



**GERÊNCIA DE REDES DE COMPUTADORES:  
Serviços, Espaço em Disco e Falhas nos Agentes  
via SNMP**

Aluno: Daniel Miranda Barroso – RA: 2023881/6  
Orientador: Msc. Fabiano Mariath D`Oliveira

Brasília – DF, Julho de 2008

GERÊNCIA DE REDES DE COMPUTADORES:  
Serviços, Espaço em Disco e Falhas nos Agentes via SNMP

por

Daniel Miranda Barroso

Trabalho apresentado à  
Banca examinadora do  
curso de Engenharia da  
Computação da FATECS  
– Faculdade de  
Tecnologia e Ciências  
 Sociais Aplicadas –  
Centro Universitário de  
Brasília como requisito  
parcial para obtenção do  
título de Engenheiro da  
Computação

Brasília – DF, Julho de 2008

**Banca Examinadora**

---

Prof. Msc. Fabiano Mariath D`Oliveira  
Orientador

---

Prof. Msc. XXXXXXXXXXXXX  
Examinador

---

Prof. Msc. XXXXXXXXXXXXX  
Examinador

---

Prof. Msc. XXXXXXXXXXXXX  
Examinador

# Agradecimentos

---

A Deus, por essa vida maravilhosa.

Aos meus pais, avós e irmã por tanta dedicação e carinho em todos os momentos da minha vida.

A minha querida namorada, pelo apoio incondicional.

Ao esporte, por me proporcionar a tranquilidade necessária no dia a dia.

Ao meu orientador Professor MSc. Fabiano Mariath D' Oliveira que me guiou durante o desenvolvimento desse projeto.

Ao MSc. Francisco Javier Obaldia, pelo incentivo durante o projeto final.

Ao coordenador, MSc. Abiezer Amarília, pela dedicação perante o curso de Engenharia da Computação.

A todos que, de forma direta ou indireta, contribuíram para a realização deste trabalho.

# SUMÁRIO

---

|   |             |
|---|-------------|
| <b>LISTA DE FIGURAS</b> .....                                 | <b>VI</b>   |
| <b>LISTA DE TABELAS</b> .....                                 | <b>VII</b>  |
| <b>ÍNDICE DE SIGLAS E ABREVIATURAS</b> .....                  | <b>VIII</b> |
| <b>RESUMO</b> .....   | <b>IX</b>   |
| <b>CAPÍTULO 1 – INTRODUÇÃO</b> .....                          | <b>10</b>   |
| 1.1 MOTIVAÇÃO .....   | 10          |
| 1.2 OBJETIVOS.....  | 11          |
| 1.3 METODOLOGIA .....   | 12          |
| 1.4 ESTRUTURA DA MONOGRAFIA .....                             | 12          |
| <b>CAPÍTULO 2 – GERENCIAMENTO DE REDES</b> .....              | <b>14</b>   |
| 2.1 COMPONENTES DE GERENCIAMENTO .....                        | 14          |
| 2.2 SNMP .....  | 15          |
| 2.2.2 COMUNIDADES .....                                       | 20          |
| 2.2.3 MIB .....   | 22          |
| 2.3 ÁREAS FUNCIONAIS DE GERENCIAMENTO.....                    | 25          |
| <b>CAPÍTULO 3 – FERRAMENTAS DE GERÊNCIA</b> .....             | <b>27</b>   |
| 3.1 MRTG .....  | 27          |
| 3.2 ITM.....  | 29          |
| 3.3 HP OPENVIEW .....   | 32          |
| 3.4 NAGIOS.....   | 34          |
| <b>CAPÍTULO 4 – DESENVOLVIMENTO DO PROJETO</b> .....          | <b>37</b>   |
| 4.1 REQUISITOS.....   | 38          |
| 4.2 FLUXO DE FUNCIONAMENTO .....                              | 38          |
| 4.2.1 <i>Diagrama de Caso de Uso</i> .....                    | 39          |
| 4.3 FERRAMENTAS SNMP COLETA, SNMP CLIENTE E SNMP RESOLVE..... | 40          |
| 4.3.1 <i>Aplicativo SNMP Coleta</i> .....                     | 40          |
| 4.3.2 <i>Aplicativo SNMP Cliente</i> .....                    | 45          |
| 4.3.3 <i>Aplicativo SNMP Resolve</i> .....                    | 48          |
| 4.4 DEMAIS FERRAMENTAS UTILIZADAS .....                       | 50          |
| 4.4.1 <i>HmailServer</i> .....                                | 50          |
| 4.4.2 <i>Net-Snmp</i> .....                                   | 50          |
| 4.4.3 <i>Bmail</i> .....                                      | 50          |
| 4.4.4 <i>Banco de Dados MS Access 2003</i> .....              | 50          |
| 4.4.5 <i>Microsoft Data Access Components</i> .....           | 52          |
| 4.5 PROBLEMAS E SOLUÇÕES .....                                | 52          |
| <b>CAPÍTULO 5 – IMPLEMENTAÇÃO, TESTES E RESULTADOS</b> .....  | <b>55</b>   |
| 5.1 IMPLEMENTAÇÃO E CONFIGURAÇÃO DO AMBIENTE .....            | 55          |
| 5.2 FASE DE TESTES.....                                       | 61          |
| 5.3 RESULTADOS .....  | 63          |

|   |           |
|---|-----------|
| <b>CAPÍTULO 6 – CONCLUSÕES E TRABALHOS FUTUROS.....</b> | <b>67</b> |
| 6.1 TRABALHOS FUTUROS.....                              | 68        |
| <b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>                  | <b>69</b> |
| <b>APÊNDICE A – CÓDIGOS DE PROGRAMAÇÃO.....</b>         | <b>72</b> |

# LISTA DE FIGURAS

---

|   |    |
|---|----|
| FIGURA 1 – COMPONENTES BÁSICOS PARA GERENCIAR UMA REDE [CISCO, 2006].   | 15 |
| FIGURA 2 – MODELO DE COMUNICAÇÃO TCP/IP E SNMP [MAURO E SCHMIDT, 2005]. | 18 |
| FIGURA 3 – FORMATO DE UMA MENSAGEM SNMP [CISCO, 2006].                  | 20 |
| FIGURA 4 – CONFIGURANDO A COMUNIDADE DE UM AGENTE WINDOWS.              | 21 |
| FIGURA 5 – ESTRUTURA DA MIB [CISCO, 2006].                              | 24 |
| FIGURA 6 – GRÁFICO DO MRTG [MRTG, 2008].                                | 28 |
| FIGURA 7 – FLUXO DE DADOS DO IBM TIVOLI MONITORING [IBM, 2008].         | 30 |
| FIGURA 8 – TELA DO IBM TIVOLI ENTERPRISE PORTAL [IBM, 2008].            | 31 |
| FIGURA 9 – EXEMPLO DO COMANDO CHECKPING [BARTH, 2005].                  | 34 |
| FIGURA 10 – CONSOLE DE EVENTOS NAGIOS [NAGIOS, 2008].                   | 35 |
| FIGURA 11 – ELEMENTOS GERENCIADOS, NAGVIS MAP [JOSEPHSEN, 2007].        | 36 |
| FIGURA 12 – TELA DO HP OPENVIEW [TAUROCO, 1999].                        | 33 |
| FIGURA 13 – FLUXO DE FUNCIONAMENTO DA SOLUÇÃO.                          | 38 |
| FIGURA 14 – DIAGRAMA DE CASO DE USO DO APLICATIVO SNMP COLETA.          | 40 |
| FIGURA 15 – INTERFACE DO PROGRAMA SNMP COLETA.                          | 42 |
| FIGURA 16 – CRIAÇÃO DA CONEXÃO COM O BANCO DE DADOS.                    | 44 |
| FIGURA 17 – VERIFICAÇÃO DO STATUS DOS SERVIÇOS.                         | 45 |
| FIGURA 18 – SUB-ROTINA RESPONSÁVEL PELA GRAVAÇÃO NO LOG.                | 45 |
| FIGURA 19 – INTERFACE DO PROGRAMA SNMP CLIENTE.                         | 46 |
| FIGURA 20 – CONFIGURAÇÃO DO ARQUIVO .UDL.                               | 47 |
| FIGURA 21 – CONSULTA E DISPONIBILIZA OS DADOS.                          | 48 |
| FIGURA 22 – INTERFACE DO PROGRAMA SNMP RESOLVE.                         | 49 |
| FIGURA 23 – TELA DE CONFIGURAÇÃO DE RECURSOS DO SERVIDOR CLIENTE.       | 57 |
| FIGURA 24 – CONFIGURAÇÃO DE REDE DO SERVIDOR GERENTE.                   | 58 |
| FIGURA 25 – VIRTUAL PC DEPOIS DE INICIADAS AS MÁQUINAS VIRTUAIS.        | 59 |
| FIGURA 26 – INTERFACE DE GERENCIAMENTO DO APLICATIVO DE E-MAIL.         | 59 |
| FIGURA 27 – CONFIGURAÇÃO DO SERVIÇO “SNMP SERVICE”.                     | 60 |
| FIGURA 28 – TEMPO DE EXECUÇÃO, EM SEGUNDOS, DE CADA CENÁRIO.            | 65 |
| FIGURA 29 – E-MAILS RECEBIDOS CONFORME ESPERADO.                        | 66 |

## LISTA DE TABELAS

---

|  |    |
|--|----|
| TABELA 1 - EXEMPLO DE CATEGORIAS DA MIB [COMER, 2006]. ..... | 23 |
| TABELA 2 - EXEMPLO DE VARIÁVEIS DA MIB [COMER, 2006]. .....  | 24 |
| TABELA 3 - COMPARAÇÃO NAGIOS E ÁREAS DE GERENCIAMENTO. ....  | 35 |
| TABELA 4 - TABELA SNMP. ....                                 | 51 |
| TABELA 5 - CENÁRIOS SIMULADOS. ....                          | 62 |

## ÍNDICE DE SIGLAS E ABREVIATURAS

---

|        |   |
|--------|---|
| DAO    | Data Access Object                              |
| GPL    | General Public License                          |
| HTML   | Hipertext Markup Language                       |
| ITM    | IBM Tivoli Monitoring                           |
| MDAC   | Microsoft Data Access Components                |
| MIB    | Management Information Base                     |
| MRTG   | Multi Router Traffic Grapher                    |
| MSDN   | Microsoft Developer Network                     |
| NMS    | Network Management System                       |
| OSI    | Open Systems Interconnection                    |
| RDO    | Remote Data Object                              |
| RDBMS  | Relational Database Management System           |
| SGBD   | Sistema Gerenciador de Banco de Dados           |
| SNMP   | Simple Management Network Protocol              |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| TEMS   | Tivoli Enterprise Monitoring Server             |
| TEP    | Tivoli Enterprise Portal                        |
| TEPS   | Tivoli Enterprise Portal Server                 |
| VB     | Visual Basic                                    |



# RESUMO

---

Este trabalho apresenta o estudo de uma solução de gerência que detecta pouco espaço em disco, indisponibilidade de serviços importantes e falha no aplicativo de gerenciamento.

Esta solução de gerência foi implementada de maneira confiável, onde os alertas de pouco espaço em disco são gerados corretamente, onde indisponibilidades de serviços são detectadas e ainda quando ocorre falha durante a sua execução é gerado alertas comunicando as equipes e o gerente, responsáveis pelo servidor, por email. Também foi criada uma interface cliente para consultar os dados das informações coletadas.

As duas ferramentas foram desenvolvidas utilizando a ferramenta *Microsoft Visual Basic 6.0 SP6* e o banco de dados utilizado para armazenamento das informações foi o *Microsoft Access 2003*. O protocolo de gerenciamento escolhido foi o *SNMP - Simple Network Management Protocol*.

Palavras-Chave: gerência de rede; espaço em disco; serviços; SNMP; VisualBasic; banco de dados Access e monitoração.

# CAPÍTULO 1 – INTRODUÇÃO

---

## 1.1 MOTIVAÇÃO

A fusão dos computadores e das comunicações teve uma profunda influência na forma como os sistemas computacionais eram organizados. O conceito de "centro de computação" como uma sala com um grande computador ao qual os usuários levam seu trabalho para processamento agora está completamente obsoleto. O velho modelo de um único computador atendendo a todas as necessidades computacionais da organização foi substituído pelas chamadas redes de computadores, nas quais os trabalhos são realizados por um grande número de computadores separados, mas interconectados. [TANENBAUM, 2003]

Com o passar dos anos as redes de computadores estão se tornando cada vez mais críticas e complexas. Logo, observamos que estas redes são fundamentais para uma organização e simplesmente não podem falhar. Em algumas organizações, o nível de falhas e degradação de desempenho considerado aceitável está diminuindo demasiadamente, chegando próximo limite da probabilidade. Dependendo da criticidade da aplicação, o limite mínimo de disponibilidade aceitável gira em torno de 99,9%.

Portanto, com essa crescente necessidade de gerenciamento e com o aumento da complexidade das redes de computadores, a gerência de redes está se tornando cada vez mais importante nas empresas ao redor do globo.

Entretanto, problemas como: pouco espaço em disco ou indisponibilidade de serviços importantes ainda vem gerando muita preocupação em diversas empresas ao redor do mundo.

Existem ferramentas no mercado que se propõem a auxiliar os administradores de rede, é o caso do IBM Tivoli Monitoring (ITM), HP

OpenView, MRTG e Nagios. O ITM e o HP OpenView são soluções proprietárias e seus custos são elevados e variáveis dependendo das necessidades da Organização. O MRTG é basicamente uma solução para gerar gráficos. Já o Nagios, é uma ferramenta livre, de código aberto que atende as necessidades dos administradores que trabalham com sistema LINUX e suas variantes. Entretanto, existem poucas ferramentas no mercado com as características apresentadas no Nagios para a plataforma Windows.

## 1.2 OBJETIVOS

O projeto tem como finalidade o desenvolvimento, implementação e configuração de uma solução de gerência que aborde os aspectos da gerência de falhas, detectando pouco espaço em disco, indisponibilidade de serviços importantes e falhas nos agentes de gerenciamento. Será implementada uma solução de gerência onde os alertas gerarão um comunicado às equipes responsáveis através do envio de e-mail.

Os objetivos específicos do trabalho são:

- a) Obter informações do disco rígido, dentre elas:
  - a. Descrição
  - b. Tamanho de blocos utilizado
  - c. Espaço disponível
  - d. Espaço Utilizado
  
- b) Verificar se serviços críticos estão ativos, como:
  - a. Task Scheduler
  - b. Server
  - c. DNS Client
  - d. Windows Time
  
- c) Enviar comunicado à equipe de administração responsável ao detectar uma indisponibilidade de serviço ou pouco espaço em disco.

- d) Visualizar o status dos serviços e a utilização de disco rígido em um computador remoto através de uma interface cliente.

Caso, a equipe responsável não atue no servidor em um tempo pré-determinado ou de maneira correta e a ocorrência permaneça no ambiente, será encaminhado um email consolidado, informando o gerente do site que aquele problema ainda persiste e as equipes responsáveis não tomaram as medidas esperadas. Desta forma, ficará assegurado que os servidores realmente estarão sendo gerenciados conforme esperado.

### 1.3 METODOLOGIA

O método de pesquisa utilizado quando aos fins foi a pesquisa aplicada. Segundo Vergara (1997), esse tipo de pesquisa é motivada pela necessidade de resolver problemas concretos, mais imediatos, ou não. Tem, portanto, finalidade prática, ao contrário da pesquisa pura, motivada basicamente pela curiosidade intelectual do pesquisador e situada, sobretudo, no nível da especulação. Exemplo de pesquisa aplicada: proposta de mecanismos que diminuam a infecção hospitalar.

Já o método de pesquisa utilizado em relação aos meios de investigação foi a pesquisa de laboratório. Essa pesquisa é experiência realizada em local circunscrito, já que o campo seria praticamente impossível realizá-la.

### 1.4 ESTRUTURA DA MONOGRAFIA

Este projeto constitui-se de seis capítulos, incluindo a introdução. Segue uma breve descrição:

- Capítulo 2 – Gerenciamento de Redes – aborda os principais conceitos envolvidos neste projeto final como: SNMP, OID, MIB, Gerenciamento de Falhas, dentre outros.
- Capítulo 3 – Ferramentas de Gerência e Desenvolvimento – cita as principais ferramentas de gerenciamento de redes utilizadas no mercado de tecnologia e a ferramenta de desenvolvimento Visual Basic.
- Capítulo 4 – *SNMP Coleta* – desenvolvimento propriamente dito do projeto, contemplando descrições detalhadas, sustentações técnicas, diagramas, recursos aplicados, detalhamento de hardware e software utilizados.
- Capítulo 5 – Implementação, testes e resultados – detalha toda a fase de implementação do trabalho, os testes realizados e os resultados obtidos, incluindo as dificuldades encontradas ao longo da implementação do projeto.
- Capítulo 6 – Conclusões e Trabalhos Futuros – descreve as principais conclusões obtidas e sugere trabalhos futuros.

## CAPÍTULO 2 – GERENCIAMENTO DE REDES

---

As redes de computadores promovem facilidades aos usuários de determinados sistemas, isto estimula o seu crescimento e traz consigo também uma demanda de manutenção. Considerando-se uma rede dentro de um ambiente em que a mesma seja distribuída em várias salas ou em prédios distintos, a manutenção torna-se complexa, consumindo tempo e recursos [BRISA, 2001].

A gerência de redes trabalha em nível de aplicativo, se utilizando de protocolos de transporte como o SNMP (Simple Network Management Protocol), para administrar e controlar os componentes de uma rede, ou seja, quando um gerente precisa interagir com um dispositivo específico, o software de gerência segue o modelo cliente-servidor convencional: um programa aplicativo no computador do gerente atua como cliente e um programa no dispositivo de rede atua como servidor [COMER, 2006].

### 2.1 COMPONENTES DE GERENCIAMENTO

Para se gerenciar uma rede de computadores, alguns componentes são necessários para que haja uma comunicação entre os computadores conectados entre si. Estes componentes são os gerentes (NMS – *Network Management Systems*), agentes e os objetos gerenciados [BRISA, 2001]:

- Gerente – tem por finalidade executar aplicações que monitoram e controlam os objetos gerenciados. Além disso, estes são responsáveis pelos recursos de processamento e memória necessários para o gerenciamento de rede, podendo existir mais de um gerente por rede.

- o Agente – aplicativo que é executado no objeto gerenciado e detêm o conhecimento das informações deste objeto. Seu objetivo é traduzir essas informações para um padrão compatível com o SNMP e armazená-las na MIB – *Management Information Base* - local.
- o Objeto gerenciado – um objeto gerenciado é qualquer elemento que possa ser gerenciado.

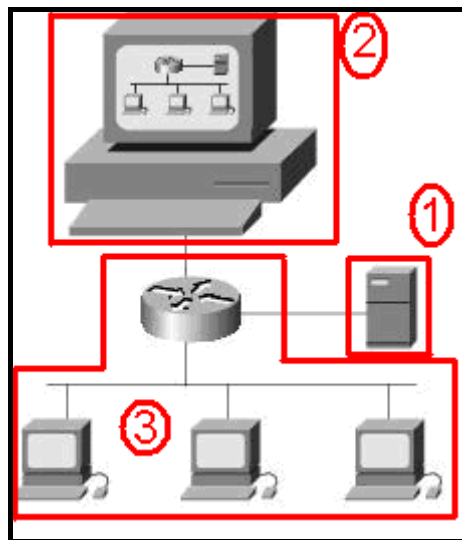


Figura 1 – Componentes básicos para gerenciar uma rede [CISCO, 2006].

A gerência de redes facilita a troca de informações em uma rede de computadores com diversos elementos interconectados [CISCO, 2006]. O item um da figura 1 demonstra o componente de gerenciamento denominado gerente. O item número dois indica uma estação de trabalho utilizada por um administrador de redes, onde é possível visualizar informações dos objetos gerenciados. Por fim, no item três verificamos os agentes e objetos gerenciados dentre eles roteadores e estações de trabalho.

