APLICAÇÃO PARA GESTÃO DE ACTIVOS DE REDE DO AEROPORTO DO PORTO

Jonathan Neves Garcia

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2012

Dissertação apresentada para obtenção do Grau de Mestre em Engenharia Electrotécnica e de Computadores, Área de Especialização de Telecomunicações.

Candidato: Jonathan Neves Garcia Nº 1070251, 1070251@isep.ipp.pt

Orientação científica: Paula Marques Viana do Departamento de Engenharia Electrotécnica do Instituto Superior de Engenharia do Porto. pmv@isep.ipp.pt

Empresa: ANA – Aeroportos de Portugal S.A.

Supervisão: Antonio Jorge Pinho, ajpinho@ana.pt

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

12 de Julho de 2012

Agradecimentos

Foram várias as pessoas que em maior ou menor medida deram o seu contributo com sugestões ou meros incentivos, mas a quem realmente devo os meus mais sinceros agradecimentos e que merecem estar aqui nomeados são, em primeiro lugar, aos meus pais João Manuel e Laudelina, pelo seu sacrifício e por sempre acreditarem no meu sucesso; um agradecimento muito especial à minha querida Sónia, a quem apesar de roubar imenso tempo e atenção com a elaboração deste projecto, sempre esteve ao meu lado e incentivou a continuar nos maus momentos com as suas palavras de ánimo e fé sem limites; a minha orientadora Paula Viana, por me ter dado a oportunidade de participar neste projecto e pela sua dedicação e contributo para que este seja um êxito; ao António Pinho, pelo bom acolhimento desde o primeiro dia na empresa e pela sua disposição para colaborar com os meios a seu alcance; ao Luis Silva, pelos seus bons conselhos e sempre úteis críticas construtivas; ao Gabriel Reis, pela sua paciência e prontidão para atender as minhas dúvidas e a clareza das suas sugestões; e também um muito obrigado para todas as outras pessoas com quem lidei durante estes meses na DSTIC do Porto, nomeadamente, Pedro Braga, Frederico Lacerda, Ricardo Oliveira, Rui Maia, Francisco Oliveira, Mário Coelho e Joaquim Queirós. A todos eles, muito obrigado.

Resumo

O presente trabalho enquadra-se na área das redes de computadores, fazendo referência aos protocolos e ao conjunto de equipamentos e softwares necessários para a administração, controlo e monitorização desse tipos de infra-estruturas. Para a gestão de uma rede de dados, é essencial dispor de conhecimentos e documentação de nível técnico para representar da forma mais fiel possível a configuração da rede, seguindo passo a passo a interligação entre os equipamentos existentes e oferecendo assim uma visão o mais fidedigna possível das instalações.

O protocolo SNMP é utilizado em larga escala sendo praticamente um standard para a administração de redes baseadas na tecnologia TCP/IP. Este protocolo define a comunicação entre um administrador e um agente, estabelecendo o formato e o significado das mensagens trocadas entre ambos. Tem a capacidade de suportar produtos de diferentes fabricantes, permitindo ao administrador manter uma base de dados com informações relevantes da monitorização de vários equipamentos, que pode ser consultada e analisada por softwares NMS concebidos especialmente para a gestão de redes de computadores.

O trabalho apresentado nesta dissertação teve como objectivo desenvolver uma ferramenta para apoiar à gestão da infra-estrutura de comunicações do Aeroporto Francisco Sá Carneiro que permitisse conhecer em tempo real o estado dos elementos de rede, ajudar no diagnóstico de possíveis problemas e ainda apoiar a tarefa de planeamento e expansão da rede instalada.

A ferramenta desenvolvida utiliza as potencialidades do protocolo SNMP para adquirir dados de monitorização de equipamentos de rede presentes na rede do AFSC, disponibilizando-os numa interface gráfica para facilitar a visualização dos parâmetros e alertas de funcionamento mais importantes na administração da rede.

Abstract

The work presented in this thesis fits in the area of computer networks, more precisely in the set of protocols, equipment and software required for the administration, control and monitoring of this type of infrastructure. Efficient management of a data network, requires knowledge and technical documentation to represent as faithfully as possible the network architecture and configuration, enabling a real view about the facilities.

The SNMP protocol is used on a large scale and is practically a standard for managing networks based on the TCP/IP technology. This protocol defines the communication between a manager and an agent, establishing the format and meaning of messages exchanged between them. It has the ability to support products from different manufacturers, allowing the administrator to maintain a database of relevant information from various monitoring equipments, which can be viewed and analyzed using an NMS software designed especially for the management of computer networks.

The work presented in this thesis aimed to develop a tool to support the management of the communications infrastructure in Francisco Sá Carneiro Airport, allowing real time information on the status of network elements, helping diagnosing potential problems and still support the task of planning the expansion of the installed network.

The tool developed uses the capabilities of the SNMP protocol to acquire data from monitored devices in the Airport´s network of the, making it available in a graphical interface for an easy analysis of parameters and most important alerts in the network administration.

Índice

Índice de Figuras

Índice de Tabelas

Índice de Excertos de Código

Acrónimos

AFSC – Aeroporto Francisco Sá Carneiro

API – Application Programming Interface

ASCII – American Standard Code for Information Interchange

CSS – Cascading Style Sheets

DBI – DataBase Interface

DFE – Decision Feedback Equalizer

DSTIC – Direcção de Sistemas e Tecnologias de Informação e Comunicação

HTML – Hiper Text Markup Language

JSP – Java Server Pages

LAN – Local Area Network

MAC – Media Access Control

MD5 – Message-Digest algorithm 5

MIB – Management Information Base

MPLS – Multi Protocol Label Switching

MTBF – Medium Time Between Failures

NMS – Network Management System

PDU – Protocol Data Unit

SMI – Structure of Management Information

SNMP – Simple Network Management Protocol

STP – Spanning Tree Protocol

UDP – User Datagram Protocol

UPS – Uninteruptible Power Supply

USM – User-based Security Model

VACM – View-based Access Control Model

VoIP – Voice Over Internet Protocol

WAN – Wide Area Network

1. INTRODUÇÃO

No presente relatório, descreve-se o trabalho desenvolvido no estágio associado à unidade curricular "Tese/Dissertação", passo final para a obtenção do grau de Mestre em Engenharia Electrotécnica e de Computadores, especialização de Telecomunicações.

O estágio decorreu em parceria com a empresa ANA – Aeroportos de Portugal, e teve como objectivo o desenvolvimento de um projecto na área da gestão de equipamentos de rede, que permitisse ao candidato aplicar os seus conhecimentos de base, adquirir novas competências e que fosse de real utilidade para a empresa.

1.1. ENQUADRAMENTO

Nos dias de hoje, as empresas dispõem de um vasto número de equipamentos informáticos para satisfazer as suas necessidades de armazenamento e tratamento de informação, possibilitando o fornecimento de recursos ou serviços aos seus clientes de maneira rápida e eficaz. Para tornar isto exequível, é elementar que exista uma partilha de informação entre as máquinas, estabelecendo-se assim redes de computadores que transportam dados entre si.

Para garantir o bom funcionamento das redes de computadores, foi desenvolvido um protocolo de administração para toda a gama de equipamentos e sistemas computacionais existentes, denominado por SNMP (*Simple Network Management Protocol*), que visa informar o gestor de uma rede de computadores quanto ao estado em que se encontram todos os parâmetros relacionados com cada equipamento. Esta ferramenta é de extrema utilidade para que o gestor possa ter a percepção real do estado de todos os elementos que formam a rede de computadores. Para poder moldar de forma clara e facilmente compreensível toda a informação que o SNMP transmite, existem sistemas NMS (*Network Management System*) que recolhem, tratam e organizam todos dos dados fornecidos pelo protocolo.

1.2. APRESENTAÇÃO DO LOCAL DE ESTÁGIO

A ANA Aeroportos de Portugal, como entidade pública gestora dos aeroportos portugueses, é responsável pela prestação de serviços aeroportuários de apoio à aviação civil, tendo a seu cargo a gestão de um total de dez aeroportos: quatro em Portugal continental - Lisboa (Portela), Porto (Francisco Sá Carneiro), Faro e Beja - quatro no arquipélago dos Açores - Ponta Delgada (João Paulo II), Horta, Santa Maria e Flores – e dois no arquipélago da Madeira – Funchal e Porto Santo – através da ANAM, empresa subsidiária da ANA [1] [2].

O presente estágio decorreu na delegação que a ANA Aeroportos de Portugal possui no Aeroporto Francisco Sá Carneiro, inserido na DSTIC (Direcção de Sistemas e Tecnologias de Informação e Comunicação). Esta divisão tem como principais funções o controlo e gestão das redes de dados e de voz, e a administração dos sistemas e equipamentos informáticos do aeroporto que pertencem à empresa.

A gestão dos equipamentos da rede do aeroporto está actualmente a ser realizada através de duas aplicações diferentes, o NetSight da Enterasys Networks que é usado, principalmente, como interface para realizar as configurações necessárias nos equipamentos de rede desta marca (software fornecido pelo fabricante dos mesmos), e o WhatsUp Gold da Ipswitch o qual é utilizado para realizar a monitorização em tempo real do estado de todos os equipamentos presentes na rede, sendo esta a aplicação base para a administração de toda a rede do aeroporto.

Apesar de ambas as aplicações suportarem o recurso a MIBs (*Management Information Base*), considerou-se útil realizar uma aplicação que recolhe informação de gestão dos equipamentos *Enterasys* para ser mostrada de um modo mais agregado e simples. É nesta base que foi estabelecido o objectivo principal para este estágio de desenvolver uma aplicação que tendo como base o conceito de software NMS, seja capaz de realizar monitorização de equipamentos, servindo como um complemento para a administração da rede.

1.3. OBJECTIVOS

Os objectivos traçados para este estágio são, em linhas gerais, os seguintes:

1. Estudar e compreender a arquitectura da rede de dados do aeroporto.

2. Realizar um levantamento dos softwares NMS existentes.

3. Conhecer os equipamentos de rede usados e as suas funcionalidades.

4. Listar os parâmetros de funcionamento considerados importantes nos equipamentos de rede.

5. Implementar uma solução para recolher, tratar e guardar os dados fornecidos por cada equipamento através do protocolo SNMP.

6. Desenvolver uma aplicação amigável com o utilizador para organizar e mostrar os dados recolhidos ao gestor da rede.

7. Realizar a monitorização dos equipamentos de rede recorrendo à solução desenvolvida e traçar possíveis melhoramentos futuros.

1.4. ORGANIZAÇÃO DO RELATÓRIO

Este trabalho encontra-se dividido em cinco capítulos, dos quais três são relativos a conteúdos abordados durante o decorrer do estágio. Nomeadamente, o Capítulo 2 denomina-se "Conceitos Fundamentais e Tecnologias" e nele se encontram os conceitos teóricos do protocolo SNMP e uma análise comparativa dos softwares NMS presentes na rede do AFSC com alguns exemplos de outros softwares existentes no mercado.

O Capítulo 3, com o título "Rede do Aeroporto Francisco Sá Carneiro", inclui uma descrição das redes de dados nas que as comunicações do Aeroporto se implementam. Estas são a rede WAN (*Wide Area Network*) da ANA, que liga o AFSC aos outros aeroportos geridos pela empresa, e a rede LAN (*Local Area Network*) do próprio AFSC na qual se inserem os equipamentos de rede que também se descrevem em detalhe na parte final deste capítulo.

Já no Capítulo 4, denominado como "Aplicação de Gestão Desenvolvida", apresenta-se todo o processo de criação e desenvolvimento da aplicação de monitorização de activos de rede, objectivo principal da realização do presente projecto. Aqui encontra-se a descrição detalhada dos parâmetros a monitorar e as respectivas variáveis, a arquitectura que foi idealizada para o seu funcionamento, as abordagens em termos de linguagem de programação que foram utilizadas, a estrutura da base de dados e, finalmente, a descrição da interface com o utilizador e o seu funcionamento.

No ultimo Capítulo, apresentam-se as conclusões relativas ao desenvolvimento da aplicação em estudo, incluindo também uma reflexão acerca dos possíveis melhoramentos que possam vir a ser introduzidos, com o objectivo de tornar a aplicação mais robusta.

2. CONCEITOS FUNDAMENTAIS E TECNOLOGIAS

No presente capítulo estão contemplados os conceitos teóricos e o estado actual da tecnologia inerente à gestão de redes de computadores, determinantes no desenvolvimento da aplicação em análise. É feita também uma breve exposição sobre o funcionamento do protocolo SNMP, do uso que os softwares NMS fazem deste protocolo e um levantamento da situação de mercado dos próprios sistemas NMS.

2.1. SNMP – *SIMPLE NETWORK MANAGEMENT PROTOCOL*

Este protocolo tem como principal função permitir trocas de informação de gestão e monitorização entre uma estação gestora e um ou mais agentes ligados em rede, de modo a tornar possível criar mecanismos de supervisão do funcionamento dos dispositivos e da própria rede. Assim, torna-se possível detectar eventuais problemas e facilitar a manutenção dos sistemas. A sua implementação realiza-se na camada de aplicação do modelo TCP/IP, utilizando o protocolo UDP (*User Datagram Protocol)* como meio de transporte dos grupos de mensagens do protocolo SNMP [3].

2.1.1. ELEMENTOS E ARQUITECTURA

O modelo de gestão de rede baseada em SNMP inclui os seguintes elementos:

- **Dispositivos geridos**: São os elementos da rede a monitorar. Podem ser switches, routers, servidores, computadores, UPS (*Uninterruptible Power Supply*), impressoras, entre outros.

- **Agentes**: Residem nos dispositivos geridos e são responsáveis por realizar o tratamento da informação de gestão. Recorrem a MIBs para conhecer os atributos que cada dispositivo tem definido pelo fabricante ou pelo administrador.

- **Sistemas de gestão de rede:** Executam aplicações que permitem monitorização e controlodos dispositivos geridos, baseando-se na informação que o agente SNMP de cada dispositivo fornece.

Na Figura 1 apresenta-se a arquitectura típica associada a um sistema de gestão baseado em SNMP, onde se ilustra a forma como os elementos interagem entre si, as mensagens existentes e a estrutura da pilha protocolar [3] [4] [5].

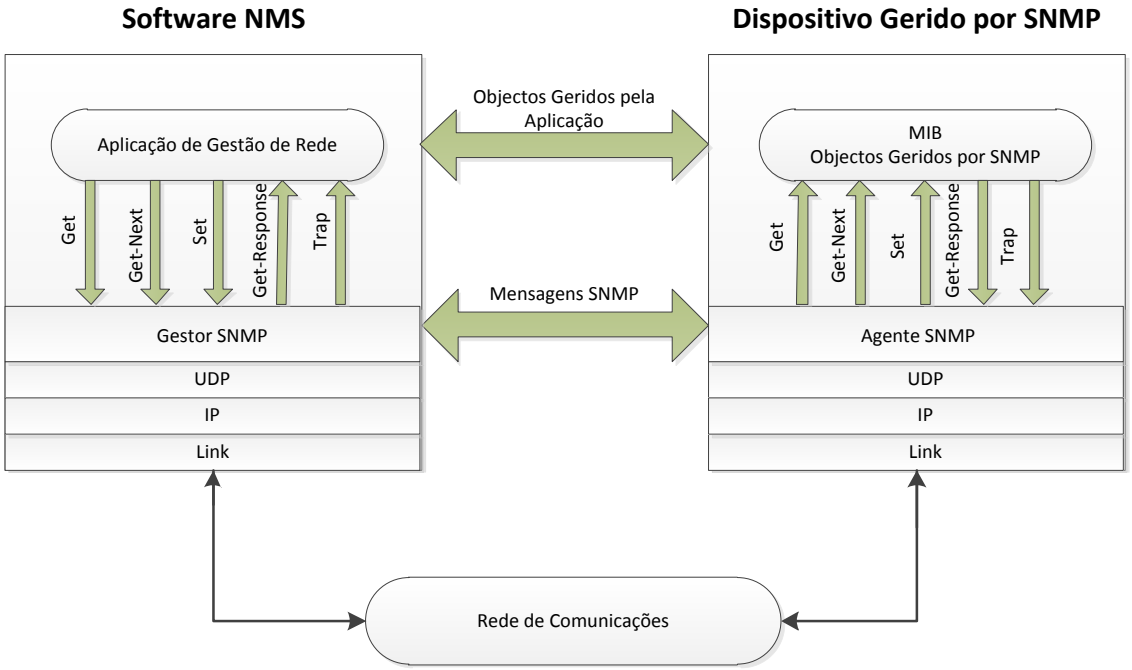


Figura 1: Arquitectura do protocolo SNMP.

