

UNIVERSIDADE ESTADUAL PAULISTA
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**MONITORAMENTO DE *HOSTS* EM REDES
TCP/IP SOBRE BASE GEORREFERENCIADA
COM MÉTRICAS DE QUALIDADE**

Daniel Saverio Spozito

Orientador: Prof. Dr. Nobuo Oki

Ilha Solteira - SP, maio de 2011.



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Campus de Ilha Solteira

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

“MONITORAMENTO DE *HOSTS* EM REDES TCP/IP SOBRE
BASE GEORREFERENCIADA COM MÉTRICAS DE
QUALIDADE”

Daniel Saverio Spozito

Orientador: Prof. Dr. Nobuo Oki

Dissertação apresentada à Faculdade de Engenharia
– UNESP - Campus de Ilha Solteira, como
requisito parcial para obtenção do título de Mestre
em Engenharia Elétrica.

Área de conhecimento: Automação.

Ilha Solteira - SP, maio de 2011.

FICHA CATALOGRÁFICA

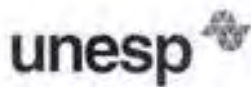
Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação
Serviço Técnico de Biblioteca e Documentação da UNESP - Ilha Solteira.

S765m Spozito, Daniel Saverio.
Monitoramento de *hosts* em redes tcp/ip sobre base georreferenciada com métricas de qualidade / Daniel Saverio Spozito. -- Ilha Solteira : [s.n.], 2011
147 f. : il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de Conhecimento: Automação, 2011

Orientador: Nobuo Oki
Inclui bibliografia

1. Redes de computadores. 2. TCP/IP (Protocolo de rede de computação).
3. Geoprocessamento. 4. Monitoramento.



UNIVERSIDADE ESTADUAL PAULISTA
CAMPUS DE ILHA SOLTEIRA
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA

CERTIFICADO DE APROVAÇÃO

TÍTULO: Monitoramento de Hosts em redes TCP/IP sobre base georreferenciada com métricas de qualidade

AUTOR: DANIEL SAVERIO SPOZITO

ORIENTADOR: Prof. Dr. NOBUO OKI

Aprovado como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica ,
Área: AUTOMAÇÃO, pela Comissão Examinadora:

Prof. Dr. NOBUO OKI

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. ARTUR PANTOJA MARQUES

Departamento de Engenharia Civil / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. DILVAN DE ABREU MOREIRA

Departamento de Ciências Matemáticas e de Computação - USP - São Carlos/SP

Data da realização: 27 de maio de 2011.

*Dedico esse trabalho à Sonia Regina Correa Fernandes
referência ímpar no meu processo de aprendizado.*

Agradecimentos

Primeiramente agradeço a Deus, pela oportunidade da vida e encaminhamento de minha trajetória de modo a ter oportunidade de desenvolvimento intelectual tão preciosa.

Ao meu orientador, o professor Nobuo Oki, que aceitou o desafio de tema recente e desafiador e considerando que nada é por acaso, já é possível observar futuras oportunidades e desafios, frutos deste trabalho. Agradeço também a compreensão e a paciência do mesmo nos momentos em que precisava alternar meu foco entre os estudos e trabalho, bem como alguns momentos em que me ajudou a refletir sobre críticas e sugestões sobre este trabalho, sempre de forma ponderada.

À minha companheira, Sonia Regina Corrêa Fernandes, pelo carinho, compreensão, apoio e pelo exemplo de fé.

Aos meus pais, José Saverio Spozito Junior, referência de conhecimento e Soeli Aparecida Ferreira Spozito, exemplo de dedicação e amor para com os filhos e neta.

Aos meus enteados Taís Fernandes Guerra e Tiago Fernandes Guerra, pela escolha do caminho correto, como estudo e trabalho, possibilitando a harmonia e equilíbrio familiar.

Ao meu irmão mais novo, Raphael Saverio Spozito, pelo suporte ao meu núcleo familiar primário, nos momentos que precisei me ausentar. Ao meu irmão do meio, Fábio Saverio Spozito, pelos momentos de luta interior oferecendo o que tem de melhor a todos. À minha sobrinha Maria Eduarda Renzi Spozito, que desde cedo demonstra características de pessoa comprometida, esforçada e respeitosa.