## 2.2 SNMP

O *Simple Network Management Protocol* (SNMP) foi definido e devidamente detalhado na RFC 1098. Nesta RFC são definidos a arquitetura SNMP, os elementos e objetivos desta arquitetura, a maneira que os objetos trocam informações, as operações suportadas, dentre outros itens.

Os RFC (*Request for Comments*) são documentos técnicos ou informativos que discutem os mais diversos aspectos relacionados à Internet. Os assuntos variam desde especificações, padrões e normas técnicas até questões históricas acerca da rede mundial de computadores. Os RFC são documentos públicos, qualquer pessoa tem acesso a eles, podendo ler, comentar, enviar sugestões e relatar experiências sobre o assunto [ICP BRASIL, 2006]. O site <http://www.ietf.org/rfc.html> contém uma base de dados com todas as RFCs.

Este protocolo evoluiu através de três gerações. A versão atual é conhecida como SNMPv3, e os predecessores são conhecidos como SNMPv1 e SNMPv2 [COMER, 2006].

A versão 1 é definida em três RFC listadas a seguir. A segurança desta versão é baseada em comunidades [ABREU e PIRES, 2004].

- RFC 1155 – define os mecanismos usados para descrever e nomear os objetos que serão gerenciado, definindo o *Structure of Management Information (SMI)*;
- RFC 1212 - define um mecanismo de descrição mais conciso, mas é inteiramente consistente ao SMI;
- RFC 1157 – atualização da RFC 1098 que define o SNMP.

Já a versão 2, também denominada *SNMPv2c*, busca implementar e corrigir algumas deficiências da versão anterior, como: adicionar mais segurança, novas operações, comunicação entre servidores com a função de gerente e configuração remota. Esta versão é definida pelas RFCs 1905, 1906 e 1907 [ABREU e PIRES, 2004].



A versão 3 implementa novas funcionalidades somente em relação à segurança, sendo elas: autenticação, privacidade, autorização e controle de acesso. Todas as operações definidas nas versões 1 e 2 são compatíveis. As RFCs 3411, 3412, 3413, 3414, 3415, 3416, 3417, 3418, 2576, 2570 e 2786 definem essa versão [COMER, 2006].

Entretanto, essa última versão ainda não está sendo amplamente utilizada, pois os fabricantes de hardware e software necessitam de tempo para adequar os seus produtos e realizar testes de funcionamento [COMER, 2006].

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 [SZTAJNBERG, 2003].

A suíte de protocolos TCP/IP, assim denominada em função de seus dois protocolos mais importantes, pode ser utilizada na comunicação entre praticamente qualquer conjunto de redes interconectadas. Por exemplo, TCP/IP pode ser utilizado para interconectar um conjunto de redes de um único prédio, de um campus universitário ou até de um conjunto de campus [TANENBAUM, 2003].

Em redes de computadores muito congestionadas e com muitos elementos para serem gerenciados o TCP não é a melhor escolha. É interessante ressaltar que o TCP não resolve todas as questões e que o SNMP foi projeto para funcionar em ambientes com problemas – caso não existissem problemas nas redes de computadores não haveria necessidade de monitorá-las. O protocolo UDP necessita de pouco recurso de hardware. Caso não seja possível transportar os dados ele não irá sobrecarregar a rede com tentativas de retransmissão. Logo, o impacto na rede de dados é mínimo. Por isso, o UDP foi escolhido em vez do *Transmission Control Protocol* (TCP). [MAURO e SCHMIDT, 2005].

O SNMP utiliza a porta UDP 161 para envio e recebimento de requisições e a porta 162 para recebimento de *traps* de elementos

gerenciados. Qualquer objeto gerenciado que implemente SNMP precisa utilizar essas portas como padrão. Porém, alguns fabricantes permitem que estas portas padrão sejam alteradas. Obviamente, caso as portas padrão sejam alteradas é necessário reportar ao servidor gerente estas alterações para que ele possa efetuar requisições nas portas corretas.



Figura 2 – Modelo de comunicação TCP/IP e SNMP [MAURO E SCHMIDT, 2005].

A figura 2.1 indica a forma como ocorre o processo de gerenciamento, utilizando SNMP, por um servidor gerente (NMS) para um agente em uma rede de dados que utiliza TCP/IP.

Este processo ocorre da seguinte forma:

- Aplicação – primeiramente, a própria aplicação SNMP, seja o gerente seja o agente, decide o que irá fazer. Por exemplo, o servidor gerente pode efetuar uma requisição de informação ou receber um *trap* enviado pelo agente.
- UDP – provê a comunicação necessária entre os elementos de gerência, desempenhando o seu papel de protocolo de transporte.
- IP – possibilita a entrega dos pacotes SNMP, utilizando o endereço IP definido.

- Protocolo de acesso ao meio – o passo final para que um pacote SNMP seja entregue ao seu destinatário é que ele trafegue pelo meio físico e chega até a outra ponta.

A simplicidade do SNMP, que o ajudou a ocupar o campo de gerenciamento das grandes redes, é a principal responsável pelas suas limitações. A segunda versão do SNMP, conhecida como SNMPv2 e lançada em 1993, consertou várias deficiências da versão original, nos campos de eficiência, segurança, flexibilidade e na distribuição da sua administração. Com estas melhorias, o SNMPv2 se tornou uma ferramenta afinada com as necessidades das grandes redes dos dias de hoje. Entre os objetivos de SNMPv2 está também a coexistência com a versão original, e a sua utilização em ambientes diferentes de TCP/IP, inclusive em redes OSI, AppleTalk e NetWare [AMMIRABILE, 2004].

Este protocolo é utilizado para administrar uma inter-rede, tendo como função definir como o gerente irá se comunicar com o agente. É através dele que são estipulados os formatos de requisições que o gerente irá enviar e das respostas que o agente irá retornar. Nele também podem ser definidas as codificações das mensagens que o mesmo retorna [COMER, 2006]. Pode-se dizer que toda a inteligência fica nos servidores de gerenciamento [TANENBAUM, 2003].

Os principais comandos SNMP são:

- Get – coleta uma determinada informação de um elemento de rede.
- Walk – efetua uma varredura no caminho indicado, coletando mais de uma informação do objeto gerenciado.
- GetNext – permite que seja coletado a próxima instância de objetos do agente.
- Set – utilizado para configuração e requisições de controle.

- o Trap – o agente irá enviar informações para um gerente, sem ter sido requisitado previamente.

A figura 3 demonstra o formato de pacote padrão utilizado pelos comandos snmpget, snmpwalk, snmpgetnext, snmpset e snmptrap. O campo *PDU type* indica o tipo de operação que será realizada; *Request ID* é utilizado para identificar uma requisição com a sua resposta; *Error status* indica se ocorreu algum erro, só é utilizado na operação de resposta; *Error index* é responsável por associar um erro a instância do objeto; *Variable bindings* é utilizado para gravar os dados de cada requisição ou resposta.

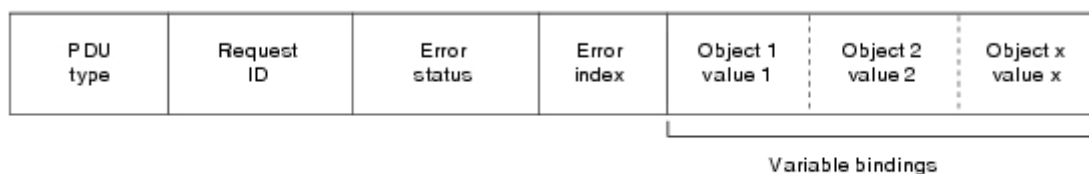


Figura 3 – Formato de uma mensagem SNMP [CISCO, 2006].

## 2.2.2 COMUNIDADES

As versões SNMPv1 e SNMPv2 utilizam o conceito de comunidades para estabelecer a confiança entre gerentes e agentes. O agente pode ser configurado com três tipos de comunidades: somente leitura (read-only), leitura e escrita (read-write) e trap (utilizada para o recebimento de traps encaminhados por um agente). Os nomes das comunidades são basicamente senhas; não existe diferença real entre a string comunidade e a senha que é usada para acessar um computador [MAURO e SCHMIDT, 2005].

Uma falha de segurança conhecida é que a comunidade é enviada em texto claro. Entretanto, existem maneiras de diminuir o risco caso alguém tente realizar um ataque: configurar o agente para encaminhar um trap caso ocorra falha na autenticação da comunidade; evitar a utilização de palavras do dicionário, ou nome de familiares; alternar caracteres alfanuméricos com letras maiúsculas e minúsculas; restringir respostas, respondendo somente para hosts conhecidos e ainda configurar firewalls para permitir tráfego UDP do gerente para o agente na porta 161 e do agente para o gerente na porta 162.

Sistemas interconectados sempre estarão suscetíveis a falhas. Entretanto, caso as precauções listadas acima sejam tomadas o risco será reduzido consideravelmente.

Na figura abaixo podemos verificar como é configurado um agente SNMP em servidores Windows. O item “Enviar interceptação de autenticação” indica que será encaminhado um trap para o servidor gerente caso ocorra falha na autenticação da comunidade. No item “Nomes de comunidade aceitos” foi configurada uma comunidade seguindo os critérios de segurança listados acima e permitindo somente leitura aos dados gravados na MIB. Por fim, foi selecionado o item “Aceitar pacotes SNMP destes hosts” e definido o IP do servidor gerente que realizará as requisições SNMP.

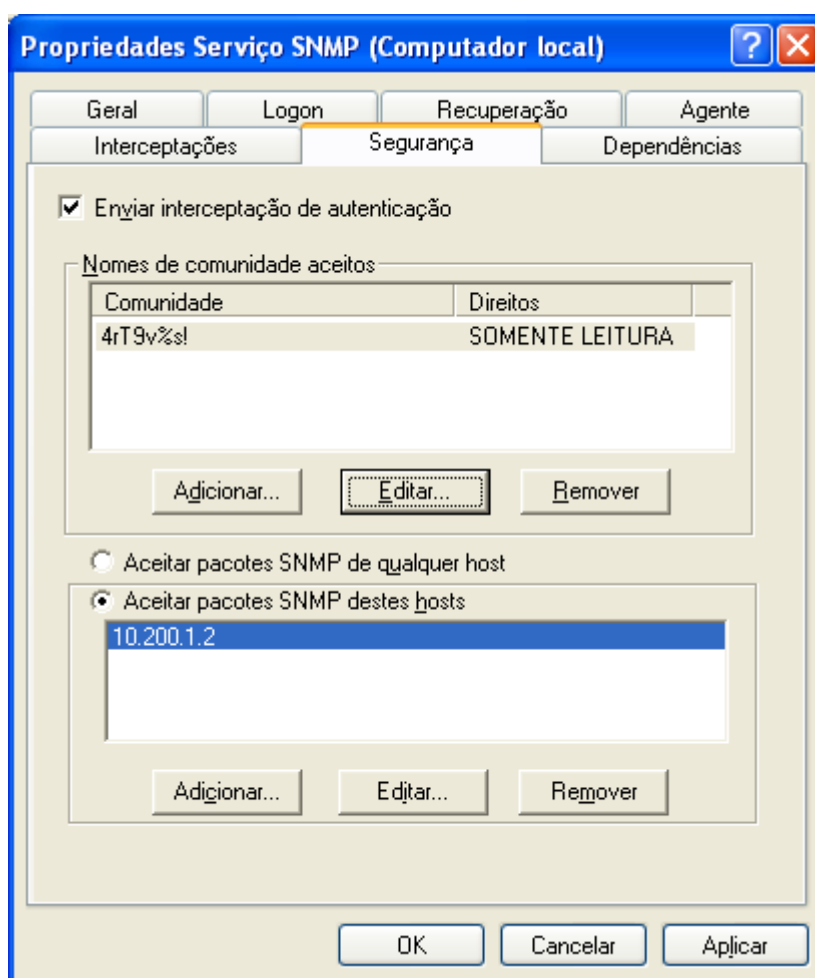


Figura 4 – Configurando a comunidade de um agente Windows.

### 2.2.3 MIB

O SNMP é o protocolo de acesso entre os objetos gerenciados, onde os dados a serem coletados ou o status dos componentes gerenciados ficam por conta da MIB – *Management Information Base*. Sendo assim as MIB podem ser definidas com vários padrões diferentes ou até mesmo por grupos de pessoas que tenham alguma necessidade diferente das já implementadas, sem que haja necessidade de alteração do protocolo de transporte [COMER, 2006].

A base de dados, controlada pelo agente SNMP, é referenciada como MIB. Esta base é definida de forma padronizada. O gerente solicita que o agente busque ou altere uma informação específica através do caminho da MIB fornecido.

Um elemento que está sendo gerenciado precisa administrar corretamente as informações que poderão ser acessadas pelo gerente. Por exemplo, um roteador mantém as estatísticas dos status de suas interfaces de rede, tráfego de pacotes de entrada e saída, datagramas descartados, e mensagens de erros geradas; um modem registra as estatísticas sobre o número de bits (ou caracteres) enviados e recebidos [COMER, 2006]. Conforme citado acima, o SNMP permite que um gerente acesse as estatísticas, não especificando exatamente qual informação pode ser acessada em um elemento.

A MIB é conhecida como um padrão que especifica quais dados um agente deve manter em sua base de dados local, as operações permitidas e seus significados. A MIB que mantém as informações do protocolo IP indica que o software agente precisa contar todos os octetos que chegam a cada interface de rede e armazenar esta informação.

Os objetos de uma MIB são definidos utilizando a notação sintática ASN.1 – *Abstract Syntax Notation 1*. Esta notação define o padrão que deverá ser utilizado para representar e transmitir dados entre os servidores gerentes e os agentes, no contexto do SNMP [MAURO e SCHMIDT, 2005].

O TCP/IP utiliza uma MIB que categoriza as informações de gerência em diversos grupos. Essa categorização é muito importante, pois baseado nela que são definidos os identificadores únicos de cada grupamento.

Tabela 1 – Exemplo de categorias da MIB [COMER, 2006].

| <b>Categoria MIB</b> | <b>Contém Informação Sobre</b>    |
|----------------------|-----------------------------------|
| <b>system</b>        | O sistema operacional do elemento |
| <b>interfaces</b>    | Interfaces de rede individuais    |
| <b>at</b>            | Tradução de endereços             |
| <b>ip</b>            | Software IP                       |
| <b>icmp</b>          | Software ICMP                     |
| <b>tcp</b>           | Software TCP                      |
| <b>udp</b>           | Software UDP                      |
| <b>bgp</b>           | Software Border Gateway Protocol  |
| <b>rip-2</b>         | Software RIP                      |
| <b>dns</b>           | Software DNS                      |

A solução de manter as definições da MIB independente do protocolo de gerenciamento utilizado foi uma vantagem, tanto para os fabricantes quanto para os usuários. Um fabricante pode vender um roteador já com o agente SNMP incluído, com a certeza que este agente continuará compatível ao padrão mesmo depois que novos itens sejam adicionados a MIB. Já o usuário poderá utilizar o mesmo software para gerenciar diferentes objetos, mesmo que as MIBs destes equipamentos não sejam idênticas.

Na figura 5 é representada a estrutura de uma MIB padrão.

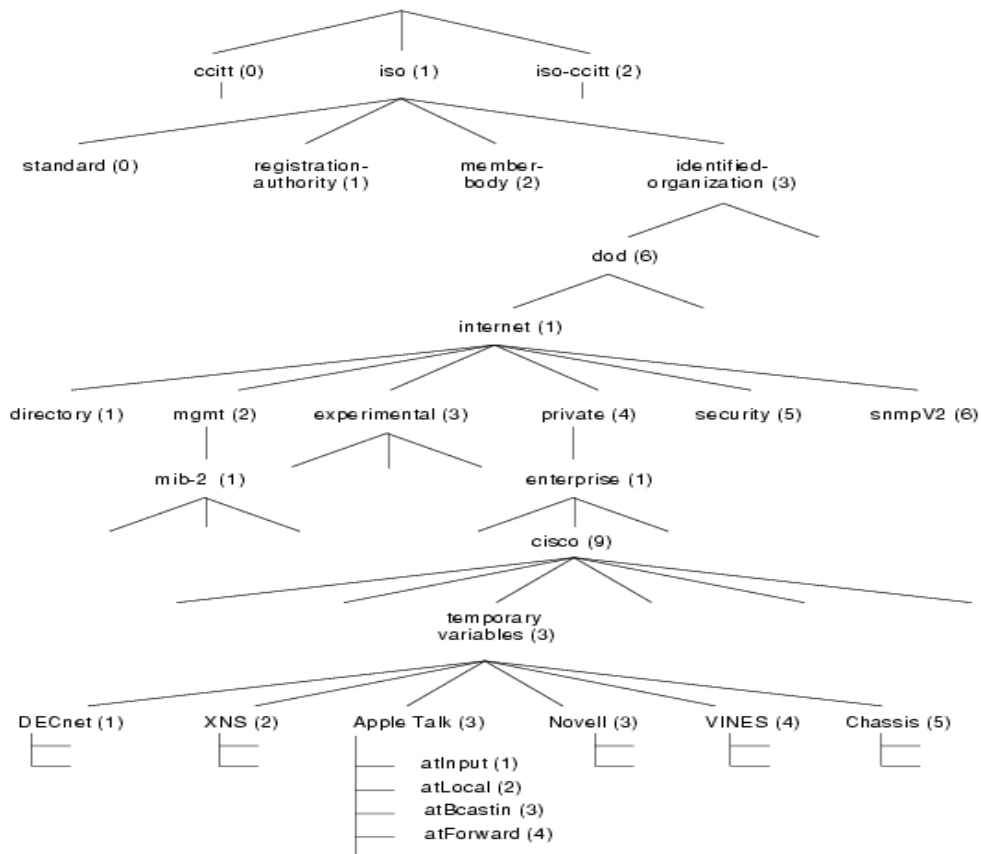


Figura 5 – Estrutura da MIB [CISCO, 2006].

Dentro de cada categoria são definidas variáveis da MIB. Nessas variáveis estão os valores ou informações de cada item, gravados pelo agente SNMP de cada elemento. A tabela 2 lista exemplos de variáveis da MIB, com suas respectivas categorias e o significado de cada uma.

Tabela 2 – Exemplo de variáveis da MIB [COMER, 2006].

| Variável da MIB       | Categoria  | Significado                          |
|-----------------------|------------|--------------------------------------|
| <b>sysUpTime</b>      | system     | Tempo ligado                         |
| <b>ifNumber</b>       | interfaces | Número de interfaces de rede         |
| <b>ipInReceives</b>   | ip         | Número de pacotes recebidos          |
| <b>ipOutNoRoutes</b>  | ip         | Número de falhas de roteamento       |
| <b>icmpInEchos</b>    | icmp       | Número de requisições ICMP recebidas |
| <b>tcpMaxConn</b>     | tcp        | Máximo de conexões TCP permitidas    |
| <b>udpInDatagrams</b> | udp        | Número de pacotes UDP recebidos      |



## 2.3 ÁREAS FUNCIONAIS DE GERENCIAMENTO

Em virtude da complexidade natural das redes de computadores, gerenciá-las de forma eficiente e eficaz representa um grande desafio. Preocupada em facilitar e organizar o desenvolvimento de projetos destinados ao gerenciamento de redes, a ISO propõe no Modelo de Gerenciamento OSI, a divisão das tarefas de gerenciamento em cinco áreas funcionais descritas a seguir [STALLINGS, 1999]:

- Gerenciamento de Falhas: detecta, localiza e corrige problemas no hardware ou software de rede. Determina e, normalmente registra, que uma falha ocorreu, determina sua localização e então tenta repará-la. Inclui também processos para relatar problemas a usuários finais e gerentes, assim como para controlar tendências relacionadas a problemas [AMMIRABILE, 2004].
- Gerenciamento de Desempenho: mede o comportamento de dispositivos gerencia em uma rede, avaliando o desempenho através de seu monitoramento a fim de comprovar e medir o bom funcionamento da rede [SOUZA, 2001].
- Gerenciamento de Segurança: tem por objetivo criar, controlar e eliminar mecanismos de segurança, bem como distribuir informações relevantes à segurança, gerando alarmes para o gerente da rede quando for detectado algum evento de segurança [PINTO, 1998].
- Gerenciamento de Contabilidade: provê meios para se medir e coletar informações a respeito da utilização dos recursos e serviços de uma rede para podermos saber qual a taxa de uso destes recursos. Deve existir um padrão para obtenção e representação das informações de contabilização [SZTAJNBERG, 2003].
- Gerenciamento de Configuração: é um conjunto de atividades de apoio ao desenvolvimento que permite que as mudanças inerentes ao desenvolvimento sejam absorvidas pelo projeto de maneira

controlada, mantendo a estabilidade na evolução do software [DIAS, 2008].