2.1.2. MENSAGENS SNMP

Os dispositivos geridos respondem às solicitações do sistema NMS através de três tipos de operações diferentes contemplados no protocolo, sendo encapsuladas em mensagens SNMP. As operações são, nomeadamente, as seguintes:

- ***SET***: Operação de escrita, utiliza-se para alterar ou adjudicar valores às variáveis atribuídas de cada dispositivo a monitorar.

- ***GET***: Operação de leitura, tem como função obter os valores atribuídos às variáveis dos dispositivos.

- ***TRAP***: Operação de notificação assíncrona, usada para informar de possíveis eventos ocorridos no dispositivo a monitorar num determinado instante.

Como suporte à troca de mensagens SNMP, é utilizado o protocolo UDP (*User Datagram Protocol*), nos portos 161 e 162, entre o gestor e o agente. A utilização do mecanismo UDP garante que as operações de gestão da rede não afectam o normal funcionamento da mesma, evitando assim a utilização de mecanismos de controlo e recuperação que o protocolo TCP contempla.

As mensagens SNMP possuem três partes elementares:

- ***Version***: Informa da versão do protocolo utilizada.

- ***Community***: Indica um nome ou uma palavra-chave para a autenticação da mensagem SNMP. Por defeito existem duas comunidades, uma de escrita (private) e outra de leitura (public).

-***SNMP PDU***: Contém os dados relativos às operações solicitadas (pedidos ou respostas). As operações encapsuladas na PDU (*Protocol Data Unit*) podem ser, na versão 1 do protocolo, do tipo *GetRequest*, *GetNextRequest*, *SetRequest* e *GetResponse*, acrescentando a operação *GetBulkRequest* já na versão 2. Como podemos comprovar na Figura 2, uma mensagem SNMP subdivide-se em vários campos, nomeadamente: *PDU type* que indica o tipo de operação; *Request ID* que serve como identificador da mensagem*; Error Status* para informar se existe um erro*; Error Index* que notifica qual o índice de erros; e as *VariableBrindings* que não são mais do que uma lista com as variáveis em causa e os respectivos valores.

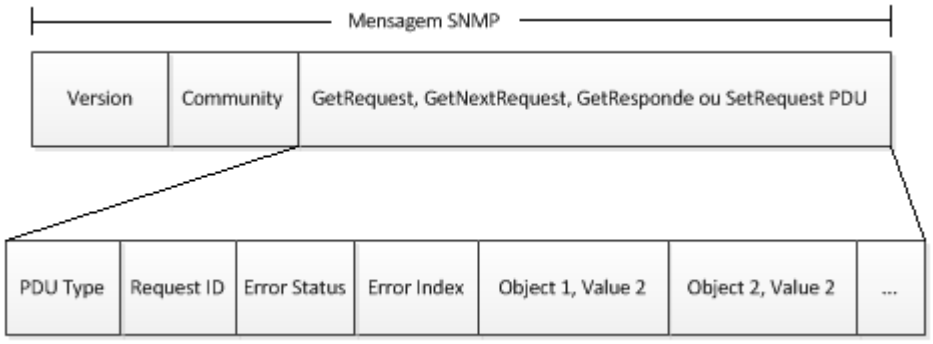


Figura 2: Estrutura da mensagem SNMP.

- Trap PDU: Este tipo de mensagem é enviada para informar de uma ocorrência inesperada num dado equipamento, possuindo uma estrutura que difere das PDUs comummente utilizadas no protocolo SNMP. Os elementos a destacar na estrutura desta mensagem, além do *PDU Type* e as *VariableBrindings* já existentes na SNMP PDU, são, como se visualiza na Figura 3, os blocos relativos a *Enterprise*, que indicam qual o elemento criador da trap; o *AgentAddress* que refere o endereço desse mesmo elemento; o *GenericTrapType* que indica qual o tipo de Trap enviada dentro dos sete possíveis (*coldStart*; *warmStart*; *linkDown*; *linkUp*; *authenticationFailure*; *egpNeighborLoss*; *enterpriseSpecific*); o *SpecificTrapType* que informa do código específico da Trap; e finalmente o campo timestamp que guarda o tempo decorrido desde a última reinicialização do elemento que gerou a Trap. [4][5][6]

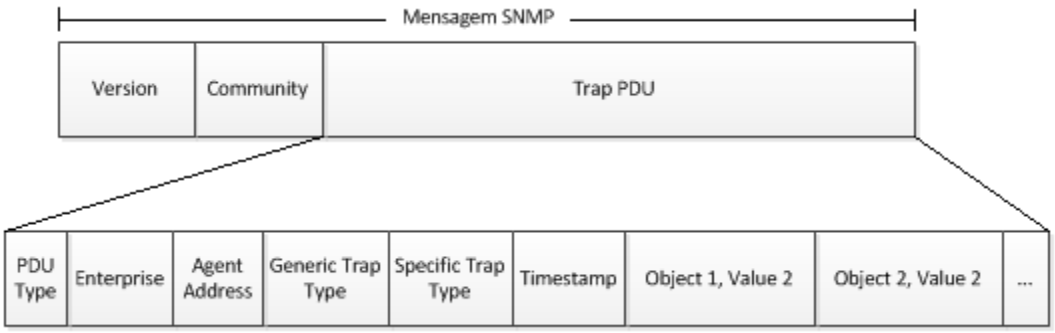


Figura 3: Estrutura da mensagem SNMP com Trap PDU.

2.1.3. VERSÕES DO PROTOCOLO

Actualmente existem três versões do protocolo SNMP:

- SNMPv1: É a primeira versão do protocolo e encontra-se definida nos RFCs 1155 e 1157.

- SNMPv2: Esta versão é uma revisão do protocolo que inclui melhoramentos no que diz respeito ao tipo de mensagens, mapeamento/transporte e elementos da estrutura MIB. Está definida nos RFCs 1901, 1905, 1906 e 2578.

- **SNMPv3**: A mais recente versão adiciona segurança e configuração remota de equipamentos ao protocolo SNMP, sendo esta versão definida pelos RFCs 3412, 3414 e 3415.

Os principais pontos fracos da versão 1 do protocolo são a falta de autenticação da origem das mensagens e a ausência de protecção destas. Como já mencionado anteriormente, SNMPv1 utiliza como mecanismo de autenticação o parâmetro *community*, só que a informação de texto nele contido é enviada sem encriptação, resultando numa carência importante de segurança nas transmissões das mensagens do protocolo.

A segunda versão do protocolo utiliza muitas das mesmas operações que o SNMPv1 possui e acrescenta duas novas, *GetBulk* e *Inform*. A operação *GetBulk* é usada pelo NMS para recuperar eficientemente grandes blocos de dados, enquanto a operação *Inform* permite a um NMS enviar *traps* a outro NMS e receber uma resposta deste último. Com estas operações, reduz-se em parte o volume de tráfego inerente à monitorização da rede.

Para resolver os problemas de segurança das versões anteriores foi desenvolvido o SNMPv3 que inclui na sua arquitectura o USM (*User-based SecurityModel*) que acrescenta protecção e encriptação em formato MD5 (*Message-Digest algorithm 5*) às mensagens SNMP e o VACM (*View-based Access Control Model*) para o controlo de acesso. Uma nova funcionalidade que esta versão inclui é a possibilidade de configurar o agente SNMP dos equipamentos utilizando o comando SET nos objectos da MIB que representam a configuração do agente. Deste modo, é possível adicionar, apagar ou modificar os parâmetros de modo local ou remoto. [6][7]

.

2.1.4. CONCEITO DE MIB

A MIB não é nada mais do que uma base de dados virtual na qual se encontram organizados de forma hierárquica todos os dados relativos aos parâmetros de gestão dos equipamentos de rede. Reside nos agentes SNMP de cada dispositivo e é acedida pelos sistemas de gestão de rede. A estrutura em árvore para a informação de gestão, convenções, sintaxe e regras para a construção das MIBs estão definidas na SMI (*Structure of Management Information*).

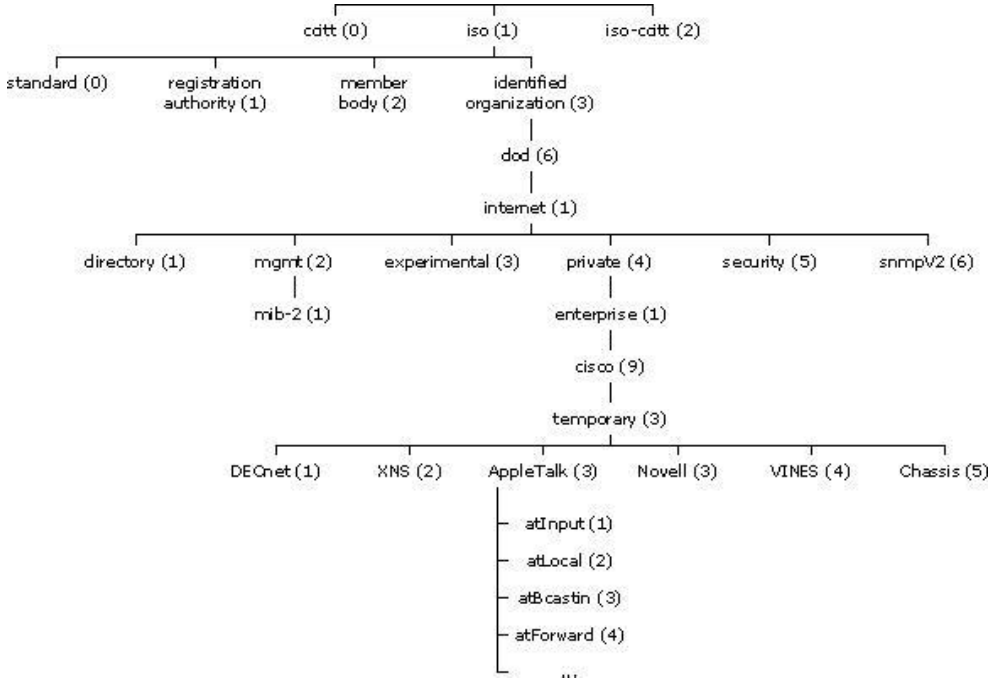


Figura 4: Estrutura hierárquica das MIBs.

Como se pode comprovar através da Figura 4, toda a informação de gestão existente nas MIBs encontra-se estruturada em árvore, ou seja, o nó inicial denomina-se por raiz e todos os outros nós que possuam ramificações são considerados como sub árvores. A existência de um nó sem nenhum tipo de subdivisão denomina-se por objecto.

Os objectos das MIBs são compostos por uma ou mais instâncias de parâmetros, que na sua maioria se definem como variáveis. Para que um agente possa seleccionar o objecto adequado dentro da árvore MIB, necessita fazer referência ao número identificador do mesmo (objectID), cujo formato consta do número de cada parâmetro da pirâmide (Ex: 1.3.6.1.4.1.9.3.3.1) ou, através dos nomes de cada um dos parâmetros de cada nível.(Para oobjectID do exemplo anterior, a sua conversão sería: *iso.org.dod.internet.private.enterprise.cisco.temporary.appletalk.atInput*). Os exemplos mencionados anteriormente fazem referência ao objecto *atInput* do grupo de MIBs apresentadas na Figura 4.

Nas MIBs podem estar definidos vários tipos de variáveis:*DisplayString* que contém uma sequência de até 256 caracteres*; Counter* para representar o número de ocorrências de um determinado evento ou processo; *Integer* quando se trata de um número inteiro positivo ou negativo; *EnumeratedValue* que atribui um campo do tipo texto a um *Counter*; *TimeTicks* para valores de tempo decorrido; *Object* que atribui um identificador para outro objecto da MIB; *Physicaladdress* quando a resposta é um endereço físico (MAC – *Media Access Control*) de um equipamento; e *String* para grandes cadeias de caracteres. [8]

2.1.5. EXEMPLO DE MIB

No Excerto 1, apresenta-se a especificação de uma variável MIB que retorna uma string na qual se descreve o tipo de equipamento que suporta os pedidos SNMP. Como podemos comprovar, a informação contida no ficheiro é do tipo ASCII (*American Standard Code for Information Interchange).* A descrição da MIB inclui os tipos de dados suportados, a identificação da versão, o tipo da variável de retorno, as permissões de manipulação do ficheiro, a indicação do órgão que desenvolveu a MIB e a descrição da sua funcionalidade. [9]

```
1- SNMPv2-MIB DEFINITIONS ::= BEGIN
2-
3- IMPORTS
4-    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
5-TimeTicks, Counter32, snmpModules, mib-2
6-        FROM SNMPv2-SMI
7-DisplayString, TestAndIncr, TimeStamp
8-
9-
10-         FROM SNMPv2-TC
11-    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
```

```
12-        FROM SNMPv2-CONF;
13-
14- snmpMIB MODULE-IDENTITY
15-    LAST-UPDATED "200210160000Z"
16-    ORGANIZATION "IETF SNMPv3 Working Group"
17-    CONTACT-INFO
18-            "WG-EMail:   snmpv3@lists.tislabs.com
19-Subscribe:  snmpv3-request@lists.tislabs.com
20-
21-             Co-Chair:   Russ Mundy
22-                         Network Associates Laboratories
23-             postal:     15204 Omega Drive, Suite 300
24-                         Rockville, MD 20850-4601
25-                         USA
26-EMail:      mundy@tislabs.com
27-             phone:      +1 301 947-7107
28- DESCRIPTION
29-            "The MIB module for SNMP entities.
30-
31-             Copyright (C) The Internet Society (2002). This
32-             version of this MIB module is part of RFC 3418;
33-see the RFC itself for full legal notices.
34-            "
35-    REVISION      "200210160000Z"
36-
37- system   OBJECT IDENTIFIER ::= { mib-2 1 }
38-
39- sysDescr OBJECT-TYPE
40-    SYNTAX      DisplayString (SIZE (0..255))
41-    MAX-ACCESS  read-only
42-    STATUS      current
43-    DESCRIPTION
44-        "A textual description of the entity.  This value should
            include the full name and version identification of
            the system's hardware type, software operating-system,
and networking software."
45-    ::= { system 1 }
46-  END
```

Excerto 1: Especificação de uma MIB.

2.1.6. NET-SNMP

Para poder realizar todas as operações que o protocolo SNMP suporta através de uma linha de comandos, foi desenvolvido o agente Net-SNMP. Este software é distribuído através da comunidade "*Open Source*" e pode ser instalado e utilizado em qualquer sistema operativo.