Aos meus tios Eliseu Savério Sposito e Maria Encarnação Beltrão Sposito, referências familiares na academia, pela revisão e orientação.

À Milena Duarte Mostado, pela revisão ortográfica e oportunidade de trabalho conjunto, possibilitando ampliar a abrangência do trabalho nas unidades do Centro Paula Souza.

Aos professores do Departamento de Engenharia Elétrica, pelos ensinamentos nas disciplinas cursadas e orientação.

EPÍGRAFE

"Inteligência consiste de 1% de inspiração e 99% de Transpiração!"

Thomas Alva Edison

Resumo

Com a globalização e a evolução tecnológica, a capilaridade das instituições, sejam elas públicas ou privadas, é realidade irreversível. O uso de computadores cada vez mais velozes e com elevada capacidade de armazenamento possibilitou a centralização das informações. Nesse contexto, surge a necessidade de interligar os sistemas computacionais com o objetivo de trocar informações, independentemente da região geográfica. A Internet, utilizando tecnologias de redes de dados, com arquitetura projetada para suportar em nível mundial a comunicação entre computadores, passa a ser a principal ferramenta utilizada na interligação de locais geograficamente distribuídos. Com a tecnologia e infraestrutura disponível para efetivar a comunicação, surge também a necessidade de garantir que os sistemas estejam ativos o maior tempo possível, atendendo a desempenho compatível às atividades desenvolvidas, pois as instituições passam a depender da interação entre os sistemas computacionais, independente do ponto do planeta em que estejam. Este trabalho propõe modelar, desenvolver e realizar ensaios de uma ferramenta que auxilie no monitoramento de pontos interligados por redes de comunicação dados sobre protocolo TCP/IP distribuídos geograficamente. O modelo consiste em monitorar um ou mais pontos, visualizados sobre um mapa com o traçado do perímetro político geográfico da área atendida. Em suma, propõe-se utilizar as ferramentas básicas de teste de conectividade nas redes de comunicação de dados, associadas aos conceitos de georreferenciamento e geoprocessamento.

Palavras-chave: Redes de Computadores. TCP/IP. Geoprocessamento. Monitoramento.

Abstract

The globalization and technological evolution, allowed the grow of institutions public or private over the entire world with highest capillarity. The development of computers becoming faster and with large storage capacity allowed the centralization of information. In this context the necessity of linking computer systems with the aim to exchange information regardless of geography became a reality. The technics and technologies used in Internet, involving data communications, with architecture designed to support global communication between computers, becomes the main tool used in the interconnection of thousand geographically distributed sites. With the technology and infrastructure available for an effective communication, also arises the need to ensure that systems are active as long as possible, with a minimum of performance to the services, because the activities in the institutions are dependent upon interaction between computer systems, regardless of the point on the planet they are. This study proposes to model, develop and test a tool to assist in the monitoring of points connected by data communication networks over TCP / IP distributed geographically. The proposed format is to monitor one or more points, displayed on one or more maps with the route of political geographic boundaries of the area served. In short, it is proposed to use the basic tools to test network connectivity in data communications, combined with the concepts of GIS and georeferencing.

Keywords: Computer Networks. TCP/IP. GIS. Monitoring.

Lista de Figuras:

Figura 1 – Rede de abrangência pessoal (PAN).....	25
Figura 2 – Rede local (LAN).....	25
Figura 3 – Rede Metropolitana (MAN).....	26
Figura 4 – Rede de longa distância (WAN).	26
Figura 5 – Topologia Estrela.	28
Figura 6 - Tecnologias <i>wireless</i>	30
Figura 7 – Modelo OSI e TCP/IP.....	36
Figura 8 – Representação de um endereço IPv4.....	37
Figura 9 – Divisão das redes lógicas em classes.	38
Figura 10 – Rede Classe A.	39
Figura 11 – Rede Classe B.	39
Figura 12 – Rede Classe C.	39
Figura 13 – Exemplo de endereçamento IP classe C.	40
Figura 14 – Resultado da operação AND booleano.	40
Figura 15 – Disparo de um comando PING para wikipedia.com.....	44
Figura 16 – Mapa de Londres e registros de casos de cólera.	46
Figura 17 – Mapa de São José do Rio Preto no formato vetorial.....	50
Figura 18– Mapa de São José do Rio Preto no formato <i>raster</i> (detalhes vetoriais sobrepostos)..	51
Figura 19 – Diagrama ER simples de um cadastro imobiliário.....	52
Figura 20 – Exemplo de monitoramento de rede WAN.....	55
Figura 21 – Estrutura da rede Intragov.....	59
Figura 22– <i>Link</i> Intragov Estado SP.....	60
Figura 23– Portabilidade da linguagem Java.....	64
Figura 24– Ícone representativo da linguagem Java.	65
Figura 25– Simbologia universal da notação UML.....	65
Figura 26– Tela de abertura da IDE NetBeans.....	66
Figura 27– Logotipo do SGBD MySQL.	67
Figura 28 – Logotipo MySQL Workbench.	68
Figura 29 – Logotipo da biblioteca GeoTools (<i>Open Source Geospatial Foundation</i>).	68
Figura 30– IDE e código fonte Java utilizando biblioteca geotools.....	69
Figura 31– <i>Software</i> com a plotagem da malha urbana de Ilha Solteira (ZOOM=5).	70
Figura 32– <i>Software</i> com a plotagem da malha urbana de Ilha Solteira (ZOOM=10).	70
Figura 33– <i>Software</i> com a plotagem da malha urbana de Ilha Solteira (ZOOM=30).	71
Figura 34 – Visão geral do banco de dados.....	73
Figura 35 – Tabela de geo-referenciamento das edificações.....	74
Figura 36 – Tabelas previstas na caracterização do <i>link</i> de dados.	75
Figura 37 – Tabela Acessos.....	77
Figura 38 – Tabelas bloco <i>hosts</i>	78
Figura 39 - Tabela Mapas.....	79
Figura 40 – Bloco Operadores e Ocorrências.....	80
Figura 41 - Tela Principal do Sistema GeoMonIpv4.....	81
Figura 42 - Tabelas geográficas.....	82
Figura 43 - Tabela Estados.....	83

Figura 44 - Tabela Cidades.....	83
Figura 45 - Tabela Bairros.....	84
Figura 46 - Tabela Ruas (classificação cidades).	85
Figura 47 - Tabela Ruas (classificação bairros).	85
Figura 48 - Cadastro de Ruas.	86
Figura 49 - Cadastro de <i>sites</i>	87
Figura 50 – Cadastro equipamentos.	88
Figura 51 - Cadastro de <i>hosts</i>	88
Figura 52 - Módulo monitoramento.	89
Figura 53 - Monitoramento de <i>hosts</i> (Intragov).	90
Figura 54 - Monitramento de <i>hosts</i> (Internet).	90
Figura 55 - Logs de monitoramento.	91
Figura 56 - Visualizar mapa.	92
Figura 57 - Mapas cadastrados no sistema.	92
Figura 58 - Visualização Multimapas.....	93
Figura 59 - Mapa do Estado de São Paulo.....	93
Figura 60 - Tabelas redes.....	94
Figura 61 - Tabelas pessoas.....	94
Figura 62 - Pacotes do código fonte.	95
Figura 63 - pacote geomon.	96
Figura 64 - Pacote geomon.Gis.	97
Figura 65 - Pacote geomon.Gui.	97
Figura 66 - Classe geomom.dao.	98
Figura 67 - Recursos SGBD.	99
Figura 68 - Pacote geomon.model.	99
Figura 69 - Pacote monitor.	100
Figura 70 - Registro de monitoramento.....	102
Figura 71 - Tabela Logs (primeira simulação).	103
Figura 72- Carga do sistema.	103
Figura 73 - Mapa relativo ao status dos <i>hosts</i> na tabela Logs (primeira simulação).....	104
Figura 74 - Tabela Logs (segunda simulação).....	105
Figura 75 - Mapa relativo ao status dos <i>hosts</i> na tabela Logs (segunda simulação).	105
Figura 76 - Tabela Logs (terceira simulação).....	106
Figura 77 - Mapa relativo ao status dos <i>hosts</i> na tabela Logs (terceira simulação).	106
Figura 78 - Tabela Logs (quarta simulação).....	107
Figura 79 - Mapa relativo ao status dos <i>hosts</i> na tabela Logs (quarta simulação).	108
Figura 80 - Monitoramento na rede Internet.	109
Figura 81 - Monitoramento na rede Internet.	109

