UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE CIÊNCIAS DA COMPUTAÇÃO

(Bacharelado)

PROTÓTIPO DE UM SOFTWARE GERENCIADOR DO SERVIDOR WEB UTILIZANDO O PROTOCOLO SNMP

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA COMPUTAÇÃO — BACHARELADO

MARCIANO DOVAL DALLMANN

BLUMENAU, JUNHO/2001

2001/1-48

PROTÓTIPO DE UM SOFTWARE GERENCIADOR DO SERVIDOR WEB UTILIZANDO O PROTOCOLO SNMP

MARCIANO DOVAL DALLMANN

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

_____

Prof. Sérgio Stringari — Orientador na FURB

_____

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

_____

Prof. Sérgio Stringari

_____

Prof. Francisco A. Péricas

_____

Prof. Miguel A. Wisintainer

AGRADECIMENTOS

Agradeço a todas as pessoas que acreditaram e de alguma forma me ajudaram a concluir esta etapa muito importante da minha vida.

Aos meus pais e familiares, que me educaram, sempre me apoiaram e estiveram comigo nas horas mais difíceis.

A minha noiva Josiane, que esteve ao meu lado, sempre me incentivando a alcançar meus objetivos.

Aos meus colegas de curso e amigos Cássio, Davi e Henrique, pelo companheirismo e amizade.

Ao meu professor e orientador Sérgio Stringari, que mostrou interesse no meu trabalho, sempre me incentivou à pesquisa, e me fez ter esperanças e acreditar que eu chegaria ao meu objetivo com sucesso.

E principalmente à Deus, por tudo.

RESUMO

Este trabalho de Conclusão de Curso (TCC) apresenta um estudo sobre o protocolo SNMP (*Simple Network Management Protocol*) e considerações sobre gerência de redes de computadores. Apresenta também, a especificação e implementação de um protótipo de um software gerenciador do servidor WEB, a partir dos seus arquivos de *log*, utilizando o protocolo SNMP .

ABSTRACT

This paper presents a study about the SNMP (Simple Network Management Protocol) and considerations on management of computer network. It also presents the specification and implementation of a prototype of a software that manages a WEB server by analyzing your log files, using the SNMP protocol.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE SIGLAS E ABREVIATURAS

API - Aplication Program Interfaces

ASN.1 - Abstract Sintax Notation.One

HOST- Estação de rede

IETF - Internet Engineering Task Force

IP - Internet Protocol

ISO - International Organization for Standardization

LOG – Repositório de registros

MIB - Management Information Base

OID – Object Identifier

OSI - Open System Interconection

PDU - Protocol Data Unit

RFC - Request for Comment

SNMP - Simple Network Management Protocol

SMI- Structure of Management Protocol

TCP - Transmission Control Protocol

UDP- User datagram Protocol

UML – Unified Modeling Language

WWW – World Wide Web

WEB – Sinônimo de WWW

1 INTRODUÇÃO

Conforme Soares (1995) uma rede de computadores é formada por um conjunto de módulos processadores (qualquer dispositivo capaz de se comunicar através do sistema de comunicação por troca de mensagens) capazes de trocar informações e compartilhar recursos, interligados por um sistema de comunicação.

De acordo com Zacker (2000) ninguém pode imaginar exatamente o que um usuário pretende enviar pela rede nem pode saber com certeza quais serão os efeitos provocados pelo próximo estágio de desenvolvimento em tecnologia de redes. A única maneira de um desenvolvedor ter certeza de que seu produto funcionará com o restante da rede é aderir rigorosamente aos padrões reconhecidos. Cada fabricante de hardware deve saber exatamente o que esperar como entrada para sua parte do sistema e o que seu sistema deve gerar como saída. Desenvolvedores de software trabalham com especificações semelhantes.

Para permitir o intercâmbio entre computadores de fabricantes distintos tornou-se necessário difundir uma arquitetura única, e para garantir que nenhum fabricante levasse vantagem em relação aos outros a arquitetura teria que ser aberta e pública. Foi com esse objetivo que a *International Organization for Standardization* (ISO) definiu o modelo denominado *Reference Model for Open Sistems Intercommunicatio*n (OSI) que propõe uma estrutura com sete níveis como referência para a arquitetura dos protocolos de redes de computadores (Soares, 1995)

A coexistência de redes heterogêneas (locais, metropolitanas e de longa distância) fez com que se tornasse necessário definir uma arquitetura voltada para interconexão dessas redes. Uma arquitetura importante no contexto de interconexão de redes heterogêneas é a arquitetura Internet, que se baseia na família dos protocolos *Transmission Control Protocol / Internet Protocol* (TCP/IP). O sistema de gerenciamento de redes da arquitetura TCP/IP opera na camada de aplicação e baseia-se no protocolo *Simple Network Management Protocol* (SNMP) (Soares, 1995).

O gerenciamento de rede implica na existência de um banco de dados contendo informações completas sobre todos os elementos da mesma (como por exemplo, linhas, *modens*, processadores de rede, terminais, computadores e software). Para cada um dos itens,

o operador da rede deve ser capaz de acessar informações como proprietário, localização, custo operacional, arrendatário, número serial e identificação do circuito. O acesso deve ser facilitado por meio de linguagens de alto nível para adicionar e manipular dados, bem como para acessá-los com vista ao cumprimento de funções de gerenciamento de rede. Os dados devem incluir informações sobre os fornecedores, o grau de confiabilidade de seus produtos, a fim de orientar futuras aquisições ou mesmo substituições de equipamentos de rede (Brisa, 1993).

Cada vez mais, verifica-se a instalação de um número crescente de redes com equipamentos de múltiplos fornecedores, o que pode levar a existência de ilhas de gerenciamento de fornecedores específicos que não trocam informações entre si. A interface entre o sistema de gerenciamento e cada componente desta rede é o que normalmente se chama interface gerente-agente. Para que as informações de gerenciamento da rede trafeguem desde o ponto de controle (gerente) até os componentes, e vice versa, é necessário que nesses pontos esteja implementado o mesmo protocolo. A curto prazo, a maneira mais prática de integrar sistemas de gerenciamento é utilizar interfaces gerente-agente. Para isso faz-se necessário desenvolver um gerente integrado, que deve obter informações de gerentes desenvolvidos para cada sistema de gerenciamento proprietário. Este gerente integrado é denominado “Gerente dos Gerentes” (*Manager of Managers*). Esta plataforma de gerenciamento é então responsável pela implementação de múltiplos protocolos de gerenciamento, reduzindo a necessidade de mudanças nos componentes hoje existentes (Soares, 1995).

No escopo do protocolo SNMP, definem-se os elementos de rede classificados como cliente (ou gerente) responsável pela monitoração e controle dos *gateways* e *hosts*, que correspondem aos servidores (ou agentes). O protocolo SNMP é baseado no paradigma conhecido como ”busca-armazenamento” (*fetch-store*, isto é, todas a operações previstas para este protocolo são derivadas de operação básicas de busca e armazenamento) (Soares, 1995).

O SNMP ajuda os administradores de rede a localizar e corrigir problemas em uma interligação em redes TCP/IP. Um administrador chama um cliente SNMP em seu computador local (geralmente uma estação de trabalho) e utiliza o cliente para contatar um ou mais servidores SNMP executados em máquinas remotas (normalmente Gateways). Além do protocolo SNMP, um padrão à parte , referente a uma *Management Information Base* (MIB)

Base de Informações de Gerenciamento, define o conjunto de variáveis que servidores SNMP mantêm, bem como a semântica de cada variável. Variáveis da MIB registram o estado de cada rede conectada, estatísticas de tráfego, contagens de erros encontrados e os conteúdos correntes de estruturas de dados internos, como a tabela de roteamento *internet protocol* (IP) da máquina (Comer, 1999).

Segundo Aragão (2000), nos últimos anos a popularidade da Internet tem aumentado bastante e uma das aplicações responsáveis por esse crescimento é sem dúvida a *Word Wide Web* (WWW). As informações que um servidor WWW oferece para seus clientes WWW é de suma importância para as organizações públicas e particulares. Assim, torna-se primordial que este serviço esteja sempre disponível e o administrador de um servidor WWW deve possuir uma grande quantidade de informações referente as operações do mesmo.

De acordo com Brisa (1993) o *log* é um repositório de registros que contém informações que devem ser preservadas. Estas informações são derivadas de relatórios de eventos internos ou de *Protocol Data Units* (PDUs) que entram no sistema e são complementadas pelo próprio processo de *Logging* (por exemplo, identificadores de registros e instante de *Logging*). Um *log* armazena os registros por ordem de chegada e os identificadores de registros são atribuídos em seqüência numérica.

Uma solução de gerenciamento de um servidor WWW consiste na verificação dos seus arquivos de *log*, podendo assim obter estatísticas e também identificar problemas que requerem alguma intervenção humana.

Diante do exposto acima, este trabalho de conclusão de curso apresenta um estudo sobre o protocolo SNMP, apresenta também a especificação e implementação de um protótipo de software para gerenciamento de um servidor WWW, no ambiente Microsoft Windows.

1.1 OBJETIVOS DO TRABALHO

O objetivo principal deste trabalho foi o desenvolvimento de um protótipo de software para gerenciar um servidor WEB, mais especificamente os arquivos de *log* desse servidor.

Os objetivos específicos do trabalho são:

a) aplicação do protocolo SNMP na conexão do protótipo com o servidor WEB;

b) a operação em redes locais com o sistema Microsoft Windows.

1.2 ESTRUTURA DO TRABALHO

Este Trabalho de Conclusão de Curso está organizado conforme a seguir.

O capítulo 1 introduz o contexto geral do trabalho, divididos dos segmentos: introdução e estrutura do trabalho.

O capítulo 2 abrange o tema gerência de redes de computadores.

O capítulo 3 fundamenta o protocolo SNMP, subdividido em operações suportadas, mensagens e MIB.

O capítulo 4 abrange o tema *log*, principalmente *log* do servidor WEB.

O capítulo 5 abrange a especificação do protótipo subdividido em diagrama de casos de uso, diagrama de classes, e diagrama de seqüência . E também abrange a implementação, subdividido em técnicas e ferramentas utilizadas e apresentação do protótipo.

O capítulo 6 descreve as conclusões obtidas neste TCC e sugestões para extensões e trabalhos futuros a serem realizados.

Para finalizar este TCC, apresenta-se as referências bibliográficas utilizadas durante o período de desenvolvimento deste Trabalho de Conclusão de Curso.

2 GERÊNCIA DE REDES DE COMPUTADORES

O gerenciamento de redes é o conjunto de funções que visa promover a produtividade da planta e dos recursos disponíveis a integrar, de forma organizada, as funções de operação, administração e manutenção de todos os elementos da rede e dos serviços de telecomunicações (Brisa, 1993).

Segundo Brisa (1994) o gerenciamento de rede deve possibilitar uma atuação preventiva , e não meramente reativa , com relação aos problemas.

O gerenciamento é uma prática vital para operação de redes. O uso dos serviços das redes é afetado pela disponibilidade e eficiência do gerenciamento de redes.

O gerenciamento no modelo OSI da ISO baseia-se na teoria da orientação a objetos. O sistema representa os recursos gerenciados através de entidades lógicas chamadas de objetos gerenciados. Ao desenvolver uma aplicação de gerenciamento, utilizam-se processos distribuídos conhecidos como gerentes (os quais gerenciam) e agentes (os que realizam ações) (Sztajnberg, 1996).

As atividades de gerenciamento de rede são divididas em cinco áreas funcionais denominadas, respectivamente: Gerenciamento de falhas, gerenciamento de configuração, gerenciamento de desempenho, gerenciamento de segurança e gerenciamento de contabilização. Estas áreas funcionais constituem processos de aplicação de gerenciamento que utilizam os serviços oferecidos pela camada de aplicação do modelo OSI (Brisa, 1994), (Mafinski, 1999) e (Sztajnberg, 1996).

O Gerenciamento de Falhas busca isolar e corrigir operações anormais do ambiente OSI. Abrange, entre outras, funções para investigar a ocorrência de falhas, identificar falhas, realizar seqüências de testes para fins de diagnósticos e corrigir falhas.

O Gerenciamento de Configuração tem como função controlar as condições do ambiente de comunicação do sistema aberto, identificando mudanças significativas e modelando a configuração dos recursos físicos e lógicos da rede .

O Gerenciamento de Desempenho oferece funções para medir, monitorar, avaliar e relatar os níveis de desempenho alcançados pela rede. Tais informações podem ser utilizadas para fins de planejamento e controle da qualidade do serviço da rede.

O Gerenciamento de Segurança apresenta três categorias de atividades, gerenciamento de segurança do sistema, gerenciamento dos serviços de segurança e gerenciamento dos mecanismos de segurança e inclui funções que buscam garantir a política de segurança definida para a rede.

O Gerenciamento de Contabilização oferece funções que possibilitam determinar o custo associado à utilização dos recursos da rede, e incluem funções que permitem determinar quais recursos e quanto desses recursos estão sendo utilizados.

Relacionando o modelo de gerenciamento OSI, quanto à sua estrutura, ele ainda é dividido em três tipos principais (Brisa, 1993), (Rekowsky, 1999) e (Sztajnberg, 1996):

- Gerenciamento de sistemas: sendo executado na camada de aplicação, precisa de apoio das sete camadas do modelo OSI para poder realizar a gerência e pode gerenciar qualquer objeto associado a um sistema aberto;
- Gerenciamento de camada: ocorre sobre os objetos gerenciados relacionados as atividades de uma camada específica e não depende dos protocolos de gerenciamento de outras camadas;
- Operações de camada: é utilizada no gerenciamento de uma única instância de comunicação em uma camada e utiliza-se do protocolo normal da camada para troca de informações e não precisa de um protocolo especial para poder realizar estas trocas de informações.

Relacionado o protocolo de gerenciamento OSI e a MIB, verifica-se que a MIB guarda as informações transferidas ou modificadas pelo uso de protocolos de gerenciamento OSI. Estas informações podem ser fornecidas por agentes administrativos locais ou remotos. As operações sobre objetos gerenciados são definidas de maneira abstrata e seu detalhamento está fora do escopo OSI, uma vez o agente e os objetos gerenciados situam-se no mesmo ambiente local (Mafinski, 1999), (Rekowsky, 1999) e (Sztajnberg, 1996).

Para a definição dos objetos gerenciados no modelo OSI deve-se considerar três hierarquias (Otsuka, 1995):

- hierarquia de herança, também denominada hierarquia de classe, tem como objetivo facilitar a modelagem dos objetos, através da utilização do paradigma da orientação a objetos. Assim podem ser definidas classes, super-classes, sub-classes. Trata-se de uma ferramenta para uma melhor definição de classes.
- hierarquia de nomeação, também chamada hierarquia de *containment*, descreve a relação de "estar contido em" aplicado aos objetos. Um objeto gerenciado está contido dentro de um e somente um objeto gerenciado. Um objeto gerenciado existe somente se o objeto que o contém existir, e dependendo da definição, um objeto só pode ser removido se aqueles que lhe pertencerem forem removidos primeiro.
- hierarquia de registro, é usada para identificar de maneira universal os objetos, independentemente das hierarquias de heranças e nomeação. Esta hierarquia é especificada segundo regras estabelecidas pela notação Abstract Sintax Notation.One (ASN.1).Assim, cada objeto é identificado por uma seqüência de números, correspondente aos nós percorridos desde a raiz, até o objeto em questão. Esta hierarquia é também usada pelo padrão Internet.