Além da linha de comandos, este agente disponibiliza também um "MIB browser", uma aplicação do tipo "*daemon*" para suportar a recepção de *Traps* e uma API (*Application Programming Interface*) para as linguagens de programação C e Perl, possibilitando o desenvolvimento de novas aplicações que recorram ao uso do protocolo SNMP em todas as suas vertentes.[11]

Na Tabela 1 encontra-se a listagem das operações mais comuns e respectivos comandos que o Net-SNMP suporta.

Comando	Descrição
snmptranslate	Traduz os OID das MIBs de numérico para texto.
snmpget	Realiza pedidos GET
snmpgetnext	Realiza pedidos GETNEXT
snmpbulkget	Realiza pedidos GETBULK
snmpset	Realiza pedidos SET
snmptrap	Envia menssagens TRAP ou INFORM
snmpd	Responde aos pedidos SNMP de um dado equipamento
snmptrapd	Inicia o "daemon" para escutar pedidos TRAP ou INFORM e actuar em consequência
snmptest	Comunica com um equipamento através de pedidos SNMP

Tabela 1: Listagem dos comandos mais comuns do Net-SNMP.

2.2. SOFTWARE NMS

Quanto maior é o tamanho da rede, mais complexos são os sistemas que nela se inserem, com mais aplicações e utilizadores. Desta forma torna-se necessário recorrer a mecanismos e protocolos como o SNMP que ajudam os administradores de rede a identificar possíveis falhas que levem à inutilização parcial ou total da rede.

Um sistema de gestão de rede é uma ferramenta para monitorar e controlar a rede, concebido para permitir ao gestor ter uma visão desta como um todo, mas atribuindo também especial atenção ao estado de cada elemento activo que nela se insere. Através destes sistemas é possível executar a maioria das tarefas de gestão de rede, tal como determinar o estado de funcionamento dos dispositivos, obter estatísticas, controlar as rotas e configurar as interfaces.

No âmbito do estágio fez-se uma análise dos dois softwares de gestão existentes no aeroporto, sendo eles o "*WhatsUp Gold*" da *Ipswitch* e o *NetSight* da *Enterasys Networks.*

.

2.2.1. WHATS UP GOLD

O WhatsUp Gold (WUG) é um software de uso simples e intuitivo que permite detectar, mapear e monitorar toda a infra-estrutura de dados, nomeadamente, dispositivos,

servidores, aplicações e tráfego de rede. Para este produto existem três versões diferentes: *Premium*, *Standard* e *Distributed*.

Segundo a empresa que o desenvolve, já conta com presença internacional na Europa, Japão e China, tendo mais de 70.000 redes monitoradas com mais de 5 milhões de dispositivos. Em termos de prospecção de mercado, conta com uma taxa de crescimento anual de 10% ao ano e uma base sólida de clientes importantes.

Na Figura 5 apresenta-se um exemplo da interface gráfica disponibilizada pela aplicação através do navegador, nomeadamente do menu principal onde se encontram listados os equipamentos monitorados, o estado dos mesmos quanto a existência de ligação, o tráfego associado a cada equipamento, o tempo de resposta aos pedidos de eco e as percentagens de utilização tanto da memória como dos discos rígidos.

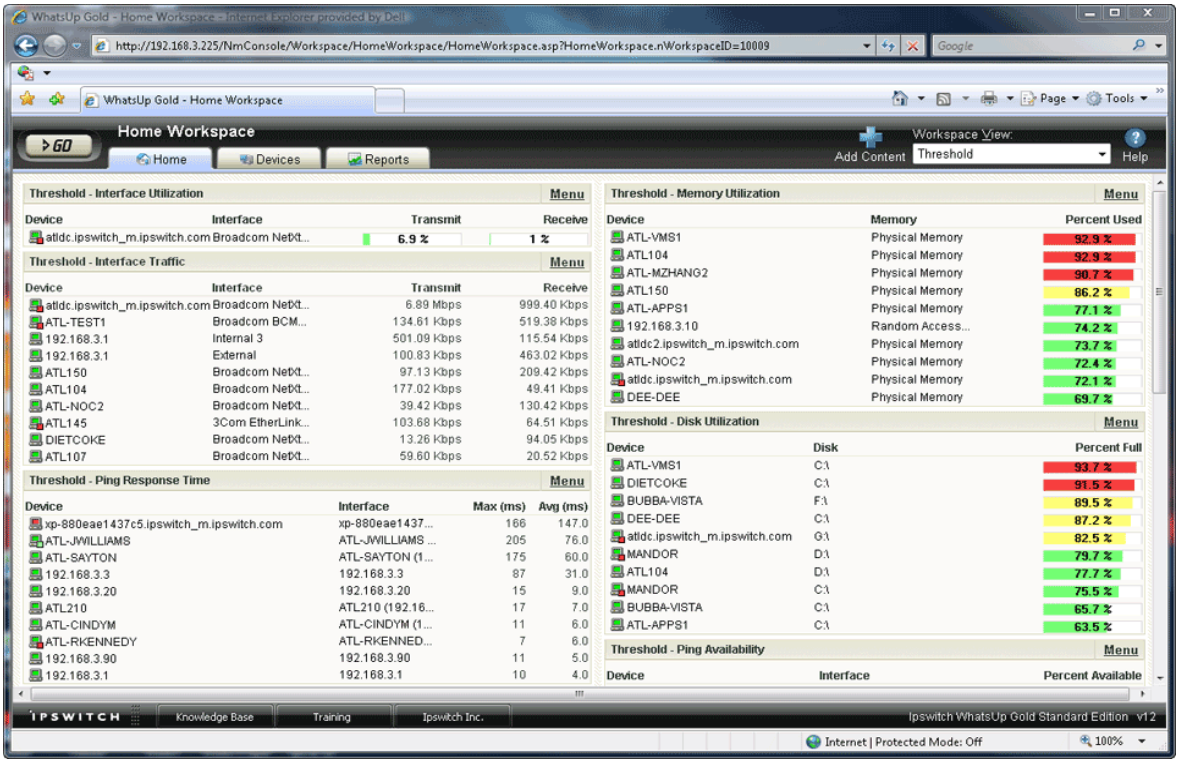


Figura 5: Menu principal do software NMS *WhatsUp Gold*.

Na Figura 6, apresenta-se um mapa da rede em causa, construída automaticamente pela aplicação e na qual se mostra em tempo real o estado das ligações e dos equipamentos. [12]

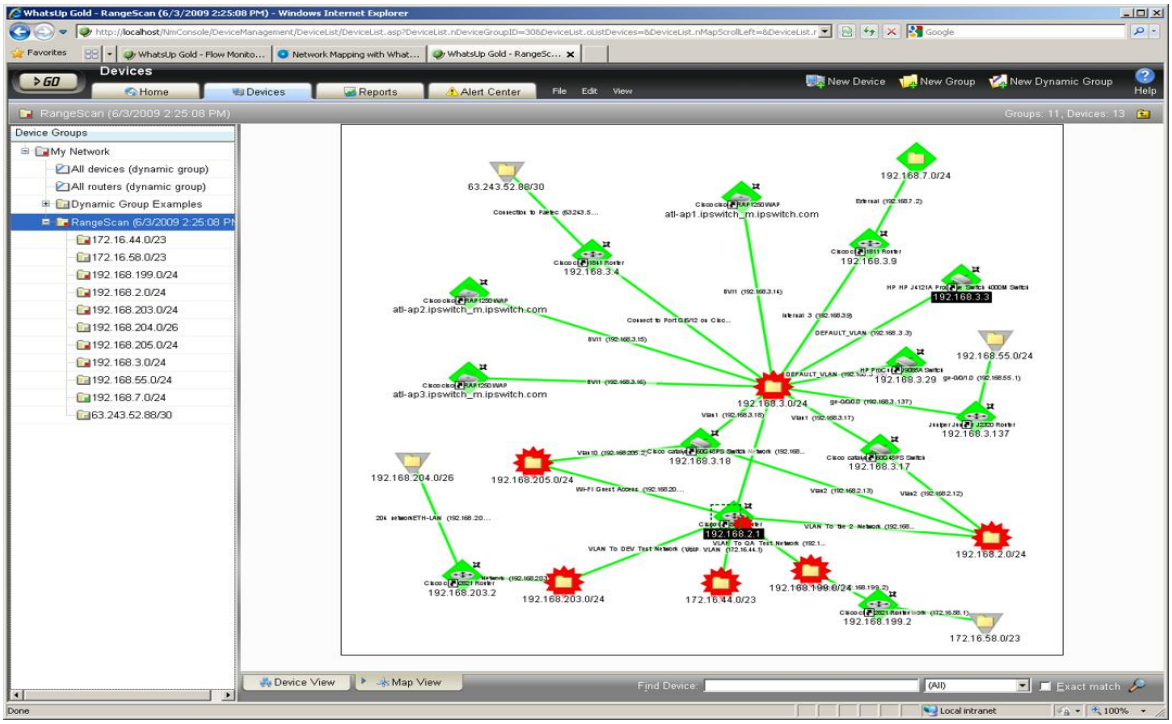


Figura 6: Exemplo de um esquema de rede construído pelo *WhatsUpGold*.

2.2.2. NETSIGHT

Este software NMS permite realizar também a gestão e monitorização de todos os componentes e obtém informação sobre o desempenho de uma infra-estrutura de rede, com especial orientação para os dispositivos fabricados pela própria *Enterasys*, dado que este software é distribuído especialmente para os clientes que adquirem os seus equipamentos. Apesar de tudo, esta aplicação permite monitorar equipamentos de outros fabricantes mas com algumas limitações no que se refere a funcionalidades da aplicação, nomeadamente, no recurso a MIBs específicas. No geral, a aplicação permite reconfigurar switchs e APs (*Access Point*), criar gráficos de certos parâmetros, criar VLANs (*Virtual-LAN*), entre outros.

Na Figura 7, apresenta-se um exemplo do menu principal da interface gráfica do *NetSight*, na qual se encontra o estado de ligação de todos os equipamentos presentes na rede, listados por endereço IP, junto com as principais propriedades de cada dispositivo. Na parte inferior da imagem, encontram-se enumerados os principais eventos acontecidos na rede, organizados por ordem de cronológica.

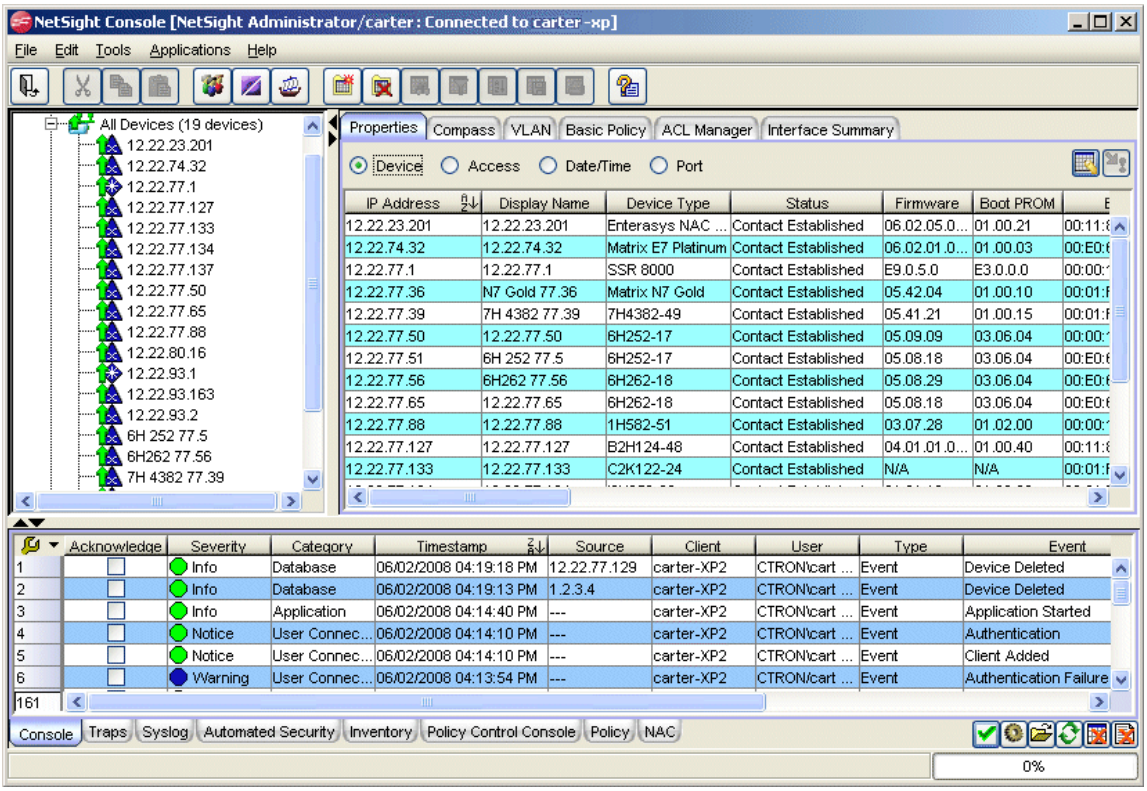


Figura 7: Menu principal da interface gráfica do *NetSight*.

Outra funcionalidade importante que a aplicação suporta é a construção automática de mapas de topologia de rede, informando do estado real das ligações e dos equipamentos existentes, como podemos ver na Figura 8. [13]

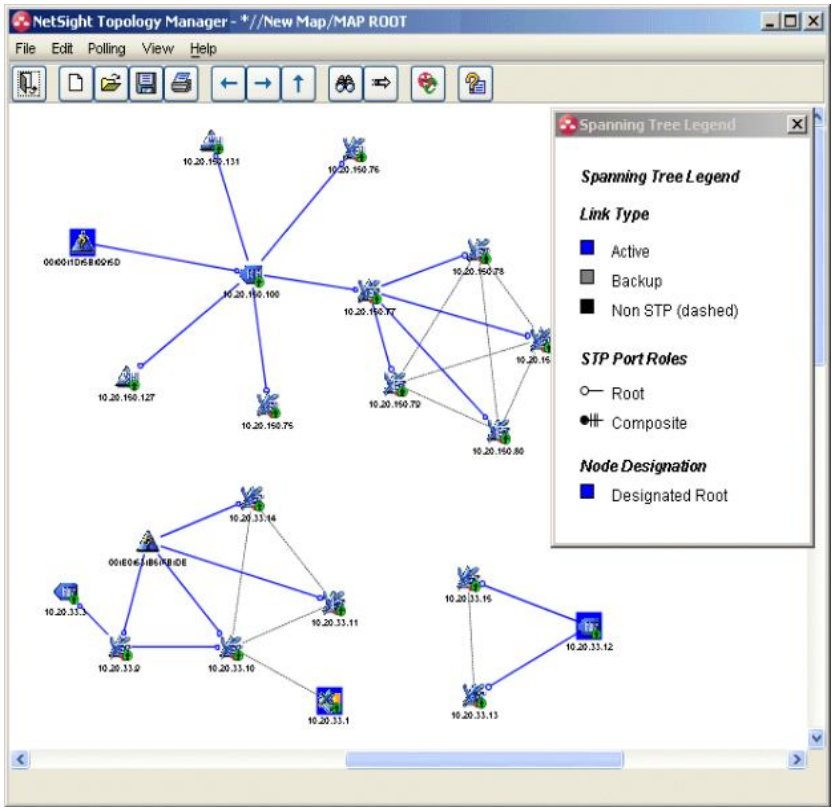


Figura 8: Mapa de rede construido com *NetSight*.

