

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RODRIGO BONFÁ DRAGO

**UM PROTÓTIPO DE UMA FERRAMENTA WEB PARA GERÊNCIA
DE SERVIÇOS EM AMBIENTES DE TI**

VITÓRIA
2006

RODRIGO BONFÁ DRAGO

**UM PROTÓTIPO DE UMA FERRAMENTA WEB PARA GERÊNCIA
DE SERVIÇOS EM AMBIENTES DE TI**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica, na área de concentração em Automação.

Orientador: Prof. Dr. Anilton Salles Garcia.

VITÓRIA
2006

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

D759p Drago, Rodrigo Bonfá, 1981-
Um Protótipo de uma ferramenta web para gerência de serviços em ambientes de TI / Rodrigo Bonfá Drago. – 2006.
117 f. : il.

Orientador: Anilton Salles Garcia.
Dissertação (mestrado) – Universidade Federal do Espírito Santo,
Centro Tecnológico.

1. Gerência 2. Redes de computação. 3. Software. I. Garcia, Anilton Salles. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

CDU: 621.3

RODRIGO BONFÁ DRAGO

**UM PROTÓTIPO DE UMA FERRAMENTA WEB PARA GERÊNCIA
DE SERVIÇOS EM AMBIENTES DE TI**

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisição parcial para a obtenção do Grau de Mestre em Engenharia Elétrica - Automação.

Aprovada em 12 de abril de 2006.

COMISSÃO EXAMINADORA

Prof. Dr. Anilton Salles Garcia
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. José Gonçalves Pereira Filho
Universidade Federal do Espírito Santo

Prof. Dr. Pedro Frosi Rosa
Universidade Federal de Uberlândia

Dedicatória

Aos meus pais, Adair Drago e Inez Bonfá Drago, e aos meus irmãos, Ádrian Bonfá Drago e Ágnes Bonfá Drago, pelo convívio diário, pelas oportunidades de educação que sempre me proporcionaram, pelo amor e pelo apoio. Sem a ajuda de vocês, com 100% de certeza, eu não chegaria até aqui.

À minha namorada, Milena Schneider Campo Dall'Orto, pelo apoio, compreensão, convívio e amor. Ainda que indiretamente, saiba que você muito contribuiu para a concretização deste meu trabalho.

Agradecimentos

Primeiramente a Deus, por me dar saúde e uma família tão digna que sempre me ajudou nos momentos mais difíceis.

Ao meu Orientador, Prof. Dr. Anilton Salles Garcia, pela sua dedicação, atenção e apoio durante a realização deste trabalho.

Ao meu irmão Ádrian Bonfá Drago e ao amigo Rodolfo da Silva Villaça, cujas contribuições foram importantíssimas para a realização deste trabalho. À colega Andréa Ramos, cujo suporte técnico e bibliográfico foi de grande valia. Ao colega Maxwell Monteiro pelas sugestões e discussões filosóficas sobre “Gerência de Serviços”.

Aos amigos Wylber Polonini, Ricardo Rios, Cristiano Biancardi, Leonardo José Silvestre, Renzo Que devez Malini e Camilo Zardo Calvi, que sempre me ajudaram nos trabalhos cotidianos no LPRM (Laboratório de Pesquisas em Redes e Multimídia).

Aos Profs. Dr. José Gonçalves P. Filho e Dr. Pedro Frosi Rosa, pelo tempo demandado para avaliação deste trabalho.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), por subsidiar os meus estudos durante o mestrado.

Aos meus colegas, amigos e professores que trabalham/trabalharam no LPRM.

A todos vocês, muito obrigado!!!

Sumário

1	<i>Introdução</i>	17
1.1	<i>Motivação</i>	17
1.2	<i>Definição do Problema</i>	20
1.3	<i>Objetivos da Dissertação</i>	21
1.4	<i>Organização da Dissertação</i>	22
2	<i>Gerenciamento de Redes e Serviços</i>	24
2.1	<i>Gerência de Redes</i>	24
2.1.1	Áreas Funcionais da Gerência de Redes.....	25
2.1.2	Arquitetura de Gerência de Redes.....	29
2.1.3	O Protocolo SNMP.....	32
2.2	<i>Gerência de Serviços</i>	36
2.2.1	ITIL (<i>Information Technology Infrastructure Library</i>).....	36
2.3	<i>Gerência Baseada em Níveis de Serviço (SLM)</i>	40
3	<i>Trabalhos Relacionados</i>	41
3.1	<i>O Protocolo SNMP para Gerência de Serviços</i>	41
3.1.1	MIBs SNMP para Gerência de Serviços Específicos.....	41
3.1.2	MIB Script.....	42
3.2	<i>A Framework for IT Service Management - MNM Team</i>	45
3.3	<i>Plataforma de Gerenciamento Trace</i>	49
3.4	<i>Software Livre para Gerenciamento de Serviço</i>	51
3.5	<i>Concepção da Ferramenta Proposta Baseada nos Trabalhos Relacionados</i>	53
4	<i>A Ferramenta Proposta (SMS – Service Management System)</i>	54
4.1	<i>Funcionalidades</i>	54
4.1.1	Grupo de Serviços e a Métrica IDS.....	56
4.1.2	As Estatísticas Geradas nos Relatórios.....	57
4.2	<i>Análise e Projeto</i>	60
4.2.1	Modelagem de um Serviço.....	63
4.2.2	Lógica do Sistema.....	67
4.2.3	Monitoramento dos Serviços de Rede.....	67

4.3	<i>O Diferencial da Ferramenta Proposta</i>	71
5	<i>Estudo de Caso</i>	72
6	<i>Conclusão</i>	78
6.1	<i>Resultados da Dissertação</i>	80
6.2	<i>Trabalhos Futuros</i>	80
	<i>Referências Bibliográficas</i>	82
	<i>Apêndice A - Fluxogramas de Monitoramento dos Serviços</i>	85
	<i>Apêndice B - Modelo de Classes e Modelo Lógico do Banco de Dados do Sistema</i>	100
	<i>Apêndice C - Interfaces de Navegação do SMS</i>	109

Lista de Tabelas

Tabela 1- Classificação de sistemas de acordo com a disponibilidade.	74
---	----

Lista de Figuras

Figura 1 - Níveis de Gerência.....	19
Figura 2 - Áreas Funcionais de Gerência.	25
Figura 3 - Arquitetura Geral de Sistema de Gerência.	29
Figura 4 - Arquiteturas de Sistemas de Gerência de Redes.....	31
Figura 5 - Modelo Agente/Gerente (Cliente/Servidor).	33
Figura 6 - Arquitetura de funcionamento do SNMP.	35
Figura 7 - O <i>framework</i> ITIL (Itil, 2004).	37
Figura 8 - Processos do módulo <i>Service Delivery</i> do ITIL (Itil, 2004).....	39
Figura 9 - Estrutura da MIB Script (Gaspary, 2002).....	43
Figura 10 - <i>Framework</i> para Gerencia de Serviços de TI (Rodosek, 2002).....	46
Figura 11 - Modelo de Serviço proposto por (Rodosek, 2002).....	48
Figura 12 - Componentes da Arquitetura da Plataforma Trace (Gaspary, 2002).....	51
Figura 13 - Diagrama de casos de uso do SMS.....	56
Figura 14 - Arquitetura conceitual do SMS.	61
Figura 15 – Implementação da arquitetura do SMS.....	62
Figura 16 - Diagrama de Pacotes do SMS.....	63
Figura 17 - Modelo geral de serviço do SMS.....	64
Figura 18 - Diagrama de Classes do SMS.....	66
Figura 19 - Lógica do SMS.	67
Figura 20 - Fluxograma do monitoramento do serviço de E-mail (SMTP).	69
Figura 21 - Página principal do SMS.	72
Figura 22 - Estudo de caso.	74
Figura 23 - Relatório de IDS.	76
Figura 24 - Relatório do Serviço de Impressão.	77

Figura 25 - Fluxograma do monitoramento do serviço WWW(HTTP e HTTPS).	85
Figura 26 - Fluxograma do monitoramento do serviço DNS.	87
Figura 27 - Fluxograma do monitoramento do serviço Telnet.	89
Figura 28 - Fluxograma do monitoramento do serviço Comando.	90
Figura 29 - Fluxograma do monitoramento do serviço FTP.	92
Figura 30 - Fluxograma do monitoramento do serviço de Banco de Dados.	94
Figura 31 - Fluxograma do monitoramento do serviço Porta TCP.	95
Figura 32 - Fluxograma do monitoramento do serviço IP usando o SNMP.	97
Figura 33 - Fluxograma do monitoramento do serviço IP usando o Ping (ICMP).	97
Figura 34 - Classes de persistência.	100
Figura 35 - Classe para gerar gráficos.	100
Figura 36 - Classes do kernel do SMS.	101
Figura 37 - Classes Java Servlets.	102
Figura 38 - Diagrama lógico do banco de dados.	103
Figura 39 - Inicializando o SMS.	109
Figura 40 - Criando uma coleta.	110
Figura 41 - Criando um cluster de serviços.	111
Figura 42 - Criando um grupo de serviços.	112
Figura 43 - Gerando um relatório de um serviço.	113
Figura 44 - Gerando um relatório de IDS.	115
Figura 45 - Visualizando o modelo <i>Red-Green</i>	116
Figura 46 - Relatório de um Cluster de Serviço de DNS.	117

Lista de Equações

Equação 1 - Tempo de Resposta Médio (TR).	58
Equação 2 - Disponibilidade Média (A).	58
Equação 3 - Disponibilidade Média de um Cluster de serviço (A_{CA}).	58
Equação 4 - MTTR.	59
Equação 5 - MTTF.	59
Equação 6 - MTBF.	59
Equação 7 - Índice de Disponibilidade Média (IDS).	59
Equação 8 - IDS do SMS.	60

Lista de Acrônimos

API	– <i>Application Programming Interface</i>
ASN.1	– <i>Abstract Syntax Notation One</i>
BSC	– <i>Balanced Scorecard</i>
CRM	– <i>Customer Relationship Management</i>
DHCP	– <i>Dynamic Host Configuration Protocol</i>
DNS	– <i>Domain Name Service</i>
E-R	– <i>Entidade-Relacionamento</i>
FTP	– <i>File Transfer Protocol</i>
HTTP	– <i>Hyper Text Transfer Protocol</i>
HTTPS	– <i>Hyper Text Transfer Protocol Secure</i>
ICMP	– <i>Internet Control Message Protocol</i>
IDS	– <i>Índice de Disponibilidade de Sistema</i>
IETF	– <i>Internet Engineer Task Force</i>
IMAP	– <i>Internet Message Access Protocol</i>
IP	– <i>Internet Protocol</i>
ISO	– <i>International Organization for Standardization</i>
ITIL	– <i>Information Technology Infrastructure Library</i>
ITU-T	– <i>International Telecommunication Union – Telecommunication Section</i>
LPRM	– <i>Laboratório de Pesquisa em Redes e Multimídia</i>
MIB	– <i>Management Information Base</i>
MTBF	– <i>Mean Time Between Failure</i>
MTTF	– <i>Mean Time To Failure</i>
MTTR	– <i>Mean Time To Repair</i>
NFS	– <i>Network File System</i>

NIS	– <i>Network Information System</i>
NMS	– <i>Network Management System</i>
NOC	– <i>Network Operations Center</i>
NTP	– <i>Network Time Protocol</i>
OID	– <i>Object Identifier</i>
OSI	– <i>Open Systems Interconnection</i>
QoS	– <i>Quality of Service</i>
QoD	– <i>Quality of Device</i>
POP3	– <i>Post Office Protocol – Version 3</i>
RFC	– <i>Request For Comments</i>
RMON	– <i>Remote Network Monitoring</i>
RDBMS	– <i>Relational Data Base Management System</i>
SLA	– <i>Service Level Agreement</i>
SLM	– <i>Service Level Management</i>
SMI	– <i>Structure of Management Information</i>
SMS	– <i>Short Message Service</i>
SMS	– <i>Service Management System</i>
SMTP	– <i>Simple Mail Transfer Protocol</i>
SNMP	– <i>Simple Network Management Protocol</i>
SQL	– <i>Structured Query Language</i>
TCP	– <i>Transmission Control Protocol</i>
TCP/IP	– <i>Transmission Control Protocol/ Internet Protocol</i>
TI	– <i>Tecnologia da Informação</i>
UML	– <i>Unified Modeling Language</i>
WWW	– <i>World Wide Web</i>

Resumo

Devido ao crescimento da complexidade das redes e aplicações e à necessidade de serviços de TI (Tecnologia da Informação) distribuídos, houve uma mudança na forma pela qual os sistemas são gerenciados. O foco moveu-se da gerência orientada a dispositivos para a gerência orientada a serviços. Atualmente, pequenas e grandes organizações precisam de uma infra-estrutura de TI para suportar seus negócios, e esta infra-estrutura deve ser monitorada a fim de garantir o seu bom funcionamento. Métricas quantitativas devem ser medidas e utilizadas para se obter um *feedback* a respeito do funcionamento do ambiente de TI. Este trabalho propõe um sistema de gerência de serviços de rede, baseado em *web*, que utiliza padrões abertos e software livre, para ajudar na prática do Gerenciamento de Nível de Serviço (*SLM – Service Level Management*) em ambientes de TI. Uma métrica – aqui chamada de “Índice de Disponibilidade de Sistema - IDS” – é considerada como uma maneira de medir a funcionalidade dos sistemas de ambientes de TI. A fim de validar o protótipo desenvolvido para o sistema proposto, um estudo de caso é apresentado e os principais resultados são analisados.

Palavras-chave: Gerência de Serviços; Sistema de Gerência; Gerência de Nível de Serviço, Índice de Disponibilidade.

Abstract

The increasing complexity in the networks and applications with the new demands for distributed information technology (IT) services has changed the way systems are managed. The focus has moved from device-oriented management to service-oriented management. Currently, small and large companies demand an IT infrastructure to support their business, and such infrastructure has to be monitored. Quantitative metrics should be measured and used to get a feedback about the IT environment functionalities. This work proposes a web-based network service management system that use open standards and free software to help the practice of Service Level Management (SLM) at IT environments. A metric – here named “Service Availability Index”- is presented as a way to measure the IT environment system functionalities. To validate the prototype developed for the system proposed, a case study is presented and the main results are analyzed.

Keywords: Service Management, Management Systems, Service Level Management, Availability Index.

Capítulo 1

1 Introdução

1.1 Motivação

Com a crescente utilização das redes de computadores, torna-se cada vez maior a necessidade de integração dos sistemas de informação e unidades de trabalho cooperativo dentro das empresas e corporações. As aplicações em ambientes de TI (Tecnologia da Informação) têm sido imprescindíveis na realização de inúmeras tarefas em diferentes ambientes de trabalho. A utilização de tecnologias de telecomunicações e informática, de natureza e porte diferentes, vem atingindo um estágio de grande amadurecimento.

Entretanto, com a utilização dos recursos de rede e a inclusão de inúmeras tecnologias e novas aplicações, a degradação de desempenho vem crescendo na mesma proporção do aumento de sua utilização. O crescimento em dimensão e complexidade dos ambientes de TI traz muitos problemas para os administradores de redes. A gerência de redes aparece, neste contexto, como sendo de vital importância, independentemente do porte do ambiente. É uma atividade que consiste em observar e controlar os eventos de grande importância no cotidiano de operações das instituições.

Devido ao crescimento significativo da complexidade das aplicações e à necessidade de se oferecer serviços de TI distribuídos, o foco de gerência tem se movido da gerência orientada a dispositivos para a gerência orientada a serviços (Rodosek, 2002). Neste contexto, uma eficiente gerência orientada a dispositivos é pré-condição para uma eficaz gerência de serviços de TI.

É importante contextualizar o significado desses dois tipos de gerência no presente trabalho. A gerência orientada a dispositivos, também conhecida como gerência de rede e sistemas, refere-se ao monitoramento e controle de uma infra-estrutura de rede, ou seja, gerenciar informações especificadas em MIBs (*Management Information Bases*) dos dispositivos de rede. Por exemplo, monitorar a utilização de CPU de um servidor WWW (*World Wide Web*),

a disponibilidade e a utilização de uma interface de um roteador através de um protocolo padrão como o SNMP (*Simple Network Management Protocol*).

A gerência orientada a serviços não possui um significado tão específico como a orientada a dispositivos, pois o termo “serviço” pode ter um contexto variável que depende de um problema particular. Por exemplo, o atendimento de uma ordem de serviço em um sistema *intra-desk* em uma organização, a disponibilidade de um canal de comunicação para um usuário na telefonia convencional ou telefonia celular, uma aplicação *web* para declaração do imposto de renda e a disponibilidade de um *link* de acesso à internet banda larga são diferentes tipos de serviços que podem ser oferecidos a usuários. Cada um desses exemplos citados possui métricas como a disponibilidade média e o tempo de resposta médio. No entanto, a forma pela qual tais métricas são monitoradas varia em cada caso. De maneira geral, o termo “serviço” é usado para descrever uma funcionalidade que é oferecida com uma determinada qualidade para os usuários de um determinado sistema de rede (Rodosek, 2002).

O ITIL (*Information Technology Infrastructure Library*) (Itil, 2004), define “gerência de serviço (*Service Management*)” como um agregado de boas práticas de gerência não limitado apenas ao monitoramento e controle dos sistemas computacionais, mas também ao controle dos processos de negócio das empresas.

Como o foco desta dissertação é a gerência de serviços, é importante definir claramente o seu significado. O contexto da gerência de serviços para o presente trabalho consiste na atividade de gerenciar de maneira pró-ativa a disponibilidade (gerência de falhas) e o tempo de resposta (gerência de desempenho) dos serviços dos usuários de ambientes de TI (WWW, DNS, e-mail, ftp, telnet, banco de dados, portas TCP – *Transmission Control Protocol*, entre outros).

A Figura 1 ilustra uma visão macro de quatro níveis de gerência: 1) A camada de Gerência de Rede e Sistemas, mais baixa, integra todos os equipamentos (*hubs, bridges, switches, roteadores, outros*) e servidores através de acompanhamento constante de indicadores. Dados como utilização e matriz de interesse de tráfego entre os elementos são importantes neste contexto; 2) A segunda camada, Gerência de Serviços, explora indicadores relacionados aos serviços básicos que suportam as aplicações do ambiente gerenciado. A disponibilidade e o acesso aos mesmos, acompanhado dos seus respectivos tempos de resposta, são exemplos de indicadores, cujo objetivo central deve ser sempre a busca pela satisfação do usuário e o aumento da produtividade com a utilização de sistemas de informação. Serviços de HTTP e DNS, por exemplo, podem suportar aplicações críticas e, assim, precisam ser monitorados e

gerenciados com atenção; 3) A terceira camada, Gerência de Aplicações, é responsável pelo monitoramento de aplicações que realizam o processamento dos dados de negócio da empresa (suporte ao negócio da empresa), tendo como principais métricas de monitoramento a disponibilidade e o tempo de resposta. Aplicações que funcionam sobre plataformas como CRM (*Customer Relationship Management*), por exemplo, utilizam-se de diferentes sub-serviços a fim de prover um outro serviço de mais alto nível; 4) A camada mais alta, Gerência de Negócio, é responsável pela gerência global do negócio, onde o objetivo final é manutenção do ambiente gerenciado. É nesse nível que são feitos os acordos entre as operadoras e são definidos os objetivos. Neste ponto, os diretores e gestores empresariais estão interessados no retorno (*ROI – Return of Investment*) que toda a estrutura das camadas inferiores proporciona à organização que detém o ambiente.

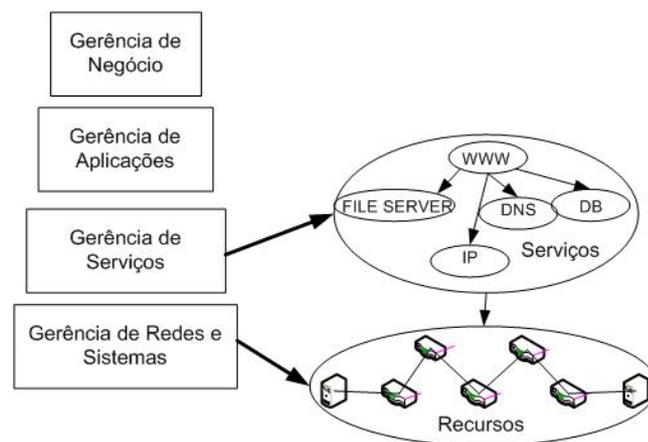


Figura 1 - Níveis de Gerência.

No âmbito atual de gerência de ambientes de TI, grupos de pesquisa buscam melhorar a forma de se realizar a gerência de redes e sistemas através do uso de diferentes tecnologias, como, por exemplo, a tecnologia de agentes móveis, a fim de se obter maior desempenho, flexibilidade e capacidade para essa tarefa (Stephan, 2004) (Mota et al., 2002). Em relação a padrões e modelos para a gerência de serviços, poucas soluções (ferramentas) livres para tal finalidade são encontradas. Dentre alguns trabalhos encontrados na literatura, cita-se (Kim, 1999), (Gaspary, 2002) e os desenvolvidos pelo grupo de pesquisadores do *Munich Network Management Team (SMTF – Service Management Task Force)* (Rodosek, 2002), os quais serão discutidos com mais detalhes no capítulo 3.

Em (Gaspary, 2002) são relacionados alguns exemplos importantes de tarefas de gerenciamento aplicáveis a serviços de rede, que devem ser executadas pelo administrador da rede.

- Monitoração da disponibilidade de serviços: serviços como o DNS (*Domain Name System*) e o DHCP (*Dynamic Host Configuration Protocol*), essenciais ao funcionamento de uma rede TCP/IP, precisam estar permanentemente em execução. Esta tarefa se resume em detectar quando algum desses serviços se encontra indisponível. A ação a ser realizada, imediatamente, é reiniciá-lo.
- Medição do tempo de resposta de uma aplicação em rede: compreende a determinação do tempo de resposta percebido pelos usuários finais de um serviço/aplicação. Essa informação permite ao administrador de rede saber quando expandir o *hardware* da estação servidora, quando o gargalo for a estação, ou reposicionar a estação para um ponto da rede em que não haja congestionamento, quando o alto tempo de resposta for ocasionado pelo atraso imposto pela própria rede.

1.2 **Definição do Problema**

A definição, a programação e a execução de tarefas de gerenciamento de falhas (disponibilidade) e de desempenho (tempo de resposta) de serviços de um ambiente de TI, são o tema central abordado nesta dissertação. As atuais dificuldades para a execução destas tarefas de gerenciamento são:

- Carência de plataformas livres/abertas voltadas para o gerenciamento de serviços. A grande maioria das plataformas existentes não atende ao gerenciamento de serviços, pois o objetivo consiste no gerenciamento de redes e sistemas;
- Soluções comerciais existentes no mercado, apesar de poderosas, não disponibilizam facilmente o monitoramento de serviços com a perspectiva do usuário (simulação de um usuário real). Em algumas soluções, a forma como os serviços são gerenciados não é tão eficiente, pois se limita a tentativas de abertura de conexão TCP nas portas monitoradas (“pings nas portas”) para identificar se um determinado serviço de rede está disponível ou não. No entanto, o ideal é monitorar o serviço simulando a utilização do mesmo como um usuário real. Por exemplo, em vez de apenas verificar através de um *socket* se a porta 25 TCP de um servidor SMTP está ativa, simular uma conexão de um cliente SMTP e executar alguns comandos básicos especificados na RFC 821, a fim de verificar se o servidor SMTP responde corretamente aos comandos do cliente;

- As soluções comerciais são fechadas e de alto custo tornando-as inacessíveis às pequenas e médias empresas, que também necessitam dos serviços de gerência para os seus ambientes de TI. Em (Gaspar, 2002), há uma caracterização funcional de algumas ferramentas de gerência de serviços comerciais existentes. Dentre as ferramentas analisadas até então pelo grupo de gerência do LPRM-UFES, o *Nagios* (Nagios, 1999), além de ser uma ferramenta livre, se mostra como a ferramenta mais flexível para a gerência de serviços. No entanto, alguns aspectos negativos do *Nagios* são a sua difícil usabilidade e o não monitoramento nativo do tempo de resposta dos serviços.

1.3 Objetivos da Dissertação

Devido à necessidade de soluções voltadas exclusivamente para a gerência de serviços, este trabalho tem como objetivo apresentar a implementação do protótipo de um sistema de gerência de serviços de rede (camada 2 da Figura 1) com interface baseada em *Web*, capaz de monitorar de forma centralizada e com a perspectiva do usuário a disponibilidade e o tempo de resposta de um conjunto de serviços de uma rede IP. O objetivo do sistema é auxiliar a atividade de gerenciamento de redes permitindo o Gerenciamento de Nível de Serviço (*SLM - Service Level Management*) em ambientes de TI. O sistema deverá realizar gerência centrada em serviços monitorando um conjunto determinado de serviços de uma rede IP, a partir da implementação de uma Biblioteca de Serviços¹.

O protótipo do sistema implementa um conjunto de componentes de modo que o administrador da rede necessite apenas especificar um conjunto de informações através de uma interface *Web*, ao invés de editar arquivos de *scripts*, para iniciar o processo de monitoramento para um determinado serviço. Do ponto de vista prático, uma solução desse tipo é de grande utilidade.

A solução proposta é concebida para funcionar em *intranets*, ou seja, dentro de redes organizacionais privadas monitorando os serviços das redes locais. Não é contemplado na solução o tratamento de questões como a existência de um *firewall* entre a estação de gerência e o serviço monitorado.

¹ Biblioteca de Serviços: tem como significado, neste trabalho, uma API (*Application Programming Interface*) Java, por exemplo, capaz de monitorar a disponibilidade e o tempo de resposta de um conjunto de serviços de rede (WWW, DNS, FTP, Telnet, Banco de Dados, SMTP, POP3, IMAP, outros) simulando um usuário real.

Uma métrica aqui denominada de “*Índice de Disponibilidade de Sistema - IDS*” também é considerada, e calculada pelo protótipo implementado. Tal métrica tem como principal objetivo fornecer um *feedback* quantitativo do funcionamento global do ambiente de TI monitorado.

O protótipo do sistema apresentado neste trabalho utiliza apenas soluções abertas, ou seja, software livre, bibliotecas abertas, plataformas de desenvolvimento e padrões abertos. A opção pelo uso de software livre deve-se ao fato do grupo de gerência do LPRM-UFES almejar desenvolver um conjunto de soluções de gerência de baixo custo, acessível às pequenas e médias empresas que também necessitam dos serviços de gerência. Dentre os trabalhos desenvolvidos e publicados pelo grupo de gerência de redes do LPRM, merecem destaques:

- (Monteiro, 2000): uma metodologia para gerencia de desempenho de redes;
- (Drago, 2001): uma ferramenta computacional para auxílio à prática de gerência de desempenho de redes de computadores (NetMonitor);
- (Carmo et al., 2004): uma solução de conectividade eficiente, segura e de baixo custo;
- (Ramos, 2004): uma ferramenta computacional para auxílio à prática de gerência de desempenho de redes de computadores (NetMonitor V2);
- (Villaça et al., 2004): uma arquitetura de gerência pró-ativa de falhas e desempenho de redes baseada em *web* (NOC – *Network Operations Center*).

Os principais diferenciais do sistema apresentado neste trabalho são: 1) permitir ao administrador da rede escalar o monitoramento de um determinado serviço de rede através de uma interface *Web*, sem a necessidade de editar arquivos de *scripts* para tanto; 2) visualizar relatórios, via interface *Web*, de disponibilidade e tempo de resposta de serviços monitorados; 3) Gerar relatórios de IDS (“Índice de Disponibilidade de Sistema”) para visualizar em nível mais alto o desempenho do funcionamento do ambiente de TI monitorado. Através do relatório de IDS, é possível identificar qual serviço mais influencia no desempenho do ambiente de TI que suporta o negócio de uma empresa.

1.4 Organização da Dissertação

Além desta introdução, o Capítulo 2 apresenta os principais conceitos teóricos relacionados à gerência de redes e serviços. O protocolo SNMP é apresentado destacando sua viabilidade

para a gerência de serviços de rede através de algumas MIBs específicas. Uma discussão sobre Gerência Baseada em Nível de Serviço também é realizada. O Capítulo 3 discute alguns dos principais trabalhos relacionados à gerência de serviços encontrados na literatura pesquisada.

O Capítulo 4 expõe a proposta desta dissertação, que consiste em uma Ferramenta *Web* para Gerência de Serviços de Ambientes de TI. Além da descrição das funcionalidades, o projeto e a arquitetura do sistema também são apresentados. O diferencial e a contribuição da ferramenta proposta também são definidos.

O capítulo 5 detalha um estudo de caso realizado a fim de validar a eficácia e a utilidade do Sistema de Gerência de Serviços proposto. O estudo de caso consiste em monitorar os serviços demandados pelos usuários do LPRM. O capítulo 6 conclui esta dissertação com os resultados deste trabalho, bem como sugerindo trabalhos futuros a fim de dar continuidade ao desenvolvimento dos trabalhos adjacentes a este.

O Apêndice A apresenta os fluxogramas que ilustram como cada tipo de serviço é gerenciado pelo sistema, ou seja, quais tarefas são executadas para verificar a disponibilidade de um determinado serviço. O Apêndice B apresenta os modelos de classes e o modelo lógico do banco de dados do sistema. O Apêndice C apresenta algumas telas de navegação do SMS a fim de ilustrar algumas funcionalidades do sistema.

Capítulo 2

2 Gerenciamento de Redes e Serviços

Neste capítulo, são apresentados alguns conceitos de gerência de redes e serviços, com a intenção de introduzir o conteúdo básico e indispensável para a compreensão do restante da dissertação. Na seção 2.1, são destacados conceitos relacionados à gerência de redes, como: áreas funcionais, arquiteturas de sistemas de gerência de redes e o padrão de gerência SNMP. A seção 2.2 discute o ITIL (*Information Technology Infrastructure Library*) (Itil,2004), que consiste em uma iniciativa para padronização da Gerência de Serviços de TI. Por fim, a seção 2.3 define o significado da gerência baseada em níveis de serviço para o presente trabalho.

2.1 Gerência de Redes

De acordo com (Stallings, 1999), a Gerência de Redes consiste em monitorar e controlar os componentes lógicos ou físicos de uma rede de computadores, assegurando um certo nível de qualidade de serviço. Para realizar essa tarefa, os administradores de redes são geralmente auxiliados por um sistema de gerência de redes. Esse sistema pode ser definido como uma coleção de ferramentas integradas para monitoramento e controle que oferece uma interface única com informações sobre a rede e pode oferecer, também, um conjunto de comandos que são usados para executar as tarefas diárias de um administrador de redes.

