

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO**

**SOFTWARE PARA MONITORAMENTO DE SERVIDORES**  
**WEB APACHE**

**FABRICIO KLABUNDE**

**BLUMENAU**  
**2007**

**2007/2-14**

**FABRICIO KLABUNDE**

## **SOFTWARE PARA MONITORAMENTO DE SERVIDORES**

### **WEB APACHE**

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciências da Computação — Bacharelado.

Prof. Francisco Adell Péricas, Mestre - Orientador

**BLUMENAU**  
**2007**

**2007/2-14**

# **SOFTWARE PARA MONITORAMENTO DE SERVIDORES**

## **WEB APACHE**

Por

**FABRICIO KLABUNDE**

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Francisco Adell Péricas, Mestre – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Mauro Marcelo Mattos, Doutor – FURB

Membro: \_\_\_\_\_  
Prof. Sérgio Stringari, Mestre – FURB

Blumenau, 27 de novembro de 2007

Dedico este trabalho a todos que contribuíram de uma forma ou de outra para sua realização. Principalmente aos colegas e professores que tiveram participação direta, auxiliando para que o objetivo fosse atingido.

## **AGRADECIMENTOS**

À Deus, pelo seu imenso amor e graça, por sempre guiar meu caminho.

À minha família, que sempre acreditou na minha capacidade e me deu forças para que eu fosse até o fim.

Aos meus amigos, que em muitos momentos foram importantes na construção do conhecimento.

Ao meu orientador, Francisco Adell Péricas, que sempre acreditou que seria possível o desenvolvimento deste trabalho, e por ter fornecido os subsídios necessários para a construção do mesmo.

Algumas das maiores façanhas do mundo foram feitas por pessoas que não eram suficientemente espertas para saber que elas eram impossíveis.

Doug Larson

## RESUMO

O presente trabalho vem mostrar de forma prática o uso de tecnologias atuais para uma solução alternativa no monitoramento de servidores web Apache. Trata-se de um protótipo que funciona através da internet, onde o usuário é capaz de conectar-se através de qualquer ponto da grande rede (internet) e fazer o monitoramento de um servidor Apache. Para o usuário poder fazer esse monitoramento, o protótipo é desenvolvido através de páginas dinâmicas escritas com *Hipertext Preprocessor* (PHP), e as informações gerenciáveis são coletadas através do protocolo *Simple Network Management Protocol* (SNMP). Uma das principais particularidades do trabalho está na flexibilização quanto à localização do usuário para operar o protótipo, uma vez que permite acesso de qualquer ponto na internet. Este trabalho propõe além de aspectos acadêmicos uma possível utilização futura no monitoramento de servidores web Apache.

Palavras-chave: Servidor web. SNMP. Apache. Gerência de servidor web.

## **ABSTRACT**

The present job approaches in a practical way the use of current technologies for an alternate solution to Apache web server management. It talks about a prototype for an application which works through the internet, where the user is able to get connected from any site on the internet and manage an Apache web server. In order to manage the system, it was developed within dynamic written pages with “Hipertext Preprocessor” (PHP), where the handling information are collected through the “Simple Network Management Protocol” (SNMP). One of the main features of this work is based on the flexibility in relation to the user’s location to operate the system, once it allows access from any site on the internet. This job intends to provide, including the academical aspects, an useful tool for the Apache web servers management.

Key-words: Web server. SNMP. Apache. Web server management.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura de gerenciamento de redes.....	15
Figura 2 – Modelo de Gerenciamento de Redes OSI.....	16
Figura 3 – Modelo Organizacional.....	17
Figura 4 – Configuração da rede com MIB e MDB.....	18
Figura 5 – MIT da OSI.....	18
Figura 6 – Visão de um objeto na perspectiva da internet.....	19
Figura 7 – Modelo de comunicação.....	20
Figura 8 – Protocolos de Transferência – Comunicação.....	20
Figura 9 - Modelo Funcional de Gerenciamento de Redes.....	21
Quadro 1 – Padrão de Formato de Mensagens SNMP.....	22
Figura 10 – Modelo SNMP.....	23
Quadro 2 – Os tipos de dados primitivos da ASN.1 permitidos no SNMP.....	25
Quadro 3 – Os grupos de objetos na MIB-II da internet.....	27
Figura 11 – Diagrama de Caso de Uso.....	33
Quadro 4 – Descrição do Caso de Uso Logar-se no sistema.....	33
Quadro 5 – Descrição do Caso de Uso Consultar Log do Sistema.....	34
Quadro 6 – Descrição do Caso de Uso Verificar <i>Status</i> do Servidor.....	34
Quadro 7 – Descrição do Caso de Uso Consultar Informações de Configuração.....	35
Quadro 8 – Descrição do Caso de Uso Consultar Informações de Contabilização.....	36
Quadro 9 – Descrição do Caso de Uso Consultar Estatísticas do Servidor.....	37
Figura 12 – Modelo Conceitual da Base de Dados do protótipo.....	38
Figura 13 – Diagrama de classes.....	39
Figura 14 – Diagrama de atividades da interação do usuário com o protótipo.....	40
Figura 15 – Arquitetura de funcionamento e gerenciamento do servidor Apache.....	41
Figura 16 – Tela de <i>login</i> .....	42
Figura 17 – Tela de Boas Vindas.....	42
Figura 18 – Tela de escolha de qual servidor deseja consultar <i>status</i> .....	43
Figura 19 – <i>Status</i> do servidor.....	44
Quadro 10 – Trecho do código fonte responsável pela consulta de <i>status</i> .....	45
Figura 20 – Configurações do servidor.....	45
Quadro 11 – Trecho do código fonte responsável pela consulta de configuração.....	46

Figura 21 – Informações de contabilização .....	46
Quadro 12 – Trecho do código fonte responsável pela consulta de contabilização .....	47
Figura 22 – Tráfego total .....	47
Figura 23 – Log .....	48
Quadro 13 – Testes com a ferramenta .....	50
Quadro 14 – MIB para servidor Apache .....	61

## LISTA DE SIGLAS

CMIP – *Common Management Information Protocol*

HTTP – *Hiper-Text Transfer Protocol*

IEEE – *Institute of Electrical and Eletronic Engineers*

ISO – *International Organization for Standarding*

LAN – *Local Area Network*

MAN – *Metropolitan Area Network*

MDB – *Management Data Base*

MIB – *Management Information Base*

MIT – *Management Information Tree*

NMS – *Network Management Station*

OSI – *Open Systems Interconnection*

PDU – *Protocol Data Unit*

SGBD – *Sistema Gestor de Base de Dados*

SMI – *Structure of Management Information*

SNMP – *Simple Network Management Protocol*

TMN – *Telecommunications Management Network*

UDP – *User Datagram Protocol*

UML - *Unified Modeling Language*

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>12</b>
1.1 OBJETIVOS DO TRABALHO .....	13
1.2 ESTRUTURA DO TRABALHO .....	13
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>14</b>
2.1 GERÊNCIA DE REDES .....	14
2.1.1 Modelo de Gerenciamento de Redes.....	16
2.1.2 Modelo Organizacional .....	16
2.1.3 Modelo de Informação .....	17
2.1.4 Modelo de Comunicação.....	20
2.1.5 Modelo Funcional .....	20
2.2 PROTOCOLO SNMP .....	21
2.3 O MODELO SNMP .....	23
2.4 SNMPV2 .....	26
2.5 MANAGEMENT INFORMATION BASE (MIB).....	27
2.6 SERVIDOR WEB APACHE .....	28
<b>3 DESENVOLVIMENTO DO PROTÓTIPO.....</b>	<b>31</b>
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	31
3.2 ESPECIFICAÇÃO .....	32
3.2.1 Diagrama de Casos de Uso .....	32
3.2.2 Modelo Conceitual da Base de Dados .....	37
3.2.3 Diagrama de Classes .....	38
3.2.4 Diagrama de Atividades.....	40
3.3 IMPLEMENTAÇÃO .....	41
3.3.1 Técnicas e ferramentas utilizadas.....	41
3.3.2 Operacionalidade da implementação .....	41
3.4 RESULTADOS E DISCUSSÃO .....	48
3.4.1 Análise de Trabalhos Correlatos .....	50
<b>4 CONCLUSÕES.....</b>	<b>52</b>
4.1 EXTENSÕES .....	52
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>54</b>
<b>ANEXO A – Apache2-MIB.....</b>	<b>56</b>

## 1 INTRODUÇÃO

Com o constante aumento do número das redes de computadores, fica indispensável a sua administração. Isso ocorre porque cada vez mais o computador torna-se uma ferramenta essencial em todos os setores das empresas, sejam elas grandes corporações ou até mesmo empresas de médio e pequeno porte.

As redes estão ficando cada vez maiores, atingindo mais pessoas, transformando-se em algo mais heterogêneo, pois possuem uma mesclagem de tecnologias e de fornecedores. As tecnologias estão em constante desenvolvimento, exigindo assim mais recursos dos equipamentos, um melhor desempenho, e por conseguinte uma capacidade maior de administração dos mesmos.

Um fator que contribui acentuadamente para que o computador passe a ser uma ferramenta cada vez mais utilizada por um maior número de pessoas é o uso da rede mundial de computadores, a internet, que passa por um processo acelerado de disseminação. Em função deste crescimento, surgem necessidades quanto ao controle de diversas tarefas que devem ser executadas.

Dentro deste contexto, pode-se citar a administração de softwares que controlam milhares de páginas internet, as quais ficam armazenadas em servidores espalhados por toda a rede: são os chamados servidores *web*, que servem para prover o acesso a estas páginas. Estes, por sua vez, precisam ser monitorados para que se possa ter um controle sobre os serviços prestados, e para evitar problemas quanto ao funcionamento dos mesmos.

Este trabalho descreve o desenvolvimento de um protótipo para monitoramento de um servidor *web* específico, o servidor Apache, tendo em vista que não são conhecidas ferramentas desenvolvidas para este fim. Um dos aspectos que pode ser observado em relação aos mecanismos já implementados é que muitos ou quase todos os softwares desenvolvidos para atender as necessidades na administração de servidores *web* são para uso a partir de uma máquina específica da rede, mais precisamente o servidor. É baseado nisto que este documento especifica um monitoramento que possa ser feito através da internet, dando assim maior flexibilidade ao administrador.

O servidor Apache é o mais utilizado no mundo atualmente Alecrim (2006), sendo que este servidor não possui facilidades implementadas para o seu gerenciamento. O Apache sequer possui em sua compilação original suporte ao *Simple Network Management Protocol* (SNMP), que é um protocolo de gerenciamento. No entanto é possível a inclusão de uma

*Management Information Base* (MIB) para dar suporte ao SNMP permitindo assim seu gerenciamento. A MIB de gerenciamento para o Apache já existe implementada, porém não faz parte da distribuição do Apache. É utilizada para auxiliar na construção do protótipo proposto neste trabalho a *Hipertext Preprocessor* (PHP), linguagem de programação para internet. Vale salientar que trata-se de uma solução de monitoramento exclusivamente do servidor *web* Apache, independente do hardware utilizado.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um protótipo para monitoramento SNMP de um servidor *web* Apache. O protótipo permite ao usuário fazer consultas de informações de configuração e contabilização do servidor.

Os objetivos específicos do trabalho são:

- a) permitir o gerenciamento do servidor *web* Apache via navegador internet;
- b) disponibilizar informações de configuração do servidor *web*;
- c) disponibilizar informações de contabilização, como bytes enviados e recebidos, números de acessos efetuados e rejeitados;
- d) integrar suporte SNMP ao servidor Apache.

## 1.2 ESTRUTURA DO TRABALHO

Este trabalho está estruturado em quatro capítulos. O primeiro capítulo apresenta uma breve introdução sobre o trabalho e os principais objetivos pretendidos.

O segundo capítulo apresenta a fundamentação teórica que deu sustentação para o desenvolvimento deste trabalho, através de uma abordagem sobre as tecnologias usadas na construção, tais como o protocolo SNMP, a linguagem para páginas dinâmicas PHP, a estrutura de dados para coleta de informações de gerência (MIB), e o servidor *web* Apache.

No terceiro capítulo é apresentada a especificação do protótipo, seu desenvolvimento através de diagramas, implementação, ferramentas utilizadas e teste das funcionalidades atingidas com o mesmo.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo destina-se ao estudo das tecnologias envolvidas no desenvolvimento do trabalho, trazendo de uma forma mais detalhada cada uma das áreas trabalhadas.

### 2.1 GERÊNCIA DE REDES

Atualmente as empresas estão aumentando cada vez mais o número de seus computadores, os quais têm a finalidade de auxiliá-las nas mais diversas tarefas a serem realizadas. O aumento na quantidade de computadores dentro das empresas se dá devido ao fato de que a cada dia as empresas precisam ter maior controle e maior agilidade no desenrolar das atividades, e para que isso ocorra de uma forma satisfatória, é preciso que todos os setores dentro da empresa estejam interligados para que possam compartilhar informações e recursos. É neste contexto que surge o conceito de redes, exatamente para fazer essa interligação entre diversos pontos que até então se encontravam isolados e que com este novo conceito passam a compartilhar informações e recursos independentemente da localização física dos membros que a compõem. “A rede também aumenta a **confiabilidade** do sistema, pois tem fontes alternativas de fornecimento” (TANENBAUM, 1997, p. 3, grifo do autor). Portanto, redes de computadores são mecanismos fundamentais para qualquer organização, pois facilitam a comunicação entre diferentes pontos da organização e possibilitam compartilhamento de recursos.

De acordo com Santos (2002), para que seja viável a administração de redes de computadores ou de sistemas de telecomunicações é preciso o auxílio de plataformas de gerenciamento. Tais plataformas permitem o controle de forma remota do funcionamento de equipamentos de informática que estão ligados a uma rede de comunicação de dados.

Sistemas de gerenciamento possuem processos gerentes, processos agentes e elementos de rede. Gerentes são processos executados nos equipamentos que gerenciam toda a rede, já os processos agentes são módulos de softwares que compilam informações no equipamento gerenciado, que são armazenadas em um banco de dados de gerenciamento.

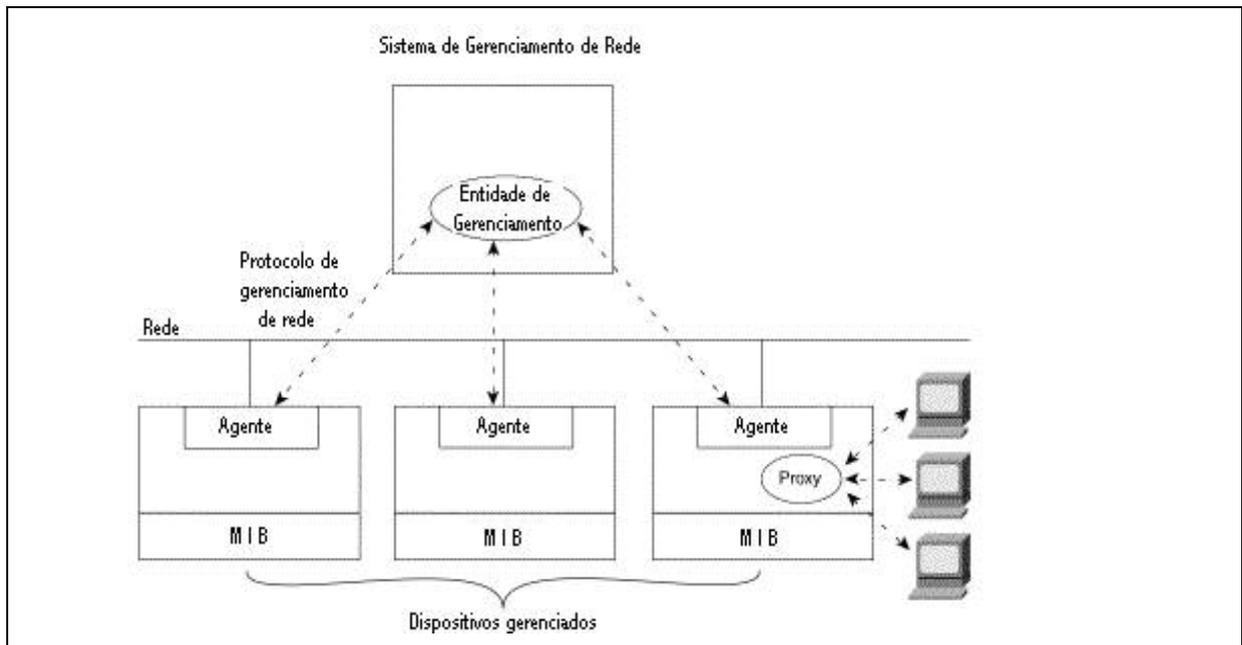
Há atualmente quatro padrões especificados para o gerenciamento de redes: SNMP, *Common Management Information Protocol - OSI, Telecommunications Management Network*

- ITU e *Web-based Management* (baseado em tecnologia *web*).