2.2.3. OUTROS NMS

No decorrer do estágio e na sequência do estudo do conceito dos sistemas de gestão de rede, foi elaborado um levantamento de algumas outras soluções existentes no mercado, sendo analisadas e comparadas as suas características e potencialidades.

2.2.3.1. HP OPEN VIEW

A versão actual do HP Open View, após reestruturação em 2007 por parte da companhia que o desenvolve, consiste num conjunto de aplicações desenhadas especialmente para gestão de redes empresariais, em que cada módulo está dedicado a cada protocolo de comunicação e a cada arquitectura de rede. Para o âmbito de estudo desta

tese, a aplicação que mais se adequa é o HPOpenView Network Node Manager (OV NNM).

O Network Node Manager recorre também ao protocolo SNMP, integrando a auto-descoberta, monitorização e controlo dos equipamentos existentes na infra-estrutura de rede. Suporta um grande número de equipamentos e modelos de vários fabricantes, dando suporte e integrando todas as suas MIBs. [14]

2.2.3.2. NAGIOS

Nagios é uma aplicação de monitorização de rede em código aberto, concebido para sistemas Linux, que tem capacidade para monitorar os principais elementos de uma infra-estrutura de dados. O seu desenvolvimento está a cargo de um grupo de programadores que voluntariamente participam no melhoramento da aplicação.

A vertente "opensource" deste sistema permite ao utilizador a possibilidade de desenvolver plug-ins com modos de monitorização personalizados e em função das necessidades de cada administrador, fazendo uso das linguagens de programação mais utilizadas (Perl, C, Python, PHP, etc). Para clientes empresariais, existe uma versão de licença comercial com maiores funcionalidades de modo a satisfazer as necessidades desse nicho de mercado.

Alguns outros sistemas NMS de código aberto, com características e funcionalidades semelhantes ao Nagios, são o OpenNMS e o Cacti. [15]

2.2.4. ANÁLISE COMPARATIVA

Do ponto de vista do administrador da rede, a escolha de um software NMS entre os vários existentes no mercado está directamente relacionada com as características da rede de dados a monitorar. Neste sentido, é necessário ter em consideração alguns princípios importantes, como podem ser o número de nós da rede, número de equipamentos, topologia da rede, variedade de fabricantes dos equipamentos instalados, necessidades de monitorização, entre outros.

De modo a ajudar a compreender as principais diferenças entre os softwares NMS anteriormente mencionados, podemos basear-nos na informação ilustrada na Tabela 2.

Software	Auto-descoberta	Plugins	Alertas	Licença Livre	Mapas	MIBs Externas	Vários fabricantes
WhatsUp Gold	✓	✓	✓	✗	✓	✓	✓
EnterasysNetSight	✓	✗	✓	✗	✓	✓	✓
HpOpenView	✓	✓	✓	✗	✓	✓	✓
Nagios	✓	✓	✓	✓	✓	✓	✓

Tabela 2: Comparação entre o NMS em análise.

Uma vez seleccionado um softwareNMS que satisfaça todas as necessidades do administrador ou do cliente, torna-se necessário realizar a análise de custo relativa à adquisição do mesmo. Normalmente, um mesmo software costuma ter várias versões à disposição do cliente, sendo normalmente a principal diferença entre elas o número de equipamentos a monitorar. No caso do *WhatsUp Gold*, existem várias licenças dentro da mesma versão, dependendo sempre do número de elementos presentes na rede. Na seguinte tabela, podemos encontrar as principais diferenças entre as três versões disponíveis.

Funcionalidades	Standard	Premium	Distributed
1- Web UI, Windows Console & Aceso móvel	✓	✓	✓
2- Centro de alertas	✓	✓	✓
3- Base de dados SQL integrada	✓	✓	✓
4- Monitorização de desempenho	✓	✓	✓
5- Recurso a Jscript ouVBscript	✓	✓	✓
6- Monitorização de rede e criação de gráficos em tempo real	✗	✓	✓
7- Aplicação de monitorização WMI	✗	✓	✓
8- Monitorização de hardware com recurso a MIBs	✗	✓	✓
9- Monitorização de servidores Unix e recursos SSH	✗	✓	✓
10- Monitorização pontos de rede Wireless (WAP)	✗	✓	✓
11- Monitores activos ou passivos já pré-configurados	✗	✓	✓
12- Transmissão de dados (HTTP/HTTPS/FTP)	✗	✓	✓
13- Serviços de monitorização remota	✗	✗	✓

Tabela 3: Comparação entre as versões do *WhatsUpGold* .

Como podemos verificar através da Tabela 3, a versão *Standard* suporta as funcionalidades mínimas indispensáveis para qualquer software NMS, enquanto a principal diferença entre as versões *Premium* e *Distributed* reside em que esta última suporta a realização de monitorização de equipamentos remotamente, desde qualquer ponto, sem ser necessariamente a partir da rede monitorada pelo administrador.

3. REDE DO AFSC

Neste capítulo apresenta-se uma breve descrição da estrutura da rede de dados da ANA Aeroportos de Portugal e do AFSC, assim como os aspectos mais importantes a ter em consideração nestas estruturas.

3.1. REDE WAN DA ANA AEROPORTOS

Pelo facto de existirem vários aeroportos geridos pela empresa, que evidentemente se encontram geograficamente separados uns dos outros, tornou-se necessário a criação de uma rede do tipo WAN para a interligação de todos os recursos e serviços que possibilitam o normal funcionamento da empresa. Desta forma, estabeleceram-se ligações entre todas as LANs de cada aeroporto, criando uma rede de dados de uso exclusivo.

A WAN é baseada na tecnologia MPLS (*Multi Protocol Label Switching*) que é suportada por dois fornecedores de serviços diferentes, ONI e PT Comunicações. As ligações de maior largura de banda (100Mbps) são fornecidas pela ONI, sendo estas as principais ligações de fornecimento de Internet aos Aeroportos e respectivos clientes e empresas associadas. O recurso a dois fornecedores de serviços diferentes permite criar uma redundância de rede entre os aeroportos mais importantes (Lisboa, Porto, Faro, Ponta Delgada e Funchal), como se verifica através do mapa da rede WAN presente na Figura 8. Deste modo, torna-se possível que os serviços básicos da empresa continuem operacionais

através da rede fornecida pela PT Comunicações, no caso de existir algum tipo de falha no circuito principal.

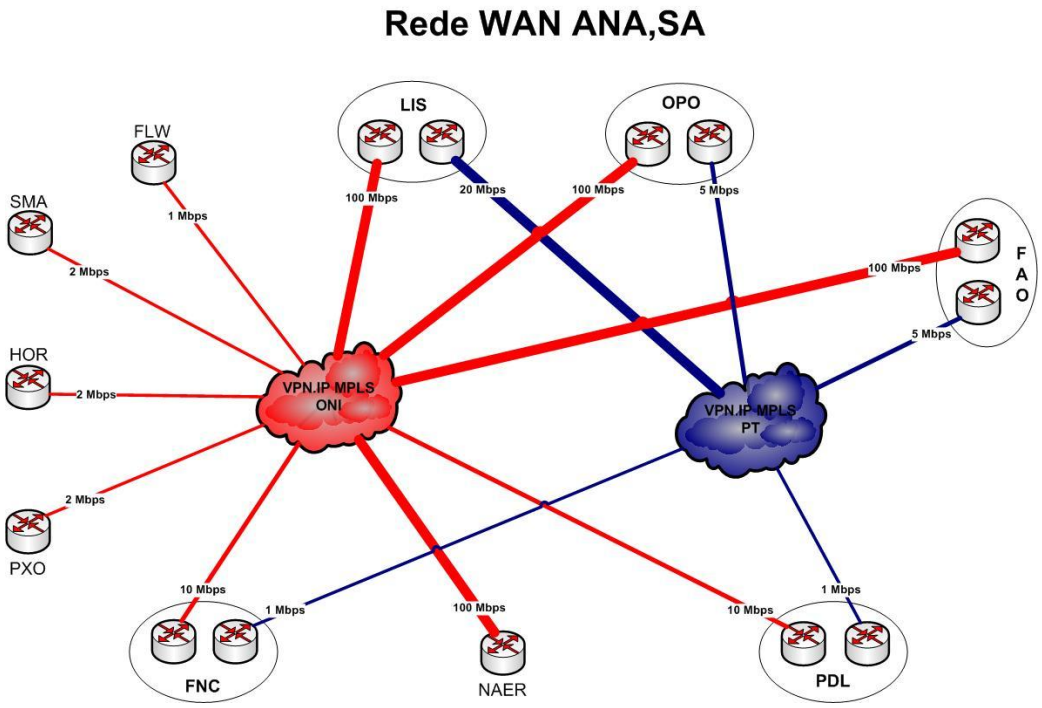


Figura 9: Mapa da rede WAN.

Na Figura 9, além da redundância na rede, podemos identificar também todos os aeroportos que se encontram ligados através da rede WAN: LIS – Lisboa, OPO – Porto, FAO – Faro, FNC – Funchal, PDL – Ponta Delgada, FLW – Flores, SMA – Santa Maria, HOR – Horta, PXO – Porto Santo e NAER – Novo Aeroporto (Não activo), e as duas redes dos fornecedores de serviços junto com as ligações existentes e a sua velocidade.

3.2. REDE LAN DO AFSC

O AFSC encontra-se dividido em vários sectores, possuindo cada um deles um ou vários nós da rede de comunicações. Esta subdivisão da rede obriga ao recurso a topologias em árvore compostas por vários níveis hierárquicos que se juntam formado uma topologia em estrela.

Dentro da rede existem dois bastidores de vital importância. O primeiro é responsável por estabelecer a interligação entre a rede WAN da empresa e a rede LAN do

aeroporto, enquanto o segundo estabelece a ligação da rede VoIP (*Voice over Internet Protocol*) à mesma LAN.

A estrutura da rede do aeroporto foi criada incluindo a implementação do protocolo STP (*Spanning Tree Protocol*) que visa recorrer a caminhos alternativos na rede no caso de existir alguma falha nas ligações. Deste modo cria-se redundância na redee no caso de algum bastidor não estar operacional, a rede LAN do aeroporto permanece em funcionamento. No anexo A.1 apresenta-se um diagrama da rede de dados que ajuda a compreender a dimensão e estruturação da mesma.

A complexidade e tamanho da rede de dados do Aeroporto Francisco Sá Carneiro requerem a utilização de redes locais virtuais sobre a rede local física. Deste modo, reduz-se o volume de tráfego a circular pela rede poupando largura de banda, tornando-a mais flexível, eficiente e segura. O recurso a este protocolo é tão elementar que neste momento existem aproximadamente 120 VLANs em funcionamento na rede de dados.

Para a criação e atribuição de VLANs, o método usado na rede do Aeroporto consiste em atribui-las a determinadas portas, de modo a que os switchs realizem o encaminhamento de tramas unicamente para as portas pertencentes à mesma VLAN.

3.3. EQUIPAMENTOS DE REDE

Devido à complexidade e volume de tráfego da rede de dados, torna-se essencial recorrer ao uso de equipamentos com elevado nível de capacidade de processamento, estabilidade, segurança e desempenho. Todos estes equipamentos, nos quais se incluemos switchs, routers, placas de rede e bastidores são, na sua maioria, da marca *Enterasys Networks.*

3.3.1. SWITCHS E ROUTERS

Os equipamentos de encaminhamento (*switching*) e roteamento (*routing*) que formam parte da rede de dados do AFSC possuem uma alta capacidade de processamento de pacotes, rondando os cem milhões de bits por segundo. Nesta rede existem modelos que operam unicamente como switchs e outros específicos que suportam as funções de switch e router em simultâneo. Através do módulo de gestão destes dispositivos torna-se possível realizar configurações que visem melhorar a segurança, atribuir prioridades e classificar o tráfego de certas transmissões de dados. Os módulos DFE (*Decision Feedback Equalizer*)

mais comuns instalados nos bastidores são os 7G4282-41; 7H4385-49; 7H4382-49;7G4270-12 e 7H4382-25. Como equipamentos independentes, os mais utilizados são os C2G124-48P; C2H124-48; V2H124-24 eB5G124-48.

Na Figura 9, podemos observar os dispositivos de rede mais usados no AFSC: o módulo 7G4282-41, na parte esquerda da imagem, e o C2G124-48P na parte direita da mesma.

Figura 10: Dispositivos 7G4282-41 e C2G124-48P.

3.3.2. BASTIDORES

Os módulos DFE estão instalados em bastidores do tipo Matrix N3, Matrix N7 e Matrix E7. O Matrix N3 possui uma estrutura de 3 slots enquanto os Matrix N7 e Matrix E7 constam de 7 slots, sendo a principal diferença entre estes dois últimos a compatibilidade com módulos mais antigos (os Matrix E7 são os únicos que contemplam esta possibilidade). O critério a seguir pelos administradores da rede na hora de seleccionar o tipo de bastidor necessário para cada sector da rede, baseia-se sempre na densidade de tráfego a processar e desempenho e capacidade necessários para o efeito. Na Figura 10, podemos identificar os bastidores Matrix N3 e Matrix N7 do lado esquerdo e direito da imagem, respectivamente.

Figura 11: Matrix N3 e Matrix N7.

3.4. EQUIPAMENTOS UTILIZADOS NO DESENVOLVIMENTO DA APLICAÇÃO

Para efeitos de desenvolvimento da aplicação, foram utilizados os equipamentos "C2G124-48" e "B5G124-48" da *Enterasys Networks*. A principal diferença entre eles reside em que os equipamentos da serie C, além de realizarem encaminhamento também efectuam operações de roteamento.

Outro dado importante a ter em consideração é o facto do equipamento da série C2 ter começado a sua comercialização no ano de 2007, enquanto que o equipamento B5 é a versão mais recente da série B (ano 2011), com as diferenças em termos de evolução tecnológica que o intervalo de tempo entre os lançamentos ao mercado destes equipamentos implica. As características mais relevantes em que esta diferença é notável são a MTBF (*Mean Time Between Failures*) que é amplamente maior no equipamento mais recente; o consumo energético que é maior no equipamento mais antigo; e nos intervalos de temperatura e humidade operacionais, sendo estes mais curtos para o equipamento da série C2.

Relativamente às semelhanças, ambos têm a mesma capacidade de encaminhamento e empilhamento, o mesmo número de portas Ethernet 10/100/1000 Mbps

e 10 GE (Gigabit Ethernet); as mesmas dimensões e intensidade sonora operacional relativamente igual.

As principais características dos equipamentos utilizados são apresentadas na Tabela 4. [16][17]

Características	B5G124-48	C2G124-48
Ano de início de comercialização	2011	2007
Nº de portas Ethernet 10/100/1000 Mbps	48	48
Nº de portas “10 GE”	2	2
Máxima capacidade de encaminhamento	96 Gbps	96 Gbps
Capacidade de empilhamento	8	8
Máxima capacidade de encaminhamento combinada	768 Gbps	768 Gbps
Nº máximo de portas Ethernet 10/100/1000 Mbps combinadas	384	384
Nº máximo de portas “10 GE” combinadas	16	16
Dimensões	A: 4.4 cm L: 44.1 cm C: 36.85 cm	A: 4.4 cm L: 44.1 cm C: 36.85 cm
Peso Neto	5.31 kg	5,71 kg
MTBF	308359 horas	113646 horas
Consumo energético	76 W	101 W
Temperatura operacional	0º até 50º C	0º até 40º C
Intervalo de humidade operacional	5% até 95 %	10% até 90%
Intensidade sonora	45.5 dB	46 dB

Tabela 4: Comparação entre os equipamentos *Enterasys* utilizados.