Somente através do monitoramento e controle intensivo é possível detectar falhas, problemas de desempenho ou mau funcionamento dos recursos monitorados. Assim, agiliza-se o processo de tomada de providências necessárias para minimizar o impacto nos serviços prestados pela rede causados por estes tipos de problemas.

É importante deixar claro o significado de monitoramento e controle no âmbito da definição de gerência de redes:

- **Monitoramento:** trabalho que consiste em observar e analisar um conjunto pré-definido de indicadores, como, por exemplo, desempenho e falhas. (Stallings, 1999) sugere que o processo de monitoramento siga o seguinte procedimento: 1) definir quais os componentes que devem ser gerenciados e como obter deles as informações necessárias; 2) escolher as ferramentas adequadas para o monitoramento; 3) uma vez

obtidas as informações, definir como serão utilizadas nas diversas áreas funcionais (vide áreas funcionais na seção 2.1.1).

- **Controle:** Seguindo ao monitoramento, é feita uma análise dos indicadores obtidos por informações coletadas dos componentes gerenciados do Ambiente de TI em busca de situações de comportamento anormal. Caso esse desvio seja detectado, parte-se para a etapa de Controle, que consiste na modificação de parâmetros sobre os dispositivos gerenciados, tendo como efeito a execução de ações pré-definidas, visando restaurar a normalidade. Controlar é inicializar, manter e desligar componentes individuais e subsistemas lógicos dentro de uma configuração de máquinas e recursos (Stallings, 1999). Mais especificamente, é usar as informações disponíveis para gerenciar a configuração dos dispositivos de rede. O processo de controle consiste em: 1) obter informações sobre o estado da configuração atual; 2) usar as informações coletadas no processo de monitoração para modificar a configuração; 3) Armazenar as informações, manter e atualizar informações de inventário e produzir relatórios baseados nas informações coletadas.

2.1.1 Áreas Funcionais da Gerência de Redes

A ISO (*International Organization for Standardization*), através do documento ISO/IEC 7498-4, dividiu as atividades de gerenciamento de redes em cinco áreas funcionais específicas (Figura 2), denominadas: gerência de falhas, gerência de configuração, gerência de desempenho, gerência de segurança e gerência de contabilização. Estas cinco áreas são abreviadas neste texto como FCAPS (*Fault, Configuration, Accounting, Performance and Security*).

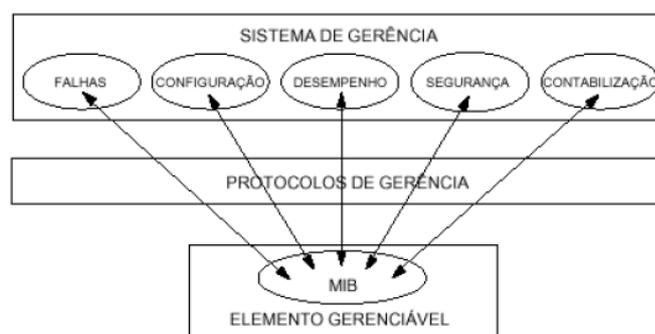


Figura 2 - Áreas Funcionais de Gerência.

Em (Rodosek, 2002), uma discussão sobre áreas funcionais de gerência de serviço é realizada através de uma aproximação com as áreas funcionais concebidas para gerência de redes. É

deixado claro que as características das áreas funcionais para gerência de redes também são aplicáveis à gerência de serviços. No entanto, para este último, características são adicionadas/estendidas.

A seguir, são caracterizadas as áreas funcionais de gerenciamento de redes (orientada a dispositivo) propostas pela ISO, e juntamente são destacadas as aproximações para gerência de serviços (orientada a serviços), discutidas em (Rodosek, 2002):

Gerência de Falhas

- **ISO:** corresponde às atividades necessárias para detecção, isolamento e correção de operações anormais. Os recursos de gerenciamento de falhas procuram mostrar o número de falhas, uma classificação prévia, o momento de ocorrência e a sua localização. O procedimento clássico determina: detectar a falha, isolar e reparar os componentes envolvidos de forma a retornar a rede ao seu estado normal com o máximo de agilidade.
- **Aproximação para gerência de serviços:** diferentemente da gerência de falhas orientada a dispositivos, o monitoramento e controle da qualidade com a qual o serviço está sendo provido faz parte da gerência de falhas orientada a serviços. Isto é, não basta o serviço estar disponível para o usuário, é necessário que a disponibilidade seja acompanhada de uma qualidade acordada. Na gerência de falhas orientada a serviços, as tarefas de detectar, diagnosticar e reparar o problema são mais complexas, pois consiste em tratar a várias dependências entre os componentes de um serviço (outros serviços e/ou recursos de rede), pois um serviço pode, hierarquicamente, depender de outros serviços e/ou dispositivos de rede. Logo, resolver um problema de disponibilidade de serviço consiste analisar o serviço em si, os sub-serviços e os recursos físicos de rede que dão suporte ao serviço. Em (Rodosek, 2002), são apresentados em mais detalhes os requisitos para gerência de falhas orientada a serviço.

Gerência de Desempenho

- **ISO:** tem como tarefa monitorar o comportamento dos componentes da rede. O resultado deste monitoramento pode ser utilizado para planejar e controlar a qualidade de serviço na rede. Pelo uso de meios estatísticos pode-se promover ações para antecipar problemas iminentes, dada uma possível degradação dos indicadores de desempenho. De acordo com (Stallings, 1999), cada componente compartilhado

demanda atenção para que sejam respeitados os seus limites pelos serviços que utilizam seus recursos. O sistema deve prover o estabelecimento de limites de comportamento permitidos para cada componente e emitir notificações para motivar uma ação, quando esses valores forem atingidos.

- **Aproximação para gerência de serviços:** a tarefa de gerência de desempenho orientada a serviços se presta de certa forma a dar um suporte à gerência de falhas, pois é responsável por monitorar e garantir que um serviço seja provido e operado com a qualidade de serviço especificada em SLAs. A gerência de desempenho orientada a serviço, às vezes, também é referenciada como Gerência de Nível de Serviço (SLM).

Gerência de Configuração

- **ISO:** compreende a instalação, inicialização, modificação e registro de parâmetros de configuração. As redes de computadores podem constituir-se de milhares de equipamentos e dispositivos dispersos por vários locais físicos, muitos deles com mudanças frequentes de localização. É fundamental que sejam conhecidas, por exemplo, a localização de cada componente da rede, suas especificações técnicas e configurações e os responsáveis pelo seu funcionamento.
- **Aproximação para gerência de serviços:** a gerência de configuração orientada a serviços é semelhante à gerência de configuração orientada a dispositivos, porém mais difícil, pois envolve um maior número de atividades. Configurar um novo serviço, por exemplo, envolve a configuração de um ou mais dispositivos de rede (recursos) além de outros serviços com os quais o novo serviço depende.

Gerência de Contabilização

- **ISO:** registra as informações sobre a utilização dos recursos com o objetivo de quantificá-los para efeito de distribuição de custos, tarifação, planejamento de capacidade e verificação de cotas de utilização.
- **Aproximação para gerência de serviços:** semelhantemente à gerência de contabilização orientada a dispositivos, a gerência de contabilização orientada a serviços tem a quantificação da utilização dos serviços como uma de suas principais atividades, a fim de determinar o custo do serviço provido.

Gerência de Segurança

- **ISO:** é responsável pela definição e implementação de políticas de segurança na rede. A informação que trafega em uma rede somente deve ser acessada por pessoas ou aplicações que possuam a devida autorização. As informações sensíveis para o negócio devem ser mantidas de forma segura, prevenindo-se contra os acessos indevidos, seja para leitura ou para alteração.
- **Aproximação para gerência de serviços:** além das características da gerência de segurança orientada a dispositivos, inclui a tarefa de prover serviços seguros, ou seja, o serviço provido deve embutir mecanismos que garantam segurança no seu acesso.

Dentre as áreas funcionais apresentadas anteriormente, a ferramenta proposta no presente trabalho (vide seção 1.3) está focada nas gerências de falhas e de desempenho de serviços, através do monitoramento da disponibilidade (falha) e do tempo de resposta (desempenho), acompanhado dos respectivos parâmetros de qualidade de serviço. Por exemplo:

- Disponibilidade: disponibilidade média, *Mean Time To Repair*, *Mean Time To Fail*, *Mean Time Between Fail* (Menascé, 1998);
- Tempo de Resposta: média do tempo de resposta.

A gerência de configuração não foi abordada no presente trabalho. No entanto, esta área pode surgir como um trabalho futuro para tomada de ações automáticas, ou seja, detectado um problema de falha ou desempenho, tomar ações que possam identificar e corrigir possíveis problemas de configuração em um sistema a partir da análise de bases de conhecimentos de problemas.

Em (Nogueira, 2002), é apresentado um trabalho onde ações automáticas informam ao administrador do sistema uma seqüência de passos/comandos, a fim de encontrar e solucionar a causa raiz de uma falha detectada no ambiente monitorado. As ações automáticas são geradas a partir de bases de conhecimento que descrevem um mapa de dependência dos serviços do ambiente monitorado.

A gerência de contabilização também não foi abordada no presente trabalho, pois a ferramenta proposta foi concebida inicialmente para gerenciar serviços em ambientes de redes privadas. Neste contexto, a questão de tarifação (*billing*) não é muito comum de se aplicar. Logo, não é escopo deste trabalho a contabilização de utilização de um determinado serviço por um determinado usuário, por exemplo.

Da mesma forma, aspectos relativos à área de gerência de segurança não foram abordados no presente trabalho, já que o objetivo não é controlar o nível de segurança dos serviços providos e nem a políticas de acesso aos serviços, por exemplo.

2.1.2 Arquitetura de Gerência de Redes

A arquitetura geral dos sistemas de gerência de redes apresenta quatro componentes básicos: elementos gerenciados, estações de gerência, protocolos de gerência e informações de gerência (Lopes, 2002). A Figura 3 ilustra cada um destes componentes e a seguir tem-se as descrições dos mesmos.

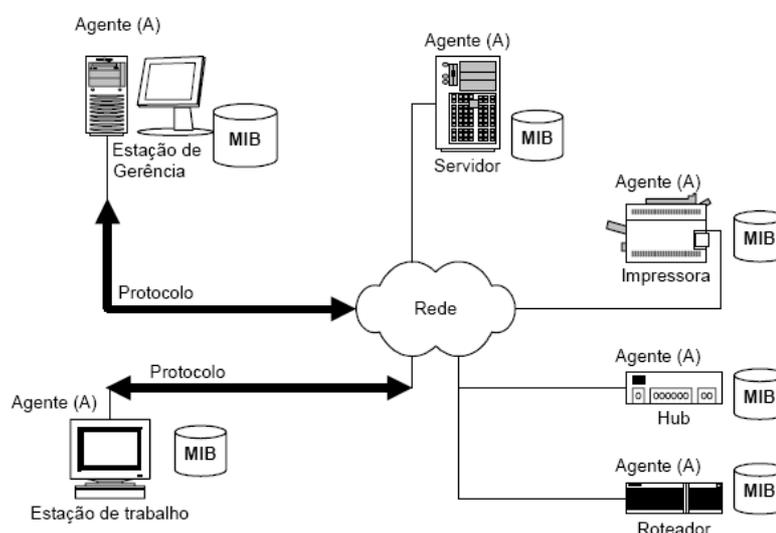


Figura 3 - Arquitetura Geral de Sistema de Gerência.

Agentes: os elementos gerenciados possuem um software especial chamado Agente. Este software permite a monitoração e o controle por uma ou mais estações de gerência, respondendo às solicitações dos indicadores solicitados.

Estação de Gerência: em um sistema de gerência, deve haver, pelo menos, uma Estação de Gerência (ou Gerente). Em sistemas distribuídos ou hierárquicos, existem dois ou mais gerentes. Em sistemas centralizados – mais comuns – existe apenas um. A estação de gerência contém o software que interage diretamente com os agentes nos elementos gerenciados (o software Gerente), com o objetivo de monitorá-los ou controlá-los. O software gerente deve oferecer uma interface para uso pelos seus usuários.

Protocolo: só é possível a troca de informações entre gerentes e agentes, quando ambos usam a mesma linguagem. A linguagem recebe o nome de Protocolo de Gerência. Esse protocolo deverá permitir operações de monitoramento (leitura) e controle (escrita).

MIB (*Management Information Base*): gerentes e agentes podem trocar informações entre si, mas não qualquer tipo de informação. A MIB corresponde ao conjunto das informações de gerência disponíveis em um agente e define os dados que podem ser referenciados em operações do protocolo de gerência.

2.1.2.1 Arquitetura de Sistemas de Gerência de Redes

Um Sistema de Gerência de Redes (*NMS – Network Management System*), ou simplesmente Gerente, pode utilizar diversas arquiteturas, para implementar a sua funcionalidade. Segundo (Leinwand, 1996), são mais comuns as arquiteturas centralizada, hierárquica ou distribuída. A Figura 4 ilustra cada uma destas arquiteturas e a seguir tem-se suas respectivas descrições.

Arquitetura Centralizada

Neste modelo de arquitetura existe apenas um único Gerente, estrategicamente localizado, de forma a ser capaz de gerenciar todos os elementos do ambiente, conforme mostra a Figura 4(a). O sistema contém também um banco de dados único e centralizado. A unicidade do Gerente e a sua localização centralizada fazem com que ele seja o único responsável por toda a geração de alertas e eventos, coleta e administração das informações de todos os elementos.

Esta arquitetura tem, como grande vantagem, a simplificação do processo de gerência, uma vez que a informação necessária está toda concentrada em um único ponto, facilitando a localização de erros e a correlação dos mesmos. Outra vantagem é a segurança no que diz respeito ao acesso às informações, pois há necessidade de se controlar apenas um único ponto de acesso. Como grandes desvantagens, estão a maior concentração da probabilidade de falhas em um único elemento (o Gerente) e a necessidade de duplicação total da base de dados para redundância do sistema (Leinwand, 1996).

Arquitetura Hierárquica

Na Arquitetura Hierárquica existe um servidor que centraliza as informações dos elementos gerenciados no ambiente, porém existe, além disso, um conjunto de outros servidores que podem atuar como clientes deste servidor central, conforme mostra a Figura 4(b). Algumas funcionalidades são de responsabilidade do servidor central, enquanto outras são dos servidores subordinados. Por exemplo, o administrador do ambiente pode dividi-lo em segmentos distintos, em que a gerência de cada um destes fica sob a responsabilidade dos servidores clientes, e o servidor central atua centralizando os dados coletados pelos clientes.

Neste modelo, a base de dados de gerência continua centralizada, mantendo-se o mesmo problema de concentração da alta probabilidade de falhas em um único ponto, porém com as seguintes vantagens incorporadas: a gerência não depende exclusivamente de um único sistema (servidor) e há uma distribuição das tarefas de gerência.

O modelo hierárquico é o modelo mais adequado quando se está trabalhando com a gerência no padrão RMON (*Remote Network Monitoring*) (Leinwand, 1996).

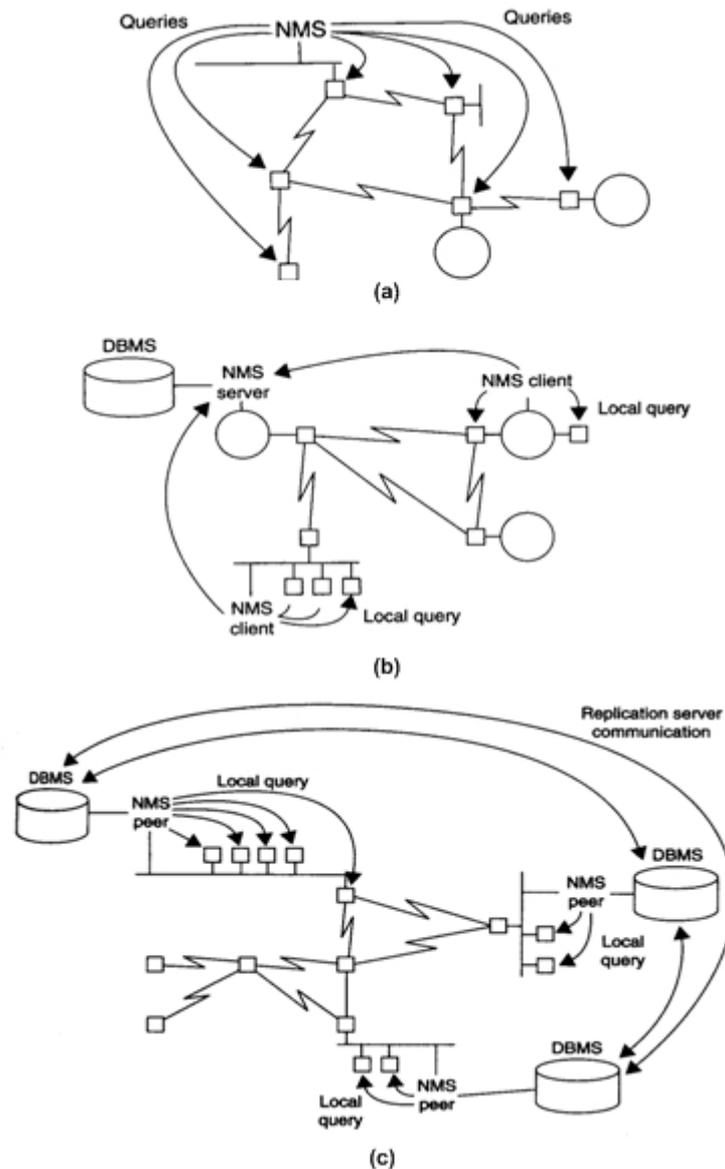


Figura 4 - Arquiteturas de Sistemas de Gerência de Redes.

Arquitetura Distribuída

A Arquitetura Distribuída combina características das arquiteturas centralizada e hierárquica, porém, ao invés de possuir um único servidor ou um conjunto formado por clientes/servidor

de gerência, o modelo distribuído utiliza-se de vários servidores num modelo ponto-a-ponto, em que não há hierarquia entre eles e nem centralização da base de dados, conforme Figura 4(c).

Um dos servidores é escolhido como “líder” do grupo e centraliza somente a tomada de decisões e emissão de relatórios, eventos e alarmes. Cada servidor é responsável individualmente por uma parte (ou segmento) da rede gerenciada, possuindo em sua própria base de dados informações de todo o ambiente, o que lhe permite analisá-lo de forma completa e reportar-se ao líder, quando necessário.

A característica principal deste modelo é a distribuição das tarefas de gerência e da base de dados para cada servidor na arquitetura, distribuindo assim também a probabilidade de falhas entre os diversos servidores e evitando a dependência de um único sistema. O maior problema deste modelo, no entanto, está na necessidade de replicação e sincronismo constante da base de dados nos diferentes sistemas. Esta replicação é normalmente custosa, tanto no que se refere ao consumo de rede, quanto ao consumo de recursos dos servidores (processador, memória, disco, entre outros) (Leinwand, 1996).

A ferramenta proposta no presente trabalho utiliza a arquitetura centralizada (vide seção 4.2), pois um requisito básico para o sistema proposto é sua simplificação do processo de gerência de serviços. No desenvolvimento deste trabalho, foi especificado, como requisito, uma ferramenta que não exigisse do administrador da rede a necessidade de realizar qualquer tipo de configuração nos servidores a fim de iniciar o monitoramento de algum serviço.

Dentre os quatro elementos básicos que compõem a arquitetura de gerência de redes apresentada na seção 2.1.2, o sistema proposto neste trabalho possui apenas a “Estação de Gerência (ou Gerente)”.

2.1.3 O Protocolo SNMP

O modelo de gerência da arquitetura TCP/IP implementa o protocolo SNMP (*Simple Network Management Protocol*), que é mais simples que o modelo de referência de gerência OSI. O protocolo SNMP possui uma arquitetura de computação baseada no modelo gerente/agente composta dos mesmos quatro componentes mostrados na seção 2.1.2, onde o protocolo utilizado é o SNMP que localiza-se na camada de aplicação.

Dada a diversidade de dispositivos gerenciáveis no mercado, a partir do início da década de 80, múltiplos protocolos proprietários foram definidos e todos de difícil integração entre si. A

solução para isso foi a adoção de uma padronização, pela implementação de serviços comuns de gerência.

O SNMP foi desenvolvido para prover aos diversos fabricantes de equipamentos um padrão para dar suporte às aplicações de gerenciamento de redes interconectáveis via TCP/IP (Stallings, 1999). O SNMP é o protocolo de suporte às aplicações de gerenciamento mais implementado em todo o mundo, sendo suportado por um número crescente de dispositivos.

Embora o sistema gerência de serviços proposto nesta dissertação não implemente o monitoramento baseado no SNMP, a apresentação do protocolo justifica-se pelo fato de trabalhos relacionados interessantes (seção 3.1 e seção 3.3) utilizarem tal padrão. Além disso, a utilização do SNMP para gerência de serviços é sugerida como um trabalho futuro.

As subseções seguintes descrevem, de maneira geral, os componentes e o funcionamento do padrão SNMP. Uma referência completa e detalhada sobre o assunto pode ser obtida em (Stallings, 1999).

2.1.3.1 Gerente/Agente de Gerência

O modelo utilizado pelo SNMP é definido na RFC1157. O paradigma Gerente/Agente estabelece um modelo no qual um Gerente troca mensagens com um ou mais Agentes, enviando requisições e recebendo respostas, conforme ilustra a Figura 5. Ambos, Gerente e Agente, são aplicações que provêm funções e serviços gerenciais. Um Agente provê o acesso e a manipulação das informações de gerenciamento (contidas na MIB). O Gerente, atuando sobre o Agente, pode efetuar ações que repercutem diretamente na operação dos objetos gerenciados de cada dispositivo. Em adição, o Agente pode reportar ao Gerente situações que foram pré-definidas como não sendo situações normais do sistema.

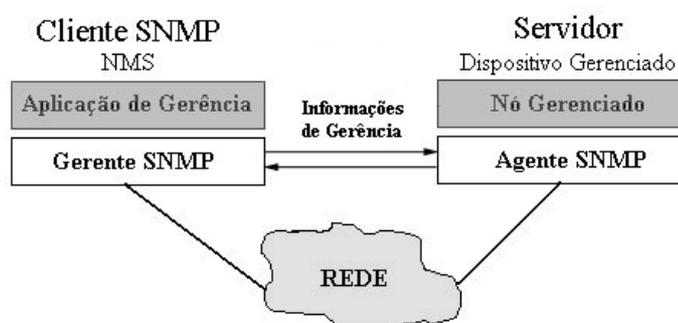


Figura 5 - Modelo Agente/Gerente (Cliente/Servidor).

Os Nós Gerenciados compreendem os elementos que sofrerão monitoramento e controle durante a atividade de Gerência. São exemplos de nós gerenciados: 1) estações de trabalho,

servidores ou impressoras; 2) roteadores; 3) *bridges*, *hubs*, ou *switches*; 4) banco de dados, *firewall* e sistemas operacionais.

A Estação de Gerência pode ser um computador que contém aplicações que realizam tarefas de gerenciamento, mantendo uma comunicação constante com os seus agentes por meio do protocolo de gerência, no caso, o SNMP.

2.1.3.2 Pollings e Traps

Polling é uma interação do tipo pedido/resposta entre o gerente e os agentes, *trap* é uma comunicação iniciada pelo agente, informando ao gerente a ocorrência de alguma modificação em seu estado corrente. É preciso definir uma política para a frequência de *polling* pelo gerente. *Pollings* muito freqüentes tendem a caracterizar bem o processo de amostragem estatística, porém aumentam a sobrecarga na rede devido às operações de monitoramento. Se a situação inversa for adotada, reduz-se o tráfego na rede, porém há uma perda na caracterização da amostragem, o que pode gerar a perda de eventos importantes no comportamento do sistema monitorado.

Se um gerente é responsável por vários agentes, e cada agente mantém diversos recursos, torna-se impraticável a obtenção de informações dos agentes para todos os objetos. Para contornar esse problema, existe uma técnica conhecida como *trap-direct polling* ou *event report*. Nesse caso, cabe ao agente reportar eventos não usuais às estações de gerência quando os mesmos acontecem. (Stallings, 1999)

2.1.3.3 Base de Informações de Gerência

A Base de Informações de Gerência (*Management Information Base - MIB*) é um conjunto de especificações expressas por meio da ASN.1 (*Abstract Syntax Notation One*). Essas especificações descrevem as estruturas das informações gerenciais pelo esquema chamado SMI (*Structure of Management Information*). Tais informações representam objetos que constituem uma visão lógica dos recursos de hardware e software dos nós gerenciados da rede, os quais podem ser manipulados diretamente por um agente. Tais recursos vão desde o meio de transmissão até detalhes dos vários protocolos (Stallings, 1999). Os objetos das MIBs podem ser considerados como uma coleção de variáveis, em conformidade com as suas especificações SMI, RFC1155 e RFC1212, cujos valores podem ser manipulados pelo Agente por requisição de um Gerente.

Inicialmente, o protocolo SNMP padronizou uma MIB chamada MIB-I cujas especificações, posteriormente, foram ampliadas e passaram a compor um novo padrão chamado MIB-II, RFC 1213.

As MIBs são formadas por um conjunto de objetos que caracterizam um dispositivo gerenciável. Esses objetos são identificados por uma seqüência de números inteiros, também chamada de Identificador do Objeto (*Object Identifier - OID*).

2.1.3.4 Operação do Protocolo

O formato de operação do protocolo é bastante simples e corresponde à troca informações por envio e recebimento de mensagens. Uma mensagem SNMP carrega consigo uma identificação do tipo de serviço e também quais os parâmetros utilizados nestes comandos. As operações definidas pelo protocolo permitem a alteração ou inspeção de variáveis na MIB (Figura 6). Seguem as operações do SNMP:

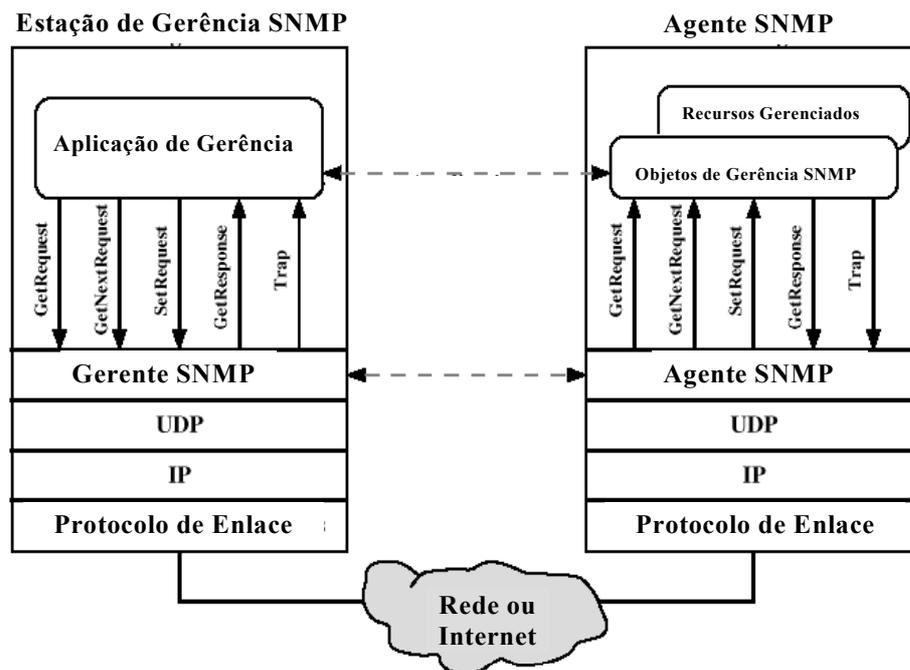


Figura 6 - Arquitetura de funcionamento do SNMP.

GetRequest/GetResponse: O gerente solicita o valor de um objeto a um agente, através de *polling*. O comando SNMP *GetRequest* inicia o processo de leitura de informação de um agente remoto, utilizando uma PDU (*Protocol Data Unit*). O resultado da operação, é uma resposta do agente com o valor de uma determinada variável da MIB, *GetResponse*.

SetRequest: O gerente solicita a modificação do valor de um objeto a um agente, através de *polling*. O comando SNMP *SetRequest* inicia o processo de escrita em uma variável na

MIB utilizando uma PDU (*Protocol Data Unit*). A sua função é alterar o valor de uma variável por um agente em uma determinada MIB.

Trap: Operação que permite ao agente, em situações pré-definidas, enviar uma mensagem não solicitada ao gerente. As *traps* notificam as estações de gerência em situações anormais. Elas devem ser definidas no momento da construção da MIB, notificando quais são as *traps* permitidas para aquele agente.

Get-NextRequest: O gerente solicita o próximo objeto na MIB ao agente, baseado na ordem lexicográfica da árvore hierárquica. A operação *GetNext* utiliza uma PDU (*Protocol Data Unit*) *Get-NextRequest* e é utilizada para recuperar elementos da MIB sem a necessidade de se ter um OID do objeto. Para cada variável contida na PDU, a instância lexicograficamente posterior é retornada com o seu respectivo valor.

2.2 Gerência de Serviços

Durante as pesquisas para realização do presente trabalho, não foram encontrados padrões amplamente difundidos (como o SNMP para gerência de redes) que pudessem definir tecnicamente a gerência de serviços, ou seja, padrões que fornecessem suporte para definição e gerenciamento de um serviço de TI, provido a um usuário. Entretanto, foram encontradas iniciativas para padronização da Gerência de Serviços de TI. Dentre estas, destaca-se o ITIL (Itil, 2004).

2.2.1 ITIL (*Information Technology Infrastructure Library*)

O ITIL (*IT Infrastructure Library*) provê um conjunto de melhores práticas (“*best practices*”) voltado para a Gerência de Serviços de TI. O ITIL foi projetado e desenvolvido inicialmente na década de 80 pelo *United Kingdom Government*, mas recentemente foi revisado e melhorado pelo OGC (*Office of Government Commerce*) (Itil, 2004). O ITIL está sendo muito bem aceito como uma solução organizacional para gerenciamento, suporte e provisionamento de serviços de TI e infraestrutura tecnológica, que suportam os negócios das organizações.

De acordo com o ITIL, a Gerência de Serviços (*Service Management*) consiste em um conjunto de boas práticas não limitadas ao monitoramento e controle de sistemas computacionais, mas também em um conjunto de processos relacionados ao gerenciamento e provisionamento de serviços de TI.

