Management*, foram considerados no desenvolvimento do protótipo, devido ao grande sucesso da *Web* como interface padrão para diferentes aplicações. Dessa forma, a utilização dos chamados recursos de *Web* (*HTML* e extensões, *aplicações de CGI*, *Web Browser* e *Web Server*) demonstrou, parcialmente, as potencialidades que tais mecanismos poderão oferecer, quando inseridos em ferramentas para o gerenciamento de redes.

Como contribuição mais significativa deste trabalho, encontra-se a proposta para o gerenciamento de aplicações em rede, apresentada no capítulo 5. Especialmente, a proposta efetuada sistematiza atividades a serem realizadas para o desenvolvimento de ferramentas de gerenciamento. Portanto, desenvolvedores de aplicações e administradores de rede, ao seguirem a proposta, terão um referencial para a

especificação, implementação e, até mesmo, utilização de sistemas destinados ao gerenciamento das implementações *client/server* e dos respectivos serviços oferecidos na *Internet*.

Na aplicação da proposta para o gerenciamento do *HTTP*, pode-se verificar a validade do tema desenvolvido. O detalhamento das atividades estipuladas na proposta possibilitou apresentar referencial teórico contendo: os aspectos funcionais das entidades *client/server* utilizadas na *Internet* previstos em nível de protocolo e/ou implementação, as características e aplicabilidades das *MIBs* existentes e os detalhes para construção de um sistema.

O desenvolvimento do protótipo de sistema, com base em tecnologias abertas e de domínio público, contribuiu para reduzir uma lacuna até então existente nos trabalhos relacionados, que, por utilizarem, na maioria das vezes, plataformas de gerenciamento proprietárias na validação de suas propostas, ocultavam detalhes sobre os componentes necessários a um sistema de gerenciamento.

Dentre as dificuldades enfrentadas, destacam-se a falta de documentação detalhada sobre parte das tecnologias utilizadas, como, por exemplo, a plataforma de gerenciamento *CMU-SNMP*, as Aplicações Agentes implementadas nos Sistemas Operacionais *AIX* e *OSF/1* e o Servidor de Banco de Dados *MySQL*.

## **7.2 Trabalhos futuros**

Como trabalho futuro, sugere-se o refinamento da proposta para o gerenciamento de aplicações em rede. Tal refinamento poderá prever, por exemplo, os seguintes avanços:

- Automatização da atividade 1 e 3, onde parte do levantamento de dados poderá ser efetuado com o uso de programas de busca (*robôis*), direcionados ao ambiente operacional que será gerenciado.



- Construção de uma ferramenta de *software* que forneça apoio à atividade 6. Tal ferramenta deverá fornecer, de forma integrada, uma *API* para a comunicação entre Gerente-Agente, para o acesso à base de dados e para a construção da interface de usuário.

Em relação ao protótipo de sistema que foi desenvolvido, trabalhos futuros poderão tratar dos seguintes aspectos:

- Identificação dos requisitos de usuário, para diferentes categorias de aplicações, considerando o ponto de vista dos desenvolvedores, administradores e usuários finais.
- Especificação de uma *MIB* que suporte, além de monitoramento, grupos de objetos para o controle (configuração) das aplicações gerenciadas.
- Ampliação do Banco de Dados, com a inclusão de novas tabelas para o armazenamento de dados históricos, possibilitando o fornecimento de novos indicadores e métricas.
- Adição de outras tecnologias de *Web* aplicadas ao gerenciamento de redes, como, por exemplo, *Applets Java*.

## Referências Bibliográficas

- [BaWa96] *BARRETT, M. L. WAGNER, C. H. C and UNIX: Tools for Software Design. United States of America: Jhon Wiley & Sons, Inc., 1996. 446p.*
- [BeTa97] BERTHOLDO, L. M. TAROUÇO, L. M. R. **Implementando Segurança e Controle em Redes de Computadores.** Anais 15º SBRC, p.61-78, mai. 1997.
- [BRI93] BRISA. **Gerenciamento de Redes - Uma Abordagem de Sistemas Abertos,** São Paulo: *MAKRON Books do Brasil Editora Ltda., 1993.*
- [BuFo97] *BUTZEN, F. FORBES, D. The Linux Database. New York: MIS:Press, 1997. 559p.*
- [CaMa94] CARRILHO, J. A. MADEIRA, E. **Um Esquema para o Gerenciamento do Protocolo FTP baseado em Domínios.** Anais 12º SBRC, v.II, p.343-362, mai. 1994.
- [CLW96] CORRÊA, A. S. N. LUDVIG, J. A. WESTPHALL, C. B. **Projeto PLAGERE: Novas Funções de Gerência de Contabilidade para Rede UFSC.** CNPq - PROTEM-CC, 1996.
- [Cou97] COUGO, P. **Modelagem Conceitual e Projeto de Bancos de Dados.** Rio de Janeiro: Editora Campus Ltda., 1997. 280p.
- [CoSt96] *COMMER, D. E. STEVENS, D. L. Internetworking with TCP/IP: Client-Server Programming. Volume III. New Jersey: Prentice-Hall, Inc., 1996. 519p.*
- [Dav96] *DAVIS, Davin. User's Guide to CMU SNMP for Linux. On-line: URL [http://www.flash.net/~da\\_davis/cmusnmp.htm](http://www.flash.net/~da_davis/cmusnmp.htm), 1996.*
- [Der96] *DERI, L. Network Management for 90s. University of Berne, 1996.*