4. APLICAÇÃO DE GESTÃO DESENVOLVIDA

No presente capítulo é descrito todo o processo de desenvolvimento da aplicação que servirá de apoio ao gestor de rede. O objectivo é conseguir monitorar todos os parâmetros considerados como essenciais para o normal funcionamento dos equipamentos de rede *Enterasys* em análise.

4.1. PARÂMETROS A MONITORAR E RESPECTIVAS MIBS

Uma vez definida a informação útil que a aplicação deveria mostrar, foi estabelecido que se realizaria a monitorização dos seguintes parâmetros:

- Temperatura;

- Utilização da CPU;

- Estado de portas (Up/Down);

- Estado de ventoinhas;

- Tráfego e largura de banda;

Como é evidente, tanto a temperatura como a utilização do CPU e estado de ventoinhas são informação de vital importância para o normal desempenho do equipamento, estando todas directamente relacionadas entre si. Uma utilização alta do CPU traduz-se directamente num aumento da temperatura do equipamento, que pode ser atenuada sempre que o estado de funcionamento das ventoinhas seja o mais adequado aos níveis de exigência dos equipamentos. A constante monitorização destes parâmetros de índole técnica é estritamente necessária para tornar a vida útil do equipamento e respectivos componentes o mais longa possível.

No que diz respeito ao estado das portas de cada equipamento e os respectivos valores de tráfego e largura de banda, são monitorados com o intuito de obter informação acerca do estado real das ligações entre as máquinas e os equipamentos de rede. Deste modo, é possível conhecer em tempo real se a comunicação está operacional, qual a velocidade de transmissão e a quantidade de dados já transmitidos para cada ligação monitorada.

De modo a poder obter os dados de monitorização dos parâmetros requeridos, procedeu-se a realização de um levantamento das MIBs mais adequadas que a aplicação necessitaria de fazer uso. Na Tabela 5 é apresentada uma listagem de todas as variáveis utilizadas com o seu respectivo *objectID,* o tipo de variável de retorno da informação solicitada e a função que ocupa na arquitectura da aplicação.

Variável	Object ID	Descrição	Tipo
SysDescr	1.3.6.1.2.1.1.1.0	Descreve o equipamento.	Display String
Cpu_Usage_1min	1.3.6.1.4.1.5624.1.2.49.1.1.1.1.3.1.1	Valor médio da carga de trabalho do CPU, em percentagem, no último minuto.	Integer
NumFans	1.3.6.1.4.1.52.4.1.1.8.1.11.0	Número de ventoinhas existentes no equipamento.	Integer
FanStatus1	1.3.6.1.4.1.52.4.1.1.8.1.12.1.2.101	Estado da ventoinha 1	Integer
FanStatus2	1.3.6.1.4.1.52.4.1.1.8.1.12.1.2.102	Estado da ventoinha 2	Integer
FanMode1	1.3.6.1.4.1.52.4.1.1.8.1.12.1.3.101	Modo ventoinha 1	Integer
FanMode2	1.3.6.1.4.1.52.4.1.1.8.1.12.1.3.102	Modo ventoinha 2	Integer
FanSpeed1	1.3.6.1.4.1.52.4.1.1.8.1.12.1.4.101	Velocidade ventoinha 1	Integer
FanSpeed2	1.3.6.1.4.1.52.4.1.1.8.1.12.1.4.102	Velocidade ventoinha 2	Integer
IfOperStatus (1..48)	1.3.6.1.2.1.2.2.1.8.(Nºde porta)	Estado da ligação	Integer
IfInOctets (1..48)	1.3.6.1.2.1.2.2.1.10.(Nºde porta)	Número de octetos de entrada	Counter
IfOutOctets (1..48)	1.3.6.1.2.1.2.2.1.16.(Nºde porta)	Número de octetos de saída	Counter
IfSpeed (1..48)	1.3.6.1.2.1.2.2.1.5.(Nºde porta)	Velocidade de ligação	String

Tabela 5: Listagem das variáveis utilizadas.

4.2. DIAGRAMA E ARQUITECTURA

A Figura 12 resume o funcionamento da aplicação, apresentando os blocos ligados entre si que representam por sua vez, os módulos da arquitectura funcional da aplicação e a transferência da informação por ela tratada.

Em termos gerais, podemos descrever os módulos da seguinte forma:

-**Dispositivo**: Elementos da rede a serem monitorados pela aplicação. Neste caso, equipamentos de rede *Enterasys*.

-**Módulo de Comunicação**: Responsável por estabelecer a interacção com o dispositivo fazendo uso dos protocolos TCP/IP e SNMP através do agente "Net-SNMP".

-**Módulo de Controlo**: Elemento chave da aplicação. É aqui que se executam as acções de obtenção da informação de gestão do dispositivo e onde se trata a informação para posteriormente ser guardada. Não é nada mais do que um conjunto de Scripts em linguagem Perl que são executados a partir de um servidor Linux.

-**Base de Dados**: Responsável por guardar a informação e retorná-la quando solicitada por parte da interface.

-**Interface**: Representação gráfica da informação manipulada pela aplicação e que permite a utilização desta por parte do utilizador.

Figura 12: Diagrama funcional da aplicação desenvolvida.

No que diz respeito à arquitectura na qual se assenta o funcionamento da aplicação, esta pode ser ilustrada pela Figura 13. Como se pode verificar, os dispositivos monitorados estão ligados através da rede de dados Ethernet ao servidor principal da aplicação do tipo Linux, onde se executam os módulos de comunicação e controlo. A informação recolhida é tratada, enviada e guardada no servidor de base de dados Microsoft SQL Server 2008 existente no Aeroporto, servidor este que vai responder a todos os pedidos de dados solicitados pela interface, através do servidor web.

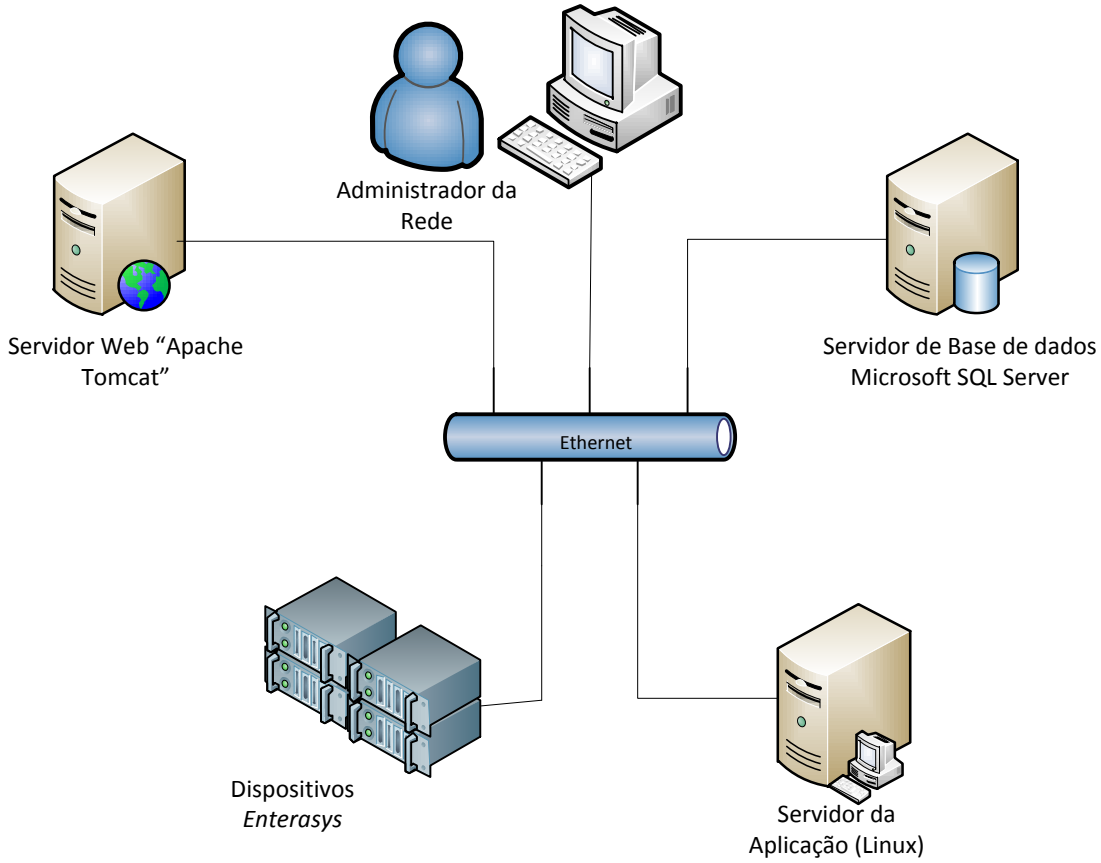


Figura 13: Arquitectura da aplicação desenvolvida.

4.3. ESTRUTURA GENÉRICA DE UM SCRIPT

Todo o código base elaborado nesta aplicação foi desenvolvido em linguagem de programação "Perl". Na necessidade de criar funções que traduzam para variáveis a informação solicitada, estabeleceram-se scripts que fazem uso do agente SNMP para solicitar a informação. Após obter a resposta, esta é devidamente tratada para cada caso em específico com ajuda de ficheiros de configuração previamente carregados, isto é, os endereços IP aos quais se solicita a informação são atribuidos pelo gestor de rede através de um ficheiro de texto editável, presente no servidor da aplicação. Após o processamento de todas as respostas, a informação relevante é enviada para uma base de dados.

No Excerto 2 (cuja função é obter a percentagem de ocupação do CPU no último minuto) podemos encontrar um exemplo de qual o tipo de abordagem encontrada para realizar pedidos SNMP a um determinado equipamento. Na linha 9 encontra-se uma *string* que contém o pedido GET, o IP do equipamento e o ID da MIB que retorna a informação

descritiva do equipamento. Após a recolha da informação contida na mensagem resposta, na linha 10 utiliza-se a função *chomp* que elimina qualquer caracter de "nova linha" no fim da *string* retornada.

Para guardar a informação na base de dados, é necessário utilizar uma DBI (*Data Base Interface*) de Perl que seja compatível com o servidor de base de dados). Na linha 14 realiza-se a ligação e início de sessão com a base de dados que, no caso de ter sucesso, permite o envio da *query* SQL (linha 16) que guarda a informação na tabela seleccionada por esta. Na linha 18 a string onde se insere a *query* é enviada para o *buffer* e na linha 20 implementa-se no servidor de base de dados. De modo a finalizar o processo, na linha 22 temos aapresentação do resultado na consola e na linha 24 o pedido de fim da sessão no servidor de base de dados. Esta abordagem de envio de informação para a base de dados é utilizada recorrentemente ao longo de todos os scripts que formam parte do código base da aplicação.

```
1- #!/usr/bin/perl -w
2- #use strict;
3-
4- use DBI;
5-
6- $SNMP_GET_CMD = "snmpget -v1 -c public -Ovq ";
7- $SNMP_TARGET = "192.168.1.11 ";
8-
9- my $sysDescr = `${SNMP_GET_CMD}${SNMP_TARGET} iso.3.6.1.2.1.1.1.0`;
10- chomp($sysDescr);
11- my$cpu_usage_5sec=`${SNMP_GET_CMD}${SNMP_TARGET}
iso.3.6.1.4.1.5624.1.2.49.1.1.1.1.3.1.1`;
12- chomp($cpu_usage_5sec);
13-
14- my $dbh = DBI-> connect("dbi:ODBC:odbc-test", 'username', 'password') or die "Can´t
connect: $DBI::errstr";
15-
16- my $sql = "INSERT INTO CPU (Equipamento,IP,CPU,Data) VALUES
('${sysDescr}','$SNMP_TARGET',$cpu_usage_5sec,getdate())";
17-
18- my $sth = $dbh->prepare($sql) or die "Can't prepare statement: $DBI::errstr";
19-
20- $sth->execute();
21-
22- print("${sysDescr}\n Com o IP $SNMP_TARGET Tem o CPU a $cpu_usage_5sec%\n");
23-
24- $dbh->disconnect;
```

Excerto 2: Obtenção da percentagem de ocupação do CPU e envio da resposta à base de dados.

4.4. PEDIDOS SNMP

A aplicação desenvolvida faz uso do agente SNMP de livre distribuição "Net-SNMP". Este agente permite realizar pedidos SNMP aos equipamentos, pedidos estes baseados em MIBs já pré-seleccionadas.

Para obter os valores de cada parâmetro, é necessário recorrer a um pedido GET, o qual solicita a uma ou mais MIBs a informação necessária. Este tipo de pedido realiza-se em todos os casos, exceptuando o parâmetro da temperatura, já que esta informação é fornecida através de uma *Trap* de alarme.

No Excerto 3, encontram-se todos os pedidos do tipo GET relativos a obtenção do estado de funcionamento das ventoinhas dos equipamentos.

```
1.  my $NumFans = `${SNMP_GET_CMD}${SNMP_TARGET} iso.3.6.1.4.1.52.4.1.1.8.1.11.0`;
2.  chomp($NumFans);
3.  my $FanStatus1 = `${SNMP_GET_CMD}${SNMP_TARGET}
    iso.3.6.1.4.1.52.4.1.1.8.1.12.1.2.101`;
4.  chomp($FanStatus1);
5.  my $FanStatus2 = `${SNMP_GET_CMD}${SNMP_TARGET}
    iso.3.6.1.4.1.52.4.1.1.8.1.12.1.2.102`;
6.  chomp($FanStatus2);
7.  my $FanMode1 = `${SNMP_GET_CMD}${SNMP_TARGET}
    iso.3.6.1.4.1.52.4.1.1.8.1.12.1.3.101`;
8.  chomp($FanMode1);
9.  my $FanMode2 = `${SNMP_GET_CMD}${SNMP_TARGET}
    iso.3.6.1.4.1.52.4.1.1.8.1.12.1.3.102`;
10. chomp($FanMode2);
11. my $FanSpeed1 = `${SNMP_GET_CMD}${SNMP_TARGET}
    iso.3.6.1.4.1.52.4.1.1.8.1.12.1.4.101`;
12. chomp($FanSpeed1);
13. my $FanSpeed2 = `${SNMP_GET_CMD}${SNMP_TARGET}
    iso.3.6.1.4.1.52.4.1.1.8.1.12.1.4.102`;
14. chomp($FanSpeed2);
```

Excerto 3: Pedidos do tipo GET.

4.5. CONVERSÃO DE DADOS

Como verificamos anteriormente na Tabela 4, algumas MIBs retornam informação de estado (Ex: rápido, lento, falha, etc.) através de uma variável do tipo *Integer*, o que obriga a converter esse inteiro numa *string* que traduza de uma forma mais compreensível a informação solicitada ao equipamento. Neste sentido, foram desenvolvidos mecanismos em Perl que realizam o tratamento da informação recebida para o posterior envio à base de dados.

No Excerto 4 de código Perl, podemos encontrar o procedimento de tratamento dos dados para a conversão da resposta do tipo *integer* para uma *string* que posteriormente será guardada na base de dados.

```
1-    if($FanStatus1==1)
2-    {
3-     $FanStatus1DB = "unknown";
4-    }
5-    elsif ($FanStatus1==2)
6-     {
7-     $FanStatus1DB = "normal";
8-    }
9-    elsif ($FanStatus1==3)
10-    {
11-    $FanStatus1DB = "testing";
12-   }
13-   elsif ($FanStatus1==4)
14-    {
15-    $FanStatus1DB = "slow";
16-   }
17-   elsif ($FanStatus1==5)
18-   {
19-    $FanStatus1DB = "inoperative";
20-   }
21-   elsif ($FanStatus1==6)
22-   {
23-    $FanStatus1DB = "off";
24-   }
25-   print("\nVentoinha 1 ---- $FanStatus1DB\n");
```

Excerto 4: Conversão de *integer* para *string*.