Lista de Tabelas:

Tabela 1 - Unidades da federação brasileira com maior e menor grau de inclusão digital.	33
Tabela 2 - Limites para cálculo dos parâmetros de qualidade.....	101
Tabela 3 – Caracteres de controle da padronização ASCII.....	116
Tabela 4 – Caracteres Imprimíveis da Tabela ASCII.....	117

Lista de Símbolos e Abreviaturas:

ADSL	<i>Assymmetric Digital Subscriber Line</i>
API	<i>Application Programming Interface</i>
ASCII	<i>American Standard Code for Information Interchange</i>
CE	<i>Community Edition</i>
CEETEPS	Centro Estadual de Educação Tecnológica Paula Souza
CGI.br	Comitê Gestor da Internet no Brasil
CSMA/CD	<i>Carrier Sense Multiple Access With Collision Detection</i>
DAT	<i>Digital Audio Storage</i>
DDS	<i>Digital Data Storage</i>
DHU	Departamento de Habitação e Urbanismo
DNS	<i>Domain Name System</i>
DoD	<i>Department of Defense</i>
EDGE	<i>Enhanced Data rates for GSM Evolution</i>
EIA	<i>Electronic Industries Alliance</i>
ER	Entidade Relacional
FDDI	<i>Fiber Distributed Data Interface</i>
FGV	Fundação Getulio Vargas
FTP	<i>File Transfer Protocol</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile Communications</i>
HD	<i>Hard Disk</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
ICMP	<i>Internet Control Message Protocol</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IMPs	<i>Interface Message Processors</i>
IP	<i>Internet Protocol</i>
IPv4	<i>Internet Protocol version 4</i>

IPv6	<i>Internet Protocol version 6</i>
ISA	<i>Industry Standard Architecture</i>
ISDN	<i>Integrated Services Digital Network</i>
ISO	<i>International Organization for Standardization</i>
KB	Kilo Bytes
LAN	<i>Local Area Network</i>
MAN	<i>Metropolitan Area Network</i>
MB	Mega Bytes
MPLS	<i>Multiprotocol Label Switching</i>
MRTG	<i>Multi Router Traffic Grapher</i>
MVC	<i>Model-View-Controller</i>
NASA	<i>National Aeronautics and Space Administration</i>
NIC	<i>Network Interface Card</i>
OMT-G	<i>Object Modeling Technique for Geographic Applications</i>
OO	Orientado a Objetos
PAN	<i>Personal Area Network</i>
PC	<i>Personal Computer</i>
PCI	<i>Peripheral Component Interconnect</i>
PING	<i>Packet Internet Grouper</i>
PRODESP	Companhia de Processamento de Dados do Estado de São Paulo
RAM	<i>Random Access Memory</i>
RF	Radio Frequencia
SAN	<i>Storage Area Network</i>
SGBG	Sistema de Gerenciamento de Banco de Dados
SIG	Sistema de informação geográfica
SLA	<i>Service Level Agreement</i>
Slackware	Distribuição Linux, versão de sistema operacional baseada em conceitos UNIX.
SMTP	<i>Simple Mail Transfer Protocol</i>
SNMP	<i>Simple Network Management Protocol</i>
SO	Sistema Operacional
SONET	<i>Synchronous Optical Network</i>