A grande maioria das arquiteturas de gerenciamento de redes utilizam uma mesma estrutura básica. Dispositivos que permitem ser gerenciados tais como computadores e dispositivos de rede, executam um software que os habilita a serem capazes de enviar algum tipo de alerta quando algum problema é detectado. Quando esses alertas são disparados ocorrem uma ou várias ações que são executadas pelas entidades de gerenciamento, as quais têm o papel de reagir notificando os operadores do sistema por exemplo, ou mesmo fazendo a inclusão deste evento no histórico de eventos, desligamento de algum dispositivo, tentativa de reparo automático, ou alguma outra coisa qualquer que possa ou que deva ser feita na ocorrência destes eventos.

Também é possível que as entidades de gerenciamento requisitem valores de determinadas variáveis para estações na rede. Tais requisições podem ocorrer de forma automatizada ou através de comandos executados pelo usuário do sistema, porém o agente do dispositivo gerenciado responde a todas as requisições feitas.

Pode-se observar a arquitetura de gerenciamento de redes através da figura 1.

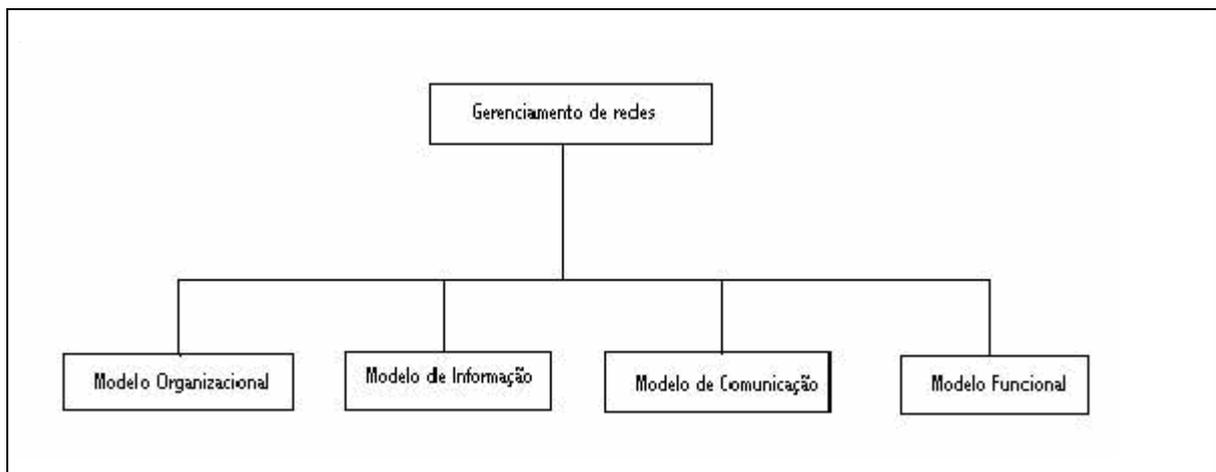


Fonte: Santos (2002).

Figura 1 – Arquitetura de gerenciamento de redes

### 2.1.1 Modelo de Gerenciamento de Redes

O modelo de redes OSI é um padrão ISO e tem colaborado amplamente para a padronização das redes. Trata-se de um modelo fundamental para compreensão de sistemas de gerenciamento de redes. Na figura 2 é possível observar o modelo de gerenciamento OSI que agrega 4 modelos: modelo organizacional, modelo de informação, modelo de comunicação e modelo funcional.

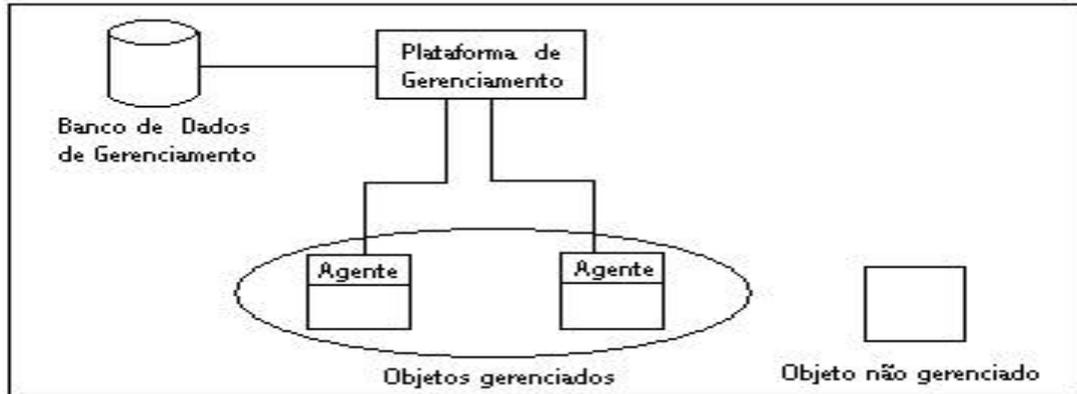


Fonte: Santos (2002).

Figura 2 – Modelo de Gerenciamento de Redes OSI

### 2.1.2 Modelo Organizacional

De acordo com Santos (2002), o modelo organizacional descreve os componentes de redes e suas funções, relações e infra-estruturas. Define os termos objeto, agente e gerente. Isto pode ser visto na figura 3, onde é apresentado um modelo de 2 camadas. Um objeto de rede pode ser um *switch*, *bridge*, etc. Tais objetos podem ser gerenciáveis ou não gerenciáveis, sendo que os elementos gerenciáveis executam internamente um processo de gerenciamento, o que é chamado de agente.



Fonte: Santos (2002).

Figura 3 – Modelo Organizacional

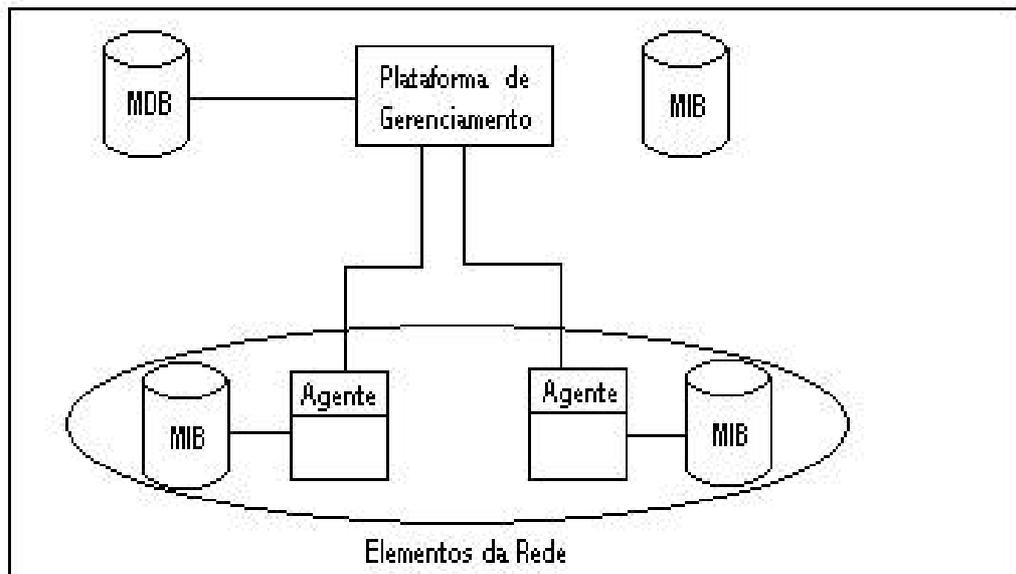
Na figura 3, uma plataforma de gerenciamento solicita informações ao agente, e estas informações são armazenadas no banco de dados de gerenciamento.

### 2.1.3 Modelo de Informação

Segundo Santos (2002), o modelo de informação trabalha com o armazenamento da informação, onde define a estrutura e organização do gerenciamento. Ele especifica a estrutura de gerenciamento da informação (SMI) e a base de dados (MIB).

SMI faz a descrição de como o gerenciamento é estruturado, tratando da sintaxe e da semântica das informações que posteriormente serão armazenadas na MIB. A MIB por sua vez trata das relações e do armazenamento das informações. Ela usa os processos agente e gerente no armazenamento e troca de informações de gerenciamento.

Na figura 4 é possível observar o modelo de informação com as MIBs, o gerente possui 2 bancos de dados MDB.

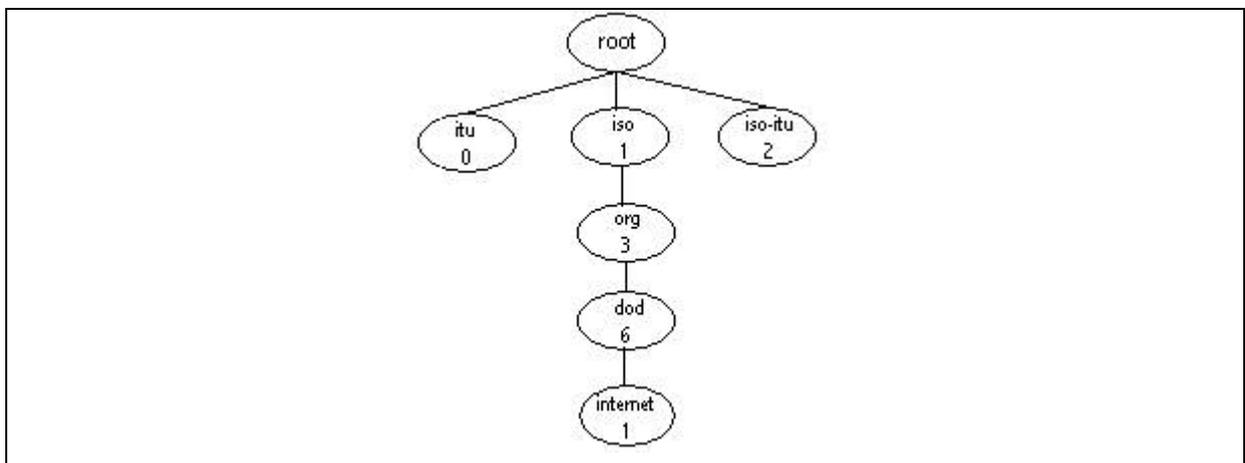


Fonte: Santos (2002).

Figura 4 – Configuração da rede com MIB e MDB

A MIB será questionada pela plataforma de gerenciamento sobre quais são as variáveis e valores associados através do agente residente no componente. Tal questionamento é feito de tempos em tempos buscando detectar possíveis falhas.

Os objetos gerenciáveis são definidos por uma árvore (MIT) especificada pelo modelo OSI, que é adotada internacionalmente. A figura 5 ilustra a MIT.



Fonte: Santos (2002).

Figura 5 – MIT da OSI

De acordo com Santos (2002), a MIT possui três camadas logo abaixo da raiz, que são: iso, itu e iso-itu. A camada “iso” define a *International Standards Organization* e a camada “itu” define a *International Telecommunication Union*. O número de cada círculo identifica a designação do objeto na camada. A “iso” é designada como ‘1’, a camada “org” como ‘1.3’, “dod” (*Department of Defense*) como ‘1.3.6’ e “internet” fica sendo ‘1.3.6.1’. Os objetos gerenciáveis estarão abaixo de “internet” e terão o número deles seguidos de mais pontos e

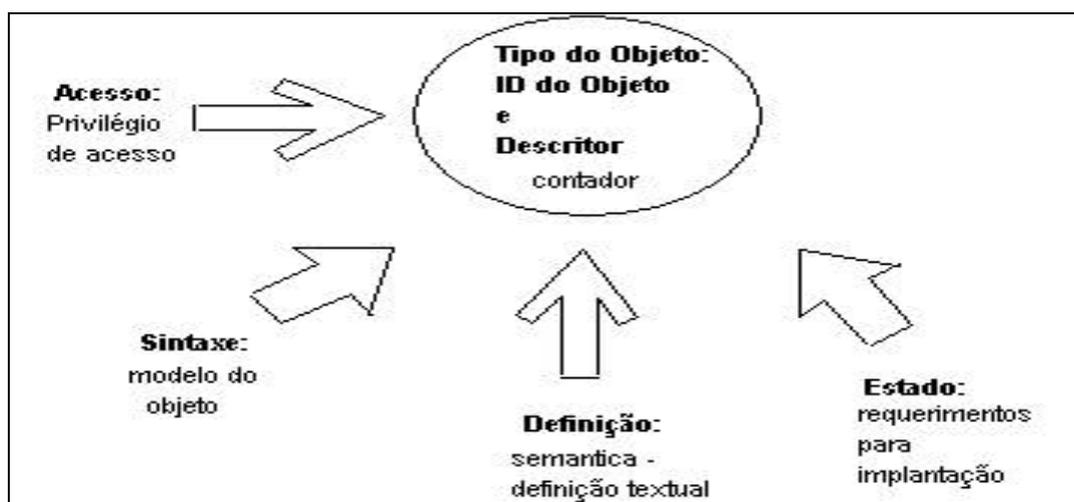
números.

De acordo com Santos (2002), quando os objetos gerenciáveis foram definidos anteriormente, foi criado um vínculo com elementos da rede, como por exemplo: *hub*, roteador, etc. A definição de tais objetos será ampliada de uma maneira que não seja necessariamente algo físico, podendo ser não palpável, como um “contador” que marca o número de pacotes recebidos. É conveniente o uso de uma representação de forma física para o entendimento das características e operações associadas. Dessa forma, o objeto terá uma sintaxe, como por exemplo “contador”, e uma definição que dará a sintaxe, como o nome sugere, uma definição do objeto, como por exemplo “número de pacotes recebidos”.

O objeto está agora associado a uma forma física, possui uma sintaxe e uma descrição, poderá ser acessado e apenas visualizado ou ter seus parâmetros modificados. Deve-se definir atributos de acesso ao objeto, definindo assim privilégios possíveis como, leitura ou leitura-escrita.

Quando se fala em grupo de objetos, deve-se definir se um determinado objeto é essencial para o grupo, tendo então o estado. Tal estado pode ser obrigatório ou opcional. É possível ainda existir vários tipos de contadores (objetos), como por exemplo contadores de tempo, pacotes enviados, pacotes recebidos etc. Por isso é preciso definir um identificador de objetos (identificador) e um descritor (nome).

Definem-se 5 atributos básicos de um objeto gerenciável na perspectiva da internet, que são: tipo do objeto (identificador do objeto e descritor), definição, sintaxe, acesso, e estado. A figura 6 mostra um objeto ‘contador’ na perspectiva da internet.



Fonte: Santos (2002).

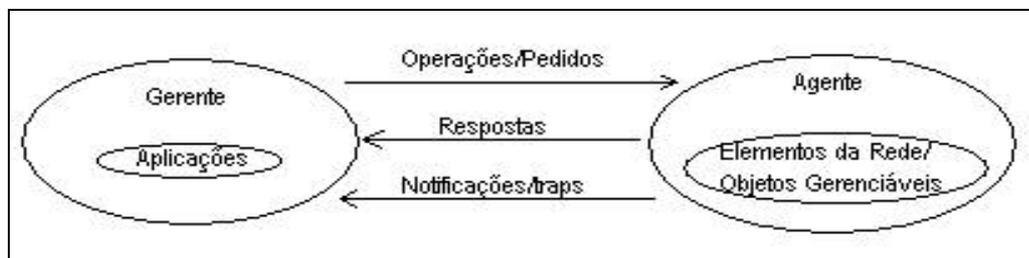
Figura 6 – Visão de um objeto na perspectiva da internet

### 2.1.4 Modelo de Comunicação

Segundo Santos (2002), o modelo de comunicação define a forma como as informações são trocadas entre os sistemas. As informações de gerenciamento são trocadas entre agentes e gerentes, ou mesmo processos gerentes entre si.

Três elementos devem ser observados no modelo de comunicação: protocolo de mensagem, protocolo de aplicação, comandos e respostas. Pedidos são executados pelo módulo gerente, e o agente executa o pedido no elemento gerenciável da rede e retorna a resposta ao gerente. Notificações são mensagens não solicitadas originadas no agente.