O ITIL parte do princípio de que a informação é o mais importante recurso estratégico que uma organização deve gerenciar. Daí a necessidade de um *framework* que defina os processos para gerenciamento das informações. Com isso, vê-se a dependência entre os sistemas de TI e os objetivos de negócio de uma organização.

O ITIL, como um *framework* de gerência, não trata somente processos relacionados ao monitoramento e ao controle de sistemas computacionais, mas também os processos de negócio de uma empresa. Ou seja, o planejamento de novas aplicações e o levantamento de requisitos de novos serviços para atender o negócio das organizações e seus respectivos parceiros, também constitui os processos definidos no ITIL.

O ITIL consiste de um conjunto de sete módulos e cada módulo é formado por um conjunto de processos. Este conjunto de módulos e processos define um *framework* para melhores práticas para Gerência de Serviços de TI. A Figura 7 apresenta uma visão geral do *framework* ITIL.

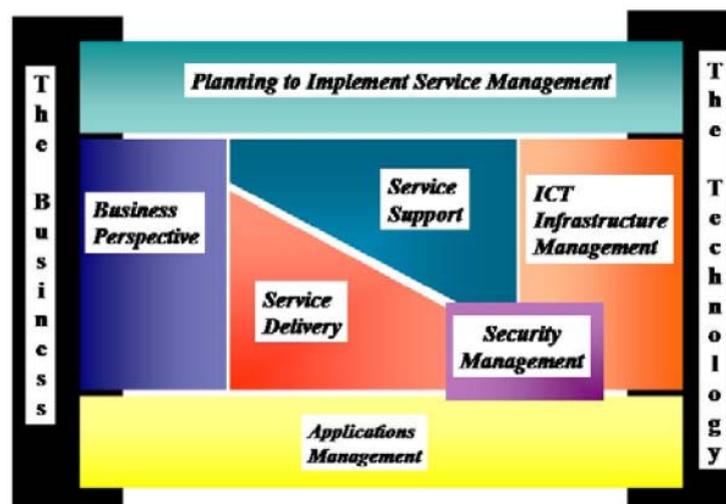


Figura 7 - O *framework* ITIL (Itil, 2004).

A seguir é apresentada uma descrição sucinta de cada módulo do *framework*.

- ***Service Delivery***: contempla os processos necessários para a disponibilização da qualidade dos serviços de TI e os processos que têm a finalidade de melhorar a qualidade dos serviços de TI disponibilizados;
- ***Service Support***: contempla os processos relacionados com suporte diário e a manutenção das atividades para provisionamento de serviços de TI;
- ***Information Communication Technology Infrastructure Management (ICT IM)***: contempla os processos que tratam os aspectos de ICT IM a partir dos requisitos de

negócio, ou seja, gerenciar, administrar, projetar, planejar, otimizar e operar o ambiente de tecnologia de comunicação a fim de suportar os serviços de TI. Isto significa que a gerência de redes também faz parte da gerência de serviços de TI;

- ***Planning to Implement Service Management:*** contempla os processos que tratam das tarefas relacionadas com planejamento, implementação e aperfeiçoamento da Gerência de Serviços em uma organização;
- ***Application Management:*** contempla os processos que descrevem como gerenciar aplicações em sua implantação, durante seus estágios no ciclo de vida e sua definitiva exclusão. O objetivo é garantir que uma aplicação traga vantagens (ganhos) para o negócio da organização;
- ***The Business Perspective:*** contempla os processos que visam guiar e ajudar os funcionários de TI de uma organização a entender como eles podem contribuir para os objetivos de negócio da organização. Como seus serviços podem melhorar e maximizar o desempenho dos negócios da organização;
- ***Security Management:*** contempla os processos relacionados com o planejamento e gerenciamento do nível de segurança para as informações e os serviços de TI. Também inclui a avaliação e o gerenciamento de riscos, vulnerabilidades e custos para as soluções adotadas.

O ITIL define processos (boas práticas) a serem executados a fim de melhorar a qualidade de serviços providos em uma organização. No entanto, nenhum modelo genérico que possa descrever um serviço é apresentado e nenhuma metodologia técnica para se obter informações de um determinado serviço é definida.

Comparando o ITIL com a contextualização dos níveis de gerência apresentados nesta dissertação (seção 1.1), pode-se observar que os processos definidos no ITIL abordam, de maneira geral, todos os níveis de gerência apresentados na Figura 1.

Não é objetivo deste trabalho implementar o protótipo de um *framework* de Gerência de Serviços de TI, mas sim um Sistema de Gerência de Serviços de TI que possa auxiliar a execução de alguns processos de um *framework* de Gerência de Serviços de TI como o ITIL, por exemplo. Mais especificamente, o sistema proposto nesta dissertação poderia ser útil nos processos de *Availability Management* e *Service Level Management* do módulo *Service Delivery* do ITIL (Figura 8), através do monitoramento de disponibilidade e tempo de

resposta dos serviços de uma rede IP, da geração de relatórios e definição de SLAs para as métricas monitoradas.

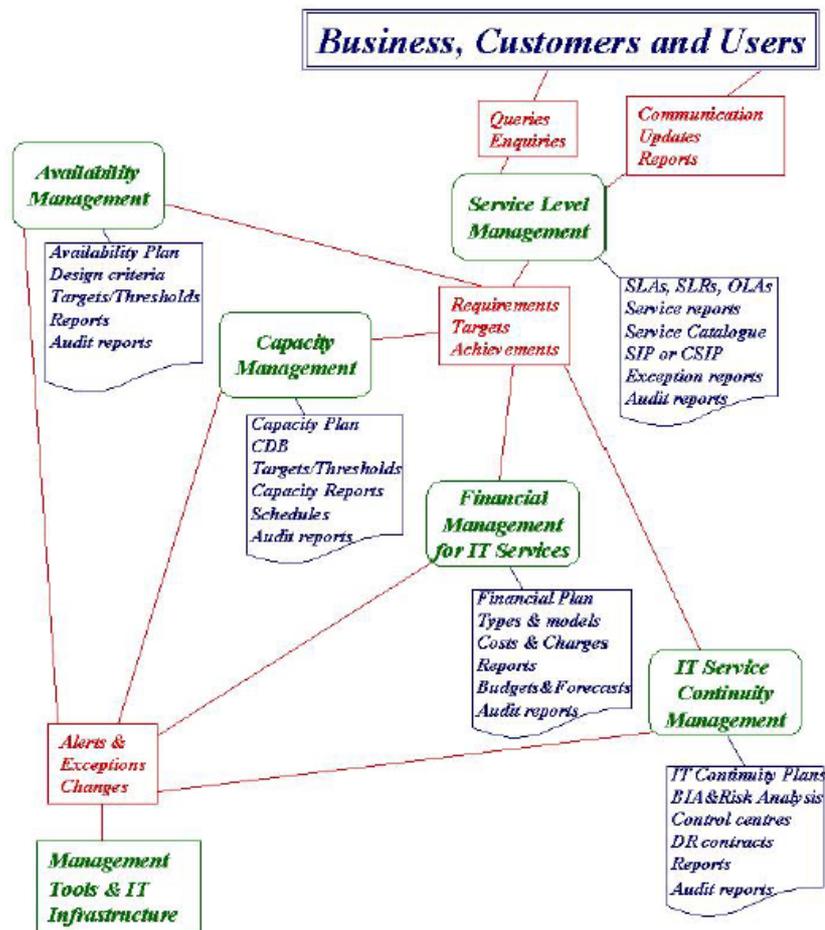


Figura 8 - Processos do módulo *Service Delivery* do ITIL (Itil, 2004).

Os documentos que definem e padronizam o ITIL estão listados abaixo. É importante salientar que esta documentação não é gratuita.

- *BS 15000-1 (2002): IT Service Management (Part 1: Specification for Service Management).*
- *BS 15000-2 (2003): IT Service Management (Part 2: Code of Practice for IT Service Management).*
- *PD 0005 (2003): IT Service Management – A Manager’s Guide.*
- *PD 0015 (2002): IT Service Management – Self Assessment Workbook.*

2.3 Gerência Baseada em Níveis de Serviço (SLM)

O SLM (*Service Level Management*) consiste em um conjunto de atividades para o desenvolvimento, execução e gerenciamento de Acordos de Nível de Serviço (*SLAs – Service Level Agreements*). Comumente usado, SLAs incluem disponibilidade de rede, tempo médio para recuperação de falhas, desempenho de rede, disponibilidade de sistemas e serviços, disponibilidade de *help desk*, desempenho de sistema, tempo de instalação, satisfação do cliente, *throughput* de rede, tempo de resposta de aplicações e serviços, outros (Bissel, et al., 2000)(Muller, 1999).

O gerenciamento de SLAs pode contribuir para determinar como um cliente é respeitado pelos seus prestadores de serviço. O bom relacionamento com esse consumidor e a confiança entre as partes envolvidas podem ser mais importantes do que o próprio valor dos serviços.

A disponibilização de relatórios de falha e de desempenho é um dos requisitos para acompanhamento de SLA. Isto é destacado em iniciativas como o ITIL (Itil, 2004) e trabalhos como o apresentado em (Rodosek, 2002).

O sistema proposto nesta dissertação permite a definição de SLA para os seguintes indicadores:

- Disponibilidade média de um serviço;
- Disponibilidade média de um cluster de serviços;
- Disponibilidade média de um grupo de serviços e/ou cluster de serviços;
- Disponibilidade média de um sistema (conjunto de grupos de serviços e/ou cluster de serviços);
- Tempo de resposta médio de um serviço.

Os indicadores citados acima serão mais bem discutidos no capítulo 4.

Capítulo 3

3 Trabalhos Relacionados

Neste capítulo são apresentados alguns trabalhos relacionados à gerência de serviços encontrados durante as pesquisas realizadas para o desenvolvimento desta dissertação. Na seção 3.1, são mostradas algumas MIBs SNMP para gerência de serviços. Na seção 3.2, uma proposta de um *framework* para gerência de serviços de TI, desenvolvido pelo *Munich Network Management Team* (Rodosek, 2002), é apresentado. Na seção 3.3, uma plataforma de gerenciamento de serviços, aplicações e protocolos de alto nível, “*Plataforma de Gerenciamento Trace*” (Gaspar, 2002), é apresentada como um interessante trabalho que utiliza o padrão SNMP para gerência de serviços. Na seção 3.4, é discutido o uso de software livre para a gerência de serviços, destacando a utilização do *Nagios* (Nagios, 1999) para tal tarefa. A seção 3.5 trata da concepção da ferramenta proposta nesta dissertação levando em consideração os trabalhos relacionados.

3.1 O Protocolo SNMP para Gerência de Serviços

Durante as pesquisas realizadas, observou-se que existem alguns trabalhos/iniciativas que buscam utilizar o padrão SNMP para o gerenciamento de serviços através do desenvolvimento/utilização de MIBs específicas. Esta idéia é interessante dado que o padrão SNMP é amplamente consolidado, devido a sua flexibilidade e eficiência para a gerência de redes. As sub-seções seguintes destacam algumas MIBs que podem ser utilizadas para a gerência de serviços.

3.1.1 MIBs SNMP para Gerência de Serviços Específicos

Existem algumas RFCs (*Request For Comments*) propostas que propõem MIBs específicas para serviços. Isto é, MIBs que possibilitam que o agente SNMP forneça informações de gerência, de um determinado tipo de serviço, para a aplicação de gerência. Cita-se como exemplo:

- DNS – RFC1611: MIB para o monitoramento do serviço de DNS;
- WWW – RFC2594: MIB para o monitoramento do serviço de WWW;
- Mail – RFC2789: MIB para o monitoramento do serviço de Mail;

- RDBMS – RFC1697: MIB para o monitoramento de Sistemas de Gerenciamento de Banco de Dados.

Dentre as MIBs exemplificadas, em relação à disponibilidade (gerência de falhas) e ao tempo de resposta (gerência de desempenho), todas fornecem informações sobre a disponibilidade, mas o mesmo não se pode dizer para o tempo de resposta.

Para o desenvolvimento deste trabalho, a adoção dessas MIBs para o monitoramento de serviços torna-se inviável por três motivos, principalmente:

- Limitação de MIBs para o monitoramento dos diversos tipos de serviços;
- Dificuldade em encontrar módulos que implementem tais MIBs e disponibilizem as informações de gerência para o agente SNMP;
- Dificuldade de monitoramento do tempo de resposta.

3.1.2 MIB Script

A MIB Script, RFC3165, é uma solução de gerenciamento distribuído integrada à arquitetura SNMP. Foi desenvolvida pelo grupo de trabalho DISMAN (*Distributed Network Management*), do IETF (*Internet Engineer Task Force*). A MIB permite que *scripts*, que implementam procedimentos ligados a gerenciamento, sejam transferidos para o local onde devam ser executados. A execução desses *scripts* pode ser disparada remotamente, argumentos podem ser passados a eles e resultados podem ser retornados a quem solicitou sua execução (Gaspary, 2002).

Mesmo com o advento de novas tecnologias como agentes móveis e plataformas de programação distribuída, muitos pesquisadores, no entanto, continuam investindo na arquitetura SNMP devido a dois motivos, principalmente: 1) a simplicidade dessa arquitetura; 2) ao argumento que direcionar as pesquisas em gerenciamento para outras tecnologias justamente no momento em que se chegou a um consenso entre fabricantes é inadequado.

A MIB descreve um *script* como um pedaço de código capaz de ser executado pelo agente remoto que implementa a MIB. Isso significa que quem implementa a MIB pode escolher as linguagens de programação em que os *scripts* são implementados. Essa característica é muito positiva, pois a automação de tarefas de gerenciamento através de *scripts* é um procedimento já amplamente adotado por administradores de rede, inclusive no gerenciamento de serviços.

3.1.2.1 Estrutura da MIB

A operação da MIB Script resume-se a operações realizadas nas seis tabelas que a constituem (Figura 9). Um gerente que deseja delegar um *script* deve primeiro verificar a tabela *smLangTable* (1), que lista todas as linguagens suportadas pela implementação da MIB Script no agente. Além de verificar essa tabela, ele também pode verificar a tabela *smExtsnTable* (omitida da Figura 9), que contém todas as extensões suportadas pelas linguagens instaladas. Essas duas tabelas só podem ser acessadas para leitura, já que é o agente quem possui controle sobre quais são as linguagens e extensões suportadas.

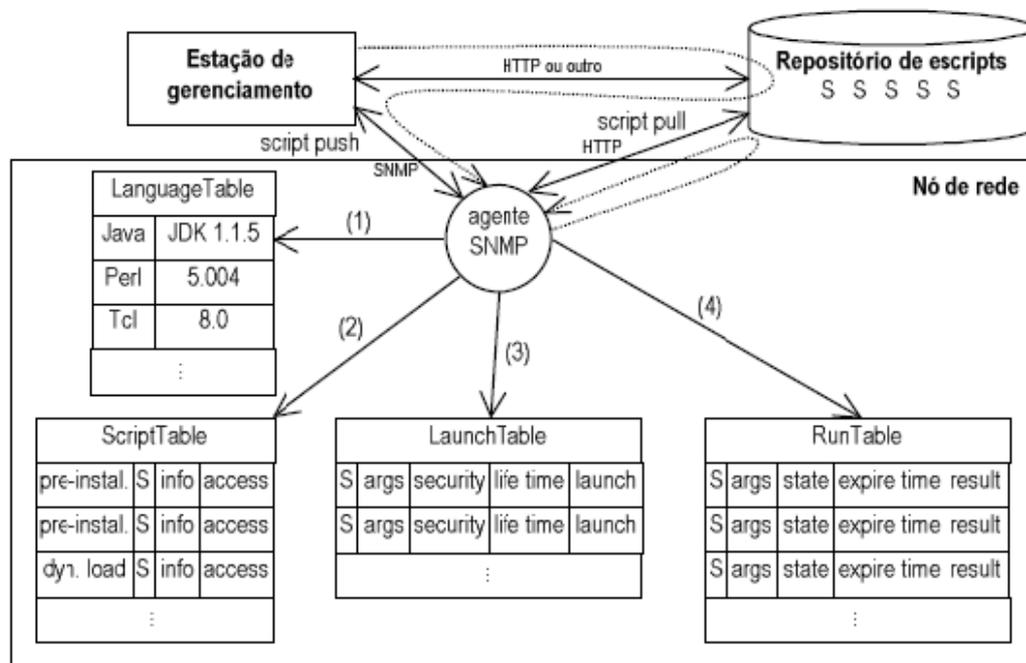


Figura 9 - Estrutura da MIB Script (Gaspary, 2002).

Baseado nas informações obtidas nas tabelas de linguagens e extensões, o administrador deve selecionar um *script* apropriado de um repositório. Esse *script* deve ser registrado em uma nova linha criada na tabela *smScriptTable* (2), que contém uma lista de todos os *scripts* conhecidos pelo agente. Esses *scripts* podem ser instalados permanentemente ou podem ser obtidos através dos modelos *client pull* ou *server push*. O modelo *client pull* requer que apenas um URL seja escrito nessa linha da tabela para que o agente possa fazer o *download* do *script*. Já o modelo *server push* requer que o *script* seja escrito linha por linha na tabela *smCodeTable* (omitida da Figura 9), através de diversas PDUs (*Protocol Data Units*) contendo *SetRequests* SNMP. A implementação desse último modelo não é obrigatória pela

definição da MIB Script. Outras operações, como leitura e a remoção de *scripts*, também são suportadas pela tabela *smScriptTable*.

Para que os *scripts* sejam disparados é preciso que uma entrada seja criada na tabela *smLaunchTable* (3). Nessa tabela é descrito o ambiente de execução para cada *script*, incluindo os argumentos que serão passados, o tempo máximo de vida e a identificação do proprietário do *script*. Um objeto gerenciável nessa tabela, chamado *smLaunchStart*, inicia a execução de um *script* com apenas um *SetRequest* SNMP. Várias instâncias de um mesmo *script* podem ser criadas através de uma única entrada nessa tabela e várias entradas dessa tabela podem apontar para um mesmo *script* na tabela *smScriptTable*. Dessa forma pode-se ter vários *scripts* iguais em execução, mas com argumentos ou permissões diferentes.

Cada *script* disparado possui uma linha criada na tabela *tmRunTable* (4) automaticamente. Essa tabela possui a lista de todos os *scripts* que estão executando ou que terminaram a sua execução recentemente. Através dessa tabela o gerente pode controlar os *scripts* em execução, assim como pode obter os resultados intermediários ou finais gerados por eles. O resultado final é mantido no agente até que o seu tempo de vida expire.

Notificações

Existem duas notificações (*traps*) atualmente definidas na MIB Script. A primeira delas é a *smScriptAbort*, que é disparada sempre que um *script* em execução termina com um código de saída (*smRunExitCode*) diferente de *noError*. A segunda delas é a *smScriptResult*, que pode ser utilizada por *scripts* para notificar aplicações de gerência (gerentes) sobre os seus resultados.

Como mencionado na seção 2.1.2, o trabalho proposto nesta dissertação propõe um sistema de gerência de serviços que utiliza uma arquitetura fortemente centralizada, ou seja, um sistema que busca simplicidade na instanciação de um novo monitoramento excluindo a necessidade de realizar algum tipo de configuração no servidor provedor de serviço. Logo, apesar deste trabalho não utilizar nenhum mecanismo de gerenciamento distribuído, justifica-se a apresentação da MIB Script neste trabalho pelos seguintes motivos:

- Implementar um sistema de gerência de serviços (disponibilidade e tempo de resposta) utilizando uma biblioteca de *scripts* e a MIB Script pode ser uma solução interessante, pois com isso é possível obter distribuição no gerenciamento de serviços utilizando *scripts*. Através da gerência distribuída é possível monitorar um mesmo serviço a

partir de pontos diferentes da rede podendo trazer resultados mais significativos para o administrador da rede. Isto pode ser colocado como um trabalho futuro a este;

- A seção 3.3 destaca uma plataforma que tem como uma de suas funcionalidades a gerência de serviços utilizando a MIB Script. Tal trabalho mostra-se muito interessante.

3.2 A Framework for IT Service Management - MNM Team

Através do trabalho (Rodosek, 2002), o *Munich Network Management Team* (MNM Team) especifica um *Framework* para Gerência de Serviços de TI. Similarmente ao ITIL, apresentado na seção 2.2.1, o trabalho busca especificar os processos necessários para se disponibilizar um determinado serviço com uma qualidade desejada (aprovisionamento de serviços). Diferentemente do ITIL, o *framework* especificado em (Rodosek, 2002) não trata do levantamento de requisitos de novas aplicações para o negócio de uma organização, mas apenas os requisitos de um provedor de serviços que atende às requisições de novos serviços de clientes.

(Rodosek, 2002) discute os requisitos de novos conceitos de gerenciamento devido à necessidade de gerenciar a infra-estrutura de comunicação a fim de garantir a disponibilidade dos serviços, acompanhada de seus respectivos SLAs. De acordo com (Rodosek, 2002), um *serviço* consiste em um novo objeto a ser gerenciado juntamente com a sua integração e dependência da gerência orientada a dispositivos. (Rodosek, 2002) define como desafio fundamental da gerência de serviços de TI o desenvolvimento de uma definição comum de um serviço, ou seja, um *Common Service Model*.

O tema central do trabalho apresentado em (Rodosek, 2002) é a definição de um Modelo de Serviço Comum (*Common Service Model*) como uma base para um *Framework* para Gerência de Serviços de TI.

(Rodosek, 2002) faz uma definição técnica dos processos para provisionamento de serviço. Ou seja:

- Como o modelo de serviço proposto trata a questão da dependência de serviço em relação a outros serviços e a recursos da rede;
- Como o modelo de serviço proposto especifica os parâmetros de QoS em função de outros parâmetros de QoS e QoD (*Quality of Device*);

- Como especificar, obter e gerar relatórios para os clientes, a fim de informar a qualidade dos serviços providos;
- Como tratar as requisições de serviços dos clientes e automatizar a implementação dos serviços requisitados (com a qualidade desejada) sobre os recursos disponíveis – *Dynamic Service Provisioning*. Quais as atividades envolvidas nos processos do ciclo de vida de um serviço (requisição, planejamento, implementação, operação, mudança, exclusão).

Após uma extensiva discussão sobre questões relacionadas com gerenciamento de serviços, e uma abrangente análise de requisitos, (Rodosek, 2002) define o *framework* destacando três módulos principais: *Service Model*, *Dynamic Service Provisioning* e *Quality Management* (Figura 10).

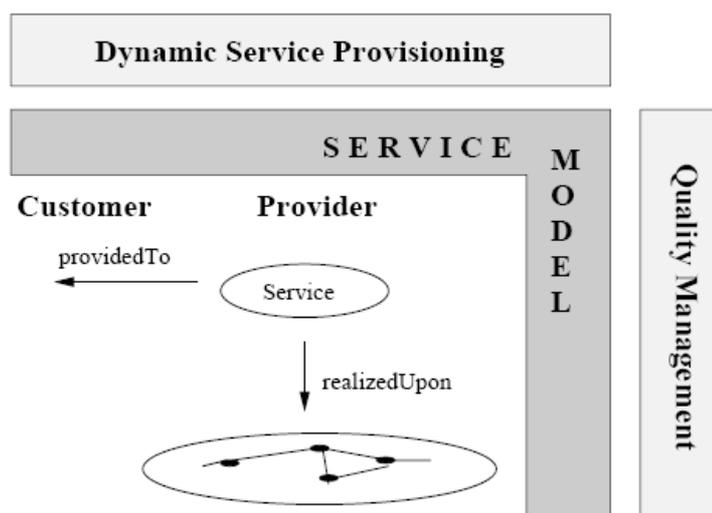


Figura 10 - *Framework* para Gerencia de Serviços de TI (Rodosek, 2002).

- *Service Model*: o modelo de serviço é fundamental para o *framework*, pois é necessário para prover informações básicas dos serviços gerenciados, como por exemplo: descrição do serviço, dependências do serviço, parâmetros de qualidade do serviço e ciclo de vida do serviço;
- *Dynamic Service Provisioning*: o provisionamento de serviço dinâmico trata das atividades relacionadas a disponibilizar os serviços requisitados pelos clientes, implementando-os sobre a infra-estrutura de comunicação do lado do provedor. Ou seja, especifica todos o passos necessários desde o recebimento da solicitação até a disponibilização do serviço suportado pela infra-estrutura. Para isso, as informações disponíveis no modelo de serviço são utilizadas;

- *Quality Management*: a gerência de qualidade é responsável por medir, calcular e acompanhar a qualidade dos serviços providos pelo provedor e disponibilizar os relatórios para os clientes. Medir a qualidade de um serviço significa tratar as dependências deste serviço com outros serviços e recursos de comunicação. (Rodosek, 2002) define uma linguagem chamada QUAL para auxiliar a tarefa de cálculo de QoS considerando as dependências dos serviços. Para isso, as informações disponíveis no modelo de serviço devem ser utilizadas.

O modelo de serviço definido por (Rodosek, 2002) foi útil para o desenvolvimento do presente trabalho, pois serviu como ponto de partida para a modelagem simplificada de um serviço, que é utilizado na implementação do sistema proposto nesta dissertação (vide seção 4.2.1). A Figura 11 ilustra o modelo de serviço (*Service Template Model – Common Service Model*) proposto por (Rodosek, 2002). O diagrama da Figura 11 ainda é estendido em (Rodosek, 2002) para especificar um *Provider-Centric Service Template Model* e um *Customer-Centric Service Template Model*, ou seja, classes são adicionadas ao modelo a fim de inserir características específicas do provedor e do cliente, especificamente. No entanto, o que serviu de inspiração para o presente trabalho foi apenas o *Service Template Model*.

De maneira geral, o diagrama UML (*Unified Modeling Language*) da Figura 11 especifica que um serviço:

- Possui parâmetros de qualidade de serviço;
- Depende de outros serviços e/ou recursos de rede;
- Possui custos;
- Possui funcionalidades;
- As funcionalidades possuem pontos de acesso;
- As funcionalidades possuem parâmetros.

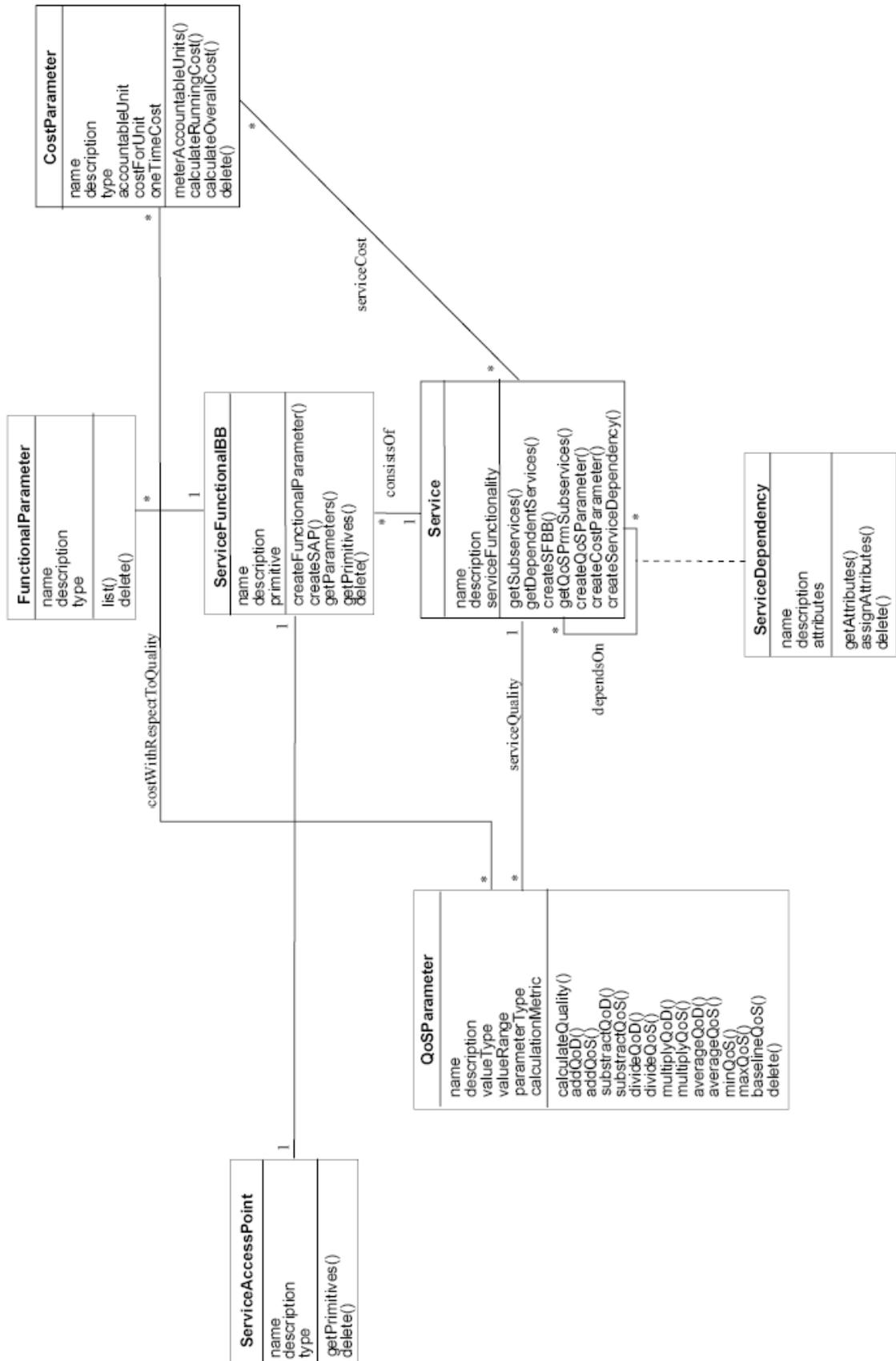


Figura 11 - Modelo de Serviço proposto por (Rodosek, 2002).

(Rodosek, 2002) descreve um modelo de serviço e quais tipos de informação um objeto serviço deve possuir. Define também como este modelo é utilizado pelo *framework*. Realiza alguns casos de uso reais a fim de validar a eficiência do *framework*. No entanto, tudo isto é realizado “*on paper*”, ou seja, nada foi implementado computacionalmente.

A modelagem UML da Figura 11 tem a intenção apenas de apresentar, de forma organizada, as informações pertinentes de um serviço, ou seja, não preocupa-se em apresentar um diagrama de classes que represente uma implementação relativa a um *framework* de gerência.