- [FGM97] *FIELDING, R. T. GETTYS, J. MOGUL, J. C. Hypertext Transfer Protocol - HTTP/1.1. Request For Comments 2068, January 1997.*
- [FMI97] *FMI - Free Range Media, Inc. Web-Based Enterprise Management Initiative. On-line: URL <http://wbem.freerange.com/wbem/pressrelease.htm>, 1997.*
- [Gas96] *GASPARY, L. P. CMU SNMP Tutorial. Trabalho Acadêmico - CPGCC - UFRGS. On-line: URL <http://penta.ufrgs.br/ger96/cmu-snmptutorial.html>, 1996.*
- [Har97] *HARNEDY, S. Total SNMP: Exploring the Simple Network Management Protocol. Second Edition. New Jersey: Prentice Hall PRT, 1997. 642p.*
- [Haz97] *HAZEWINKEL, H. Survey of Defined Managed Objects for Applications Management. Internet Draft <draft-hazewinkel-appl-mib-00.txt>, January, 1997.*
- [HKS97] *HAZEWINKEL, H. KALBFLEISCH, C. W. SCHOENWAEELDER, J. Definitons of Manged Objects for WWW Services. Internet Draft <draft-ietf-applmib-wwwmib-05.txt>, September, 1997.*
- [HMP96] *HARRISON, B. MELLQUIST, P. E. PELL, A. Web Based System and Network Management. Draft Internet: <draft-mellquist-web-sys-00.txt>, jun. 1996.*
- [IETFa] *INTERNET ENGINEERING TASK FORCE, Application MIB (applmib) Charter. On-line: URL <http://www.ietf.org/html.charters/applmib-charter.html>.*
- [IETFb] *INTERNET ENGINEERING TASK FORCE, SNMP Version 3 (snmpv3). On-line: URL <http://www.ietf.org/html.charters/snmpv3-charter.html>.*
- [ISO94] *ISO/IEC. Information processing systems - Open Systems Interconnection - System Management - Part 1: Usage Metering Funcion, 1994.*
- [Kal96] *KALBFLEISCH, C. Applicability of Standards Track MIBs to Management of World Wide Web Servers. Request For Comments 2039, November, 1996.*
- [KiFr94] *KILLE, S. FREED, N. Networking Services Monitoring MIB. Request For Comments 1565, January, 1994.*
- [Mat96] *MATTAR, F. N. Pesquisa de Marketing. Edição Compacta. São Paulo: Editora Atlas S.A., 1996. 270p.*

- [Nat95] NATALE, B. *Windows SNMP - The Open Interface for Programming network Management Protocol under Microsoft Windows, WinSNMP/Manager*. American Computer & Electronics Corporation, August 1995.
- [PeMc97] PERKINS, D. McGINNIS, E. *Understanding SNMP MIBs*. New Jersey: Prentice Hall PTR, 1997. 509p.
- [PHH97] PRAS, A. HAZENWINKEL, H. HENGSTUM, E. v. *Management of the World-Wide Web*. Anais 15° SBRC, p.340-345, mai. 1997.
- [PHL95] PERROW, G.S e outros. *The Abstraction and Modelling of Management Agents*. Department of Computer Science, University of Western Ontario, 1995.
- [PiVe95] PICOTO, C. VEIGA, P. *Management of a WWW Server using SNMP*. On-line: URL <http://www.di.fc.ul.pt/~cap/jenc6.html>, 1995.
- [Pra95] PRAS, A. *Network Management Architectures*. Netherlands: Centre for Telematics and Information Technology, 1995. 193p.(Thesis, Doctorate in Science of the Computation).
- [QDS97] Quantitative Data Systems, Inc. *Web-Based Management*. On-line: URL [http://www.qds.com/net\\_webbased.htm](http://www.qds.com/net_webbased.htm), 1997.
- [RCO95] RAMOS, S. de Q. CUNHA, P. R. F. OLIVEIRA, M. **Uma Metodologia de Apoio à Análise e ao Desenvolvimento de Aplicações em Sistemas de Gerenciamento de Redes de Computadores**. Anais 13° SBRC, p.143-161, mai. 1995.
- [RoMc95] ROSE, M. McCLOGHRIE, K. *How To MANAGE YOUR NETWORK Using SNMP: The Networking Management Practicum*. New Jersey: Prentice Hall PTR, 1995. 549p.
- [Ros94] ROSE, M. *The SIMPLE Book: An Introduction to Internet Management*. New Jersey: PTR Prentice-Hall, Inc., 1994. 456p.
- [SCM95] STAMATELOPOULOS, F. CHIOTIS, T. MAGLARIS, B. *A Scaleable, Platform-Based Architecture for Multiple domain Network Management*. IEEE International Conference on Communications'95 June, 1995.
- [ScSc97] SCHÖNWÄLDER, Jürgen. SCHÖNFELDER, Erik. *Linux CMU SNMP Project*. On-line: URL <http://www.gaertner.de/snmp/>, 1997.
- [SiMa95] SILVEIRA, C. K. MADEIRA, E. R. M. **Um Esquema para o Gerenciamento do Tráfego de Aplicações em Redes TCP-IP**. Anais 13° SBRC, p.567-585, mai. 1995.

- [SiTa94] SILVA, A. TAROUÇO, L. **Uma Proposta para Gerência de Correio Eletrônico**. Anais 12º SBRC, v.II, p.363-384, mai. 1994.
- [Sta96] STALLINGS, W. **SNMP, SNMPv2 and RMON: Practical Network Management**. Second Edition. United States of America: Addison Wesley, Inc., 1996. 478p.
- [Ste96] STEVENS, W. R. **TCP/IP Illustrated: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols**. Volume 3. United States of America: Addison-Wesley Publishing Company, Inc., 1996. 328p.
- [StMa97] STAMATELOPOULOS, F. MAGLARIS, B. **A Management Architecture for Internet Information Services**. NETMODE - Department of Electrical and Computer Engineering, National Technical University of Athens, 1997.
- [Sun97] Sun Microsystems, Inc. **Java Management API Overview**. On-line: URL <http://java.sun.com/products/JavaManagement/overview.html>, 1997.
- [Tan97] TANENBAUM, A. S. **COMPUTER NETWORKS**. Third Edition. New Jersey: Prentice-Hall, Inc., 1996. 813p.
- [TcX97] T.c.X - **DataKo0nsultAB**. On-line: URL <http://www.tcx.se>, 1997.
- [Xmy97] **XmySQL Home Page**. On-line: URL <http://web.wt.net/~dblhack>, 1997.
- [Web97] WEBER, R. F. **Segurança na Internet**. Anais da V ERI-SBC, p. 101-131. Mai, 1997.
- [WCS96] WALL, L. CHRISTIANSEN, T. SCHWARTZ, R. L. **Programming Perl**. 2<sup>nd</sup> Edition. United States of America: O'Reilly & Associates, Inc., 1996. 645p.