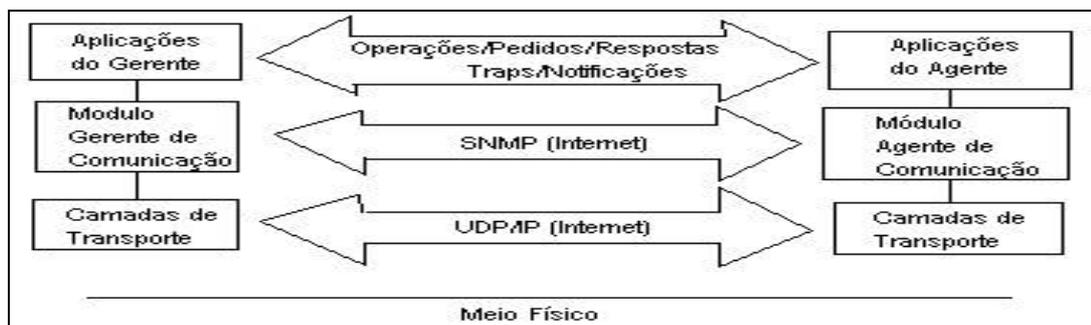
A figura 7 ilustra o modelo de comunicação.



Fonte: Santos (2002).

Figura 7 – Modelo de comunicação

Na figura 8 é possível observar a estrutura de comunicação entre os objetos gerenciáveis, através dos protocolos de gerenciamento.

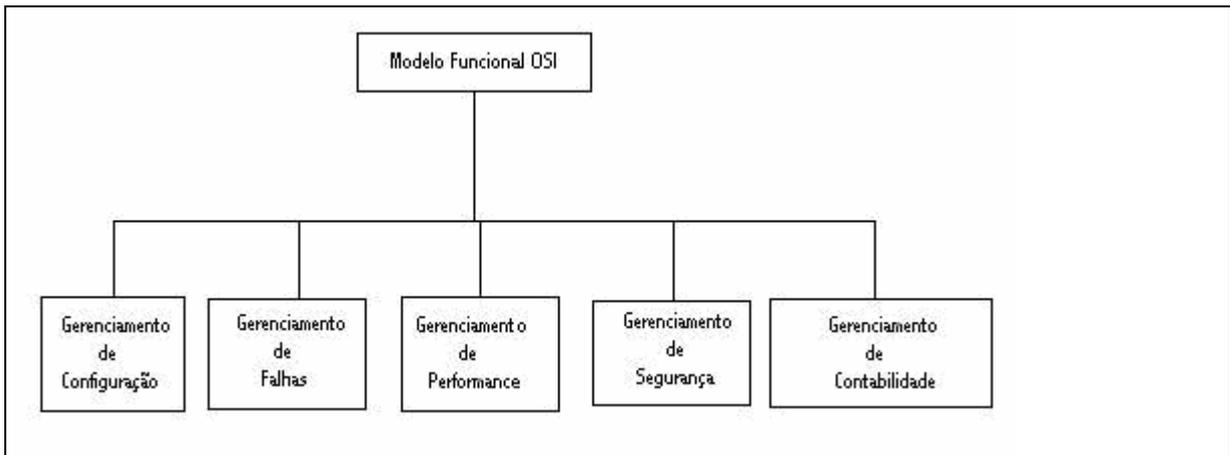


Fonte: Santos (2002).

Figura 8 – Protocolos de Transferência – Comunicação

### 2.1.5 Modelo Funcional

De acordo com Santos (2002), trata-se de um modelo que trabalha com requerimentos de gerenciamento orientado ao usuário. O modelo OSI define 5 áreas como pode ser observado na figura 9.



Fonte: Santos (2002).

Figura 9 - Modelo Funcional de Gerenciamento de Redes

**Gerenciamento de Configuração:** refere-se aos ajustes e mudanças das configurações de redes e de seus componentes.

**Gerenciamento de Falhas:** responsável pela detecção e isolamento dos problemas causadores de falhas na rede.

**Gerenciamento de Desempenho:** monitora o comportamento da rede.

**Gerenciamento de Segurança:** responsável por aspectos relacionados a segurança na comunicação das redes.

**Gerenciamento de Contabilidade:** lida com os privilégios dos usuários da rede, bem como administra os custos da rede estabelecendo métricas para tarifar o uso de recursos e serviços.

## 2.2 PROTOCOLO SNMP

Segundo Tanenbaum (1997, p. 718), no início das redes de computadores, se o acesso a um computador se tornasse muito grande e este problema fosse detectado pelo usuário, o mesmo executaria o programa Ping tirando assim o pacote de seu destino. Eram então verificados os timbres de hora no cabeçalho do pacote que retornava, e assim poderia ser localizado o problema facilmente e uma providência poderia ser tomada. Outro fator favorável era o pequeno número de roteadores: dessa forma era viável executar o Ping em todos eles a fim de descobrir sua situação.

Com a transformação da rede de computadores existente até então na atual internet com seus diversos *backbones* (conexões de alta velocidade) e operadores, a solução existente

deixou de ser a mais apropriada e surgiu então uma grande necessidade da criação de novas ferramentas para seu gerenciamento. Em maio de 1990 foi publicada a RFC 1157 definindo a versão 1 do SNMP (TANENBAUM, 1997, p. 719).

De acordo com Dias e Junior (2002), o SNMP é um protocolo de gerência definido a nível de aplicação, e é utilizado para obter informações de servidores SNMP – agentes espalhados numa rede baseada na pilha de protocolos TCP/IP. As informações são obtidas através de requisições de um gerente para um ou vários agentes que utilizam os serviços do protocolo de transporte UDP para o envio e recepção das mensagens.

Ainda segundo Dias e Junior (2002), os comandos do SNMP são limitados e baseados no mecanismo de busca/alteração. Neste mecanismo estão disponíveis as operações de alteração de um valor de um objeto, de obtenção dos valores de um objeto e suas variações.

O SNMP oferecia uma forma sistemática de monitorar e gerenciar uma rede de computadores. Tal estrutura, bem como o protocolo, foram fortemente utilizados em produtos comerciais e se tornaram os padrões de fato para o gerenciamento de redes.

De acordo com Freitas (2000), o protocolo SNMP pode ser encarado como o conjunto de 3 padrões: um padrão para o formato de mensagens, um conjunto padrão de objetos gerenciados e um modo padrão de adicionar objetos usando objetos já existentes e outros novos para resolver casos específicos.

Em relação ao padrão de formato de mensagens, Freitas (2000) descreve que o padrão SNMP define 4 tipos básicos de PDUs (*Protocol Data Unit*). No SNMPv2 são incluídos mais dois tipos. Pode-se observar isso no quadro 1, onde estão descritos os tipos de mensagens e uma breve descrição de cada uma:

<b>Mensagem</b>	<b>Descrição</b>
Get-Request	Lê o valor de uma ou mais variáveis
Get-Next-Request	Lê o valor da variável seguinte
Get-Bulk-Request	Busca uma tabela
Set-Request	Atualiza uma ou mais variáveis
Inform-Request	Confirmação do recebimento da PDU de trap
Trap	Aviso de <i>trap</i> do agente para o gerente

Fonte: Freitas(2000).

Quadro 1 – Padrão de Formato de Mensagens SNMP

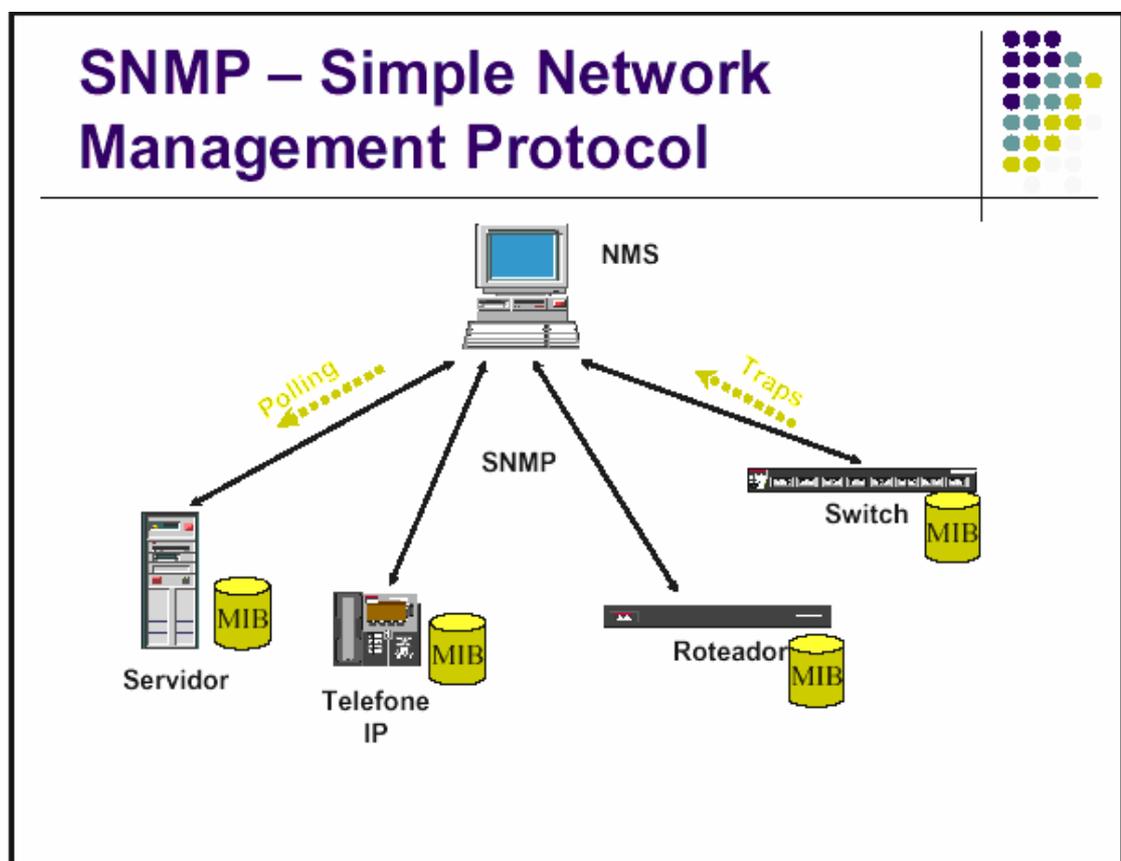
Com o passar do tempo os usuários vão aprimorando-se no uso do SNMP e suas deficiências começaram a tornarem-se evidentes, forçando assim o surgimento do SNMPv2 que por sua vez tornou-se um padrão para a internet.

## 2.3 O MODELO SNMP

Segundo Tanenbaum (1997, p. 719), o modelo SNMP é composto por quatro componentes, os quais são:

- a) nós Gerenciados;
- b) estações de gerenciamento;
- c) informações de gerenciamento;
- d) um protocolo de gerenciamento.

Um exemplo do modelo SNMP, pode ser visto na Figura 10.



Fonte: Desconhecida.

Figura 10 – Modelo SNMP

*Hosts*, roteadores, pontes, impressoras ou outro dispositivo de comunicação qualquer capazes de comunicar informações de *status* para o mundo externo, podem ser definidos como nós gerenciados. De acordo com Tanenbaum (1997, p. 719), para que um nó possa ser gerenciado diretamente através do protocolo SNMP, ele deve ser capaz de executar um processo de gerenciamento SNMP, o qual é denominado de agente SNMP.

As estações de gerenciamento são computadores genéricos que executam um software

especial. Estas estações de gerenciamento possuem um ou vários processos que comunicam-se com os agentes espalhados por toda a rede, enviando comandos e recebendo respostas. De acordo com Tanenbaum (1997, p. 720), toda a inteligência fica nas estações de gerenciamento. Com isso os agentes podem ser mais simples, tornando possível a redução do impacto sobre os dispositivos que estão sendo executados. Ainda segundo Tanenbaum (1997, p. 720), muitas estações de gerenciamento têm uma interface gráfica para permitir que o gerente da rede inspecione seu *status* e tome providências caso seja necessário.

Uma rede de computadores possui diversos equipamentos, tais como roteadores, *hosts* e impressoras por exemplo, que são produzidos por vários fabricantes diferentes. Para que todos os equipamentos possam comunicarem-se uns com os outros de forma eficiente, é preciso que a natureza das informações mantidas neles, seja rigidamente especificada (TANENBAUM, 1997, p. 720).

Segundo Tanenbaum (1997, p. 720), o protocolo SNMP descreve de forma muito detalhada as informações exatas que cada tipo de agente deve manter e o formato a ser aplicado a essas informações. Cada dispositivo mantém uma ou mais variáveis que descrevem seu estado.

Todos os dispositivos de uma rede possuem variáveis que guardam informações de seu *status*. Estas variáveis são chamadas de objetos, e o conjunto de objetos de uma rede é armazenado em uma estrutura de dados chamada de MIB.

Através do protocolo SNMP, a estação de gerenciamento comunica-se com os objetos obtendo informações sobre os mesmos, podendo fazer alterações neles.

De acordo com Tanenbaum (1997, p. 721), falhas podem ocorrer nos dispositivos espalhados pela rede, e quando isso acontece, o dispositivo informa às estações de gerenciamento cabendo a estas solicitar maiores informações do problema ocorrido. Tal problema é reportado num relatório que é gerado e que possui o nome de *trap*.

A comunicação entre nós gerenciados e estação de gerenciamento não é confiável, pois não é confirmada, por isso é importante que a estação de gerenciamento consulte cada nó ocasionalmente, para verificar se existe algum evento anormal. Segundo Tanenbaum (1997, p. 721), o modelo que executa um *polling* (consulta sequencial) em longos intervalos, havendo uma aceleração quando há o recebimento de uma *trap*, é chamado de *trap directed polling*.

Tanenbaum (1997, p. 721) explica que dispositivos mais antigos ou dispositivos que não foram projetados originalmente para uso em uma rede, não disponham dos recursos que o tornam capaz de executar um agente SNMP, e para tratar este problema o SNMP define um agente proxy, que nada mais é do que um agente que controla um ou mais dispositivos não-

SNMP e que faz a comunicação com a estação de gerenciamento a fim de interceder por esses dispositivos.

De acordo com Tanenbaum (1997, p. 721), a segurança e autenticação no protocolo SNMP possui papel muito importante. A estação de gerenciamento tem capacidade de obter muitas informações sobre todos os nós que estão sob seu controle e pode também desativar todos eles. Por isso é preciso que os agentes sejam convencidos que as consultas que estão vindo estejam realmente sendo solicitadas pela estação de gerenciamento.

Na versão 1 do protocolo SNMP, a estação de trabalho colocava uma senha (em texto simples) em cada mensagem, para provar a autenticidade. Na versão 2, a segurança foi melhorada consideravelmente fazendo-se o uso de técnicas modernas de criptografia, isso contribui para que esta versão do SNMP se tornasse ainda mais pesada.

Segundo Tanenbaum (1997, p. 722), o coração do modelo SNMP é o conjunto de objetos que é gerenciado, lido e gravado pela estação de trabalho. É preciso que a definição dos objetos que serão gerenciados seja feita de forma padronizada e neutra, devido aos diferentes fornecedores dos equipamentos, caso contrário não é possível a comunicação entre os mesmos. Outro aspecto é que a codificação dos objetos deve ser padronizada para que possam trafegar através da rede de dados. De acordo com Tanenbaum (1997, p. 722), é preciso uma linguagem de definição de objetos padronizada, bem como as regras de codificação. O protocolo SNMP utiliza uma linguagem retirada do modelo OSI, chamada de ASN.1 (*Abstract Syntax Notation One*).

“Basicamente, a ASN.1 é uma linguagem de declaração de dados muito primitiva. Ela permite que o usuário defina objetos primitivos e combine-os em objetos mais complexos” (TANENBAUM, 1997, p. 722).

No quadro 2 é possível observar os tipos de dados básicos da ASN.1.

<b>Tipo de primitiva</b>	<b>Significado</b>	<b>Código</b>
INTEGER	Inteiro de tamanho arbitrário	2
BIT STRING	Um string de 0 ou mais bits	3
OCTET STRING	Um string de 0 ou mais bytes sem sinal	4
NULL	Um marcador de lugar	5
OBJECT IDENTIFIER	Um tipo de dados oficialmente definido	6

Fonte: Tanenbaum (1997).

Quadro 2 – Os tipos de dados primitivos da ASN.1 permitidos no SNMP

Segundo Tanenbaum (1997, p. 726), a sintaxe de transferência da ASN.1 define como

os valores dos tipos ASN.1 são convertidos sem qualquer ambigüidade em uma seqüência de bits para transmissão. A sintaxe de transferência usada pela ASN.1 é chamada de BER (*Basic Encoding Rules*).

## 2.4 SNMPV2

Nesta seção verificam-se alguns aspectos relevantes do SNMPV2 (segunda versão do protocolo SNMP).

Segundo Freitas (2000), a segunda versão do protocolo SNMP possui várias melhorias, como por exemplo permitir que sejam especificadas variáveis com mais detalhes, permitindo inclusive o uso de uma tabela de estrutura de dados para recuperação de dados mais facilmente.

Ainda de acordo com Freitas (2000), foram adicionadas tantas características a nova versão do SNMP, que sua especificação passou de 36 páginas (versão 1) para 416 páginas na versão 2 (dois).