## CAPÍTULO 3 – FERRAMENTAS DE GERÊNCIA

---

Para que os administradores de redes consigam monitorar de forma eficaz e abrangente todo ambiente de rede, podem ser desenvolvidos softwares para controlar e monitorar componentes, como *hosts*, roteadores, bridges, demonstrando seus status, bem como obter estatísticas do funcionamento destes. Com isso o administrador da rede tem a possibilidade e flexibilidade de alterar configurações da rede para que seus componentes trabalhem da melhor forma possível [COMER, 2006].

### 3.1 MRTG

O MRTG – *Multi Router Traffic Grapher* – é uma ferramenta gratuita, sob a licença GPL – código aberto – utilizada para monitorar o tráfego, possibilitando a geração de gráficos que podem ser visualizados em formato HTML. Este aplicativo foi desenvolvido utilizando as linguagens Perl e C. Os sistemas operacionais suportados são Linux, Unix, Windows e até sistemas Netware [MRTG, 2008].

Além de uma visão diária detalhada, o MRTG também cria gráficos semanais, mensais e anuais dos dados coletados. Isto só é possível, pois o MRTG mantém um arquivo com todas as informações previamente requisitadas. Esse arquivo de log é automaticamente consolidado de maneira que não cresça demasiadamente e os dados relevantes dos últimos dois anos permaneçam intactos [MRTG, 2008].

Apesar de a ferramenta apresentar melhor desempenho criando gráficos sobre utilização de interfaces de rede em roteadores, ela também pode ser configurada para criar gráficos de utilização de memória, processador e espaço em disco em servidores. Esta ferramenta é bastante útil em determinar os períodos em que ocorre pico de utilização dos recursos gerenciados. Como por exemplo, o administrador de redes pode verificar que durante o horário comercial a utilização do link com a Internet apresenta um grande aumento em relação aos outros períodos do dia. Em outros casos, ele pode verificar que o link está sendo subutilizado e informar ao seu gerente que a empresa não necessita de um link com tamanha capacidade. [MAURO e SCHMIDT, 2005].

As coletas são realizadas através de comandos *snmpget*, utilizando o protocolo de gerencia SNMP. Para isso, são criados e configurados arquivos indicando qual o elemento que deverá ser gerenciado, sua comunidade, os dados que deverão ser coletados e a periodicidade de cada coleta. Após efetuar as coletas, os dados são armazenados. Esses dados são utilizados para criar gráficos diários, semanais, mensais e anuais em formato HTML, que podem ser facilmente visualizados através de um navegador da internet.

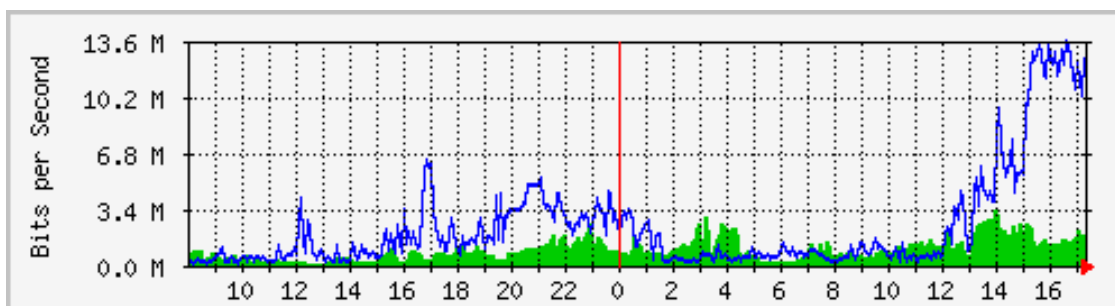


Figura 6 – Gráfico do MRTG [MRTG, 2008].

Na figura 6 podemos observar que no dia anterior, lado esquerdo da linha vermelha, entre as 15 e 17 horas da tarde, que o pico de utilização de uma determinada interface foi de 6.8 Mbits. Entretanto, no dia seguinte, lado direito do gráfico, tomando o mesmo período como base, a utilização ultrapassou 13 Mbits de utilização. Desta forma, o administrador de redes sabe

exatamente como está a saúde do seu equipamento de rede e poderá tomar ações com base nos gráficos gerados.

### 3.2 ITM

O *IBM Tivoli Monitoring* (ITM) é uma família de produtos que possibilita a localização, isolamento e correção de problemas de desempenho e disponibilidades mais rapidamente no nível de sistema, subsistema, banco de dados ou aplicativo. O IBM Tivoli Monitoring fornece relatórios que você pode utilizar para rastrear as tendências e resolver problemas [IBM, 2008]. Vale lembrar que esta é uma solução paga.

Como pode ser observado na figura 3.1, O ITM fornece uma estrutura composta de:

1. Cliente – interface com o usuário baseada em Java para visualizar e monitorar a rede corporativa.
2. TEPS – *Tivoli Enterprise Portal Server* – possibilita a recuperação, manipulação e análise de dados dos agentes de monitoração.
3. TEMS – *Tivoli Enterprise Monitoring Server* – o servidor de monitoramento atua como gerenciador e ponto de controle para alertas recebidos dos agentes de monitoramento e coleta os dados de desempenho e disponibilidade.
4. Agente – os agentes são instalados nos sistemas cujos aplicativos ou recursos deverão ser monitorados. O agente coleta os dados e os transmite para o servidor TEMS ao qual está conectado.

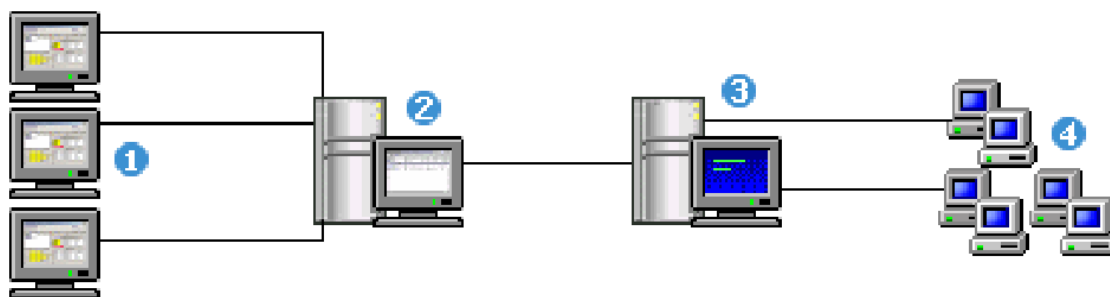


Figura 7 – Fluxo de dados do IBM Tivoli Monitoring [IBM, 2008].

O Tivoli Enterprise Portal é a interface para seus produtos IBM Tivoli Monitoring. Da mesma forma que você utiliza a home page de seu navegador como um ponto inicial para navegar na Internet, pode utilizar o Tivoli Enterprise Portal para obter uma visão geral de alto nível de seu ambiente de rede. A janela de aplicativo mostra uma visualização de tipo árvore de sua rede monitorada (à esquerda), com ícones de alerta que aparecem quando ocorrem problemas. O restante da janela é preenchido com visualizações relacionadas ao item escolhido na árvore. [IBM, 2008].

O ITM é uma ferramenta proprietária. Seus custos são elevados e sofrem variação dependendo da necessidade da Organização. A quantidade de processadores que está instalada nos servidores e agentes, onde serão instalados os produtos, influenciam diretamente o preço que deverá ser pago à IBM para utilização dessa solução de gerencia. Os valores não foram divulgados pela IBM do Brasil.

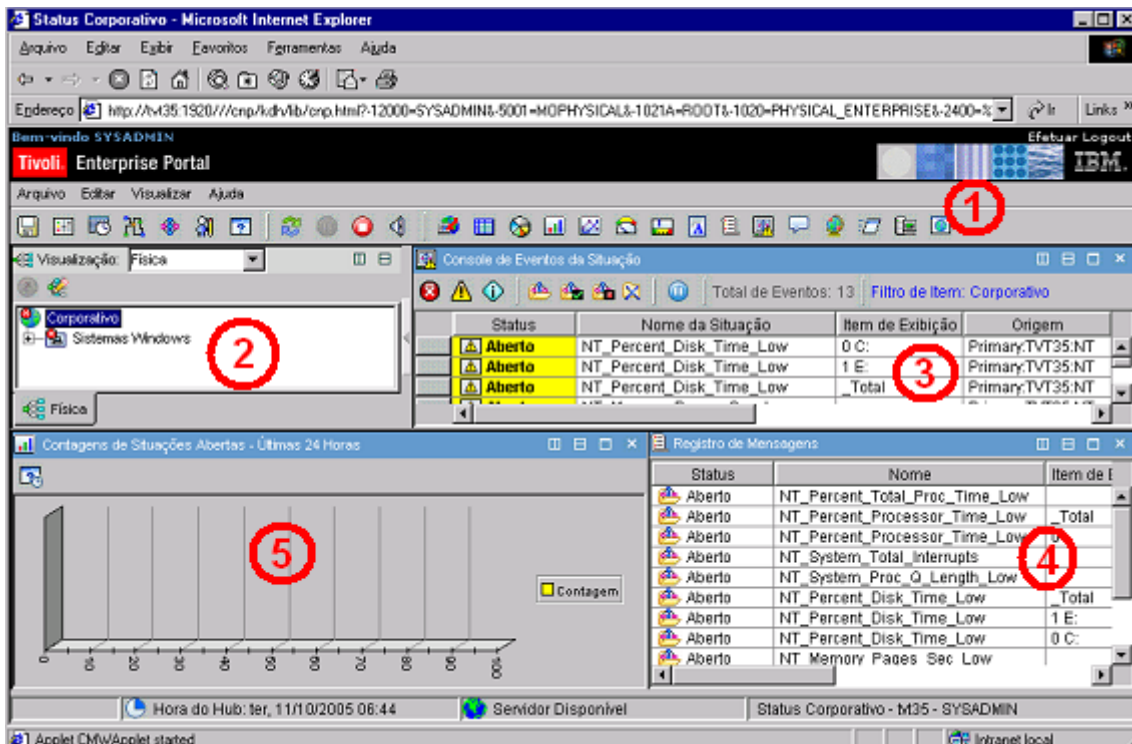


Figura 8 – Tela do IBM Tivoli Enterprise Portal [IBM, 2008].

A figura 8 mostra a interface de gerenciamento WEB – Tivoli Enterprise Portal. Uma breve explicação a seguir:

- No primeiro item observa-se uma barra de tarefas. Nesta barra de tarefas estão links para os aplicativos de cadastramento de usuários, criação de alertas, criação de consultas ao banco, etc.
- O segundo item apresenta uma árvore física de todos os servidores que estão reportando dados para o ITM.
- No terceiro item é apresentada uma console de eventos com todos os alertas que estão disparados no momento. Nessa tela o operador poderá tomar conhecimento dos eventos e tomar alguma ação.
- O quarto item mostra um histórico dos últimos eventos.
- No quinto item é apresentado um gráfico com todos os alertas abertos nas últimas 24 horas. No caso acima, nenhum dos alertas se enquadraram nessa categoria.

### 3.3 HP OPENVIEW

O HP OpenView é uma suite de aplicações que fornece uma solução completa para gerenciamento de redes. Essa solução é altamente flexível, permitindo até a utilização de plugins. Entretanto, é proprietária e pertence à empresa de computadores Hewlett Packard (HP).

Segundo Tauroco (1999), algumas aplicações incluídas no HP OpenView são:

- Network Node Manager (OV NNM): tem por objetivo maximizar a produtividade produtividade da rede e dos seus administradores, assim como, aumentar a disponibilidade dos equipamentos seja numa simples rede local ou redes locais interconectadas. Proporciona uma visão ampla da rede através de um monitor, permitindo controlar o ambiente por inteiro. Proporciona, também, gerenciamento de falhas, configuração e performance de redes TCP/IP. Possui um conjunto de características que reduzem o tempo de realização das tarefas de gerenciamento de horas para minutos.
- Kit do Desenvolvedor e Plataforma SNMP: é composto de um ambiente de desenvolvimento para os técnicos que estão criando soluções de gerenciamento de sistemas e redes, em ambientes TCP/IP usando Simple Network Management Protocol (SNMP) e um ambiente de execução dessas soluções criadas.
- Agente TCP/IP: possui a capacidade necessária para gerenciar um HP 9000 séries 300, 400, 600, 700, 800 e um computador SunSPARC via SNMP (Simple Network Management Protocol).
- Agente SNMP Extensível: é uma tecnologia disponível para gerenciar recursos críticos da rede, como servidores e aplicações de usuário final. Ele possibilita o controle centralizado de todas as



propriedades da rede que são fundamentais para o seu funcionamento.

- Gerenciamento Distribuído (DM): provê um ambiente de desenvolvimento e execução de aplicações, onde são criadas soluções de gerenciamento de sistemas e redes.

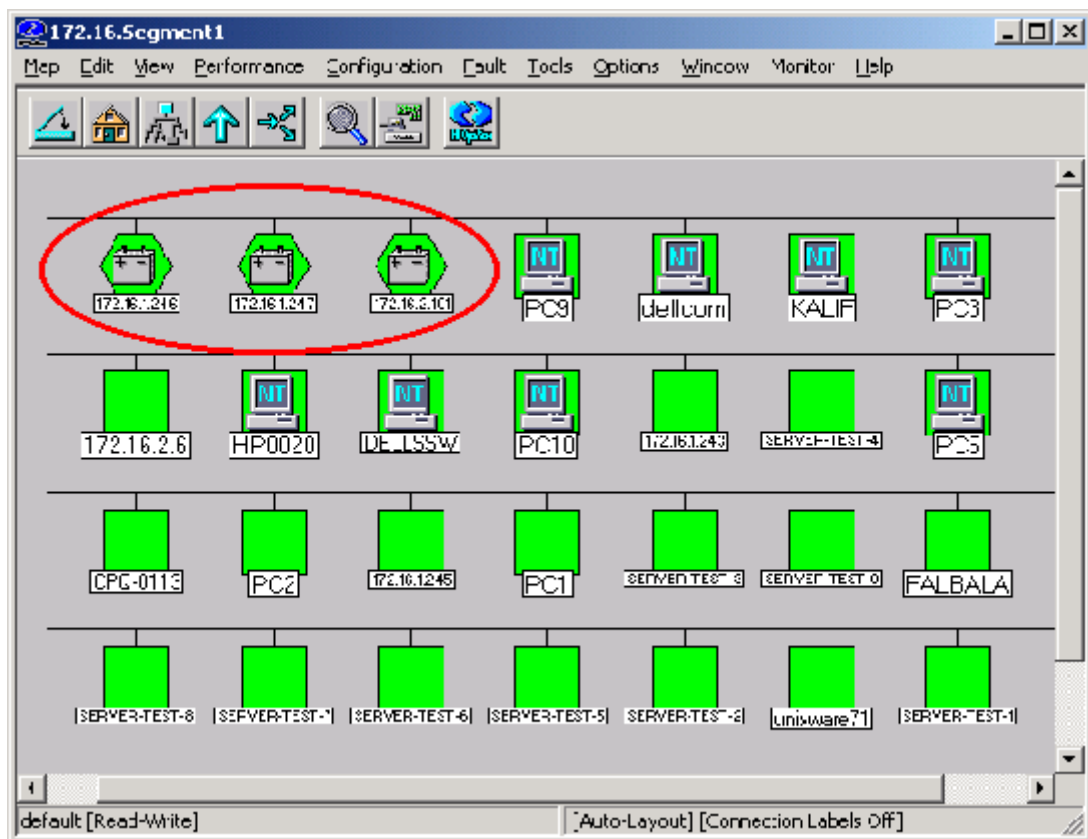


Figura 9 – Tela do HP OpenView [TAUROCO, 1999].

A figura 12 demonstra uma tela da ferramenta HP OpenView. Nessa tela observa-se que foram encontrados ou “descobertos” elementos do segmento de rede 172.16.\*. Dentre os objetos gerenciados encontram-se: no-breaks (dentro do círculo vermelho), estações de trabalho (símbolo NT), servidores e demais itens que não puderam ser identificados através de consulta SNMP. Todos os elementos estão ativos, apresentando a cor verde.

## 3.4 NAGIOS

O Nagios é um aplicativo, freeware, utilizado para monitorar *hosts* e serviços com o objetivo de informar o administrador de redes de problemas encontrados no ambiente antes que seus clientes sejam afetados pelas indisponibilidades. Foi desenvolvido em linguagem C, Perl e Shell para ser executado em sistemas operacionais Linux, mas também pode ser instalado na maioria dos sistemas operacionais *UNIX-LIKE* como FreeBSD, Apple OS X e Solaris. É mantido sob a licença GPL – *open-source*, com interface de gerenciamento via web [NAGIOS, 2008].

O processo responsável por realizar as verificações periódicas monitora elementos e serviços configurados pelo administrador da ferramenta. Esta aplicação permite o uso de “plugins” que possibilita o retorno de informações para o Nagios. Quando o Nagios detecta algum problema, o aplicativo pode enviar notificações para os contatos técnicos. As informações sobre as coletas e dados históricos podem ser consultadas via browser [NAGIOS, 2008].

Podemos observar na figura 3.2 como deve ser configurado o arquivo de execução de verificações do Nagios. No exemplo utilizado é definido o comando `check_ping` com os parâmetros necessários para uma monitoração via ICMP (verifica se um *host* está disponível ou indisponível).

```
# -- /etc/nagios/mysite/checkcommands.cfg
...
define command{
command_name check_ping
command_line $USER1$/check_icmp -H $HOSTADDRESS$ -w $ARG1$ \
-c $ARG2$ -p 5
}
```

Figura 10 – Exemplo do comando checkping [BARTH, 2005].

O Nagios possibilita a monitoração de diversos itens, que o fazem uma poderosa ferramenta de monitoração. Dentre os fatores que podem ser monitorados pelo Nagios, temos: serviços de rede como SMTP, POP3, HTTP, NNTP, PING, etc; recursos de um elemento como utilização de CPU, espaço

em disco e utilização de memória, processos em execução e arquivos de log; fatores climáticos como temperatura; arquitetura que permite a utilização de plugins desenvolvidos pela comunidade da Internet; habilidade de definir ações proativas no caso de indisponibilidades; interface WEB para visualização de problemas; dentre outros [BARTH, 2005].