# ANEXO - A

## Descrição da *MIB* em *ASN.1*

```
NA-MIB          DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY, OBJECT-TYPE,
Counter32, Integer32,
IpAddress, enterprises
    FROM SNMPv2-SMI
MODULE-COMPLIANCE, OBJECT-GROUP
    FROM SNMPv2-CONF
TEXTUAL-CONVENTION, DisplayString,
DateAndTime, TAddress
    FROM SNMPv2-TC
applIndex
    FROM APPLICATION-MIB
TimeFilter
    FROM RMON2-MIB;

networkApplicationMIB MODULE-IDENTITY
LAST-UPDATED "970914300Z"
ORGANIZATION "UCPEL, GPR"
CONTACT-INFO
" Luiz Fernando T. Meirelles

Postal:  Catholic University of Pelotas (UCPel)
        School of Computer Science      (ESIN)
        Group of Research in Network    (GPR)

        412 Felix da Cunha
        RS, Pelotas, 96.010-000
        BRAZIL

Tel:     +55 22.1555 - 233
Fax:     +55 25.3105
E-mail:  lftm@amadeus.ucpel.tche.br "
```

DESCRIPTION  
" This MIB maintains management data on

```

        applications entities (e.g. http,
        ftp, wais, gopher entities) running on
        the system and provides an SNMP-based
        interface for monitoring.
        The objects defined, provide
        information on entity configuration,
        utilization and statistics. "

        ::= { gpr 7 }

ucpel          OBJECT IDENTIFIER ::= { enterprises 2547 }
gpr            OBJECT IDENTIFIER ::= { ucpe1 1 }

naMIB          OBJECT IDENTIFIER ::= { gpr 7 }

-- Full Identifier :
-- (1.3.6.1.4.1.2547.1.7)
-- (iso.org.dod.internet.private.enterprises.ucpel.gpr.naMIB)

naObjects     OBJECT IDENTIFIER ::= { naMIB 1 }
naConformance OBJECT IDENTIFIER ::= { naMIB 2 }
naCompliances OBJECT IDENTIFIER ::= { naConformance 1 }
naGroups      OBJECT IDENTIFIER ::= { naConformance 2 }

```

---

```
-- TEXTUAL CONVENTIONS
```

```

VersionNumber ::= TEXTUAL-CONVENTION
  STATUS      current
  DESCRIPTION
    " An Octet String type representing a version number
    consisting of up to four numbers, giving versioning
    of the type x.x.x.x; each of the four number parts
    used for detailed version definition is expressed by the
    use of each of the four 8 bit parts of the octet string.
    Significance increases from left to right.
    Non existant subversions take the value 0.
    For example a version number 3.3.1 is represented
    by the string 3310. "
  SYNTAX      OCTET STRING (SIZE (4))

ResponseType  ::= TEXTUAL-CONVENTION
  STATUS      current
  DESCRIPTION
    " The ResponseType defines the different response
    values used by applications entities."
  SYNTAX      INTEGER (0..2147483647)

UserIdentification ::= TEXTUAL-CONVENTION
  STATUS      current
  DESCRIPTION
    " Represents textual identification of an user.
    The construction of the value consists of
    two parts:

```

<username>'@'<hostname>

Whereby the <username>'@' is optional in case the user name is known.

The character set to be used is NVT ASCII, as defined in pages 4, 10-11 of RFC 854.

Any object defined using this syntax may not exceed 100 characters in length."

-- Restricted in size due to the fact an OID may not  
-- exceed the total number of 128 sub-OIDs.

SYNTAX OCTET STRING (SIZE(0..100))

DomainString ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

" Represents textual identification for a domain.  
The character set should be the NVT ASCII character set, as defined in pages 4, 10-11 of RFC 854.

Any object defined using this syntax may not exceed 100 characters in length. "

-- Restricted in size due to the fact an OID may not  
-- exceed the total number of 128 sub-OIDs.

-- The length may also not be zero because it is used for  
-- an index-value.

SYNTAX OCTET STRING (SIZE(1..100))

-----  
-- naSystem Group

naSystem OBJECT IDENTIFIER ::= { naObjects 1 }

naSysHostAddress OBJECT-TYPE

SYNTAX IPAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

" The ip address of the application host. "  
::= { naSystem 1 }

naSysWanBandwidth OBJECT-TYPE

SYNTAX Integer32

UNITS "KBits/s"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

" The bandwidth of the internet connection for  
the particular site. Its value is defined manually  
by the network administrator. "  
::= { naSystem 2 }

naSysEntityTable OBJECT-TYPE



```

SYNTAX                SEQUENCE OF NaEntityEntry
MAX-ACCESS            not-accessible
STATUS                current
DESCRIPTION
" This table maintains information about the
  various applications running on the
  system. "
 ::= { naSystem 3 }

naSysEntityEntry OBJECT-TYPE
SYNTAX                NaSysEntityEntry
MAX-ACCESS            not-accessible
STATUS                current
DESCRIPTION
" The entries to the naSysEntity Table "
INDEX                { applIndex }
 ::= { naSysEntityTable 1 }

NaSysEntityEntry ::= SEQUENCE {
naSysEntityContact    DisplayString,
naSysEntityProtocol   INTEGER,
naSysEntityProtocolVersion  VersionNumber,
naSysEntityProtocolRole  INTEGER,
naSysEntityAddress    Address,
naSysEntityOid        OBJECT IDENTIFIER }

naSysEntityContact OBJECT-TYPE
SYNTAX                DisplayString
MAX-ACCESS            read-only
STATUS                current
DESCRIPTION
" Contact information for this application entity.
  It can be a name, phone, e-mail, etc. "
 ::= { naSysEntityEntry 1 }

naSysEntityProtocol OBJECT-TYPE
SYNTAX                INTEGER {
ftp(20),
tacacs(49),
http(80),
gopher(70),
wais(201),
icp(3128) }
MAX-ACCESS            read-only
STATUS                current
DESCRIPTION
" The main protocol supported by the entity,
  i.e. the supported application.
  It is by no means the port that the application
  listens to (even though these two value may
  be identical). "
 ::= { naSysEntityEntry 2 }

naSysEntityProtocolVersion OBJECT-TYPE
SYNTAX                VersionNumber
MAX-ACCESS            read-only
STATUS                current
DESCRIPTION

```

```
" The highest protocol version number supported by
the entity. "
 ::= { naSysEntityEntry 3 }
```

```
naSysEntityProtocolRole OBJECT-TYPE
SYNTAX INTEGER {
server(1),
serverAndMirror(2),
cachingProxy(3),
proxy(4),
gateway(5),
tunnel(6),
client(7),
other(8) }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" The particular role of the entity. "
 ::= { naSysEntityEntry 4 }
```

```
naSysEntityAddress OBJECT-TYPE
SYNTAX TAddress
MAX-ACCESS read-write
STATUS current
DESCRIPTION
" The transport address of this particular entity."
 ::= { naSysEntityEntry 5 }
```

```
naSysEntityOid OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" The authoritative identification of the
private MIB for this application entity,
provided by the product vendor.
If there is no appropriate
value to return, then 0.0 is assumed. "
 ::= { naSysEntityEntry 6 }
```