Foram criadas melhorias na segurança, onde foram inseridas técnicas de criptografia na transmissão de senhas de autenticação do gerente, dificultando assim o acesso de intrusos.

Segundo Freitas (2000), a MIB também foi melhorada passando para uma nova MIB designada de MIB-II.

Apesar de muitas melhorias nesta nova versão, o SNMPv2 está vivo teoricamente pois seus criadores (Caso, McCloghrie, Rosa e Waldbusser) não puderam concordar em vários pontos chave do protocolo e também porque ficou complexo demais, em relação ao SNMP original.

Conforme Freitas (2000), é difícil encontrar agentes que suportem totalmente o padrão v2. Muitos agentes apóiam as extensões de segurança, contudo oferecendo-as à versão 1.

Em 1997 foi lançado o SNMPv3. Esta versão possui muitos avanços em questões de segurança, porém ainda não está sendo utilizado (FREITAS, 2000).

## 2.5 MANAGEMENT INFORMATION BASE (MIB)

Granville (2006) define MIB como sendo uma base de dados conceitual. Os dados podem estar armazenados em um SGBD como por exemplo a taxa de utilização de um *link*, ou então os dados podem ser encontrados nos próprios recursos, como o estado atual de uma interface por exemplo. A MIB é representada como uma árvore de dados estruturada. Os nodos intermediários contém sub-nodos mas não contém nenhum valor associado. Um nodo que não possui sub-nodos é chamado de objeto e possui um valor associado.

De acordo com Tanenbaum (1997, p. 731), os objetos gerenciados pelo SNMP que estão definidos na MIB, são agrupados em dez categorias que correspondem a dez nós sob mib-2 como se pode observar no quadro 3.

<b>Grupo</b>	<b>Nº de Objetos</b>	<b>Descrição</b>
System	7	Nome, local e descrição do equipamento
Interfaces	23	Interfaces de rede e seu tráfego
AT	3	Conversão de endereço (obsoleto)
IP	42	Estatísticas de pacotes IP
ICMP	26	Estatísticas sobre as mensagens ICMP recebidas
TCP	19	Algoritmos TCP, parâmetros e estatísticas
UDP	6	Estatísticas de tráfego UDP
EGP	20	Estatísticas de tráfego de protocolo de gateway externo
Transmission	0	Reservado para MIDs de meios físicos específicos
SNMP	29	Estatísticas de tráfego SNMP

Fonte: Tanenbaum (1997).

Quadro 3 – Os grupos de objetos na MIB-II da internet

Segundo Tanenbaum (1997, p. 732), o grupo *System* dispõe de informações como nome do dispositivo, quem o fabricou, o que o hardware e o software contém, onde ele está localizado e o nome e o endereço da pessoa de contato também são fornecidos.

O grupo *Interfaces* é responsável por fornecer número de pacotes e bytes enviados e recebidos da rede, o número de descartes, o número de difusões e o tamanho da fila de saída.

Tanenbaum (1997, p. 732) ainda define os outros grupos, como o grupo AT, responsável por fornecer informações sobre o mapeamento de endereços.

O grupo IP controla o tráfego IP recebido e emitido, possuindo uma grande quantidade

de contadores que controlam o número de pacotes descartados por diversas razões possíveis.

Grupo ICMP, nada mais é do que um contador de mensagens de erro IP.

O grupo TCP faz o controle do número atual e o cumulativo de conexões abertas, segmentos enviados e recebidos e diversas estatísticas de erros.

O grupo UDP registra o número de datagramas UDP enviados e recebidos, bem como os que não foram entregues devido algum erro.

O grupo EGP faz o controle de um determinado tipo de pacotes, quantos foram enviados, recebidos e encaminhados corretamente, e quantos foram recebidos e descartados.

O grupo *Transmission* é um marcador de lugar para MIBs de meios físicos e específicos.

O último grupo, o SNMP é destinado ao cálculo de estatísticas sobre a operação do próprio SNMP.

De acordo com Tanenbaum (1997, p. 733), a MIB-II é formalmente definida na RFC 1213.

## 2.6 SERVIDOR WEB APACHE

De acordo com Alecrim (2006), quando se acessa um site na internet, por trás daquele endereço existe um servidor responsável por disponibilizar as páginas e recursos que aquele site possui. Sendo assim, ao enviar um e-mail através de um formulário, colocar uma mensagem em algum fórum de discussão, fazer uma compra *on-line*, ou outra operação qualquer, existe um servidor ou um conjunto de servidores *web* que processam todas as informações necessárias para o tipo de transação que está sendo trabalhada.

Alecrim (2006) esclarece que um servidor *web* é um computador que processa solicitações *Hyper Text Transfer Protocol* (HTTP), que é um dos protocolos padrões para internet. Ao utilizar um navegador para acessar páginas da internet, este faz solicitações via HTTP ao servidor, e o servidor por sua vez processa as solicitações e encaminha os resultados ao navegador novamente. No caso do servidor Apache, além de processar o protocolo HTTP ele comporta o processamento de vários outros protocolos.

Segundo Alecrim (2006), o Apache é o servidor *web* mais conhecido e utilizado mundialmente, sendo responsável por mais de 60% das utilizações. Estão relacionados a isso o fato de ter um excelente desempenho e prover segurança e compatibilidade com diversas

plataformas e todos os seus recursos.

O Apache surgiu no *National Center of Supercomputing Applications* (NCSA) através do trabalho de Rob McCool. Quando saiu da NCSA, McCool parou de trabalhar no projeto e a partir daí várias pessoas e grupos passaram a adaptar o servidor às suas necessidades. Os principais responsáveis pela retomada do projeto foram Brian Behlendorf e Cliff Skolnick que foram em seguida apoiados por Brandon Long e Beth Frank. Estes últimos tinham a tarefa de continuar o projeto pela NCSA, e não demorou para eles se juntarem ao Apache Group.

Segundo Alecrim (2006), a primeira versão oficial do Apache (0.6.2) foi lançada em 1995. Dentre as modificações/evoluções do Apache, uma das mais importantes que deve ser citada é a criação da arquitetura *Shambhala* desenvolvida por Robert Thau. Após uma série de modificações para aperfeiçoamento, testes e uma documentação nova (feita por David Robinson), em dezembro de 1995 é lançado então o Apache 1.0.

Alecrim (2006) descreve ainda as principais características do Apache (extraído do Guia Foca Linux), os quais são:

- a) possui suporte a *scripts* CGI usando linguagens como Perl, PHP, Shell Script, ASP, etc;
- b) suporte a autorização de acesso podendo ser especificadas restrições de acesso separadamente para cada endereço/arquivo/diretório acessado no servidor;
- c) autenticação requerendo um nome de usuário e senha válidos para acesso a alguma página/sub-diretório/arquivo (suportando criptografia via Crypto);
- d) negociação de conteúdo, permitindo a exibição de páginas *web* no idioma requisitado pelo cliente navegador;
- e) suporte a tipos MIME;
- f) personalização de logs;
- g) mensagens de erro;
- h) suporte a *virtual hosting* (é possível servir 2 ou mais domínios com endereços/portas diferentes através do mesmo processo ou usar mais de um processo para controlar mais de um endereço);
- i) suporte a servidor proxy FTP e HTTP, com limite de acesso, *caching* (todas flexivelmente configuráveis);
- j) suporte a proxy e redirecionamentos baseados em URLs para endereços internos;
- k) suporte a criptografia via SSL através de certificados digitais.

Além disso, implementa um mecanismo de tratamento de módulos *Dynamic Shared Objects* (DSO) que permitem adicionar/remover funcionalidades e recursos sem necessidade

de re-compilação do programa.

### 3 DESENVOLVIMENTO DO PROTÓTIPO

Este capítulo apresenta o desenvolvimento do protótipo, demonstrando a seqüência de etapas trabalhadas até a sua conclusão. No capítulo anterior apresentou-se a fundamentação teórica, a qual é responsável pelo embasamento científico para o desenvolvimento do trabalho. A etapa seguinte é a de levantamento dos requisitos, onde são verificados de forma textual quais os requisitos funcionais e requisitos não funcionais necessários ao desenvolvimento. A seguir é feita a especificação do protótipo através de diagramas e/ou modelos lógicos que representam o problema a ser resolvido.

A etapa seguinte apresenta a implementação do projeto, mostrando de que forma o mesmo foi construído. Logo após a descrição da implementação tem-se uma consideração referente às técnicas e ferramentas utilizadas, tendo como etapa seguinte os testes que são realizados para encontrar possíveis falhas.

Por fim é realizada uma análise dos resultados obtidos com o projeto, fazendo-se comentários sobre tais resultados.

#### 3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O protótipo deve permitir o monitoramento do servidor *web* Apache através de consultas feitas às informações constantes na MIB. Citam-se como principais requisitos do protótipo, designados em Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF):

- a) prover acesso ao protótipo através de um nome de usuário e senha pré-cadastrados, dando assim maior segurança ao sistema e limitando os usuários que podem acessar o mesmo (RF);
- b) permitir que o usuário faça o gerenciamento de mais de um servidor Apache concomitantemente (RF);
- c) permitir que o usuário visualize um *log* do sistema (RF);
- d) permitir ao usuário consultar estatísticas de informações como tráfego total por exemplo, em forma de gráficos (RF);

- e) o tempo de resposta às requisições feitas pelo protótipo à MIB do servidor gerenciado deve ser aceitável, não sendo superior a três segundos (a conexão de internet utilizada influencia diretamente no tempo desta transação) (RNF);
- f) permitir o acesso desde qualquer ponto da internet, para que se possa trabalhar o gerenciamento do mesmo (RNF);
- g) o servidor *web* a ser gerenciado deve ser o Apache (RNF);
- h) a implementação da visualização deve ser feita com a linguagem PHP (RNF);
- i) possibilitar fácil entendimento ao usuário através de interface facilitada com menus para cada opção disponível no protótipo (RNF).

## 3.2 ESPECIFICAÇÃO

Utilizou-se para a especificação do projeto uma metodologia orientada a objetos, representada através de diagramas que fazem parte da *Unified Modeling Language* (UML), tendo como ferramenta de suporte a esta especificação a *Enterprise Architect*, que foi utilizada para a construção do diagrama de casos de uso, diagrama de classes e diagrama de atividades.

Para a criação do modelo conceitual da base de dados foi utilizada a ferramenta brModelo 2.0, Cândido (2005).

### 3.2.1 Diagrama de Casos de Uso

Caso de uso tem como propósito principal descrever de forma conceitual a estrutura do protótipo (FURLAN, 1998, p. 169).

Na figura 11 observa-se os principais casos de uso do protótipo.

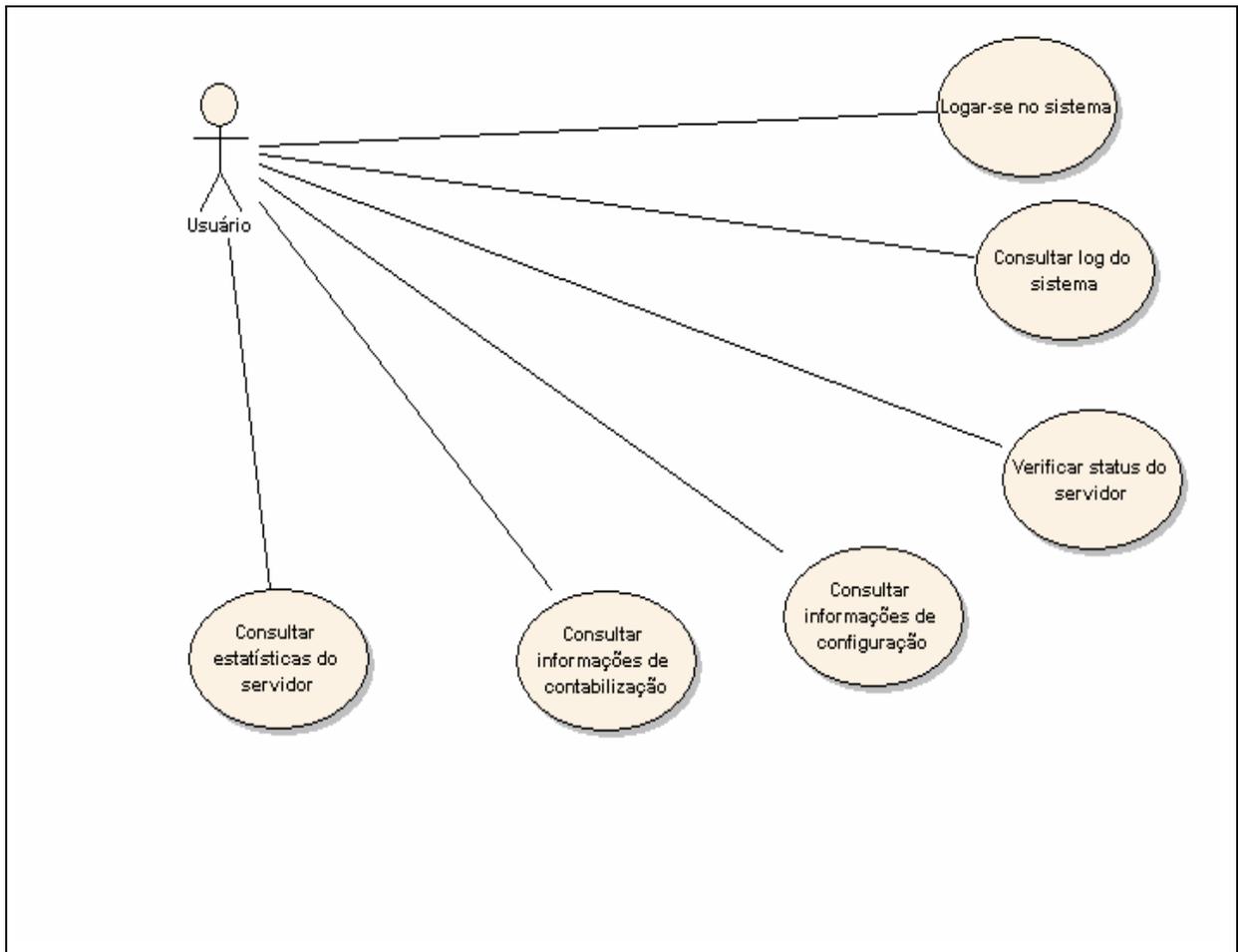


Figura 11 – Diagrama de Caso de Uso

No Quadro 4 apresenta-se o caso de uso “Logar-se no sistema”.

Descrição	Usuário acessa aplicação via navegador internet.
Ator	Usuário.
Pré-condição	Scripts da aplicação devem estar hospedados no servidor <i>web</i> .
Fluxo principal	<ul style="list-style-type: none"> <li>a) abrir o navegador de internet;</li> <li>b) abrir endereço do protótipo;</li> <li>c) fornecer nome de usuário e senha;</li> <li>d) submeter os dados para validação na base de dados.</li> </ul>
Fluxo alternativo (a)	<ul style="list-style-type: none"> <li>a) nome de usuário e/ou senha inválido (s)</li> <li>- tela com mensagem “usuário ou senha inválida” é carregada.</li> </ul>
Pós-condição	Usuário terá acesso a todas as funcionalidades do sistema.

Quadro 4 – Descrição do Caso de Uso Logar-se no sistema

No quadro 5 apresenta-se o caso de uso “Consultar *log* do sistema”.

Descrição	Usuário acessa o <i>link</i> consultar log.
Ator	Usuário.
Pré-condição	Informações de log devem estar disponíveis na base de dados.
Fluxo principal	<ul style="list-style-type: none"> <li>a) escolher a opção de consultar log no menu do sistema;</li> <li>b) log apresentado na tela.</li> </ul>
Pós-condição	Informações de log são apresentadas no navegador para visualização.

Quadro 5 – Descrição do Caso de Uso Consultar Log do Sistema

No quadro 6 é apresentado o caso de uso “Verificar *status* do servidor”.

Descrição	Usuário acessa o <i>link Status</i> .
Ator	Usuário.
Pré-condição	Servidor Apache devidamente configurado com suporte a MIB que possui as informações de gerência do servidor.
Fluxo principal	<ul style="list-style-type: none"> <li>a) escolher a opção <i>Status</i> no menu do sistema;</li> <li>b) apresentada página com lista de servidores disponíveis para gerenciamento;</li> <li>c) usuário opta por um dos servidores disponíveis clicando no <i>link</i> que possui nome e endereço ip do servidor;</li> <li>d) é apresentada página que informa se o servidor escolhido está: <ul style="list-style-type: none"> <li>- executando;</li> <li>- parado;</li> <li>- inicializando;</li> <li>- parando;</li> <li>- não respondendo.</li> </ul> </li> </ul>
Fluxo alternativo (a)	<ul style="list-style-type: none"> <li>a) servidor escolhido não está executando <ul style="list-style-type: none"> <li>- clicar no menu <i>Status</i> e escolher outro servidor para verificar <i>status</i>;</li> <li>- escolher outra opção no menu do sistema.</li> </ul> </li> </ul>
Pós-condição	Apresentado <i>status</i> do servidor escolhido.