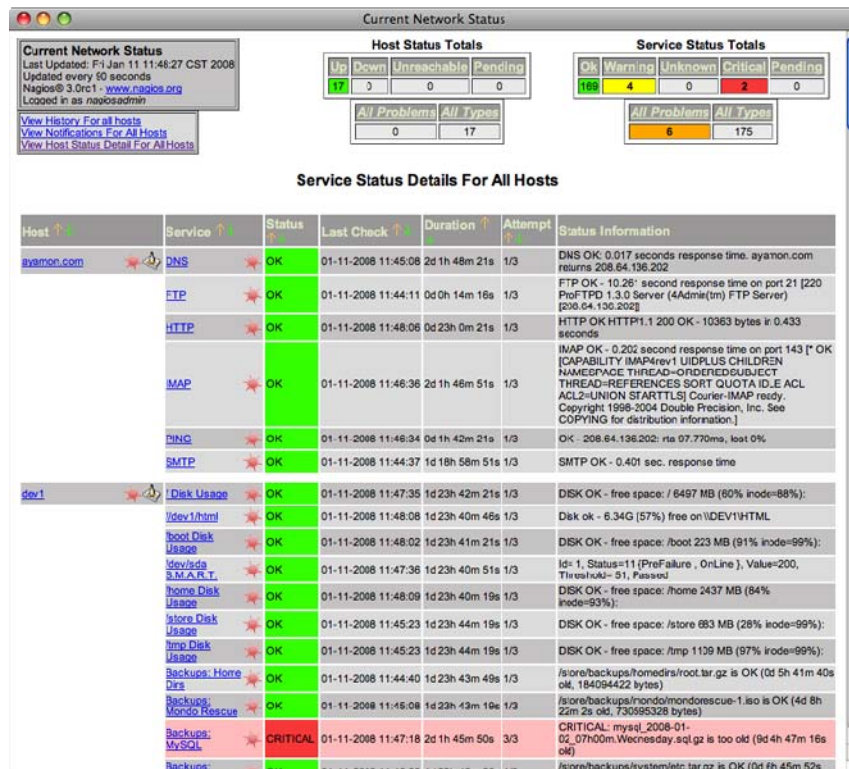


Figura 11 – Console de eventos Nagios [NAGIOS, 2008].

Essa ferramenta atende as seguintes áreas funcionais de gerenciamento, conforme tabela abaixo:

Tabela 3 – Comparação Nagios e áreas de gerenciamento.

| Áreas de Gerenciamento  | Nagios |
|-------------------------|--------|
| Gerenciamento de Falhas | Sim    |

|                                       |     |
|---------------------------------------|-----|
| <b>Gerenciamento de Desempenho</b>    | Sim |
| <b>Gerenciamento de Segurança</b>     | Sim |
| <b>Gerenciamento de Contabilidade</b> | Sim |
| <b>Gerenciamento de Configuração</b>  | Sim |

Segundo David Josephsen, o Nagios é a melhor ferramenta de monitoração do mercado para gerenciamento de sistemas e de redes de computadores, incluindo as ferramentas proprietárias e livres. É um conjunto de scripts modular, fácil de trabalhar, direto e bastante expansível. A licença open source do Nagios o torna um aplicativo gratuito, de fácil acesso (pode ser encontrado na Internet), e permite a utilização de *plugins* para atender as necessidades do administrador de redes [JOSEPHSEN, 2007].

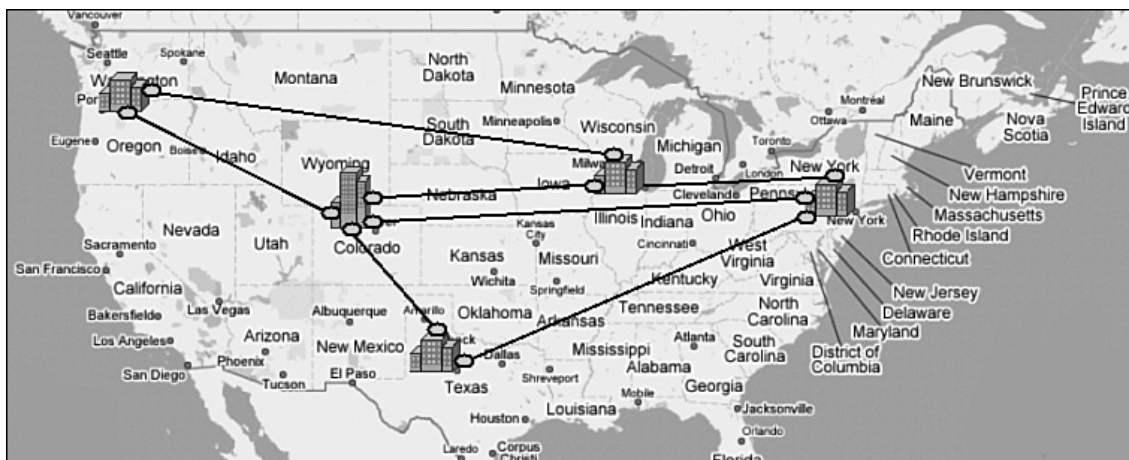


Figura 12 – Elementos gerenciados, NAGVIS Map [JOSEPHSEN, 2007].

## CAPÍTULO 4 – DESENVOLVIMENTO DO PROJETO

---

A administração de um ambiente lógico e físico corporativo tornou-se uma tarefa complexa nos ambientes de redes atuais. Essa complexidade exige dos administradores de rede, alta capacidade de absorção das novas tecnologias em um curto período de tempo [MSDN TRAINING, 2001].

O uso de “tecnologias e ferramentas para assegurar serviço de rede sob determinadas condições ou restrições” é uma das principais definições de administração e gerência de redes [SOUSA JR., 2005]. Com o objetivo de auxiliar os administradores de redes em suas tarefas diárias, grandes empresas ao redor do mundo desenvolvem e aprimoram aplicações na área de gerência de rede. Por outro lado, também existem os “pequenos scripts” customizados tão utilizados para atender demandas específicas de cada ambiente.

Os aplicativos de gerência desenvolvidos neste projeto têm o intuito de auxiliar a equipe responsável pela administração de redes nas suas tarefas diárias. O aplicativo que realiza a verificação de pouco espaço em disco, detecta indisponibilidade de serviços importantes e falhas nos agentes de gerenciamento é chamado “SNMP Coleta”. Foi desenvolvido também um aplicativo cliente, denominado “SNMP Cliente”, com a funcionalidade de visualizar os dados coletados rígido em um computador remoto através de uma interface cliente.

As ferramentas desenvolvidas neste projeto têm o objetivo de auxiliar os administradores de redes, usuários e gerentes no processo de gerenciamento de falhas. As demais áreas funcionais de gerenciamento não estão no escopo deste projeto final.

## 4.1 REQUISITOS

Segue abaixo lista dos principais requisitos para o correto funcionamento das aplicações SNMP Coleta e SNMP Cliente:

- Computadores compatíveis com x86.
- Sistemas Operacionais com Suporte: Windows 2000, Windows XP, Windows 2000 Server e Windows 2003 Server.
- Instalação do pacote NET-SNMP no servidor responsável pelas coletas.

## 4.2 FLUXO DE FUNCIONAMENTO

A seguir é detalhado o fluxo de funcionamento da solução de gerência proposta, observando a nomenclatura citada por Brisa (2001).

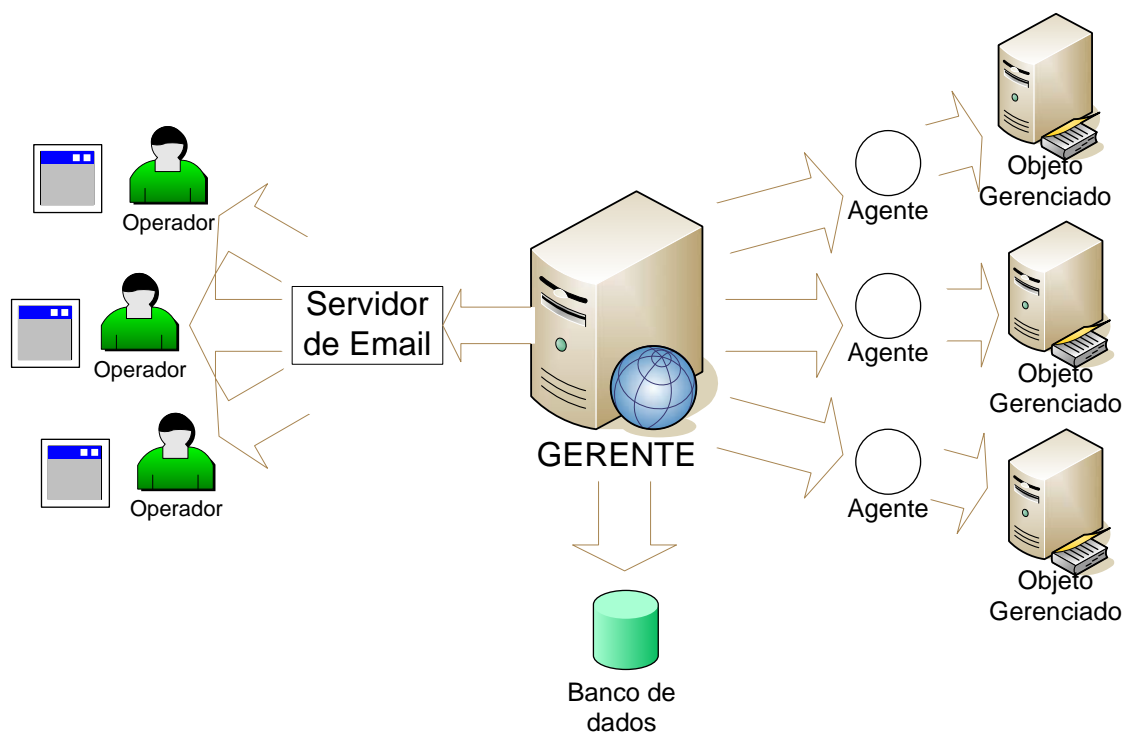


Figura 13 – Fluxo de funcionamento da solução.

- No servidor gerente, deverá ser configurado o arquivo de parâmetros, denominado *parametros.ini*. Nesse arquivo, são configurados os IPs, comunidades e percentual de alerta de disco para cada servidor que será gerenciado.
- Após a configuração no arquivo de parâmetros, a ferramenta SNMP Coleta é iniciada para coletar as informações de disco e serviços ativos dos agentes.
- Neste momento são executadas diversas requisições SNMP para os agentes. Essas requisições têm o objetivo de solicitar informações do agente sobre o servidor que está sendo gerenciado.
- Os agentes coletam as informações e as encaminham para o gerente. O gerente insere essas informações no banco de dados local com o objetivo de manter um histórico de todas as coletas realizadas.
- Caso seja detectada alguma indisponibilidade de serviço ou alto consumo de espaço em disco é iniciada uma rotina que envia email para a equipe responsável por aquele servidor.
- O operador, que trabalha em regime de escala, irá verificar e corrigir a ocorrência. Após corrigir o problema, ele deverá abrir o aplicativo SNMP Resolve e relatar qual ocorrência foi resolvida e a ação tomada.
- No servidor principal, a própria interface de execução do programa *SNMP Coleta* mostrará os dados de cada coleta.
- Na estação de trabalho remota, os dados de todas as coletas realizadas podem ser visualizados. Para isso, basta executar o aplicativo SNMP Cliente e clicar no botão “consulta”. Um histórico com todas as verificações realizadas será mostrado na tela.

#### 4.2.1 DIAGRAMA DE CASO DE USO

O diagrama de caso de uso a seguir demonstra as características da ferramenta desenvolvida SNMP Coleta:

- Gerente: pode realizar coletas individuais.
- Técnico: pode realizar coletas individuais e definir as máquinas que serão gerenciadas, através do arquivo de parâmetros.
- Timer: rotina responsável por realizar as coletas automáticas e enviar email com informações sobre os objetos gerenciados.

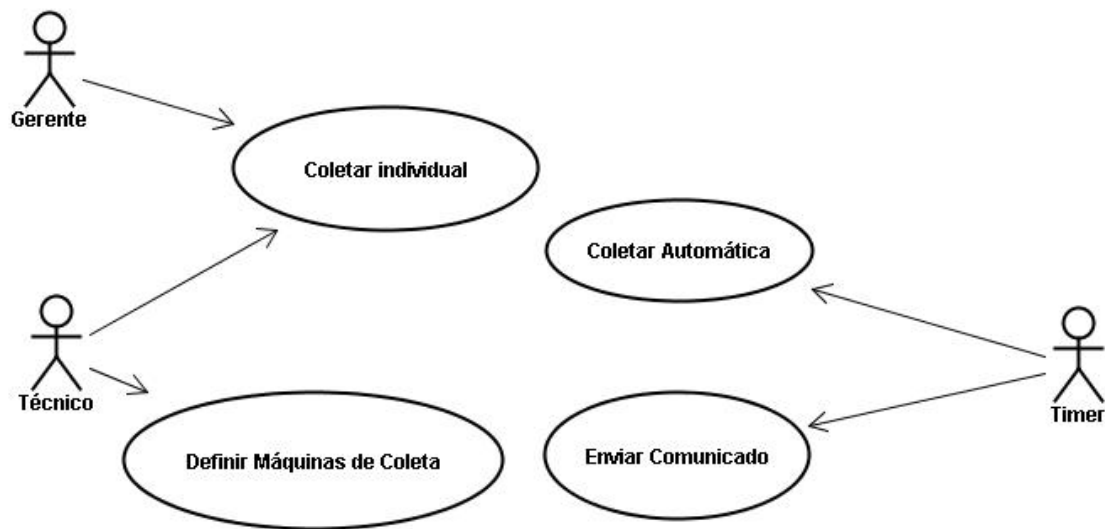


Figura 14 – Diagrama de caso de uso do aplicativo SNMP Coleta.

### 4.3 FERRAMENTAS SNMP COLETA, SNMP CLIENTE E SNMP RESOLVE

A seguir são apresentados os procedimentos utilizados para o desenvolvimento das três ferramentas propostas neste projeto.

#### 4.3.1 APLICATIVO SNMP COLETA

No desenvolvimento do aplicativo SNMP Coleta foi utilizado a ferramenta de desenvolvimento Visual Basic 6.0. Foi criado um projeto padrão,



selecionando a opção “Standard EXE”, adicionada a biblioteca “Microsoft ActiveX Data Objects 2.8 Library” (MDAC) e o componente “Microsoft Hierarchical FlexGrid Control 6.0 (SP4)”.

Criou-se um arquivo de parâmetros, extensão .ini. O objetivo deste é facilitar a configuração da ferramenta, indicando quais servidores deverão ser gerenciados de forma prática. O arquivo deverá ser alterado da maneira detalhada abaixo. Para gerenciar mais de um servidor, cadastrá-los começando a partir da linha seguinte, respeitando a ordem de cada informação.

- 1ª. Linha – colocar o IP do servidor que deverá ser monitorado.
- 2ª. Linha – colocar a comunidade aceita pelo servidor.
- 3ª. Linha – colocar o percentual de disco que se ultrapassado irá gerar um alerta.

Elaborou-se uma rotina que possibilita a verificação automática. Quando um tempo pré-determinado finaliza-se, o aplicativo lê o arquivo de parâmetros e disparar as coletas. Para que essa rotina funcione adequadamente, o programa SNMP Coleta deverá ficar aberto na tela do servidor gerente.

Em seguida, foi desenvolvida a *interface* deste programa.

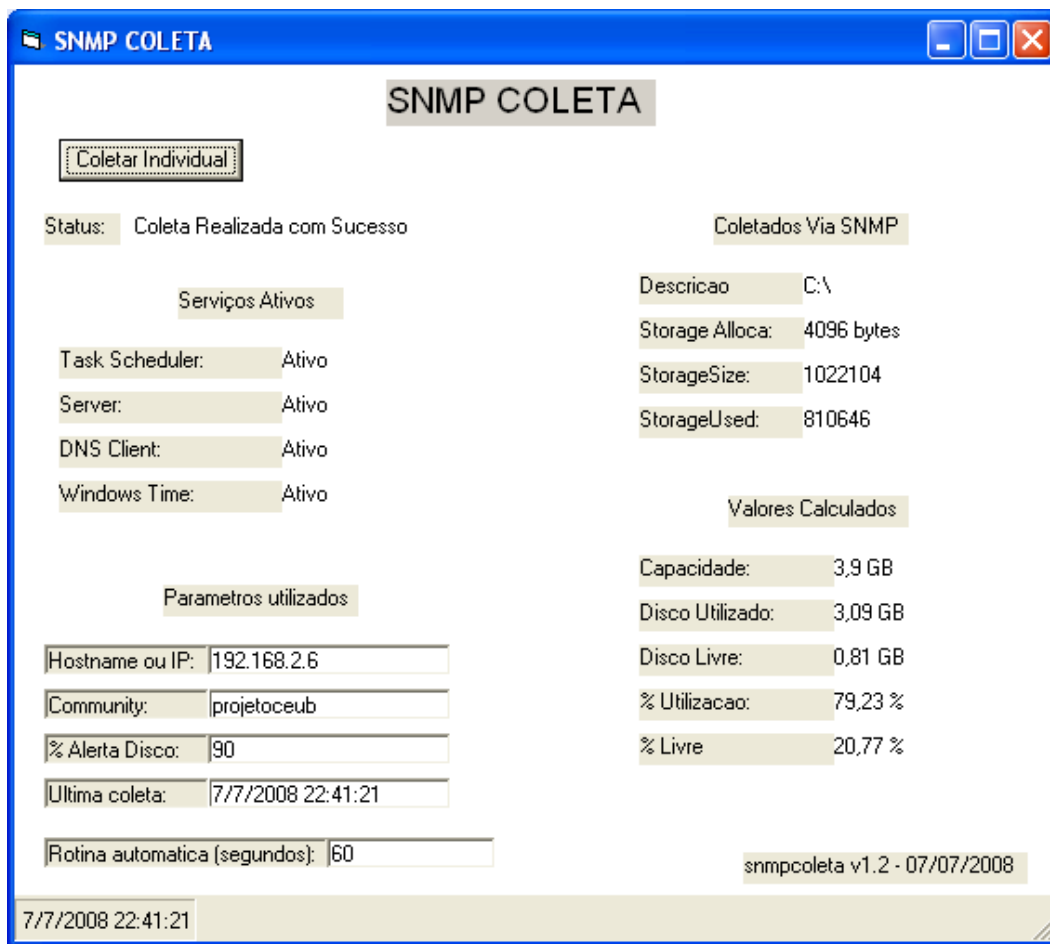


Figura 15 – Interface do programa SNMP Coleta.

Na categoria “Serviços Ativos” estão listados os serviços Windows que serão monitorados no servidor cliente:

- Task Scheduler – Permite que um usuário configure e agende tarefas automatizadas no computador. Se este serviço for interrompido, essas tarefas não serão executadas nos horários agendados.
- Server – Oferece suporte a compartilhamento na rede de arquivo, impressão e pipes nomeados para um computador. Se este serviço for interrompido, quaisquer serviços que dele dependam diretamente não serão inicializados.
- DNS Client – Resolve e armazena em cachê nomes Domain Name System (DNS) para um computador. Se este serviço for

parado, o computador não poderá resolver nomes DNS nem localizar controladores do domínio do Active Directory.

- Windows Time – Responsável por manter a data e hora sincronizadas em todos os clientes e servidores na rede de computadores. Se este serviço for parado, a sincronização de data e hora ficará indisponível.

No item “Coletados Via SNMP” são coletados os seguintes dados brutos da MIB do servidor cliente:

- Descrição – letra da unidade gerenciada. Exemplo: C:\
- Storage Allocation – tamanho do bloco de dados utilizado.
- StorageSize – tamanho total do disco rígido em bytes.
- StorageUsed – total utilizado do disco em bytes.

Posteriormente, após coletado os dados, são realizados cálculos para informar a capacidade total, utilização, espaço livre, percentual de utilização e percentual livre do disco conforme abaixo:

- Capacidade total –  $((\text{StorageSize} \times \text{Storage Allocation}) / 1048576)$ . Divide-se o resultado por 1024 para obter o resultado na unidade “GB”.
- Total de disco utilizado –  $((\text{StorageUsed} \times \text{Storage Allocation}) / 1048576)$ . Divide-se o resultado por 1024 para obter o resultado na unidade “GB”.
- Total de disco livre –  $((\text{StorageSize} - \text{StorageUsed}) \times \text{StorageAllocation}) / 1048576)$ . Divide-se o resultado por 1024 para obter o resultado na unidade “GB”.