Segundo Sztajnberg (1996) gerência de Redes é uma aplicação distribuída onde processos de gerência (agentes e gerentes) trocam informações com o objetivo de monitorar e controlar a rede. O processo gerente envia solicitação ao processo agente que por sua vez responde às solicitações e também transmite notificações referentes aos objetos gerenciados que residem em uma base de informação de gerenciamento (MIB).

Toda e qualquer informação produzida pelo Sistema de Gerência, em um determinado instante, está ou em uma MIB ou trafegando pela rede (em uma comunicação típica entre um agente e um gerente ou entre dois gerentes) ou ainda poderá ser deduzida (reproduzida) com informações parciais oriundas destas duas fontes. Toda informação produzida pelo Sistema de gerência é útil para a manutenção da rede em operação com confiabilidade. Sem dúvida, os Sistemas de Gerência facilitam a administração das redes seja pela automatização de algumas atividades, seja por permitir maior controle sobre os recursos da rede ou ainda por fornecer

informações (estatísticas, por exemplo) que permitirão ajustes, correções ou adaptações às necessidades dos usuários.

Entretanto, neste ponto também é possível observar que o próprio Sistema de Gerência e as informações por ele geradas são de extrema valia para indicar pontos vulneráveis à ataques, ter acesso e controlar indevidamente recursos da rede, manipular informações, em suma, realizar atividades prejudiciais à rede, aos sistemas e/ou aos usuários.

Segundo Comer (2001) a gerência de rede não é definida como parte dos protocolos de transporte ou inter-rede. Em vez disso, os protocolos que um gerente de rede utiliza para monitorar e controlar dispositivos de rede opera no nível de aplicativo. Ou seja, quando um gerente precisa interagir com um dispositivo de hardware específico, o software de gerência segue o modelo cliente-servidor convencional: um programa aplicativo no computador do gerente atua como um cliente, e um programa aplicativo no dispositivo de rede atua como um servidor. O cliente do computador do gerente usa protocolos de transporte convencionais (por exemplo TCP ou *User Datagram Protocol* (UDP) para estabelecer comunicação com o servidor. Os dois então trocam requisições e respostas de acordo com o protocolo de gerência.

Para evitar uma confusão entre programas aplicativos que os usuários invocam e aplicativos que são reservados a gerentes de rede, os sistemas de gerência de rede evitam os termos cliente e servidor . Em vez disso, o aplicativo cliente que roda no computador do gerente é chamado de gerente, e um aplicativo que roda em um dispositivo de rede é chamado de agente.

3 PROTOCOLO SNMP

Segundo Brisa (1994) a filosofia de gerenciamento de redes apresentada pela comunidade internet é baseada no modelo conhecido como SNMP, que teve a sua origem em um protocolo para monitoração de *gateways IP*, o *Simple Gateway Management Protocol* (SGMP). Obviamente tal evolução implicou na introdução de mudanças no modelo original, que passou a utilizar alguns conceitos de gerenciamento do próprio modelo OSI. Tais Mudanças incluíram a definição de uma Estrutura de Informação de Gerenciamento *Structure of Management Information* (SMI) e de uma Base de Informação de Gerenciamento (MIB).

O SNMP define uma base limitada de informações de gerenciamento, com algumas variáveis dispostas em tabelas bidimensionais e um protocolo com funcionalidade limitada, que permite ao gerente apenas recuperar e atribuir valores às variáveis e, ao agente, enviar avisos não solicitados previamente, denominados *traps*. Como Conseqüência, uma implementação SNMP consome poucos recursos da rede e de processamento, o que permite a sua inclusão em equipamentos bastante simples. Além disso, por apresentar uma estrutura bastante dirigida e restrita, não é difícil conseguir a interoperabilidade entre estações de gerenciamento e softwares de agentes de fornecedores diferentes.

Segundo Sztajnberg (1996) o protocolo SNMP é a solução adotada na Internet para permitir que gerentes de redes possam localizar e corrigir problemas. Geralmente, é utilizado um processo na máquina do administrador chamado de cliente (uma *workstation* ou um *gateway*, por exemplo) que se conecta a um ou mais servidores SNMP localizados em máquinas remotas, para executar operações sobre os objetos gerenciados (por exemplo, para obter informações sobre estes objetos). O SNMP utiliza o protocolo UDP na comunicação entre cliente e servidor.

Para o cliente da rede, o SNMP executa as operações sobre os objetos de forma transparente, o que permite a interface do software de gerenciamento da rede criar comandos imperativos para executar operações sobre os objetos gerenciados. Esta é a grande diferença entre gerenciar uma rede usando o protocolo SNMP e gerenciar a mesma rede usando outros protocolos.

No protocolo SNMP são definidas tanto a sintaxe (forma e a representação dos nomes e do valores) como o significado das mensagens trocadas entre os clientes e os servidores. O formato das mensagens e dos objetos gerenciados de uma MIB são especificados com a linguagem ASN.1 e ao contrário de outros protocolos usados nas redes TCP/IP, suas mensagens não apresentam campos fixos, e portanto, não se pode representar as mensagens simplesmente com o uso de estruturas fixas.

O SNMP também define as relações administrativas entre os vários gateways que estão sendo gerenciados, determinando a autenticação necessária para os clientes acessarem os objetos gerenciados.

Ao contrário dos outros protocolos de gerenciamento que apresentam muitos comandos (operações), o SNMP apresenta somente um conjunto limitado de comandos, baseado num simples mecanismo de busca/alteração. Portanto, é muito mais simples de ser implementado do que um protocolo com muitas operações, em que cada operação sobre um objeto necessita de um comando diferente para implementá-la.

O mecanismo de busca/alteração conceitualmente só apresenta duas operações: uma que permite ao cliente alterar atributos de um objeto de uma MIB (SET), e outra para obter os valores dos atributos de um objeto (GET). Somente estão disponíveis estas operações (e suas variações) para o gerenciamento da rede, que serão aplicadas sobre os objetos de uma MIB. A principal vantagem de um mecanismo como este é a simplicidade e flexibilidade que este mecanismo dá ao protocolo, o que permite ao SNMP ser um protocolo bem estável porque a sua estrutura básica continuará fixa, mesmo que novos objetos sejam adicionados na MIB, ou que novas operações sejam definidas sobre estes objetos (elas serão constituídas por estas operações básicas).

A MIB define o conjunto e a semântica dos objetos que os servidores SNMP devem controlar, ou seja, define o conjunto conceitual de objetos que um servidor SNMP controla. A MIB é usada para armazenar em seus objetos os estados internos das entidades de uma rede.

Na maioria dos casos, usamos as variáveis convencionais para o armazenamento dos objetos de uma MIB, mas em alguns casos, em que a estrutura interna do TCP/IP não é exatamente compatível com a estrutura de um objeto de uma MIB, é necessário que o SNMP

seja capaz de computar os objetos de uma MIB a partir das estruturas de dados disponíveis (simulação deste conjunto conceitual de objetos). Como exemplo, o que um *gateway* deve fazer para saber por quanto tempo um sistema está operacional, a maioria dos sistemas simplesmente subtrai a hora corrente daquela em que o sistema iniciou mas neste caso, o software poderia simular um "objeto" que contenha o tempo decorrido desde o último *start-up* deste sistema.

Ao receber e enviar mensagens no protocolo SNMP, os nomes dos objetos não devem ser armazenados na forma textual, e sim na forma numérica definida pela sintaxe ASN.1, que representa o objeto univocamente, com o objetivo de tornar o pacote SNMP mais compacto. Quando a forma numérica que representa um objeto terminar com um zero (como em 1.3.6.1.2.1.4.3.0), representa que o objeto é a única instância existente. Por exemplo, o objeto gerenciável *iso.org.dod.internet.mgmt.mib.ip.ipInReceives* será representado na mensagem SNMP como *1.3.6.1.2.1.4.3*.

Para minimizar o espaço interno necessário para representar um objeto, e considerando que todos os objetos em uma MIB apresentam o mesmo prefixo no seu nome, podemos retirar o prefixo após a mensagem chegar na máquina, e recolocá-lo imediatamente antes de enviar a mensagem para outra máquina.

Podemos, resumidamente, dizer que os principais objetivos do protocolo SNMP, devido ao protocolo desejar ser flexível e simples, são:

- Reduzir o custo da construção de um agente que suporte o protocolo;
- Reduzir o tráfego de mensagens de gerenciamento pela rede necessárias para gerenciar os recursos da rede;
- Reduzir o número de restrições impostas as ferramentas de gerenciamento da rede, devido ao uso de operações complexas e pouco flexíveis;
- Apresentar operações simples de serem entendidas, sendo facilmente usadas pelos desenvolvedores de ferramentas de gerenciamento;
- Permitir facilmente a introdução de novas características e novos objetos não previstos ao se definir o protocolo;
- Construir uma arquitetura que seja independente de detalhes relevantes à somente a algumas implementações particulares.

Para suprir as deficiências apresentadas pelo SNMP original, uma nova versão foi apresentada. O SNMP versão 2 (SNMPv2) provê mecanismos de recuperação de grandes quantidades de informação de gerenciamento e facilidades de segurança (Brisa, 1994).

A versão 3 do SNMP é uma versão que apresenta uma proposta de solução para o problema de segurança encontrado nas versões anteriores do protocolo. As propriedades de segurança abordadas são(Menezes, 1998):

a) Autenticação

- Permite a um agente verificar se uma solicitação está vindo de um gerente autorizado e a integridade do seu conteúdo.

b) Criptografar

- Permite gerentes e agentes a criptografarem mensagens para evitar invasão de terceiros.

c) Controle de Acesso

- Torna possível configurar agentes para oferecerem diferentes níveis de acesso a diferentes gerentes.

Segundo Brisa (1994), o gerente solicita informações dos agentes ou envia comandos para que eles alterem alguma situação referente a algum componente de rede que está sendo gerenciado . A troca de mensagens entre os sistemas agentes e gerente é efetuada mediante a utilização de mecanismos de autenticação para fins de segurança. Tal troca de mensagens é realizada através do protocolo SNMP, que especifica o conteúdo e o formato de tais mensagens, bem como a seqüência correta das mensagens trocadas.

O SNMP inclui a definição de um conjunto de operações de gerenciamento para : acessar o valor de uma variável da MIB, conhecendo ou não o seu nome; responder a uma operação de acesso; armazenar o valor de uma variável e, ainda informar ao gerente sobre a ocorrência de um evento pré determinado.

O SNMP é o protocolo de gerência recomendado para o gerenciamento de redes TCP/IP. O SNMP é um protocolo de gerência definido à nível de aplicação, utilizando os

serviços do protocolo de transporte UDP para enviar suas mensagens através da rede. Sua especificação está contida no RFC 1157. Este protocolo é o centro do desenvolvimento do gerenciamento SNMP (Chesani, 1995).

Segundo Brisa (1994) o sistema de gerenciamento de arquitetura Internet foi especificado para permitir a detecção e a correção de falhas na rede, o controle de componentes da rede. A verificação de eventual violação dos protocolos, entre outros. Dentro desse sistema, de maneira análoga ao OSI, são definidos dois papéis correspondentes ao agente, que é responsável pela coleta de informações de gerenciamento, e ao gerente, que deve processá-las.

Na arquitetura Internet, define-se ainda o conceito de agente *proxy* .Este tipo de agente é autorizado a responder às operações de gerenciamento definidas para o SNMP, que são executadas mediante à comunicação deste agente com o sistema gerenciado através de protocolos particulares.

No modelo de gerenciamento da Internet em cada camada são definidas entidade *Layer Management Entity* (LME) responsáveis pelo seu gerenciamento. Neste caso, a obtenção de informações de gerenciamento e seu posterior armazenamento na MIB ficam a cargo de um elemento de software qualquer não padronizado, e especificado em nível de implementação. De maneira análoga ao modelo OSI, as informações contidas na MIB são lidas, alteradas e transferidas a partir de operações de gerenciamento especificadas em mensagens do protocolo SNMP transmitidas pelo gerente. Tal protocolo é usado para transportar também o resultado destas operações dos agentes ao gerente.

O modelo de gerenciamento consiste em um esquema centralizado, isto é, uma estação (*host*) é configurada como gerente e os demais elementos da rede desempenham o papel de agentes ou *proxy* agentes. Um *proxy* agente serve de procurador para aqueles equipamentos que não implementam SNMP. Cada agente possui uma MIB que contem as variáveis relativas aos objetos gerenciados. O modelo genérico compreende três componentes :

- um conjunto de objetos gerenciados, correspondente a um agente e a uma MIB associada;
- uma estação de gerenciamento de rede;

- um protocolo de gerenciamento de rede que é usado pela estação gerente e pelos agentes na troca de informações de gerenciamento.

Um objeto gerenciado representa um recurso que pode ser classificado na categoria de sistema hospedeiro (estação de trabalho, servidor de terminal ou impressora), sistema *gateway* (roteadores) ou equipamentos de meio (*modem, bridge, hub* ou multiplexador).

A estação gerente corresponde a um sistema hospedeiro que executa as aplicações de gerenciamento e o protocolo de gerenciamento de rede, tomando as decisões de acordo com as informações obtidas junto ao agente.

O protocolo de gerenciamento é visto sob o paradigma de observação remota, isto é, ela não transporta simplesmente operações de gerenciamento que devem ser executadas pelos objetos gerenciados, cada objeto é visto como uma coleção de variáveis cujo valor pode ser lido ou alterado, possibilitando, assim, a monitoração e o controle da cada elemento da rede.

O agente, quando solicitado pelo gerente, encaminha as informações ou altera valores das variáveis que representam os objetos gerenciados. O agente pode, ainda, avisar o gerente da ocorrência de algum evento não previsto, encaminhando estes avisos na forma de *traps*.

Em sua nova versão, o SNMP admite a existência de um gerenciamento distribuído, com estações configuradas para exercer o papel de gerentes e agentes, e com a possibilidade de comunicação entre gerentes para troca de informações de gerenciamento. Cada gerente pode gerenciar diretamente um conjunto de agentes e, quando o número de agentes cresce ao pontode causar problemas relativos ao seu gerenciamento, a estação servidora pode delegar a tarefa de gerenciamento a gerenciadores intermediários. O gerente intermediário exerce o papel de gerente para monitorar e controlar os agentes sob sua responsabilidade e exerce o papel de agente para enviar e receber informações de controle de seu servidor de gerenciamento hierarquicamente superior.

Segundo Microsoft (1997) o agente é um programa SNMP que deve ser instalado em cada computador gerenciado pelo SNMP. O programa agente fornece uma interface para as MIBs e para os objetos monitorados instalados no computador. Os programas de gerenciamento SNMP enviam solicitações de gerenciamento aos computadores da rede. O programa agente no computador recebe as solicitações e as processa recuperando as

informações dos MIBs no computador. O agente então envia as informações solicitadas de volta para o programa de gerenciamento SNMP que iniciou a solicitação.