3.3 **Plataforma de Gerenciamento Trace**

A Plataforma de Gerenciamento Trace foi proposta por (Gasparly, 2002) e consiste em um sistema de gerenciamento distribuído e flexível de protocolos de alto nível, serviços e aplicações em redes de computadores.

O trabalho explora a técnica de monitoração passiva do tráfego de rede para observar eventos significativos ligados a falhas, contabilização, desempenho e segurança. A especificação desses eventos é realizada através de uma linguagem descritiva para representação de traços² de protocolos denominada PTSL (*Protocol Trace Specification Language*), apresentada por (Gasparly, 2002). A linguagem, baseada no conceito de máquina de estados, permite ao gerente determinar seqüências de trocas de pacotes que, quando observadas na rede, indicam a ocorrência de eventos. Por exemplo, a observação de uma requisição DNS respondida por uma mensagem ICMP (*Internet Control Message Protocol*) do tipo *Port Unreachable* é um indicativo de que o *daemon* responsável pelo serviço de resolução de nomes não está sendo executado. Portanto, a observação de uma troca de mensagens como essa no tráfego de rede representa um evento relacionado ao gerenciamento de falhas. Com o conseqüente aumento na granularidade dos eventos observáveis, amplia-se consideravelmente o conjunto de tarefas de gerenciamento que podem ser executadas com a utilização dessa técnica.

A arquitetura de gerenciamento Trace estende a infra-estrutura centralizada de gerenciamento SNMP para um modelo hierárquico ao introduzir gerentes intermediários, além dos tradicionais gerentes, e agentes de monitoração e ação (utiliza a MIB Script). O gerente, a partir da estação de gerenciamento, define tarefas de gerenciamento, traços de protocolos

² Traço: um traço representa uma seqüência pré-definida de troca de pacotes que, quando observada na rede, indica a ocorrência de um evento.

(usando a linguagem PTSL) e ações (*scripts* em Tcl, Java ou Perl). Um exemplo de tarefa de gerenciamento ligado ao traço apresentado no parágrafo anterior seria: “*sempre que o traço for observado mais do que três vezes dentro de um intervalo de trinta segundos, disparar o reinício do serviço de resolução de nomes*”. As tarefas de gerenciamento são especificadas através de um *wizard* (via interface *web*) e oportunamente convertidas para um *script* Tcl (*Tool Command Language*). Um exemplo de ação seria um script Perl que reinicia o *daemon named*.

Para garantir a escalabilidade da arquitetura, a execução de tarefas de gerenciamento como a supracitada é delegada a gerentes intermediários, via MIB Script . Ao executar uma tarefa de gerenciamento, o gerente intermediário inicialmente (a) configura, via MIB Script, um agente de monitoração informando o traço que ele deve passar a monitorar. O agente armazena as estatísticas sobre os traços em uma MIB SNMP similar à RMON2 (*Remote Network Monitoring Management Version 2*), RFC2021. Após a configuração do agente de monitoração, o gerente intermediário (b) consulta o agente periodicamente para coletar estatísticas a respeito do traço associado. Sempre que o número observado de ocorrências do traço corresponder ao que a tarefa determina (no exemplo acima, três ocorrências em um intervalo de trinta segundos), (c) uma ação pode ser executada. Para tal, o gerente intermediário delega a execução do *script* de ação, via MIB Script, associado à tarefa (voltando ao exemplo, o *script* para reiniciar o *daemon named*) ao agente de ação que reside na estação onde o serviço monitorado se encontra. A arquitetura apresenta ainda mecanismos de notificação (*traps*) para que o gerente intermediário possa sinalizar a ocorrência de eventos significativos à estação de gerenciamento. A Figura 12 ilustra a arquitetura da plataforma trace.

O que se propõe, portanto, é um ambiente de gerenciamento aberto e hierárquico que explora a observação passiva do tráfego para a execução de tarefas de gerenciamento ligadas a protocolos de alto nível, serviços e aplicações em rede. Dentre algumas das vantagens da plataforma trace, destacam-se: 1) tarefas de gerenciamento ligadas a quatro áreas funcionais de gerenciamento (falhas, contabilização, desempenho e segurança) podem ser especificadas e executadas; 2) o ambiente oferece suporte à descentralização de tarefas; 3) o ambiente oferece suporte a especificação de novos cenários de gerenciamento a qualquer momento, usando mecanismos de alto nível; 4) utilização de *software* livre para sua implementação.

O trabalho proposto nesta dissertação mantém a simplicidade do monitoramento de serviços de forma ativa, ou seja, buscando utilizar os serviços efetivamente, através de uma arquitetura centralizada. A arquitetura centralizada utilizada neste trabalho não utiliza qualquer espécie de agente.

No entanto, o trabalho proposto por (Gaspary, 2002) sugere uma interessante extensão do presente trabalho (um trabalho futuro), que seria a distribuição do monitoramento de disponibilidade e tempo de resposta dos serviços através da execução de *scripts* por agentes distribuídos pela rede, através da MIB Script. Uma plataforma que implementa a MIB Script, e que foi utilizada por (Gaspary, 2002), é a JASMIN (Jasmin, 2001).

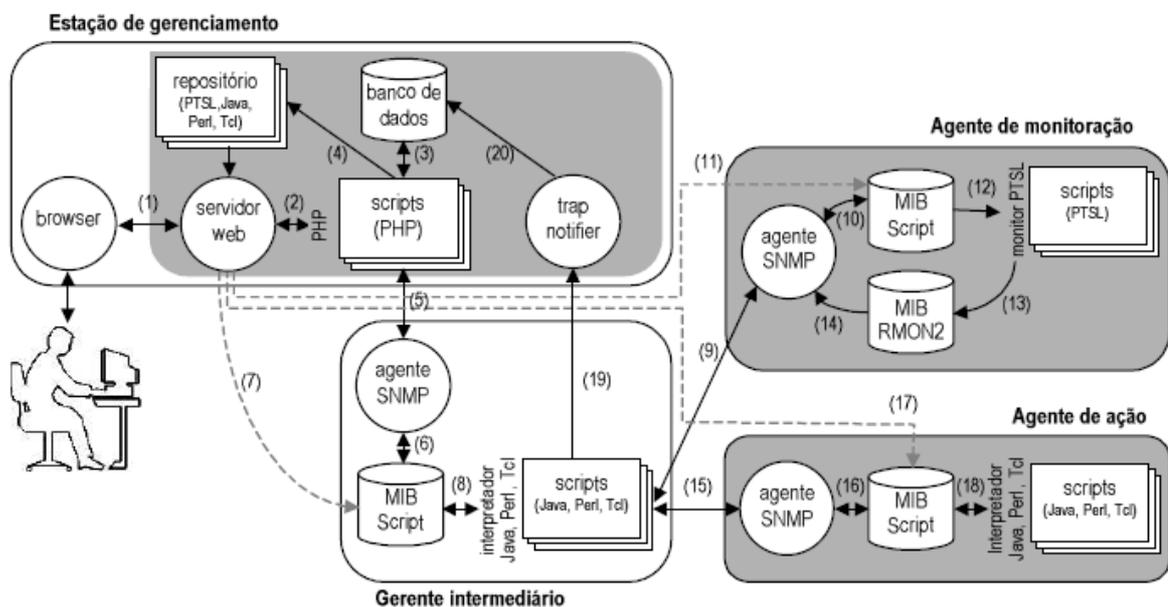


Figura 12 - Componentes da Arquitetura da Plataforma Trace (Gaspary, 2002).

3.4 Software Livre para Gerenciamento de Serviço

Dentre as ferramentas analisadas até então pelo Grupo de Gerência do LPRM, o *Nagios* (Nagios, 1999), que é uma ferramenta livre (*software* livre), se mostra como a ferramenta mais flexível para a gerência de serviços. Isso se deve ao fato do monitoramento ser realizado através de *scripts*, ou seja, o gerente do ambiente de rede e serviços pode monitorar o que quiser desde que tenha um *script* que execute a tarefa desejada e sirva como um *plug-in* para o *Nagios*. Além disso, o *Nagios* também possibilita o gerenciamento de rede através do protocolo SNMP.

Dentre as características do *Nagios*, destacam-se:

- Monitoramento da disponibilidade de um conjunto de serviços de rede, já pré-definidos;
- Monitoramento da disponibilidade de Cluster de Serviços. Um cluster de serviços corresponde a um conjunto de serviços, do mesmo tipo, que respondem por um mesmo serviço, como um conjunto de serviços redundantes de DNS, por exemplo;
- Monitoramento dos resultados retornados (0 ou 1) por um *script* definido por um usuário, a fim de monitorar um serviço ou algo específico;
- Monitoramento de variáveis SNMP;
- Cálculos estatísticos da disponibilidade dos serviços e clusters monitorados;
- Geração de alarmes (via e-mail basicamente).

Algumas características do *Nagios* que deixam a desejar são:

- Não monitoramento do tempo de resposta dos serviços monitorados;
- Não geração de relatórios gráficos e estatísticos para serviços, clusters e grupos de serviços monitorados;
- Dificil usabilidade;
- Necessidade de editar arquivos de configuração para monitorar cada serviço desejado.

O trabalho proposto nesta dissertação busca agregar, em uma nova ferramenta voltada exclusivamente para o monitoramento de serviços, algumas funcionalidades interessantes do *Nagios* (algumas das citadas acima), bem como adicionar novas funcionalidades como o monitoramento do tempo de resposta de serviços e a geração de relatórios gráficos e estatísticos via interface *web*.

O protótipo do sistema apresentado neste trabalho utiliza apenas soluções abertas, ou seja, *software* livre, bibliotecas abertas, plataformas de desenvolvimento e padrões abertos. A opção pelo uso de software livre deve-se ao fato do grupo de gerência do LPRM almejar desenvolver um conjunto de soluções de gerência de baixo custo, acessível às pequenas e médias empresas que também necessitam dos serviços de gerência.

3.5 Concepção da Ferramenta Proposta Baseada nos Trabalhos Relacionados

Os trabalhos apresentados anteriormente foram úteis para agregar idéias, esclarecer conceitos e sugerir novas maneiras de se implementar o trabalho proposto nesta dissertação: *“desenvolver um protótipo de um sistema de gerenciamento de serviços, que monitore a disponibilidade (falha) e o tempo de resposta (desempenho) de serviços de um ambiente de TI.”*.

Há um conjunto de funcionalidades no protótipo apresentado nesta dissertação que não foi encontrado em nenhum outro software livre voltado para o gerenciamento de serviços. Dentre estas funcionalidades, destacam-se principalmente: 1) monitoramento e geração de relatórios do tempo de resposta dos serviços e o 2) o cálculo do Índice de Disponibilidade de Sistema.

As funcionalidades e características do sistema proposto nesta dissertação serão apresentados no capítulo 4.

Capítulo 4

4 A Ferramenta Proposta (SMS – Service Management System)

Este capítulo apresenta o protótipo do sistema proposto nesta dissertação. O sistema será referenciado como SMS (*Service Management System*). Na seção 4.1 são definidas as funcionalidades do sistema (levantamento de requisitos). Na seção 4.2, é apresentada a modelagem do SMS, ou seja, a análise e projeto do sistema. Na seção 4.3, é destacado o diferencial do sistema apresentado.

4.1 Funcionalidades

As funcionalidades do SMS foram levantadas juntamente com administradores de redes que trabalham na área de gerência de TI em organizações como siderurgia, empresas estatais, empresas de turismo, pequenas e médias empresas, entre outros. Segue abaixo a lista dos requisitos que foram definidos para implementação do protótipo:

- Monitorar (via *polling*) a disponibilidade e o tempo de resposta dos serviços de um conjunto de serviços de uma rede IP: WWW (HTTP e HTTPS), e-mail (SMTP), DNS, Banco de Dados (SGBD PostgreSQL), FTP, Telnet e Portas TCP (mini biblioteca de serviços);
- Monitorar a disponibilidade e o tempo de resposta de dispositivos/interfaces de dispositivos usando o comando Ping (ICMP *echo request*) ou o protocolo SNMP (consulta à variável *ifOperStatus.<numero_da_interface>*, através da mensagem *snmpGet*). Embora o SMS não seja um sistema de gerência baseado no SNMP, o monitoramento via SNMP da variável *ifOperStatus*, que é padrão da MIB-II (Stallings, 1999), foi implementado. A idéia é flexibilizar o monitoramento do serviço IP;
- Monitorar a disponibilidade e o tempo de resposta de um Comando (execução de um *Script*). Ou seja, o comando (*script*) especificado será executado e o seu resultado, que deverá ser igual a 1 ou 0, representará sua disponibilidade. O tempo de resposta será igual ao intervalo entre o disparo da execução do comando até o retorno da resposta;

- Monitorar a disponibilidade de um Cluster de Serviços, ou seja, um conjunto de serviços do mesmo tipo que respondem por um mesmo serviço (serviços redundantes). Por exemplo, um cluster de servidores de DNS, que são responsáveis pela resolução de nomes de um determinado domínio;
- Criação de grupos de serviços e cluster de serviços para a análise do Índice de Disponibilidade de Sistema (IDS);
- Geração de relatórios gráficos e estatísticos para serviços e clusters de serviços monitorados;
- Geração do relatório de IDS para o ambiente de TI monitorado;
- Visualização de serviços e clusters de serviços monitorados através do modelo *Red-Green*;
- Definição de SLA para os parâmetros de QoS do SMS:
 - Disponibilidade média de serviço, cluster de serviços, grupo de serviço e do IDS do SMS;
 - Tempo de resposta médio de um serviço.
- Geração de alarmes (e-mail) e execução de comandos na ocorrência de indisponibilidade de um serviço ou cluster de serviço, e na quebra de *threshold* de tempo de resposta de um serviço;
- Armazenamento dos dados de monitoramento em Banco de Dados;
- Consolidação dos dados de monitoramento para análise de tendência;
- Geração de relatórios utilizando dados consolidados;
- Administração do SMS baseada em interface *Web*.

O sistema desenvolvido tenta detectar a ocorrência de problemas de uma forma mais próxima da visão do usuário. Problemas de configuração e desempenho podem não ser detectados quando os serviços são monitorados através de tentativas de abertura de conexão nas portas (“ping na porta”) dos servidores monitorados, pois nesse caso não há processamento no servidor. Uma vez detectada uma falha ou um alto tempo de resposta para um dado serviço, o administrador da rede deve investigar o problema, que pode ter sido causado por: 1) uma outra falha, 2) um problema de configuração ou 3) um problema de desempenho.

A Figura 13 ilustra o diagrama de casos de uso do SMS. Cada caso de uso contempla um conjunto de atividades:

- Gerenciar Coleta: criar coleta, paralisar coleta, excluir coleta, finalizar coleta, reiniciar coleta e editar coleta.
- Gerenciar Cluster: criar cluster de serviços, excluir cluster e editar cluster.
- Gerenciar Grupo: criar grupo de serviços, excluir grupo e editar grupo.
- Gerar Relatórios: gerar relatório de coleta de serviço, cluster de serviços e relatório de IDS.
- Visualizar Modelo *Red-Green*: visualizar o modelo *red-green* dos serviços e cluster de serviços monitorados. Ou seja, visualizar o *status* dos serviços e clusters monitorados.

O Apêndice C apresenta exemplos de interfaces de navegação que representam algumas funcionalidades dos casos de uso mencionados anteriormente. São destacados, principalmente, as funcionalidades de criação de uma coleta, um cluster e um grupo, pois envolve a entrada de um conjunto de informações por parte do administrador da rede.

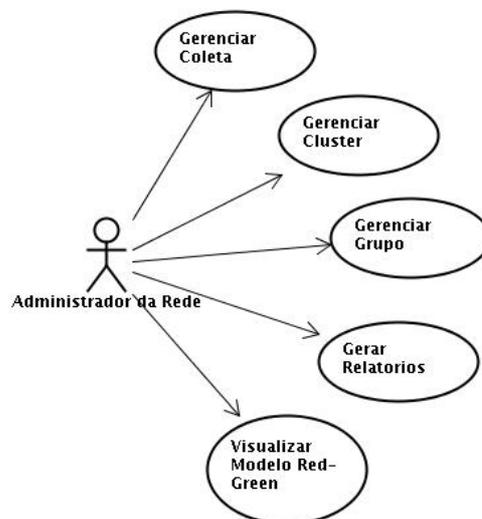


Figura 13 - Diagrama de casos de uso do SMS.

4.1.1 Grupo de Serviços e a Métrica IDS

É importante fundamentar melhor a questão de grupos de serviços, assim como a métrica Índice de Disponibilidade de Sistema (IDS).

Diferentes serviços/clusters de serviços de rede são necessários para suportar o fluxo de trabalho dos funcionários dos departamentos de uma empresa. Serviços e clusters distintos

podem ser agrupados de maneira a formar grupos que se assemelhem em suas funcionalidades. Um grupo de serviços/clusters de “Internet”, por exemplo, pode estar relacionado com os serviços que provêm o acesso à internet ou acesso dedicado a outras empresas parceiras/terceiras. Cabe aos administradores da rede definir o esquema de agrupamento de serviços/clusters de forma a melhor representar o ambiente de TI monitorado.

O IDS consiste em uma métrica quantitativa que busca dar um retorno (*feedback*) do funcionamento do ambiente de TI monitorado, a partir dos grupos existentes neste ambiente. O grupo “Internet”, por exemplo, possui um índice de disponibilidade equivalente a uma média ponderada da disponibilidade dos serviços/clusters que compõem esse grupo. O IDS do ambiente de TI de uma empresa corresponde ao menor índice dentre os grupos existentes.

A implementação desta métrica no SMS foi sugerida por administradores de rede, os quais a implementam, de forma não automatizada, em ambientes corporativos (mais especificamente, em uma companhia siderúrgica³). De acordo com esses administradores de rede, a métrica de IDS está fundamentada no *Balanced ScoreCard* (Bsc, 2005), que consiste em uma metodologia criada por Robert Kaplan e David Norton (Bsc, 2005) para medir o desempenho de empresas, com base em indicadores financeiros e não-financeiros.

4.1.2 As Estatísticas Geradas nos Relatórios

Os relatórios gerados pelo SMS reportam um conjunto de informações e estatísticas a respeito dos serviços ou clusters de serviço monitorados pelo sistema. Tais estatísticas são úteis para uma análise comportamental de um serviço. Algumas dessas informações e estatísticas são:

- Descrição geral do serviço ou cluster monitorado (nome do serviço, IP do servidor, criticidade para disponibilidade e tempo de resposta, alarmes gerados na ocorrência de problemas, outros);
- Tempos de resposta mínimo e máximo de um serviço (em segundos);
- Variância do tempo de resposta de um serviço (em segundos*segundos);
- Desvio padrão do tempo de resposta de um serviço (em segundos);
- Média do tempo de resposta de um serviço (em segundos);

³ Os gestores de TI da Companhia Siderúrgica de Tubarão (CST) utilizam/implementam uma métrica de IDS similar à apresentada neste trabalho.

- Disponibilidade média do serviço ou cluster de serviço (em %);
- Tempo total de monitoramento do serviço ou cluster de serviço (em minutos);
- Tempo do serviço ou cluster no estado *Up* (em minutos);
- Tempo do serviço ou cluster no estado *Down* (em minutos);
- MTTR (*Mean Time To Repair*), MTBF (*Mean Time Between Failures*) e MTTF (*Mean Time To Failure*) (Menascé, 1998) de um serviço ou cluster de serviço (em minutos).

A seguir, tem-se a definição matemática das principais métricas calculadas pelo SMS.

Tempo de Resposta Médio (RT) de um serviço monitorado por uma coleta de serviço.

$$RT = \frac{\sum_{i=1}^n rt_i}{n}$$

Equação 1 - Tempo de Resposta Médio (TR).

Onde rt_i é o tempo de resposta de um *polling* i e n é o número de vezes que o serviço monitorado por *polling* esteve *UP*.

Disponibilidade Média (A) de um serviço monitorado por uma coleta de serviço.

$$A = \frac{T_{UP}}{T_{UP} + T_{DOWN}}$$

Equação 2 - Disponibilidade Média (A).

Onde T_{UP} é o tempo total que o serviço monitorado esteve *UP* e T_{DOWN} é o tempo total que esteve *DOWN*.

Disponibilidade Média de um Cluster de Serviço (A_{CA}) monitorado pelo SMS.

Suponha que um Cluster A (CA) seja formado por N Coletas de Serviços (CS), então $CA = \{CS_1, CS_2, \dots, CS_n\}$. CA está *UP* se CS_1 está *UP*, ou CS_2 está *UP*, ou ..., ou CS_n está *UP*.

$$A_{CA} = \frac{T_{UP_CA}}{T_{UP_CA} + T_{DOWN_CA}}$$

Equação 3 - Disponibilidade Média de um Cluster de serviço (A_{CA}).

Onde T_{UP_CA} é o tempo total que CA esteve *UP* e T_{DOWN_CA} é o tempo total que esteve *DOWN*.

Mean Time To Repair (MTTR), Mean Time To Failure (MTTF) e Mean Time Between Failures (MTBF) são calculados para o monitoramento de Disponibilidade de Serviço e Cluster de Serviço.

$$MTTR = \frac{T_{DOWN}}{n}$$

Equação 4 - MTTR.

Onde T_{DOWN} é o tempo total que o Serviço ou Cluster de Serviço esteve *DOWN*. N é o número de vezes que o Serviço ou Cluster de Serviços esteve *DOWN*.

$$A = \frac{MTTF}{MTTR + MTTF}$$

Equação 5 - MTTF.

Onde A é a disponibilidade média do Serviço ou Cluster de Serviço.

$$MTBF = MTTR + MTTF = \frac{T_{UP} + 2T_{DOWN}}{n}$$

Equação 6 - MTBF.

IDS (Índice de Disponibilidade de Sistema) de um Grupo de Serviço (IDS_{GS}):

Suponha que um Grupo de Serviço (GS) seja composto por n Coletas de Serviço (COS) e m Clusters de Serviço (CLS). Então, $GS = \{COS_1, COS_2, \dots, COS_n, CLS_1, CLS_2, \dots, CLS_m\}$. Cada um (Coletas e Clusters de Serviços) tem seu próprio peso e disponibilidade média dentro de um Grupo de Serviço. O IDS de um Grupo de Serviço (IDS_{GS}) é igual à média ponderada dos valores da disponibilidade média de todos os serviços e clusters de serviços que compõem este grupo.

$$IDS_{GS} = \frac{\sum_{i=1}^n A_{COi} * P_{COi} + \sum_{j=1}^m A_{CLj} * P_{CLj}}{\sum_{i=1}^n P_{COi} + \sum_{j=1}^m P_{CLj}}$$

Equação 7 - Índice de Disponibilidade Média (IDS).

Onde, n é o número de coletas de serviço que compõe o grupo de serviço, m é o número de clusters de serviços que compõem o grupo de serviço, A_{COi} é a disponibilidade média da coleta de serviço i , P_{COi} é o peso para a disponibilidade média da coleta de serviço i , A_{CLj} é a disponibilidade média do cluster de serviço j e P_{CLj} é o peso para a disponibilidade média do cluster de serviço j .

Uma coleta ou um cluster de serviço pode ter diferentes pesos para diferentes grupos de serviços. Por exemplo, um serviço A pode ter um peso para sua disponibilidade média igual a 10 em um grupo de serviço X e igual a 5 em um grupo de serviço Y. O administrador da rede é responsável por determinar da melhor maneira possível os valores desses pesos.

IDS do SMS (IDS_{SMS}):

Existe apenas um valor para o “IDS do SMS”. Este valor é uma função de todos os “IDS de Grupos de Serviços” monitorados pelo sistema. O IDS do SMS corresponde ao IDS do Ambiente de TI monitorado pelo SMS, e é igual ao menor valor de todos os “IDS de Grupos de Serviços”.

$$IDS_{SMS} = MIN\{IDS_{GS1}, IDS_{GS2}, \dots, IDS_{GSn}\}$$

Equação 8 - IDS do SMS.

4.2 Análise e Projeto

Como mencionado na seção 2.1.2.1, a ferramenta proposta no presente trabalho utiliza a arquitetura centralizada, pois um requisito básico para o sistema proposto é sua simplificação do processo de gerência de serviços. Isto é, uma ferramenta que não exija do administrador da rede a necessidade de realizar qualquer tipo de configuração nos servidores a fim de iniciar o monitoramento de algum serviço. A idéia é monitorar um serviço simulando um usuário real do mesmo. A Figura 14 apresenta uma arquitetura conceitual do sistema desenvolvido, a qual é composta por quatro componentes (ou módulos) que são descritos a seguir:

Interface com Usuário: módulo que permite ao usuário controlar e configurar o SMS. Módulo que dá acesso às funcionalidades do sistema de gerenciamento. Na arquitetura, este módulo troca informações apenas com o módulo “Lógica de Negócio do SMS”;

Lógica de Negócio do SMS: módulo que implementa a lógica de negócio do SMS, que é apresentada na seção 4.2.2. Todo o processamento de dados do SMS ocorre neste módulo e por isso pode-se chamá-lo de “Kernel do SMS”. Na arquitetura, este módulo interage com todos os outros módulos: “Interface com Usuário”, “Persistência” e “Modelo de Serviço”;

Persistência: módulo responsável por armazenar fisicamente os dados do SMS em um banco de dados. O banco de dados deve armazenar as configurações do SMS e os dados de monitoramento dos serviços. Na arquitetura, este módulo troca informações apenas com o módulo “Lógica de Negócio do SMS”;

Modelo de Serviço: módulo que implementa o modelo de serviço utilizado pelo SMS, que é apresentado na seção 4.2.1. Este módulo é responsável pela descrição, monitoramento e cálculo dos parâmetros de QoS (Tempo de Resposta e Disponibilidade) dos serviços monitorados. Este módulo possui o sub-módulo **Monitoramento** que é responsável por implementar o monitoramento de disponibilidade e tempo de resposta de cada tipo de serviço (S_1, S_2, \dots, S_n). A seção 4.2.3 detalha como cada tipo de serviço é monitorado pelo SMS. Na arquitetura, este módulo troca informações apenas com o módulo “Lógica de Negócio do SMS”.

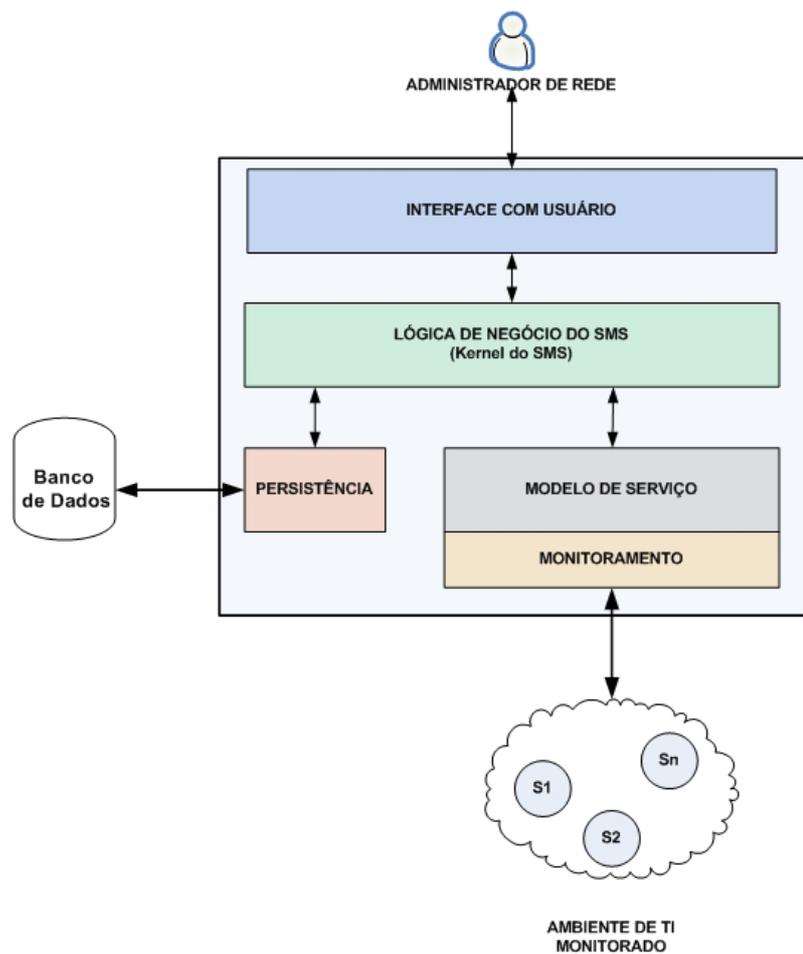


Figura 14 - Arquitetura conceitual do SMS.

A Figura 15 ilustra a implementação da arquitetura do SMS destacando as tecnologias utilizadas.

O Kernel do SMS inclui o **Modelo de Serviço** (discutido na seção 4.2.1) que é responsável pela descrição, monitoramento e cálculo dos parâmetros de QoS (Tempo de Resposta e Disponibilidade) dos serviços monitorados. O Modelo de Serviço possui **Classes de Coleta**, onde cada classe é responsável por monitorar o Tempo de Resposta e a Disponibilidade de um

determinado tipo de serviço através de *pollings*. As **Classes de Coleta** implementam o submódulo **Monitoramento** da arquitetura do SMS. Atualmente o SMS possui um conjunto de classes de coleta para o monitoramento de alguns serviços (definidos na seção 4.1), mas o objetivo é desenvolver classes para os mais diferentes tipos de serviços (vide trabalhos futuros na seção 6.2). O monitoramento dos serviços é executado usando código Java e *scripts* Perl. Os detalhes de como um determinado serviço é monitorado são destacados na seção 4.2.3.