- Percentual de disco utilização – (Disco utilizado / Capacidade total) x 100 para obter o resultado em “%”.
- Percentual de disco livre – (Disco livre / Capacidade total) x 100 para obter o resultado em “%”.

É desenvolvida a sub-rotina *test\_snmp*. Nesta sub-rotina os parâmetros IP e Comunidade são utilizados para realizar as verificações de espaço em disco e serviços ativos.

A conexão com o banco de dados é criada. Utiliza-se arquivo *.udl* para possibilitar a conexão com o SGBD Microsoft Access.

```
Dim Controle As New ADODB.Connection
Dim rstControle As New ADODB.Recordset
Dim sqlControle As String

Controle.ConnectionString = "File Name=" & App.Path & "\snmpconexao.udl"
Controle.Open
Set rstControle.ActiveConnection = Controle
sqlControle = "select * from snmp"
rstControle.Open sqlControle, , adOpenKeyset, adLockOptimistic 'adLockReadOnly
```

Figura 16 – Criação da conexão com o banco de dados.

São iniciadas as coletas SNMP. Primeiro, os dados de disco são coletados. Em seguida, são executados os cálculos para chegar ao valor de espaço em disco utilizado, livre e a capacidade total do disco.

Inicia-se a verificação do status dos serviços, também através de coletas SNMP.

```

comando = "snmpwalk -c " & comunidade & " -v 2c " & ip & " " & oid & " > "
& Chr(34) & caminho & Chr(34)
Shell ("cmd /k " & comando), vbHide
Sleep 2000 'Necessario pois o snmpwalk demora para retornar os dados

If Ler <> vbNullString Then
    strAtrib = Split(Ler, "STRING: ") (1)
    If strAtrib Like retorno Then
        Form1.LBRESULT(intLb) = "Ativo"
        LBRESULT(intLb).ForeColor = &H0&
        snmpServices = "Ativo"
    Else
        Form1.LBRESULT(intLb) = "Sem Resposta"
        LBRESULT(intLb).ForeColor = &HFF&
        snmpServices = "Sem Resposta"
    End If
Else
    Form1.LBRESULT(intLb) = "Parado"
    LBRESULT(intLb).ForeColor = &HFF&
    snmpServices = "Parado"
End If

Exit Function

```

Figura 17 – Verificação do status dos serviços.

Caso ocorra algum erro durante a execução do programa existe uma sub-rotina responsável pela gravação do erro detectado no arquivo de log e no banco de dados.

```

Public Sub GravarLOG(ByVal strMensagem As String)
    Open App.Path & "\log.txt" For Append As #1
        Write #1, Now & " " & strMensagem
    Close #1
    'End
End Sub

```

Figura 18 – Sub-rotina responsável pela gravação no log.

Finalmente, é gerado o executável responsável por gerenciar o servidor alvo.

#### 4.3.2 APLICATIVO SNMP CLIENTE

No desenvolvimento do aplicativo SNMP Cliente foi utilizado a ferramenta de desenvolvimento Visual Basic 6.0. Foi criado um projeto padrão, selecionando a opção “Standard EXE”, adicionada a biblioteca “Microsoft ActiveX Data Objects 2.8 Library” (MDAC) e o componente “Microsoft Hierarchical FlexGrid Control 6.0 (SP4)”.

Em seguida, é criada a interface do programa *SNMP CLIENTE*.

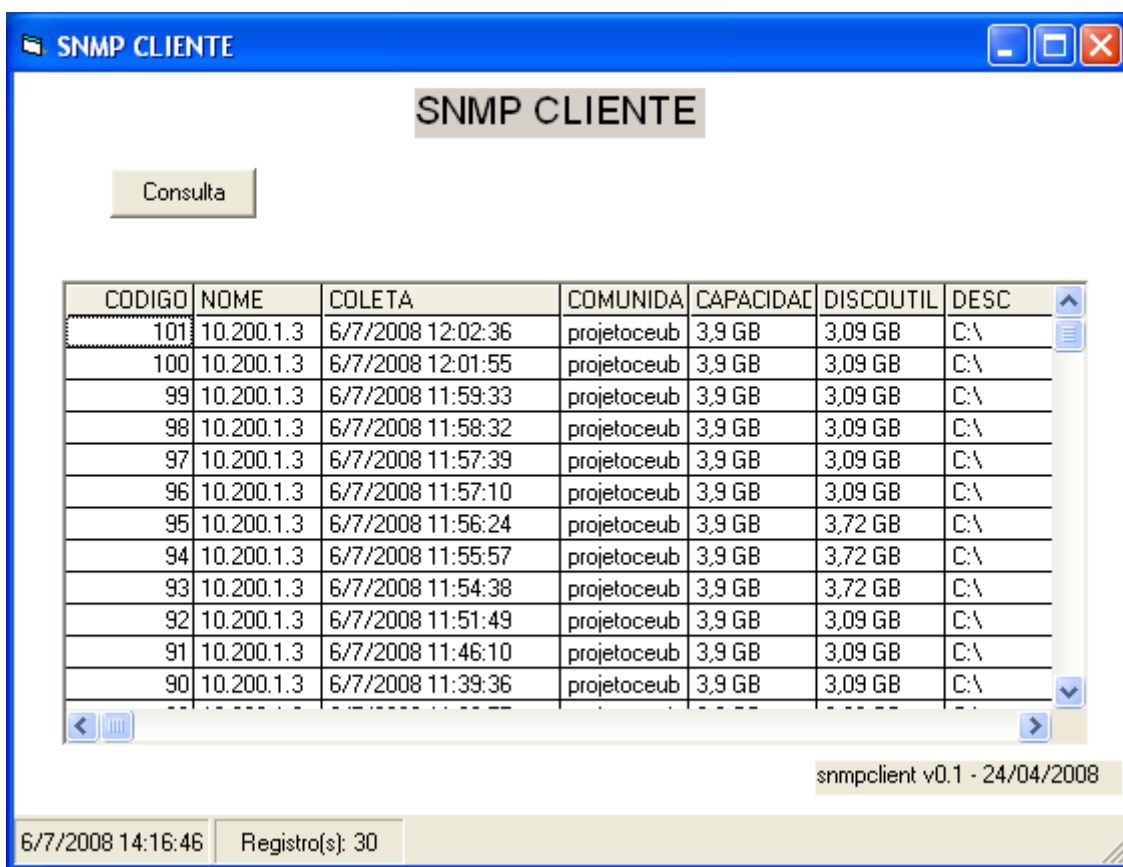


Figura 19 – Interface do programa SNMP Cliente.

A sub-rotina *snmp\_client* é definida. Está rotina é responsável pela execução do programa SNMP Cliente.

Cria-se a conexão com o banco de dados através do arquivo *.udl*. Este arquivo deverá ser configurado para consultar o arquivo de dados do banco Access remoto, utilizado pelo programa principal *SNMP COLETA*.

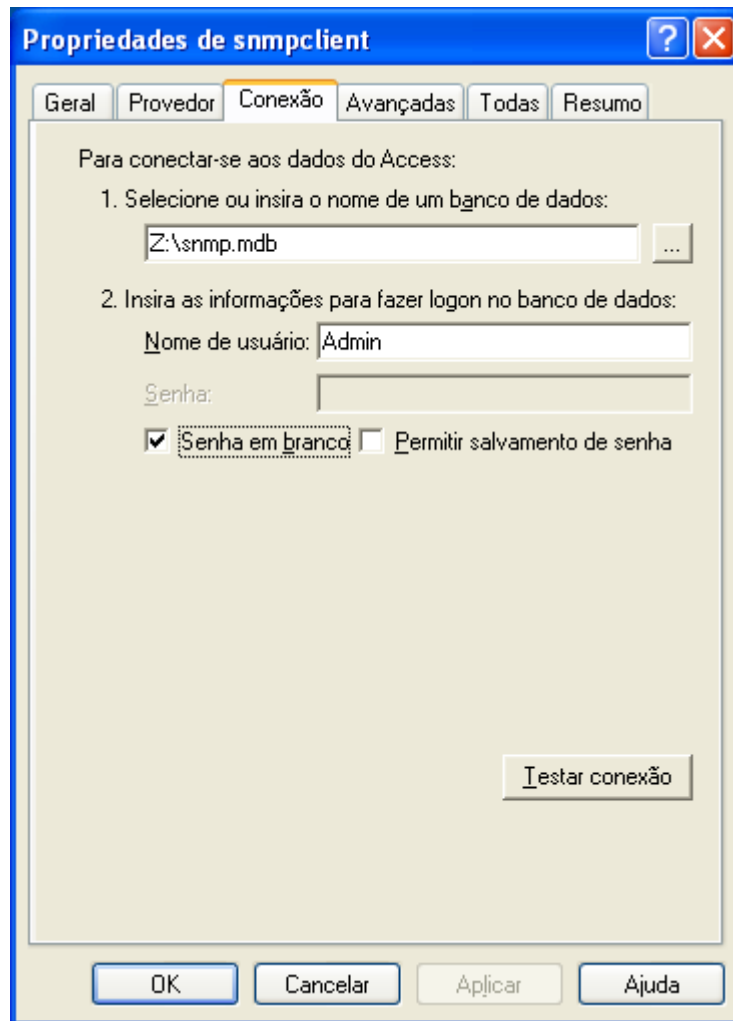


Figura 20 – Configuração do arquivo .udl.

Desenvolve-se a consulta necessária para visualização dos dados coletados. É executado o comando “select” na tabela “snmp” da base de dados Access, ordenando os dados de forma descendente. Em seguida, a consulta realizada é atribuída ao “GRID.DataSource” que será disponibilizado no interface do programa SNMP Cliente, em forma de tabela de dados, sempre que requisitado pelo usuário.

```
Controle.Open
Set rstControle.ActiveConnection = Controle
rstControle.Open "select * from snmp order by coleta desc", , adOpenDynamic, adLockOptimistic
If Not rstControle.EOF Then
    Set GRID.DataSource = rstControle
    StatusBar1.Panels(2).Text = "Registro(s): " & GRID.Rows - 1
End If
```

Figura 21 – Consulta e disponibiliza os dados.

Por fim, é criado o executável SNMP CLIENTE que será executado na estação remota com sistema operacional Windows XP SP2 instalado.

#### 4.3.3 APLICATIVO SNMP RESOLVE

O aplicativo SNMP Resolve também foi desenvolvido utilizando o VB 6.0. Este aplicativo tem por finalidade auxiliar os técnicos na tarefa de responder a um chamado e indicar a solução.

Foi criada uma rotina que ao pressionar o botão “Consulta” é feita uma pesquisa na base de dados por ocorrências que ainda estão pendentes de uma solução. Caso exista alguma ocorrência, esta pendência será mostrada na tela, conforme podemos observar na figura 22.



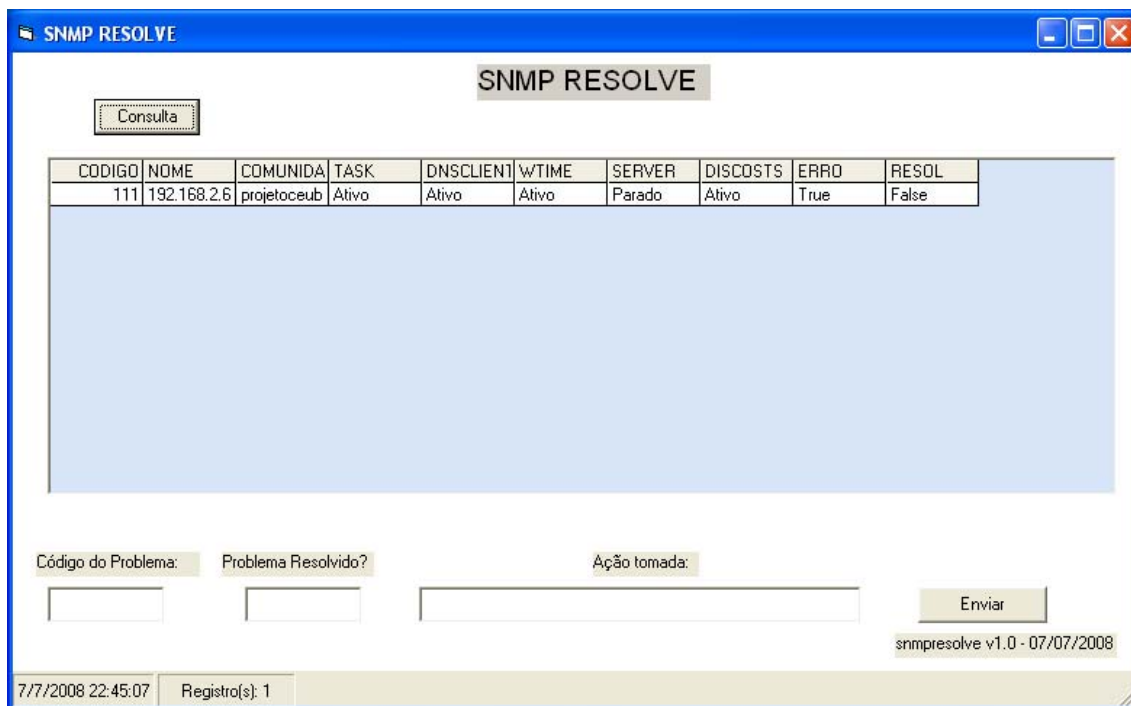


Figura 22 – Interface do programa SNMP Resolve.

Posteriormente, foram criados os campos a seguir. Após o correto preenchimento desses campos, o botão “Enviar” deverá ser pressionado para inserir os dados no banco e atualizar a ocorrência. Foi criada a mensagem “Dados inseridos com sucesso” para verificar se o processo de atualização foi bem sucedido.

- “Código do Problema”: o operador deverá preencher nesse campo o código da ocorrência que está pendente.
- “Problema Resolvido?”: o operador deverá indicar se o problema existente foi solucionado. Este campo deverá ser preenchido com “sim” ou “não”.
- “Ação”: Caso o problema tenha sido resolvido, qual foi a ação tomada pela operador.

Caso não existam ocorrências que ainda estão pendentes de uma solução irá aparecer a mensagem na tela: “sem ocorrências”.

## 4.4 DEMAIS FERRAMENTAS UTILIZADAS

### 4.4.1 HMAILSERVER

O aplicativo hMailServer é um servidor de email *livre* para a plataforma Windows. Este software é compatível com os protocolos de email mais utilizados atualmente: IMAP, SMTP e POP3 e ainda provê biblioteca COM que pode ser usada para integrar outras ferramentas ou scripts. Os emails são armazenados em uma base de dados, MySQL ou MS SQL, ficando a escolha do usuário deste produto.

A instalação padrão do hMailServer inclui instalação básica do banco de dados MySQL. Portanto, caso não exista uma base de dados na rede, o MySQL é instalado automaticamente durante a instalação do hMailServer.

### 4.4.2 NET-SNMP

Net-Snmp é uma solução *free* que utiliza e implementa o protocolo SNMP, compatível com as versões v1, v2c e v3. Esta solução contém uma biblioteca cliente responsável por uma gama de aplicações de linha de comando, incluindo um agente SNMP que possibilita o uso dos comandos snmpget, snmpwalk, snmpgetnext dentre outros.

### 4.4.3 BMAIL

Bmail é um executável, *freeware*, invocado por linha de comando, responsável pelo envio de emails utilizando o protocolo SMTP. Este programa permite automatizar rotinas de envio de email contendo arquivos de logs, dados ou mensagens de erros em computadores da plataforma Windows, inclusive é permitido incluir arquivos *.txt* no corpo do email.

### 4.4.4 BANCO DE DADOS MS ACCESS 2003

A base de dados Microsoft Access, que utiliza a extensão .mdb, é um banco de dados relacional completo. Ele prove todas as ferramentas necessárias para definição de dados, manipulação de dados, e controle de dados importantes para gerenciar grande volume de informação. [VIESCAS, 2004].

Para criar o banco de dados utilizando o RDBMS Access é necessário abrir o Microsoft Access, criar um novo banco de dados, selecionar a opção “Banco de dados em branco...” e definir o local onde será salvo o arquivo .mdb e clicar no botão salvar. O banco de dados foi criado.

Em seguida, será criada a tabela onde serão inseridos os dados das coletas. Esta mesma tabela será consultada pelo programa cliente, SNMP Cliente, para visualizar as verificações realizadas. Selecionar a opção “Criar tabela no modo Design” e definir o nome dos campos e o tipo de dados.

A tabela criada “snmp” contém os campos e significados listados na tabela abaixo.

Tabela 4 – Tabela SNMP.

| Campo             | Tipo           | Significado                                    |
|-------------------|----------------|--|
| <b>Código</b>     | Auto-Numeração | É incrementado a cada novo conjunto de dados   |
| <b>Nome</b>       | Texto          | Contém o endereço IP do objeto gerenciado      |
| <b>Comunidade</b> | Texto          | O nome da comunidade utilizada                 |
| <b>Coleta</b>     | Data/Hora      | Data e hora da verificação                     |
| <b>Status</b>     | Texto          | Se a coleta foi realizada com sucesso ou falha |
| <b>Desc</b>       | Texto          | Descrição da unidade de disco monitorada       |
| <b>Capacidade</b> | Número         | Espaço total do disco rígido                   |
| <b>Discoutil</b>  | Número         | Espaço em uso                                  |
| <b>Discolivre</b> | Número         | Espaço livre                                   |
| <b>Percutil</b>   | Número         | Percentual de disco utilizado                  |
| <b>Perclivre</b>  | Número         | Percentual de disco livre                      |
| <b>Discosts</b>   | Texto          | Se houve ocorrência para espaço em disco       |

|                  |           |  |
|------------------|-----------|--|
| <b>Task</b>      | Texto     | Serviço <i>Task Scheduler</i> está ativo ou parado |
| <b>Server</b>    | Texto     | Serviço <i>Server</i> está ativo ou parado         |
| <b>Dnsclient</b> | Texto     | Serviço <i>DNS Client</i> está ativo ou parado     |
| <b>Wtime</b>     | Texto     | Serviço <i>Windows Time</i> está ativo ou parado   |
| <b>Erro</b>      | Sim/Não   | Foi detectado algum erro no servidor agente?       |
| <b>Resol</b>     | Sim/Não   | Foi resolvido o problema?                          |
| <b>Dataresol</b> | Data/Hora | Que horas o problema foi resolvido.                |
| <b>Descresol</b> | Texto     | Descrição da ação tomada.                          |

Após conclusão dos passos acima, a base está pronta para inserção dos dados pelo aplicativo “SNMP Coleta” e para ser consultado pelo programa “SNMP Cliente”.

#### 4.4.5 MICROSOFT DATA ACCESS COMPONENTS

Uma das características mais interessantes do VB é sua facilidade para o manejo de banco de dados que residem no servidor. Isto é possível mediante o uso de ADO (ActiveX Data Object) de uma forma fácil, rápida e com um mínimo consumo de recursos do sistema.

Microsoft Data Access Components (MDAC) 2.8 instala os mesmos componentes principais do Data Access que o Microsoft SQL Server OLE DB Provider, driver ODBC. O instalador redistribuível do MDAC 2.8 instala os mesmos componentes do núcleo do Data Access que o Microsoft Windows Server 2003. Utilizado para provar conexão entre o VB e o Access.