Os programas de gerenciamento do SNMP são denominados de gerentes. Os gerentes obtêm dados de dispositivos de rede e tornam estas informações disponíveis para um administrador de rede por meio de interfaces de usuário textuais, gráficas ou orientadas à objetos. O programa gerente envia mensagens SNMP para *hosts* da rede. Essas mensagens são recebidas pelo agente no *host*, e iniciam as operações *get*, *get-next* e *set* . O programa gerente espera (escuta) pelas mensagens do SNMP do agente que contém os resultados da operação e exibe as informações no console de gerenciamento SNMP ou salva os dados em um arquivo de banco de dados específico.

3.1 OPERAÇÕES SUPORTADAS PELO SNMP

Segundo Brisa (1993) o protocolo SNMP é baseado no paradigma conhecido como "busca armazenamento" (*fetch-store*),isto é, todas as operações previstas para este protocolo são derivadas de operações básicas de busca e armazenamento. Estas operações básicas incluem:

- *get-request*: leitura do valor de uma variável;
- *get-next request*: leitura do valor da próxima variável;
- *get-response*: resposta à operação de leitura (*get-request on get-next –request*);
- *set-request*: gravação do valor de uma variável;
- *trap*: notificação da ocorrência de um evento específico.

No caso da operação de *trap*, deve-se observar que os eventos que, normalmente, geram notificação são predefinidos e correspondem à erros, falhas ou operações normais do sistema.

Essas 5 operações estão presentes tanto no SNMPv1 quanto no SNMPv2; para o SNMPv2, existe outras duas operações que oferecem maior flexibilidade na recuperação da informação de gerenciamento e na distribuição das atividades de gerenciamento (Brisa, 1994) :

- *get-bulk-request*: permite a recuperação da informação de múltiplos valores através de uma única troca de PDU, cobrindo uma deficiência do SNMPv1 na recuperação de grandes blocos de informações de gerenciamento;
- *inform-request*: possibilita que um gerente encaminhe informações de gerenciamento para outro gerente, suportando, assim, em esquema distribuído de gerenciamento de rede.

Uma operação *get* ou *set* somente se refere a uma única instância de um objeto representada através de seu nome. No protocolo SNMP, as operações são atômicas, isto é, todas as operações de um pedido devem ser executadas. Não existem execuções parciais de um pedido (no caso, operações aplicadas a múltiplos objetos). Se ocorrer algum erro durante a execução de uma operação, os resultados produzidos por esta operação devem ser ignorados (Sztajnberg, 1996).

3.2 MENSAGENS

As mensagens deste protocolo não possuem campos fixos e são especificados na notação ASN.1. Elas consistem em três partes principais: versão de protocolo, identificador da comunidade SNMP e área de dados .Para cada uma das operações , é definido um tipo específico de mensagem de protocolo, isto é um tipo de PDU. Desta maneira tem-se : *GetRequestPDU*, *GetNextRequestPDU*, *GetResponsePDU*, *SetRequestPDU* e *TrapPDU*, (Brisa, 1993).

Segundo Sztajnberg (1996) ao contrário de muitos outros protocolos TCP/IP, as mensagens no protocolo SNMP além de não apresentarem campos fixos, são codificadas usando a sintaxe ASN.1 (tanto a mensagem de pedido, como a de resposta) o que dificulta o entendimento e a decodificação das mensagens.

As partes mais importantes de uma mensagem são: as operações (*get*, *set* e *get-next*) e a identificação, no formato ASN.1, dos objetos em que as operações devem ser aplicadas.

Deve existir um cabeçalho que informe o tamanho da mensagem, que só será conhecido após a representação de cada campo ter sido computada. Na verdade, o tamanho da

mensagem depende do tamanho de sua parte remanescente (que contém os dados), portanto o tamanho só poderá ser computado após a construção da mensagem. Uma maneira de evitar este problema é construir a mensagem de trás para frente.

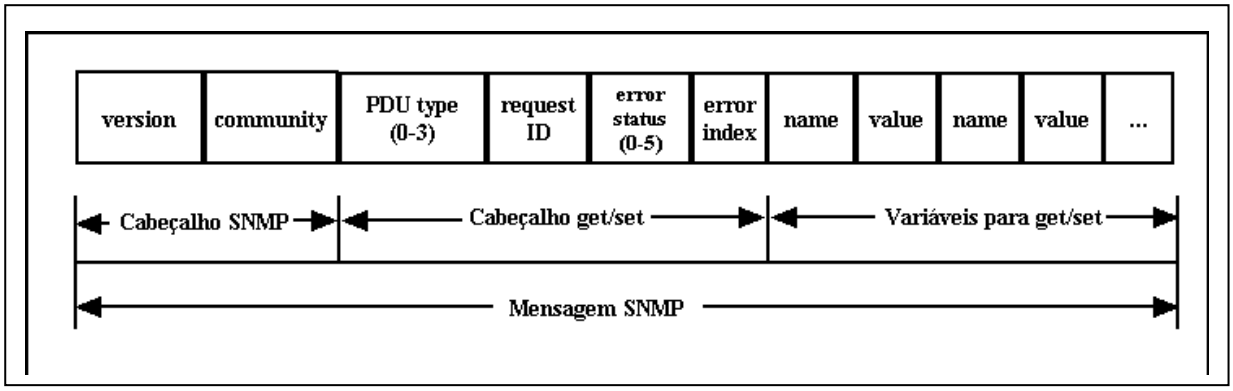
Uma mensagem SNMP deve definir o servidor do qual obtém-se ou altera-se os atributos dos objetos, e que será responsável por converter as operações requisitadas em operações sobre as estruturas de dados locais. Após verificar os campos de uma mensagem, o servidor deve usar as estruturas internas disponíveis para interpretar a mensagem e enviar a resposta da operação ao cliente que requisitou o pedido.

Uma mensagem é constituída por três partes principais:

- A versão do protocolo;
- A identificação da comunidade, usada para permitir que um cliente acesse os objetos gerenciados através de um servidor SNMP;
- A área de dados, que é dividida em unidades de dados de protocolo (PDU). Cada PDU é constituída ou por um pedido do cliente, ou por uma resposta de um pedido (enviada pelo servidor).

A figura abaixo mostra a partes de uma mensagem SNMP:

Figura 1 - Formato da mensagem SNMP



Fonte: Rekowsky (1999).

O primeiro campo de uma mensagem SNMP é um operador seqüencial, seguido por um campo com o tamanho total da mensagem (se este tamanho não for igual ao do datagrama,

será retornado um código de erro). O próximo campo é um número inteiro que identifica a versão do protocolo SNMP, seguido por um campo usado para a autenticação, indicando a comunidade que o cliente pertence (a comunidade *public* permite a qualquer cliente acessar os objetos, não precisando o servidor verificar se o cliente pode ou não acessar o objeto). O quarto campo contem a operação que será executada, devendo ser um *get*, *set* ou *get-next* pois a operação de *trap* só é gerada pelo servidor. O quinto campo é usado para o servidor ter certeza de que o valor deste campo é igual ao tamanho da parte da mensagem que contém os dados. O sexto campo é uma identificação para o pedido, e o sétimo e o oitavo campos são *flags* que indicam erros quando estão setadas (campos de *status* e de índice de erro).

Na definição de uma mensagem, cada uma das PDUs são constituídas ou por um dos cinco tipos de PDUs para as operações ou por uma PDU para a resposta. Na definição da mensagem SNMP, deve-se ter uma sintaxe individual para cada um das cinco operações da PDU. Alguns termos encontrados nas sintaxes das PDUs das operações são:

- O campo *RequestID* é um inteiro de 4 bytes (usado para identificar as respostas);
- Os campos *ErrorStatus* e *ErrorLevel* são inteiros de um byte (sendo nulos em um pedido de um cliente);
- O campo *VarBindList* é uma lista de identificadores de objetos na qual o servidor procura os nomes dos objetos, sendo definida como uma seqüência de pares contendo os nomes dos objetos (em ASN.1 este par é representado como uma seqüência de dois itens). Na sua forma mais simples (com um objeto) apresenta dois itens: o nome do objeto e um ponteiro nulo.

Os agentes SNMP executam operações elementares, como estabelecer e obter valores das variáveis. O programa que analisa, manipula, combina ou aplica algum algoritmo sobre os dados que devem residir no monitor (gerente) (Rekowsky, 1999).

3.3 MIB

Para garantir a interoperabilidade entre diferentes sistemas de gerenciamento de rede, tais sistemas precisam ter uma visão comum da informação de gerenciamento. Isto implica definir uma estrutura de informação de gerenciamento SMI que especifica o modelo de

informação a ser adotado. Este modelo deve incluir a definição da estrutura da informação de gerenciamento armazenadas em bases de dados destinadas a este fim , as operações que podem ser realizadas sobre a mesma e as notificações que podem ser emitidas em decorrência de alguma operação ou alteração destas informações (Brisa, 1993).

As informações de gerenciamento são armazenadas em uma base de dados denominada MIB, que contém informações classificadas em categorias referentes a sistemas *hosts* e *gateways*, *interfaces* individuais de rede, tradução de endereços e softwares relativos ao IP, TCP e UDP .

As MIBs são especificadas usando a notação ASN.1 padronizada no contexto do modelo OSI (Brisa, 1994).

Todo sistema complexo necessita armazenar as informações manipuladas em algum tipo de base de dados. A Base de Informação Gerencial MIB é o nome conceitual para a informação de gerenciamento, incluindo os objetos gerenciados e seus atributos. Pode-se considerar as informações para a configuração do sistema como também pertencentes à MIB (Sztajnberg, 1996).

Segundo Brisa (1993) a SMI descreve o cenário no qual a base de informação gerencial pode ser definida. A SMI, baseada na abordagem orientada a objetos, introduz os conceitos de hierarquia, herança, nomeação e registros usados na caracterização e identificação de objetos gerenciados. Além disso, ela define o conjunto de operações que podem ser realizadas sobre os objetos gerenciados da MIB e o comportamento desses objetos mediante a execução destas operações.

Dentro deste contexto, a MIB é definida como um conjunto de objetos gerenciados dentro de um sistema aberto, na qual um objeto gerenciado é a visão abstrata de um recurso real dentro deste sistema.

O RFC 1066 apresentou a primeira versão da MIB para uso com o protocolo TCP/IP, a MIB-I. Este padrão designou a base de informação necessária para monitorar e controlar redes baseadas no protocolo TCP/IP. O RFC 1066 foi aceito pela IAB (Internet Activities Board) como padrão no RFC 1156.

O RFC 1158 propôs uma segunda MIB, a MIB-II, para uso com o protocolo TCP/IP, sendo aceita e formalizada como padrão no RFC 1213. A MIB-II expandiu a base de informações definidas na MIB-I.

No padrão Internet os objetos gerenciados são definidos em uma árvore de registro, equivalente a hierarquia de registro do padrão OSI.

A MIB II usa uma arquitetura de árvore, definida na ISO ASN.1, para organizar todas as suas informações. Cada parte da informação da árvore é um nó rotulado que contém:

- um identificador de objetos (*Objetct Identifier* OID): seqüência de números separados por pontos;
- uma pequena descrição textual: descrição o nó rotulado.

A MIB da Internet define os objetos que podem ser gerenciados por cada camada do protocolo TCP/IP. Estes objetos estão sob a guarda de um agente de gerenciamento e a comunicação entre este agente e um gerente, localizado na estação de gerenciamento é feita utilizando o protocolo SNMP (Sztajnberg, 1996).

Segundo Zacker (2000) os MIBs (versões I e II) definem os objetos que estão contidos no software agente. Quando os padrões para o SNMP foram escritos, foi determinado que precisava haver um formato padrão para as informações contidas em um agente que fizesse interface com o protocolo de gerenciamento de rede. Este formato padrão foi definido como SMI. Se pensar no MIB como um banco de dados de itens a serem gerenciados, o SNMP será uma estrutura , ou esquema, desse banco de dados. O MIB é uma árvore hierárquica que contem definições de uma lista padrão de funções ou características a serem gerenciadas no dispositivo. Estas funções características são chamadas de objetos. Se seguirmos o exemplo de banco de dados, eles também podem ser imaginados como campos no banco de dados . Cada objeto, ou campo , pode ser tomado como um valor, dependendo do estado do objeto no dispositivo gerenciado.

Cada objeto na MIB apresenta uma série de características que permitem à ele trabalhar com SNMP para fornecer suas quatro funções básicas. As características comuns à todo objeto são:

a) *ACCESS*, que define os direitos de acesso ao objeto da MIB;

b) *DESCRIPTION*, que descreve o que o objeto oferece;

c) *STATUS*, que indica se o objeto pode ser implementado neste agente MIB;

d) *SYNTAX*, que descreve qual deve ser o formato apropriado para o valor do objeto.

ACCESS, pode assumir um dos quatro valores possíveis:

a) *read-only*, este objeto somente pode ser lido;

b) *read-write*, este objeto pode ser lido e configurado;

c) *write-only*, este objeto pode ser configurado, mas não lido;

d) *not-accessible*, este objeto não pode ser configurado nem lido.

STATUS indica se o objeto pode ser implementado neste agente MIB – ou seja , se um agente deve controlar ou não um item particular no dispositivo que está monitorando. Este item assume os seguintes valores:

a) *mandatory*, o agente deve implementar o objeto;

b) *Optional*, o agente pode implementar o objeto;

c) *Obsolete*, o objeto não é mais necessário.

Segundo Sztajnberg (1996), basicamente, são definidas quatro tipos de MIB: MIB I, MIB II, MIB experimental e MIB privada. As MIBs do tipo I e II fornecem informações gerais sobre o equipamento gerenciado, sem levar em conta as características específicas deste equipamento. A MIB II, em verdade, é uma evolução da MIB I, que introduziu novas informações além daquelas encontradas na MIB I.

As MIB experimentais são aquelas que estão em fase de testes, com a perspectiva de serem adicionadas ao padrão e que, em geral, fornecem características mais específicas sobre a tecnologia dos meios de transmissão e equipamentos empregados.

As MIB privadas são específicas dos equipamentos gerenciados, possibilitando que detalhes peculiares a um determinado equipamento possam ser obtidos. É desta forma que é

possível se obter informações sobre colisões, configuração, swap de portas, e muitas outras, de um *hub*. Também é possível fazer um teste, reinicialização ou desabilitar uma ou mais portas do *hub* através de MIB proprietárias

A MIB divide os objetos em vários grupos. O quadro seguir mostra a MIB-I e os grupos nela definidos.

Quadro 1 - Número de objetos nos grupos da MIB-I

Grupo	Objetos para	#
system	informações básicas do sistema	3
interfaces	interfaces de rede	22
at	tradução de endereço	3
ip	software de protocolo IP	33
icmp	protocolo de estatíst. para contr.interno de msgs.	26
tcp	software de protocolo TCP	17
udp	software de protocolo UDP	4
egp	software de protocolo EGP	6

Fonte: Sztajnberg (1996)

O quadro a seguir mostra a MIB-II e os grupos nela definidos.