-----  
-- naStatistics group

```
naStatistics OBJECT IDENTIFIER ::= { naObjects 2 }

naStatTable OBJECT-TYPE
SYNTAX SEQUENCE OF NaStatEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
" This table maintains a summary of the
statistics of applications requests. "
 ::= { naStatistics 1 }
```

```

naStatEntry          OBJECT-TYPE
    SYNTAX            NaStatEntry
    MAX-ACCESS        not-accessible
    STATUS            current
    DESCRIPTION
        " The naStatTable entry "
    INDEX             { applIndex }
    ::= { naStatTable 1 }

NaStatEntry          ::= SEQUENCE {
    naStatRequests    Counter32,
    naStatReqErrCli   Counter32,
    naStatReqErrSer   Counter32,
    naStatAuthSuccess Counter32,
    naStatAuthFailure Counter32,
    naStatOutBytes    Counter32,
    naStatDateInit    TimeTicks }

naStatRequests       OBJECT-TYPE
    SYNTAX            Counter32
    MAX-ACCESS        read-only
    STATUS            current
    DESCRIPTION
        " The total number of requests generated or
        received by this application entity. "
    ::= { naStatEntry 1 }

naStatReqErrCli      OBJECT-TYPE
    SYNTAX            Counter32
    MAX-ACCESS        read-only
    STATUS            current
    DESCRIPTION
        " The total number of request errors
        detected by the application entity.
        These are error in the syntax of the
        request message (IS protocol level). "
    ::= { naStatEntry 2 }

naStatReqErrSer      OBJECT-TYPE
    SYNTAX            Counter32
    MAX-ACCESS        read-only
    STATUS            current
    DESCRIPTION
        " The total number of request errors
        detected by the application entity.
        These are error in the syntax of the
        request message (IS protocol level). "
    ::= { naStatEntry 3 }

naStatAuthSuccess    OBJECT-TYPE
    SYNTAX            Counter32
    MAX-ACCESS        read-only
    STATUS            current
    DESCRIPTION
        " A counter for successfully authenticated accesses. "
    ::= { naStatEntry 4 }

```

```

naStatAuthFailure      OBJECT-TYPE
    SYNTAX               Counter32
    MAX-ACCESS           read-only
    STATUS               current
    DESCRIPTION
        " A counter for authentication failures. "
    ::= { naStatEntry 5 }

naStatOutBytes         OBJECT-TYPE
    SYNTAX               Counter32
    MAX-ACCESS           read-only
    STATUS               current
    DESCRIPTION
        " The sum number of bytes transmitted by this
          application entity. "
    ::= { naStatEntry 6 }

naStatDataInit
    SYNTAX               TimeTicks
    MAX-ACCESS           read-only
    STATUS               current
    DESCRIPTION
        " The time (in hundredths of a second) since
          the access_log was last create. "
    ::= { naStatEntry 7 }

-----
-- naMonitor group

naMonitor              OBJECT IDENTIFIER ::= { naObjects 3 }

naMonDocumTableSize   OBJECT-TYPE
    SYNTAX               Integer32
    MAX-ACCESS           read-write
    STATUS               current
    DESCRIPTION
        " The maximum size of the naMonDocumTable. "
    ::= { naMonitor 1 }

naMonDocumTableRefreshPeriod OBJECT-TYPE
    SYNTAX               Integer32
    MAX-ACCESS           read-write
    STATUS               current
    DESCRIPTION
        " Refresh period of the naMonDocumTable
          in minutes. If the manager sets its value
          to '-1' then refreshing is disabled. If
          the manager sets its value to '0' the table
          is refreshed immediately. '0' does not
          update the object's value."
    ::= { naMonitor 2 }

naMonDocumTable       OBJECT-TYPE
    SYNTAX               SEQUENCE OF NaMonDocumEntry
    MAX-ACCESS           not-accessible

```

```

STATUS          current
DESCRIPTION
" This table maintains information on the
  documents published by entities and their
  access statistics. The entry counters are
  reset every time the manager reads them. "
 ::= { naMonitor 3 }

naMonDocumEntry      OBJECT-TYPE
SYNTAX              NaMonDocumentEntry
MAX-ACCESS          not-accessible
STATUS              current
DESCRIPTION
" The entry of the table. "
INDEX               { applIndex,
                    naMonDocumTimeMark }
 ::= { naMonDocumTable 1 }

NaMonDocumEntry      ::= SEQUENCE {
naMonDocumTimeMark   TimeFilter,
naMonDocumResourceURL DisplayString,
naMonDocumType       INTEGER,
naMonDocumSize       Integer32,
naMonDocumLastUpdate DateAndTime,
naMonDocumAccesses   Counter32 }

naMonDocumTimeMark   OBJECT-TYPE
SYNTAX              TimeFilter
MAX-ACCESS          not-accessible
STATUS              current
DESCRIPTION
" TimeMark of when the document was last accessed."
REFERENCE
"The RMON-2 MIB"
 ::= { naMonDocumEntry 1 }

naMonDocumResourceURL OBJECT-TYPE
SYNTAX              DisplayString
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
" The URL containing the resource
  for the document. "
 ::= { naMonDocumEntry 2 }

naMonDocumType       OBJECT-TYPE
SYNTAX              INTEGER {
                    static(1),
                    dynamic(2),
                    mirror_image(3) }

MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
" The type of the document. "
 ::= { naMonDocumEntry 3 }

```

```

naMonDocumSize          OBJECT-TYPE
    SYNTAX               Integer32
    UNITS                 "Bytes"
    MAX-ACCESS           read-only
    STATUS                current
    DESCRIPTION
        " The document's size of a static document in bytes.
          If the documentType is dynamic this value should be 0."
    ::= { naMonDocumEntry 4 }

naMonDocumLastUpdate    OBJECT-TYPE
    SYNTAX               DateAndTime
    MAX-ACCESS           read-only
    STATUS                current
    DESCRIPTION
        " The date of the document last update. "
    ::= { naMonDocumEntry 5 }

naDocumAccesses         OBJECT-TYPE
    SYNTAX               Counter32
    MAX-ACCESS           read-only
    STATUS                current
    DESCRIPTION
        " The total number of accesses to the document. "
    ::= { naMonDocumEntry 6 }

naMonRecentUserTableSize OBJECT-TYPE
    SYNTAX               Integer32
    MAX-ACCESS           read-write
    STATUS                current
    DESCRIPTION
        " The maximum size for the isnRecentUserTable. "
    ::= { naMonitor 4 }

naMonRecentUserTableDelUsers OBJECT-TYPE
    SYNTAX               Counter32
    MAX-ACCESS           read-write
    STATUS                current
    DESCRIPTION
        " The number of overflows (users
          deleted from the table due to the
          max size limit). "
    ::= { naMonitor 5 }

naMonRecentUserTable    OBJECT-TYPE
    SYNTAX               SEQUENCE OF NaMonRecentUserEntry
    MAX-ACCESS           not-accessible
    STATUS                current
    DESCRIPTION
        " This table logs users that have recently
          connected to the information entities up to
          the maximum number (naMonRecentUserTableSize)
          set by the manager.
          If a user is already in this table that will
          not be reinserted, but the associated conceptual
          row will be updated and its TimeMark value
          will be changed. "
    ::= { naMonitor 6 }