## 4.5 PROBLEMAS E SOLUÇÕES

Durante o desenvolvimento deste projeto alguns problemas ocorreram. Entretanto, soluções foram encontradas. Segue abaixo detalhamento:

- Problema: Erro ao tentar efetuar inserção de dados na base access.
  - Solução: Após análise detalhada do erro, verificou-se a necessidade de atualização do componente MDAC (*Microsoft Data Access Components*) para a versão mais atual, v2.8. Este componente é utilizado para possibilitar acesso a base de dados.
- Problema: Ao efetuar as coletas de espaço em disco ou status de um determinado serviço, os dados retornados eram inconsistentes.
  - Solução: Observou-se que devido à velocidade de execução do aplicativo, não era possível o programa receber o retorno da coleta e processar os dados. Logo, foi implementada uma rotina que ajusta o tempo de execução.
- Problema: Ao realizar diversos testes com a ferramenta *SNMP COLETA*, detectou-se que um grande número de processos “cmd.exe” continuavam em execução no servidor.
  - Solução: Foi adicionada uma rotina no código do programa que finaliza todos os processos “cmd.exe” depois de concluída a coleta de informações.
- Problema: Os valores calculados de espaço em disco não refletiam a realidade.
  - Solução: Utilizou-se a função ROUND com duas casas decimais.
- Problema: A interface cliente não mostrava os dados das coletadas realizadas.
  - Solução: Após pesquisar o problema, verificou-se a necessidade de utilização do componente Microsoft Hierarchical FlexGrid Control 6.0 para criação do grid na interface cliente.
- Problema: Ao tentar consultar os dados, das coletas realizadas pela interface cliente, na base de dados, localizada no servidor principal, era retornado erro na execução aplicativo.

- Solução: Foram ajustadas as permissões de acesso ao arquivo de dados no servidor principal.

## CAPÍTULO 5 – IMPLEMENTAÇÃO, TESTES E RESULTADOS

---

Com o objetivo de validar a funcionalidade das ferramentas desenvolvidas foi configurado e implementado um ambiente de testes. Este ambiente possibilitou a simulação de um cenário produtivo onde os resultados fossem avaliados de maneira concreta.

### 5.1 IMPLEMENTAÇÃO E CONFIGURAÇÃO DO AMBIENTE

Foi criado um ambiente virtual, utilizando o software Microsoft Virtual PC 2007 SP1, contendo dois servidores e uma estação cliente.

O Virtual PC é um software de virtualização gratuito que possibilita a criação de um ambiente virtual, para sistemas operacionais da família Windows, e também pode ser utilizado para emular o sistema operacional Mac OS X em hardware compatível com PowerPC. O VirtualPC foi originalmente desenvolvido por uma empresa chamada Connectix, que foi posteriormente adquirida pela Microsoft.

Os sistemas operacionais compatíveis com este aplicativo são Windows Vista, XP, Server 2008, 2000, 98 Second Edition, Server 2003, NT 4.0 Server, Mac OS X e algumas versões do OS/2 da IBM [MICROSOFT, 2007].

O hardware físico utilizado para hospedar estes computadores possui as seguintes características:

- Fabricante: Sony

- Modelo: Vaio VGN-CR160A
- Processador: Intel Core 2 Duo T7100 @ 1.80GHz
- Memória: 2 GB de RAM
- Disco Rígido: 120 GB
- Placa de Vídeo: Mobile Intel 965 Express Chipset Family 128 MB
- Placa de Rede com fio: Realtek RTL8139/810x Family Fast
- Placa de Rede sem fio: Intel Wireless WiFi Link 4965AGN
- Tecnologia Intel Centrino Duo
- Sistema Operacional: Windows XP SP2

Seguem as configurações de hardware dos computadores virtualizados:

- Servidores Gerente e Agente:
  - Sistema Operacional: Windows 2003 Server SP1.
  - Memória: 512 MB de RAM.
  - Disco Rígido: 4 GB
- Estação Cliente:
  - Sistema Operacional: Windows XP SP2.
  - Memória: 256 MB de RAM.
  - Disco Rígido: 4 GB

Após definidas as configurações dos servidores e da estação, as máquinas virtuais foram criadas utilizando o seguinte procedimento:



- Instalar o software de virtualização Virtual PC. Sempre selecionando as opções padrão.
- Executar o software Virtual PC recém instalado.
- Criar três máquinas virtuais com as configurações previamente estabelecidas.
- Uma vez criadas as máquinas virtuais, os recursos deverão ser alocados para cada servidor, seguindo o padrão definido acima: os servidores com 512 MB de RAM e 4 GB de disco rígido e a estação cliente com 256 MB de RAM e 4 GB de disco rígido.

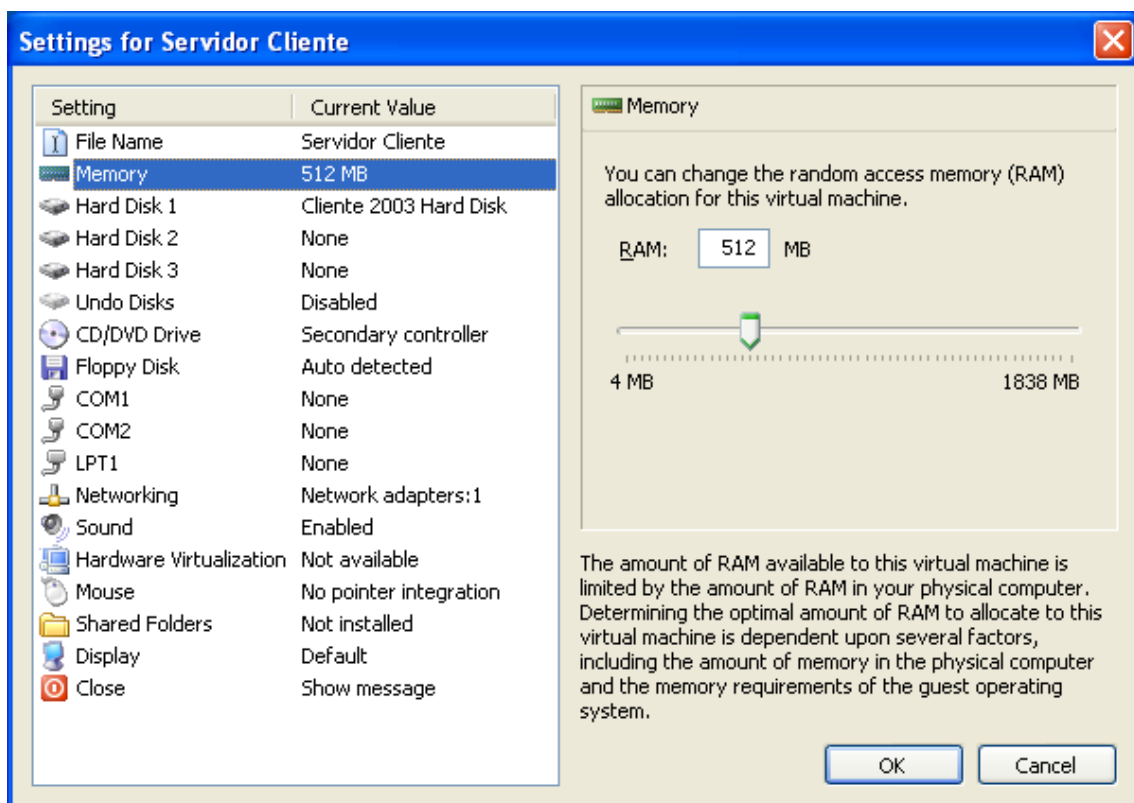


Figura 23 – Tela de configuração de recursos do servidor cliente.

- Realizar a configuração de rede dos servidores. Para o servidor gerente foi definido o IP: 10.200.1.2. Já no agente foi configurado o IP: 10.200.1.3. Por fim, foi inserido o IP: 10.200.1.4 na estação cliente.

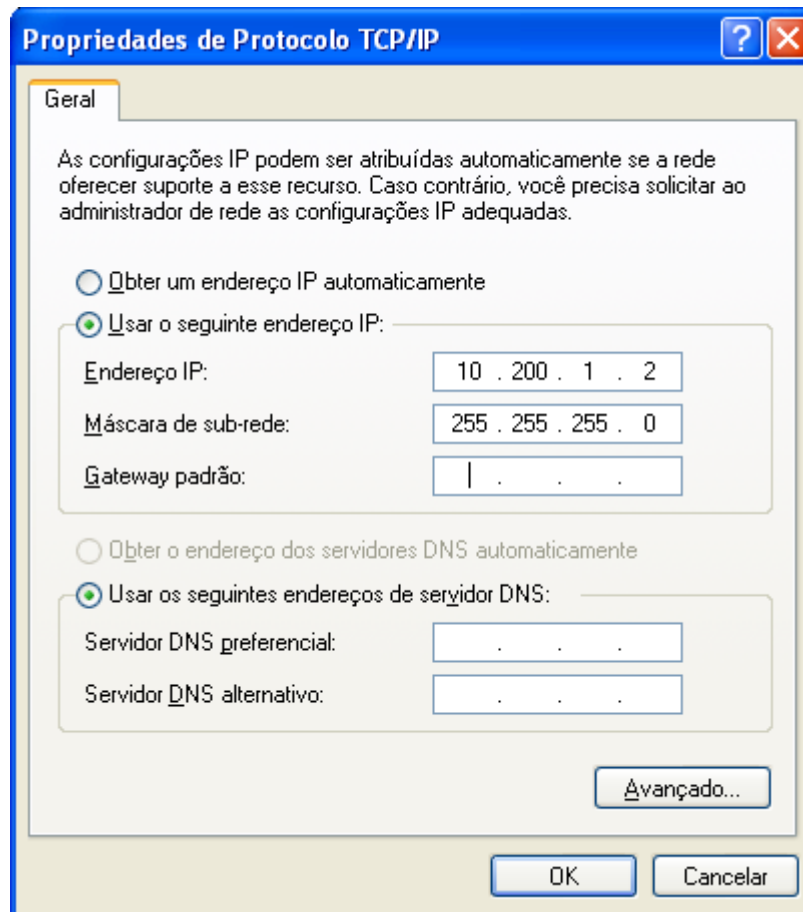


Figura 24 – Configuração de rede do servidor gerente.

- Concluída a fase de criação, alocação de recursos e configuração das máquinas virtuais, estes computadores são ligados para validar os procedimentos executados. O aplicativo VirtualPC apresenta a seguinte tela:

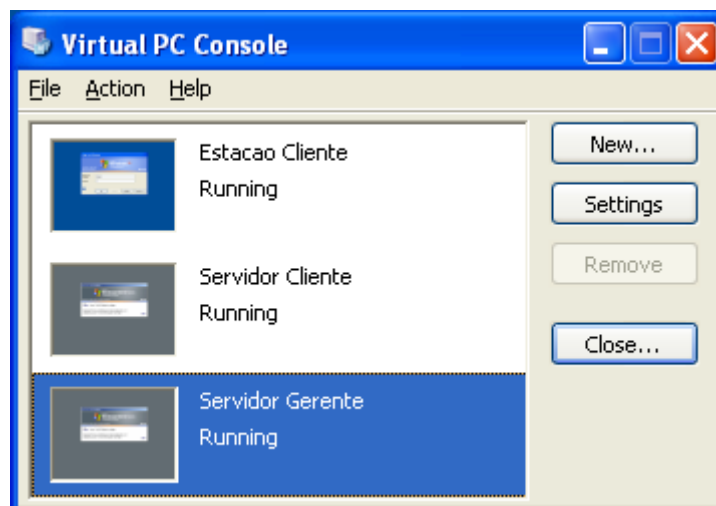


Figura 25 – Virtual PC depois de iniciadas as máquinas virtuais.

A ferramenta SNMP Coleta foi instalada no servidor gerente. Este servidor tem a função de realizar as coletas SNMP do servidor gerenciado. Neste servidor está configurado o banco de dados Access e o servidor de email utilizado para entregar as mensagens no caso de indisponibilidades.

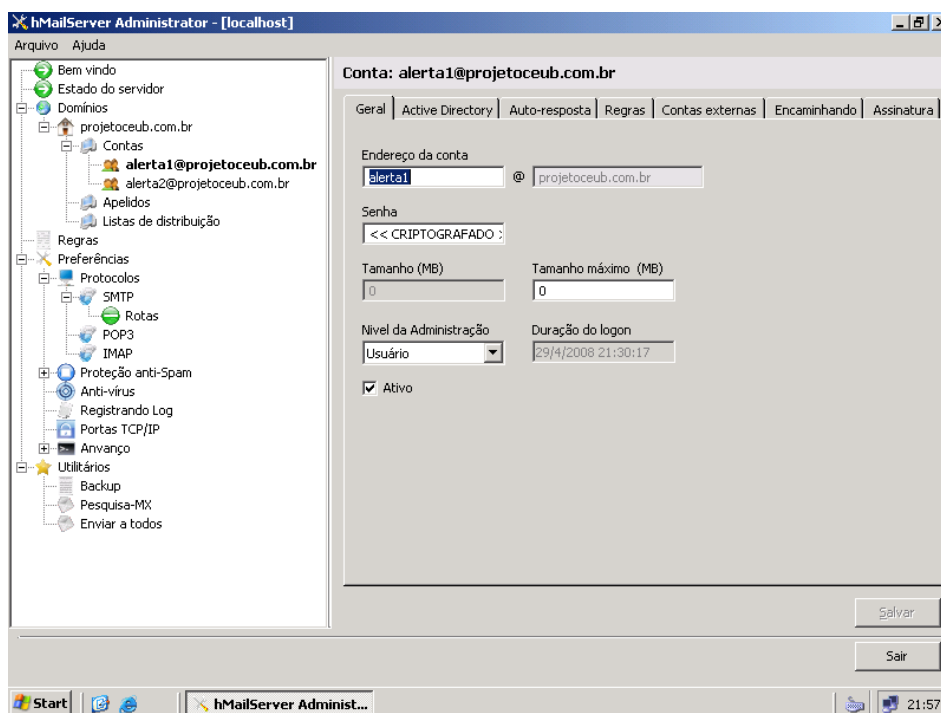


Figura 26 – Interface de gerenciamento do aplicativo de e-mail.

No servidor agente, o serviço SNMP foi configurado para aceitar requisições SNMP do servidor gerente (IP 10.200.1.2). A comunidade utilizada foi "projetoceub". Segue explicação de como efetuar as configurações acima:

- No servidor cliente acessar o aplicativo que permite iniciar, parar e configurar os serviços seguindo o caminho: "Start", "All Programs", "Administrative Tools" e "Services".
- Localizar o serviço que implementa o protocolo de gerência SNMP: "SNMP Service".

- Clicar com o botão direito sobre o item “SNMP Service” e selecionar “Propriedades”.
- Ir à aba superior chamada “Security”. Nesta aba os seguintes itens de segurança são alterados:
  - Em “Accepted community names”, adicionar a comunidade “projetoceub” com permissão de leitura.
  - Selecionar a opção “Accept SNMP packets from these hosts” e adicionar o IP do servidor gerente, 10.200.1.2. Desta forma, o servidor cliente irá responder as requisições do programa SNMP Coleta.

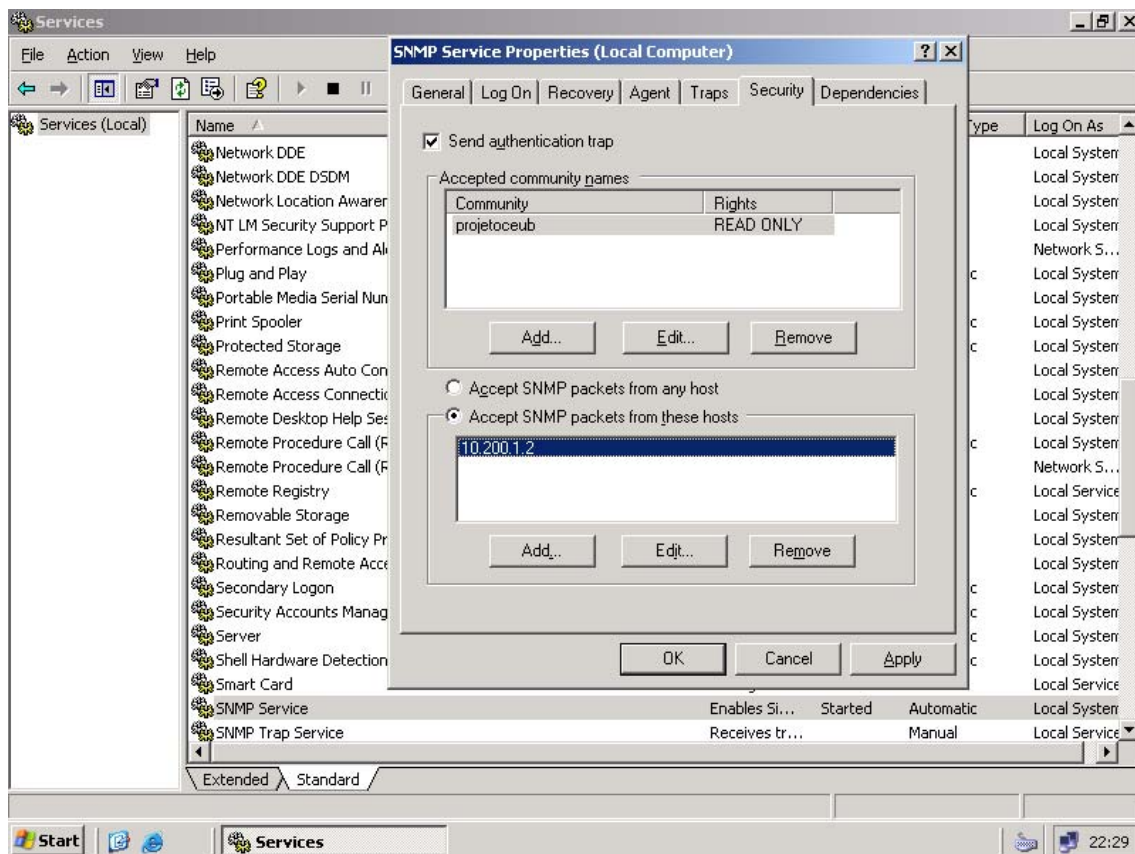


Figura 27 – Configuração do serviço “SNMP SERVICE”.

Concluída as alterações acima, o serviço “SNMP Service” é reiniciado para que as alterações efetuadas sejam carregadas.

Na estação cliente, que utiliza o sistema operacional Windows XP SP2, o aplicativo SNMP Cliente foi instalado. Outro passo importante foi mapear o caminho de rede, apontando para o servidor gerente, onde está localizado o banco de dados Access.

## 5.2 FASE DE TESTES

No ambiente virtual criado foram executados os procedimentos abaixo com o objetivo de estruturar o cenário e posteriormente homologar a ferramenta:

- Foram ligados os três computadores virtuais.
- Efetuou-se o login em cada um deles e verificou-se que todos estavam configurados na mesma rede. Foi utilizado o comando “ping” para constatar que todos os computadores virtuais estavam se comunicando conforme esperado.
- Foi confirmado que o serviço “SNMP Service” estava sendo executado no servidor cliente.
- No servidor gerente foi verificado que o serviço “hMailServer” estava em execução. Este serviço é responsável pelo recebimento e encaminhando de mensagens via email. Outro item que requereu a atenção foi o arquivo .udl, utilizado para prover a conexão com o banco de dados Access. Este arquivo deve estar apontando para onde está localizado o arquivo .mdb do Access.
- No computador gerente, executou-se o programa SNMP Coleta. Ao executar o programa SNMP Coleta é realizado um teste via SNMP para confirmar que o servidor cliente está com o serviço SNMP iniciado e configurado corretamente.

- No aplicativo SNMP Coleta, foram preenchidos os campos IP e Comunidade com “10.200.1.3” e “projetoceub”, respectivamente.

Após concluir os passos acima, foram simulados cinco cenários com o objetivo de homologar as ferramentas desenvolvidas. Os resultados foram detalhados no próximo tópico:

- 1º. Cenário: Espaço em disco abaixo de 90% de utilização e serviços verificados em execução.
- 2º. Cenário: Espaço em disco acima de 90% de utilização e serviços verificados em execução.
- 3º. Cenário: Espaço em disco abaixo de 90% de utilização e serviços verificados parados.
- 4º. Cenário: Espaço em disco acima de 90% de utilização e serviços verificados parados.

| Cenários           | Espaço em Disco | Serviços verificados |
|--------------------|-----------------|----------------------|
| <b>1º. Cenário</b> | Abaixo de 90%   | Em execução          |
| <b>2º. Cenário</b> | Acima de 90%    | Em execução          |
| <b>3º. Cenário</b> | Abaixo de 90%   | Parados              |
| <b>4º. Cenário</b> | Acima de 90%    | Parados              |

Tabela 5 – Cenários simulados.