Quadro 2 - Número de objetos nos grupos da MIB-II

Grupo	Objetos para	#
system	informações básicas do sistema	7
interfaces	interfaces de rede	23
at	tradução de endereço	3
ip	software de protocolo IP	38
icmp	protocolo de estat. para contr. interno de msgs.	26
tcp	software de protocolo TCP	19
udp	software de protocolo UDP	7
egp	software de protocolo EGP	18
transmiss	transmissão. Média-específica	0
snmp	aplicações snmp	30

Fonte: Sztajnberg (1996)

A lista definida de objetos gerenciáveis foi derivada daqueles elementos considerados essenciais. Esta implementação de se pegar apenas objetos essenciais não é restrita, uma vez que a SMI proporciona mecanismos de extensão como uma nova versão de uma MIB e uma definição de um objeto privado ou que não seja padrão.

A seguir, são listados alguns exemplos de objetos de alguns grupos:

a) Grupo System

- sysDescr : completa descrição do sistema (versão, hardware, sistema operacional);
- sysObjectID : objeto para identificação do vendedor;
- sysUpTime : tempo desde a última reinicialização;
- sysContact : nome da pessoa de contato;
- sysServices : serviços oferecidos pelo dispositivo.

b) Grupo IP

- ipForwarding : indica se esta entidade é um gateway IP;
- ipInHdrErrors : número de datagramas recebidos descartados devido a erros em seu cabeçalho IP ;
- ipInAddrErrors : número de datagramas recebidos descartados devido a erros em seu endereço IP ;
- ipReasmOKs : número de datagramas IP remontados com sucesso;
- ipRouteMask : máscara de sub-rede para rota .

c) Grupo TCP

- tcpRtoAlgorithm : Algoritmo para determinar o *timeout* para retransmissão de um datagrama desconhecido ;
- tcpMaxconn : limite do número de conexões TCP que a entidade pode sustentar ;
- tcpInSegs : Número de segmentos recebidos incluindo aqueles recebidos com erro ;
- cpConnRemAddress : o endereço remoto IP para determinada conexão TCP ;
- tcpInErrs : número de segmentos descartados devido ao formato de erro ;
- tcpOutRsts : número de reinicializações geradas.

d) Grupo UDP

- udpInDatagrams : número de datagramas UDP entregues aos usuário UDP ;
- udpNoPorts : número de datagramas UDP recebidos para aqueles onde não existe aplicação para aquela porta de destino ;

- udpInErrors : número de datagramas UDP recebidos que não podem ser entregues por diversas razões, menos a falta de uma aplicação para a porta de destino ;
- udpOutDatagrams : número de datagrama UDP enviados por esta entidade.

e) Grupo Interfaces

- ifIndex : número da interface ;
- ifDescr : descrição da interface ;
- if Type : tipo da interface ;
- ifMtu : tamanho do maior datagrama IP ;
- ifAdminisStatus : status da interface ;
- ifLastChange : hora em que a interface inicializou o status corrente.

Para finalizar, a MIB da internet não inclui informações de gerenciamento para aplicações tais como: acesso a terminal remoto (TELNET), transferência de arquivos *File Transfer Protocol* (FTP) e correio eletrônico *Simple Mail Transfer Protocol* (SMTP).

Segundo Microsoft (1997) uma MIB é um arquivo de dados que contém a descrição dos objetos monitorados e os valores dos objetos. Cada *host* que deve ser monitorado pelo SNMP tem um MIB que descreve os objetos monitorados neste *host*.

Basicamente , um MIB definirá o seguinte para cada objeto contido dentro dele :

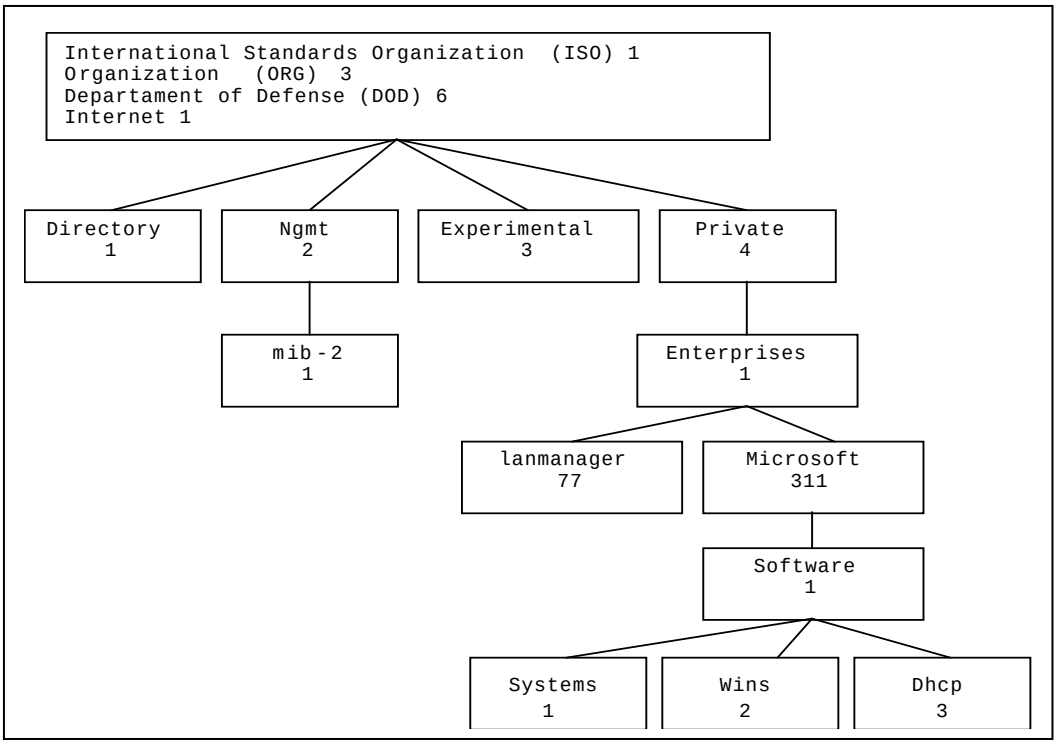
- A associação entre (1) o *hardware* do *host* ou componente de *software* (objeto) e (2) um nome de objeto e um identificador de objeto;
- Uma definição do tipo de dados usados para definir o objeto;
- Uma descrição textual do objeto;
- Um método de índice usado para os objetos que são um tipo de dados complexos;
- O acesso de leitura ou gravação que é permitido no objeto.

O RFC 1213 define um padrão industrial do MIB do SNMP referenciado como MIB-II. Os fabricantes da indústria , como a Microsoft, podem definir MIB adicionais que permitem que *hardware* únicos e serviços de software desenvolvidos pelo fabricante sejam monitorados e gerenciados pelos consoles de gerenciamento SNMP.

Cada objeto da MIB é um identificado por um único rótulo universalmente conhecido como um OID. O espaço do nome do objeto é implementado como um esquema de nomeação hierárquico de múltiplas partes . Um esquema de nomeação hierárquico pode ser visualizado como uma árvore invertida com os galhos apontados para baixo. Cada ponto onde um novo galho é adicionado é referenciado como um nó. Esse OID é internacionalmente aceito e permite que os desenvolvedores e os fabricantes criem novos componentes e recursos e atribuam um único OID para cada novo componente ou recurso.

O esquema de nomeação OID á administrado pela *Internet Engeneering Task Force* (IETF). O IETF concede autorização para partes do espaço do nome para organizações individuais, tais como a Microsoft que, por exemplo tem autorização para atribuir os OID que podem ser derivados para ramificações descendentes a partir do nó da árvore de nomes MIB que começa em *1.3.6.1.4.1.311*. Ver figura abaixo:

Figura 2 - Hierarquia de nomes de objetos monitorados



Fonte: Microsoft (1997).

4 ARQUIVOS LOG

O *log* tem como função servir de repositório de registros. Estes registros por sua vez, contêm informações que devem ser preservadas. Um *log* armazena os registros por ordem de chegada e os identificadores de registro são atribuídos em seqüência numérica.

O comportamento de um *log* é descrito através de seus atributos de estado (administrativo, operacional e de utilização), de seu status disponibilidade, de seus pacotes de programação e de seu construtor de discriminador (Brisa, 1994).

Segundo Sztajnberg (1996) o objetivo de função de controle de *log* é o de permitir as demais funções de gerenciamento, preservar informações sobre os eventos que ocorreram, ou sobre as operações executadas nos objetos gerenciados. Uma vez que estas informações podem mudar, a função de controle de *log* deve satisfazer as seguintes características:

- Controle de *log* deve ser flexível para permitir a seleção de quais registros do *log* devem ser preservados pelo sistema de gerenciamento;
- Deve permitir que um sistema externo altere os critérios usados na preservação dos registros;
- Deve permitir a um sistema externo saber se foi alterada alguma característica de preservação, ou se um registro foi perdido;
- Definir mecanismos para controlar o tempo durante o qual devem ser realizadas as atividades de preservação das informações;
- Deve permitir que um sistema externo recupere e elimine os registros em um *log*, como também criar e eliminar *logs*.

4.1 ARQUIVOS LOG NO SERVIDOR WEB

Segundo Ablan (1996) a maioria dos servidores WEB gravam os acessos e os erros em arquivos de *log* específicos, estas opções são configuradas através de programas de administração do servidor WEB, ou alterando seus arquivos de configuração.

O arquivo *log* de acesso tipicamente chamado *access_log* ou *access* grava todas as requisições de acesso que servidor WEB recebe. O arquivo *log* de acesso é o arquivo mais importante para obtenção de estatísticas.

As estatísticas obtidas de um servidor WEB são de grande utilidade e importância para muitas empresas. Estas estatísticas podem responder uma série de questões como:

- Estão acessando de onde;
- O caminho informado para acesso ao site;
- O número de visitantes ao site;
- O número de visitas a determinada sessão ou página do site;
- A localização geográfica dos visitantes;
- O tipo de software que os visitantes estão usando para o acesso;
- Que problemas as pessoas estão tendo com o site;

A quantidade de WEB *sites* vem crescendo, e obter informações sobre os usuários que a visitam torna-se muito importante. As informações obtidas são de suma importância para identificação de problemas, determinar páginas ou áreas no *site* que são mais visitadas, e até fornecer melhor suporte para usuários.

Arquivos *log* precisam ser gerenciados, pois crescem de maneira muito rápida e podem ocupar muito espaço em disco devido a grande quantidade de visitas que podem receber. Recomenda-se alternar os arquivos de *log* ou arquiva-los a uma determinada freqüência.

Programas de estatística precisam ler e interpretar o conteúdo dos arquivos de *log*, e se estes programas tentarem ler muitas informações podem não ter memória suficiente disponível. Para que os programas de estatística interpretem as informações corretamente é necessário o gerenciamento dos arquivos de *log*.

Estatísticas podem ser obtidas baseadas num intervalo de tempo. Gerenciando os arquivos de *log* para que sejam alternados num determinado intervalo de tempo (por exemplo uma semana) torna se mais fácil e mais rápido ler o *log* e obter as estatísticas. Os arquivos de *log* também podem ser apagados num intervalo de tempo, ou compactados. Arquivar arquivos de *log* pode ser importante quando se há a necessidade de recuperar estatísticas mais antigas.

Segundo Trunfio (2001), os *logs* do servidor WEB armazenam informações das requisições de acesso à ele. Todos servidores WEB geram *logs* de acesso, que tipicamente são

concebidos num formato padrão denominado *common log file* que armazena cada requisição em uma linha. Uma entrada típica no *log* seria assim:

- *host ident authuser date request status bytes*

Neste exemplo, *host* indica o nome completo do domínio do cliente ou seu endereço IP, *ident* é a identidade do cliente se ele estiver rodando um servidor de identificação (maiores informações na RFC 931) com a opção *ident_lookup* habilitada, *authuser* contem a identificação do usuário se o cliente requisita um documento seguro, *date* informa a data no seguinte formato [data/mês/ano:hora:minuto:segundo zona], *request* é a requisição do cliente dada no seguinte formato "GET endereço HTTP/versão", *status* é o código do estado HTTP da requisição (códigos comuns de estado são 200, que indica que a requisição foi um sucesso, 302 indica um redirecionamento, 4xx, que indica um erro de acesso, por exemplo, 401 indica falha de autenticação do usuário, 403 indica acesso proibido e 404 é não encontrado, 5xx indica erro do servidor Web), e *bytes* indica o número de bytes enviados.

5 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo são apresentadas técnicas e ferramentas utilizadas no desenvolvimento do protótipo de um software gerenciador do servidor WEB utilizando o protocolo SNMP. O protótipo desenvolvido constitui-se de uma MIB com objetos específicos para finalidade de gerenciamento dos arquivos de *log* de um servidor Web, uma extensão do agente SNMP para controlar esta MIB e um gerente SNMP para o monitoramento deste agente.

Para especificação do protótipo foram utilizadas as seguintes ferramentas:

- Técnica de modelagem orientada à objetos, a *Unified Modeling Language* (UML);
- Ferramenta *Rational Rose*.

5.1 UML

A UML é uma tentativa de padronizar a modelagem orientada a objetos de forma que qualquer sistema, seja qual for o tipo, possa ser modelado corretamente, com consistência, fácil de se comunicar com outras aplicações, simples de ser atualizado e compreensível.

A UML é a linguagem padrão para especificar, visualizar, documentar e construir artefatos de um sistema e pode ser utilizada como todos os processos ao longo do ciclo de desenvolvimento e através de diferentes tecnologias de implementação.

Diante da diversidade de conceitos das diversas metodologias de orientação a objetos, Grady Booch, James Rumbaugh e Ivar Jacobson decidiram criar uma Linguagem de Modelagem Unificada.

Os objetivos da UML são:

- a modelagem de sistemas (não apenas software) usando os conceitos da orientação a objetos;
- estabelecer uma união fazendo com que métodos conceituais sejam também executáveis;
- criar uma linguagem de modelagem usável tanto pelo homem quanto pela máquina.

A UML está destinada a ser dominante, a linguagem de modelagem comum usada nas indústrias. Ela está totalmente baseada em conceitos e padrões extensivamente testados provenientes das metodologias existentes anteriormente, e também é muito bem documentada com toda a especificação da semântica da linguagem representada em meta-modelos.

Existem cinco fases no desenvolvimento de sistemas de software em UML: análise de requisitos, análise, projeto, programação e testes. Estas cinco fases não devem ser executadas nesta ordem, mas concomitantemente de forma que problemas detectados numa certa fase modifiquem e melhorem as fases desenvolvidas anteriormente de forma que o resultado global gere um produto de alta qualidade e performance.

As fases de análise de requisitos, análise e projetos utilizam-se em seu desenvolvimento cinco tipos de visões, nove tipos de diagramas e vários modelos de elementos que serão utilizados na criação dos diagramas e mecanismos gerais. Todos, em conjunto, especificam e exemplificam a definição do sistema, tanto no que diz respeito à funcionalidade estática quanto a dinâmica do desenvolvimento de um sistema.

5.1.1 DIAGRAMAS DA UML

Na UML, o modo para descrever os vários aspectos de modelagem, é através da notação definida pelos seus vários tipos de diagramas. Um diagrama é uma apresentação gráfica de uma coleção de elementos de modelo, freqüentemente mostrado como um gráfico conectado por arcos (relacionamentos) e vértices (outros elementos do modelo).