SQL	<i>Structured Query Language</i>
SRI	<i>Stanford Research Institute</i>
STPC	Serviço de telefonia pública comutada
TCP	<i>Transmission Control Protocol</i>
TCP / IP	<i>Transmission Control Protocol / Internet Protocol</i>
TCU	Tribunal de Contas da União
TFTP	<i>Trivial File Transfer Protocol</i>
TIA	<i>Telecommunications Industry Association</i>
TIC	Tecnologias da Informação e Comunicação
UDP	<i>User Datagram Protocol</i>
UML	<i>Unified Modeling Language</i>
Unix	Sistema Operacional Portátil Multitarefa
USB	<i>Universal Serial Bus</i>
UWB	<i>Ultra-wideband</i>
VPN	<i>Virtual Private Network</i>
WAN	<i>Wide Area Network</i>

SUMÁRIO

1	Introdução	17
1.1	Trabalhos relacionados	18
1.2	Motivação	19
1.3	Objetivos do trabalho	20
1.4	Organização do texto	20
2	Redes TCP/IP	22
2.1	Introdução às redes	22
2.1.1	Protocolos de camada física	23
2.1.2	Protocolos de conexão lógica	24
2.2	Conceitos básicos de redes	24
2.3	Meios físicos para redes	27
2.4	Redes WAN e largura de banda	31
2.5	Modelo TCP/IP	35
2.6	Conjunto de protocolos TCP/IP e endereçamento IP	37
2.7	Qualidade em redes TCP/IP	42
2.8	PING	43
3	Sistemas Georreferenciados	45
3.1	Conceitos básicos	45
3.2	Sistemas de Informação Geográfica - SIG	46
3.3	Generalização cartográfica	48
3.3.1	Ponto	48
3.3.2	Linha	48
3.3.3	Polígono	48
3.4	Dados geográficos	49
3.5	Representação vetorial/matricial	49
3.6	Modelagem de dados	51
4	Monitoramento TCP/IP georreferenciado	54
4.1	Gerência em redes IP	54
4.2	Georreferenciamento de equipamentos de rede	56
5	Estudo de caso.	58
5.1	O Centro Estadual de Educação Tecnológica Paula Souza - CEETEPS	58
5.2	Problemas e Hipóteses	60
5.3	Tipo de pesquisa	61
5.4	Materiais	61
5.5	Metodologia proposta	61
5.6	Descrição das ferramentas	61
5.6.1	Sistemas Operacionais	62
5.6.2	A Linguagem Java.	63
5.6.3	A IDE NetBeans	65
5.6.4	O SGBD MySQL	66
5.6.5	A biblioteca GeoTools	68
6	Sistema GeoMon_Ipv4	72
6.1	Modelagem do banco de dados	72
6.1.1	Bloco de georreferenciamento	74

6.1.2	Bloco de caracterização de tecnologias e topologias das redes TCP/IP.....	75
6.1.3	Bloco de equipamentos de rede (<i>hosts</i>) e histórico de <i>status</i> (logs);.....	77
6.1.4	Bloco de operadores e ocorrências.	79
6.2	Modelagem do sistema e codificação Java.....	80
6.2.1	Bloco tabelas geográficas.	82
6.2.2	Bloco tabelas equipamentos.	87
6.2.3	Bloco monitoramento.	89
6.2.4	Bloco principal.	92
6.2.5	Blocos auxiliares.	94
6.3	Organização do código fonte.....	94
6.3.1	Pacote geomon.....	96
6.3.2	Pacote geomon.Gis	96
6.3.3	Pacote geomon.Gui.....	97
6.3.4	Pacote geomon.dao	98
6.3.5	Pacote geomon.model.....	99
6.3.6	Pacote geomon.monitor	100
6.3.7	Coleta das métricas	101
6.4	Testes do sistema GeoMonIPv4	101
	CONCLUSÕES.....	110
	REFERÊNCIAS	112
	ANEXOS.....	116
	Anexo I.....	116
	APÊNDICES	118
	Apêndice I	118
	Apêndice II.....	122
	Apêndice III.....	123
	Apêndice IV	129
	Apêndice V.....	138
	Apêndice VI	141
	Apêndice VII.....	143

CAPÍTULO I

1 Introdução

Na sociedade atual, é possível observar a importância da comunicação entre equipamentos eletrônicos. A Internet é a maior estrutura planejada e montada para troca de informações por meio eletrônico. Alguns setores considerados estratégicos para vários países como, por exemplo, a área financeira e o sistema bancário, dependem atualmente da Internet e das redes de computadores para operar.