Quadro 6 – Descrição do Caso de Uso Verificar *Status* do Servidor

No quadro 7 apresenta-se o caso de uso “Consultar Informações de Configuração”.

Descrição	Usuário acessa o <i>link</i> Configuração no menu do sistema.
Ator	Usuário.
Pré-condição	<ul style="list-style-type: none"> <li>a) servidor Apache devidamente configurado com suporte a MIB que possui as informações de gerência do servidor;</li> <li>b) servidor escolhido deve estar com <i>status</i> de executando.</li> </ul>
Fluxo principal	<ul style="list-style-type: none"> <li>a) escolher a opção Inf.de Configuração no menu do sistema;</li> <li>b) apresentada página com lista de servidores disponíveis para gerenciamento;</li> <li>c) usuário opta por um dos servidores disponíveis clicando no <i>link</i> que possui nome e endereço ip do servidor;</li> <li>d) é apresentada página com informações de configuração do servidor: <ul style="list-style-type: none"> <li>- número ip do servidor;</li> <li>- versão do servidor;</li> <li>- data da versão (compilação) do servidor;</li> <li>- diretório onde está instalado o servidor;</li> <li>- <i>PIDFILE</i> do processo do servidor;</li> <li>- diretório para arquivos temporários;</li> <li>- agente HttpAddress;</li> <li>- total de portas do servidor.</li> </ul> </li> </ul>
Fluxo alternativo (a)	<ul style="list-style-type: none"> <li>a) erro de comunicação com o servidor: <ul style="list-style-type: none"> <li>- nova tentativa clicando novamente no link correspondente;</li> <li>- escolha de outra opção de gerência.</li> </ul> </li> </ul>
Pós-condição	Apresentadas informações referente configuração do servidor no navegador.

Quadro 7 – Descrição do Caso de Uso Consultar Informações de Configuração

O quadro 8 descreve o caso de uso “Consultar Informações de Contabilização”.

Descrição	Usuário acessa o <i>link</i> Contabilização no menu do sistema.
Ator	Usuário.
Pré-condição	<ul style="list-style-type: none"> <li>a) servidor Apache devidamente configurado com suporte a MIB que possui as informações de gerência do servidor;</li> <li>b) servidor escolhido deve estar com <i>status</i> de executando.</li> </ul>
Fluxo principal	<ul style="list-style-type: none"> <li>a) escolher a opção Inf.de Contabilização no menu do sistema;</li> <li>b) apresentada página com lista de servidores disponíveis para gerenciamento;</li> <li>c) usuário opta por um dos servidores disponíveis clicando no <i>link</i> que possui nome e endereço ip do servidor;</li> <li>d) é apresentada página com informações de contabilização do servidor: <ul style="list-style-type: none"> <li>- tráfego total;</li> <li>- total de acessos;</li> <li>- total de processos em execução;</li> <li>- total de processos ociosos;</li> <li>- tempo que o servidor está em execução;</li> <li>- número de requisições por segundo;</li> <li>- Kbytes por segundo;</li> <li>- Kbytes por requisição.</li> </ul> </li> </ul>
Fluxo alternativo (a)	<ul style="list-style-type: none"> <li>a) erro de comunicação com o servidor: <ul style="list-style-type: none"> <li>- nova tentativa clicando novamente no link correspondente;</li> <li>- escolha de outra opção de gerência.</li> </ul> </li> </ul>
Pós-condição	Apresentadas informações referente dados de contabilização do servidor.

Quadro 8 – Descrição do Caso de Uso Consultar Informações de Contabilização

O caso de uso “Consultar Estatísticas do Servidor” está ilustrado no quadro 9.

Descrição	Usuário acessa o <i>link</i> Estatísticas.
Ator	Usuário.
Pré-condição	a) informações para geração de estatísticas devem estar armazenadas na base de dados do sistema.
Fluxo principal	<p>a) escolher a opção Estatísticas no menu do sistema;</p> <p>b) apresentada página com lista de servidores disponíveis para consulta de estatísticas;</p> <p>c) usuário opta por um dos servidores disponíveis clicando no <i>link</i> que possui nome e endereço ip do servidor;</p> <p>d) é apresentada página com estatísticas do servidor escolhido:</p> <ul style="list-style-type: none"> <li>- tráfego total;</li> <li>- total de acessos;</li> <li>- requisições por segundo;</li> <li>- kbytes por segundo.</li> </ul>
Fluxo alternativo (a)	<p>a) erro de comunicação com o servidor:</p> <ul style="list-style-type: none"> <li>- nova tentativa clicando novamente no link Estatísticas;</li> <li>- escolha de outra opção no menu do sistema.</li> </ul>
Pós-condição	Apresentadas estatísticas do servidor Apache escolhido em forma de gráficos.

Quadro 9 – Descrição do Caso de Uso Consultar Estatísticas do Servidor

### 3.2.2 Modelo Conceitual da Base de Dados

O protótipo a ser construído utiliza uma base de dados para armazenar os usuários do sistema, os servidores a serem gerenciados, dados para que se possa gerar estatísticas e informações para geração de um log das operações realizadas pelos usuários.

Na figura 12 pode-se observar o modelo conceitual da base de dados que será utilizada no projeto.

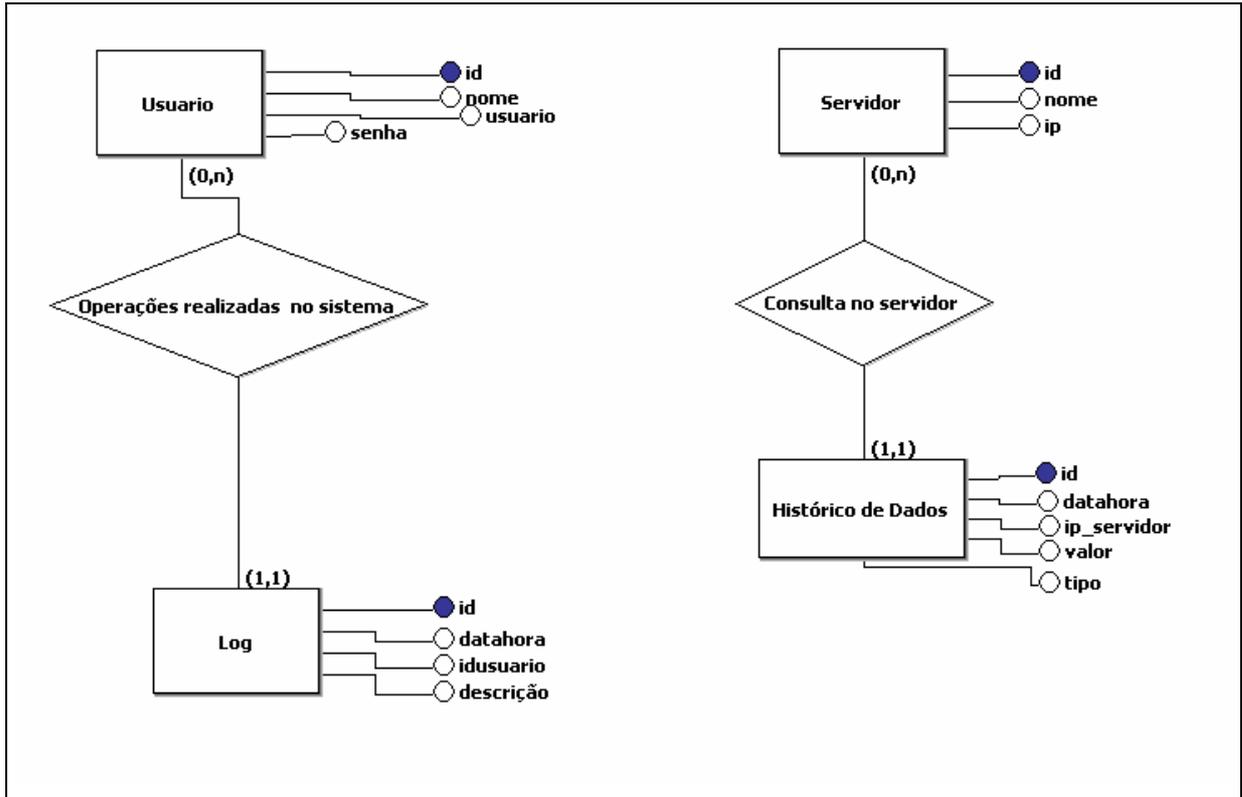


Figura 12 – Modelo Conceitual da Base de Dados do protótipo

A base de dados possui quatro tabelas conforme descrito abaixo:

- tabela de usuários – nesta tabela ficam armazenados os dados de todos os usuários que tem acesso ao sistema;
- tabela de servidores – responsável por guardar os dados (nome, endereço ip) dos servidores que são cadastrados no sistema para que possam ser gerenciados;
- tabela de histórico de dados – esta tabela guarda os dados que serão utilizados no momento que serão geradas as estatísticas do servidor através de gráficos. Guarda valores de tráfego, acessos, Kbytes por segundo e Kbytes por requisição;
- tabela de log – guarda operações realizadas no sistema, qual usuário efetuou e também a data e hora da transação.

### 3.2.3 Diagrama de Classes

A figura 13 apresenta de forma simplificada um diagrama com as principais classes seus atributos e métodos, utilizadas para construção do protótipo.

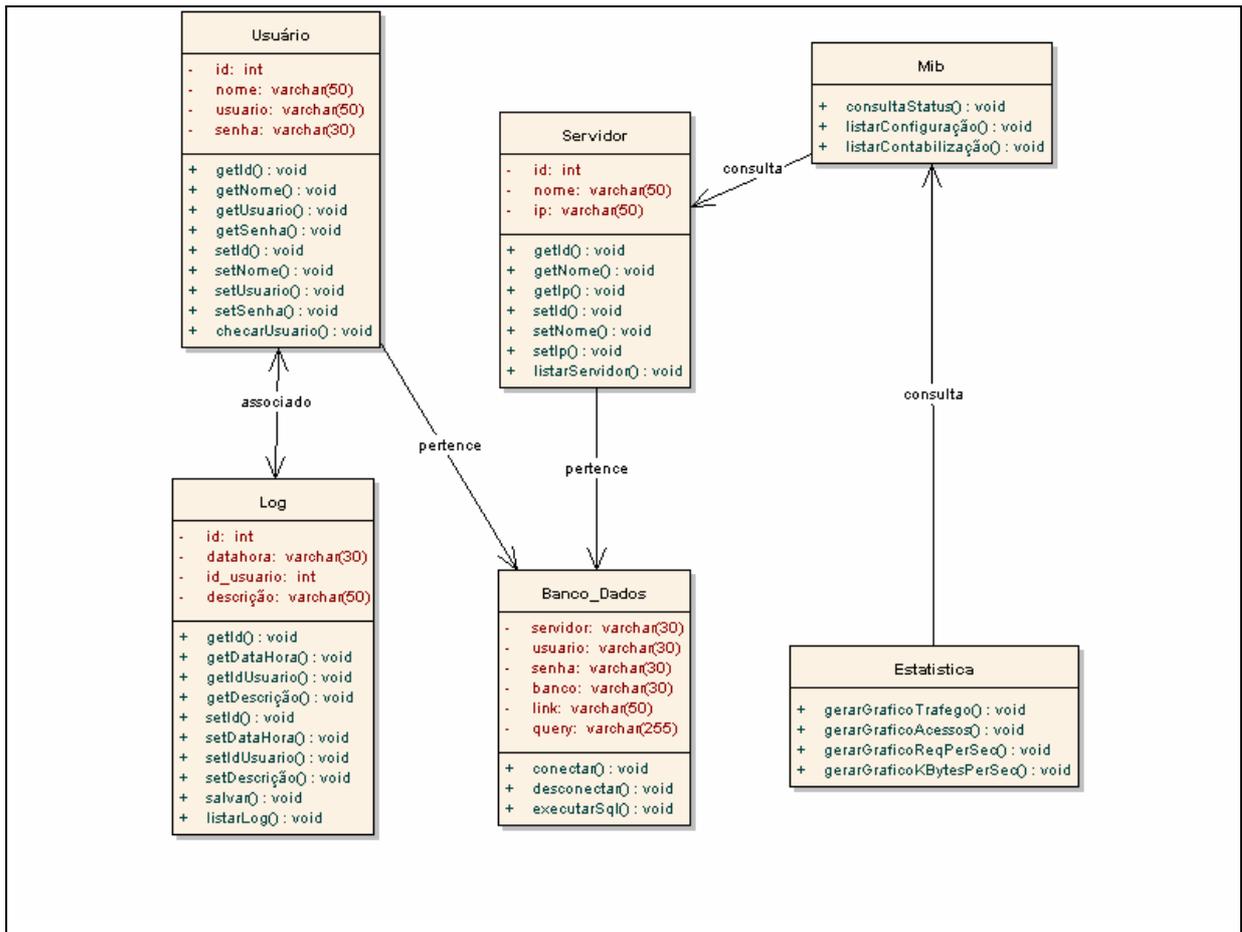


Figura 13 – Diagrama de classes

A função de cada classe no sistema está descrita abaixo:

- classe Usuário – classe que possui os atributos referentes aos usuários da ferramenta, e os métodos que são responsáveis por manipular os dados dos mesmos;
- classe Servidor – possui os atributos referentes aos servidores cadastrados no sistema bem como os métodos responsáveis pela manipulação destes atributos;
- classe Mib – esta classe possui os métodos encarregados pela consulta das informações de gerência na MIB do servidor;
- classe Log – classe que trabalha com as informações de log do sistema. É ela que possui os métodos que fazem a inserção referente às operações realizadas pelo usuário no sistema e lista-os quando solicitados no menu da ferramenta;
- classe Banco\_Dados – esta classe é usada para conexão com o banco de dados da ferramenta na inserção e consulta de todos os dados manipulados;
- classe Estatística – classe que possui os métodos que fazem a consulta na base de dados para geração dos gráficos.

### 3.2.4 Diagrama de Atividades

Na figura 14 pode-se observar um diagrama de atividades que descreve a interação do usuário com o protótipo.