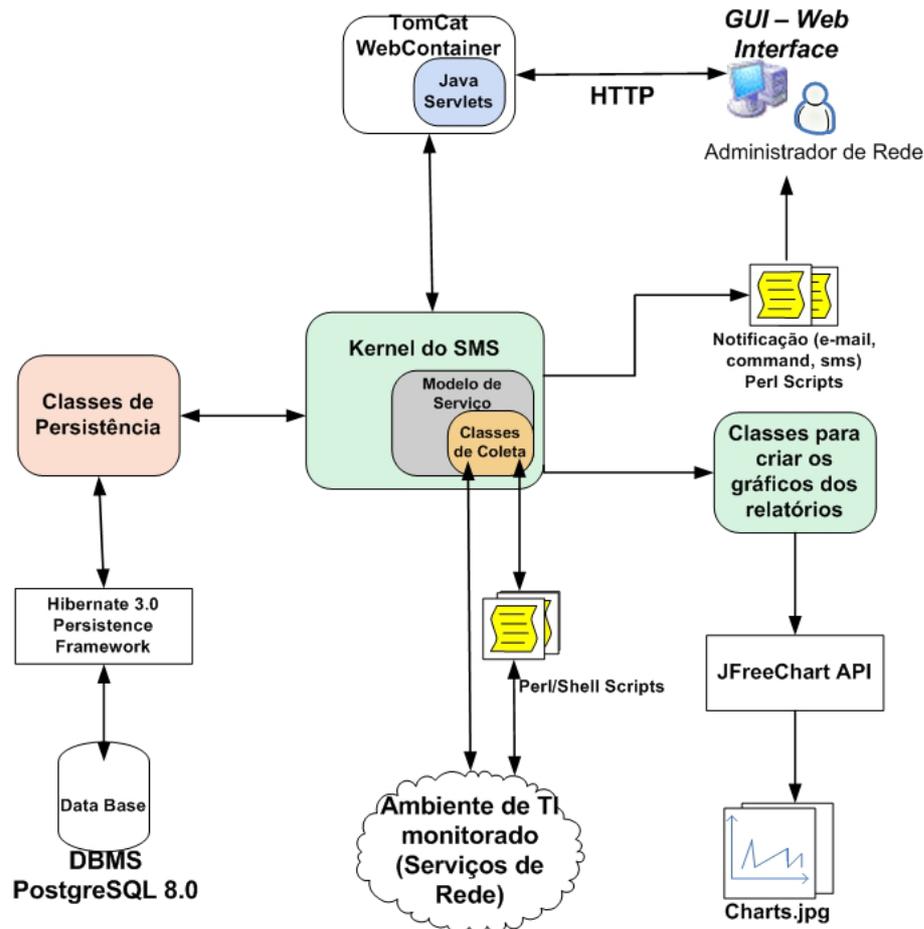


Figura 15 – Implementação da arquitetura do SMS.

O SMS possui **classes de persistência** que usam o *framework* Hibernate 3.0 para armazenar os dados do sistema em um banco de dados (SGBD PostgreSQL). **Classes para criar os gráficos dos relatórios** estatísticos gerados pelo SMS também são utilizadas. A API JfreeChart é usada para geração desses gráficos. A **interface Web** do SMS é implementada através da tecnologia Java Servlets.

Scripts Perl são utilizados para implementar a **notificação de eventos**, os quais devem ser enviados para o administrador da rede quando um evento de disponibilidade ou tempo de resposta ocorre no ambiente de TI monitorado.

O SMS foi desenvolvido para funcionar sob a plataforma Linux e o Suse 9.3 foi utilizado no desenvolvimento. A linguagem Java (J2SE 1.5) foi utilizada como padrão para o desenvolvimento do SMS. A plataforma também utiliza a linguagem Perl, pois muitos *scripts* Perl de monitoramento são utilizados pelo SMS a fim de flexibilizar o monitoramento de certos tipos de serviços. O *WebContainer Jakarta Tomcat* é utilizado como servidor *Web* para os *Servlets* que implementam a interface *Web* do SMS.

A Figura 16 ilustra o diagrama de pacotes do SMS. Os pacotes “GuiServlet” e “SMS” foram implementados para o desenvolvimento do SMS e o diagrama de classes desses pacotes encontra-se no Apêndice B desta dissertação. Os pacotes “JFree” (JFreeChart) e “Hibernate” são livres e encontram-se disponíveis na internet para serem utilizados. O pacote “SMS” implementa o **Kernel do SMS**, as **classes de persistência** e as **classes para criação dos gráficos dos relatórios**. A partir do diagrama de pacotes, vê-se que o pacote “SMS” utiliza os pacotes “JFree” e “Hibernate”.

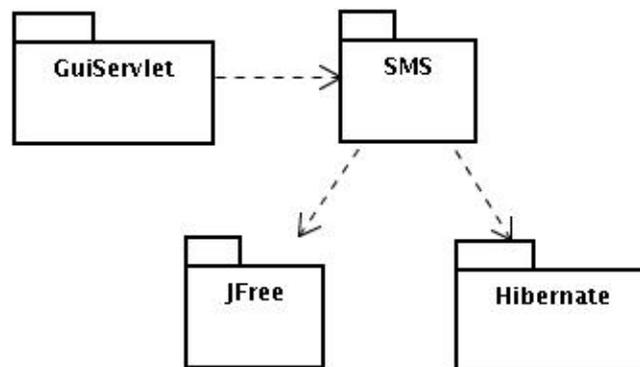


Figura 16 - Diagrama de Pacotes do SMS.

O pacote “GuiServlet” implementa os servlets da **interface Web** do SMS. A partir do diagrama de pacotes, vê-se que o pacote “GuiServlet” utiliza o pacote “SMS”. O Apêndice C ilustra algumas páginas *Web* do SMS, as quais são implementadas pelos servlets do pacote “GuiServlet”.

4.2.1 Modelagem de um Serviço

Como mencionado anteriormente, o Kernel do SMS possui um modelo de serviço que visa descrever um serviço e os parâmetros de QoS considerados no sistema (a disponibilidade e o tempo de resposta). O diagrama de classes UML do **modelo de serviço** ilustrado na Figura 17 foi desenvolvido neste trabalho a partir de idéias extraídas do modelo proposto por (Rodosek, 2002) (seção 3.2). O diagrama da Figura 17 representa a modelagem utilizada no SMS. É

importante ressaltar que, além de (Rodosek, 2002), não foram encontrados outros trabalhos que apresentassem uma modelagem de serviço, a fim de se avaliar diferentes formas para representação de serviços. Por isso, não foram encontradas outras alternativas a não ser desenvolver um modelo próprio para o SMS.

Modelo de Serviço

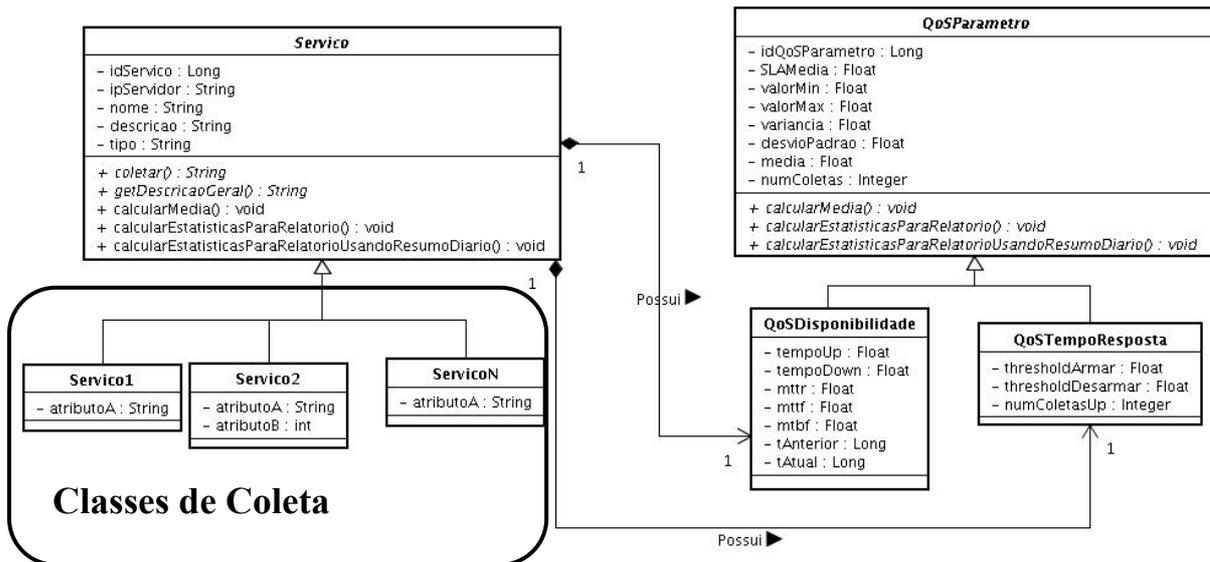


Figura 17 - Modelo geral de serviço do SMS.

A modelagem de serviço apresentada na Figura 17 propõe um modelo genérico que atende às especificações do SMS, ou seja, um modelo que descreve um serviço, que trata das particularidades de monitoramento de cada tipo de serviço e que considera os parâmetros de QoS especificados. Embora esta modelagem atenda bem aos requisitos do SMS, não se pode afirmar que tal modelagem é amplamente generalizada, por se tratar de uma modelagem voltada para uma aplicação de gerenciamento limitada. Em diferentes contextos esta modelagem pode não ser suficiente para atender outros tipos de requisitos como, por exemplo, parâmetros de localização.

A classe abstrata “Serviço” possui como atributos o IP do Servidor do serviço (ipServidor), o nome do serviço (servico), a descrição do serviço (descricao) e o tipo do serviço (tipo). O tipo do serviço é definido pelas **classes de coleta** que herdam a classe “Serviço” e corresponde, literalmente, ao tipo do serviço (www, dns, email ou outros).

O método abstrato “coletar()” da classe abstrata “Serviço” é peça importante neste modelo, pois as classes de coleta devem implementar este método. Através deste método, cada classe de coleta implementa o monitoramento da disponibilidade e do tempo de resposta de um

serviço de forma específica. Por exemplo, a classe de coleta “Serviço1” implementa o método “coletar()” de forma a monitorar um serviço do tipo “servico1”, que com certeza é diferente do monitoramento do serviço do tipo “servicoN”, implementado pela classe de coleta “ServicoN”. A seção 4.2.3 e o Apêndice A apresentam mais detalhadamente como cada tipo de serviço é monitorado pelo SMS.

A classe abstrata “Serviço” possui ainda dois atributos da classe “QoSParametro”, mais especificamente, um atributo da classe “QoSTempoResposta” e outro da classe “QoSDisponibilidade”, onde ambas as classes herdam a classe abstrata “QoSParametro”. A classe “QoSParametro” possui um conjunto de atributos e os métodos abstratos “calcularMedia()”, “calcularEstatisticasParaRelatorio()” e “calcularEstatisticasParaRelatorioUsandoResumoDiario()” que são utilizados para computar os dados de monitoramento que são utilizados na geração de relatórios.

A Figura 18 apresenta um diagrama de classes mais simplificado do Kernel do SMS. No Apêndice B este diagrama pode ser visualizado com mais detalhes. A partir do diagrama de classes do Kernel do SMS pode-se observar que a classe “Coleta” possui um atributo da classe “Serviço”. É a partir dessa associação que o **modelo de serviço** é integrado ao Kernel do SMS, onde uma instância da classe “Coleta” é responsável pelo monitoramento de uma instância da classe “Serviço”, ou melhor, uma instância de uma **classe de coleta**. Nesse diagrama é possível observar também as **classes de coleta** que foram implementadas no SMS. São elas: “www”, “dns”, “email”, “ftp”, “telnet”, “ip”, “postgresql”, “portatcp” e “comando”.

A classe “SMS” é a classe principal do sistema e quando este está em execução existe apenas uma única instância desta classe. Essa instância agrupa todas as coletas, clusters e grupos de serviços monitorados pelo sistema. A seção 4.2.2 apresenta a lógica de funcionamento do Kernel do SMS, ou seja, a lógica do sistema de gerência implementado.

4.2.2 Lógica do Sistema

A lógica do Kernel do SMS é apresentada na Figura 19. Esta lógica consiste nos seguintes fatos:

- 1) Uma **Coleta de Serviço** monitora o tempo de resposta e a disponibilidade de um único serviço. Neste caso tem-se uma associação de uma coleta para um serviço;
- 2) Um **Cluster de Serviço** monitora a disponibilidade de um ou mais serviços do mesmo tipo. Neste caso tem-se uma associação de um cluster para uma ou mais coletas de serviços;
- 3) Um **Grupo de Serviço** é formado por uma ou mais coletas de serviços e/ou um ou mais cluster de serviços. Neste caso tem-se uma associação de um grupo para uma ou mais coletas de serviços e/ou um ou mais cluster de serviços. Um **Grupo de Serviço** calcula o IDS do grupo com base na disponibilidade média de cada serviço e/ou cluster de serviço;
- 4) O IDS do ambiente como um todo monitorado pelo SMS (**IDS do SMS**), é calculado em função do IDS dos vários grupos de serviço que existem no SMS. Neste caso tem-se uma associação de um IDS para um ou mais grupos de serviços.

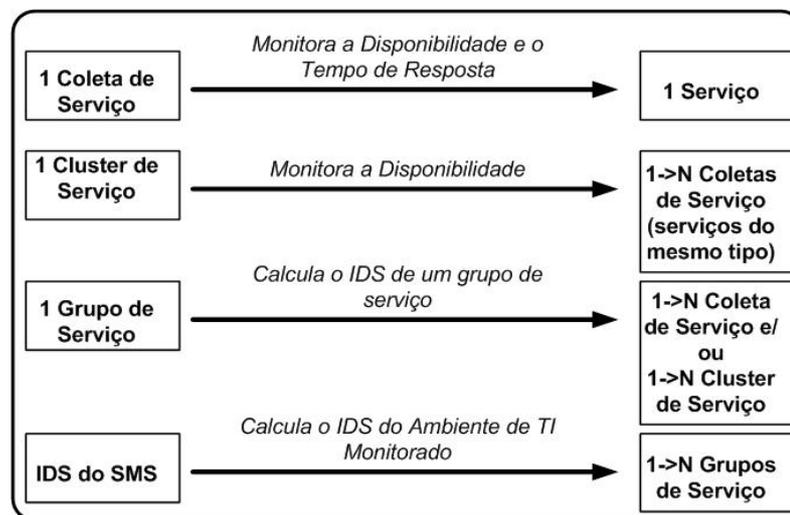


Figura 19 - Lógica do SMS.

4.2.3 Monitoramento dos Serviços de Rede

Como já foi mencionado, o SMS busca monitorar os serviços de rede simulando usuários reais, na medida do possível. Para cada tipo de serviço o sistema implementa uma classe para o monitoramento do mesmo. É interessante descrever como um serviço é monitorado em

essência, já que nenhum tipo de MIB é utilizada na obtenção das informações (exceto o monitoramento do serviço IP que usa o SNMP). Esta seção descreve como cada serviço implementado pelo SMS é monitorado, mas apenas o fluxograma e o código relativo ao monitoramento do serviço de E-mail (SMTP) são apresentados nesta seção. O Apêndice A apresenta os fluxogramas e os códigos de monitoramento dos outros serviços.

WWW (HTTP ou HTTPS)

Forma de monitoramento: realizar um “*http get*” de uma página *Web* do servidor WWW (HTTP ou HTTPS) e verificar o código HTTP de retorno. Se for “200”, serviço “OK”, caso contrário, “FALHA”.

DNS

Forma de monitoramento: realizar uma consulta de DNS, ou seja, solicitar ao servidor que resolva um determinado nome. Se o nome for resolvido, “OK”, caso contrário, “FALHA”.

FTP

Forma de monitoramento: abrir uma conexão FTP com o servidor de FTP, logar no servidor, executar o comando NLST (RFC 959) e fechar a conexão. Se tudo for executado, “OK”, caso contrário, “FALHA”.

TELNET

Forma de monitoramento: abrir uma conexão via Telnet (Terminal Remoto) com o servidor de Telnet, logar no servidor e fechar a conexão. Se tudo for executado, “OK”, caso contrário, “FALHA”.

BANCO DE DADOS

Forma de monitoramento: abrir uma conexão JDBC com um BD de um determinado SGBD, criar um *statement* JDBC para a conexão, fechar o *statement* e a conexão. Se tudo for executado, “OK”, caso contrário, “FALHA”. É importante dizer que essa funcionalidade varia de SGBD para SGBD. Atualmente o SMS tem uma implementação para o SGBD PostgreSQL.

PORTA TCP

Forma de monitoramento: solicitar uma conexão TCP via *socket* para uma determinada porta de um determinado servidor de aplicação. Se a conexão for aceita, “OK”, caso contrário, “FALHA”.

IP

Forma de monitoramento: monitoramento de links utilizando os protocolos ICMP (ping) ou SNMP (ifOperStatus.<numeroInterface>).

COMANDO

Forma de monitoramento: executar o comando (*script*) especificado. Se o resultado retornado for igual a 1, “OK”, se for igual a 0, “FALHA”.

E-MAIL (SMTP)

Forma de monitoramento: abrir uma conexão SMTP com o servidor de e-mail, executar o comando HELO (RFC 821) e fechar a conexão. Se tudo for executado, “OK”, caso contrário, “FALHA”.

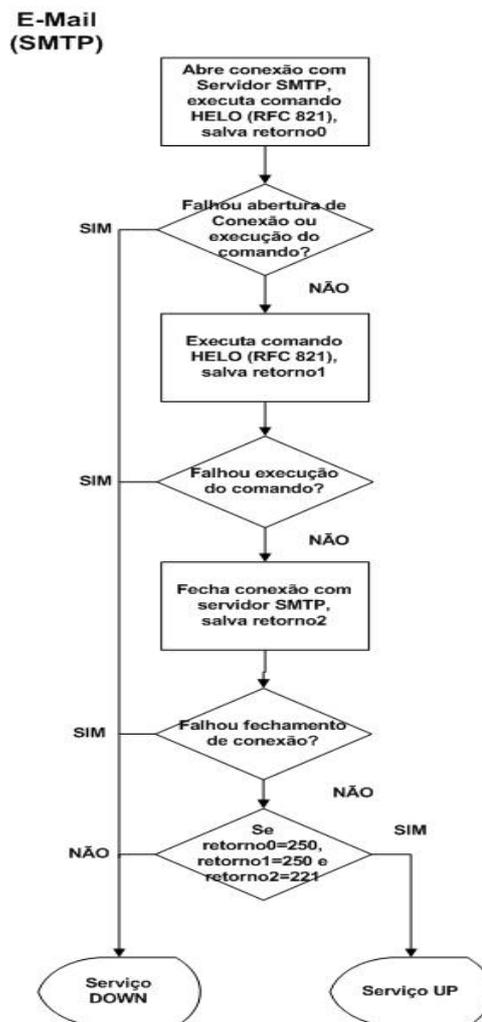


Figura 20 - Fluxograma do monitoramento do serviço de E-mail (SMTP).

A fluxograma da Figura 20 retrata a forma de monitoramento do serviço de E-mail (SMTP) realizada pelo SMS. Seguem abaixo dois trechos de código. O primeiro refere-se ao método “coletar()” da classe de coleta “email” e o segundo refere-se ao *script* Perl, executado pelo método “coletar()”, que monitora efetivamente o serviço SMTP de um determinado servidor.

Método Coletar() da Classe de Coleta “Email”

```

1   public String coletar(ResultadoAtual resAtual) {
2       String comando = null;
3       int disponibilidade = 0;
4       long t1 = 0, t2 = 0;
5       float tempoResposta = 0;
6       String mensagem = null;
7
8       try {
9           // Executa o comando
10          comando = "/home/bonfa/mestrado/SMS/PerlScripts/e-mail.pl " +
getIpServidor();
11          t1 = System.currentTimeMillis();
12          Process child = Runtime.getRuntime().exec(comando);
13          t2 = System.currentTimeMillis();
14
15          // Le a saida do comando executado.
16          InputStream in = child.getInputStream();
17          int c;
18          if ((c = in.read()) != -1) {
19              if (c == '1'){
20                  disponibilidade = 1;
21                  tempoResposta = (float)(t2 - t1)/1000;
22              }else {
23                  disponibilidade = 0;
24                  tempoResposta = 0f;
25              }
26          }
27          in.close();
28      } catch (IOException e) {
29          disponibilidade = 0;
30          tempoResposta = 0f;
31          mensagem = "Falha na execucao do comando: " + comando;
32      }
33
34      resAtual.setDisponibilidade(disponibilidade);
35      resAtual.setTempoResposta(tempoResposta);
36      resAtual.setTimeAtual(System.currentTimeMillis());
37      return mensagem;
38  }

```

Script Perl para Monitoramento do Serviço de E-mail (SMTP) – e-mail.pl

```

1   #!/usr/bin/perl5.8.6
2
3   use Net::SMTP;
4
5   my $emailServer = $ARGV[0];
6
7   # Cria um canal com o Servidor de E-mail e executa o comando "Helo"
uma vez.

```

```

8     $smtp = Net::SMTP->new($emailServer, Hello => "SMS", Timeout => 120)
or (print "0\n" and exit);
9     my $codigoRetornado0 = $smtp->code;
10
11     # Executa o comando "Helo" (RFC 821) pela segunda vez.
12     $smtp->hello('SMS') or (print "0\n" and exit);
13     my $codigoRetornado1 = $smtp->code;
14
15     # Fecha o canal de comunicacao com o servidor de E-mail.
16     $smtp->quit or (print "0\n" and exit);
17     my $codigoRetornado2 = $smtp->code;
18
19     # Se tudo OK, print 1.
20     if ($codigoRetornado0 == 250 && $codigoRetornado1 == 250 &&
    $codigoRetornado2 == 221){
21         print "1\n";
22     } else {
23         print "0\n";
24     }

```

4.3 O Diferencial da Ferramenta Proposta

Dentre as funcionalidades disponibilizadas no SMS, destacam-se como diferenciais as seguintes: 1) permitir ao administrador da rede escalonar o monitoramento de um determinado serviço de rede através de uma interface *Web*, sem a necessidade de editar arquivos de *scripts* para tanto, já que existe uma mini biblioteca de serviços embutida no SMS; 2) visualizar relatórios, via interface *Web*, de disponibilidade e tempo de resposta de serviços monitorados; 3) Gerar relatórios de IDS (“Índice de Disponibilidade de Sistema”) para visualizar em nível mais alto o desempenho do funcionamento do ambiente de TI monitorado. Através do relatório de IDS, é possível identificar qual serviço está mais impactando no desempenho do ambiente de TI que suporta o negócio de uma empresa. 4) permitir que o administrador da rede monitore outros serviços a partir de *scripts* criados pelo próprio administrador. Isto contribui muito para a flexibilidade da ferramenta.

Capítulo 5

5 Estudo de Caso

Um estudo de caso foi realizado para demonstrar a utilidade do sistema apresentado nesta dissertação. O objetivo do estudo de caso é monitorar os Serviços que atendem aos usuários do LPRM-DI-UFES (Laboratório de Pesquisas em Redes e Multimídia do Departamento de Informática da UFES).

A Figura 21 ilustra a página principal do SMS que dá acesso à configuração do monitoramento dos serviços.

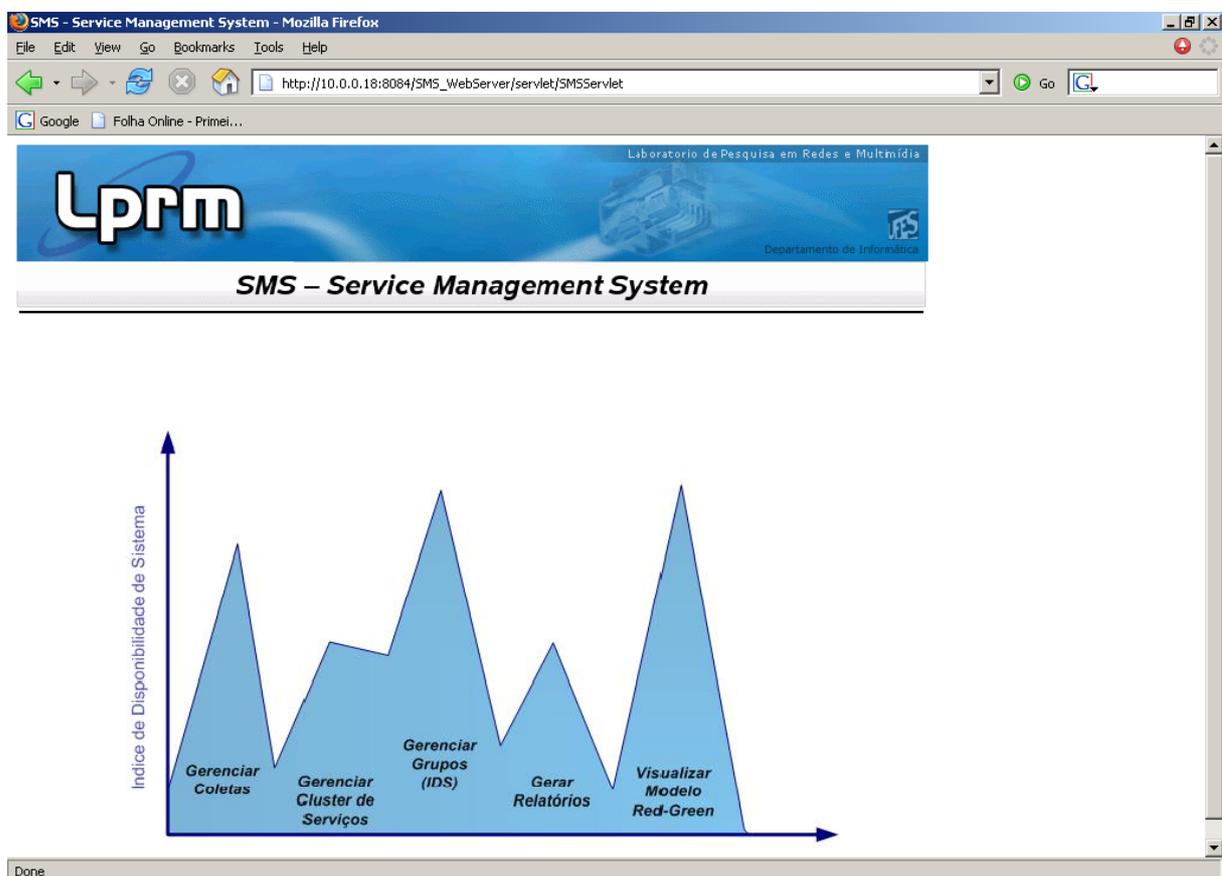


Figura 21 - Página principal do SMS.

A partir da página principal, pode-se visualizar os principais grupos de funcionalidades do SMS, os quais foram especificados na seção 4.1:

- Gerenciar Coletas: para criar, excluir, paralisar e reiniciar uma coleta de um serviço, ou seja, o monitoramento da disponibilidade e do tempo de resposta de um serviço.

- Gerenciar Cluster de Serviços: para criar e excluir o monitoramento da disponibilidade de um cluster de serviço.
- Gerenciar Grupos (IDS): para criar e excluir um grupo de serviço, que consiste em um conjunto de serviços e/ou clusters de serviços.
- Gerar Relatórios: para gerar um relatório via interface *Web* de uma coleta de serviço, de um cluster de serviço ou do IDS do ambiente monitorado.
- Visualizar Modelo *Red-Green*: para visualizar o *status* dos serviços ou clusters de serviços monitorados.

Os serviços monitorados neste estudo de caso são (Figura 22):

- WWW (SLA = 99,9%): serviço de impressão, WebMail do DI-UFES e Home Page do LPRM;
- DNS (SLA = 99,9%): serviço de DNS do servidor do LPRM e do NPD-UFES;
- E-mail (SLA = 99,9%): serviço SMTP do DI-UFES;
- Banco de Dados (SLA = 99,9%): banco de dados do SMS;
- IP (SLA = 99,9%): link que permite acesso à internet (monitorado via ICMP);
- Porta TCP (SLA = 99,9%): serviços SSH, NIS, NFS e Proxy do Servidor do LPRM;

Todos os serviços tiveram um SLA para a média do tempo de resposta igual a 5 seg.

Um cluster de serviço de DNS também é monitorado:

- O cluster é formado pelos serviços de DNS do servidor do LPRM e do servidor do NPD-UFES.

Três grupos de serviços foram criados:

- Grupo de Serviços de Internet (SLA = 99,9%): são os serviços mais importantes que permitem que os usuários do LPRM utilizem a internet (Cluster de DNS: peso=10; Proxy: peso=5; Link Internet: peso=10).
- Grupo de Serviços Críticos (SLA = 99,9%): são os serviços sem os quais os usuários não conseguem usar o laboratório (NIS: peso=10; NFS: peso=10).

- Grupo de Serviços Básicos (SLA = 99,9%): são os serviços não críticos mas que os usuários estão utilizando diariamente (Webmail: peso=10; e-mail: peso=10; impressão: peso=6; SSH: peso=4).

Cabe ao administrador da rede definir os pesos de cada serviço ou cluster dentro de um grupo de acordo com a relevância/importância do serviço no grupo em questão.

O IDS do SMS (do ambiente monitorado), tem um SLA para disponibilidade igual a 99,9%.

Todos os serviços, cluster de serviços, grupos de serviços e o IDS do SMS tiveram sua SLA para disponibilidade média definida para 99,9%. Essa definição deve ser acordada nas organizações que possuem seus ambientes de TI monitorados.

Em (Villaça, 2004) é apresentada uma pesquisa que esboça a classificação de sistemas de acordo com a disponibilidade (Tabela 1). Através dessa tabela, é possível concluir que um sistema com disponibilidade igual a 99,9% pode ser considerado como um sistema bem gerenciado.

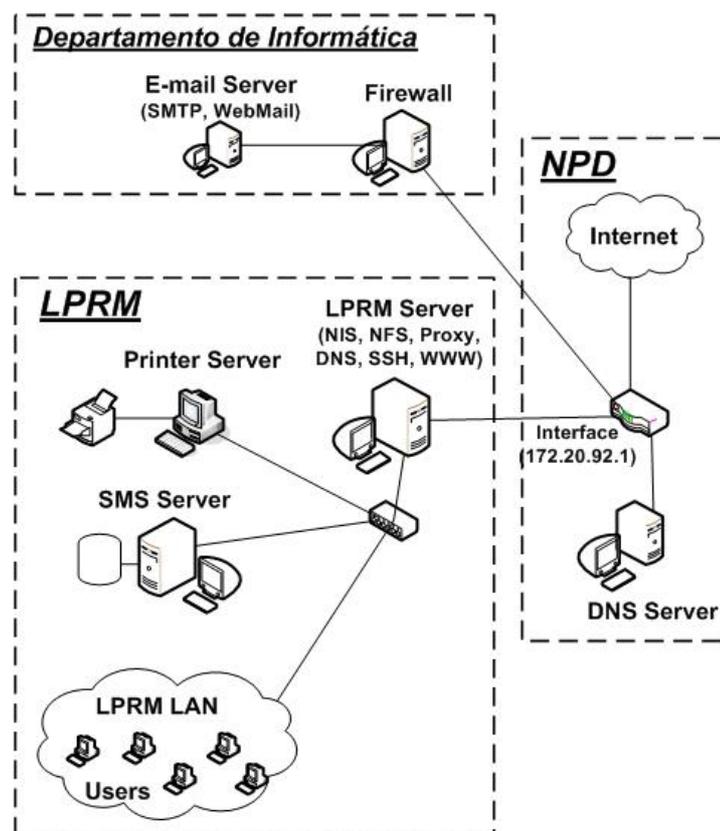


Figura 22 - Estudo de caso.