Com a constante migração das informações do papel para o meio eletrônico, os microcomputadores ganharam importância nas atividades relacionadas a todas as áreas. Surge então a necessidade do intercâmbio de informações seja dentro ou entre as instituições da mesma atividade econômica ou em atividades diferentes, como o acesso ao sistema bancário pelas indústrias e pelo comércio.

Garantir a comunicação e troca de dados entre os milhares de pontos passa a ser atividade fundamental nesse contexto.

A Internet, apesar de projetada para operar sem depender de apenas um único ponto, não oferece garantias de estabilidade e segurança. Utilizar ferramentas de gestão da comunicação entre pontos distribuídos geograficamente é atividade que pode impactar no desempenho dos serviços oferecidos ou até mesmo na disponibilidade dos mesmos.

Atualmente, dada a dependência das empresas em se tratando da necessidade de troca de informações entre os sistemas informáticos, o sucesso das mesmas passa a depender também das soluções tecnológicas e das redes de comunicação de dados (GARCEZ, 2007).

Em médias e grandes instituições, surge o papel do gestor de redes, profissional designado para garantir a comunicação entre os diferentes pontos, objetivando planejar, implementar e dar suporte de modo eficaz e eficiente nas redes de comunicação de dados.

Considerando a capilaridade de algumas instituições, sendo que há situações em que os locais interligados distam centenas de quilômetros, ferramentas computacionais passam a exercer papel fundamental nas atividades de suporte.

Associar conceitos geográficos com a utilização de mapas, dada a facilidade que o ambiente gráfico proporciona na interpretação de informações, pode reduzir o tempo na detecção de falhas e por consequência, diminuir o período de inatividade na comunicação entre pontos geograficamente distribuídos.

1.1 Trabalhos relacionados

Projetos tradicionais de redes de computadores normalmente são baseados em normas, recomendações e melhores práticas. (SANTANA, 2006).

Há no mercado sistemas comerciais e/ou gratuitos que objetivam monitorar as redes lógicas, auxiliando os gestores de redes de comunicação de dados desenvolverem suas atividades; dentre eles, pode-se destacar:

- *Network View* (NETWORKVIEW, 2011)
- *ManageEngine OPManager* (OPMANAGER, 2011)
- Nagios (NAGIOS, 2011)
- MRTG (*The Multi Router Traffic Grapher*) (MRTG, 2011)

Apesar de sistemas como o *Network View* possibilitarem a rápida instalação em ambientes gráficos sobre a plataforma Windows, os dispositivos monitorados são dispostos como figuras geométricas sem relacionamento geográfico. Utilizar aplicativos como o Nagios, MRTG ou *Network View* em redes pequenas, com dispositivos interligados dentro de uma mesma edificação ou a pequenas distâncias não torna relevante a necessidade de soluções de monitoramento geoprocessado, além de considerar que a esses softwares já estão em estágio maduro de desenvolvimento.

A grande maioria dos projetos desenvolvidos na área utiliza, além das métricas básicas, a opção de obter informações dos pontos monitorados utilizando protocolos desenvolvidos para esse fim, como o SNMP.

O MRTG é um sistema de distribuição gratuito, amplamente utilizado na comunidade de usuários dos sistemas operacionais LINUX. Baseia-se na coleta de informações utilizando protocolo SNMP e utiliza ferramentas adicionais para gerar gráficos das informações coletadas.

Dentre os *softwares* comerciais, o sistema *OPManager* destacou-se na utilização de mapas a partir de análises feitas no sítio na Internet do desenvolvedor. Trata-se de um sistema comercial com valores a partir de US\$ 595,00, conforme a versão escolhida. Não há referência quanto ao tipo de base cartográfica utilizada, se há georreferenciamento ou apenas a utilização de figuras dentro do sistema.

Dentro da comunidade acadêmica, destaca-se trabalho recente (ZARPELÃO, 2010), que utiliza técnicas para análise comportamental em redes de comunicação de dados, mas cabe ao operador identificar em qual localidade ocorre a anomalia.