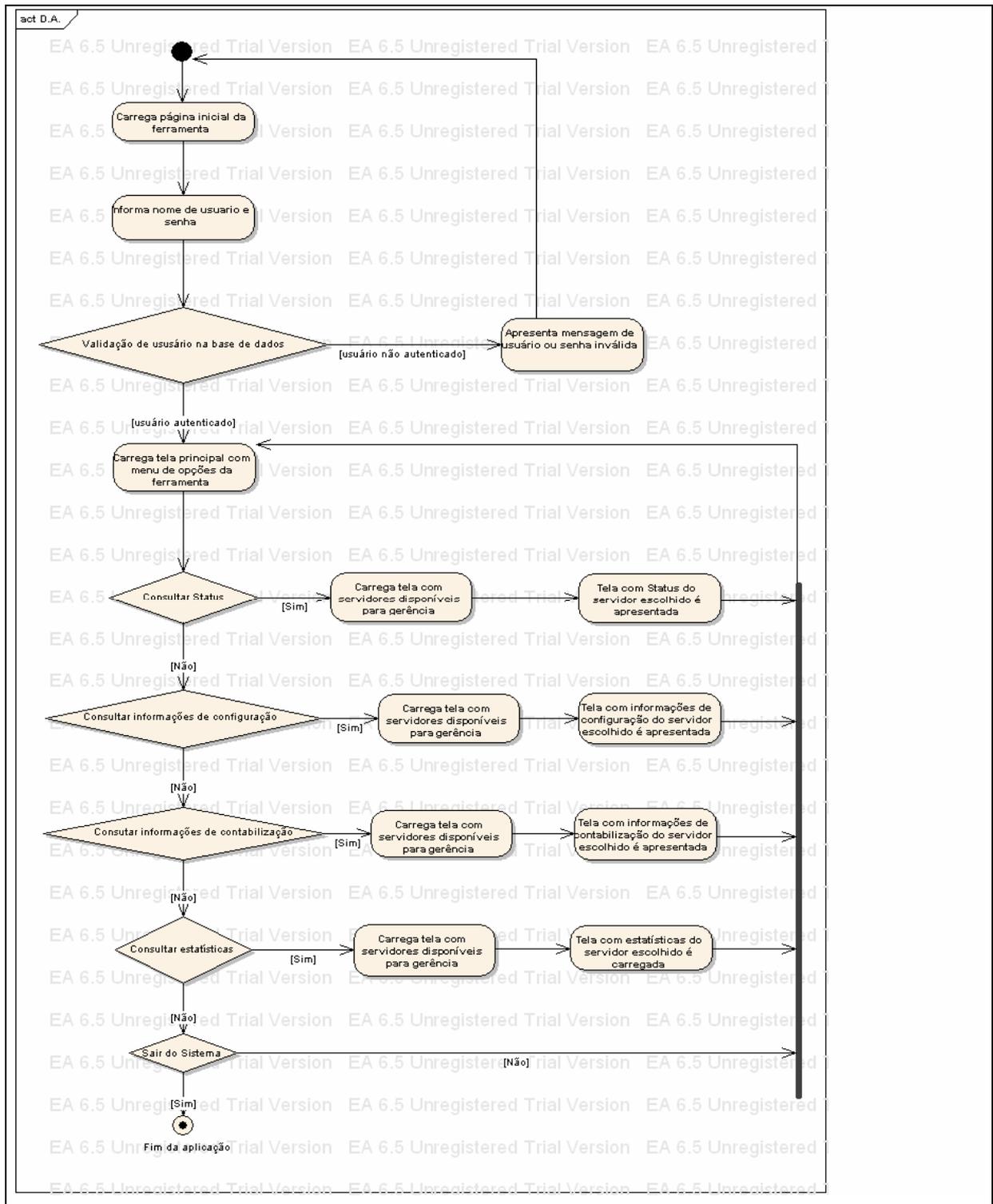


Figura 14 – Diagrama de atividades da interação do usuário com o protótipo

### 3.3 IMPLEMENTAÇÃO

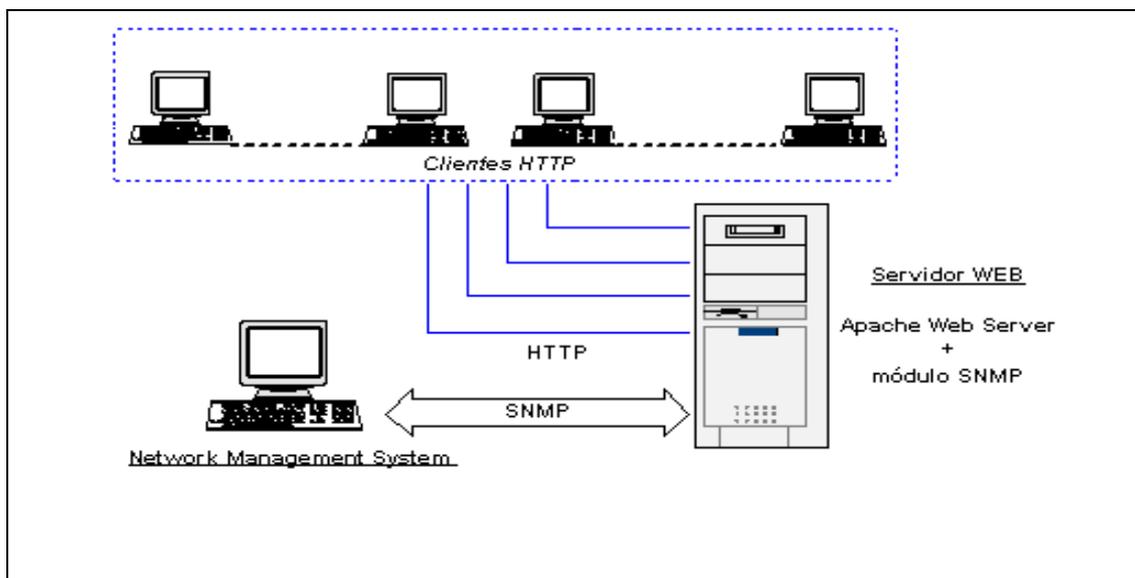
Esta seção descreve a implementação do protótipo, mostrando como o mesmo foi desenvolvido, fazendo considerações sobre as técnicas e ferramentas utilizadas, ilustrando através dos códigos implementados bem como apresentando a operacionalidade do sistema através de figuras com as principais telas do protótipo para que se tenha um maior entendimento do funcionamento e da operacionalidade do sistema.

#### 3.3.1 Técnicas e ferramentas utilizadas

As ferramentas utilizadas na implementação foram as linguagens de programação PHP, HTML e *java script*. Foi utilizado o MySQL como banco de dados do protótipo. O sistema operacional utilizado para construção e teste do projeto foi o Linux com servidor *web* Apache devidamente configurado com a MIB Pizzini (2006).

#### 3.3.2 Operacionalidade da implementação

Na figura 15 observa-se a arquitetura de funcionamento do servidor Apache com seus clientes http e o servidor para seu gerenciamento.



Fonte: Pizzini (2006).

Figura 15 – Arquitetura de funcionamento e gerenciamento do servidor Apache

Abaixo apresenta-se as principais telas do protótipo.

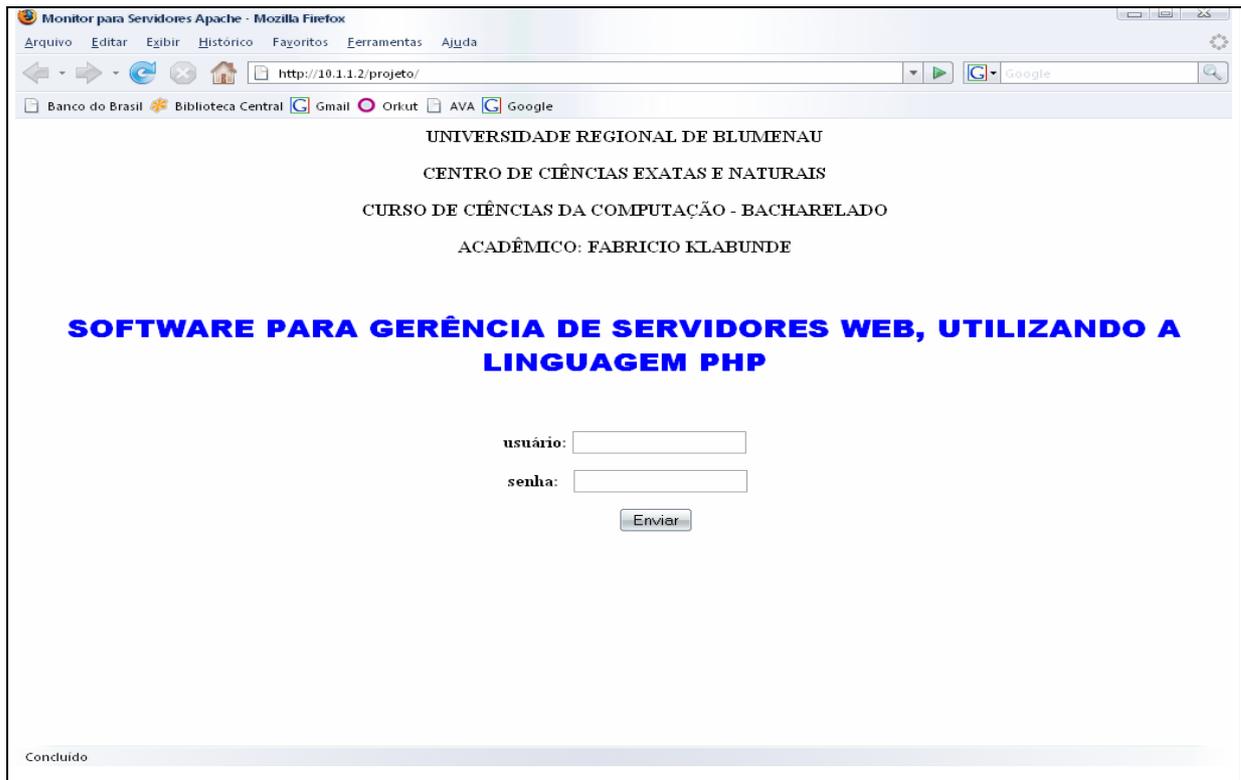


Figura 16 – Tela de login

Na figura 16 é apresentada a tela inicial, é nesta tela que o usuário informa seus dados para fazer a autenticação no protótipo de gerenciamento do servidor Apache.

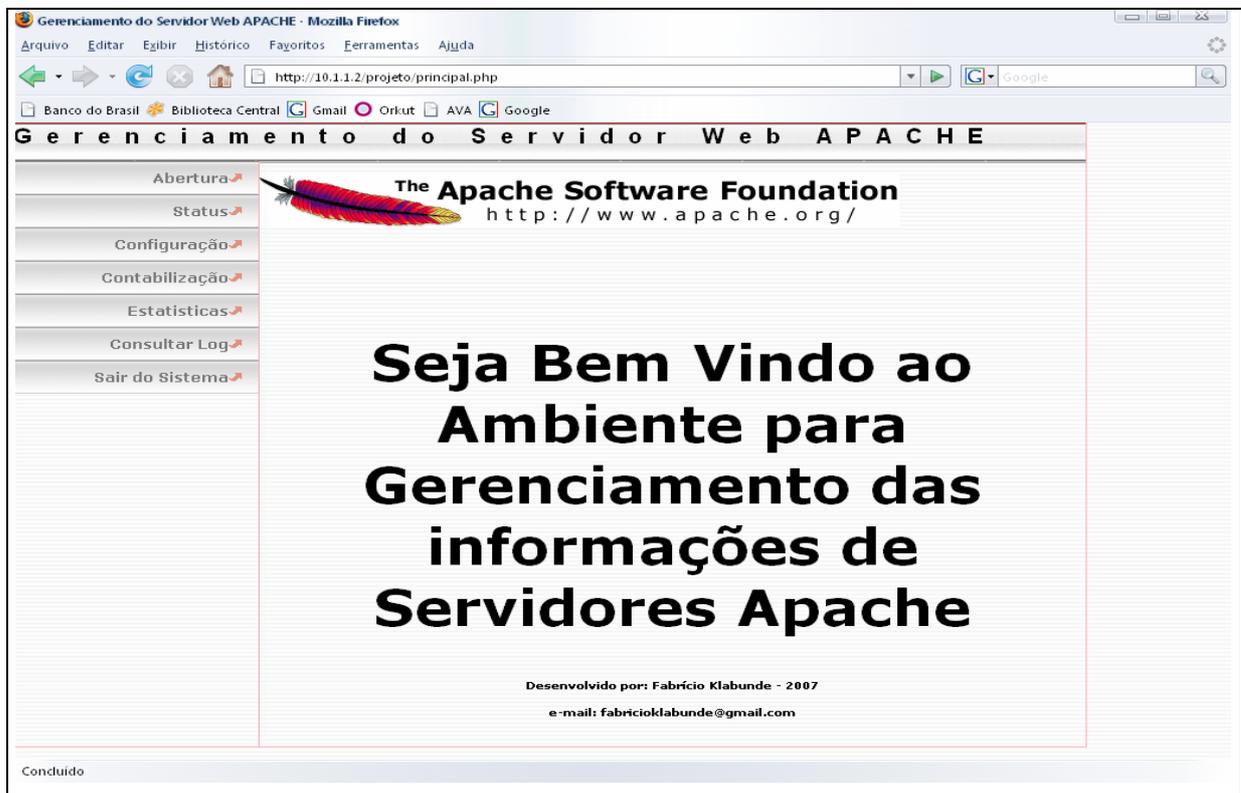


Figura 17 – Tela de Boas Vindas

Na figura 17 observa-se a tela que é carregada quando o usuário faz sua autenticação na ferramenta informando usuário e senha. Trata-se de uma tela de boas vindas do ambiente de gerenciamento e que possui o menu com todas as opções disponíveis para o usuário.

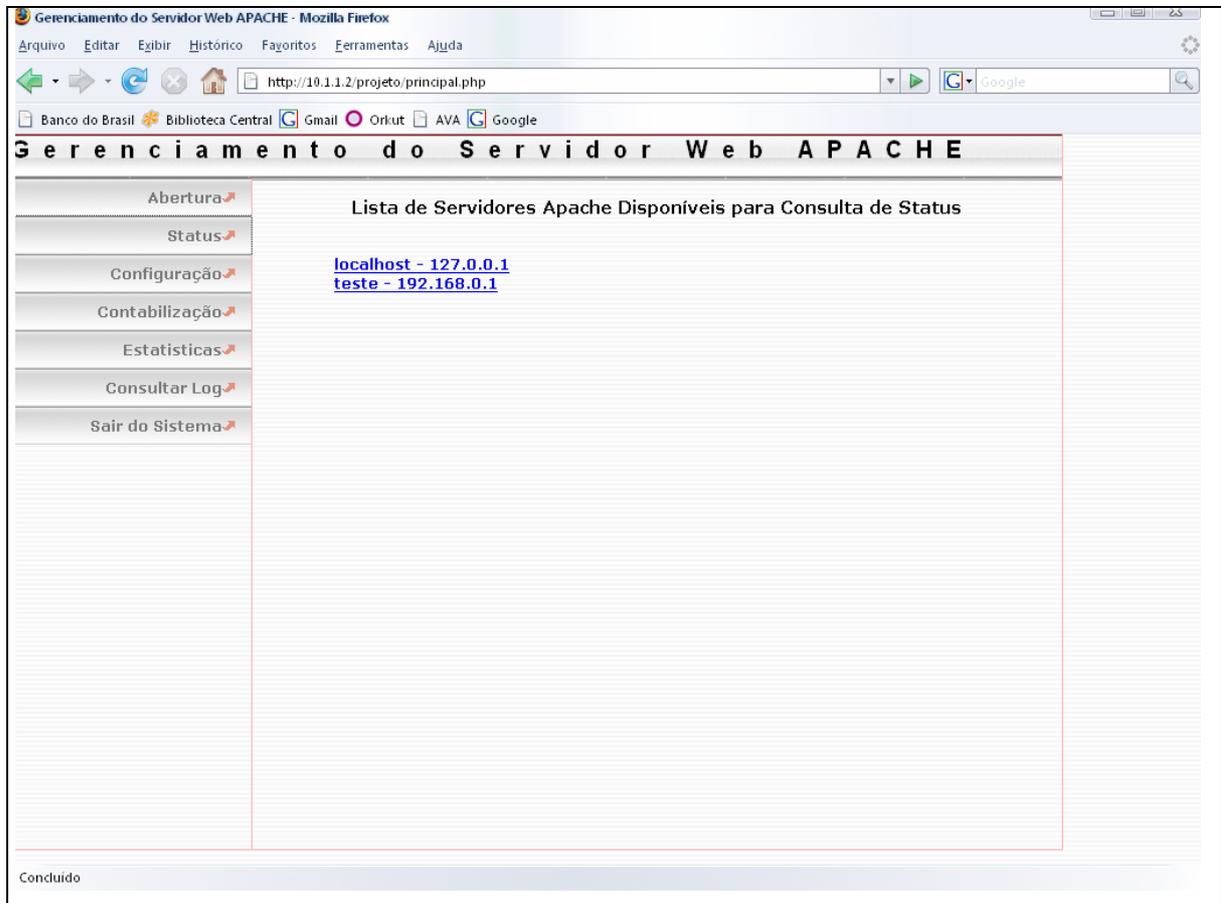


Figura 18 – Tela de escolha de qual servidor deseja consultar *status*

A figura 18 apresenta a tela que é carregada quando o usuário escolhe no menu a opção *Status*. Nessa tela é apresentada uma lista de servidores disponíveis para gerenciamento e o usuário pode optar por um servidor clicando no *link* correspondente. O mecanismo de listagem dos servidores disponíveis ocorre em todas as opções de gerenciamento do menu.