5.1.1.1 DIAGRAMA DE CASOS DE USO

Os casos de uso descrevem a funcionalidade do sistema percebida por atores externos. Um ator interage com o sistema podendo ser um usuário, dispositivo ou outro sistema.

5.1.1.2 DIAGRAMA DE CLASSE

Gráfico bidimensional de elementos de modelagem que pode conter tipos, pacotes, relacionamentos, instâncias, objetos e vínculos (conexão entre dois objetos). Um diagrama de classe denota a estrutura estática de um sistema e as classes representam coisas que são

manipuladas por este sistema. O diagrama é considerado estático pois a estrutura descrita é sempre válida em qualquer ponto no ciclo de vida do sistema. Um diagrama de objeto mostra um número de instâncias de classes, em vez de uma classe real, e apresenta o nome do objeto sublinhado dentro do retângulo de classe. Diagrama de objeto não tão importantes quanto os diagramas de classe embora possam eventualmente ser úteis para exemplificar diagramas de classe complexos.

5.1.1.3 DIAGRAMA DE SEQÜÊNCIA

Apresenta a interação de seqüência de tempo dos objetos que participam na interação. As duas dimensões de um diagrama de seqüência consistem na dimensão vertical (tempo) e na dimensão horizontal (objetos diferentes). O diagrama de seqüência mostra a colaboração dinâmica entre um número de objetos e o aspecto importante desse diagrama é mostrar a seqüência de mensagens enviadas entre objetos.

Maiores informações sobre UML são encontradas em Furlan (1998).

5.2 RATIONAL ROSE

Segundo Santos (2000) o Rational Rose é uma ferramenta de modelagem visual que dá suporte a modelagem orientada à objetos . Proveniente da empresa dos criadores da UML, é uma solução para:

- os analistas de sistema e de negócios ;
- arquitetos de sistemas ;
- analistas de dados e ‘modeladores’ de Banco de Dados ;
- desenvolvedores: Win, WEB/XML, Java, Unix, e de aplicações em tempo-real.

Maiores informações sobre Rational Rose são encontradas em Moro (2000).

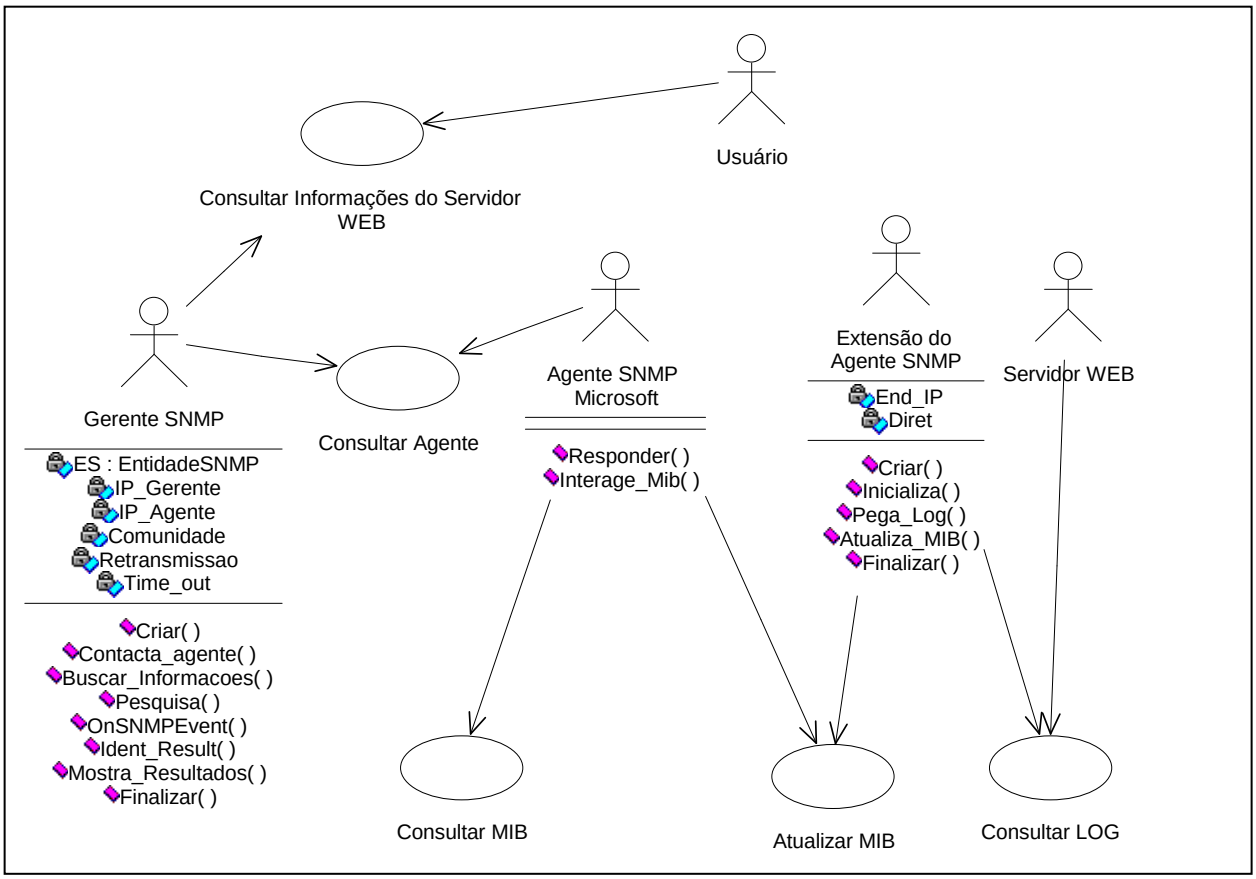
5.3 DIAGRAMAS DO PROTÓTIPO

Aqui serão apresentados os diagramas de casos de uso, de classe e de seqüência referentes a especificação do protótipo.

5.3.1 DIAGRAMA DE CASOS DE USO

O diagrama de casos de uso foi desenvolvido utilizando a notação da UML através da ferramenta *Rational Rose*. Na figura abaixo são apresentados os casos de uso principais do sistema.

Figura 3 - Diagrama de casos de uso



Os casos de uso definidos para o sistema são:

a) Consultar as informações do servidor WEB: o usuário interage com a aplicação gerente para consultar as informações sobre o servidor WEB;

b) Consultar agente: quando o gerente solicita informações do agente, ele informa o identificador de objeto e o agente SNMP Microsoft responde com o valor desse objeto;

c) Consultar a MIB: quando o agente SNMP da Microsoft recebe uma mensagem do gerente, então ele interage com a MIB e retorna a mensagem para o gerente;

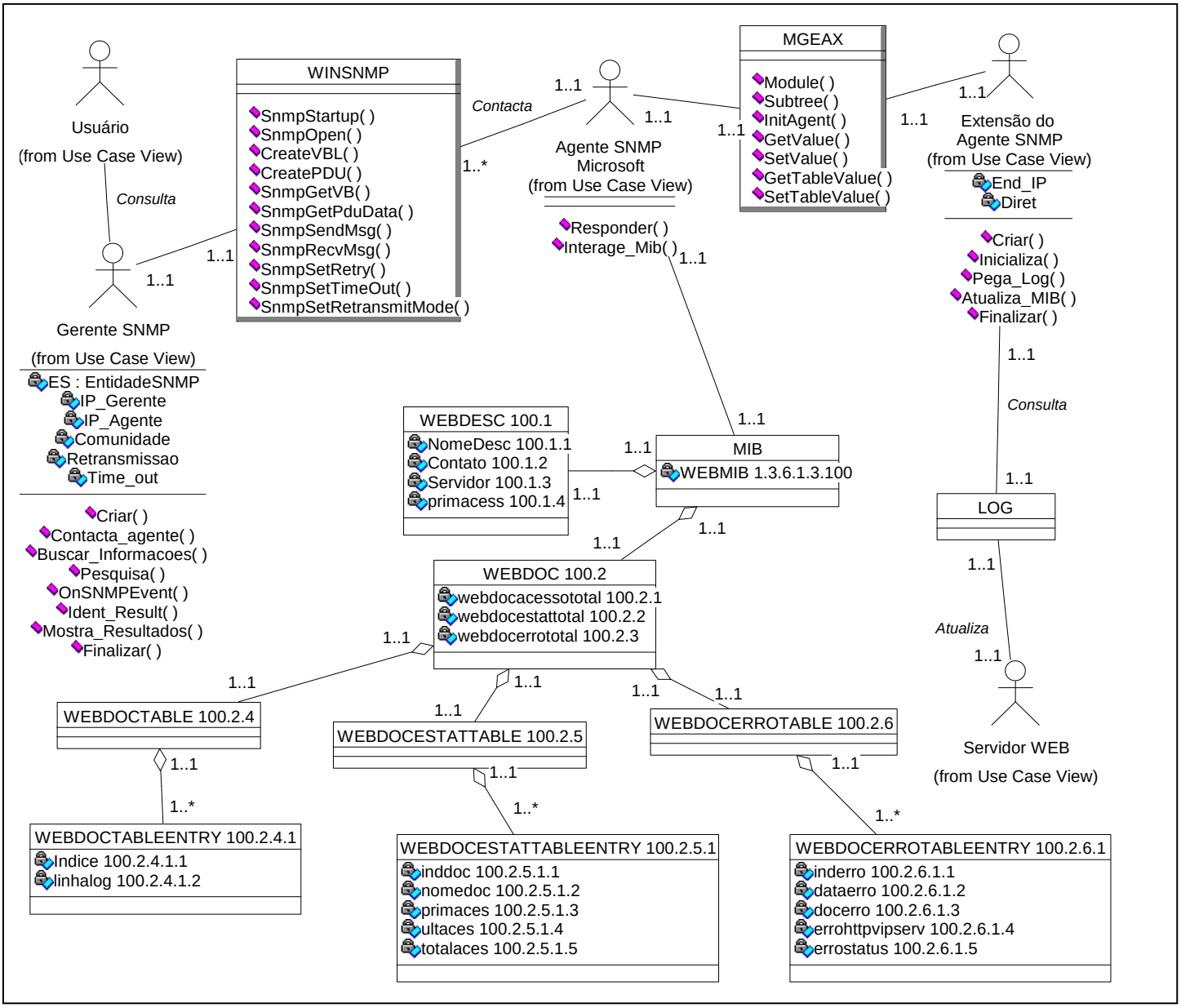
d) Consultar *LOG*: a aplicação extensão do agente SNMP consulta o arquivos de *LOG* do servidor Web;

e) Atualizar a MIB: as informações consultadas do arquivo de *LOG* são armazenadas na MIB, de acordo com a programação da extensão do agente SNMP.

5.3.2 DIAGRAMA DE CLASSES

Na fase de análise foram identificadas as classes principais do sistema, seus relacionamentos e atributos, conforme a figura abaixo:

Figura 4 - Diagrama de classes

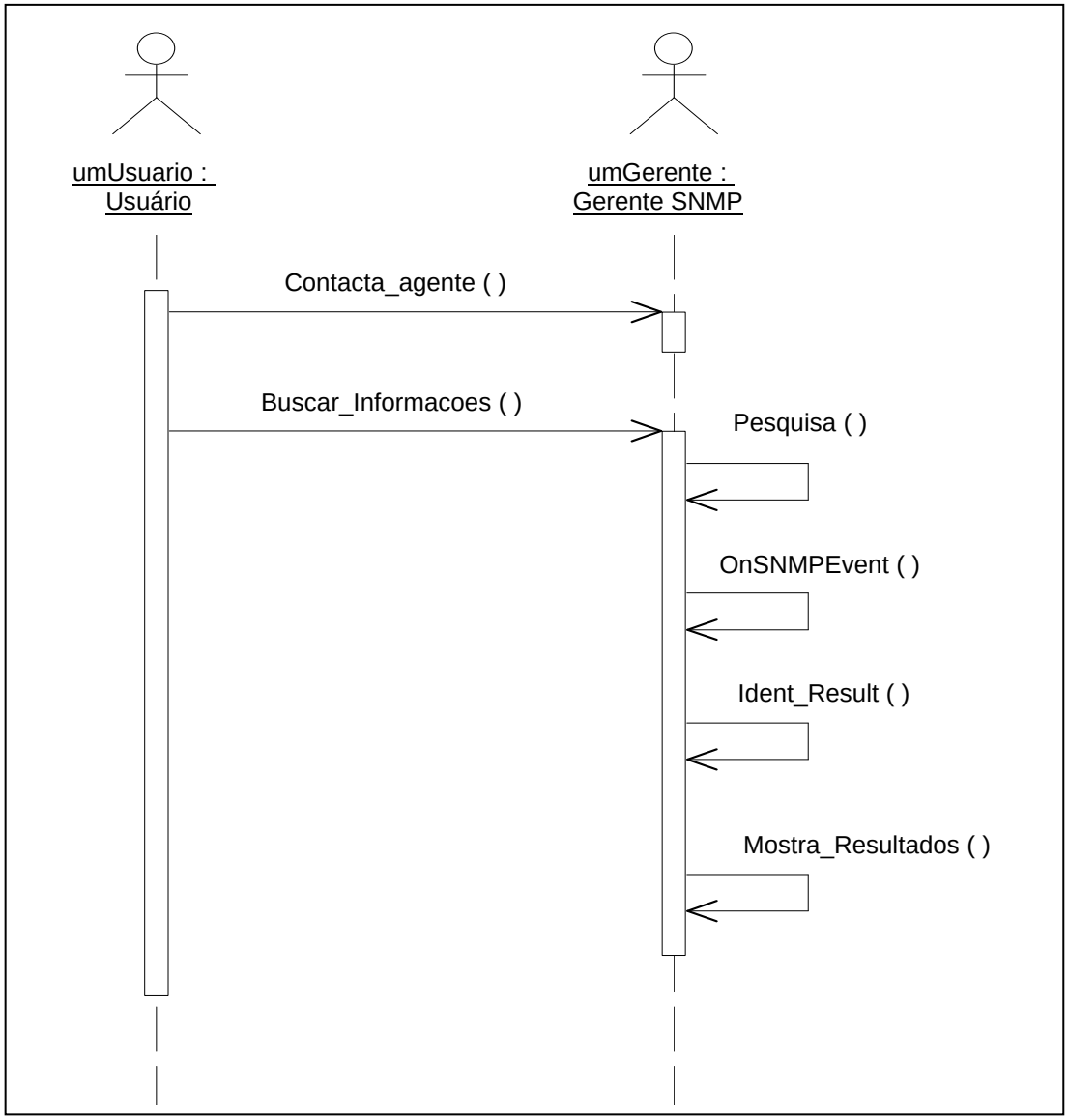


5.3.3 DIAGRAMA DE SEQÜÊNCIA

Os principais diagramas de seqüência encontrados no sistema são:

a) Consultar as informações do servidor WEB: o usuário interage com o gerente SNMP, informa o endereço IP do Agente SNMP, contata o agente, e após solicita as informações de estatísticas e erros, para isto o gerente SNMP informa o identificador de objetos para pesquisa e envia a mensagem SNMP, após isto espera o retorno da mensagem do agente, identifica o resultado e mostra o resultado de acordo com o identificador de objetos. Ver figura abaixo :