Tabela 1- Classificação de sistemas de acordo com a disponibilidade.

Classe de Disponibilidade	Disponibilidade	Indisponibilidade (min/ano)	Tipo de Sistema
---------------------------	-----------------	-----------------------------	-----------------

1	90,0%	52.560	Não-gerenciado
2	99,0%	5.256	Gerenciado
3	99,9%	526	Bem-gerenciado
4	99,99%	52,6	Tolerante-à-Falhas
5	99,999%	5,3	Alta Disponibilidade
6	99,9999%	0,53	Muito Alta Disponibilidade
7	99,99999%	0,053	Ultra Disponibilidade

Após um curto período de monitoramento de 7 dias pôde-se gerar o relatório de IDS da Figura 23. Neste relatório, pode-se observar que o SLA do IDS do ambiente monitorado não foi satisfeito, pois o IDS tem um valor igual a 99,47% (<99,9%). Pode-se ainda identificar que o problema encontra-se no grupo de serviços básicos do LPRM, e mais exatamente no serviço de impressão, que teve uma disponibilidade média igual a 97,51%. Gerando-se um relatório (Figura 24) do monitoramento deste serviço, pode-se observar que ele teve um grande tempo de parada (135min). Após investigações, constatou-se que no dia 07/10/2005 houve uma curta queda de energia no LPRM. Nesta queda, até mesmo o Sistema de Gerência (SMS) foi desligado. O servidor de impressão não voltou a funcionar normalmente quando a energia foi re-estabelecida, tendo que sofrer manutenção. O MTTR para o serviço de impressão foi de 45min, pois no período do incidente o serviço ficou DOWN e UP três vezes.

Foi observado também neste período de monitoramento que todos os serviços tiveram um baixo tempo de resposta.

Através deste estudo de caso, foi possível observar como o SMS pode auxiliar no gerenciamento de Serviços de um Ambiente de TI. Através da ferramenta, foi possível obter uma medida do funcionamento do ambiente de TI do LPRM e determinar quais serviços influenciaram mais no não cumprimento do SLA estabelecido para o IDS do LPRM.

O Apêndice C ilustra um conjunto de interfaces de navegação do SMS a fim de demonstrar a usabilidade do sistema desenvolvido. O apêndice apresenta também um relatório (Figura 46) de um cluster de serviços de DNS (o mesmo do estudo de caso apresentado neste capítulo) que utiliza dados de monitoramento de mais de um mês de coleta (07/02/2006 a 01/03/2006). Os dados utilizados para geração de tal relatório são dados consolidados, ou seja, resumos diários.

SMS - Service Management System - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://10.0.0.18:8084/SMS_WebServer/servlet/SMSGerarRelatorioIDSServlet

Relatório de IDS (Índice de Disponibilidade de Sistema)

IDS (%)	Grupo	Índice do Grupo (%)	Serviço/Cluster	Índice do Serviço/Cluster (%)	Peso	Índice x Peso (%)
99.47318 (SLA = 99.9 %)	Grupo de Serviços de Internet (SLA = 0.999 %)	100.0	Link internet do LPRM (SLA = 99.9 %)	100.0	10.0	1000.0
			Proxy do LPRM (SLA = 99.9 %)	100.0	5.0	500.0
			Cluster de DNS (SLA = 99.9 %)	100.0	10.0	1000.0
	Grupo de Serviços Críticos do LPRM (SLA = 0.999 %)	100.0	NFS (SLA = 99.9 %)	100.0	10.0	1000.0
			NIS (SLA = 99.9 %)	100.0	10.0	1000.0
	Grupo de Serviços Básicos do LPRM (SLA = 0.999 %)	99.47318	Webmail (SLA = 99.9 %)	99.90838	10.0	999.0838
			E-mail do DI-UFES (SLA = 99.9 %)	100.0	10.0	1000.0
			CUPS (SLA = 99.9 %)	97.51863	6.0	585.11176
			SSH	100.0	4.0	400.0

Done

Figura 23 - Relatório de IDS.

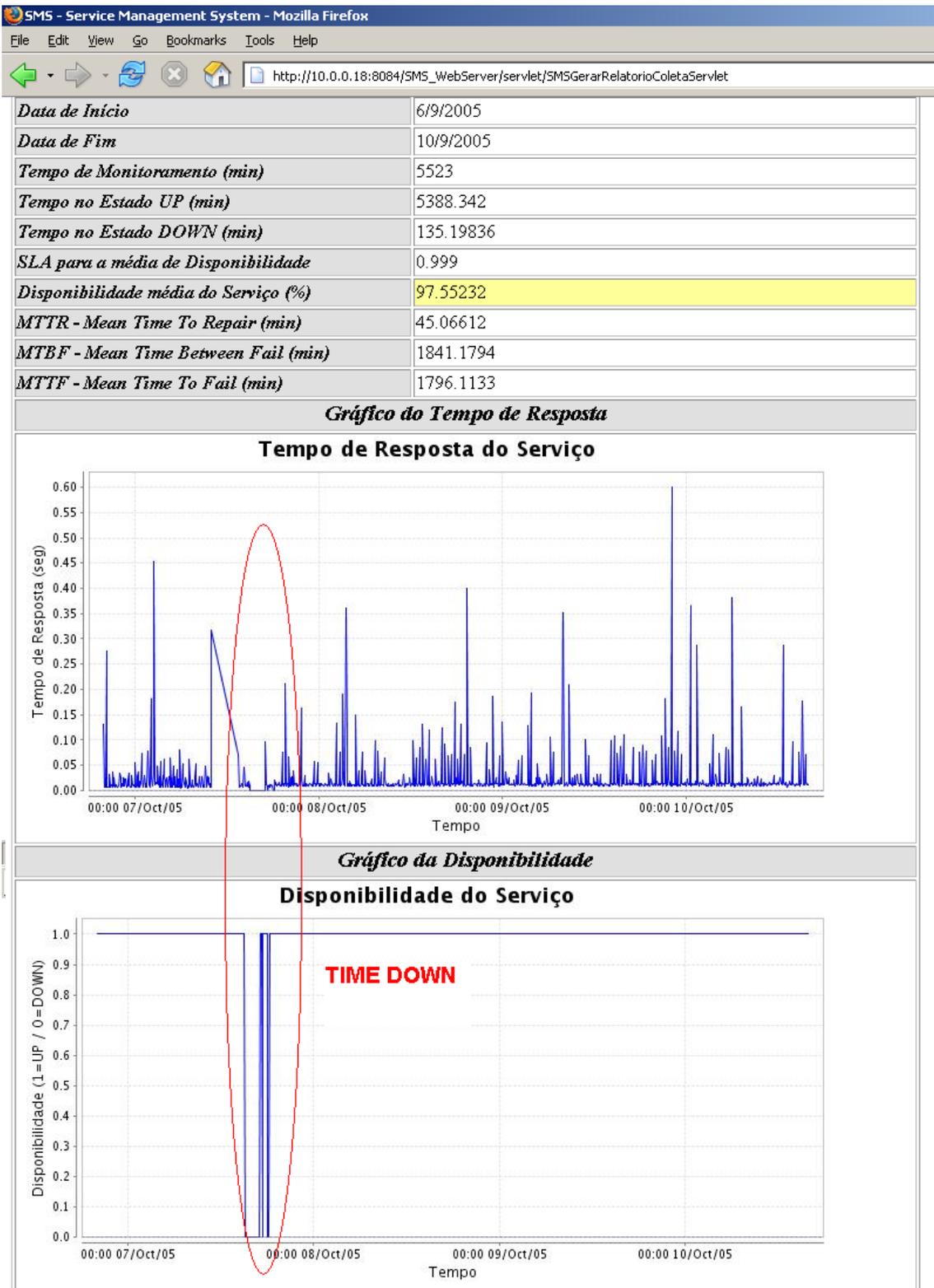


Figura 24 - Relatório do Serviço de Impressão.

Capítulo 6

6 Conclusão

Esse trabalho discutiu a importância da gerência de redes e serviços em ambientes de TI das organizações, dando um maior enfoque à gerência de serviços, uma vez que já existem padrões, *frameworks* e vários sistemas de gerenciamento de redes consolidados. Dentre os trabalhos e as ferramentas de gerência estudadas, há poucas soluções abertas que permitem a gerência de serviços. Vale destacar que muitas das soluções existentes para tal tarefa são proprietárias e com um alto custo.

Baseando-se nessas premissas, este trabalho apresentou uma proposta de um sistema para gerência de serviços de rede baseado em interface *Web* desenvolvido pelo Grupo de Gerência do LPRM, que utiliza ferramental aberto, capaz de monitorar de forma centralizada a disponibilidade e o tempo de resposta de um conjunto serviços de uma rede IP.

O objetivo do sistema, nomeado de SMS, é auxiliar a atividade de gerenciamento de redes permitindo o Gerenciamento de Nível de Serviço em ambientes de TI. Diferentes métricas são calculadas e apresentadas em forma de relatórios, acessados via *Web*. Um estudo de caso foi desenvolvido para ilustrar a utilidade da ferramenta proposta.

Este trabalho buscou evoluir funcionalidades e integrar-se, como mais uma proposta de solução de gerência, aos trabalhos já desenvolvidos pelo Grupo de Gerência do LPRM (destacados na seção 1.3), os quais visam constituir um conjunto de soluções abertas voltadas para a gerência de redes e serviços.

Para o desenvolvimento do presente trabalho optou-se por implementar uma ferramenta que utilizasse uma arquitetura centralizada, pois um requisito básico era gerenciar serviços da forma mais simples possível. O contexto de simplicidade utilizado neste trabalho significa a não necessidade de ter que configurar algum tipo de agente para iniciar o monitoramento de um determinado serviço. Foram encontrados alguns trabalhos, como (Gaspary, 2002), que utilizavam arquiteturas distribuídas para a gerência de serviços, mas tais soluções, apesar de se mostrarem eficazes, não eram acompanhadas de simplicidade no processo de gerenciamento.

Devido a isso, este trabalho se propôs a criar uma ferramenta que gerencia de forma centralizada a disponibilidade e o tempo de resposta dos serviços de uma rede através de testes, ou seja, um software que testa um serviço simulando a sua utilização como se fosse um usuário ou uma aplicação real que estivesse utilizando o mesmo. Por exemplo, ao invés de verificar se a porta UDP 53 de um servidor de DNS está ativa, a ferramenta desenvolvida solicita que o servidor de DNS resolva um nome de rede. Com isso, é possível verificar se o serviço está funcionando adequadamente e se o tempo de resposta do serviço apresenta valores aceitáveis.

Com uma solução de gerência desse tipo não é necessário que configurações específicas sejam realizadas no servidor a fim de viabilizar o monitoramento de um determinado serviço. Outra vantagem é que todos os dados de monitoramento e configuração encontram-se centralizados na estação de gerência para que diferentes estatísticas sejam geradas. A ferramenta desenvolvida explora tais vantagens gerando diferentes relatórios, acessados via *Web*, para um conjunto de métricas, as quais foram definidas na seção 4.1.2.

No entanto, alguns problemas e dificuldades foram encontrados para o desenvolvimento da solução proposta. Há serviços em que a implementação de seu teste de disponibilidade tem um grau de complexidade bastante elevado. Por exemplo, como testar um serviço de DHCP sendo que este tem o seu funcionamento baseado em *broadcast* no nível de enlace de dados? Ou seja, o serviço não funciona sob uma porta TCP ou UDP. Logo, como seria simular uma aplicação real que utiliza o serviço de DHCP? Uma possibilidade é a implementação de um *socket* que realiza tal tarefa. Outro exemplo é o monitoramento do serviço NIS, que apesar de responder em uma porta TCP esta porta não é fixa, ou seja, o NIS funciona sob uma porta TCP aleatória sempre menor que 1024.

Tais particularidades dificultam a implementação do monitoramento de certos serviços seguindo a idéia adotada na solução proposta. Isso não quer dizer que seja impossível de se implementar, mas talvez utilizando uma arquitetura de gerenciamento distribuída baseada em agentes tais problemas poderiam ser solucionados mais facilmente, pois um agente poderia ficar alojado no servidor verificando constantemente se o processo do serviço está funcionando adequadamente ou não. Devido a isso, sugere-se, na seção 6.2, que um dos trabalhos futuros seja uma implementação e avaliação de uma solução de gerenciamento de serviços baseado em uma arquitetura distribuída, ou melhor, integrar gerenciamento distribuído no sistema desenvolvido nesta dissertação.

6.1 Resultados da Dissertação

Tomando-se por base as atividades de pesquisa realizadas durante os últimos dois anos de trabalho, pode-se considerar como os principais resultados desta dissertação:

- Implementação de um **protótipo de um sistema de gerência de serviços** baseado em *Web*, denominado **SMS (*Service Management System*)**, capaz de monitorar de forma centralizada e com a perspectiva do usuário a disponibilidade (falha) e o tempo de resposta (desempenho) de um conjunto serviços. O objetivo da ferramenta é auxiliar a atividade de gerenciamento de redes permitindo o Gerenciamento de Nível de Serviço (SLM) em ambientes de TI. O sistema realiza gerência centrada em serviços monitorando um conjunto determinado de serviços de uma rede a partir da implementação de uma biblioteca de serviços baseada em Java e *scripts* Perl. O sistema é projetado de forma que o administrador da rede possa instanciar novos monitoramentos sem a necessidade de editar *scripts* no servidor de gerência (a menos que se queira criar um novo *script*). Além dessas funcionalidades, o sistema considera e implementa também uma métrica aqui denominada de “*Índice de Disponibilidade de Sistema - IDS*”. Esta métrica tem como principal objetivo fornecer um *feedback* quantitativo do funcionamento global do ambiente de TI monitorado.
- Publicação e apresentação do artigo “*A Proposed Web Tool for IT Environment Service Management*”, no *AICT’06 – 2nd Advanced International Conference on Telecommunication*, realizado em Guadalupe, French Caribbean, em fevereiro de 2006. Publicado nos *proceedings* do *IEEE Computer Society*.

6.2 Trabalhos Futuros

Alguns trabalhos futuros são sugeridos, a fim de enriquecer o trabalho desenvolvido nesta dissertação:

- Atualmente o sistema possui uma biblioteca de serviços limitada, ou seja, um conjunto de classes Java e *scripts* Perl que viabilizam o monitoramento de um conjunto limitado de serviços, como: DNS, WWW (HTTP e HTTPS), SMTP (E-mail), Telnet, FTP, Banco de Dados PostgreSQL, IP, Porta TCP e Comando. Se o administrador da rede desejar monitorar um outro serviço como NFS, por exemplo, atualmente ele tem duas opções: 1) monitorar a porta do serviço NFS (que não é tão interessante a fim de simular um usuário real) ou 2) criar um *script* que teste o serviço NFS como um

usuário real e monitorá-lo através da opção comando (que monitora a saída, 1 ou 0, da execução do *script*). Logo, seria interessante estender novas classes para novos serviços a fim de que o administrador da rede já tenha a opção pré-definida de monitoramento de outros serviços como, por exemplo, NFS, POP3, IMAP, DHCP, NIS, NTP, Banco de Dados MySQL, Oracle, SQLServer, etc.

- Incluir a gerência de configuração no SMS. Ou seja, desenvolver código para que detectado um problema de falha ou performance, tomar ações que possam identificar e corrigir possíveis problemas de configuração em um serviço. Pode ser interessante o desenvolvimento e a utilização de uma base de conhecimento de problemas.
- Estudar como integrar o gerenciamento distribuído de serviços no SMS através da MIB Script usando a plataforma (Jasmin, 2001), ou através de outra tecnologia, como agentes móveis usando a plataforma (Jade, 2000), se for mais viável.
- Aperfeiçoar e estender as funcionalidades da ferramenta como: cadastramento de usuários, cadastramento de novos *scripts* de monitoramento via interface *Web*, melhorar a interface *Web*, finalizar as funcionalidades de edição de serviços, clusters e grupos, realizar testes em ambientes reais maiores a fim de se avaliar o desempenho da ferramenta, armazenar dados consolidados por hora, semana e por mês (atualmente, armazena-se dados consolidados apenas por dia), reavaliar a estrutura das Classes e do Banco de Dados do SMS.
- Levantar novas funcionalidades que possam ser interessantes para o sistema como, por exemplo, gerar gráficos estatísticos de acompanhamento e especificar SLAs para outros parâmetros de qualidade como MTTR, MTBF e MTTF.

Referências Bibliográficas

BSC. Bsc – Balanced Scorecard. Disponível em: <<http://www.balancedscorecard.org/>>. Acesso em: 12/06/2005.

BISSEL, T. et al. **Service level management with agent technology**, ELSEVIER, Computer Networks, v.34, p.831-841, 2000.

CARMO, A. P. et al., **MBOX – A efficient and security connectivity solution**. In: IWT'04 – International Workshop on Telecommunication, 2004, Santa Rita do Sapucaí-MG-Brasil.

DRAGO, A. B. **Uma Ferramenta Computacional para Auxiliar a Prática de Gerência de Redes de Computadores**, Dissertação (Mestrado), Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória-ES, 2001.

GASPARY, L. P. **Gerenciamento Distribuído e Flexível de Protocolos de Alto Nível, Serviços e Aplicações em Redes de Computadores**, Tese (Doutorado), Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre-RS, 2002.

ITIL. A Introduction to ITIL. Disponível em: <<http://www.itpapers.com>>. Acesso em: 15/04/2005.

JADE. Jade – java agent development framework. Disponível em: <<http://jade.tilab.com/>>. Acesso em: 14/08/2005.

JASMIN. Jasmim – a script-mib implementation. Disponível em: <<http://www.ibr.cs.tu-bs.de/projects/jasmin/>>. Acesso em: 01/10/2005.

KIM, J. Y.; HONG, J. W. **Design and Implementation of a Web-based Internet/Intranet Mail Server Management System**. In: IEEE International Conference on Communications, 1999, Vancouver, Canada.

LEINWAND, A.; FANG, K.. **Network Management: A practical perspective**. 2nd Ed. Massachusetts: Addison Wesley, 1995.

LOPES, R. V. **Melhores Práticas para a Gerência de Redes de Computadores**. Dissertação (Mestrado). Programa de Pós-Graduação em Informática. Universidade Federal da Paraíba. Campina Grande-PB, 2002.

MENASCÉ, D.; ALMEIDA, V. **Capacity Planning For Web Services: Metrics, Models and Methods**. New Jersey: Prentice Hall, 1998.

MONTEIRO, M.; DRAGO, A. B.; GARCIA, A. S. **A Methodology for Network Performance Management**. In: 25^o Annual IEEE Conference on Local Computer Networks, 2000, Tampa, USA.

MOTA, T. et al. **Quality of Service Management in IP Networks using Mobile Agent Technology**, Lectures Notes on Computer Science, Springer Verlag, 2521, p.193-205, Berlim, Germany, 2002. Trabalho apresentado no IEEE/ACM International Workshop on Mobile Agents for Telecommunication Applications (MATA'2002), Barcelona, Spain, 2002.

MULLER, N. J. **Managing Service Level Agreements**. International Journal of Network Management, 1999, n.9, p.155-166, Jhon Wiley & Sons, Ltd.

NAGIOS. Nagios – sistema de gerência de redes e serviços. Disponível em: <<http://www.nagios.org>>. Acesso em: 10/03/2005.

NOGUEIRA, J. M. S.; BARBOSA, J. C. F. **Deteção e análise de falhas usando redes de dependência em sistemas distribuídos de gerenciamento**. In: 20^o Simpósio Brasileiro de Redes de Computadores, 2002, Búzios-RJ, Brasil.

RAMOS, A. S. **Proposta de Uma Ferramenta Computacional para Auxílio à Prática de Gerencia de Desempenho de Redes de Computadores**, Dissertação (Mestrado), Programa de Pós-Graduação em Informática. Universidade Federal do Espírito Santo. Vitória-ES, 2004.

RFC1155. ROSE, M.; MCCLOGHRIE, K. **Structure and identification of management information for TCP/IP-based internets**. 1990.

RFC1157. CASE, J. et al. **Simple Network Management Protocol (SNMP)**. 1990.

RFC1212. ROSE, M.; MCCLOGHRIE, K. **Concise MIB Definitions**. 1991.

RFC1213. ROSE, M.; MCCLOGHRIE, K. **Management Information Base for Network Management of TCP/IP-based Internets: MIB-II**. 1991.

RFC1611. AUSTEIN, R.; SAPERIA, J. **DNS Server MIB Extensions**. 1994.

RFC1697. BROWER, D. et al. **Relational Database Management System (RDBMS) Management Information Base (MIB) using SMIV2**. 1994.

RFC2021. WALDBUSSER, S. **Remote Network Monitoring Management Information Base Version 2 using SMIV2**. 1997.

RFC2594. HAZEWINDEL, H.; KALBFLEISCH, C.; SCHOENWAELDER, J. **Definitions of Managed Objects for WWW Services**. 1999.

RFC2789. FREED, N.; KILLE, S. **Mail Monitoring MIB**. 2000.

RFC3165. LEVI, D.; SCHOENWAELDER, J. **Definitions of Managed Objects for the Delegation of Management Scripts**, 2001.

RFC821. POSTEL, J. **Simple Mail Transfer Protocol**, 1982.

RFC959. POSTEL, J.; REYNOLDS, J. **File Transfer Protocol (FTP)**, 1985.

RODOSEK, G. D. **A Framework for IT Service Management**, Tese (Doutorado), Faculdade de Matemática e Informática, Universidade de Munique, Munich, Germany, 2002.

STALLINGS, William. **SNMPv1, SNMPv2, SNMPv3 e RMON 1 e 2**. 3rd Ed. Massachusetts: Addison Wesley, 1999.

STHEPHAN, R.; RAY, P.; PARAMESH, N. **Network management platform based on mobile agents**. International Journal of Network Management, 2004, n.14, 59-73, Jhon Wiley & Sons, Ltd.

VILLAÇA, R. S. et al. **A Web-based pro-active fault and performance network management architecture**, Lectures Notes on Computer Science, Springer Verlag, v.3124, p.792-800, Berlim, Germany, 2004. Trabalho apresentado no 11º International Conference on Telecommunication (ICT'04), Fortaleza, Brazil, 2004.

VILLAÇA, R. S. **Uma Abordagem para Gerência Pró-Ativa de Falhas e Desempenho em um Centro Remoto de Operações de Rede**. Dissertação (Mestrado), Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória-ES, 2004.

Apêndices

Apêndice A - Fluxogramas de Monitoramento dos Serviços

Este apêndice descreve como é implementado no SMS o monitoramento de cada serviço mostrado na seção 4.2.3, ou seja, a forma de monitoramento descrita para cada serviço naquela seção.

Monitoramento do Serviço WWW (HTTP ou HTTPS)

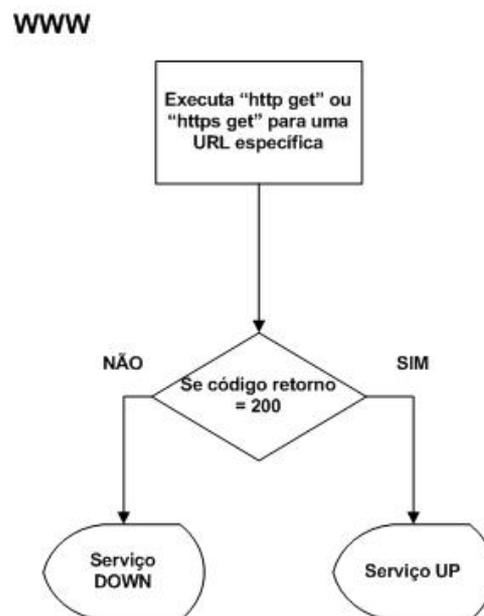


Figura 25 - Fluxograma do monitoramento do serviço WWW(HTTP e HTTPS).

Método Coletar() da Classe de Coleta “WWW”

```

1   public String coletar(ResultadoAtual resAtual){
2       int disponibilidade = 0, codigoHttpRetornado = 0, timeOut =
180000, HTTP_OK = 200;
3       long t1 = 0, t2 = 0;
4       float tempoResposta = 0;
5       String mensagem = null;
6
7       if (getUrl().substring(0,5).equalsIgnoreCase("http:")) {
8           /**
9            * HTTP GET para URL que usa o protocolo HTTP.
10          */
11          try {
12              URL u = new URL(getUrl());
  
```

```

13         HttpURLConnection conn = (HttpURLConnection)
u.openConnection();
14         conn.setDefaultUseCaches(false);
15         conn.setConnectTimeout(timeOut);
16         // Marcar o tempo de resposta.
17         t1 = System.currentTimeMillis();
18         codigoHttpRetornado = conn.getResponseCode();
19         t2 = System.currentTimeMillis();
20         conn.disconnect();
21     } catch (MalformedURLException e) {
22         mensagem = "URL mal formada.";
23     } catch (SocketTimeoutException e) {
24         mensagem = "Estouro de Timeout na conexao.";
25     } catch (IOException e) {
26         mensagem = "Erro de I/O.";
27     } catch (Exception e) {
28         mensagem = e.getMessage();
29     }
30     } else if (getUrl().substring(0,5).equalsIgnoreCase("https")) {
31         /**
32          * HTTPS GET para URL que usa o protocolo HTTPS (HTTP + SSL).
33          * Criar um Trust Manager para aceitar todos os certificados
oferecidos pelas URLs que usam HTTPS.
34          * "Create a trust manager that does not validate certificate
chains"
35          */
36         TrustManager[] trustAllCerts = new TrustManager[]{
37             new X509TrustManager() {
38                 public java.security.cert.X509Certificate[]
getAcceptedIssuers() {
39                     return null;
40                 }
41                 public void
checkClientTrusted(java.security.cert.X509Certificate[] certs, String
authType) {
42                     }
43                 public void
checkServerTrusted(java.security.cert.X509Certificate[] certs, String
authType) {
44                     }
45                 }
46         };
47         /**
48          * "Install the all-trusting trust manager"
49          */
50         try {
51             SSLContext sc = SSLContext.getInstance("SSL");
52             sc.init(null, trustAllCerts, new
java.security.SecureRandom());
53         } catch (Exception e) {
54             mensagem = "Falhou instalacao do trust manager";
55         }
56     }
57     try {
58         URL u = new URL(getUrl());
59         HttpURLConnection conn = (HttpURLConnection)
u.openConnection();
60         conn.setDefaultUseCaches(false);

```

```

62         conn.setConnectTimeout(timeOut);
63         // Marcar o tempo de resposta.
64         t1 = System.currentTimeMillis();
65         codigoHttpRetornado = conn.getResponseCode();
66         t2 = System.currentTimeMillis();
67         conn.disconnect();
68     }catch (MalformedURLException e) {
69         mensagem = "URL mal formada.";
70     }catch (SocketTimeoutException e) {
71         mensagem = "Estouro de Timeout na conexao.";
72     }catch (IOException e){
73         mensagem = "Erro de I/O.";
74     }catch (Exception e){
75         mensagem = e.getMessage();
76     }
77 } else {
78     mensagem = "URL mal formada: nao iniciada com http ou
https.";
79 }
100
101
102 /**
103  * Tempo de Resposta em segundos.
104  * Codigo HTTP: HTTP_OK = 200
105  */
106 if (codigoHttpRetornado == HTTP_OK){
107     disponibilidade = 1;
108     tempoResposta = (float)(t2 - t1)/1000;
109 }
110 resAtual.setDisponibilidade(disponibilidade);
111 resAtual.setTempoResposta(tempoResposta);
112 resAtual.setTimeAtual(System.currentTimeMillis());
113 return mensagem;
114 }

```

Monitoramento do Serviço DNS

DNS

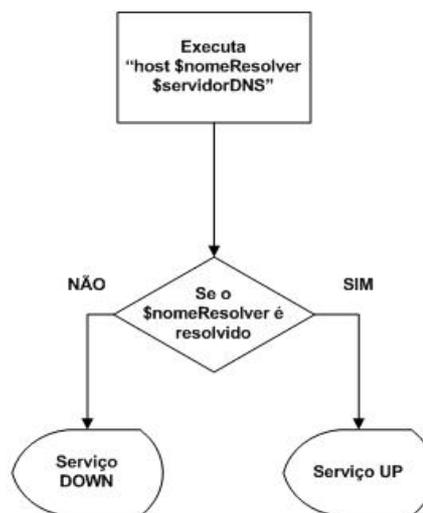


Figura 26 - Fluxograma do monitoramento do serviço DNS.

Método Coletar() da Classe de Coleta “DNS”

```

1 public String coletar(ResultadoAtual resAtual) {
2     String comando = null;
3     int disponibilidade = 0;
4     long t1 = 0, t2 = 0;
5     float tempoResposta = 0;
6     String mensagem = null;
7
8     try {
9         // Executa o comando
10        comando = "/home/bonfa/mestrado/SMS/PerlScripts/dns.pl " +
getNomeResolver() + " " + getIpServidor();
11        t1 = System.currentTimeMillis();
12        Process child = Runtime.getRuntime().exec(comando);
13        t2 = System.currentTimeMillis();
14
15        // Le a saida do comando executado.
16        InputStream in = child.getInputStream();
17        int c;
18        if ((c = in.read()) != -1) {
19            if (c == '1'){
20                disponibilidade = 1;
21                tempoResposta = (float)(t2 - t1)/1000;
22            }else {
23                disponibilidade = 0;
24                tempoResposta = 0f;
25            }
26        }
27        in.close();
28    } catch (IOException e) {
29        disponibilidade = 0;
30        tempoResposta = 0f;
31        mensagem = "Falha na execucao do comando: " + comando;
32    }
33
34    resAtual.setDisponibilidade(disponibilidade);
35    resAtual.setTempoResposta(tempoResposta);
36    resAtual.setTimeAtual(System.currentTimeMillis());
37    return mensagem;
38 }

```

Script Perl para Monitoramento do Serviço DNS – dns.pl

```

1     #!/usr/bin/perl5.8.6
2
3     my $nomeResolver = $ARGV[0];
4     my $servidorDNS = $ARGV[1];
5     my $contador = 0;
6
7     # Executa o comando "host" e verifica se o servidor resolve o nome
especificado.
8     my @saida = `host $nomeResolver $servidorDNS`;
9
10    foreach (@saida) {
11        if (/has address/) {
12            # Incrementa o numero de enderecos IP que o nome possui.
13            $contador = $contador + 1;
14        }

```

```

15     }
16
17     # Se contador for maior que 0, significa que o nome foi resolvido
para pelo
18     # menos um endereço IP. OK.
19     if ($contador > 0) {
20         print "1\n";
21     }else {
22         print "0\n";
23     }

```

Monitoramento do Serviço TELNET

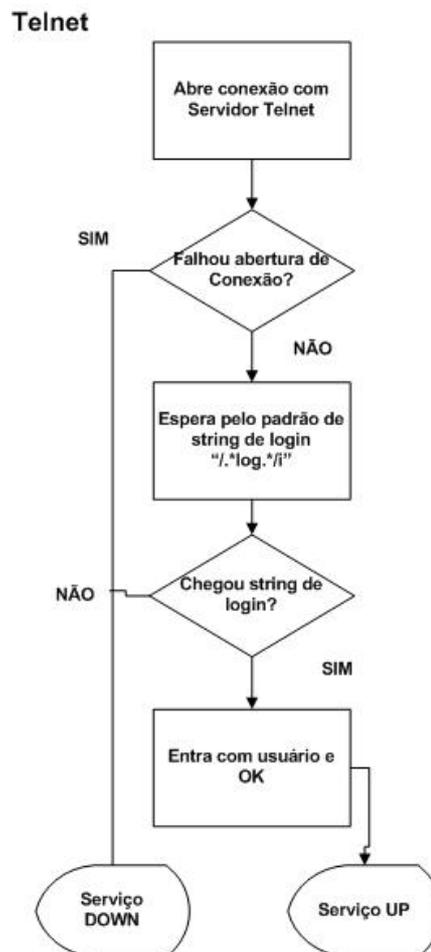


Figura 27 - Fluxograma do monitoramento do serviço Telnet.