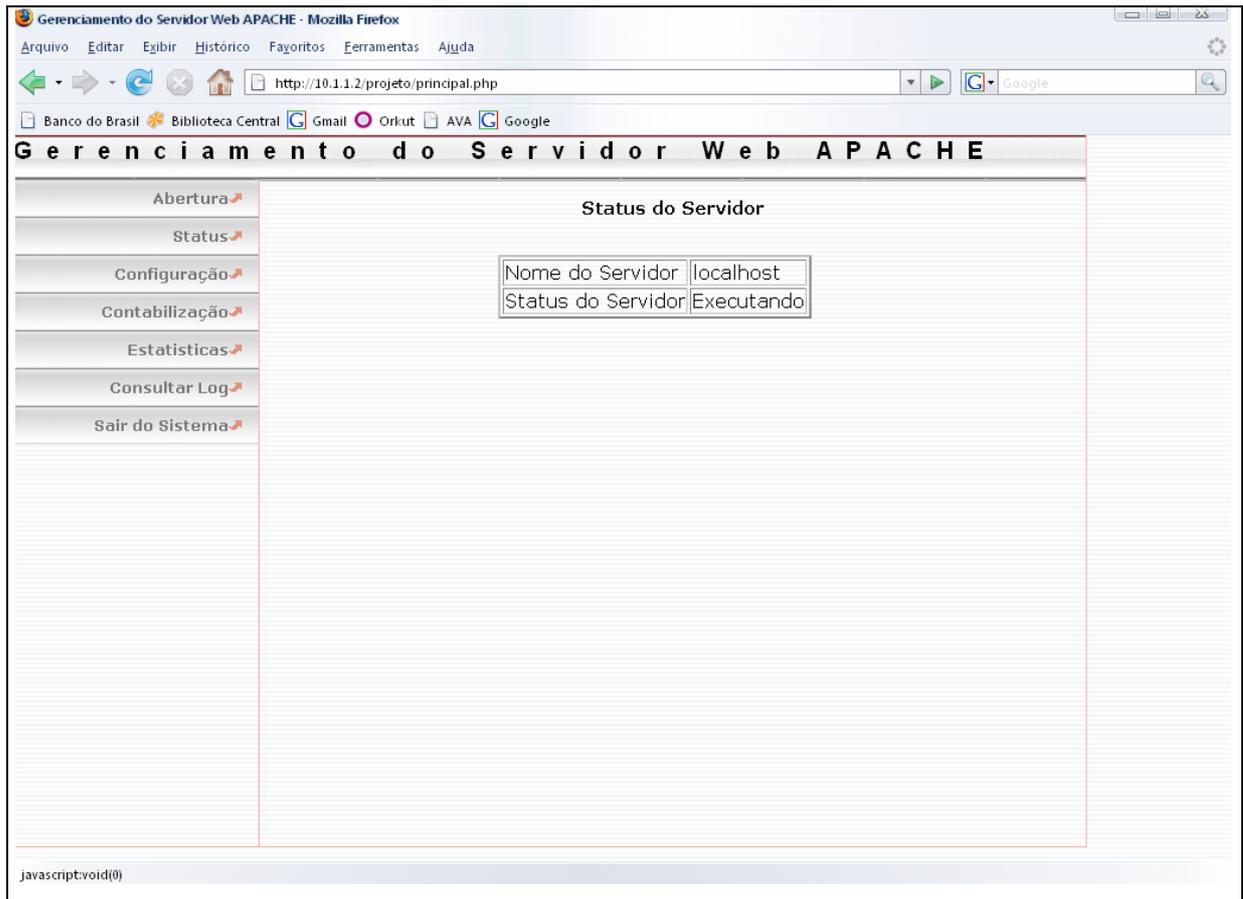


Figura 19 – *Status* do servidor

Na figura 19 observa-se a tela que mostra o *status* do servidor. Nesta tela são possíveis os seguintes *status*:

- a) Executando;
- b) Parado;
- c) Iniciando;
- d) Parando;
- e) Não respondendo.

No quadro 10 apresenta-se parte do código fonte responsável pela informação de *status* do servidor.

```

public static function consultaStatus($ip)
{
    try
    {
        //comando SNMP que consulta o status do servidor na MIB e armazena na variável $status
        $status = snmpget($ip, "ropublic", "APACHE2-MIB::serverStatus.0");
        if (strstr($status,"up"))
            return "Executando";
        elseif (strstr($status,"Down"))
            return "Parado";
        elseif (strstr($status,"Starting"))
            return "Iniciando";
        elseif (strstr($status,"Shutting"))
            return "Parando";
        elseif (strstr($status,"not"))
            return "Não Respondendo";
        else
            return "Erro ao tentar acessar Servidor";
    } catch (Exception $E)
    {
        return "Erro ao tentar acessar Servidor";
    }
}

```

Quadro 10 – Trecho do código fonte responsável pela consulta de *status*

The screenshot shows a web browser window titled "Gerenciamento do Servidor Web APACHE - Mozilla Firefox". The address bar shows "http://10.1.1.2/projeto/principal.php". The page content includes a navigation menu on the left with options like "Abertura", "Status", "Configuração", "Contabilização", "Estatísticas", "Consultar Log", and "Sair do Sistema". The main content area is titled "Informações de configuração do servidor Apache" and contains a table with the following data:

Nome do Servidor	localhost
IP do Servidor	127.0.0.1
Versão do Servidor	Apache/2.0.59 (Unix) PHP/5.2.3
Data da versão do Servidor	Jun 4 2007 19:39:24
Diretório de instalação do Servidor	/usr/local/apache2
Pid do processo do Servidor	/usr/local/apache2/logs/httpd.pid
Diretório para arquivos temporários	/tmp/ap2_snmp.tmp
Endereço ip e porta do Agente Http do Servidor	127.0.0.1:80
Numero de portas do Servidor	1

At the bottom of the page, the status "Concluído" is displayed.

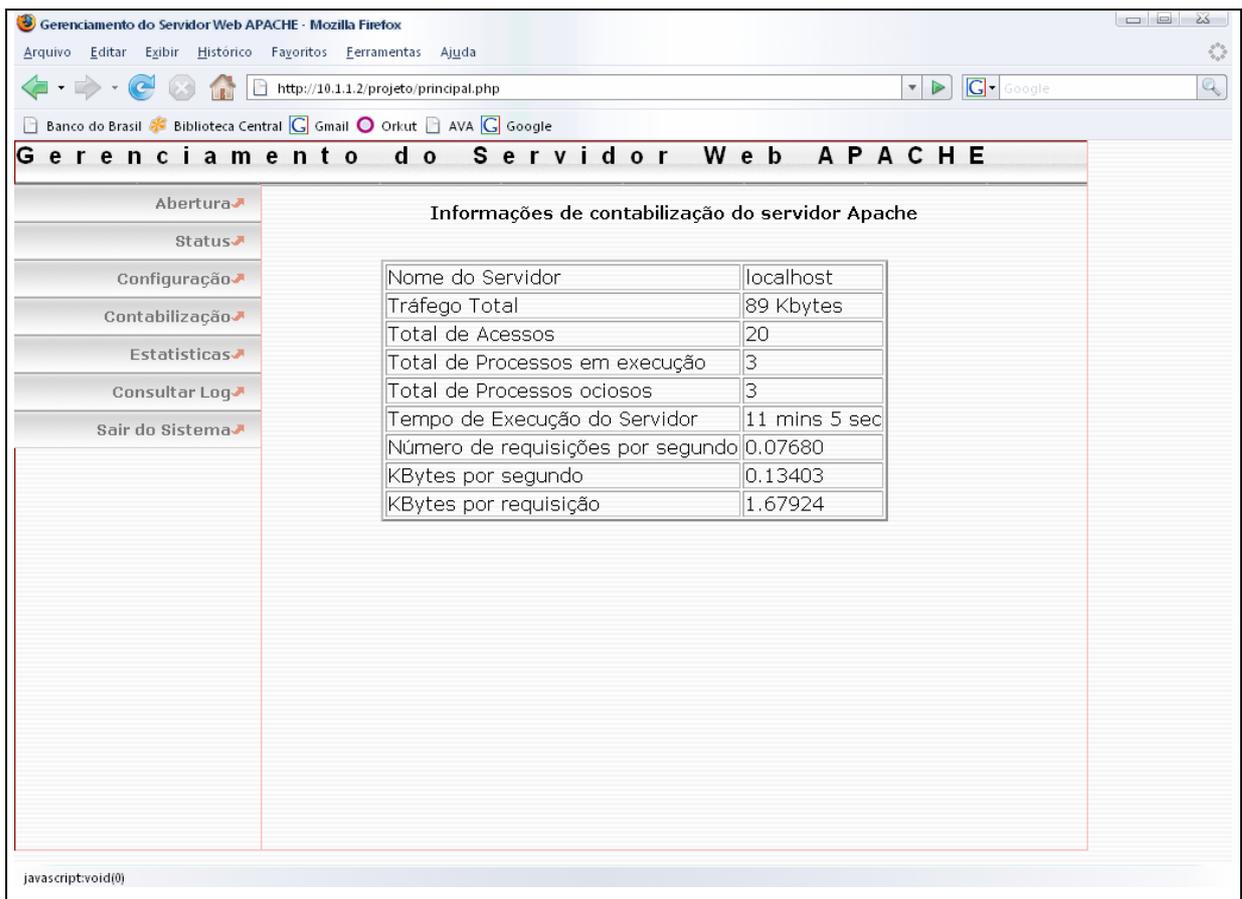
Figura 20 – Configurações do servidor

A figura 20 apresenta a tela que é carregada quando o usuário realiza a consulta de informações de configuração do servidor.

O quadro 11 apresenta uma parte do código fonte responsável pela geração das informações de configuração do servidor.

```
public static function listarConfiguracao($ip,$servidor)
{
    $ipServidor = snmpget($ip, "ropublic", "APACHE2-MIB::serverName.0");
    $versao = snmpget($ip, "ropublic", "APACHE2-MIB::serverVersion.0");
    $compilacao = snmpget($ip, "ropublic", "APACHE2-MIB::serverBuilt.0");
    $diretorio = snmpget($ip, "ropublic", "APACHE2-MIB::serverRoot.0");
    $pid = snmpget($ip,"ropublic", "APACHE2-MIB::serverPidfile.0");
    $tmpdir = snmpget($ip,"ropublic", "APACHE2-MIB::serverTmpDir.0");
    $agentHttp = snmpget($ip,"ropublic", "APACHE2-MIB::agentHttpAddress.0");
    $portas = snmpget($ip,"ropublic", "APACHE2-MIB::totalServerPorts.0");
}
}
```

Quadro 11 – Trecho do código fonte responsável pela consulta de configuração



The screenshot shows a web browser window titled "Gerenciamento do Servidor Web APACHE - Mozilla Firefox". The address bar shows "http://10.1.1.2/projeto/principal.php". The page content includes a navigation menu on the left with options like "Abertura", "Status", "Configuração", "Contabilização", "Estatísticas", "Consultar Log", and "Sair do Sistema". The main content area is titled "Informações de contabilização do servidor Apache" and contains a table with the following data:

Nome do Servidor	localhost
Tráfego Total	89 Kbytes
Total de Acessos	20
Total de Processos em execução	3
Total de Processos ociosos	3
Tempo de Execução do Servidor	11 mins 5 sec
Número de requisições por segundo	0.07680
kBytes por segundo	0.13403
kBytes por requisição	1.67924

Figura 21 – Informações de contabilização

Na figura 21 apresenta-se a tela que é carregada quando o usuário consulta informações de contabilização do servidor.

No quadro 12 é apresentado parte do código fonte que é responsável pela geração das informações de contabilização do servidor.

```

public static function listarContabilizacao($ip,$servidor)
{
    $trafegoTotal = snmpget($ip, "ropublic", "APACHE2-MIB::totalTraffic.0");
    $acessos = snmpget($ip, "ropublic", "APACHE2-MIB::totalAccess.0");
    $processosExec = snmpget($ip, "ropublic", "APACHE2-MIB:busyWorkers.0");
    $processosPar = snmpget($ip, "ropublic", "APACHE2-MIB:idleWorkers.0");
    $tempoExec = snmpget($ip, "ropublic", "APACHE2-MIB:serverUptime.0");
    $RequestPerSec = snmpget($ip, "ropublic", "APACHE2-MIB:serverRequestsPerSec.0");
    $KBytesPerSec = snmpget($ip, "ropublic", "APACHE2-MIB:serverKBytesPerSec.0");
    $KBytesPerRequest = snmpget($ip, "ropublic", "APACHE2-MIB:serverKBytesPerRequest.0");
    //gravar dados no banco para geração de gráfico
    $trafego = substr($trafegoTotal,8,strlen($trafegoTotal)-1);
    $banco = new BancoDados();
    $sql="insert into historico_dados (datahora,valor,tipo,ip_servidor) values
    (sysdate(),$trafego,'TRAFICO','$ip)";
    $banco->executarSql($sql);
}

```

Quadro 12 – Trecho do código fonte responsável pela consulta de contabilização

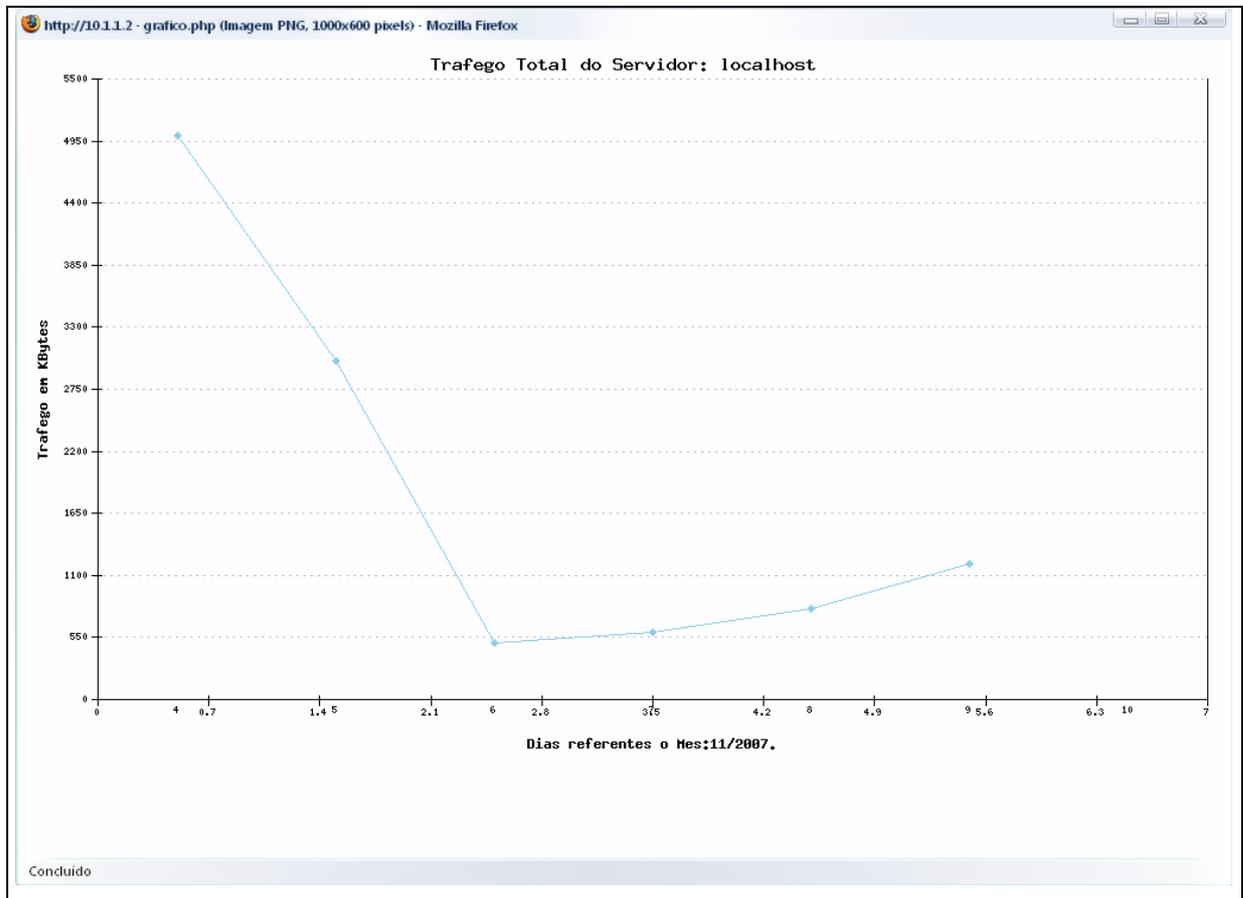


Figura 22 – Tráfego total

Na figura 22 observa-se um gráfico gerado pelo menu de estatísticas. A figura ilustra um gráfico de tráfego total em kbytes X dias do mês de novembro do ano de 2007. Outro gráfico que pode ser gerado nesta opção do menu é o gráfico de total de acessos ao Apache.

Usuário	Operação Realizada	Data e Hora da Operação
123	Consulta de Configuração.	2007-11-04 20:36:04
123	Consulta de informações de contabilização.	2007-11-04 20:36:22
Fabrício	Usuário Logado.	2007-11-04 20:38:51
Fabrício	Consulta de Status.	2007-11-04 20:39:36
Fabrício	Consulta de Configuração.	2007-11-04 20:39:53
Fabrício	Consulta de informações de contabilização.	2007-11-04 20:40:09
Fabrício	Usuário Logado.	2007-11-04 20:59:31
Fabrício	Consulta de Status.	2007-11-04 21:00:29
Fabrício	Consulta de Configuração.	2007-11-04 21:01:04
Fabrício	Consulta de informações de contabilização.	2007-11-04 21:01:33

Figura 23 – Log

Na figura 23 é apresentada a tela que é carregada quando o usuário opta por consultar o log do sistema. Essa tela apresenta as operações que cada usuário realizou como por exemplo consulta de *status*, informando qual usuário realizou, e a data e hora da realização.