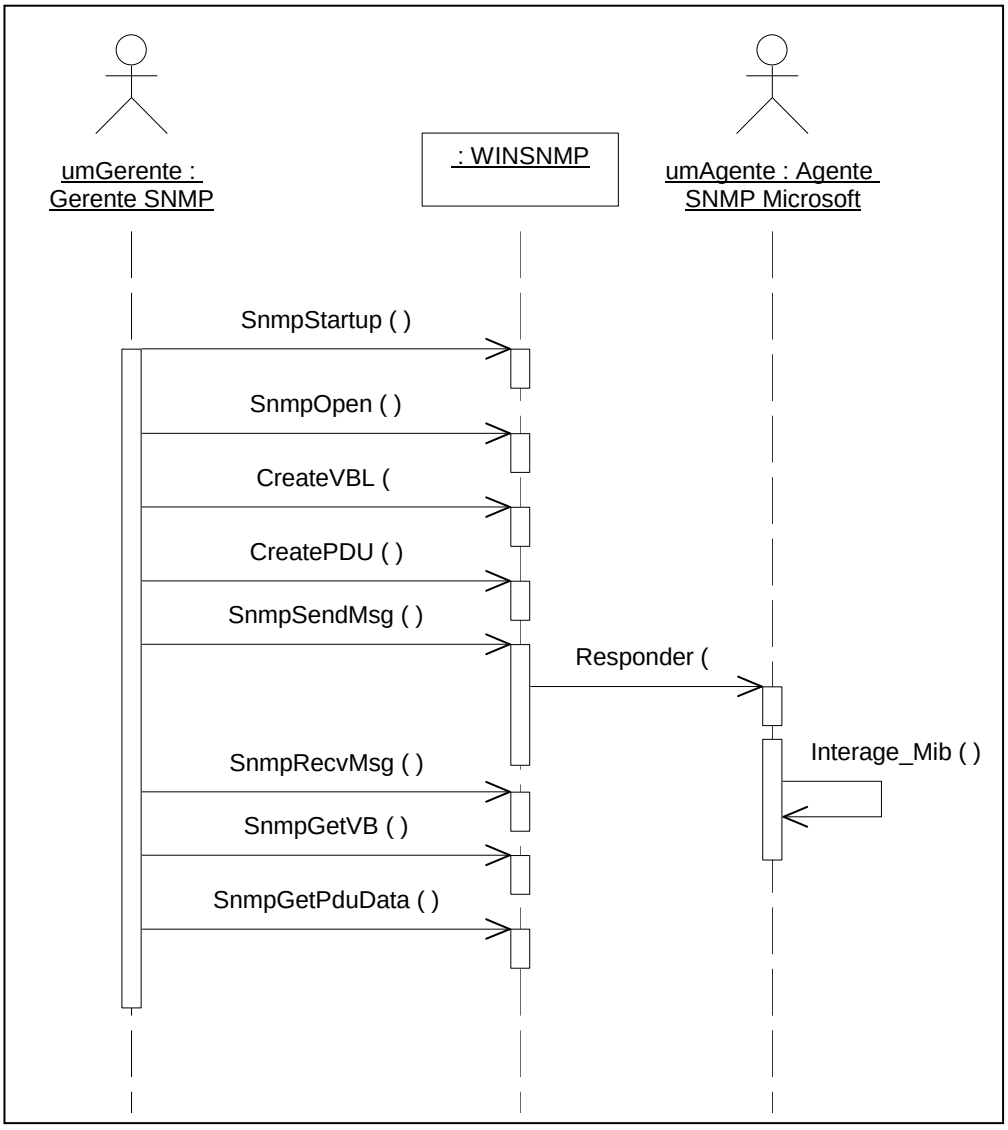
Figura 5 - Diagrama de seqüência - a



b) Consultar agente: o gerente SNMP através da a biblioteca de funções WINSNMP, executa uma série de operações como inicializar, abrir uma sessão SNMP, criar as partes necessárias para envio de uma mensagem SNMP, enviar a mensagem, receber a mensagem do agente SNMP da Microsoft, e extrair o valor da mensagem SNMP. Ver figura abaixo;

c) Consultar a MIB: quando o agente SNMP da Microsoft recebe uma mensagem do gerente SNMP, então ele interage com a MIB criada para o protótipo, e retorna a mensagem para o gerente SNMP. Ver figura abaixo:

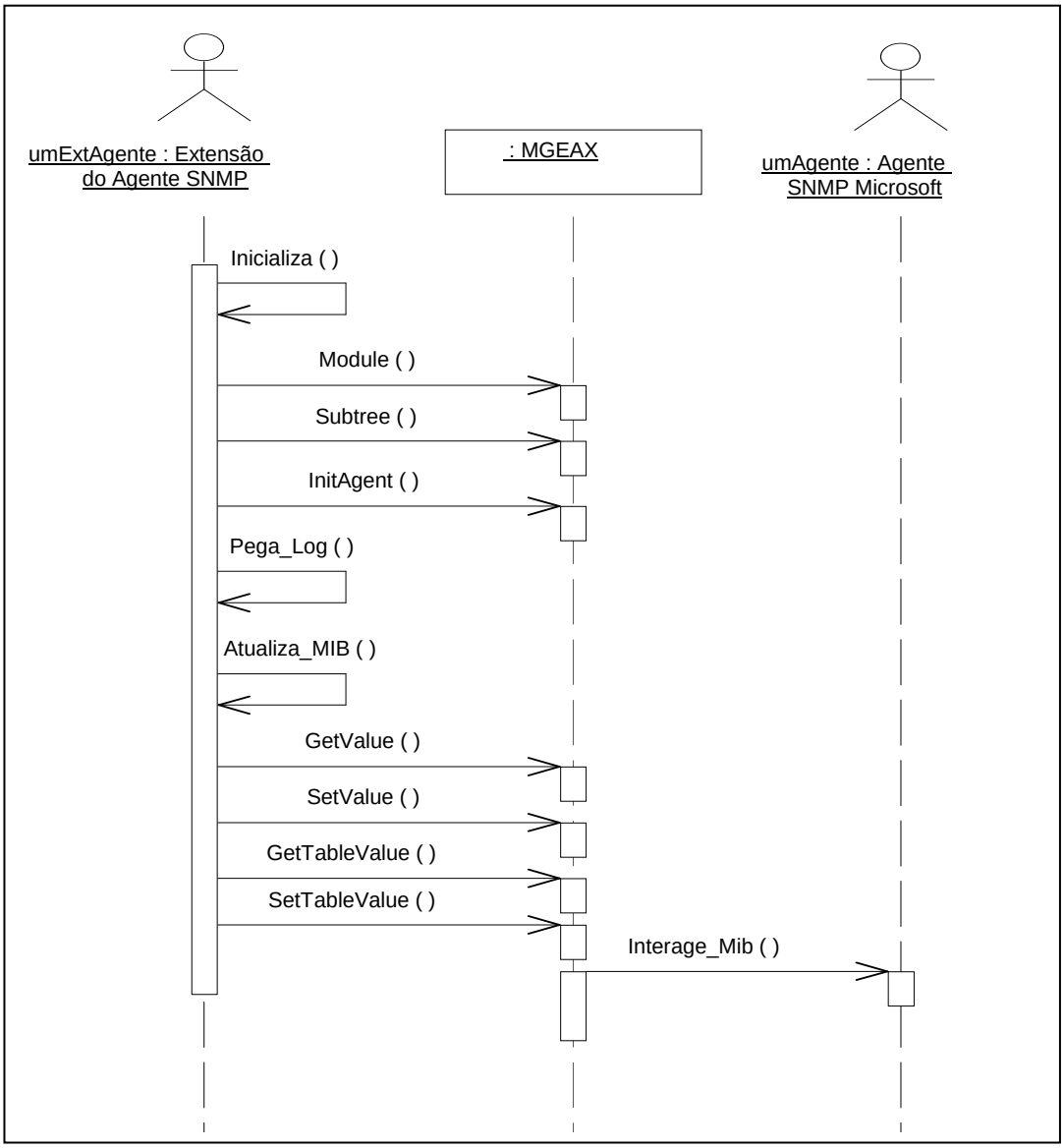
Figura 6 - Diagrama de seqüência – b,c



d) Consultar *log*: as requisições ao servidor Web são armazenadas em um arquivo de *log*, a um determinado intervalo de tempo a aplicação extensão do agente SNMP consulta estes arquivos de *log*. Ver figura abaixo:

e) Atualizar a MIB: a aplicação extensão do agente SNMP utilizando as funções de um componente MGEAX inicializa a extensão do agente SNMP, indica o módulo MIB para qual foi desenvolvido e respectivo identificador de objetos, após obter as informações do arquivo *log* dispara a função para atualizar a MIB, interagindo assim com o agente SNMP da Microsoft para atualização da MIB. Ver figura abaixo:

Figura 7 - Diagrama de seqüência – d,e



5.4 IMPLEMENTAÇÃO

Neste capítulo são apresentadas considerações sobre a implementação do protótipo.

Para a implementação foram as técnicas e ferramentas utilizadas para desenvolvimento do protótipo, sendo elas:

- sistema operacional Windows NT;
- o ambiente de programação Delphi;
- ferramentas para SNMP da MGSOFT.

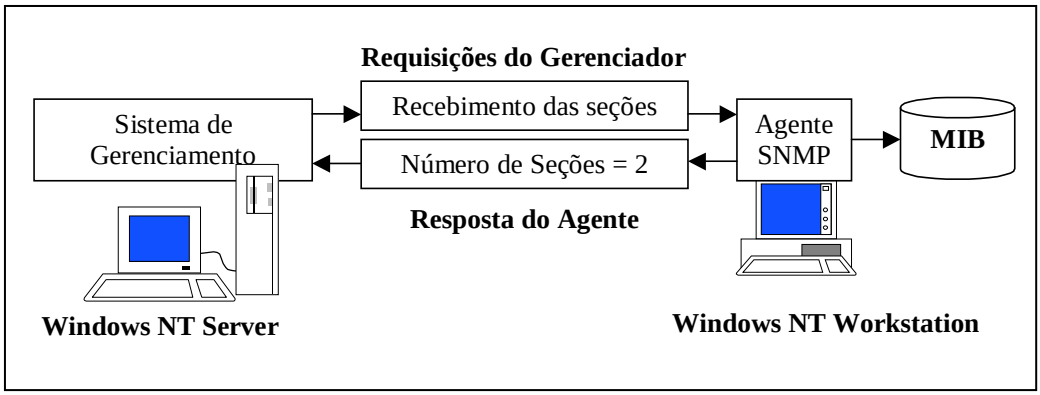
5.4.1 WINDOWS NT

O Agente SNMP baseado no Windows NT é implementado como um serviço e pode ser instalado nos computadores baseados no Windows NT que usam os protocolos TCP/IP e IPX. O protocolo TCP/IP deve ser instalado antes da instalação do SNMP.

O serviço SNMP é implementado como um serviço de 32 bits do Windows utilizando o Windows Sockets (porta virtual que permite a conexão entre servidor e clientes) sobre o TCP/IP e o IPX/SPX. As MIB adicionais da Microsoft para o DHCP, WINS e o Internet Information Server estendem o gerenciamento SNMP a esses serviços baseados em Windows NT. Os programas agentes que implementam essas MIB adicionais são chamados de agentes de extensão.

O diagrama a seguir mostra uma interação simples entre um computador gerenciador SNMP e um computador baseado no Windows NT com um programa agente do SNMP:

Figura 8 - Interação do gerenciador e do agente SNMP



Fonte: Microsoft (1997)

Maiores informações sobre Windows NT são encontradas em Microsoft (1997).

5.4.2 AMBIENTE DE PROGRAMAÇÃO DELPHI

O Delphi oferece uma base sólida na construção de aplicativos visuais apresentando muitas vantagens reais de produtividade para o programador. É um ambiente de programação visual desenvolvido pela empresa Borland, baseada na linguagem de implementação Object Pascal, utilizadas para a criação de aplicações para sistemas operacionais Windows, Mafinsky (1999).

O aumento do uso do sistema operacional Windows, com sua interface gráfica para o usuário *Graphical User Interface* (GUI), começava a trazer um ambiente de trabalho padrão para usuários de computadores e programadores. As interfaces puramente textuais estavam sendo cada vez mais substituídas pela interfaces gráficas. Estas interfaces gráficas eram mais fáceis de serem utilizadas, facilitando a vida de iniciantes e experientes com computadores, fazendo com que o usuário manuseasse o computador de forma mais intuitiva e não meramente por linha de comandos. O desenvolvimento de programas de quaisquer gêneros era uma tarefa demorada e em alguns casos muito longa. Tendo em foco principal o desenvolvimento a empresa Borland criou o ambiente de programação Delphi em 1995 O ambiente de programação Delphi utiliza a linguagem de programação *Object Oriented Pascal*, que é uma evolução do Pascal ANSI .

No Delphi, o desenvolvimento de aplicativos inicia com a montagem de componentes em janelas, como se fosse um programa gráfico. Estes componentes são auxiliares para o desenvolvimento da interface gráfica ao usuário e podem ser nativos (próprios do ambiente) ou desenvolvidos por terceiros (*shareware*) e distribuídos de graça (*freeware*) ou através do pagamento de uma taxa, como se fossem programas. Os usuários podem criar novos componentes e utilizá-los conforme surgirem as necessidades para cada um

Maiores informações sobre o ambiente de programação Delphi podem ser encontradas em Rekowsky (1999), Cantú (1998) e Carvalho (1998).

5.4.3 FERRAMENTAS DA MGSOFT

O software SNMP-Lab desenvolvido pela MGSOFT distribuído como *shareware* contém as ferramentas WinSNMP SDK, WinMIB SDK, SNMP EasyAgent SDK e MG-SOFT MIB Compiler. Também contém numerosos exemplos de código de fonte que ilustram o WinSNMP, WinMIB e SNMP EasyAgent e usos de *Aplication Program Interfaces* (APIs). Códigos podem servir como base para desenvolver uma aplicação WinSNMP e WinMIB SNMP EasyAgent sub-agentes para o agente SNMP Microsoft que está presente no Windows 95, Windows 98, Windows NT e Windows 2000 (MG-SOFT, 2001).

5.4.3.1 WINSNMP

O WinSNMP é uma proposta que define uma interface de programação para aplicações de gerência de redes sobre ambiente Microsoft Windows. A interface suporta tanto SNMPv1 quanto SNMPv2; SNMPv1 é visto no WinSNMP como um subconjunto de SNMPv2.

WinSNMP é definido para estimular o desenvolvimento de aplicações que gerenciem objetos em uma rede. Como esperado em extensões de API em ambiente Windows, WinSNMP implementa as funções da interface em uma *dynamic link library* (DLL). Por sua vez a DLL usa a interface padrão winsock para realizar suas tarefas.

Maiores informações sobre WINSNMP são encontradas em Granville (1996).

5.4.3.2 MIB COMPILER

O MIB COMPILER converte arquivos MIB no formato proprietário SMIB da MG-SOFT (suportando as especificações SMI, SMIv1 e SMIv2). Uma aplicação pode acessar arquivos de MIB compilados no formato SMIB utilizando a interface WINMIB através das bibliotecas de funções.