```

```

naMonRecentUserEntry OBJECT-TYPE
    SYNTAX NaMonRecentUserEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        " The entry of the table. "
    INDEX { applIndex,
           naMonRecentUserTimeMark }
    ::= { naMonRecentUserTable 1 }

NaMonRecentUserEntry ::= SEQUENCE {
    naMonRecentUserTimeMark TimeFilter,
    naMonRecentUserAddress DomainString,
    naMonRecentUserIdentif UserIdentification,
    naMonRecentUserAccesses Counter32,
    naMonRecentUserLastDocum DisplayString,
    naMonRecentUserLastAccess DateAndTime }

naMonRecentUserTimeMark OBJECT-TYPE
    SYNTAX TimeFilter
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        " The index of the table which is the
          time stamp for update of the conceptual
          row of the associated user.
          The RMON time filter mechanism is used. "
    ::= { naMonRecentUserEntry 1 }

naMonRecentUserAddress OBJECT-TYPE
    SYNTAX DomainString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        " The identification of domain of an user of the
          application."
    ::= { naMonRecentUserEntry 2 }

naMonRecentUserIdentif OBJECT-TYPE
    SYNTAX UserIdentification
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        " The identification of an user of the
          application."
    ::= { naMonRecentUserEntry 3 }

naMonRecentUserAccesses OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        " The total number of accesses for this user
          and entity. "
    ::= { naMonRecentUserEntry 4 }

```

```

naMonRecentUserLastDocum OBJECT-TYPE
    SYNTAX          DisplayString
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
    " The last document name accessed by the
      particular user. "
    ::= { naRecentUserEntry 5 }

```

```

naMonRecentUserLastAccess OBJECT-TYPE
    SYNTAX          DateAndTime
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
    " The date of the document last access. "
    ::= { naRecentUserEntry 6 }

```

-----  
-- Conformance and Compliance Definition

```

naSystemGroup      OBJECT-GROUP
    OBJECTS        {
                    naSysHostAddress,
                    naSysWanBandwidth,
                    naSysEntityContact,
                    naSysEntityProtocol,
                    naSysEntityProtocolVersion,
                    naSysEntityProtocolRole,
                    naSysEntityAddress,
                    naSysEntityOid,
                }
    STATUS          current
    DESCRIPTION
    " The naSystem group maintains information
      about system, the information entities
      (servers, clients, proxies, etc.) of general
      purpose definitions. "
    ::= { naGroups 1 }

```

```

naStatisticsGroup OBJECT-GROUP
    OBJECTS        {
                    naStatRequests,
                    naStatReqErrCli,
                    naStatReqErrSer,
                    naStatAuthSuccess,
                    naStatAutuFailure,
                    naStatOutBytes,
                    naStatDateInit }
    STATUS          current
    DESCRIPTION
    " The naStatistics group maintains information
      associated to the information protocols
      supported by the system entities and
      provides mostly statistics about its
      operation at the protocol PDU or byte
      stream level. "
    ::= { naGroups 2 }

```



```

naMonitorGroup      OBJECT-GROUP
OBJECTS              {      naMonDocumTableSize,
                           naMonDocumTableRefreshPeriod,
                           naMonDocumResourceURL,
                           naMonDocumType,
                           naMonDocumSize,
                           naMonDocumentLastUpdate,
                           naMonDocumAccesses,
                           naMonRecentUserTableSize,
                           naMonRecentUserTableDelUsers,
                           naMonRecentUserAddress,
                           naMonRecentUserIdentif,
                           naMonRecentUserAccesses,
                           naMonRecentUserLastDocument,
                           naMonRecentUserLastAccess }

STATUS               current
DESCRIPTION
" The isnMonitor group maintains information
  of current (or near past) states of the
  entities running on the system, that the
  agent monitors in real time, on demand or
  in a periodical fashion. "
 ::= { naGroups 3 }

```

---

```

naCompliance        MODULE-COMPLIANCE
STATUS               current
DESCRIPTION
" The compliance statements for the
  the NA MIB groups. "
MODULE -- This module
    MANDATORY-GROUPS {
        naSystemGroup,
        naStatisticsGroup,
        naMonitorGroup }
MODULE APPLICATION-MIB -- The Application MIB (MADMAN)
    MANDATORY-GROUPS {
        applGroup,
        assocGroup }
MODULE HOST-RESOURCES-MIB -- The Host Resources MIB
    MANDATORY-GROUPS {
        hrSystemGroup,
        hrStorageGroup,
        hrSWRunPerf }
 ::= { naCompliances 1 }

```

END

## ANEXO B

### Descrição dos módulos do protótipo

O protótipo de sistema para o gerenciamento de aplicações em rede deverá ser acionado a partir de um *link* previamente definido no *site* dos administradores. O *link* para o protótipo é direcionado para um arquivo inicializador que contém a função *window.open* da Linguagem *JavaScript*, utilizada para a abertura de mais uma janela do *browser*, conforme apresentado na FIG. B.1. Por questões de segurança, após acionar o *link*, o usuário será submetido a um processo de autenticação configurado no *Web Server*, onde se encontram as interfaces do sistema. Uma breve descrição dos módulos e de suas funcionalidades é efetuada nos próximos itens.