Outro caso em que se torna necessário realizar a conversão dos tipos de variáveis encontra-se no Excerto 5. Aqui encontra-se um ciclo que percorre todas as portas que o equipamento tem activas, onde uma vez detectada que a porta está activa, efectua-se o pedido GET onde se obtém o número de octetos que a porta em análise contabilizou. Com o valor dos octetos já atribuido na variavel, realiza-se a conversão do número de octetos do tipo *counter*, para o formato simplificado de tipo *character*, obtendo assim o tráfego associado e velocidade das ligações de cada porta activa no equipamento.

```
1-    for($i = 1; $i<=48; $i++)
2-    {
3-    my $porta1 = `${SNMP_GET_CMD}${SNMP_TARGET} iso.3.6.1.2.1.2.2.1.8.$i`;
4-    chomp($porta1);
5-    select(undef, undef, undef, 0.25);
6-    if($porta1==1)
7-    {
8-    $estado = Up;
9-    #print("\nPORTA $i ---- $estado\n");
10-   my $octeto_in = `${SNMP_GET_CMD}${SNMP_TARGET} iso.3.6.1.2.1.2.2.1.10.$i`;
11-   chomp($octeto_in);
12-   print("\n$octeto_in\n");
13-   if($octeto_in<= 1023999999)
14-   {
15-    $trafego_in = $octeto_in/1024000;
16-    $T_IN = "${trafego_in} Mb";
17-    print("\nTrafego = $T_IN \n");
18-   }
19-   else
20-   {
21-    $trafego_in = $octeto_in/1024000000;
22-    $T_IN = "${trafego_in} Gb";
```

```
23-     print("\nTrafego = $T_IN \n");
24-    }
25-    my $octeto_out = `${SNMP_GET_CMD}${SNMP_TARGET} iso.3.6.1.2.1.2.2.1.16.$i`;
26-    chomp($octeto_out);
27-    print("\n$octeto_out\n");
28-    #sleep(1);
29-    if($octeto_out<= 1023999999)
30-    {
31-     $trafego_out = $octeto_out/1024000;
32-     $T_OUT = "${trafego_out} Mb";
33-     print("\nTrafego = $T_OUT \n");
34-    }
35-    else {
36-     $trafego_out = $octeto_out/1024000000;
37-     $T_OUT = "${trafego_out} Gb";
38-     print("\nTrafego = $T_OUT \n");
39-    }
40-    my $bandwith = `${SNMP_GET_CMD}${SNMP_TARGET} iso.3.6.1.2.1.2.2.1.5.$i`;
41-    chomp($bandwith);
42-    print("\n$bandwith\n");
43-    #sleep(1);
44-    if($bandwith<= 1023999999)
45-    {
46-     $bandwith_F = $bandwith/1024000;
47-     $BAND = "${bandwith_F} Mb/s";
48-    print("\nBandwith = $BAND \n");
49-    }
50-    else {
51-     $bandwith_F = $bandwith/1024000000;
52-     $BAND = "${bandwith_F} Gb/s";
53-    print("\nBandwith = $BAND\n");
54-    }
55-    }
56-    else {
57-     $estado = Down;
58-     $T_IN = 0;
59-     $T_OUT = 0;
60-     $BAND = 0;
61-    print("\nPORTA $i ---- $estado\n");
```

Excerto 5: Conversão de counter*r* para *character*.

4.6. TRAPS SNMP

Para realizar o controlo do estado de temperatura dos equipamentos, utilizam-se *Traps* SNMP. Estas são geradas quando se atinge os valores de temperatura limite, de modo a não por em causa nem a integridade do equipamento nem a qualidade do serviço por ele prestado na rede. O intervalo de temperatura de funcionamento suportado está compreendido entre os 0 ºC e os 50 ºC, sendo os 40 ºC o valor de temperatura que acciona o envio da *Trap* de alarme.

Neste contexto, desenvolveu-se um servidor integrado no Módulo de Controlo que fica à escuta das *Traps* de alarme e regista num ficheiro "log" e na base de dados o conteúdo do evento comunicado. Este servidor encontra-se no Excerto 6.

```
1-#!/usr/bin/perl -w
2-
3- use DBI;
4-
5- my $TRAP_FILE = "/home/jonathan/trapall.log";
6-
7- my $host = <STDIN>;
8- chomp($host);
9- my $ip = <STDIN>;
10- chomp($ip);
11-
12- while(<STDIN>) {
13-        chomp($_);
14-        push(@vars,$_);
}
15-
16- open(TRAPFILE, ">> $TRAP_FILE");
17- $date = `date`;
18- chomp($date);
19- print(TRAPFILE "Nova trap recebida: $date por $OID\n\nHOST: $host\nIP: $ip\n");
20-foreach(@vars) {
        print(TRAPFILE "TRAP: $_\n");
}
21- print(TRAPFILE "\n----------\n");
22- close(TRAPFILE);
23-
24- my $dbh = DBI-> connect("dbi:ODBC:odbc-test", 'username', 'password')or die "Can´t
connect: $DBI::errstr";
25-
26- my $sql = "INSERT INTO Temperatura(IP,Data) VALUES ('$ip', getdate())";
27-
28- my $sth = $dbh->prepare($sql)or die "Can't prepare statement: $DBI::errstr";
29-
30- $sth->execute();
31-
32- $dbh->disconnect;
```

Excerto 6: Script de escuta das Traps.

Após o desenvolvimento do script de escuta, foi necessário configurar o agente SNMP do equipamento para realizar o envio de traps cada vez que a temperatura atingisse um determinado valor. Assim, o conteúdo da mensagem de recepção da *Trap* apresenta o formato que se apresenta no Excerto 7, e que se encontra nos ficheiros de log manipulados automaticamente através do script.

```
Nova trap recebida: Wed Jun 4 15:12:48 PDT 2012 por

HOST: 192.168.1.11
IP: 192.168.1.11
TRAP: RFC3877-MIB::ituAlarmEventType
```

Excerto 7: Exemplo de recepção de *Trap*.

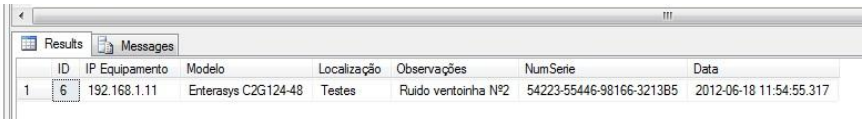
4.7. BASE DE DADOS

Toda a informação relativa aos parâmetros dos equipamentos a monitorar, após recolha e tratamento, é armazenada numa base de dados "Microsoft SQL Server 2008".

Para organizar a informação de forma a facilitar o acesso a esta por parte da aplicação desenvolvida, foram criadas as seguintes tabelas:

- ***Equipamentos***: Tabela na qual se associa cada endereço IP em uso na rede ao modelo e Nº de série do equipamento, assim como à zona do aeroporto na qual se encontra. Na Figura 14 apresenta-se um exemplo de registo na tabela. (Em todas as tabelas seguintes consta uma figura com exemplos de registos)

Campos: ID; IP Equipamento; Modelo; Localização; Observações; NumSerie; Data.

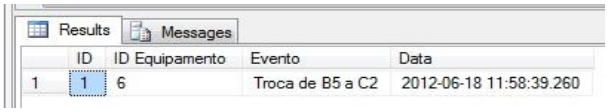


	ID	IP Equipamento	Modelo	Localização	Observações	NumSerie	Data
1	6	192.168.1.11	Enterasys C2G124-48	Testes	Ruido ventoinha Nº2	54223-55446-98166-3213B5	2012-06-18 11:54:55.317

Figura 14: Entrada na tabela Equipamentos.

- ***Eventos****:* Nesta tabela registam-se os acontecimentos de interesse que podem ocorrer nos equipamentos que existem na rede. Como por exemplo, avarias, substituição por outros modelos ou adição/remoção de funcionalidades.

Campos: ID; ID Equipamento; Evento; Data.

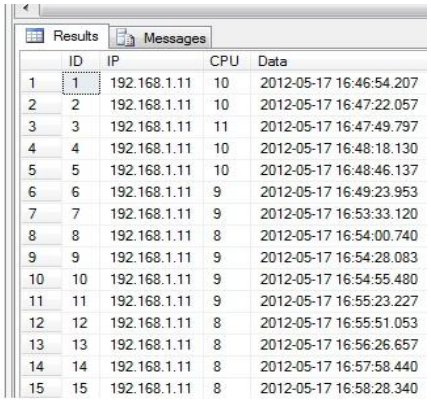


	ID	ID Equipamento	Evento	Data
1	1	6	Troca de B5 a C2	2012-06-18 11:58:39.260

Figura 15: Entrada na tabela Eventos.

- ***CPU:*** Esta tabela recolhe as amostras da percentagem de utilização da CPU dos equipamentos, obtidas em intervalos de tempo constantes de 30 segundos de duração.

Campos: ID; IP; CPU; Data.



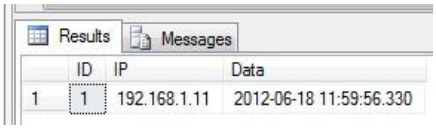
Results | Messages

	ID	IP	CPU	Data
1	1	192.168.1.11	10	2012-05-17 16:46:54.207
2	2	192.168.1.11	10	2012-05-17 16:47:22.057
3	3	192.168.1.11	11	2012-05-17 16:47:49.797
4	4	192.168.1.11	10	2012-05-17 16:48:18.130
5	5	192.168.1.11	10	2012-05-17 16:48:46.137
6	6	192.168.1.11	9	2012-05-17 16:49:23.953
7	7	192.168.1.11	9	2012-05-17 16:53:33.120
8	8	192.168.1.11	8	2012-05-17 16:54:00.740
9	9	192.168.1.11	9	2012-05-17 16:54:28.083
10	10	192.168.1.11	9	2012-05-17 16:54:55.480
11	11	192.168.1.11	9	2012-05-17 16:55:23.227
12	12	192.168.1.11	8	2012-05-17 16:55:51.053
13	13	192.168.1.11	8	2012-05-17 16:56:26.657
14	14	192.168.1.11	8	2012-05-17 16:57:58.440
15	15	192.168.1.11	8	2012-05-17 16:58:28.340

Figura 16: Entradas na tabela CPU.

- ***Temperatura***: Tabela na qual se registam todos os alarmes de temperatura ocorridos em cada equipamento.

Campos: ID; IP; Data.



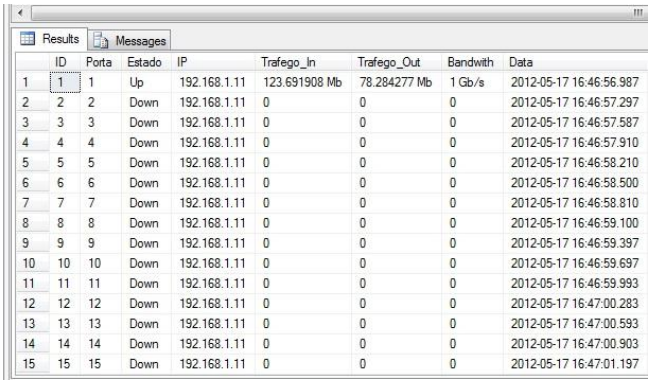
Results | Messages

	ID	IP	Data
1	1	192.168.1.11	2012-06-18 11:59:56.330

Figura 17: Entrada na tabela Temperatura.

- ***Tráfego***: Nesta tabela armazenam-se todos os dados relativos ao estado das portas de cada equipamento, valores de trafego e largura de banda respectiva.

Campos: ID; IP; Porta; Estado; Trafego_IN; Trafego_OUT; Bandwith; Data.



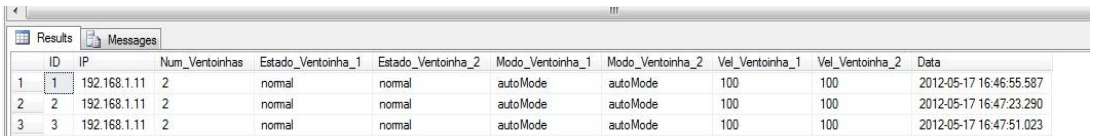
Results | Messages

	ID	Porta	Estado	IP	Trafego_In	Trafego_Out	Bandwith	Data
1	1	1	Up	192.168.1.11	123.691908 Mb	78.284277 Mb	1 Gb/s	2012-05-17 16:46:56.987
2	2	2	Down	192.168.1.11	0	0	0	2012-05-17 16:46:57.297
3	3	3	Down	192.168.1.11	0	0	0	2012-05-17 16:46:57.587
4	4	4	Down	192.168.1.11	0	0	0	2012-05-17 16:46:57.910
5	5	5	Down	192.168.1.11	0	0	0	2012-05-17 16:46:58.210
6	6	6	Down	192.168.1.11	0	0	0	2012-05-17 16:46:58.500
7	7	7	Down	192.168.1.11	0	0	0	2012-05-17 16:46:58.810
8	8	8	Down	192.168.1.11	0	0	0	2012-05-17 16:46:59.100
9	9	9	Down	192.168.1.11	0	0	0	2012-05-17 16:46:59.397
10	10	10	Down	192.168.1.11	0	0	0	2012-05-17 16:46:59.697
11	11	11	Down	192.168.1.11	0	0	0	2012-05-17 16:46:59.993
12	12	12	Down	192.168.1.11	0	0	0	2012-05-17 16:47:00.283
13	13	13	Down	192.168.1.11	0	0	0	2012-05-17 16:47:00.593
14	14	14	Down	192.168.1.11	0	0	0	2012-05-17 16:47:00.903
15	15	15	Down	192.168.1.11	0	0	0	2012-05-17 16:47:01.197

Figura 18: Entrada na tabela Tráfego.

- ***Ventoinhas***: Para esta tabela envia-se a informação relativa ao estado das ventoinhas de cada equipamento em análise.

Campos: ID; IP;Num_Ventoinhas; Estado_Ventoinha1; Estado_Ventoinha2; Modo_Ventoinha1; Modo_Ventoinha2; Vel_Ventoinha1; Vel_Ventoinha2; Data.



Results | Messages

	ID	IP	Num_Ventoinhas	Estado_Ventoinha_1	Estado_Ventoinha_2	Modo_Ventoinha_1	Modo_Ventoinha_2	Vel_Ventoinha_1	Vel_Ventoinha_2	Data
1	1	192.168.1.11	2	normal	normal	autoMode	autoMode	100	100	2012-05-17 16:46:55.587
2	2	192.168.1.11	2	normal	normal	autoMode	autoMode	100	100	2012-05-17 16:47:23.290
3	3	192.168.1.11	2	normal	normal	autoMode	autoMode	100	100	2012-05-17 16:47:51.023

Figura 19: Entrada na tabela Ventoinhas.