### 3.4 RESULTADOS E DISCUSSÃO

Para conseguir-se que os objetivos propostos fossem alcançados foi preciso superar algumas dificuldades encontradas durante o desenvolvimento do projeto. Quando pensou-se em construir um protótipo de ferramenta de gerenciamento para o servidor *web* Apache não se sabia que o Apache não tinha suporte ao protocolo SNMP nativo, portanto não havia em sua compilação original nenhuma MIB que pudesse ser acessada via protocolo SNMP. Com a identificação deste problema começou-se então a busca por uma solução eficaz para que o Apache passasse a ter uma MIB gerenciável, porém com o tempo disponível para a construção da ferramenta ficava inviável uma tentativa de construir a MIB.

Pesquisando-se na internet sobre o assunto foi descoberto um módulo SNMP para o Apache, Hazewinkel (2004). A MIB desenvolvida era para ser configurada no servidor Apache que rodasse no ambiente Linux. A MIB não dava suporte a todas as informações que inicialmente pensava-se gerenciar, e em várias tentativas de configuração da mesma, junto ao Apache não obteve-se sucesso.

Tentou-se usar então um outro ambiente (sistema operacional BSD) para re-compilar o Apache juntamente com a MIB e novamente não foi possível fazer essa integração. Neste ponto do projeto decidiu-se então abandonar este módulo SNMP para o Apache e buscar uma nova solução para o problema.

Um novo módulo SNMP foi encontrado, Pizzini (2006). Este módulo possui uma MIB implementada na qual verificou-se a possibilidade de gerenciamento das informações inicialmente propostas. Novamente foi preciso fazer a re-compilação do servidor Apache para ambiente Linux, e com isso alguns problemas de configuração foram encontrados acarretando em um atraso considerável no andamento do projeto.

Após várias tentativas sem sucesso para fazer o módulo ser carregado e funcionar corretamente com o Apache, a única solução foi a de tentar um contato com o desenvolvedor para que o mesmo pudesse passar algumas instruções de como fazer a configuração do módulo. Através de um e-mail recebeu-se um primeiro *feedback* do desenvolvedor e a partir daí vários contatos foram feitos na tentativa de encontrar onde estava o problema. Foram utilizadas nesta etapa do projeto ferramentas de comunicação através de mensagens como o *Google Talk* e o MSN da Microsoft.

Várias verificações na configuração foram feitas, e o desenvolvedor mandou inclusive alguns arquivos para que se fizesse execução e se analisasse o resultado para descobrir em que ponto estava o problema. Após muitas tentativas descobriu-se que o problema que não permitia que o módulo rodasse estava na parametrização do arquivo `httpd.conf`, onde modificou-se o valor do parâmetro `snmp_community` de `public` para `ropublic` e criou-se um `rwpublic` também, e resolvido isto o módulo passou a rodar perfeitamente quando o Apache era carregado.

Neste ponto todos os testes eram feitos via console do Linux, e restava então pouco mais de um mês para a conclusão da construção do protótipo. Felizmente com o problema da incorporação da MIB junto ao Apache resolvido foi possível a construção da ferramenta através dos *scripts* PHP para fazer as consultas na MIB.

Foram realizados alguns testes com a ferramenta para verificação de possíveis falhas, priorizando-se os testes referentes à segurança e tratamento às exceções geradas por erros no

processamento. No quadro 13 apresentam-se os principais testes realizados e os resultados obtidos.

<b>Descrição do teste</b>	<b>Resultado obtido</b>
Tentativa de acesso à ferramenta com nome de usuário e/ou senha inválido(s).	A ferramenta direciona para uma página que mostra uma mensagem de erro de inserção de dados.
Tentativa de acesso direto à página principal da ferramenta, sem precisar passar pelo login do sistema.	Sistema não carrega a página solicitada, e carrega novamente a página onde deve ser feito o login, que no caso é a página <code>index.php</code> .
Cadastramento de um servidor inexistente para tentar fazer o gerenciamento.	Gera um erro de exceção que é tratado através de uma mensagem “Erro ao tentar acessar Servidor”
Teste de login com informações de usuário corretas.	Acessou ferramenta normalmente.
Teste de acesso aos servidores cadastrados.	Informações solicitadas foram apresentadas normalmente.

Quadro 13 – Testes com a ferramenta

A ferramenta mostrou-se eficiente para atender as consultas ao Apache para qual foi desenvolvida.

### 3.4.1 Análise de Trabalhos Correlatos

A ferramenta de Costa (2003) tem como características semelhantes ao protótipo o uso do protocolo SNMP para gerenciamento e a construção de *scripts* com a linguagem de programação PHP, mas a principal diferença é que seu gerenciamento não é para um servidor *web* e sim para administrar a rede local.

Comparando a ferramenta com o trabalho de Karing (2002), tem-se como características semelhantes, a utilização do protocolo SNMP, porém a ferramenta aqui desenvolvida faz o monitoramento de um servidor *web*, que neste caso é o servidor Apache permitindo inclusive que tal monitoramento seja feito através da internet, enquanto que a ferramenta de Karing (2002) faz o monitoramento de desempenho de uma rede de

computadores em uma rede local e foi desenvolvida utilizando-se a linguagem de programação Java.

## 4 CONCLUSÕES

O objetivo deste trabalho foi desenvolver um protótipo de ferramenta para monitoramento do servidor *web* Apache sendo possível fazer isto através da internet ou mesmo de uma rede local. Para tal desenvolvimento teve-se como objetivo o uso da linguagem de programação PHP e do protocolo de gerência SNMP.

Foi desenvolvido um protótipo onde o usuário pode monitorar um ou mais servidores Apache realizando consultas em uma MIB que deve ser previamente configurada no servidor. A ferramenta permite ao usuário acessar o servidor ao qual deseja gerenciar através de um navegador internet, bastando para isso possuir o endereço ip do servidor que deseja acessar. O usuário tem acesso a informações como *status* do servidor, diretório onde o mesmo encontra-se, total de portas do servidor, tráfego total, total de acessos entre outras informações, tornando-se assim um protótipo inovador, pois não se tem conhecimento de nenhuma ferramenta que faça esse tipo de gerenciamento disponível no mercado.

Os principais objetivos propostos para o protótipo foram alcançados e com isso tem-se uma ferramenta para monitoramento do servidor Apache que é útil para um administrador de redes por exemplo.

O trabalho traz benefícios à área científica por servir como base para desenvolvimento de novos projetos nessa linha como a implementação de uma MIB para outros servidores *web*.

### 4.1 EXTENSÕES

São sugestões para extensão desse trabalho:

- a) implementação para cadastramento dos usuários diretamente na ferramenta;
- b) implementação para cadastramento dos servidores diretamente na ferramenta;
- c) adaptação na MIB que contém as informações de gerência para que seja possível inicializar e parar o servidor (alteração de valor no objeto gerenciado);
- d) recompilação do servidor Apache para ambiente Microsoft Windows com suporte a MIB de gerência;
- e) implementar uma MIB similar para outro servidor *web*, como por exemplo IIS da Microsoft;

- f) desenvolver um mecanismo para que a ferramenta faça consultas em determinados intervalos de tempo na MIB independentemente da ação do usuário.

## REFERÊNCIAS BIBLIOGRÁFICAS

ALECRIM, Emerson. **Conhecendo o servidor apache**. [S.l.], [2006]. Disponível em: <<http://www.infowester.com/servapach.php>>. Acesso em: 05 ago. 2007.

CÂNDIDO, Carlos Henrique. **BrModelo 2.0**. [S.l.], [2005]. Disponível em: <<http://www.sis4.com/brModelo/>>. Acesso em: 15 set. 2007.

COSTA, Italo Marcelo de Oliveira. **Gerenciamento de redes utilizando o protocolo SNMP**. [S.l.], [2003?]. Disponível em: <<http://phpbrasil.com/articles/article.php/id/411>>. Acesso em: 15 mar. 2004.

DIAS, Beethoven Zanella; JÚNIOR, Nilson Alves. **Protocolo de gerenciamento SNMP**. [S.l.], [2002]. Disponível em: <[http://mesonpi.cat.cbpf.br/naj/snmp\\_color.pdf](http://mesonpi.cat.cbpf.br/naj/snmp_color.pdf)>. Acesso em: 17 ago. 2007.

FREITAS, João Carlos da Silva. **Simple network management protocol**. Rio de Janeiro, [2000]. Disponível em: <[http://www.gta.ufrj.br/grad/00\\_1/joao/index.html](http://www.gta.ufrj.br/grad/00_1/joao/index.html)>. Acesso em: 10 ago. 2007.

FURLAN, Jose Davi. **Modelagem de objetos através da UML-The Unified Modeling Language**. São Paulo: Makron Books, 1998. 329 p.

GRANVILLE, Lisandro Zambenedetti. **Módulo 5 – Management Information Base (MIB)**. [S.l.], [2006?]. Disponível em: <[http://www.inf.ufrgs.br/granville/Gerencia/Programa/Mod5/Mod5\\_4.pdf](http://www.inf.ufrgs.br/granville/Gerencia/Programa/Mod5/Mod5_4.pdf)>. Acesso em: 08 ago. 2007.

HAZEWINKEL, Harrie. **SNMP module for Apache 1.3.x**. [S.l.], [2004]. Disponível em: <<http://www.mod-snmp.com/>>. Acesso em: 03 mar. 2006.

KARING, Anderson. **Protótipo de um sistema de monitoramento de desempenho de redes de computadores baseado no protocolo SNMPV3**. 2002. 43 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

PIZZINI, Esteban. **SNMP para Apache Web Server**. [S.l.], [2006]. Disponível em: <<http://mod-apache-snmp.sourceforge.net/>>. Acesso em: 15 mar. 2007.

SANTOS, Carlos Eduardo Meyer dos. **Plataformas de gerenciamento**. [S.l.], [2002]. Disponível em: <<http://www.gta.ufrj.br/~rezende/cursos/eel879/trabalhos/gerenciamento/>>. Acesso em: 20 jun. 2007.

TANENBAUM, Andrew S. **Redes de computadores**. 3. ed. Rio de Janeiro: Campus, 1997.

## ANEXO A – Apache2-MIB

No quadro 14 observa-se a MIB para o servidor Apache, que foi utilizada no projeto.

```

APACHE2-MIB DEFINITIONS ::= BEGIN

IMPORTS
    enterprises                               FROM SNMPv2-SMI
    DisplayString                             FROM SNMPv2-TC
    OBJECT-TYPE, Integer32, Counter32, Gauge32
    MODULE-IDENTITY                           FROM SNMPv2-SMI
    MODULE-COMPLIANCE, OBJECT-GROUP          FROM SNMPv2-CONF;

apache2-mib MODULE-IDENTITY
    LAST-UPDATED "200402180000Z"
    ORGANIZATION "Esteban Pizzini"
    CONTACT-INFO
        " Esteban Pizzini
         ep@fibertel.com.ar"
    DESCRIPTION
        "Apache2 Mib"
    REVISION    "200402210000Z"
    DESCRIPTION
        "First draft"
    ::= { enterprises 19786 1 1 }

apache2MIBInformation OBJECT IDENTIFIER ::= { apache2-mib 1 }
apache2MIBStatus      OBJECT IDENTIFIER ::= { apache2-mib 2 }
apache2MIBNotifications OBJECT IDENTIFIER ::= { apache2-mib 4 }
apache2MIBHttpErrors  OBJECT IDENTIFIER ::= { apache2-mib 5 }

serverName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Server Name"
    ::= { apache2MIBInformation 1 }

serverVersion OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Apache Server Version"
    ::= { apache2MIBInformation 2 }

serverBuilt OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Apache build datetime"
    ::= { apache2MIBInformation 3 }

serverRestart OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write

```

```

        STATUS          current
        DESCRIPTION
            "Last restart"
        ::= { apache2MIBInformation 4 }

serverRoot OBJECT-TYPE
    SYNTAX          DisplayString
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Apache's Root Path"
    ::= { apache2MIBInformation 5}

serverPidfile OBJECT-TYPE
    SYNTAX          DisplayString
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Apache's PIDFILE"
    ::= { apache2MIBInformation 6}

serverTmpDir OBJECT-TYPE
    SYNTAX          DisplayString
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Temporary Directory for shared files (Net-SNMP <-->
Apache)"
    ::= { apache2MIBInformation 7}

agentHttpAddress OBJECT-TYPE
    SYNTAX          DisplayString
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Port and Address that use the agent to get HTTP access to
Apache"
    ::= { apache2MIBInformation 8}

totalServerPorts OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Total ports listened by Apache"
    ::= { apache2MIBInformation 9}

--
-- Tabla de puertos escuchados (implementada en serverListenPorts.c)
--

serverListenPorts OBJECT-TYPE
    SYNTAX          SEQUENCE OF ServerPort
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "List of ports listened by Apache"
    ::= { apache2MIBInformation 10}

serverPort OBJECT-TYPE

```

```

        SYNTAX          ServerPort
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION
            "List of ports listened by Apache"
        INDEX { serverPortIndex }
        ::= { serverListenPorts 1 }

ServerPort ::= SEQUENCE {
    serverPortIndex      Integer32,
    serverPortNumber     Integer32
}

serverPortIndex OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Port Index"
    ::= { serverPort 1 }

serverPortNumber OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "Port Number"
    ::= { serverPort 2 }

-- fin tabla

totalTraffic OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Total Traffic"
    ::= { apache2MIBStatus 1 }

totalAccess OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Total HTTP Accesses"
    ::= { apache2MIBStatus 2 }

busyWorkers OBJECT-TYPE
    SYNTAX          Gauge32
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "total number of Apache's busy processes"
    ::= { apache2MIBStatus 3 }

idleWorkers OBJECT-TYPE
    SYNTAX          Gauge32
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION

```

```

        "total number of Apache's idle processes"
        ::= { apache2MIBStatus 4 }

serverStatus OBJECT-TYPE
    SYNTAX      Integer32 { down(0), up(1), notResponding(2),
starting(3), stopping(4) }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Status
         0 - Down
         1 - Up
         2 - Starting
         3 - Shutting down
         4 - not responding"
        ::= { apache2MIBStatus 5 }

serverUptime OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Uptime"
        ::= { apache2MIBStatus 6 }

agentHttpAccess OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "HTTP accesses generated by SNMP agent"
        ::= { apache2MIBStatus 7 }

serverRequestsPerSec OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Requests per second"
        ::= { apache2MIBStatus 8 }

serverKBytesPerSec OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "KBytes per second"
        ::= { apache2MIBStatus 9 }

serverKBytesPerRequest OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "KBytes per request"
        ::= { apache2MIBStatus 10 }

httpError400 OBJECT-TYPE
    SYNTAX      Counter32

```

```

MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "HTTP_BAD_REQUEST"
 ::= { apache2MIBHttpErrors 400 }

httpError403 OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "HTTP_FORBIDDEN"
 ::= { apache2MIBHttpErrors 403 }

httpError404 OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "HTTP_NOT_FOUND"
 ::= { apache2MIBHttpErrors 404 }

httpError405 OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "HTTP_METHOD_NOT_ALLOWED"
 ::= { apache2MIBHttpErrors 405 }

httpError500 OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "HTTP_INTERNAL_SERVER_ERROR"
 ::= { apache2MIBHttpErrors 500 }

httpError501 OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "HTTP_NOT_IMPLEMENTED"
 ::= { apache2MIBHttpErrors 501 }

httpError505 OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "HTTP_VERSION_NOT_SUPPORTED"
 ::= { apache2MIBHttpErrors 505 }

serverStatusNotification NOTIFICATION-TYPE
OBJECTS { serverName, serverStatus }
MAX-ACCESS accessible-for-notify
STATUS current
DESCRIPTION
    "Server status changes"
 ::= { apache2MIBNotifications 1 }

```

```
serverRestartNotification NOTIFICATION-TYPE
  OBJECTS { serverRestart ,serverStatus}
  MAX-ACCESS accessible-for-notify
  STATUS current
  DESCRIPTION
    "Server Restart"
  ::= { apache2MIBNotifications 2 }

serverStartNotification NOTIFICATION-TYPE
  OBJECTS { serverName, serverStatus }
  MAX-ACCESS accessible-for-notify
  STATUS current
  DESCRIPTION
    "Server Starting"
  ::= { apache2MIBNotifications 3 }

serverStopNotification NOTIFICATION-TYPE
  OBJECTS { serverName, serverStatus }
  MAX-ACCESS accessible-for-notify
  STATUS current
  DESCRIPTION
    "Server Shutting down"
  ::= { apache2MIBNotifications 4 }

END
```

Fonte: Pizzini (2006).

Quadro 14 – MIB para servidor Apache