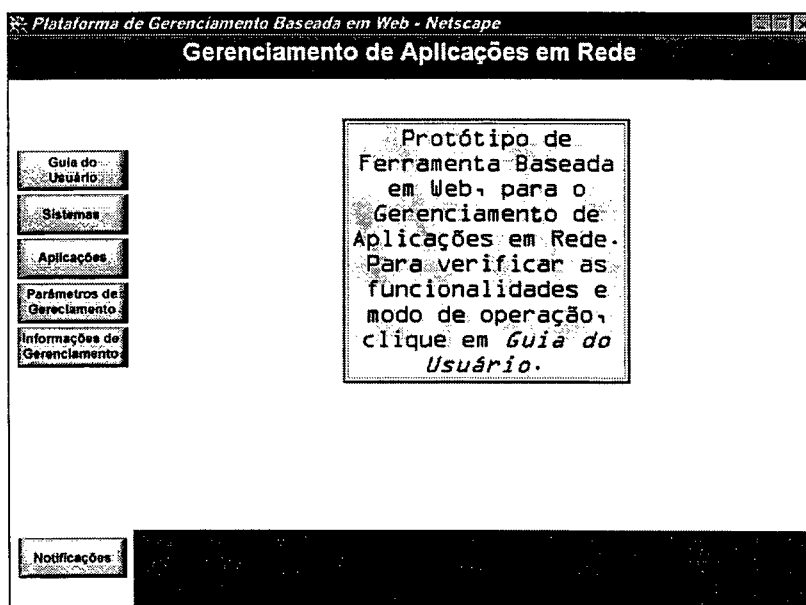


FIGURA B.1: Interface de abertura do protótipo

## B.1 Módulo Sistemas

O módulo para a atualização dos sistemas que serão gerenciados apresentará um formulário solicitando os dados de identificação (Endereço IP ou Nome do *Host*, Endereço da Porta da Aplicação e Nome da Comunidade). Esses dados são utilizados para verificar se a Aplicação Agente já está ativada e se o sistema já foi cadastrado no Banco de Dados da Aplicação Gerente. Os dados apresentados na FIG. B.2 indicam um sistema que ainda não foi cadastrado e cuja Aplicação Agente já havia sido previamente inicializada.

A imagem mostra uma janela de navegador Netscape com o título "Plataforma de Gerenciamento Baseada em Web - Netscape". O conteúdo principal é uma página intitulada "Gerenciamento de Aplicações em Rede". No topo, há um menu "Sistemas" com sub-itens: "Guia do Usuário", "Sistemas", "Aplicações", "Parâmetros de Gerenciamento" e "Informações de Gerenciamento". Abaixo do menu, há um formulário com o seguinte conteúdo:

Sistemas	
Endereço IP ou Nome do Host	amadeus.ucpel.tche.br
Endereço da Porta do Agente	161
Nome da Comunidade	public
<input type="button" value="Acessar"/> <input type="button" value="Limpar"/> <input type="button" value="Cancelar"/>	

Na parte inferior esquerda, há um botão "Notificações".

FIGURA B.2: Interface para o cadastramento de sistemas que serão gerenciados

Após acionar o botão **Acessar**, uma aplicação de *CGI* em *Perl* realizará os procedimentos citados e, ocorrendo sucesso, retornará uma tabela contendo os dados preenchidos anteriormente, adicionados das informações recuperadas da *MIB* (*System Group*) da Aplicação Agente, juntamente com um formulário, contendo somente os componentes de execução (FIG. B.3). Assim que o usuário confirmar os dados, poderá acionar o botão **Atualizar** para inserir as informações na tabela *sis\_ger*.

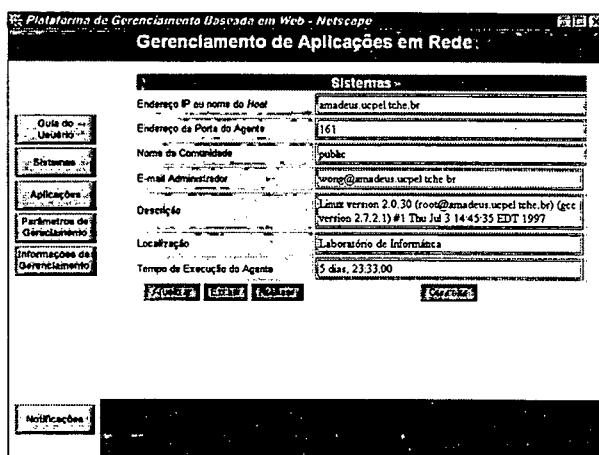


FIGURA B.3: Interface para confirmação e atualização dos dados dos sistemas no B.D.

## B.2 Módulo Aplicações

O módulo para atualização das aplicações que serão gerenciadas, apresentará um formulário solicitando os dados de identificação (Endereço IP ou Nome do *Host*, Endereço da Porta do Agente e Endereço da Porta da Aplicação). Estes dados são utilizados para verificar se o sistema onde a aplicação está rodando já foi cadastrado (tabela *sis\_ger*), e se a própria aplicação já foi cadastrada (tabela *apl\_ger*). No exemplo apresentado na FIG. B.4, trata-se de uma aplicação que ainda não foi cadastrada no Banco de Dados da Aplicação Gerente.

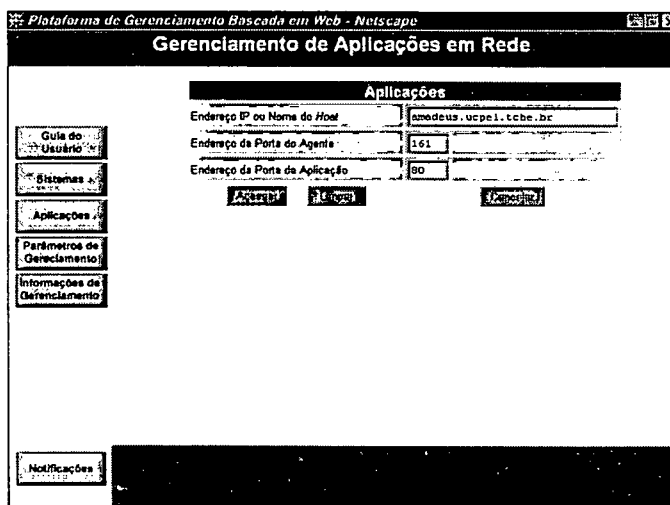


FIGURA B.4: Interface para o cadastramento das aplicações de rede

Após acionar o botão **Acessar**, uma aplicação de CGI verifica se o sistema foi cadastrado, recupera outros dados junto à MIB (naSystem Group) da Aplicação Agente e retorna o formulário (FIG. B.5), contendo os dados preenchidos anteriormente e as informações recuperadas. Assim que o usuário confirmar os dados, poderá acionar o botão **Atualizar** para inserir as informações na tabela `apl_ger`.

Aplicações	
Endereço IP ou nome do Host	amadeus.ucpel.tche.br
Endereço da Porta do Agente	161
Endereço da Porta da Aplicação	80
Versão do protocolo da Aplicação	1.1
Versão da implementação da Aplicação	1.2.4
Email Administrador	panqueca@amadeus.ucpel.tche.br
Valor do object identifier da MIB do fabricante	

Atualizar Excluir Notificar Cancelar

FIGURA B.5: Interface para confirmação e atualização dos dados das aplicações

### B.3 Módulo Informações de Gerenciamento

Considerando-se como executados os módulos Sistemas, Aplicações e Parâmetros de Gerenciamento, o usuário poderá acionar o Módulo Informações de Gerenciamento, o qual provocará a carga do formulário para solicitação de informações. Um exemplo válido para utilização do formulário é apresentado na FIG. B.6.