Finalmente, na Figura 20, encontramos uma imagem que mostra a estrutura da base de dados, junto com os tipos de variáveis de cada campo:

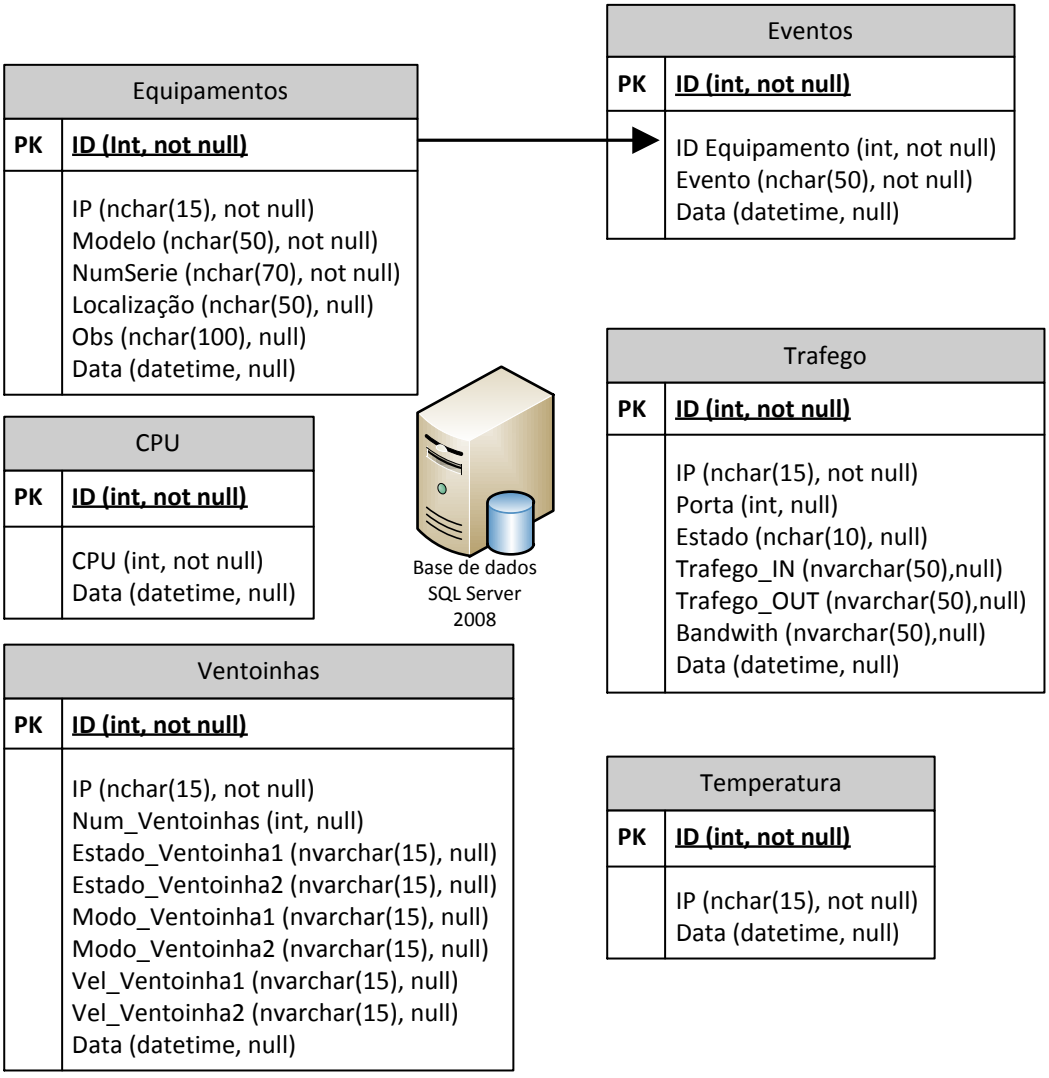


Figura 20: Arquitectura da base de dados desenvolvida.

4.8. TECNOLOGIA USADA NA INTERFACE COM O UTILIZADOR

A interface gráfica que permite mostrar ao administrador da rede a informação recolhida é baseada em tecnologia JSP (*Java Server Pages*). Baseada na linguagem de programação "Java", esta abordagem consiste em desenvolver aplicações do tipo "Web" que têm a capacidade de aceder a informação anteriormente armazenada, recolher os dados

seleccionados e mostrá-los num ambiente amigável e de fácil compreensão para o utilizador.

No Excerto 8, de código JSP, podemos encontrar um exemplo de como se podem seleccionar e mostrar os dados de um parâmetro em estudo que consiste em enviar uma *query* à base de dados na qual se solicita a selecção dos dados desejados pelo utilizador, como podemos ver na linha 18 do excerto de código. Para mostrar no ecrã a informação retornada, na linha 20 temos o ciclo que imprime todas as linhas da tabela solicitada pela *query*.

Todos os módulos JSP desenvolvidos para a aplicação objecto de estudo na presente tese estão acessíveis através de um servidor web "Apache Tomcat".

```
1- <%@page contentType="text/html" pageEncoding="UTF-8"%>
2-<!DOCTYPE html>
3- <%@ page import="java.sql.*" %>
4-
5- <html>
6- <head>
7- <title>Portas 192.168.1.11</title>
8- </head>
9- <body text="black" bgcolor="White">
10- <table cellpadding="50">
11- <tr>
12- <td>
13-
14.<%Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver").newInstance();
15- Connection conn = null;
16- String url = "jdbc:sqlserver://ascsql.ana.pt:1433;" +
   "databaseName=LANSNMP;user=user;password=password;";
conn = DriverManager.getConnection(url);
17- Statement s = conn.createStatement ();
18- s.executeQuery ("SELECT * FROM [LANSNMP].[dbo].[Trafego] WHERE Estado like '%Up%'
and ID in (SELECT TOP 48 ID FROM [LANSNMP].[dbo].[Trafego] ORDER BY ID DESC);");
19- ResultSetrs = s.getResultSet ();
20- while (rs.next ()){
out.println ("Trafego Download: " + rs.getString (6) + "<br/>Trafego Upload: " +
rs.getString (7) + "<br/>Velocidade: "  + rs.getString (8) + "<br/>");
21- }
22-
23- rs.close ();
24- s.close ();
25- conn.close ();
26- %>
27-
28- </table>
29- </body>
30- </html>
```

Excerto 8: Exemplo de uma pagina JSP.

4.9. INTERFACE GRÁFICA

A aplicação desenvolvida tem a sua interface em tecnologia HTML (*Hiper Text Markup Language*) e JSP, tornando-se necessário recorrer a um *browser* para aceder e interagir com a mesma. Durante o processo de desenvolvimento da interface, foi utilizado o *browser* Mozilla Firefox, versão 12.0.

No momento em que tentamos aceder ao endereço da aplicação, surge a página de início de sessão presente na Figura 21, na qual será necessário introduzir correctamente as credenciais de acesso para poder transitar até o menu principal da aplicação. A elaboração e distribuição de credenciais não será um assunto abordado neste relatório.

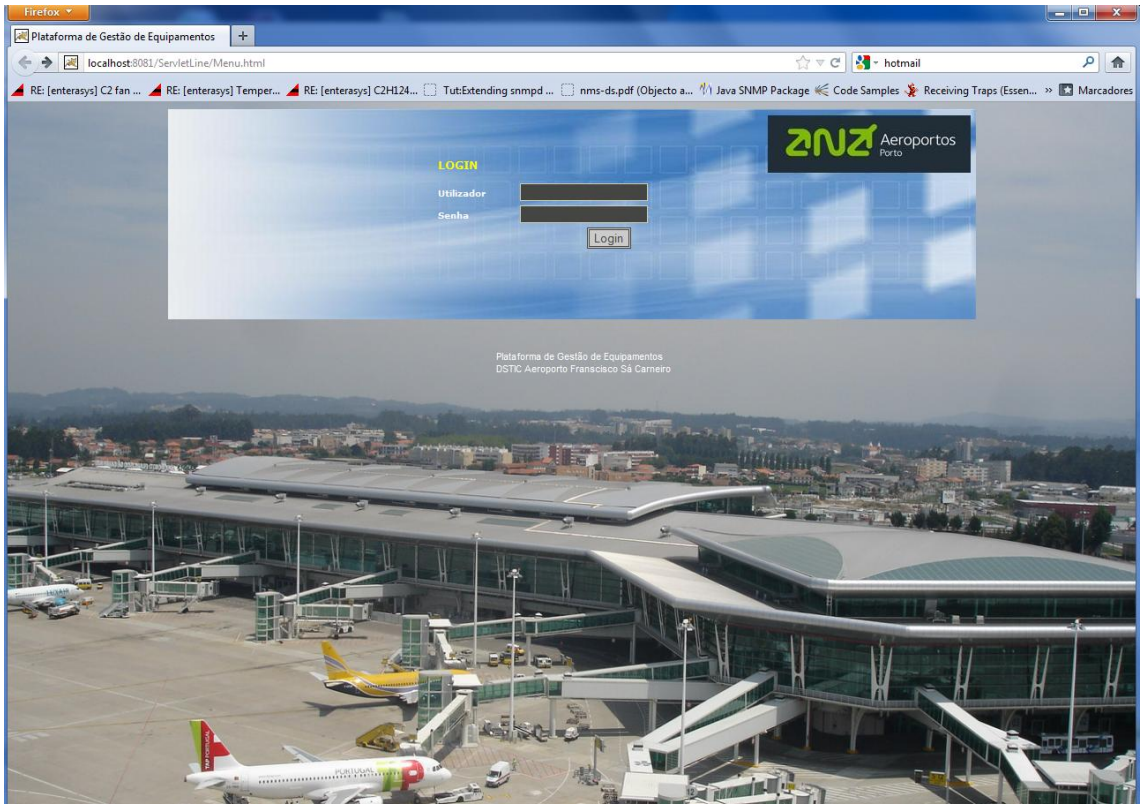


Figura 21: Página de acesso à aplicação.

No menu principal da aplicação (Figura 22), na parte central, apresentam-se os equipamentos presentes na rede, com descrição do IP, modelo do equipamento e informação detalhada do estado das portas. Os equipamentos estão separados por zonas, como podemos visualizar na Figura 23.

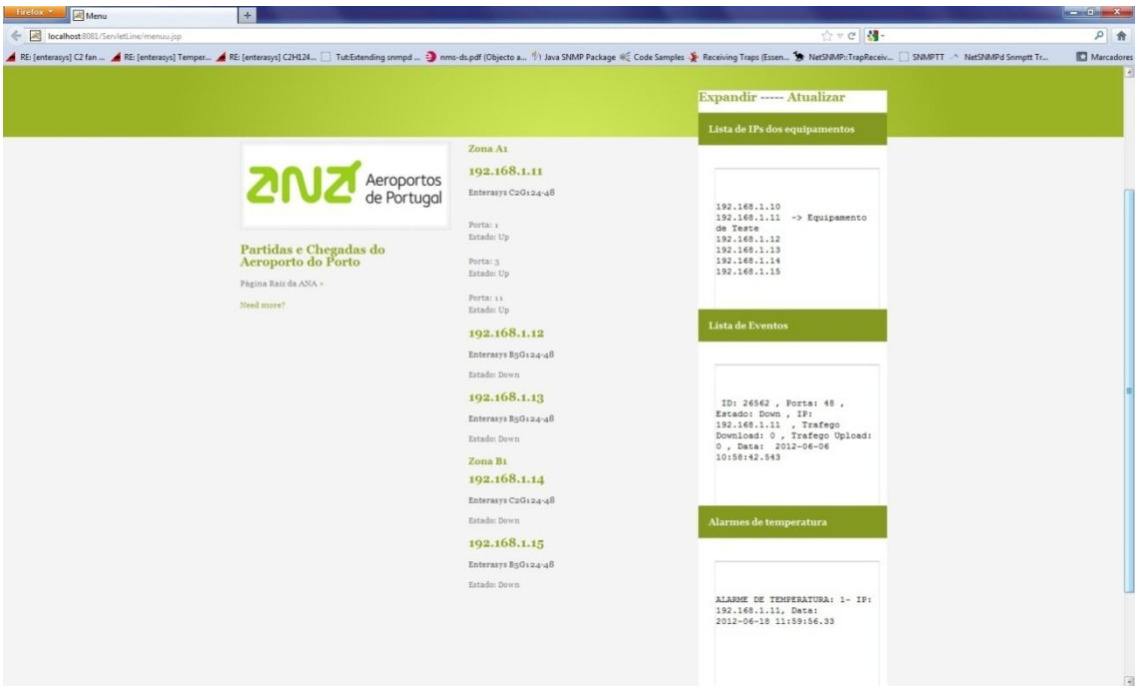


Figura 22: Menu principal da aplicação.

Figura 23: Equipamentos e estado dos mesmos separados por zonas.

A informação aqui mostrada reside nas tabelas da base de dados e é acedida através das páginas JSP criadas. Do lado direito da Figura 14 encontra-se a lista dos equipamentos que a aplicação vai monitorar, a lista de eventos registados em todos os equipamentos, os alarmes existentes naquele instante e os botões que permitem actualizar ou expandir a informação presente nas listagens, como se apresenta na Figura 24.

Figura 24: Página de consulta do estado do equipamento.

Para estas funções, a interface limita-se a abrir e mostrar os ficheiros de log criados pela aplicação aquando de cada acontecimento. Ao clicar no botão "Expandir", abre-se uma página (Figura 25) que mostra em ecrã inteiro a informação que consta dos ficheiros de log, de modo a poder facilitar a visualização desta informação.

Figura 25: Página de consulta do conteúdo dos ficheiros de log.

A página presente na Figura 22 foi desenvolvida em HTML e com recurso a um ficheiro CSS (*Cascading Style Sheets*) especialmente desenvolvido para o efeito. O Excerto 9, de código CSS diz respeito a construção das páginas presentes na aplicação, e é responsável por atribuir as cores de fundo junto com estilo e cores das letras.

```
1-      *{margin:0;padding:0;outline:0}
2-      body { font: 11px/18px Georgia, Palatino, "Times New Roman", Times, Serif;
background: #f4f4f4 url(bg.jpg) no-repeat center top; color: #777; }
3-
4-      a { text-decoration: none; color: #89A213; }
```

```
5-      a:hover { color: #556314; }
6-
7-      p  { margin: 0 0 15px; line-height: 1.6em; }
8-      h1 { float: left; width: 700px; line-height: 1.5em; font-size: 2.7em; color:
#fff; margin: 0 0 20px; text-shadow: #89A213 1px 1px1px; }
9-      h2 { margin: 0 0 15px; font-size: 1.6em; }
10-     h3 { margin: 0 0 7px; font-size: 1.3em; clear: both; color: #444; line-height:
1.3em; }
11-     h4 { margin: 0 0 10px; font-size: 1.2em; }
12-
13-     img { border: 0; }
14-     .x { clear: both; }
15-
16-     #content { margin: 0 auto; width: 960px;}
17-     #header{ height: 350px; }
18-     #top { padding: 13px 0 0; margin: 0 0 60px; color: #fff; height: 31px; }
19-     #pitch { clear: left; float: left; width: 610px; font-size: 1.2em; padding: 20px
0 0; color: #59690C; margin: 0 0 60px; }
20-
21-     #menu { float: right; margin: 10px 15px 0 0; }
22-     #menu li { display: inline; }
23-     #menu li a { float: left; color: #EFF4D7; font-size: 1.2em; margin: 0 0 0 20px;
padding: 4px; }
24-     #menu li a:hover, #menu li a.current { color: #fff; border-bottom: 1px solid
#A5BE2E; }
25-
26-     #cols{ clear: both; }
27-     .col { float: left; width: 300px; margin: 0 39px 30px 0; }
28-     .last { position: relative; float: right; margin: -76px 0 0; background: #fff;
width: 280px; }
29-     .col.last div { padding: 24px; }
30-     .img { clear: both; margin: 0 0 15px; border: 1px solid #ddd; padding: 5px; }
31-     .date { margin: 0 0 12px; color: #444; }
32-     .col h4 { background: #83981F; padding: 15px; color: #fff; }
33-
34-     #main { float: left; clear: both; width: 640px; font-size: 1.2em; }
35-     .left { float: left; margin: 0 30px 10px 0; }
36-     #main p { text-align: justify; }
37-
38-     #footer { clear: both; border-top: 1px solid #ddd; color: #999; padding: 35px 0
15px 0; }
39-     #right{ float: right; }
40-     #footer p { margin: 0 0 12px; }
41-     #footer a { color: #555; margin: 0 0 0 5px; }
```

Excerto 9: CSS da interface com o utilizador.