Outro trabalho (FERREIRA, 2005), considerou análises dentro de um estudo de caso na qual as ferramentas retornam resultados em arquivos na forma textual ou geralmente chamados de arquivos de *log*. Também conforme Ferreira (2005) foram visitadas três empresas, sendo duas derivadas da privatização do sistema Telebrás, nas quais há carência de ferramentas básicas que se adequem ao perfil de suas redes.

Deste modo, estudos anteriores refletem a pouca aplicação de softwares que se utilizem de recursos gráficos de alta definição, associados a bases cartográficas, possibilitando uma interpretação rápida e precisa pelos gestores das redes. Considerando também o porte das empresas pesquisadas por Ferreira (2005), pode-se dizer que a associação desses conceitos é recente, possibilitado principalmente pelo desenvolvimento e estudo das aplicações e tecnologias de geoprocessamento.

Uma aplicação prática na utilização de sistemas geográficos para auxiliar no monitoramento de redes para fornecimento de produtos ou serviços pode ser observada no estudo de (BHOGENAHALLY, 2009). O trabalho utiliza uma base de dados geográfica com informações de uma rede de dutos em uma área do Texas, utilizando o sistema de informações geográficas ArcGis.

1.2 Motivação

Durante a análise das ferramentas e com a necessidade de uma solução que realizasse o monitoramento de equipamentos de rede distribuídos geograficamente foi observada a carência de soluções que adotassem em conjunto sistemas de geoprocessamento e sistemas de monitoramento de redes.

Como exemplo, a iniciativa de alguns colaboradores da Prefeitura de Belo Horizonte (NASSIF et al., 2008) gerou a publicação do artigo sobre um sistema que, a partir de um conjunto de soluções distintas que envolvem desde a coleta de informações de campo até a importação de informações de arquivos, possibilitou a visualização em um terceiro sistema, o MapInfo, uma visão geográfica da rede de comunicação de dados de fibra ótica do município.

A motivação para este trabalho, além da necessidade de resolver uma solução pontual, foi destacar a carência de soluções para monitoramento de redes de comunicação de dados, utilizando, como base, sistemas de geoprocessamento.

Destacam-se, também, as vantagens de associar conceitos de geoprocessamento e georreferenciamento a sistemas e variáveis que possuem comportamento irregular em função do tempo.

Como principal contribuição, considerou-se o desenvolvimento da ferramenta sem a dependência de sistemas de terceiros no que tange, especificamente, o sistema de informações geográficas – SIG. Neste trabalho, ao desenvolver a base de dados, registros relacionados a redes de comunicação de dados foram utilizados, mas é possível alterar a estrutura para inserção de novos componentes ou atributos.

1.3 Objetivos do trabalho

Este trabalho tem como propósito desenvolver um sistema para o monitoramento de redes de comunicação de dados distribuídas.

Foram estudados conceitos de geoprocessamento e transmissão de dados em redes comutadas por pacotes que utilizam equipamentos geograficamente dispersos sobre os protocolos TCP/IP (*Transmission Control Protocol / Internet Protocol*) versão quatro (IPv4). Finalmente foi criada uma metodologia que possibilitou utilizar ferramentas de georreferenciamento no monitoramento dessas redes. Deste modo, espera-se justificar o desenvolvimento do protótipo com funcionalidades básicas, utilizado no monitoramento dos *links* de dados que interligam as unidades escolares do Centro Paula Souza.

1.4 Organização do texto

O texto foi organizado em sete capítulos, incluída a introdução apresentada no Capítulo I. No capítulo II descreve-se o funcionamento das redes de computadores utilizando o protocolo

TCP/IP. No Capítulo III descreve-se o conceito de sistemas georreferenciados. No Capítulo IV é descrita a metodologia aplicada bem como a utilização dos conceitos de georreferenciamento no monitoramento das redes que utilizam protocolo TCP/IP. O capítulo V apresenta a proposta e o local do estudo de caso na aplicação dos conceitos. No capítulo VI é demonstrada a organização do *software* desenvolvido durante o trabalho e por fim as conclusões.