Método Coletar() da Classe de Coleta “Telnet”

Idem método Coletar() da Classe de Coleta DNS. A única alteração é o nome do *script* da variável *comando*, onde em vez de “dns.pl” tem-se “telnet.pl”.

Script Perl para Monitoramento do Serviço Telnet – telnet.pl

```

1     #!/usr/bin/perl5.8.6
2
3     use Net::Telnet;

```

```

8   my $servidor = $ARGV[0];
9   my $usuario = $ARGV[1];
10  my $passwd = $ARGV[2];
11
12  # Cria um objeto do tipo Net::Telnet que implementa a conexao de
rede.
13  # A opcao Errmode, serve para o processo nao morrer com um erro caso
haja
14  # falha ou erro de conexao com o servidor.
15  $telnet = Net::Telnet->new(Errmode => "return") or ( print "0\n" and
exit);
16
17  # Abre uma conexao TELNET com o servidor.
18  $telnet->open($servidor) or ( print "0\n" and exit);
19
20  # Loga no servidor com usuario e senha.
21  $telnet->waitfor('/.*log.*/i') or ( print "0\n" and exit);
22  $telnet->print($usuario) or ( print "0\n" and exit);
23
24  # Espera pela mensagem de passwd. Se vier, entra com a senha.
25  # Se nao vier, considera o servico UP,
26  # pois a mensagem de login chegou anteriormente enviada.
27  # Considero isso suficiente para afirma disponibilidade do sevcio
TELNET.
28  $telnet->waitfor('/.*pas.*/i') or ( print "1\n" and exit);
29  $telnet->print($passwd) or ( print "0\n" and exit);
30  # Espera pela mensagem DEFAULT de sucesso do Login.
31  $telnet->waitfor('/Have a lot of fun/') or ( print "0\n" and exit);
32
33  # Fecha conexao com o servidor.
34  $telnet->close() or ( print "0\n" and exit);
35
36  # Se tudo OK, print 1.
37  print "1\n";

```

Monitoramento do Serviço COMANDO

Comando

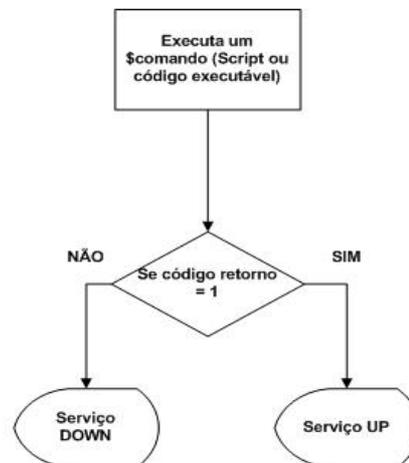


Figura 28 - Fluxograma do monitoramento do serviço Comando.

Método Coletar() da Classe de Coleta “Comando”

```

1  public String coletar(ResultadoAtual resAtual) {
2      String comando = null;
3      int disponibilidade = 0;
4      long t1 = 0, t2 = 0;
5      float tempoResposta = 0;
6      String mensagem = null;
7
8      try {
9          // Executa o comando
10         comando = "/home/bonfa/mestrado/SMS/Comandos/" +
getComando();
11         t1 = System.currentTimeMillis();
12         Process child = Runtime.getRuntime().exec(comando);
13         t2 = System.currentTimeMillis();
14
15         // Le a saida do comando executado.
16         InputStream in = child.getInputStream();
17         int c;
18         if ((c = in.read()) != -1) {
19             if (c == '1'){
20                 disponibilidade = 1;
21                 tempoResposta = (float)(t2 - t1)/1000;
22             }else {
23                 disponibilidade = 0;
24                 tempoResposta = 0f;
25             }
26         }
27         in.close();
28     } catch (IOException e) {
29         disponibilidade = 0;
30         tempoResposta = 0f;
31         mensagem = "Falha na execucao do comando: " + comando;
32     }
33
34     resAtual.setDisponibilidade(disponibilidade);
35     resAtual.setTempoResposta(tempoResposta);
36     resAtual.setTimeAtual(System.currentTimeMillis());
37     return mensagem;
38 }

```

Monitoramento do Serviço FTP

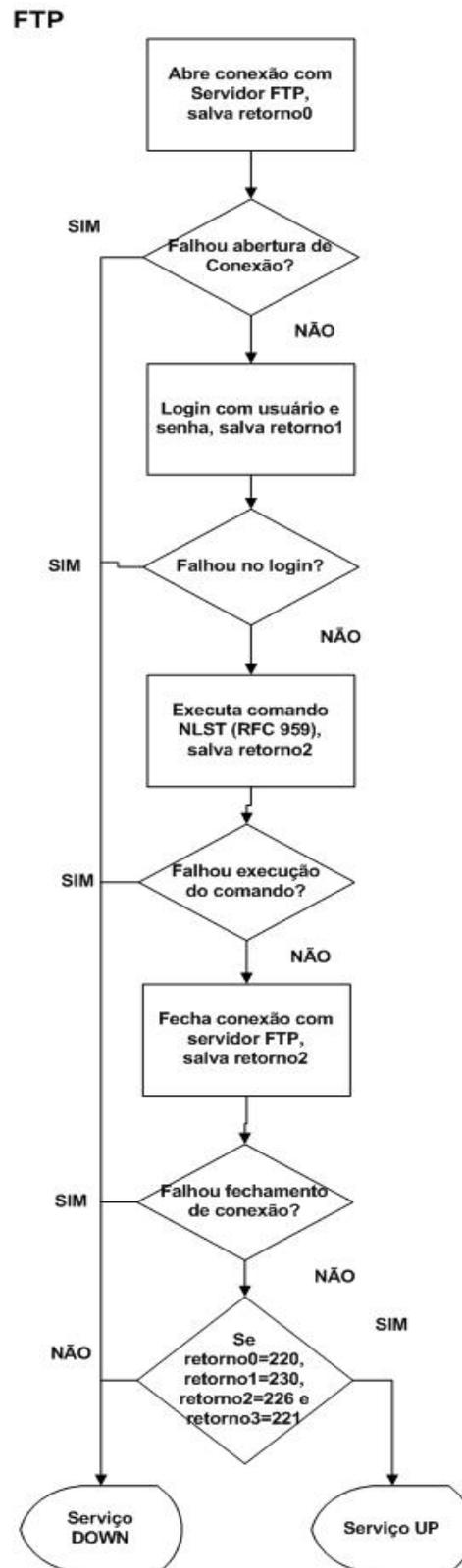


Figura 29 - Fluxograma do monitoramento do serviço FTP.

Método Coletar() da Classe de Coleta “FTP”

Idem método Coletar() da Classe de Coleta DNS. A única alteração é o nome do *script* da variável *comando*, onde em vez de “dns.pl” tem-se “ftp.pl”.

Script Perl para Monitoramento do Serviço FTP – ftp.pl

```

1      #!/usr/bin/perl5.8.6
2
3      use Net::FTP;
4
5      my $servidor = $ARGV[0];
6      my $usuario = $ARGV[1];
7      my $passwd = $ARGV[2];
8
9      # Abre uma conexao FTP com o servidor.
10     $ftp = Net::FTP->new($servidor, Timeout => 120) or ( print "0\n" and
exit);
11     my $codigoRetornado0 = $ftp->code;
12
13     # Loga com usuario e senha no servidor.
14     $ftp->login($usuario, $passwd) or ( print "0\n" and exit);
15     my $codigoRetornado1 = $ftp->code;
16
17     # Executa o comando NLST (RFC 959).
18     $ftp->ls("./") or ( print "0\n" and exit);
19     my $codigoRetornado2 = $ftp->code;
20
21     # Fecha a conexao FTP.
22     $ftp->quit or ( print "0\n" and exit);
23     my $codigoRetornado3 = $ftp->code;
24
25     # Se tudo OK, print 1.
26     if ($codigoRetornado0 == 220 && $codigoRetornado1 == 230 &&
$codigoRetornado2 == 226 && $codigoRetornado3 == 221) {
27         print "1\n";
28     } else{
29         print "0\n";
30     }

```

Monitoramento do Serviço BANCO DE DADOS

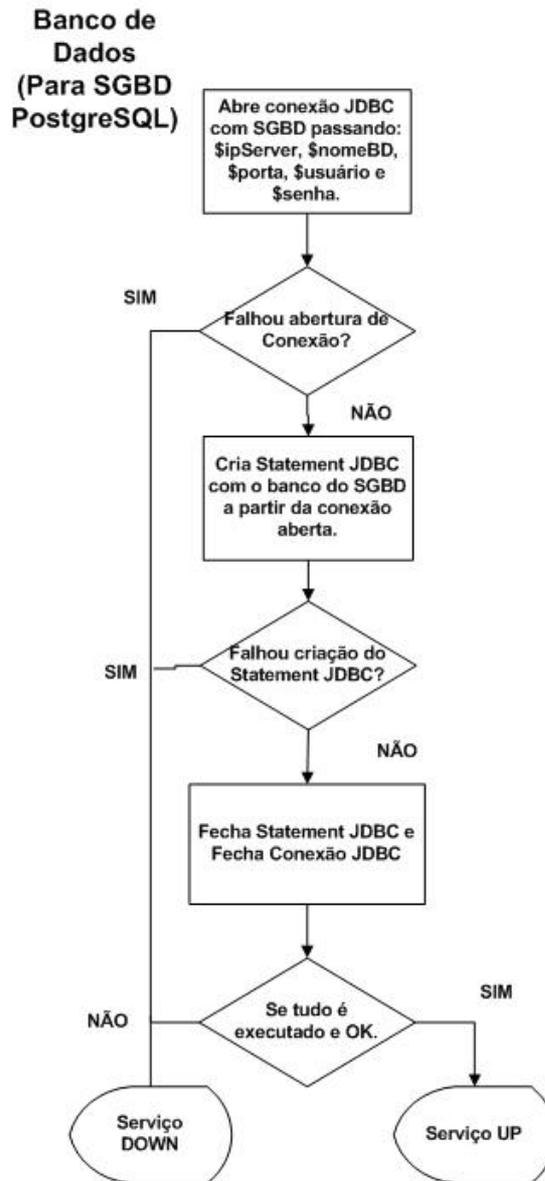


Figura 30 - Fluxograma do monitoramento do serviço de Banco de Dados.

Método Coletar() da Classe de Coleta “PostgreSQL”

```

1 public String coletar(ResultadoAtual resAtual) {
2
3     int disponibilidade = 0;
4     long t1 = 0, t2 = 0;
5     float tempoResposta = 0;
6     String mensagem = null;
7
8     String url;
10    Connection connection;
11    Statement statement;
12
13    url= "jdbc:postgresql://" + getIpServidor() + ":" + getPorta() +
"/" + getNomeBanco();

```

```

14
15     try {
16         // Abrindo e fechando uma conexão com o BD.
17         t1 = System.currentTimeMillis();
18         Class.forName("org.postgresql.Driver");
19         connection = DriverManager.getConnection(url,
getUsuario(), getSenha());
20         statement = connection.createStatement();
21         statement.close();
22         connection.close();
23         t2 = System.currentTimeMillis();
24
25         disponibilidade = 1;
26         tempoResposta = (float)(t2 - t1)/1000;
27
28     }catch (ClassNotFoundException cnfex){
29         mensagem = "Banco: " + getNomeBanco() + " Porta: " +
getPorta() + " Falha no carregamento do driver do PostgreSQL!!!";
30         disponibilidade = 0;
31         tempoResposta = 0f;
32     }catch (SQLException sqllex){
33         mensagem = "Banco: " + getNomeBanco() + " Porta: " +
getPorta() + " Falha na conexão como BD PostgreSQL!!!";
34         disponibilidade = 0;
35         tempoResposta = 0f;
36     }
37
38     resAtual.setDisponibilidade(disponibilidade);
39     resAtual.setTempoResposta(tempoResposta);
40     resAtual.setTimeAtual(System.currentTimeMillis());
41     return mensagem;
42 }

```

Monitoramento do Serviço PORTA TCP

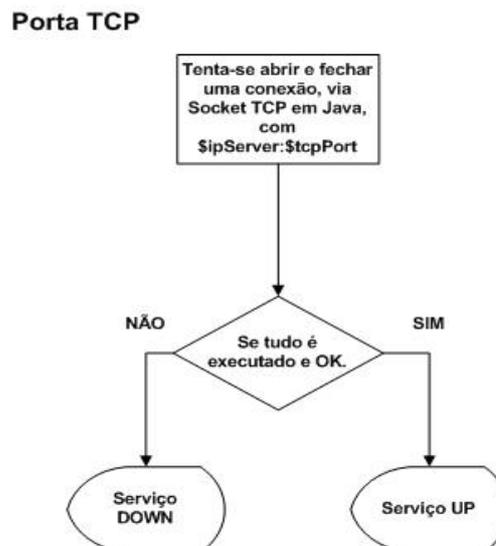


Figura 31 - Fluxograma do monitoramento do serviço Porta TCP.

Método Coletar() da Classe de Coleta "PortaTCP"

```

1  public String coletar(ResultadoAtual resAtual) {
2
3      int disponibilidade = 0;
4      long t1 = 0, t2 = 0;
5      float tempoResposta = 0;
6      String mensagem = null;
7
8      // Criando um socket com timeout.
9      try {
10         InetAddress addr = InetAddress.getByName(getIpServidor());
11         SocketAddress sockaddr = new InetSocketAddress(addr,
getPorta());
12
13         // Criando o objeto socket sem estabelecer conexao.
14         Socket sock = new Socket();
15
16         // Este metodo ficara bloqueado no maximo o tempo de timeout.
17         // Depois do tempo de timeout, a excessao
SocketTimeoutException e lancada.
18         int timeoutMs = 6000;    // 60 seconds
19
20         t1 = System.currentTimeMillis();
21         sock.connect(sockaddr, timeoutMs);
22         sock.close();
23         t2 = System.currentTimeMillis();
24
25         disponibilidade = 1;
26         tempoResposta = (float)(t2 - t1)/1000;
27
28     } catch (UnknownHostException e) {
29         disponibilidade = 0;
30         tempoResposta = 0f;
31         mensagem = this.getIpServidor() + ": host desconhecido para
monitoramento de porta TCP.";
32     } catch (SocketTimeoutException e) {
33         disponibilidade = 0;
34         tempoResposta = 0f;
35         mensagem = "Timeout no monitoramento de porta TCP: host:" +
getIpServidor() +" Porta: " + getPorta() + ".";
36     } catch (IOException e) {
37         disponibilidade = 0;
38         tempoResposta = 0f;
39         mensagem = "Excessao no monitoramento de porta TCP: host:" +
getIpServidor() +" Porta: " + getPorta() + ".";
40     }
41
42     resAtual.setDisponibilidade(disponibilidade);
43     resAtual.setTempoResposta(tempoResposta);
44     resAtual.setTimeAtual(System.currentTimeMillis());
45     return mensagem;
46 }

```

Monitoramento do Serviço IP

IP via SNMP

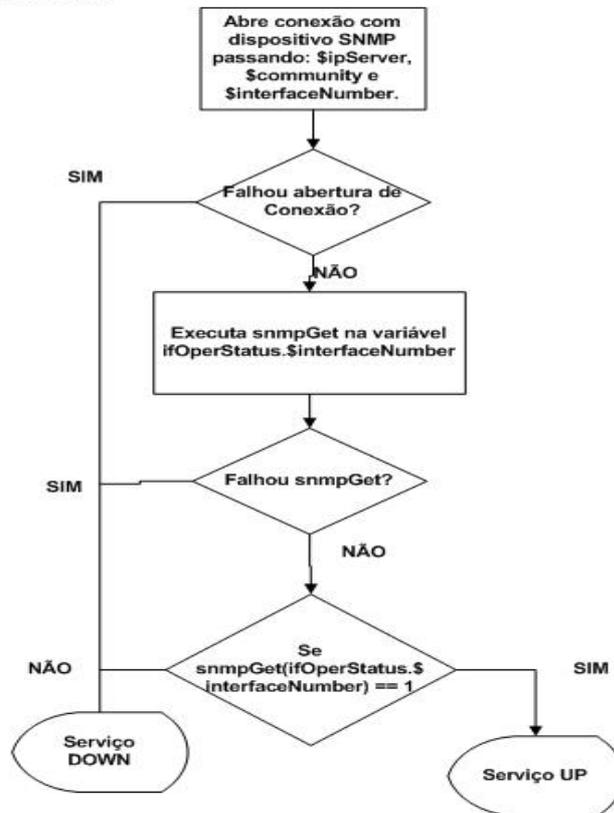


Figura 32 - Fluxograma do monitoramento do serviço IP usando o SNMP.

IP via ICMP (Ping)

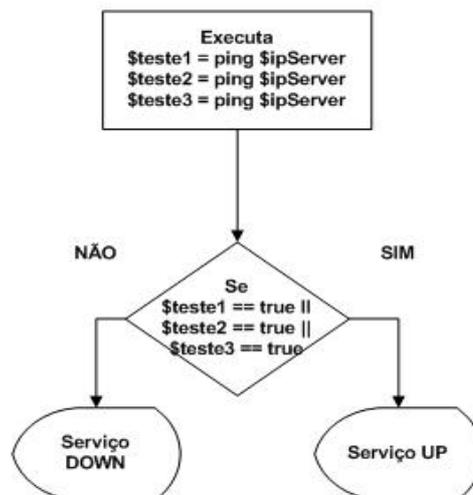


Figura 33 - Fluxograma do monitoramento do serviço IP usando o Ping (ICMP).

Método Coletar() da Classe de Coleta “IP”

```

1   public String coletar(ResultadoAtual resAtual) {
2       String comando = null;
3       int disponibilidade = 0;
4       long t1 = 0, t2 = 0;
5       float tempoResposta = 0;
6       String mensagem = null;
7
8       try {
9
10          // Verificar se o monitoramento e via icmp (ping) ou snmp
11          (ifOperStatus da Mib-2).
12          if (protocolo.compareTo("snmp") == 0) {
13              comando =
14              "/home/bonfa/mestrado/SMS/PerlScripts/disponibilidade_snmp.pl " +
15              getIpServidor() + " " + getCommunity() + " " + getInterf();
16          }else {
17              comando =
18              "/home/bonfa/mestrado/SMS/PerlScripts/disponibilidade_ping.pl " +
19              getIpServidor();
20          }
21
22          // Executa o comando
23          t1 = System.currentTimeMillis();
24          Process child = Runtime.getRuntime().exec(comando);
25          t2 = System.currentTimeMillis();
26
27          // Le a saida do comando executado.
28          InputStream in = child.getInputStream();
29          int c;
30          if ((c = in.read()) != -1) {
31              if (c == '1'){
32                  disponibilidade = 1;
33                  tempoResposta = (float)(t2 - t1)/1000;
34              }else {
35                  disponibilidade = 0;
36                  tempoResposta = 0f;
37              }
38          }
39          in.close();
40      } catch (IOException e) {
41          disponibilidade = 0;
42          tempoResposta = 0f;
43          mensagem = "Falha na execucao do comando: " + comando;
44      }
45
46      resAtual.setDisponibilidade(disponibilidade);
47      resAtual.setTempoResposta(tempoResposta);
48      resAtual.setTimeAtual(System.currentTimeMillis());
49      return mensagem;
50  }

```

Script Perl para Monitoramento do Serviço IP (Ping - ICMP) – disponibilidade_ping.pl

```

1   #!/usr/bin/perl5.8.6
2
3   use Net::Ping::External qw(ping);
4
5   my $servidor = $ARGV[0];
6
7   my $alive0 = ping( host => $servidor);
8   my $alive1 = ping( host => $servidor);
9   my $alive2 = ping( host => $servidor);
10
11
12  if ($alive0 || $alive1 || $alive2) {
13  print "1\n";
14  } else {
15      print "0\n";
16  }

```

Script Perl para Monitoramento do Serviço IP (SNMP) – disponibilidade_snmp.pl

```

1   #!/usr/bin/perl5.8.6
2
3   use Net::SNMP;
4
5   my $servidor = $ARGV[0];
6   my $community = $ARGV[1];
7   my $interface = $ARGV[2];
8
9   my ($session, $error) = Net::SNMP->session(
10  -hostname => $servidor,
11  -community => $community,
12  -port => 161
13  );
14
15  if (!defined($session)) {
16      print "0\n";
17      exit;
18  }
19
20  my $ifOperStatus = "1.3.6.1.2.1.2.2.1.8.$interface";
21
22  my $result = $session->get_request(
23      -varbindlist => [$ifOperStatus]
24  );
25  if (!defined($result)) {
26      print "0\n";
27      $session->close;
28      exit;
29  }
30  if ($result->{$ifOperStatus} == 1) {
31      print "1\n";
32  }else {
33      print "0\n";
34  }
35  $session->close;

```

Apêndice B - Modelo de Classes e Modelo Lógico do Banco de Dados do Sistema

Este apêndice apresenta as classes desenvolvidas para a implementação do SMS, ou seja, as funcionalidades e a lógica do sistema apresentados no capítulo 4. Também é apresentado neste apêndice o diagrama lógico do banco de dados (diagrama E-R) que armazena as informações do SMS, assim como o *script* sql para a criação desse banco de dados.

Classes de Persistência do SMS

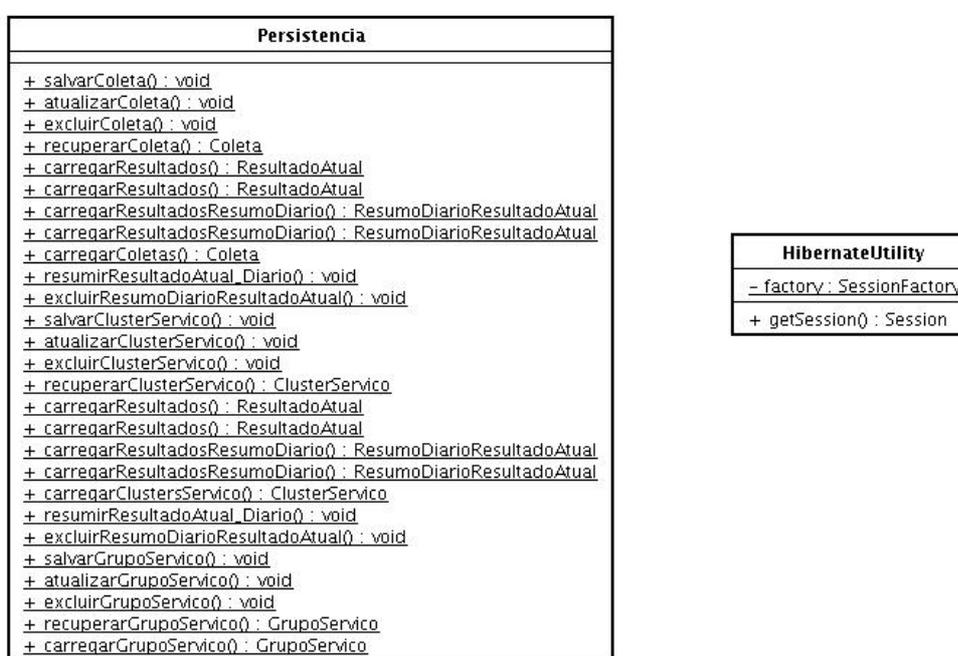


Figura 34 - Classes de persistência.

Classe para Criar os Gráficos dos Relatórios do SMS



Figura 35 - Classe para gerar gráficos.

Classes do Kernel do SMS

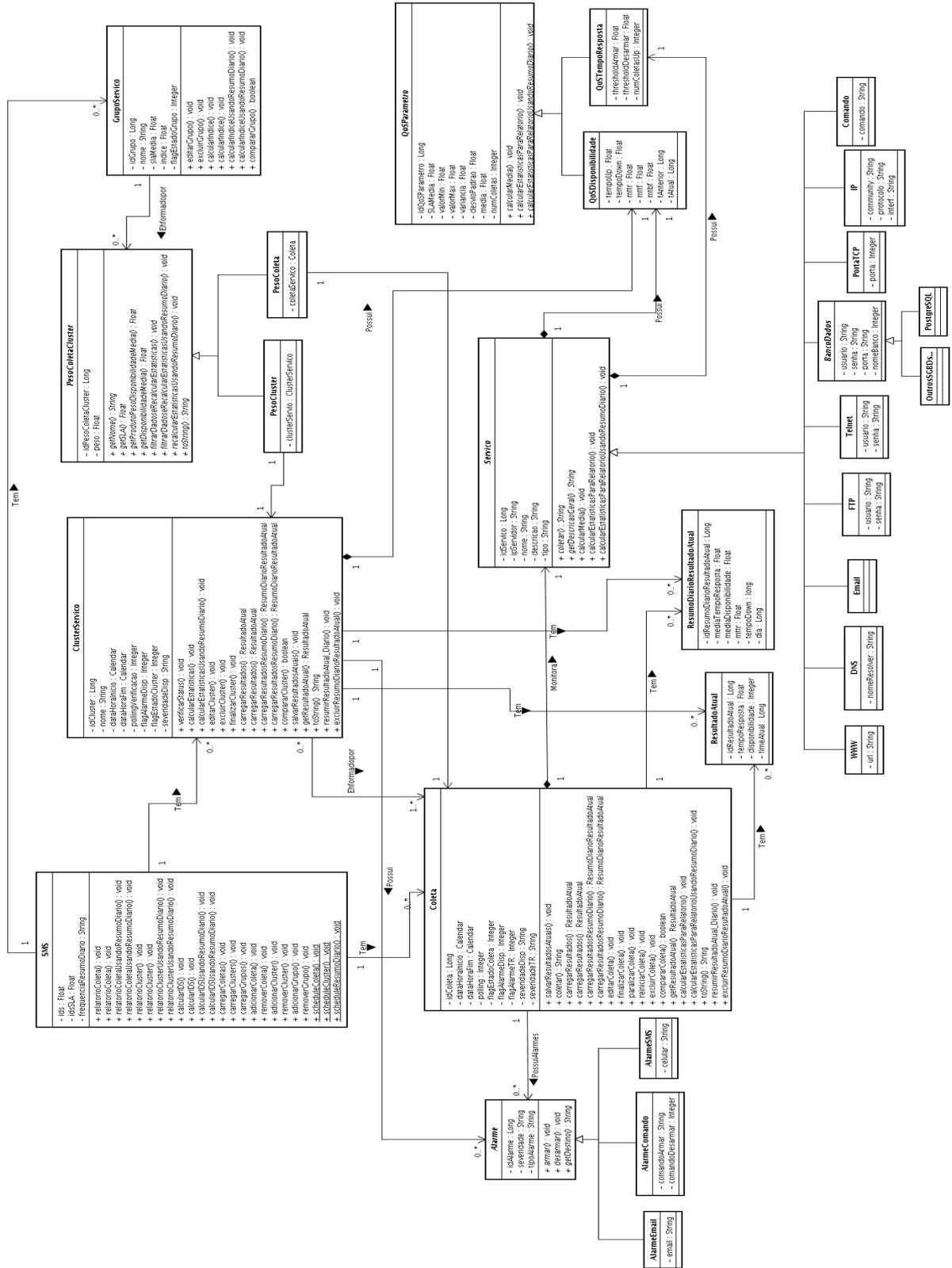


Figura 36 - Classes do kernel do SMS.