Plataforma de Gerenciamento Baseada em Web - Netscape

## Gerenciamento de Aplicações em Rede

Gua do Usuário

Sistemas

Aplicações

Parâmetros de Gerenciamento

Informações de Gerenciamento

### Informações de Gerenciamento

**ORIGEM DAS INFORMAÇÕES**

Endereço IP ou Nome do Host

Endereço da Porta da Aplicação

MIB do Agente  
 Banco de Dados do Gerente

**PERÍODO PARA PROCESSAMENTO**  
(Somente para Informações do Banco de Dados)

Inicial(aammddhhmm)

Final (aammddhhmm)

Intervalo   horas  minutos

---

**RELATÓRIOS** (As informações serão sempre recuperadas do Banco de Dados)

Alarmes de Segurança

---

INDICADORES		LT	TG		LT	TG
<input checked="" type="checkbox"/> Confiabilidade		<input type="text" value="01"/>	<input type="text"/>	<input type="checkbox"/> Documentos acessados	<input type="text"/>	<input type="text"/>
<input type="checkbox"/> Bytes transferidos		<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/> Acessos por domínio	<input type="text" value="10"/>	<input type="text" value="Setores"/>

---

Notificações

200.17.92.55: Enterprise Specific Trap (401) Uptime: 3 days, 22:16:18  
200.18.78.45 - netline.ucpel.tche.br lfm  
Um pedido de Autenticação falhou

FIGURA B.6: Interface para solicitação de informações de gerenciamento

Os parâmetros preenchidos no formulário indicam que se trata de um pedido de informações de gerenciamento referente ao *Host amadeus.ucpel.tche.br* e aplicação de rede que implementa o protocolo *HTTP*, cujo endereço de porta está configurado com valor **80**.

Outro aspecto a observar é que as informações serão recuperadas diretamente da *MIB* da Aplicação Agente, exceto os relatórios de alarme de segurança que sempre serão extraídos do Banco de Dados da Aplicação Gerente.

Como período para o processamento do referido relatório, foi estipulado o intervalo das 10 h às 10 h 59 min, de 22/10/97, com intervalo de 15 minutos. Isto significa que serão exibidos todos os alarmes de segurança gerados pela Aplicação Agente da entidade gerenciada (*Host amadeus* e implementação do *HTTP*) no período de tempo especificado, o mais próximo possível do intervalo estipulado.

As demais informações de gerenciamento solicitadas referem-se à obtenção do indicador de confiabilidade e de estatísticas de acessos por domínio, as quais serão recuperadas diretamente da *MIB* (*naStatistics* e *naMonitor Group*) da Aplicação Agente. O indicador de confiabilidade será exibido em uma tabela contendo 1 linha de dados, sem a criação de nenhum tipo de gráfico. Já as estatísticas de acesso por domínio serão apresentadas em uma tabela de até 10 linhas de dados, seguida por um gráfico de setores, demonstrando o percentual de acessos efetuados pelos domínios, encontrados na *MIB* (*naMonRecentUserTable*) da Aplicação Agente.

A informação existente na área destinada à exibição das notificações em tempo real (parte inferior da interface), embora não solicitada, é resultante de uma mensagem enviada pela Aplicação Agente via operação *Trap* do *SNMP*, a qual indica a ocorrência de um evento relacionado à segurança da aplicação gerenciada.

Como resultado da submissão do formulário, quando o usuário aciona o botão **Executar**, uma aplicação de *CGI* retorna as informações de gerenciamento solicitadas. A descrição dos dados processados com base nos parâmetros indicados na FIG. B.6, será efetuada com o uso da seguinte estrutura:

- Nome relatório, indicador ou dado estatístico.
- Uma descrição textual do indicador ou dado estatístico.
- Identificação dos propósitos de gerenciamento alcançados.
- Identificação dos objetos da *MIB* e da fórmula utilizados.

**Nome:** Relatório de Alarmes de Segurança

**Descrição:** Notificações enviadas pela Aplicação Agente e armazenadas no banco de dados da Aplicação Gerente.

**Propósitos de gerenciamento:** Monitoramento de segurança.

**Objetos da MIB e fórmula:** Não necessários, em virtude da utilização da ferramenta *snmptrap*, disponível no pacote *CMU - SNMP*.

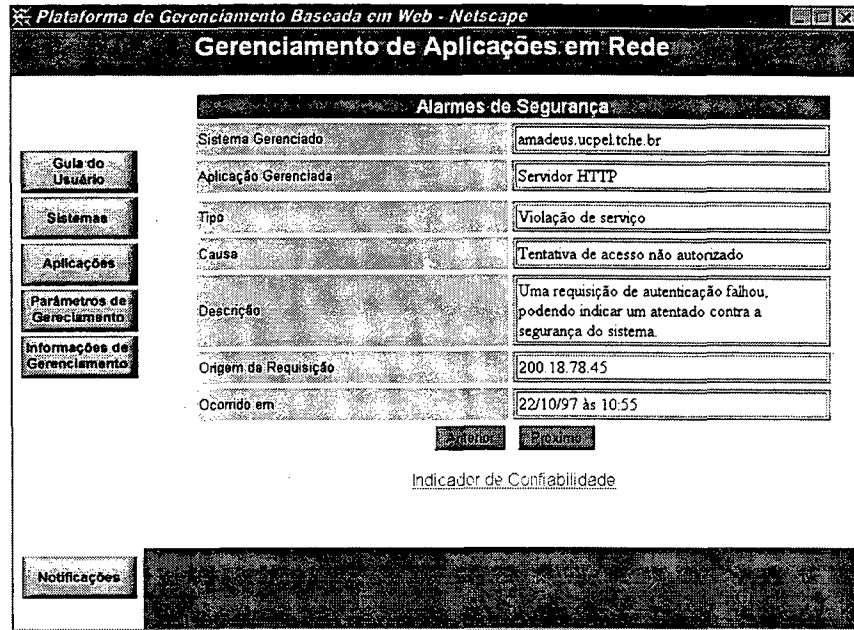


FIGURA B.7: Interface com a exibição dos relatórios de alarme de segurança

**Nome:** Indicador de confiabilidade

**Descrição:** Percentual de erros causados por entidades *clients* e/ou *server* em relação ao total de requisições atendidas pela entidade gerenciada.