- 5º. Cenário: Este último cenário criado tem características diferentes dos cenários acima. O objetivo foi simular um erro no programa SNMP Coleta. Para isso, inseriu-se uma comunidade diferente da configurada no servidor cliente. Em

vez de colocar a comunidade “projetoceub”, foi utilizada a comunidade “public”.

### 5.3 RESULTADOS

Com todo o ambiente definido e devidamente configurado, o projeto desenvolvido gerou os seguintes resultados nos cinco cenários citados acima:

- **1º. Cenário - Espaço em disco abaixo de 90% e serviços verificados em execução:**

O aplicativo SNMP Coleta executou corretamente, as informações requisitadas foram recebidas e inseridas no banco de dados.

Nenhum alerta via email foi gerado, uma vez que o servidor cliente não apresentou nenhuma ocorrência.

O tempo de execução registrado foi 15 segundos.

- **2º. Cenário - Espaço em disco acima de 90% e serviços verificados em execução:**

O aplicativo SNMP Coleta executou corretamente, as informações requisitadas foram recebidas e inseridas no banco de dados.

Foram gerados dois alertas por email: um para a equipe responsável e outro para o gerente do site, uma vez que o servidor cliente apresentou alto consumo de espaço em disco, excedendo 90% de utilização. Estes dois alertas foram recebidos pelo cliente de email utilizado.

O tempo de execução registrado foi 15 segundos.

- **3º. Cenário - Espaço em disco abaixo de 90% e serviços verificados parados:**

O aplicativo SNMP Coleta executou corretamente, as informações requisitadas foram recebidas e inseridas no banco de dados.

Foram gerados oito alertas por email: quatro para a equipe responsável, um para cada serviço indisponível e outros quatro para o gerente do site, um para cada serviço indisponível, uma vez que os serviços críticos no servidor cliente estavam parados. Estes oito alertas foram recebidos pelo cliente de email utilizado.

O tempo de execução registrado foi 15 segundos.

- **4º. Cenário - Espaço em disco acima de 90% e serviços verificados parados:**

O aplicativo SNMP Coleta executou corretamente, as informações requisitadas foram recebidas e inseridas no banco de dados.

Foram gerados dez alertas por email: cinco para a equipe responsável, um para espaço em disco e quatro para os serviços indisponíveis e outros cinco para o gerente do site, um para cada serviço indisponível e um para alertar a respeito da utilização de espaço em disco. Estes dez alertas foram recebidos pelo cliente de email utilizado.

O tempo de execução registrado foi 15 segundos.

- **5º. Cenário – Nome da comunidade errado:**

O aplicativo SNMP Coleta gerou um erro durante a sua execução, como era esperado, indicando que a comunidade inserida estava errada. Estas informações foram inseridas no banco de dados. Também foram gerados dois emails com o intuito de alertas a equipe responsável pelo servidor e o gerente do site que o agente não executou corretamente.



O tempo de execução foi de 7 segundos.

Baseado nos tempos de execução dos cenários propostos, foi observado que a quantidade de alertas enviados não interfere no tempo de execução total do aplicativo SNMP Coleta. Entretanto, caso a comunidade não seja à definida no servidor cliente, o programa SNMP Coleta retorna o erro após 7 segundos.

Segue gráfico detalhando os tempos de execução em cada cenário:

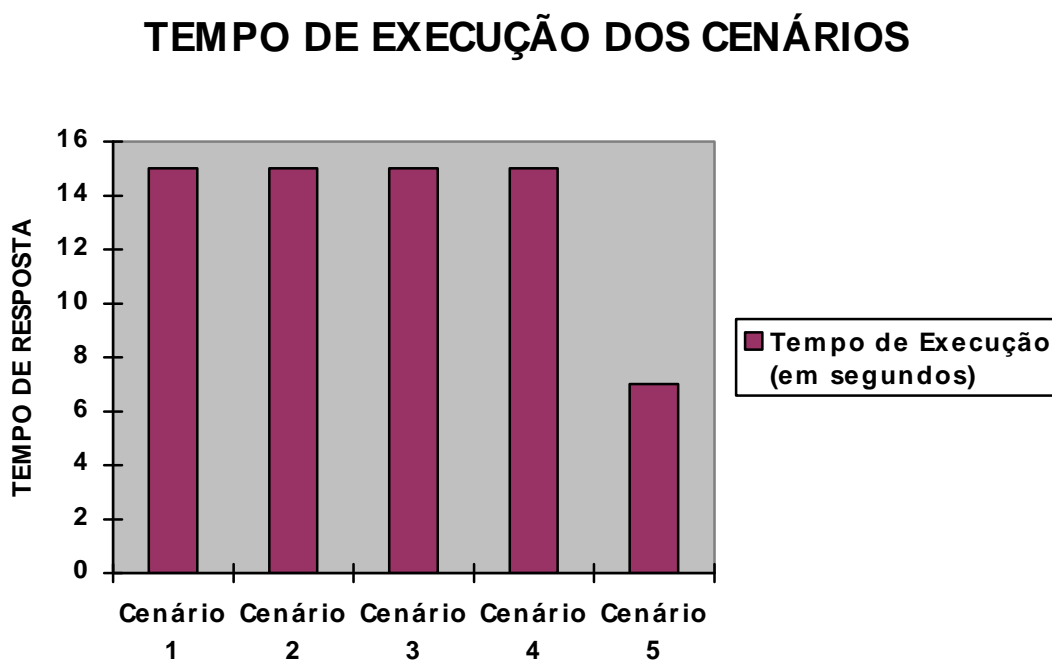


Figura 28 – Tempo de execução, em segundos, de cada cenário.

Nos cinco cenários criados a ferramenta de gerência SNMP Coleta comportou-se da maneira esperada. Todas as verificações realizadas pelo programa SNMP Coleta puderam ser visualizadas pelo aplicativo SNMP Cliente na estação cliente, que consulta a base de dados Access remotamente.

Por fim, todos os emails foram recepcionados pelo programa de email instalado na estação cliente:

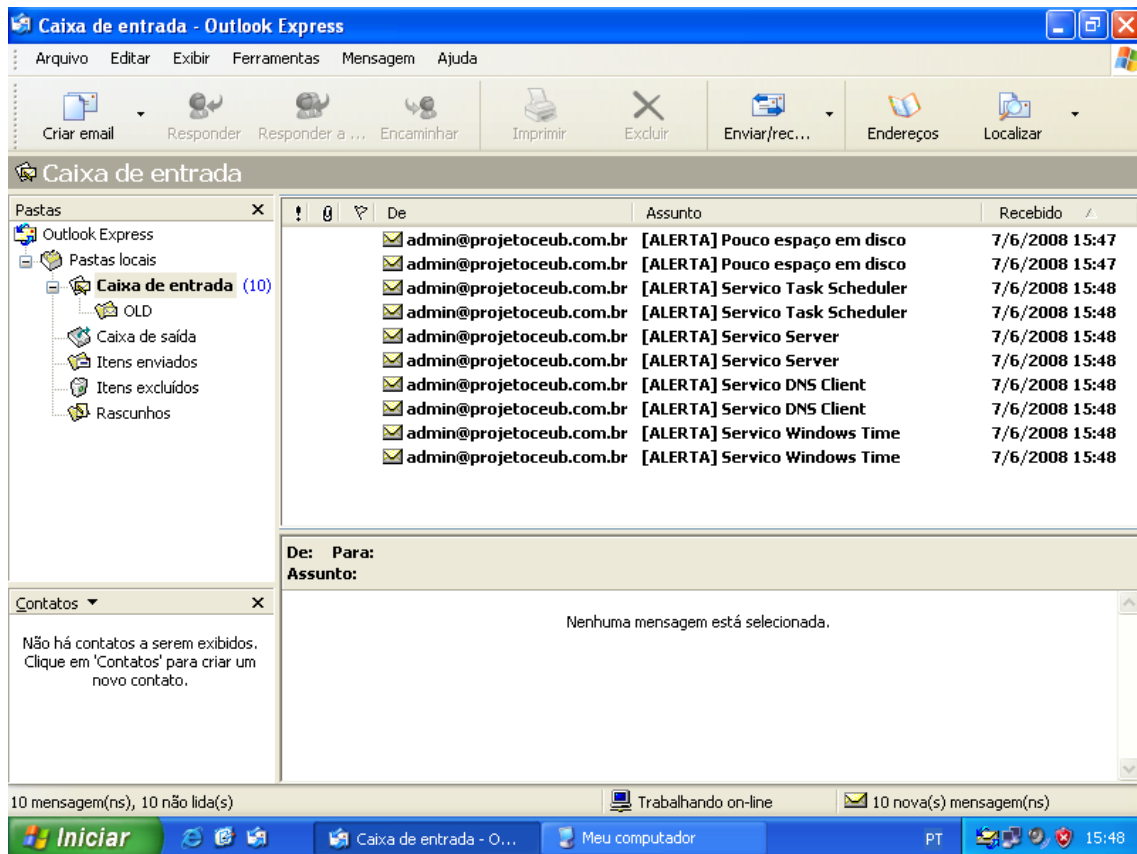


Figura 29 – E-mails recebidos conforme esperado.

## CAPÍTULO 6 – CONCLUSÕES E TRABALHOS FUTUROS

---

Este projeto teve como finalidade o desenvolvimento, implementação e configuração de uma solução de gerência que detecta pouco espaço em disco, indisponibilidade de serviços importantes e falha nos agentes de gerenciamento.

Foi desenvolvida a ferramenta SNMP Coleta, na linguagem Visual Basic, que utiliza o protocolo de gerenciamento SNMP para coletar as informações do servidor cliente e gravar esses dados na base Microsoft Access 2003. Em seguida, foi criada uma ferramenta cliente, chamada SNMP Cliente, com a funcionalidade de visualizar as coletas realizadas.

Esta solução foi implementada em um ambiente virtualizado composto por dois servidores, um gerente e outro cliente, com sistema operacional Windows 2003 Server SP1 e uma estação cliente com sistema operacional Windows XP SP2. O software de virtualização utilizado foi o Microsoft Virtual PC 2007 SP1.

Concluídas as fases de desenvolvimento e implementação iniciou-se a fase de testes e análises. Foram criados cinco cenários com o intuito de avaliar o funcionamento do aplicativo SNMP Coleta. Nestes cinco cenários a ferramenta comportou-se de forma eficaz, inclusive no cenário 5, quando foi simulado um erro de comunidade, os alertas foram gerados conforme esperado.

Além disso, a ferramenta cliente SNMP Cliente cumpriu com o seu objetivo, disponibilizando os dados coletados sempre que requisitada.

O protocolo de gerenciamento escolhido, SNMP, mostrou-se eficiente, atendendo a todos os requisitos dos aplicativos desenvolvidos. Assim como a linguagem Visual Basic.

O software Virtual PC disponibilizou o ambiente virtual necessário para este trabalho. Já os sistemas operacionais Windows 2003 Server SP1 e

Windows XP SP2 proporcionaram o suporte adequado para hospedar as ferramentas criadas.

Portanto, com as evidências acima, pode-se concluir que o objetivo deste trabalho foi atingido com sucesso.

## 6.1 TRABALHOS FUTUROS

Ao longo da elaboração e desenvolvimento do projeto, algumas sugestões para trabalhos futuros foram vislumbradas e serão apresentadas abaixo:

- Utilizando as funcionalidades do protocolo SNMP e a estrutura implementada neste projeto, gerenciar outros itens no servidor cliente, como: tráfego na interface de rede; número de conexões ativas e o tempo desde a última reinicialização do servidor.
- Implementar nos aplicativos SNMP Coleta e SNMP Cliente um controle de usuários, onde somente usuários cadastrados possam acessar estes programas.
- Reescrever as ferramentas de gerência em Java, uma linguagem portátil, independente de plataforma, capaz de executar nos ambientes Unix, Linux e Windows.

## REFERÊNCIAS BIBLIOGRÁFICAS

---

ABREU, Fabiano Rocha; Pires Herbert Domingues. **Gerência de redes**; Rio de Janeiro, 2004.

AMMIRABILE, Enriane K.; GIUDICE, Ricardo D. Detecção, Diagnóstico de Problemas e Procedimentos para uma Gerência Pró-ativa em uma rede WAN; Brasília, 2004.

BARTH, Wolfgang. Nagios: System and Network Monitoring. San Francisco, California, EUA; William Pollock, 2005.

BRISA. Sociedade Brasileira para Interconexão de Sistemas Abertos. Arquitetura de redes de computadores, 2<sup>o</sup> edição revisada e ampliada. ed. Makron Books, 2001.

CISCO. Simple Network Management Protocol (SNMP). 12 out. 2006. Disponível em: <<http://www.cisco.com/en/US/docs/internetworking/technology/handbook/SNMP.html>>. Acesso em: 15 mar. 2008.

COMER, Douglas. Internetworking with TCP/IP: Volume 1. Upper Saddle River, New Jersey, EUA; Pearson Education, 2006.

DIAS, André Felipe. O que é Gerencia de Configuração. 9 jun. 2008. Disponível em: <[http://www.pronus.eng.br/artigos\\_tutoriais/gerencia\\_configuracao/gerencia\\_configuracao.php?pagNum=0](http://www.pronus.eng.br/artigos_tutoriais/gerencia_configuracao/gerencia_configuracao.php?pagNum=0)>. Acesso em: 5 jul. 2008.

HOLME, Dan; THOMAS, Orin. Windows Server 2003 Environment. Redmond, Washington, EUA; Microsoft Press, 2006.

IBM Corporation. IBM Tivoli Monitoring Documentation: Instalation, Configuring and User's Guide. 5 jun. 2006. Disponível em: <<http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp?toc=/com.ibm.itm.doc/toc.xml>>. Acesso em: 2 fev. 2008.

ICP BRASIL. RFC (Request for Comments). 3 set 2004. Disponível em: <<https://www.icpbrasil.gov.br/duvidas/glossary/rfc-request-for-comments>>. Acesso em: 5 jul. 2008.

JOSEPHSEN, David. Building a Monitoring Infrastructure with Nagios. Boston, Massachusetts, EUA; Pearson Education, 2007.

MAURO, Douglas; SCHMIDT, Kevin. Essential SNMP. EUA; O'Reilly, 2005.

MRTG. What is MRTG?. 03 jul. 2007. Disponível em: <<http://oss.oetiker.ch/mrtg/doc/mrtg.en.html>>. Acesso em: 7 mar. 2008.

MICROSOFT. Microsoft Virtual PC 2007. 12 nov. 2007. Disponível em <<http://www.microsoft.com/windows/downloads/virtualpc/default.aspx>>. Acesso em: 5 jul. 2008.

MSDN Training. Scripting Microsoft® Windows® Management Instrumentation, course 2439A. Microsoft Official Curriculum, 2001.

NAGIOS. About Nagios. 18 mar. 2008. Disponível em: <<http://www.nagios.org/about>>. Acesso em: 12 abr. 2008.

PINTO, Ana Clara. SEGREDE: Mecanismos para Gerência de Segurança em Redes. 8 out. 2002. Disponível em: <<http://penta2.ufrgs.br/gereseg/kerberos.html>>. Acesso em: 5 jul. 2008.

SOUZA, Alysson. Gerenciamento de Redes Baseado em Agentes Móveis, Aracajú, 6 jun. 2001.

SOUZA Jr., Rafael Timóteo de. Universidade de Brasília. Administração e Gerência de Rede, Brasília, 5 abr. 2005.

STALLINGS, William. Local & Metropolitan Area Networks. Boston, EUA; Addison Wesley, 1997.

STALLINGS, William. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. Boston, EUA; Addison Wesley, 1999.

SZTAJNBERG, Alexandre. Gerenciamento de Redes. 27 mai 2003. Disponível em: <<http://www.gta.ufrj.br/~alexszt/ger/gerencia.html>>. Acesso em: 4 jul. 2008.

TANENBAUM, Andrew. Computer Networks. Upper Saddle River, New Jersey, EUA; Pearson Education, 2003.

TECHNET. Windows Server 2003 Technology Centers. 21 nov. 2004. Disponível em: <<http://www.microsoft.com/windowsserver2003/technologies/default.mspx>>. Acesso em: 25 fev. 2008.

TAROUCO, Liane Margarida Rockenbach. 20 jun. 1999. Disponível em: <[http://penta.ufrgs.br/home\\_red.htm](http://penta.ufrgs.br/home_red.htm)>. Acesso em: 7 jul. 2008.

VIESCAS, John. Microsoft Office Access 2003 Inside Out. Redmond, Washington, EUA; Microsoft Press, 2004.

## APÊNDICE A – CÓDIGOS DE PROGRAMAÇÃO

---

### **SNMP\_Coleta.vbp – Visual Basic Project da aplicação SNMP Coleta.**

```
Private Declare Sub Sleep Lib "kernel32" (ByVal lngTempo As Long) 'espera de tempo
Dim caminho As String
Dim uptime As String
Dim ip As String
Dim comunidade As String
Dim hora As String
Dim codigo As String
Dim con As New ADODB.Connection
Dim rs As New ADODB.Recordset
Dim rs2 As New ADODB.Recordset
Dim FSO As New Scripting.FileSystemObject
Dim strTexto As TextStream
Dim strAlloc As String
Dim strSize As String
Dim strUsed As String
Dim strDescr As String
Dim dkutil As Double
Dim dklivre As Double
Dim dkcapac As Double
Dim percutili As Double
Dim perclivre As Double

Private Sub BoTest_Click()
    Call test_snmp(TxIPtest.Text, TxComTest.Text)
End Sub

Private Sub Form_Load()
    caminho = App.Path & "\snmpget.txt"
End Sub

Function Ler() As String

    Dim Linha As String
```



```

Set strTexto = FSO.OpenTextFile(caminho, ForReading)
Linha = vbNullString
Do While Not strTexto.AtEndOfStream
    Linha = strTexto.ReadLine
Loop
strTexto.Close
Ler = Linha

End Function

Public Sub GravarLOG(ByVal strMensagem As String)
    Open App.Path & "\log.txt" For Append As #1
        Write #1, Now & " " & strMensagem
    Close #1
'End
End Sub

Public Sub test_snmp(ip As String, comunidade As String)
    On Error GoTo TrataErro

    Dim Controle As New ADODB.Connection
    Dim rstControle As New ADODB.Recordset
    Dim sqlControle As String

    Controle.ConnectionString = "File Name=" & App.Path & "\snmpconexao.udl"
    Controle.Open
    Set rstControle.ActiveConnection = Controle
    sqlControle = "select * from snmp"
    rstControle.Open sqlControle, , adOpenKeyset, adLockOptimistic 'adLockReadOnly

    'ip = "10.192.51.28" 'Para facilitar os testes
    'comunidade = "public" 'Para facilitar os testes

    If TxIPtest.Text = vbNullString Or TxComTest.Text = vbNullString Then
        MsgBox ("Favor inserir o IP/Hostname e Comunidade")
    End
End If

If TxIPtest.Text = "10.200.1.3" Then