Segue abaixo o *script* SQL (*Structure Query Language*) usado para criação do banco de dados esboçado na Figura 38 (*bd_sms.sql*).

```

DROP TABLE GrupoServico;
DROP TABLE PesoColetaCluster;
DROP TABLE PesoColeta;
DROP TABLE PesoCluster;
DROP TABLE Coleta;
DROP TABLE ClusterServico;
DROP TABLE ClusterServicoTemColeta;
DROP TABLE Alarme;
DROP TABLE ResultadoAtual;
DROP TABLE ResumoDiarioResultadoAtual;
DROP TABLE Servico;
DROP TABLE QoSParametro;
DROP TABLE QoSTempoResposta;
DROP TABLE AlarmeSMS;
DROP TABLE AlarmeEmail;
DROP TABLE AlarmeComando;
DROP TABLE QoSDisponibilidade;
DROP TABLE WWW;
DROP TABLE DNS;
DROP TABLE Email;
DROP TABLE FTP;
DROP TABLE Telnet;
DROP TABLE BancoDados;
DROP TABLE PostgreSQL;
DROP TABLE PortaTCP;
DROP TABLE IP;
DROP TABLE Comando;

CREATE TABLE GrupoServico (
    idGrupo BIGINT NOT NULL,
    nome VARCHAR(50) NOT NULL,
    slaMedia FLOAT NOT NULL,
    indice FLOAT NOT NULL,
    flagEstadoGrupo INT NOT NULL,
    PRIMARY KEY(idGrupo)
);

CREATE TABLE PesoColetaCluster (
    idPesoColetaCluster BIGINT NOT NULL,
    GrupoServico_idGrupo BIGINT,
    peso FLOAT NOT NULL,
    PRIMARY KEY(idPesoColetaCluster)
);

CREATE TABLE PesoColeta (
    PesoColetaCluster_idPesoColetaCluster BIGINT NOT NULL,
    Coleta_idColeta BIGINT NOT NULL,
    PRIMARY KEY(PesoColetaCluster_idPesoColetaCluster)
);

CREATE TABLE PesoCluster (
    PesoColetaCluster_idPesoColetaCluster BIGINT NOT NULL,
    ClusterServico_idCluster BIGINT NOT NULL,
    PRIMARY KEY(PesoColetaCluster_idPesoColetaCluster)
);

```

```

CREATE TABLE Coleta (
  idColeta BIGINT NOT NULL,
  Servico_idServico BIGINT NOT NULL,
  dataHoraInicio DATE NOT NULL,
  dataHoraFim DATE NOT NULL,
  polling INT NOT NULL,
  flagEstadoColeta INT NOT NULL,
  flagAlarmeDisp INT NOT NULL,
  flagAlarmeTR INT NOT NULL,
  severidadeDisp VARCHAR(30),
  severidadeTR VARCHAR(30),
  PRIMARY KEY(idColeta)
);

CREATE TABLE ClusterServico (
  idCluster BIGINT NOT NULL,
  QoSDisponibilidade_QoSParametro_idQoSParametro BIGINT NOT NULL,
  nome VARCHAR(50) NOT NULL,
  dataHoraInicio DATE NOT NULL,
  dataHoraFim DATE NOT NULL,
  pollingVerificacao INT NOT NULL,
  flagAlarmeDisp INT NOT NULL,
  flagEstadoCluster INT NOT NULL,
  severidadeDisp VARCHAR(30),
  PRIMARY KEY(idCluster)
);

CREATE TABLE ClusterServicoTemColeta (
  Coleta_idColeta BIGINT NOT NULL,
  ClusterServico_idCluster BIGINT NOT NULL,
  PRIMARY KEY(Coleta_idColeta, ClusterServico_idCluster)
);

CREATE TABLE Alarme (
  idAlarme BIGINT NOT NULL,
  ClusterServico_idCluster BIGINT,
  Coleta_idColeta BIGINT,
  severidade VARCHAR(30) NOT NULL,
  tipoAlarme VARCHAR(30) NOT NULL,
  PRIMARY KEY(idAlarme)
);

CREATE TABLE ResultadoAtual (
  idResultadoAtual SERIAL PRIMARY KEY,
  ClusterServico_idCluster BIGINT,
  Coleta_idColeta BIGINT,
  tempoResposta FLOAT NOT NULL,
  disponibilidade INT NOT NULL,
  timeAtual BIGINT NOT NULL
);

CREATE TABLE ResumoDiarioResultadoAtual (
  idResumoDiarioResultadoAtual SERIAL PRIMARY KEY,
  ClusterServico_idCluster BIGINT,
  Coleta_idColeta BIGINT,
  mediaTempoResposta FLOAT NOT NULL,
  mediaDisponibilidade FLOAT NOT NULL,
  mtrr FLOAT NOT NULL,
  tempoDown FLOAT NOT NULL,

```

```
    dia BIGINT NOT NULL
);

CREATE TABLE Servico (
    idServico BIGINT NOT NULL,
    QoSTempoResposta_QoSParametro_idQoSParametro BIGINT NOT NULL,
    QoSDisponibilidade_QoSParametro_idQoSParametro BIGINT NOT NULL,
    ipServidor VARCHAR(30) NOT NULL,
    nome VARCHAR(50) NOT NULL,
    descricao VARCHAR(100) NOT NULL,
    tipo VARCHAR(30) NULL,
    PRIMARY KEY(idServico)
);

CREATE TABLE QoSParametro (
    idQoSParametro BIGINT NOT NULL,
    SLAMedia FLOAT NOT NULL,
    valorMin FLOAT NOT NULL,
    valorMax FLOAT NOT NULL,
    variancia FLOAT NOT NULL,
    desvioPadrao FLOAT NOT NULL,
    media FLOAT NOT NULL,
    numColetas INT NOT NULL,
    PRIMARY KEY(idQoSParametro)
);

CREATE TABLE QoSTempoResposta (
    QoSParametro_idQoSParametro BIGINT NOT NULL,
    thresholdArmar FLOAT NOT NULL,
    thresholdDesarmar FLOAT NOT NULL,
    numColetasUp INT NOT NULL,
    PRIMARY KEY(QoSParametro_idQoSParametro)
);

CREATE TABLE AlarmeSMS (
    Alarme_idAlarme BIGINT NOT NULL,
    celular VARCHAR(30) NOT NULL,
    PRIMARY KEY(Alarme_idAlarme)
);

CREATE TABLE AlarmeEmail (
    Alarme_idAlarme BIGINT NOT NULL,
    email VARCHAR(50) NOT NULL,
    PRIMARY KEY(Alarme_idAlarme)
);

CREATE TABLE AlarmeComando (
    Alarme_idAlarme BIGINT NOT NULL,
    comandoArmar VARCHAR(50) NOT NULL,
    comandoDesarmar VARCHAR(50) NOT NULL,
    PRIMARY KEY(Alarme_idAlarme)
);

CREATE TABLE QoSDisponibilidade (
    QoSParametro_idQoSParametro BIGINT NOT NULL,
    tempoUp FLOAT NOT NULL,
    tempoDown FLOAT NOT NULL,
    mttr FLOAT NOT NULL,
    mttf FLOAT NOT NULL,
    mtbf FLOAT NOT NULL,
```

```
tAnterior BIGINT NOT NULL,  
tAtual BIGINT NOT NULL,  
PRIMARY KEY(QoSParametro_idQoSParametro)  
);  
  
CREATE TABLE WWW (  
  Servico_idServico BIGINT NOT NULL,  
  url VARCHAR(100) NOT NULL,  
  PRIMARY KEY(Servico_idServico)  
);  
  
CREATE TABLE DNS (  
  Servico_idServico BIGINT NOT NULL,  
  nomeResolver VARCHAR(100) NOT NULL,  
  PRIMARY KEY(Servico_idServico)  
);  
  
CREATE TABLE Email (  
  Servico_idServico BIGINT NOT NULL,  
  PRIMARY KEY(Servico_idServico)  
);  
  
CREATE TABLE FTP (  
  Servico_idServico BIGINT NOT NULL,  
  usuario VARCHAR(30) NOT NULL,  
  senha VARCHAR(30) NOT NULL,  
  PRIMARY KEY(Servico_idServico)  
);  
  
CREATE TABLE Telnet (  
  Servico_idServico BIGINT NOT NULL,  
  usuario VARCHAR(30) NOT NULL,  
  senha VARCHAR(30) NOT NULL,  
  PRIMARY KEY(Servico_idServico)  
);  
  
CREATE TABLE BancoDados (  
  Servico_idServico BIGINT NOT NULL,  
  usuario VARCHAR(30) NOT NULL,  
  senha VARCHAR(30) NOT NULL,  
  porta VARCHAR(30) NOT NULL,  
  nomeBanco VARCHAR(30) NOT NULL,  
  PRIMARY KEY(Servico_idServico)  
);  
  
CREATE TABLE PostgreSQL (  
  BancoDados_Servico_idServico BIGINT NOT NULL,  
  PRIMARY KEY(BancoDados_Servico_idServico)  
);  
  
CREATE TABLE PortaTCP (  
  Servico_idServico BIGINT NOT NULL,  
  porta INT NOT NULL,  
  PRIMARY KEY(Servico_idServico)  
);  
  
CREATE TABLE IP (  
  Servico_idServico BIGINT NOT NULL,  
  community VARCHAR(30),  
  interf VARCHAR(30),
```

```
    protocollo VARCHAR(30) NOT NULL,  
    PRIMARY KEY(Servico_idServico)  
);  
  
CREATE TABLE Comando (  
    Servico_idServico BIGINT NOT NULL,  
    comando VARCHAR(100) NOT NULL,  
    PRIMARY KEY(Servico_idServico)  
);
```

Apêndice C - Interfaces de Navegação do SMS

Este apêndice apresenta as telas de navegação para executar algumas das funcionalidades disponibilizadas pelo SMS. Também é ilustrado um relatório de um cluster de serviço (DNS) que utiliza os dados (consolidados em resumos diários) de mais de um mês de monitoramento.

Inicializando o SMS

Na inicialização do SMS, o administrador da rede deve especificar o SLA para o IDS do ambiente monitorado e definir a periodicidade para consolidação dos dados de monitoramento, ou seja, consolidação dos dados em resumos diários, Figura 39(a). Após especificar tais dados, chega-se a tela principal do SMS, Figura 39(b).



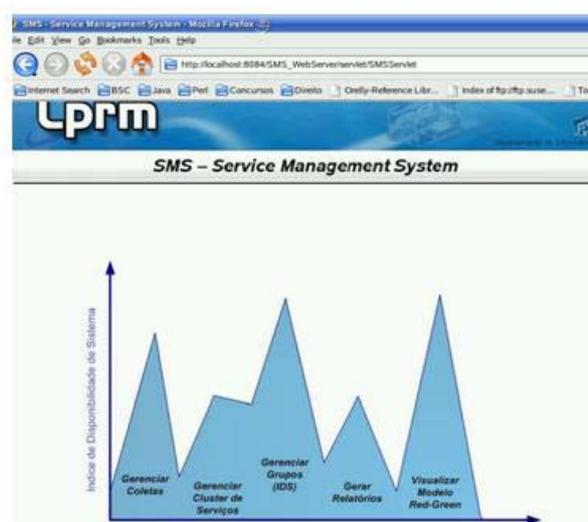
The screenshot shows a web browser window with the URL `http://localhost:8084/SMS_WebServer/index.html`. The page features the Lprm logo and the title "SMS - Service Management System". Below the header, there is a form titled "SMS" with the following content:

Especifique o SLA do IDS do Sistema de Gerência de Serviços
IDS (ex: 0.99999):

Selecione o tipo de coleta

- Diário
- Semanal
- Quinzenal
- Mensal

(a)



(b)

Figura 39 - Inicializando o SMS.

Criar Coleta de Serviços

A partir da tela principal do SMS (Figura 39-b), através da opção *Gerenciar Coletas*, pode-se criar uma nova coleta, ou seja, instanciar o monitoramento de um novo serviço. Na tela Figura 40(a) escolher-se a ação, no caso *Criar Coleta*, e o tipo do serviço a ser monitorado. Na tela Figura 40(b) o administrador da rede informa ao sistema algumas informações necessárias para o monitoramento do serviço. Há informações específicas para cada serviço, assim como informações que devem ser informadas para qualquer tipo de serviço, ou seja, informações gerais. Por exemplo, um serviço WWW tem como informação específica a URL a ser monitorada. Exemplos de informações gerais seriam o IP do servidor provedor do serviço, o SLA para o tempo de resposta e para a disponibilidade do serviço.

Uma mensagem de sucesso é exibida (Figura 40-c) quando uma nova coleta é criada.

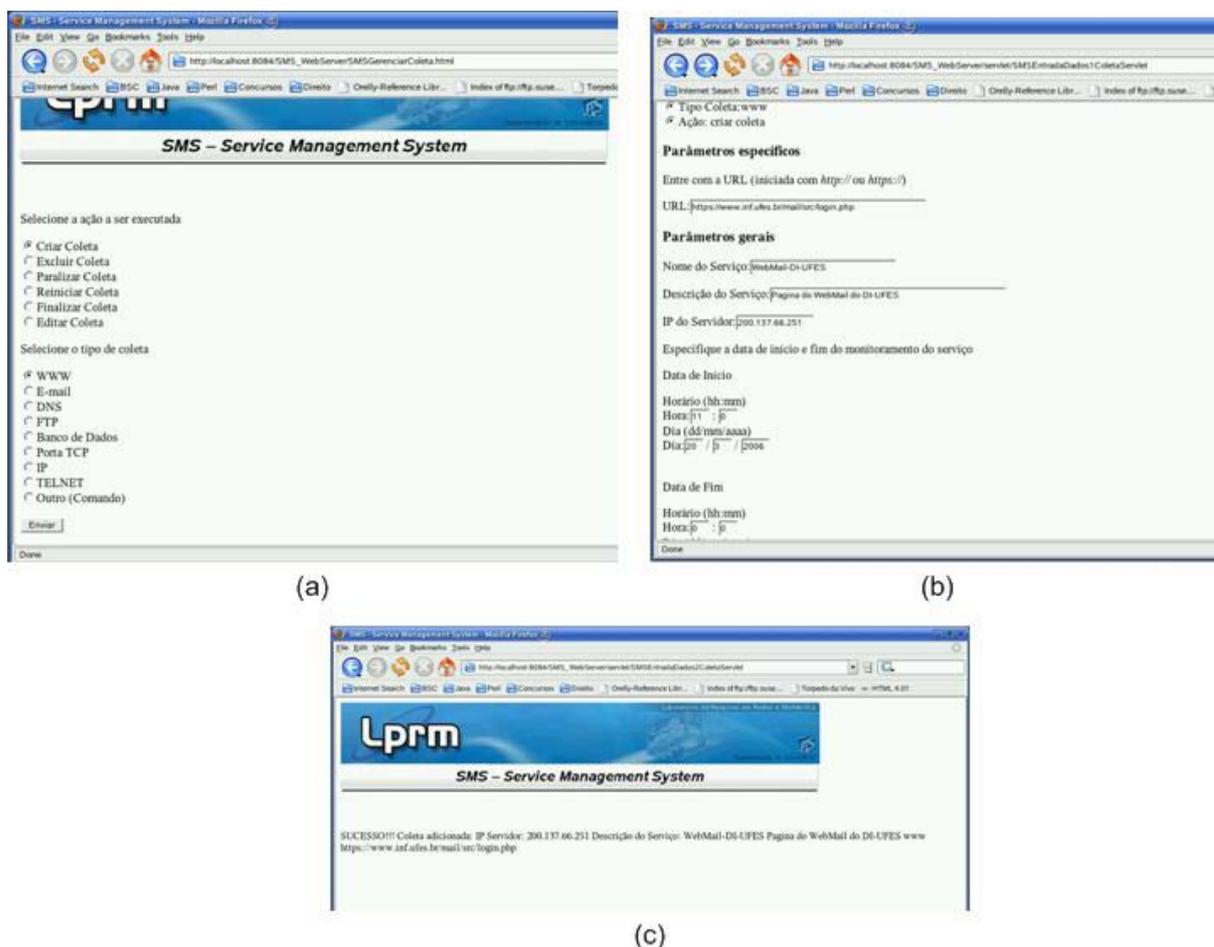


Figura 40 - Criando uma coleta.

Criar Cluster de Serviços

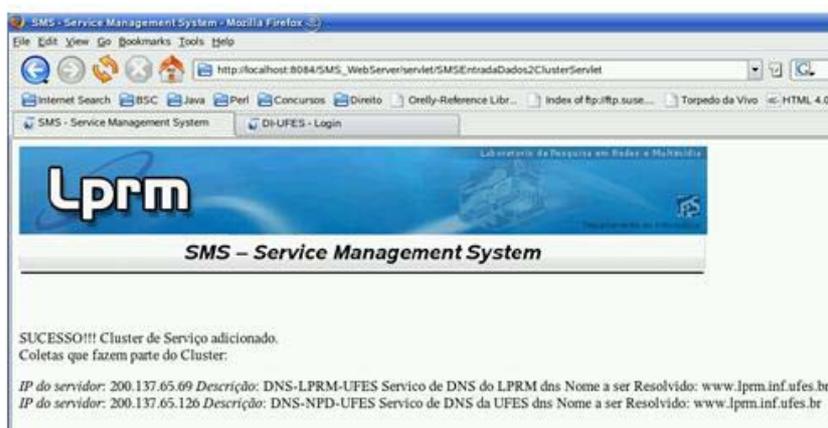
A partir da tela principal do SMS (Figura 39-b), através da opção *Gerenciar Cluster de Serviço*, pode-se criar um novo cluster, ou seja, instanciar o monitoramento da disponibilidade de um novo cluster de serviço. Na tela Figura 41(a) escolhe-se a ação, no caso *Criar Cluster*, e o tipo de serviço do cluster. Na tela Figura 41(b) o administrador da rede informa ao sistema algumas informações necessárias para o monitoramento do cluster, como os serviços que irão compor o cluster e o SLA para a disponibilidade do mesmo. Uma mensagem de sucesso é exibida (Figura 41-c) quando um novo cluster é criado.



(a)



(b)



(c)

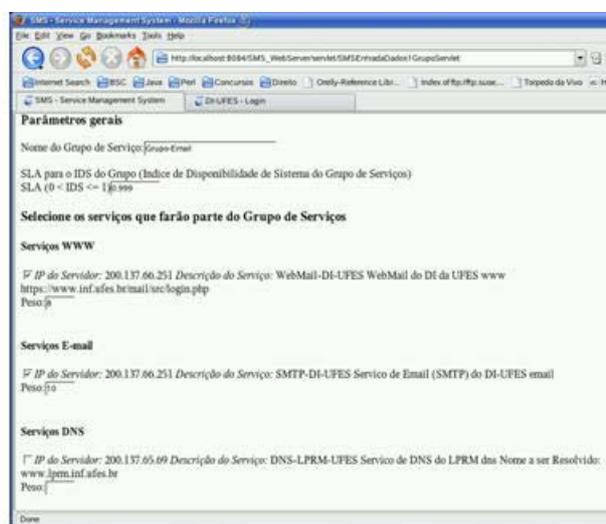
Figura 41 - Criando um cluster de serviços.

Criar Grupo de Serviços

A partir da tela principal do SMS (Figura 39-b), através da opção *Gerenciar Grupos (IDS)*, pode-se criar um novo grupo. Na tela Figura 42(a) escolhe-se a ação, no caso *Criar Grupo de Serviço*. Na tela Figura 42(b) o administrador da rede seleciona quais serviços e clusters de serviços monitorados pelo SMS farão parte do grupo a ser criado. Para cada item selecionado deve-se especificar também um peso. O SLA para a disponibilidade do grupo também deve ser definido. Uma mensagem de sucesso é exibida (Figura 42-c) quando um novo grupo é criado.



(a)



(b)



(c)

Figura 42 - Criando um grupo de serviços.

Gerar Relatório de Serviço



(b)

(c)

(d)



Figura 43 - Gerando um relatório de um serviço.

A partir da tela principal do SMS (Figura 39-b), através da opção *Gerar Relatórios*, pode-se gerar um relatório estatístico de uma coleta de serviço. Na tela Figura 43(a) escolhe-se a opção *Relatório de Coleta*. Na tela seguinte, Figura 43(b), seleciona-se o tipo da coleta de serviço. Na tela Figura 43(c) seleciona-se o serviço desejado. Na tela Figura 43(d) é oferecida a possibilidade de se aplicar filtro de datas e a utilização de dados consolidados (dados antigos) para a geração do relatório. Depois dessas definições o relatório é gerado, Figura 43(e).

Gerar Relatório de IDS

A partir da tela principal do SMS (Figura 39-b), através da opção *Gerar Relatórios*, pode-se gerar um relatório de IDS. Na tela Figura 43(a) escolhe-se a opção *Relatório de Grupos (IDS)*. Na tela seguinte, Figura 44(a), redefine-se o IDS do SMS (se for de desejo, pois já foi definido na inicialização do SMS, Figura 39-a). Na tela Figura 44(b) é oferecida a possibilidade de se aplicar filtro de datas e a utilização de dados consolidados para a geração do relatório. Depois dessas definições o relatório de IDS do SMS é gerado, Figura 43(e).

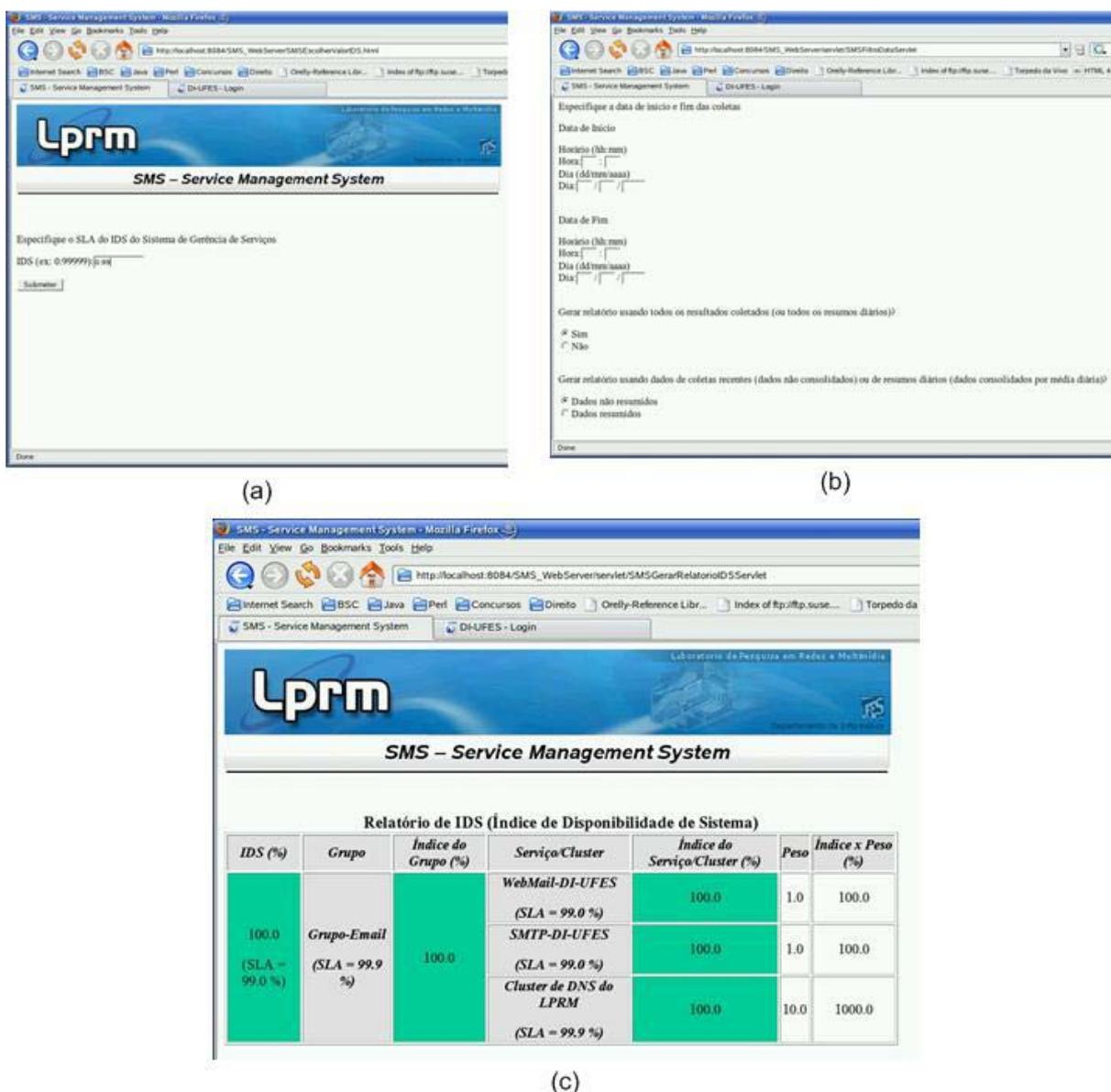
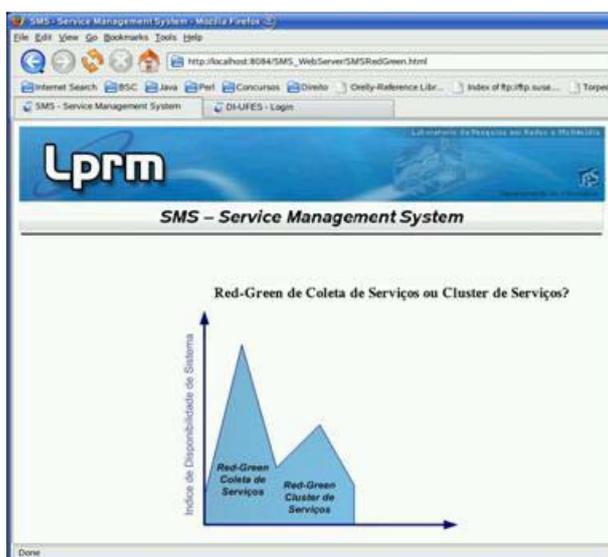


Figura 44 - Gerando um relatório de IDS.

Visualizar Modelo Red-Green de Serviço

A partir da tela principal do SMS (Figura 39-b), através da opção *Visualizar Modelo Red-Green*, pode-se visualizar o estado atual, o tempo de resposta atual e a disponibilidade média dos serviços e clusters monitorados pelo SMS (no caso de cluster de serviço, não há tempo de resposta). Na tela Figura 45(a) escolhe-se uma das opções: *Visualizar Red-Green de Coleta de Serviços ou Cluster de Serviços*. Escolhendo para *Coleta de Serviços*, na tela seguinte, Figura 45(b), é permitida a aplicação de alguns filtros para a visualização do modelo *Red-Green*. Depois dessas escolhas, o modelo *Red-Green* é apresentado, Figura 45(c).



(a)

Seleção o tipo de coleta

WWW
 E-mail
 DNS
 FTP
 Banco de Dados
 Porta TCP
 IP
 TELNET
 Outro (Comando)

Todos os tipos de coleta?
 Sim
 Não

Seleção a severidade para o Tempo de Resposta

FATAL
 CRITICAL
 WARNING
 HARMLESS

Todas as severidades para Tempo de Resposta?
 Sim
 Não

(b)

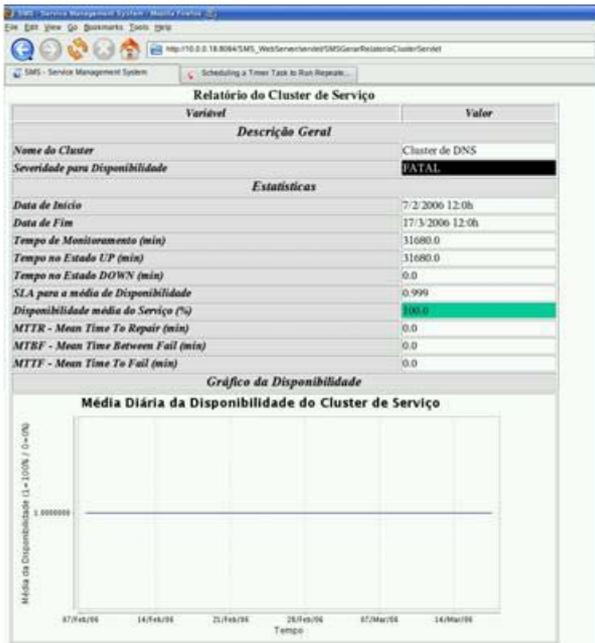
Status	Tempo de Resposta (seg)	IP do Servidor	Tipo do Serviço	Disponibilidade Média (%)	Descrição Geral	Estado do Monitoramento
UP	0.703	200.137.06.251	www	100.0	Nome do Serviço: WebMail-DI-UFES Descrição do Serviço: WebMail-DI-UFES WebMail do DI da UFES www https://www.inf.ufes.br/mail/src/login.php	Coletando
UP	0.0030	200.137.06.251	email	100.0	Nome do Serviço: SMTP-DI-UFES Descrição do Serviço: SMTP-DI-UFES Serviço de Email (SMTP) do DI-UFES email	Coletando
UP	0.027	200.137.05.09	dns	100.0	Nome do Serviço: DNS-LPRM-UFES Descrição do Serviço: DNS-LPRM-UFES Serviço de DNS do LPRM dns Nome a ser Resolvido: www.lprm.inf.ufes.br	Coletando
UP	0.027	200.137.05.126	dns	100.0	Nome do Serviço: DNS-NPD-UFES Descrição do Serviço: DNS-NPD-UFES Serviço de DNS da UFES dns Nome a ser Resolvido: www.lprm.inf.ufes.br	Coletando

(c)

Figura 45 - Visualizando o modelo *Red-Green*.

Relatório de um Cluster de Serviço de DNS que Utiliza Dados Consolidados

A Figura 46 ilustra um relatório de um cluster de serviço de DNS (o mesmo do estudo de caso apresentado no capítulo 5) que utiliza dados de monitoramento de mais de um mês de coleta (07/02/2006 a 01/03/2006). Os dados utilizados para geração de tal relatório são dados consolidados, ou seja, resumos diários. Pode-se observar que o relatório de um cluster é formado pelo relatório de disponibilidade do cluster e pelos relatórios dos serviços que compõem o cluster. Neste exemplo é possível visualizar que o cluster de DNS teve uma disponibilidade de 100%. No entanto, um dos serviços de DNS (NPD-UFES) teve uma indisponibilidade de aproximadamente 5 minutos (um intervalo de *pooling*).



Veja abaixo os relatórios das Coletas de Serviços que compõem o Cluster

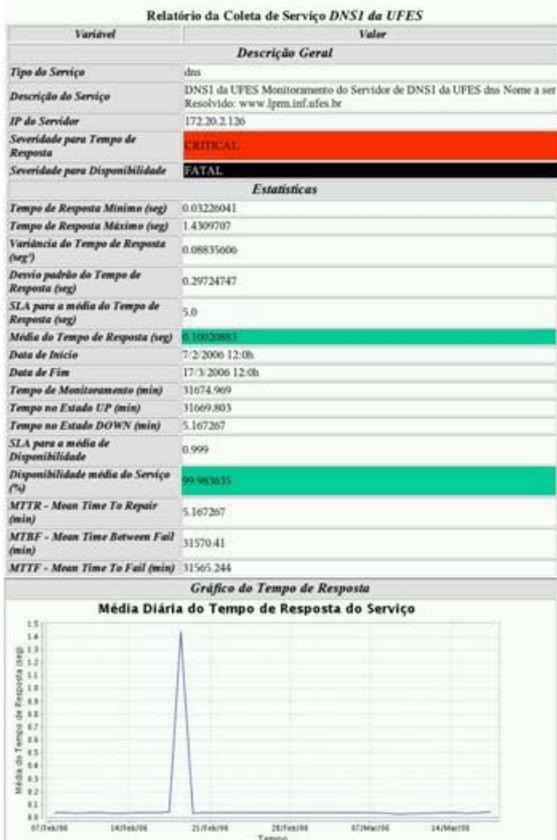


Figura 46 - Relatório de um Cluster de Serviço de DNS.