**Propósitos de gerenciamento:** Planejamento e alocação de recursos.

**Objetos da MIB e fórmula:** *naStatRequests*, *naStatReqErrCli* e *naStatReqErrSer*  

$$\frac{(naStatReqErrCli + naStatReqErrSer)}{naStatRequests}$$



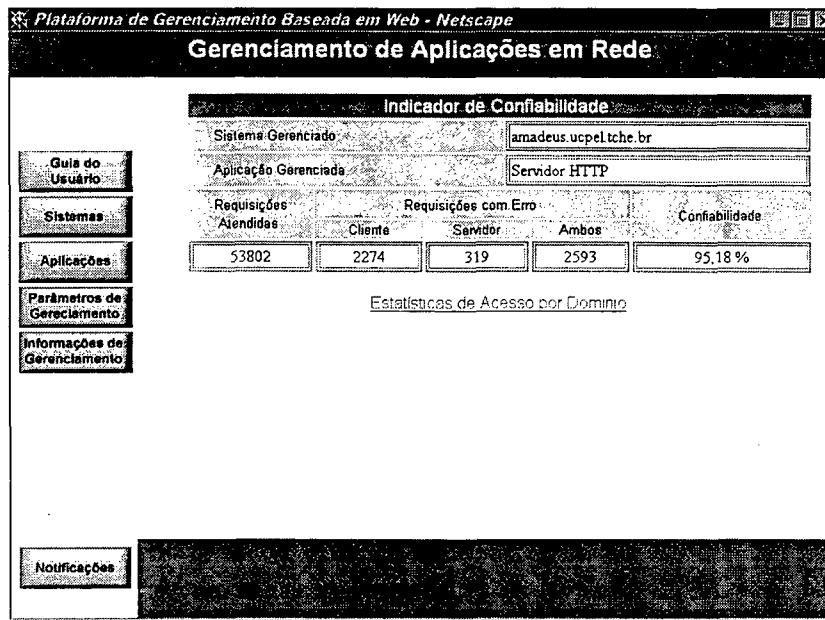


FIGURA B.8: Interface com o indicador de confiabilidade da aplicação (informações recuperadas da *MIB* do Agente)

<b>Nome:</b>	<u>Estadísticas dos acessos por domínio</u>
<b>Descrição:</b>	Estadísticas dos domínios/usuários com recentes acessos e dos domínios/usuários que mais acessam a aplicação gerenciada.
<b>Propósitos de gerenciamento:</b>	Monitoramento da utilização dos recursos.
<b>Objetos da <i>MIB</i> e fórmula:</b>	<i>naMonRecentUserAddress</i> , <i>naMonRecentUserIdentf</i> , <i>naMonRecentUserAccesses</i> , <i>naMonRecentUserLastAccess</i>
	Quatro tabelas: domínios/usuários que mais acessam ( <i>top domínios/users</i> ) e domínios/usuários recentes. A Aplicação Gerente, periodicamente, recupera e esvazia

a *naMonRecentUserTable* da *naMIB*, armazenando as informações coletadas no Banco de Dados. Os domínios/usuários que mais acessam são facilmente calculados e armazenados em tabelas distintas. A tabela dos domínios/usuários recentes é facilmente construída com base na classificação dos domínios/usuários pelo objeto (variável) *naMonRecentUserLastAccess*, considerando-se as primeiras N linhas da tabela (N é configurado pela Aplicação Gerente).

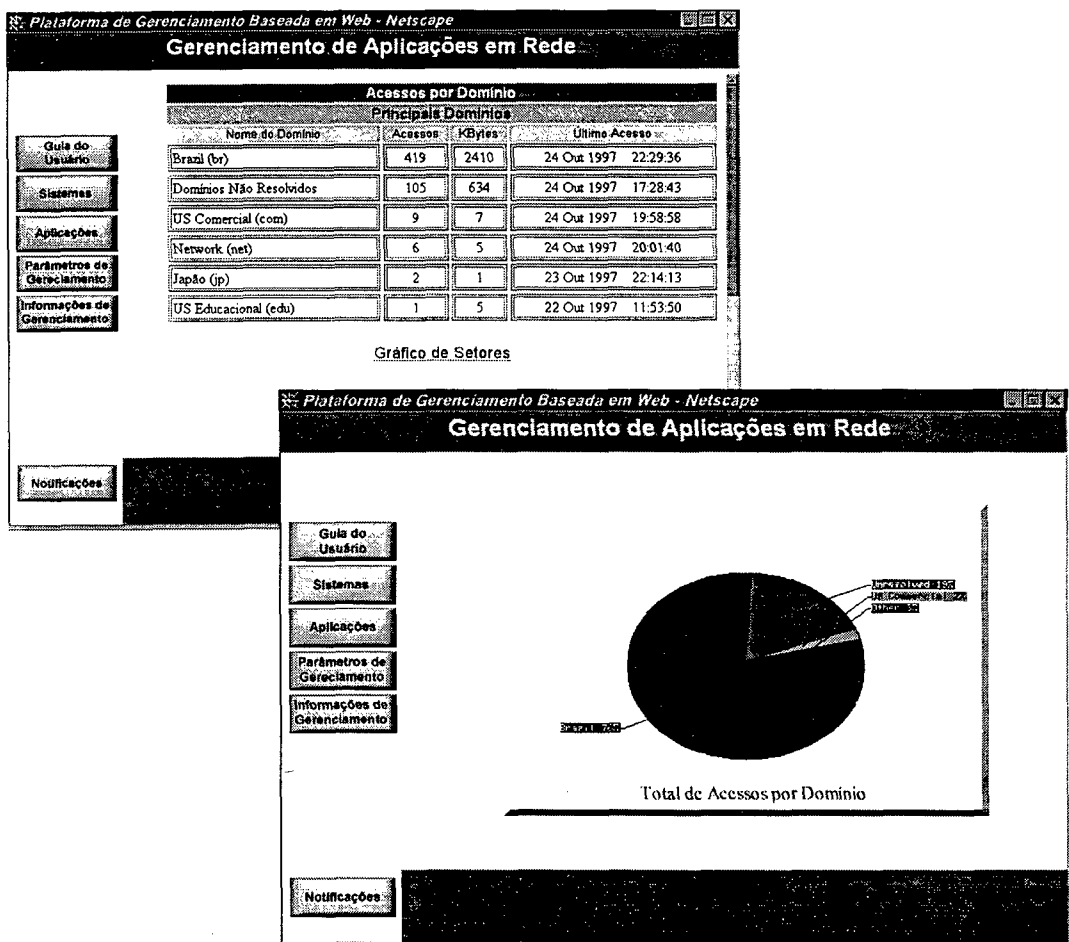


FIGURA B.9: Interface com a relação dos principais domínios que acessam a aplicação