Maiores informações sobre MIB COMPILER são encontradas em (MG-SOFT, 2001).

5.4.3.3 SNMP EASYAGENT TOOLKIT

SNMP EASYAGENT TOOLKIT fornece um *framework* para estender o agente SNMP da Microsoft, que funciona no Microsoft Windows 95, Windows 98 e Windows NT. Ao usar este *framework*, qualquer aplicação Win32 pode facilmente importar e exportar seus dados através do agente SNMP, e também pode ser gerenciado remotamente utilizando o protocolo SNMP.

Maiores informações sobre SNMP EASYAGENT TOOLKIT são encontradas em (MG-SOFT, 2001).

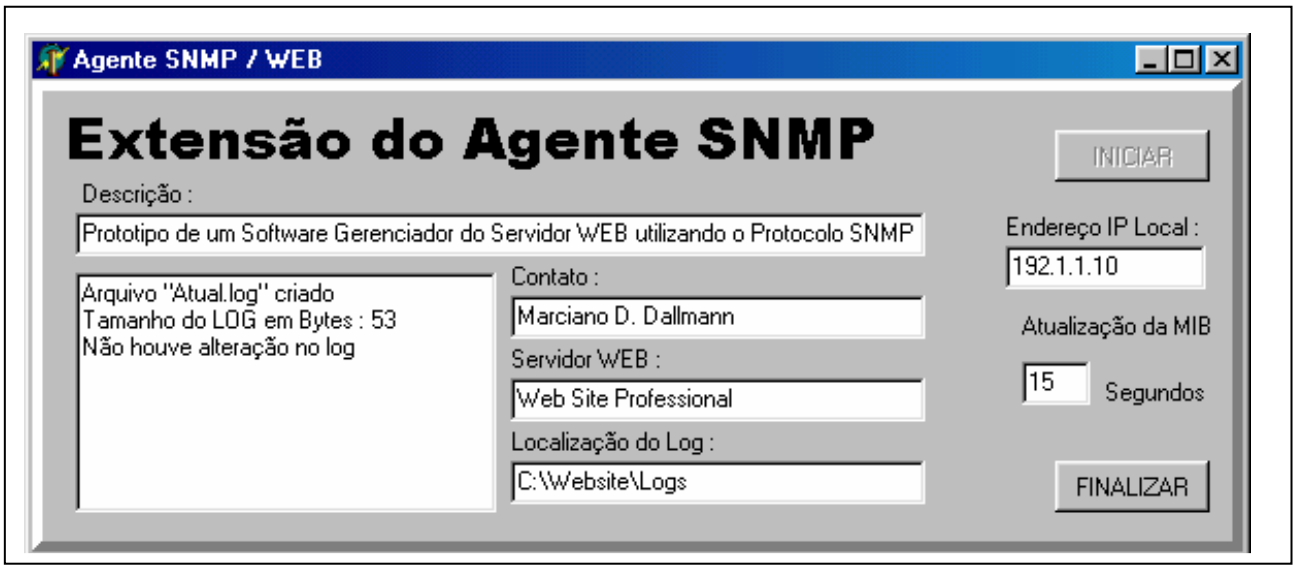
5.4.4 O PROTÓTIPO

A seguir será feita a descrição das principais funções, bem como do funcionamento do protótipo.

5.4.4.1 MÓDULO AGENTE

Ao executar o protótipo da extensão do agente SNMP, entram em operação os processos de inicialização, como identificar o módulo MIB, seu OID, recuperar o endereço IP local. Na tela principal são apresentadas informações sobre descrição do sistema, pessoa de contato, nome do servidor WEB, sendo estas informações também armazenadas na MIB. Deverá ser informado o intervalo de tempo em segundos para atualização da MIB com as informações provenientes do arquivo de *LOG* e também deverá ser informada a localização dos arquivos de *LOG*. Quando for dado comando de iniciar a aplicação no intervalo de tempo estabelecido obtém as informações do arquivo de *LOG*, as organiza, e através da utilização de funções contidas no componente MGEAX da MG-SOFT são armazenadas na MIB. Ver figura abaixo:

Figura 9 - Tela principal da aplicação extensão do agente SNMP



No quadro abaixo é mostrado um trecho de código referente a inicialização, especialmente a parte de identificação da MIB que vai ser gerenciada.

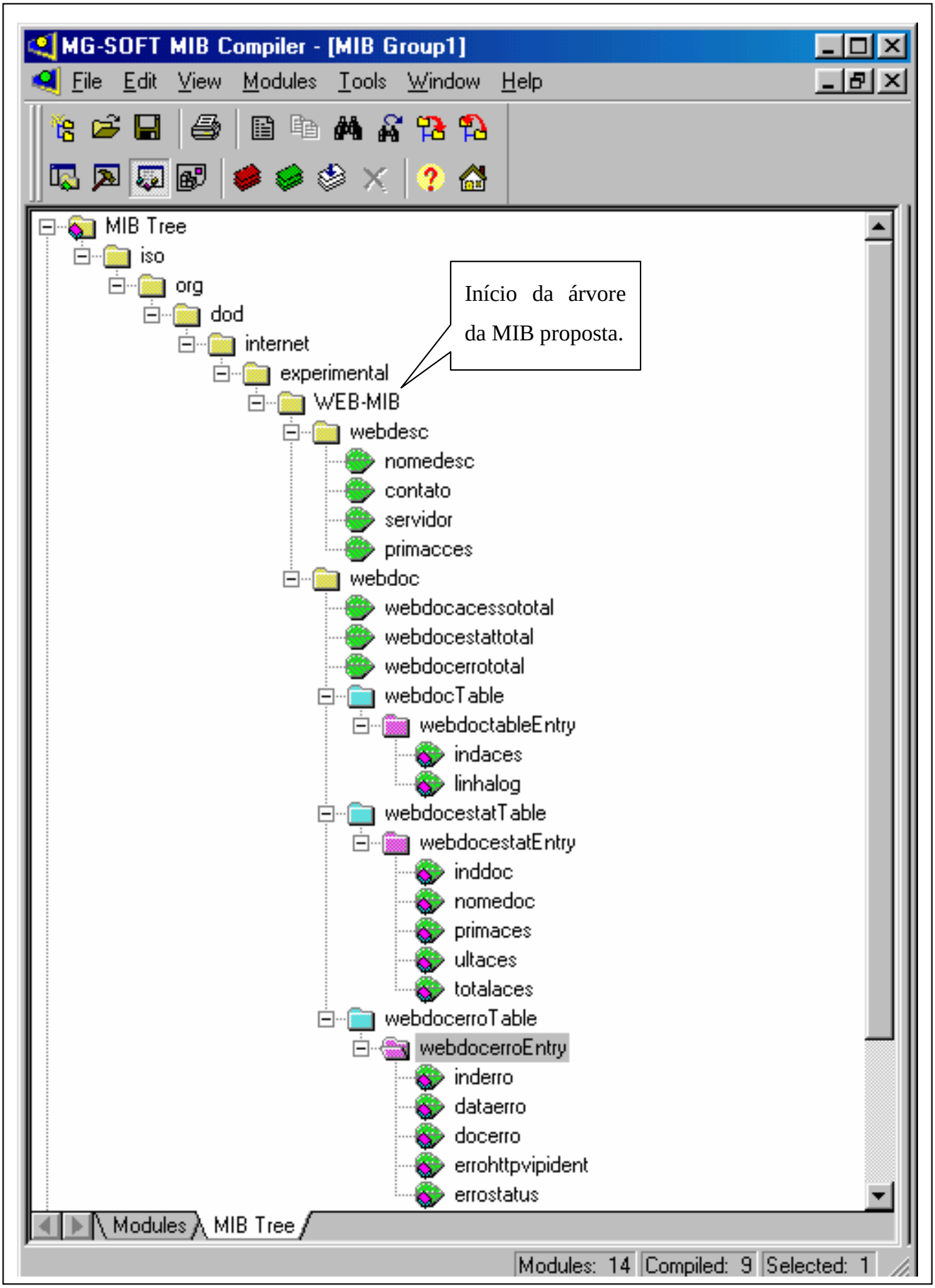
Quadro 3 - Identificação da MIB que vai ser gerenciada.

```
Agente.Module := 'WEB-MIB';
Agente.Subtree := '1.3.6.1.3.100';
brv := Agente.InitAgent('AGENTE WEB-MIB');
```

5.4.4.2 APRESENTAÇÃO DA MIB

A figura abaixo mostra a árvore da MIB proposta para o protótipo, denominada WEB-MIB e se encontra abaixo do nó experimental.

Figura 10 - Mib proposta



A partir da entrada da árvore WEB-MIB começam os objetos gerenciáveis para o protótipo. A entrada da árvore denominada *webdesc* contém os seguintes objetos:

- *nomedesc*, armazena a descrição da implementação;
- *contato*, armazena o nome do administrador, no caso do protótipo o autor;
- *servidor*, armazena o nome do servidor WEB que está sendo gerenciado;
- *primaces*, armazena a data da primeira entrada no arquivo de *log*.

A entrada da árvore denominada *webdoc* contém informações sobre os documentos, e contém os seguintes objetos:

- *webdocacessototal*, armazena o valor referente ao total de documentos acessados pelo servidor WEB;
- *webdocestattotal*, armazena o valor referente ao total de entradas na tabela de estatísticas;
- *webdocerrototal*, contém o valor referente ao total de documentos registrados no *log* do servidor WEB que com status que indica algum erro;
- *webdoctable*, é uma entrada na árvore que contém a tabela *webdoctableEntry* com os seguintes objetos:
  - *indice*, armazena o valor referente ao índice da tabela;
  - *linhalog*, armazena a linha inteira contida no arquivo de *log*;
- *webdocestattable*, é uma entrada na árvore que contém a tabela *webdocestattableEntry* com os seguintes objetos:
  - *inddoc*, armazena o valor referente ao índice da tabela;
  - *nomedoc*, armazena o valor referente ao nome do documento acessado;
  - *primaces*, armazena o valor referente a data, horário do primeiro acesso ao servidor WEB;
  - *ultaces*, armazena o valor referente a data, horário do último acesso ao servidor WEB;
  - *totalaces*, armazena o valor referente ao total de acessos realizados ao documento;
- *webdocerrottable*, é uma entrada na árvore que contém a tabela *webdocerrotableEntry* com os seguintes objetos:
  - *inderro*, armazena o valor referente ao índice da tabela;

- *dataerro*, armazena o valor referente a data, horário de quando aconteceu o erro;
- *docerro*, armazena o valor referente ao nome do documento acessado;
- *errohttpvipserv*, armazena o informações referentes a versão do protocolo HTTP do cliente, endereço IP do cliente, e se for uma tentativa de acesso que requeira uma autenticação registra também o nome do *login* que realizava estas tentativas;
- *errostatus*, armazena o informações referentes ao código do status do erro;

No quadro abaixo é apresentado um trecho do código da MIB referente a tabela *webdocestattableEntry.*

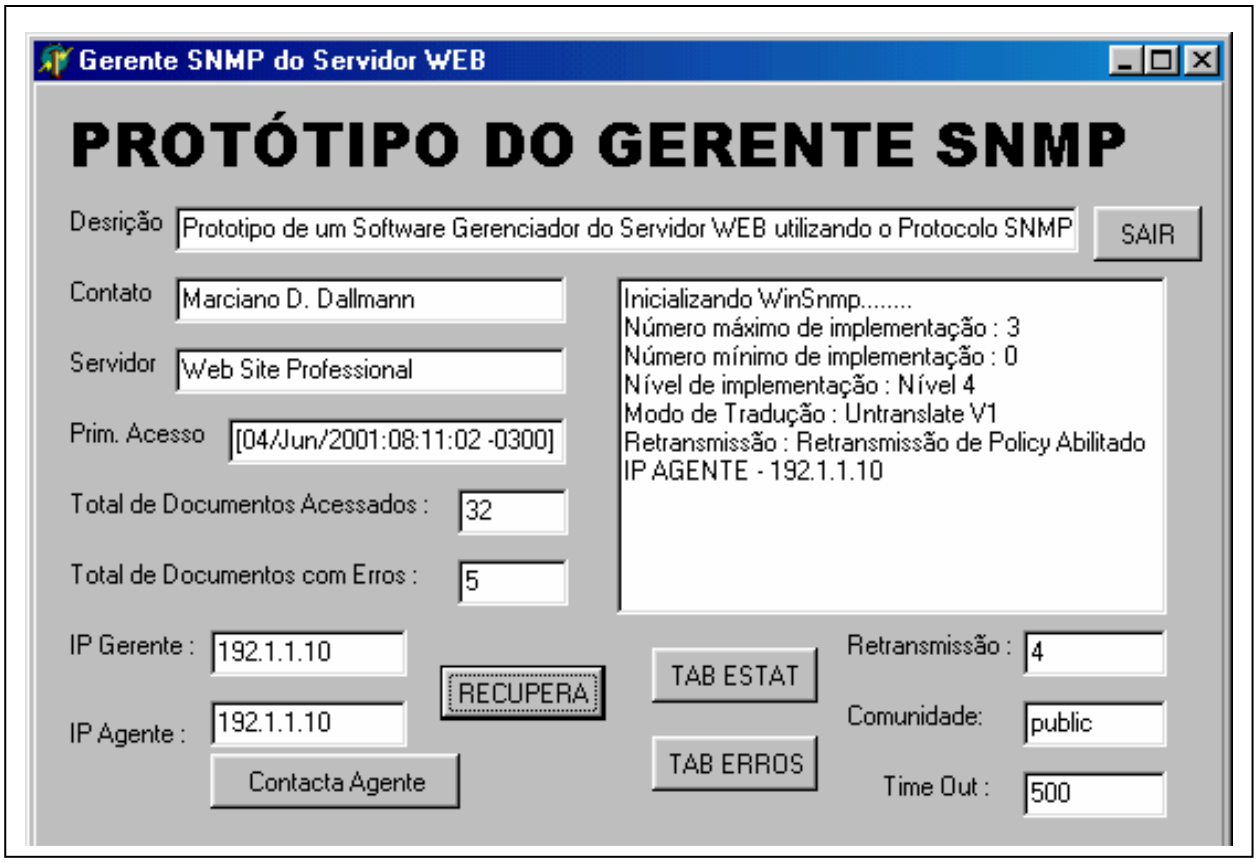
Quadro 4 - Trecho de código da MIB referente a tabela *webdocestattableEntry.*

```
webdoctableEntry OBJECT-TYPE
                SYNTAX webdoctableEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                        "Informações sobre todos os documentos acessados com
                        ou sem sucesso ."
                INDEX { indaces }
        ::= { webdocTable 1 }

        webdoctableEntry ::=
                SEQUENCE {
                    indaces
                      INTEGER (0..65535),
                    linhalog
                      OCTET STRING
                }

        indaces OBJECT-TYPE
                SYNTAX INTEGER (0..65535)
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                        "Informa o Indice."
        ::= { webdoctableEntry 1 }

        linhalog OBJECT-TYPE
                SYNTAX OCTET STRING
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                        "Informa a entrada do LOG"
        ::= { webdoctableEntry 2 }
```