Após clicar no endereço IP do equipamento que desejamos verificar, surge a página de consulta do estado deste, como podemos comprovar na Figura 26:

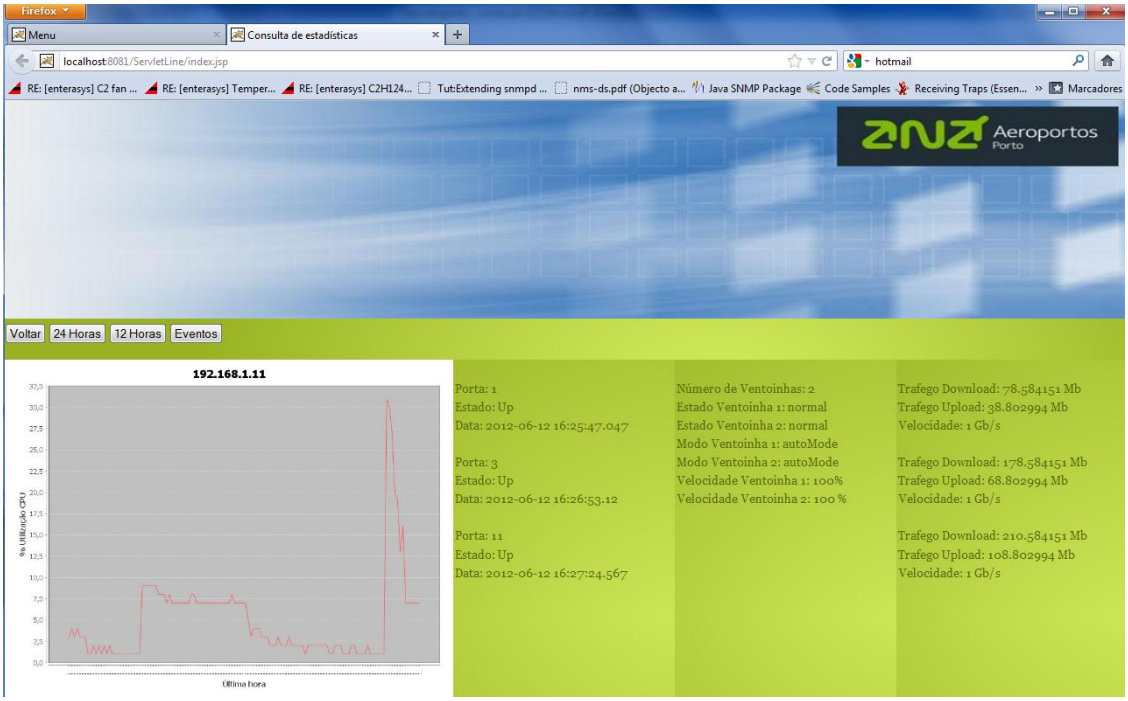


Figura 26: Página de consulta do estado do equipamento.

Nesta página podemos encontrar, nomeadamente, informação das portas activas, estado real das ventoinhas, o tráfego e a largura de banda de cada porta. A informação do estado da porta tem o trafego desta mesma porta alinhado a sua direita, por exemplo, na Figura 27 a porta Nº1 tem um tráfego de download de 78.584.151 Mbytes.

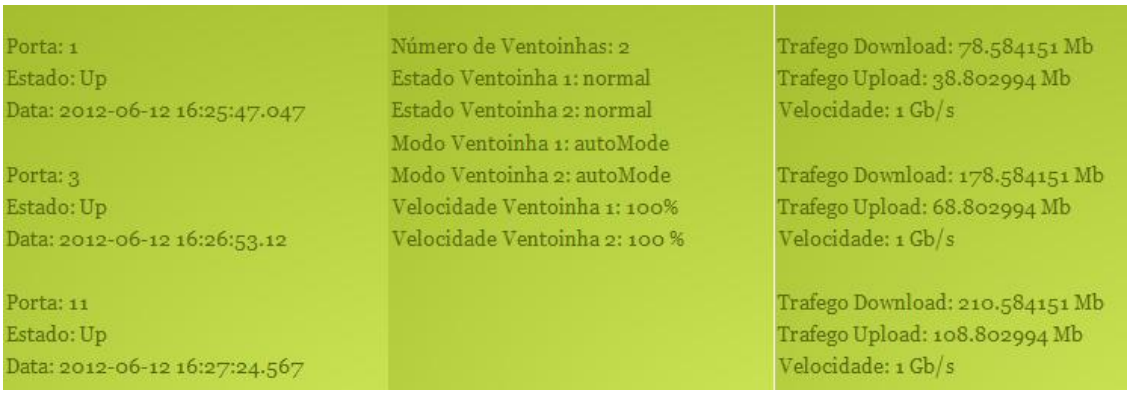


Figura 27: Zoom sobre a informação das portas.

Na figura 28 temos um gráfico da percentagem de utilização do CPU do equipamento em análise. A informação mostrada no gráfico pode ser alterada através dos botões existentes, alternando entre gráficos das últimas 12 ou 24 horas. Nesta imagem também podemos encontrar também um botão para mostrar os eventos presentes no menu principal.

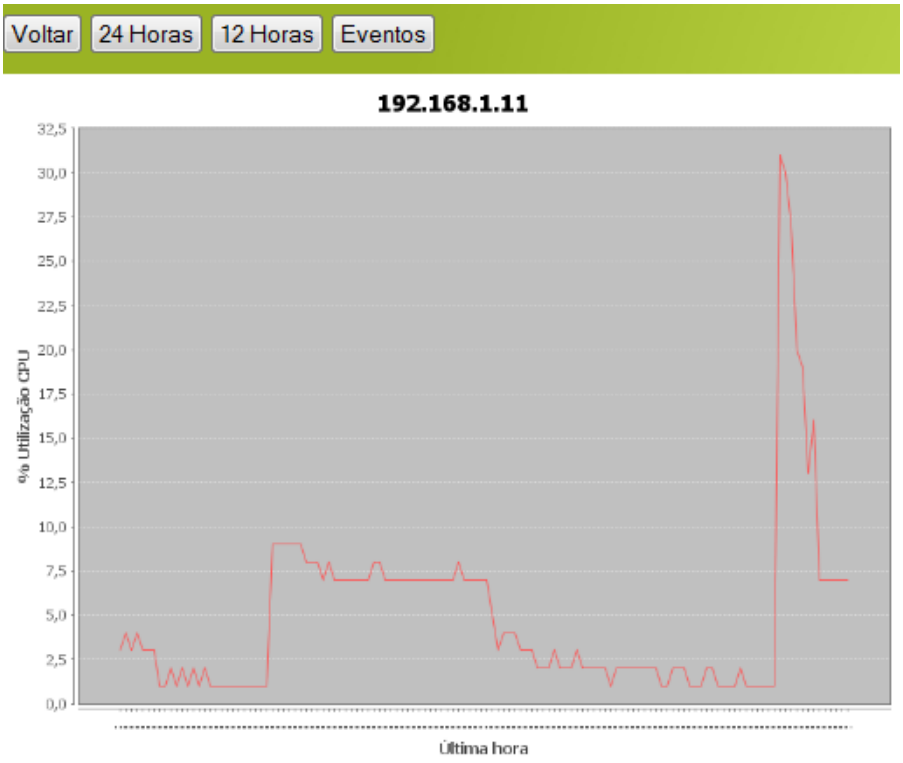


Figura 28: Zoom sobre o gráfico de utilização do CPU.

5. CONCLUSÃO

Desde o início da realização do presente trabalho, foi atribuída uma especial importância aos objectivos que se pretendiam atingir, tendo sido conseguidos em maior ou menor medida. Neste sentido, não se pode deixar de ter em atenção que esta foi uma mera abordagem à proposta que foi apresentada no âmbito do estágio, cujo tempo era limitado e os meios tecnológicos muito específicos, tendo estes factores jogado um papel determinante nas decisões e soluções implementadas durante a realização do projecto.

A elaboração deste relatório seguiu a mesma rota que se estabeleceu durante o decorrer do estágio. Em primeiro lugar, foi realizado um trabalho de estudo do protocolo SNMP, que seria a base tecnológica na qual se apoiaria o desenvolvimento da aplicação de gestão, obrigando a uma boa compreensão e domínio destes conceitos, para assim poder implementar as melhores soluções possíveis em cada caso.

Uma vez que o estágio decorreu num meio pouco comum e de grande exigência para com todos os elementos e colaboradores que formam parte da sua estrutura, justificava-se claramente realizar uma abordagem às tecnologias utilizadas na rede de dados do AFSC, focando particularidades desta como a arquitectura, os activos de rede

utilizados, as características de todos os elementos presentes e os softwares de gestão utilizados para gerir e monitorar toda a infra-estrutura de dados.

Após uma fase inicial de aquisição de "*know-how*", iniciou-se o processo de idealização e concepção da arquitectura, da escolha das tecnologias de programação mais adequadas, e do melhor aproveitamento dos meios ao dispor (servidor de base de dados, switchs e routers). O aspecto que levantou mais constrangimentos na implementação do trabalho foi a selecção das variáveis de MIBs necessárias para obter os dados do estado dos parâmetros dos equipamentos (devido a que os equipamentos *Enterasys* são equipamentos utilizados a nível profissional e existe pouca documentação de apoio para este assunto).

Relativamente à interface com o utilizador, os conhecimentos de base adquiridos ao longo dos anos anteriores foram de vital importância para o desenvolvimento de todas as ferramentas necessárias para o utilizador poder aceder e visualizar todos os dados de parametrização que a aplicação dispõe de um modo fácil e rápido. As soluções encontradas são as indispensáveis para satisfazer as perspectivas e objectivos traçados desde o início do estágio e do presente projecto.

Como não poderia ser de outra forma, existem vários melhoramentos a implementar na aplicação. Um dos mais aconselhados seria a descoberta automática de *hosts* existentes na rede recorrendo a pedidos SNMP, ao contrário do método actual que consiste em ler os endereços IP a partir de um ficheiro de configuração. Outra funcionalidade com grande utilidade para o gestor de rede seria introduzir o envio de alarmes para um e-mail ou por SMS, com todas as vantagens que isso implica à hora de monitorar em tempo real todos os sistemas. Por último, a construção de mapas de rede por parte da aplicação também viria a ser uma excelente opção para tornar a este software mais completo e cada vez mais semelhante aos existentes no mercado.

Referências Documentais

[1] *http://www.ana.pt/portal/page/portal/ANA/PAGINA_CONTINUIDADE_EMPRESA/?EMP_CT=87510&actualmenu=82274709&cboui=87510*

[2] http://www.ana.pt/portal/page/portal/ANA/PAGINA_CONTINUIDADE_EMPRESA/?EMP_CT=91795&actualmenu=82274767&cboui=91795

[3] http://www.cisco.com/en/US/docs/ios/11_3/feature/guide/snmpinfm.html

[4] Douglas R. Mauro & Kevin J. Schmidt. (2001). *Essential SNMP* (1st ed.). Sebastopol, CA: O'Reilly & Associates.

[5] RFC3411 - An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks.

[6] Stallings, William *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2*, Addison-Wesley Publishing Company, 3rd ed. (1999)

[7] http://www.snmplink.org/snmparticles/abeginnersguide/#1

[8] http://www.paessler.com/knowledgebase/en/topic/653-how-do-snmp-mibs-and-oids-work

[9] http://www.ieee802.org/1/pages/MIBS.html

[10] http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a00800b4cee.shtml

[11] http://en.wikipedia.org/wiki/Net-SNMP

[12] http://www.whatsupgold.com/

[13] http://www.enterasys.com/products/visibility-control/netsight-console.aspx

[14] http://en.wikipedia.org/wiki/HP_Network_Management_Center

[15] http://nagios.org/

[16] Manuais Enterasys C2

[17] Manuais Enterasys B5

[18] http://www.perl.org/books/beginning-perl/

[19] http://en.wikipedia.org/wiki/ODBC

[20] http://msdn.microsoft.com/en-us/library/bb264565(v=sql.90).aspx

[21] http://www.coreservlets.com/Apache-Tomcat-Tutorial/

[22] http://docs.oracle.com/javase/6/docs/api/

[23] http://www.jfree.org/jfreechart/

[24] http://www.w3schools.com/html/default.asp

[25] http://www.w3schools.com/sql/default.asp

[26] http://www.w3schools.com/css/

[27] http://jsptut.com/

Anexo A.

DIAGRAMA DE REDE DE DADOS

V2 SABA/GALP

V2 ETAR

mCon3082 MARCO

V2 PASOCORROS

mCon3082 TELEMETRO

V2 CAP NORTE

C2 AVA

B5 LAS/PTW

B5 AOMP PGA

B5 AOMP TAP

SCCSL/SGTC

HP PROCURVE

C2 APRO

B5 GUARITA A

C2 B4.3A.1

C2 B4.3A.2

C2 B4.3A.3

C2 B4.3A.5

C2 B4.3A.4

SITA / CUTE

C2 B4.2A.1

C2 B4.3B.1

C2 B4.3B.2

C2 B4.3B.4

C2 B4.3B.5

C2 B4.3B.6

C2 B4.2B.1

C2 B5.2

V2 B5.1

C2 B5.0.1

C2 B5.0.2

C2 B5.0.3

C2 B5.0 - SEF

C2 GUARITA B

ROUTER-FIDS/METRO

C2 CARGA-GF

V2 B5-1

B5 CARGA Sala 325

C2 CARGA Sala 301

C2 CARGA Sala 219A

C2 CARGA PORTWAY

E1 TORRE

V2 FISCALIZAÇÃO

V2 PSP

C2 B4.0B.1

C2 B4.0B.2

C2 B4.0B.3

V2 PTW CAER

C2 B3.2.1

C2 B3.2,3-SEF

C2 B3.2.2

C2 B3.2.3

C2 B3.1A

C2 B3.1B.1

C2 B3.1B.2

C3 B3,1B,3

V2

C2 B3.0A.1

C3 B3,0A,2

C2 B3.0A.3

V2 B3.-1

V2 B3.0B

C3 B4,0A,3.1

C2 B4.0A.3.2

C2 B4.0A.3.3

V2 B5 4.1

V2 B4.-1.1

V2 B4.-1

V2 PARQUE SUB

C2 B5 PSUB

V2 METRO PORTO

C2 DASC

C2 SSO

C2 B4.0A.1

C2 B4.0A.2

C2 B4.0A.4

C2 B4.0B.2-SEF

I3-1P4

I3-2P4

VH AVIS

VH HERTZ

VH NATIONAL/GUERIN

VH CASTANHEIRA

VH EUROPCAR

VH AUTOJARDIM

V2 SIXT/AURORENT

V2 CLA

LEGENDA

NORMAL

SPANNING-TREE

Aeroportos de Portugal