```

```
Else
    MsgBox ("Favor inserir o IP/Hostname do servidor cliente (10.200.1.3)")
End
End If
```

```
rstControle.AddNew
```

```
Form1.LBRESULT(9) = ip
rstControle!NOME = ip
Form1.LBRESULT(10) = comunidade
rstControle!comunidade = comunidade
Form1.LBRESULT(11) = Now
rstControle!COLETA = Now
```

```
'Coleta dados para o calculo de espaco em disco
```

```
strAlloc = funAlloc(ip, comunidade) 'StorageAllocation
strSize = funSize(ip, comunidade) 'Size
strUsed = funUsed(ip, comunidade) 'Used
```

```
'Coleta a letra da unidade alvo
```

```
rstControle!Desc = funDescr(ip, comunidade) 'Descricao da unidade de disco
```

```
'Aqui sao realizados os calculos de espaco em disco
```

```
'Calculo de espaco em disco utilizado
dkutil = ((Val(strUsed) * Val(strAlloc)) / 1048576)
dkutil = Round((dkutil / 1024), 2)
Form1.LBRESULT(4) = dkutil & " GB"
rstControle!DISCOUTIL = dkutil & " GB"
```

```
'Calculo de espaco em disco livre
dklivre = (((Val(strSize) - Val(strUsed)) * Val(strAlloc)) / 1048576)
dklivre = Round((dklivre / 1024), 2)
Form1.LBRESULT(5) = dklivre & " GB"
rstControle!DISCOLIVRE = dklivre & " GB"
```

```
'Calculo da capacidade total do disco
dkcapac = ((Val(strSize) * Val(strAlloc)) / 1048576)
```

```
dkcapac = Round((dkcapac / 1024), 2)
Form1.LBRESULT(6) = dkcapac & " GB"
rstControle!CAPACIDADE = dkcapac & " GB"
```

```
'Calculo do percentual de disco utilizado
percutili = dkutil / dkcapac * 100
Form1.LBRESULT(7) = Round(percutili, 2) & " %"
rstControle!PERCUTIL = Round(percutili, 2) & " %"
```

```
'Calculo do percentual de disco livre
perclivre = dklivre / dkcapac * 100
Form1.LBRESULT(8) = Round(perclivre, 2) & " %"
rstControle!perclivre = Round(perclivre, 2) & " %"
```

```
'Condicao para o envio de email informando a indisponibilidade
```

```
  If percutili > 90 Then
    LBRESULT(7).ForeColor = &HFF&
    LBRESULT(8).ForeColor = &HFF&
    rstControle!email = "1" 'Email enviado
    discoalerta1 = "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t
alerta1@projetoceub.com.br -b " & Chr(34) & "ALERTA: Foi detectado pouco espaco em disco
na unidade C:" & Chr(34) & " -a " & Chr(34) & "[ALERTA] Pouco espaco em disco" & Chr(34)
    discoalerta2 = "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t
alerta2@projetoceub.com.br -b " & Chr(34) & "Sr. Gerente foi detectado pouco espaco em
disco na unidade C:" & Chr(34) & " -a " & Chr(34) & "[ALERTA] Pouco espaco em disco" &
Chr(34)
    Shell ("cmd /k " & discoalerta1), vbHide
    Shell ("cmd /k " & discoalerta2), vbHide
    'Shell ("bmail.exe -s 10.192.1.4 -f admin@projetoceub.com.br -t
alerta1@projetoceub.com.br -b 'ALERTA: Foi detectado pouco espaco em disco na unidade C:
' -a '[ALERTA] Pouco espaco em disco' "), vbHide
  Else
    LBRESULT(7).ForeColor = &H0
    LBRESULT(8).ForeColor = &H0
    rstControle!email = "0" 'Email nao necessario

  End If
```

```
'Verifica servicos e manda email
```

'Servico Task Scheduler

tasksrv = snmpServices(".1.3.6.1.4.1.77.1.2.3.1.1.14.84", 12, "\*Task\*", ip, comunidade)

rstControle!task = tasksrv

If tasksrv = "Sem Resposta" Or tasksrv = "Parado" Then

taskalerta1 = "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t  
alerta1@projetoceub.com.br -b " & Chr(34) & "ALERTA: Servico Task Scheduler está parado."  
& Chr(34) & " -a " & Chr(34) & "[ALERTA] Servico Task Scheduler" & Chr(34)

taskalerta2 = "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t  
alerta2@projetoceub.com.br -b " & Chr(34) & "Sr. Gerente, o servico Task Scheduler está  
parado." & Chr(34) & " -a " & Chr(34) & "[ALERTA] Servico Task Scheduler" & Chr(34)

Shell ("cmd /k " & taskalerta1), vbHide

Shell ("cmd /k " & taskalerta2), vbHide

End If

'Servico Server

serversrv = snmpServices(".1.3.6.1.4.1.77.1.2.3.1.1.6.83", 13, "\*Server\*", ip, comunidade)

rstControle!server = serversrv

If serversrv = "Sem Resposta" Or serversrv = "Parado" Then

serveralerta1 = "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t  
alerta1@projetoceub.com.br -b " & Chr(34) & "ALERTA: Servico Server está parado." & Chr(34)  
& " -a " & Chr(34) & "[ALERTA] Servico Server" & Chr(34)

serveralerta2 = "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t  
alerta2@projetoceub.com.br -b " & Chr(34) & "Sr. Gerente, o servico Server está parado." &  
Chr(34) & " -a " & Chr(34) & "[ALERTA] Servico Server" & Chr(34)

Shell ("cmd /k " & serveralerta1), vbHide

Shell ("cmd /k " & serveralerta2), vbHide

End If

'Servico DNS Client

dnssrv = snmpServices(".1.3.6.1.4.1.77.1.2.3.1.1.10.68", 14, "\*DNS\*", ip, comunidade)

rstControle!dnsclient = dnssrv

If dnssrv = "Sem Resposta" Or dnssrv = "Parado" Then

dnسالerta1 = "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t  
alerta1@projetoceub.com.br -b " & Chr(34) & "ALERTA: Servico DNS Client está parado." &  
Chr(34) & " -a " & Chr(34) & "[ALERTA] Servico DNS Client" & Chr(34)

dnسالerta2 = "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t  
alerta2@projetoceub.com.br -b " & Chr(34) & "Sr. Gerente, o servico DNS Client está parado."  
& Chr(34) & " -a " & Chr(34) & "[ALERTA] Servico DNS Client" & Chr(34)

Shell ("cmd /k " & dnسالerta1), vbHide

Shell ("cmd /k " & dnسالerta2), vbHide

End If

'Servico Windows Time

wtimesrv = snmpServices(".1.3.6.1.4.1.77.1.2.3.1.1.12.87", 15, "\*Windows\*", ip, comunidade)

rstControle!wtime = wtimesrv

If wtimesrv = "Sem Resposta" Or wtimesrv = "Parado" Then

    wtimealerta1 = "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t

    alerta1@projetoceub.com.br -b " & Chr(34) & "ALERTA: Servico Windows Time está parado." &  
    Chr(34) & " -a " & Chr(34) & "[ALERTA] Servico Windows Time" & Chr(34)

    wtimealerta2 = "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t

    alerta2@projetoceub.com.br -b " & Chr(34) & "Sr. Gerente, o servico Windows Time está  
    parado." & Chr(34) & " -a " & Chr(34) & "[ALERTA] Servico Windows Time" & Chr(34)

    Shell ("cmd /k " & wtimealerta1), vbHide

    Shell ("cmd /k " & wtimealerta2), vbHide

End If

Shell ("taskkill /F /IM cmd.exe"), vbHide 'finalizar os prompt de comando em S.O Windows

XP e Windows 2003 Server

'Shell ("kill -f cmd.exe"), vbHide 'finalizar os prompt de comando em Windows 2000

GravarLOG ("Coleta Realizada com Sucesso")

Form1.LBRESULT(16) = "Coleta Realizada com Sucesso"

rstControle!Status = "Coleta Realizada com Sucesso"

rstControle.Update

rstControle.Close

Controle.Close

Form1.Show

Exit Sub

'Tratamento de erro

TrataErro:

    GravarLOG (Err.Number & " " & Err.Description)

    Form1.LBRESULT(16) = "Erro: Comunidade não aceita."

    Shell ("cmd /k " & "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t

    alerta1@projetoceub.com.br -b " & Chr(34) & "ALERTA: Agente de monitoracao com  
    problemas: Comunidade nao aceita." & Chr(34) & " -a " & Chr(34) & "[ALERTA] Agente de  
    monitoracao com problemas." & Chr(34)), vbHide

```
Shell ("cmd /k " & "bmail.exe -s 10.200.1.2 -h -f admin@projetoceub.com.br -t  
alerta2@projetoceub.com.br -b " & Chr(34) & "Sr. Gerente, agente de monitoracao com  
problemas: Comunidade nao aceita." & Chr(34) & " -a " & Chr(34) & "[ALERTA] Agente de  
monitoracao com problemas." & Chr(34)), vbHide
```

```
rstControle!Status = "Erro: Comunidade não aceita."
```

```
rstControle.Update
```

```
rstControle.Close
```

```
Controle.Close
```

```
Exit Sub
```

```
End Sub
```

```
'Funcao utilizada para coletar o tamanho dos blocos
```

```
Public Function funAlloc(ByVal ip As String, ByVal comunidade As String) As String
```

```
On Error GoTo TrataErro
```

```
comando = "snmpget -c " & comunidade & " -v 2c " & ip & " " & ".1.3.6.1.2.1.25.2.3.1.4.2" & "  
> " & Chr(34) & caminho & Chr(34)
```

```
Shell ("cmd /k " & comando), vbHide
```

```
Sleep 2000 'Necessario pois o snmpget demora para retornar os dados
```

```
If Ler <> vbNullString Then
```

```
funAlloc = Left(Split(Ler, "INTEGER: ")(1), 4)
```

```
Form1.LBRESULT(0) = funAlloc & " bytes"
```

```
Else
```

```
Form1.LBRESULT(0) = "Sem resposta"
```

```
funAlloc = "Sem resposta"
```

```
End If
```

```
Exit Function
```

```
'Tratamento de erro
```

```
TrataErro:
```

```
GravarLOG (Err.Number & " " & Err.Description)
```

```
rstControle!Status = "Erro: " & Err.Number & " - " & Err.Description
```

```
rstControle.Update
```

```
rstControle.Close
```

```
Controle.Close
```

```
Exit Function
```

End Function

'Funcao utilizada para coletar o tamanho total do disco

Public Function funSize(ByVal ip As String, ByVal comunidade As String) As String

    On Error GoTo TrataErro

    comando = "snmpget -c " & comunidade & " -v 2c " & ip & " " & ".1.3.6.1.2.1.25.2.3.1.5.2" & "  
> " & Chr(34) & caminho & Chr(34)

    Shell ("cmd /k " & comando), vbHide

    Sleep 2000 'Necessario pois o snmpget demora para retornar os dados

    If Ler <> vbNullString Then

        funSize = Split(Ler, "INTEGER: ")(1)

        Form1.LBRESULT(1) = funSize

    Else

        Form1.LBRESULT(1) = "Sem resposta"

        funSize = "Sem resposta"

    End If

Exit Function

'Tratamento de erro

TrataErro:

    GravarLOG (Err.Number & " " & Err.Description)

    rstControle!Status = "Erro: " & Err.Number & " - " & Err.Description

    rstControle.Update

    rstControle.Close

    Controle.Close

    Exit Function

End Function

'Funcao utilizada para coletar a quantidade utilizada em disco

Public Function funUsed(ByVal ip As String, ByVal comunidade As String) As String

    On Error GoTo TrataErro

    comando = "snmpget -c " & comunidade & " -v 2c " & ip & " " & ".1.3.6.1.2.1.25.2.3.1.6.2" & "  
> " & Chr(34) & caminho & Chr(34)

    Shell ("cmd /k " & comando), vbHide

    Sleep 2000 'Necessario pois o snmpget demora para retornar os dados

    If Ler <> vbNullString Then

```

    funUsed = Split(Ler, "INTEGER: ")(1)
    Form1.LBRESULT(2) = funUsed
Else
    Form1.LBRESULT(2) = "Sem resposta"
    funUsed = "Sem resposta"
End If

Exit Function

'Tratamento de erro

TrataErro:
    GravarLOG (Err.Number & " " & Err.Description)
    rstControle!Status = "Erro: " & Err.Number & " - " & Err.Description
    rstControle.Update
    rstControle.Close
    Controle.Close
    Exit Function

End Function

' Funcao utilizada para coletar a descricao do disco gerenciado
Public Function funDescr(ByVal ip As String, ByVal comunidade As String) As String
    On Error GoTo TrataErro

    Dim strDescr As String
    strDescr = vbNullString

    comando = "snmpget -c " & comunidade & " -v 2c " & ip & " " & ".1.3.6.1.2.1.25.2.3.1.3.2" & "
> " & Chr(34) & caminho & Chr(34)
    Shell ("cmd /k " & comando), vbHide
    Sleep 2000 'Necessario pois o snmpget demora para retornar os dados

    If Ler <> vbNullString Then
        strDescr = Left(Split(Ler, "STRING: ")(1), 3)
        Form1.LBRESULT(3) = strDescr
        funDescr = strDescr
    Else
        Form1.LBRESULT(3) = "Sem resposta"
        funDescr = "Sem resposta"
    End If

```



Exit Function

'Tratamento de erro

TrataErro:

```
GravarLOG (Err.Number & " " & Err.Description)
rstControle!Status = "Erro: " & Err.Number & " - " & Err.Description
rstControle.Update
rstControle.Close
Controle.Close
Exit Function
```

End Function

'Funcao utilizada para verificar o status dos servicos monitorados

```
Public Function snmpServices(ByVal oid As String, ByVal intLb As Integer, ByVal retorno As
String, ByVal ip As String, ByVal comunidade As String) As String
    On Error GoTo TrataErro
```

```
Dim strAtrib As String
strAtrib = vbNullString
```

```
comando = "snmpwalk -c " & comunidade & " -v 2c " & ip & " " & oid & " > " & Chr(34) &
caminho & Chr(34)
```

```
Shell ("cmd /k " & comando), vbHide
```

```
Sleep 2000 'Necessario pois o snmpget demora para retornar os dados
```

```
If Ler <> vbNullString Then
```

```
strAtrib = Split(Ler, "STRING: ")(1)
```

```
If strAtrib Like retorno Then
```

```
Form1.LBRESULT(intLb) = "Ativo"
```

```
LBRESULT(intLb).ForeColor = &H0&
```

```
snmpServices = "Ativo"
```

```
Else
```

```
Form1.LBRESULT(intLb) = "Sem Resposta"
```

```
LBRESULT(intLb).ForeColor = &HFF&
```

```
snmpServices = "Sem Resposta"
```

```
End If
```

```
Else
```

```
Form1.LBRESULT(intLb) = "Parado"
```

```
        LBRESULT(intLb).ForeColor = &HFF&
        snmpServices = "Parado"
    End If
```

```
Exit Function
```

```
'Tratamento de erro
```

```
TrataErro:
```

```
    GravarLOG (Err.Number & " " & Err.Description)
    Form1.LBRESULT(16) = "Erro: " & Err.Number & " - " & Err.Description
    rstControle!Status = "Erro: " & Err.Number & " - " & Err.Description
    rstControle.Update
    rstControle.Close
    Controle.Close
    Exit Function
```

```
End Function
```

SNMP\_Cliente.vbp – Visual Basic Project da aplicação SNMP Cliente.

```
Private Declare Sub Sleep Lib "kernel32" (ByVal lngTempo As Long) 'espera de tempo
```

```
Dim uptime As String
```

```
Dim hora As String
```

```
Dim codigo As String
```

```
Private Sub Atualiza_Click()
```

```
    Call snmp_client
```

```
End Sub
```

```
Public Sub GravarLOG(ByVal strMensagem As String)
```

```
    Open App.Path & "\log.txt" For Append As #1
```

```
        Write #1, Now & " " & strMensagem
```

```
    Close #1
```

```
    'End
```

```
End Sub
```

```
Public Sub snmp_client()
```

```
    On Error GoTo TrataErro
```

```

Dim Controle As New ADODB.Connection
Dim rstControle As New ADODB.Recordset
'Dim sqlControle As String

StatusBar1.Panels(1).Text = Now

Controle.ConnectionString = "File Name=" & App.Path & "\snmpclient.udl"

Controle.Open
Set rstControle.ActiveConnection = Controle
rstControle.Open "select * from snmp order by coleta desc", , adOpenDynamic,
adLockOptimistic
If Not rstControle.EOF Then
    Set GRID.DataSource = rstControle
    StatusBar1.Panels(2).Text = "Registro(s): " & GRID.Rows - 1
End If

rstControle.Close
Controle.Close

GravarLOG ("Dados disponibilizados com sucesso")

Form1.Show

Exit Sub

TrataErro:
    GravarLOG (Err.Number & " " & Err.Description)
    Exit Sub

End Sub

SNMP_Resolve.vbp – Visual Basic Project da aplicação SNMP Revolve.
Private Declare Sub Sleep Lib "kernel32" (ByVal lngTempo As Long) 'espera de tempo

Dim uptime As String
Dim hora As String
Dim codigo As String

```

```
Dim Controle As New ADODB.Connection
Dim rstControle As New ADODB.Recordset
```

```
Private Sub Atualiza_Click()
```

```
    Call snmp_resolve
```

```
End Sub
```

```
Public Sub Form_Load()
```

```
    Controle.ConnectionString = "File Name=" & App.Path & "\snmpclient.udl"
```

```
    StatusBar1.Panels(1).Text = Now
```

```
End Sub
```

```
'Rotina responsavel por gravar erro em arquivo de log.
```

```
Public Sub GravarLOG(ByVal strMensagem As String)
```

```
    Open App.Path & "\log.txt" For Append As #1
```

```
        Write #1, Now & " " & strMensagem
```

```
    Close #1
```

```
    'End
```

```
End Sub
```

```
Public Sub snmp_resolve()
```

```
    On Error GoTo TrataErro
```

```
    Controle.Open
```

```
    Set rstControle.ActiveConnection = Controle
```

```
    rstControle.Open "select CODIGO, NOME, COMUNIDADE, TASK, DNSCLIENT, WTIME,  
SERVER, DISCOSTS, ERRO, RESOL from snmp where ERRO = TRUE and RESOL = FALSE  
order by codigo desc", , adOpenDynamic, adLockOptimistic
```

```
    If Not rstControle.EOF Then
```

```
        Set GRID.DataSource = rstControle
```

```
        StatusBar1.Panels(2).Text = "Registro(s): " & GRID.Rows - 1
```

```
    Else
```

```
        'Set GRID.ClearStructure = "Sem ocorrências."
```

```
        MsgBox ("Sem ocorrências.")
```

```
        StatusBar1.Panels(2).Text = "Registro(s): 0"
```

```
    End If
```

```
    rstControle.Close
```

```
    Controle.Close
```

GravarLOG ("Dados disponibilizados com sucesso.")

Form1.Show

Exit Sub

TrataErro:

GravarLOG (Err.Number & " " & Err.Description)

Exit Sub

End Sub

Private Sub Command1\_Click()

On Error GoTo TrataErro

Dim sqlControle As String

Controle.Open

Set rstControle.ActiveConnection = Controle

sqlControle = "select CODIGO, NOME, COMUNIDADE, TASK, DNSCLIENT, WTIME,  
SERVER, DISCOSTS, ERRO, RESOL, DATARESOL, DESCRESOL from snmp where ERRO  
= TRUE and RESOL = FALSE and CODIGO = " & Text1(0).Text & ""

rstControle.Open sqlControle, , adOpenKeyset, adLockOptimistic 'adLockReadOnly

rstControle.Update

If Text1(1).Text = "Sim" Or Text1(1).Text = "SIM" Or Text1(1).Text = "sim" Then

rstControle!RESOL = 1

Else

rstControle!RESOL = 0

End If

rstControle!DATARESOL = Now

rstControle!DESCRESOL = Text1(2).Text

rstControle.Save

Controle.Close

```
GravarLOG ("Dados inseridos com sucesso.")  
MsgBox ("Dados inseridos com sucesso.")
```

```
Text1(0).Text = vbNullString  
Text1(1).Text = vbNullString  
Text1(2).Text = vbNullString
```

```
Exit Sub
```

```
TrataErro:
```

```
GravarLOG (Err.Number & " " & Err.Description)  
Exit Sub
```

```
End Sub
```