5.4.4.3 MÓDULO GERENTE

Ao iniciar a aplicação Gerente, são executados os processos de inicialização, onde é criado um vínculo com a biblioteca WINSNMP da MG-SOFT, o endereço IP local é recuperado, e os parâmetros de configuração são carregados, o endereço IP do agente SNMP deve ser informado, os parâmetros de configuração também podem ser alterados. Ao contatar o agente SNMP a aplicação estará pronta para recuperar as informações do mesmo, e quando isto ocorrer a tela principal será atualizada com as informações buscadas. Após isto será possível requerer as informações de estatística das páginas acessadas e também os erros ocorridos no servidor WEB. Ver Figura abaixo:

Figura 11 - Tela principal da aplicação gerente SNMP



No quadro abaixo é mostrado um trecho do código referente a função de pesquisa :

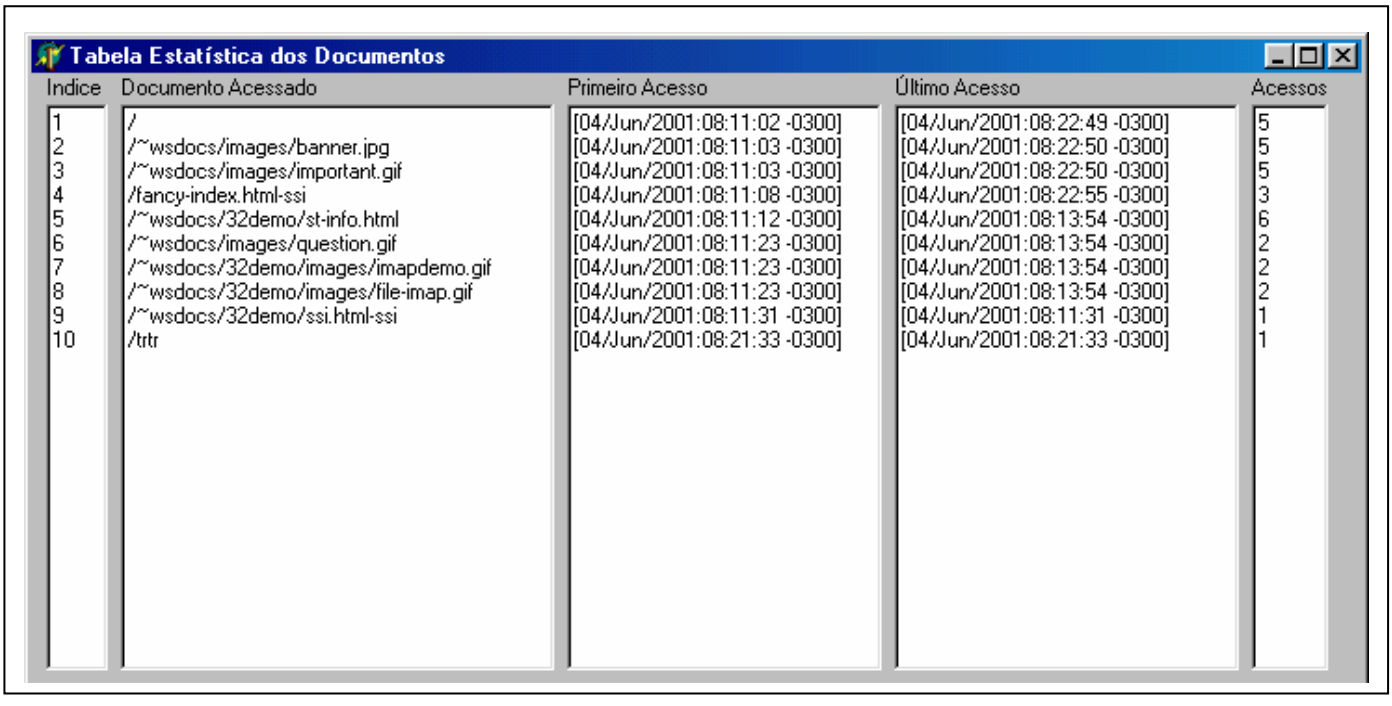
Quadro 5 - Trecho do código referente a função de pesquisa

```
Es.Vbl:=SnmpCreateVbl(Es.SessaoSNMP,@ES.OID,NIL);
   if Es.Vbl = SNMPAPI_FAILURE then
      exit;
   Es.Request:=Es.Request+1;
   Es.Pdu:=SnmpCreatePdu(Es.SessaoSNMP,SNMP_PDU_GETNEXT,
                         Es.Request,0,0,Es.Vbl);
   if Es.Pdu=SNMPAPI_FAILURE then
      exit;
   Stat:=SnmpSendMsg(Es.SessaoSNMP,Es.EntidadeGer,Es.EntidadeAge,
                     Es.Contexto,Es.Pdu);
   if Stat = SNMPAPI_FAILURE then
      exit;
   Stat:=SnmpFreeVbl(Es.Vbl);
   if Stat = SNMPAPI_FAILURE  then
      exit;
   Stat:=SnmpFreePdu(Es.Pdu);
   if Stat = SNMPAPI_FAILURE  then
      exit;
```

5.4.4.3.1 TABELA DE ESTATÍSTICAS

A tabela de estatísticas apresenta as informações organizadas dos acessos a cada documento contido no arquivo de *log* do servidor WEB. As informações são as seguintes, índice do documento, seu nome e seu caminho, o primeiro e o último acesso, e a quantidade total de acessos à cada documento. Ver figura Abaixo:

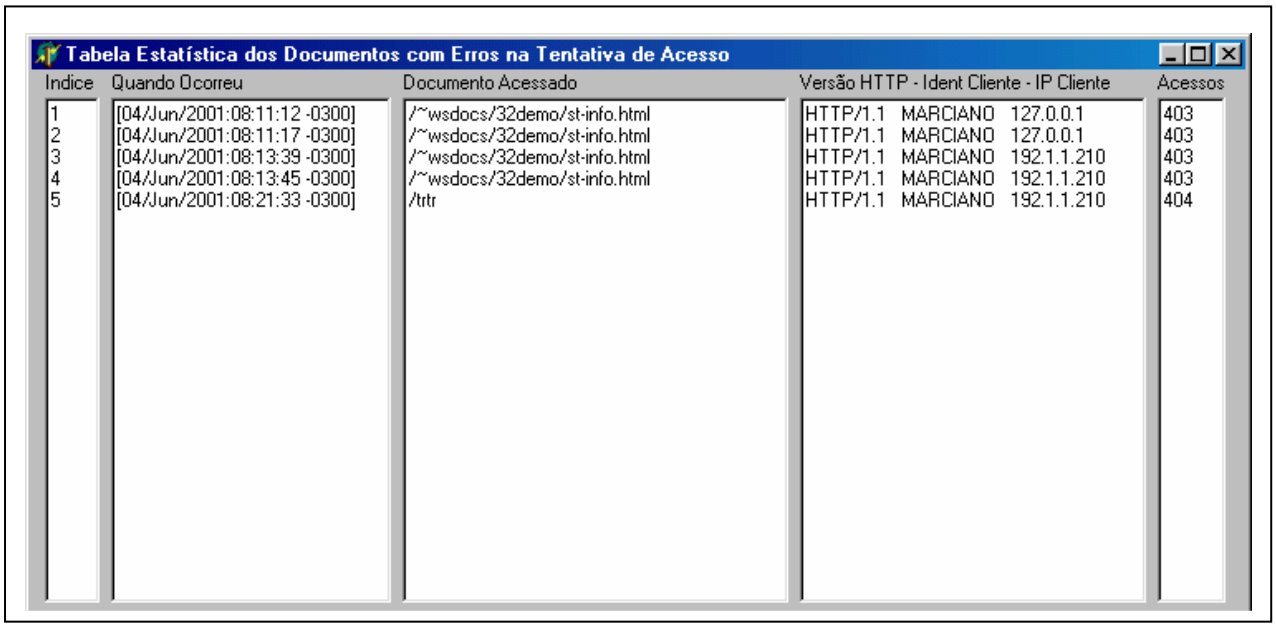
Figura 12 - Tabela de estatística dos documentos



5.4.4.3.2 TABELA DE ERROS

A tabela de Erros contém informações sobre os erros ocorridos no acesso a um determinado documento, como também erros de servidor contidos no arquivo *log* do servidor WEB. Estas informações estão organizadas da seguinte maneira, índice do documento, quando ocorreu o erro, nome do documento e seu caminho, versão HTTP, endereço IP e identificação do cliente, e por último código de *Status* do erro que serve para identificar o erro ocorrido. Ver figura Abaixo:

Figura 13 - Tabela de erros



6 CONCLUSÕES

A conclusão deste trabalho foi conseqüência do cumprimento de todas as etapas, uma de cada vez, conforme o cronograma apresentado e aprovado na proposta de TCC.

O uso das ferramentas da MG-SOFT, foi de suma importância para a especificação e implementação do protótipo, sendo que seria inviável, devido ao tempo disponível para execução deste trabalho, implementar tais funções de comunicação. Fez-se necessário uma busca a materiais que explicassem de forma mais detalhada o uso das funções disponibilizadas pela ferramenta.

A especificação e implementação do protótipo foram as etapas que exigiram um conhecimento mais aprofundado das características do protocolo SNMP, bem como dos conceitos envolvidos na gerência de redes.

Atualmente existem algumas aplicações de gerenciamento de redes um pouco mais sofisticadas. A maioria destas aplicações possibilitam apenas o monitoramento dos nós de uma rede e não possui inteligência para auxiliar os administradores de redes na execução de suas tarefas. Também, as aplicações são genéricas ou específicas demais.

Durante o desenvolvimento do trabalho houve um problema na parte da extensão do agente SNMP que ainda não foi resolvido. Quando a extensão do agente é finalizada, e há necessidade de reiniciá-la, o agente SNMP da Microsoft também precisa ser reiniciado, se isto não acontecer a aplicação não funcionará.

É importante ressaltar que para a implementação e testes com o protótipo foi necessário um estudo sobre os arquivos de *log*, no que diz respeito a configuração de um servidor WEB para tratar esses arquivos.

Este trabalho também abre a possibilidade de gerenciamento dos arquivos *log* de outras aplicações, utilizando o protocolo SNMP.

O protótipo foi desenvolvido para gerenciar um servidor WEB através do seu arquivo de *log*, onde uma extensão de agente obtém estas informações no *log* e organiza as informações numa MIB criada, e gerenciando isto através do gerente localizado em um outro *host* da rede. Portanto, considera-se que o protótipo atingiu os objetivos propostos.

6.1 EXTENSÕES

Como sugestão para continuação do trabalho, propõe-se a implementação de *traps*, como também a apresentação de outras tabelas de informações dos arquivos *log* no gerente SNMP, e também a implementação de um agente para gerenciamento de um servidor WEB no ambiente UNIX.

REFERÊNCIAS BIBLIOGRÁFICAS

ABLAN, Jerry; YANOFF, Scott. **Web Site administrator's survival guide**. Indianapolis, EstadosUnidos da América: Sams.net Publishing, 1996.

ARAGÃO, Marcelo Jorge. **Gerenciamento WWW utilizando protocolo SNMP**, Fortaleza, set. 1997. Disponível em: <http://www.secrel.com.br/usuarios/mja/wwwsnmp.html>. Acesso em: 19 out. 2000.

BRISA, Sociedade Brasileira para Interconexão de Sistemas Abertos. **Gerenciamento de redes – uma abordagem de sistemas abertos**. São Paulo: Makron Books, 1993.

BRISA, Sociedade Brasileira para Interconexão de Sistemas Abertos. **Arquitetura de redes de computadores OSI e TCP/IP**. São Paulo: Makron Books, 1994.

CHESANI, Luciana. **Simple network management protocol**. Porto Alegre, 1995. Disponível em <http://penta.ufrgs.br/gr952/trab1/snmp_snmp.html>. Acesso em 10 mar. 2001.

CANTÚ, Marco. **Dominando o Delphi 3 , a bíblia**. São Paulo : Makron Books, 1998.

CARVALHO, Faical Farhat de. Programação orientada à objetos usando Delphi 3. São Paulo : Erica, 1998.

COMER, Douglas E. **Redes de computadores e Internet**. 2. ed. Tradução Marinho Barcellos. Porto Alegre: Bookman, 2001. 522p.

COMER, Douglas E.; STEVENS, David L. **Interligação em rede com TCP/IP**. Vol.2. Rio de Janeiro: Campus, 1999.

FURLAN, José D. **Modelagem de objetos através da UML**. São Paulo: Makron Books,1998. 329p.

GRANVILLE, Lisandro Zambenedetti. **Tutorial WinSnmp**.Porto Alegre, 1996. Disponível em <http://penta.ufrgs.br/gere96/winsnmp/winsnmp.html>. Acesso em 10 mar. 2001.

MAFINSKI, André. **Protótipo de software de gerência SNMP para o ambiente Windows NT**. 1999. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

MENEZES, Elionildo da Silva; SILVA, Pedro Luciano Leite. **Gerenciamento de Redes: Estudos de Protocolos**. Pernambuco, set. 1998. Disponível em <http://www.di.ufpe.Br/~flash/ais98/gerrede/gerrede.html>. Acesso em 16 abr. 2001.

MG-SOFT. **Corporation Product Line**. Slovenia, fev. 2001. Disponível em: <http://www.mg-soft.com/products.html>. Acesso em 20 mai. de 2001.

MICROSOFT. **Microsoft Windows NT Server 4.0 Networking Guide**. São Paulo: Makron Books, 1997.

MORO, Mirella Moura. **Tutorial básico sobre Rational Rose**. Porto Alegre, set. 2000. Disponível em < http://www.inf.ufrgs.br/~mirella/cmp102/index.html>. Acesso em 20 abr. de 2001 .

OTSUKA, Joice. **Management Information Base.** Porto Alegre, 1995. Disponível em: <http://penta.ufrgs.br/gr952/trab1/2capa.html>. Acesso em 10 mar. 2001.

REKOWSKY, Ricardo H. **Protótipo de um software para monitoração de desenpenho de redes, utilizando RMON**. 1999. 98 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SANTOS, Kellyne Marques. **Ferramentas Case**. Porto Alegre, set. 2000. Disponível em <http://www.inf.pucrs.br/~kellyne/uml/case/tsld001.htm>. Acesso em 20 abril 2001.

SOARES, Luiz F. Gomes; LEMOS, Guido; COLCHER, Sérgio. **Redes de computadores, das LANs MANs e WANs às redes ATM**. 2 ed. Rio de Janeiro: Campus, 1995.

SZTAJNBERG, Alexandre. **Conceitos básicos sobre os protocolos SNMP e CMIP**. Abr. 1996. Disponível em: <www.gta.ufrj.bl/ãlexszt/ger/snmpcmip.html>. Acesso em: 06 mar. 2001.

TRUNFIO, Paul A. **Learning about your users, Part 1.** Estados Unidos, ago. 1998. Disponível em < http://webserver.cpg.com/wt/3.8/index.html>. Acesso em 08 jun. 2001.

ZACKER, Craig; DOYLE, Paul. **Redes de computadores – configuração, manutenção e expansão**. São Paulo: Makron Books, 2000.