CAPÍTULO II

2 Redes TCP/IP

Neste capítulo será apresentado o conceito de redes TCP/IP, principal padrão utilizado na implementação de redes de computadores sejam elas de abrangência local, regional ou mundial, como a Internet.

2.1 Introdução às redes

Um referencial importante no conceito de redes de computadores e o papel dos microcomputadores no ambiente de rede é a Internet.

A Internet é a maior rede de computadores do mundo e consiste basicamente em várias redes interconectadas, sejam elas pequenas, médias ou grandes.

É possível dividir os parâmetros que devem ser atendidos para que uma conexão à Internet seja satisfeita: conexão física, conexão lógica e aplicações.

A conexão física é realizada por um dispositivo eletrônico conectado ao microcomputador. Esse dispositivo pode ser um Modem (Modulador-Demodulador), uma placa de rede com ou sem fio (*wireless*), um dispositivo *Bluetooth* ou até mesmo um cabo padrão USB (*Universal Serial Bus*).

Atualmente, o principal dispositivo utilizado para conectar um microcomputador a uma rede é a placa de rede (NIC), ou adaptador de rede. Via de regra, todos os computadores hoje saem de fábrica dotados deste dispositivo. Para escolha do tipo de placa de rede que será utilizada em um determinado sistema, devem-se considerar os seguintes fatores:

- Protocolos de camada física compatíveis com o dispositivo: *Ethernet*, *Token Ring*, *FDDI*;
- Meio físico de comunicação: Cabeado (*wired*), Sem fio (*wireless*) ou fibra óptica;
- Tipo de barramento do dispositivo: PCI, USB ou ISA (Compatível com o microcomputador)

Após definido o meio físico utilizado para troca de informações, o formato com que isso ocorre deve também ser definido. A isso dá-se o nome de protocolo.

Um protocolo é a descrição formal do modo com que as comunicações são feitas entre os dispositivos. Uma analogia simples é a linguagem escrita, que possui regras gramaticais e um vocabulário próprio possibilitando que uma mensagem escrita por uma pessoa seja compreendida por outra, desde que ambas conheçam e entendam o funcionamento dessas regras ou protocolos.

2.1.1 Protocolos de camada física

Há uma estreita relação entre o modelo de adaptador escolhido e o protocolo de camada física.

O protocolo *Token Ring*, do inglês *token* que significa bastão trabalha com a topologia anel, isto é, cada estação realiza a transmissão de dados quando recebe autorização. Essa autorização está vinculada à “posse” do *token*, e dura um período determinado, chamado de janela. Cada equipamento deve esperar autorização para transmitir, mesmo que a rede esteja livre, restringindo a velocidade de transmissão.

Atualmente a maior parte do tráfego na Internet e redes locais originam-se e terminam com conexões *Ethernet*. O protocolo *Ethernet* utiliza a metodologia CSMA/CD (*Carrier Sense Multiple Access With Collision Detection*). A técnica consiste em identificar se o meio está ocupado, isto é, se alguma estação está transmitindo. Se o meio estiver livre, a transmissão é iniciada, caso contrário uma temporização de espera é gerada aleatoriamente para uma nova tentativa de transmissão. Dependendo da ocupação do meio, isto é, da quantidade de estações que estão transmitindo e recebendo dados, há possibilidade de dois dispositivos iniciarem a transmissão simultaneamente. Caso isso ocorra é sinalizado no meio físico uma colisão e são gerados nas estações concorrentes uma temporização aleatória para uma nova tentativa de transmissão (TANEMBAUM, 1997).

Redes FDDI adotam a tecnologia de transmissão idêntica às redes *Token Ring*, alterando o meio de transmissão de cabos metálicos para fibras óticas. Com a utilização de luz para transmitir informações e com a pequena atenuação dos cabos óticos, atualmente com as tecnologias que utilizam raios laser e fibras óticas monomodo é possível conseguir larguras de banda na faixa de Gigabits por segundo a distância de até 200 quilômetros.

2.1.2 Protocolos de conexão lógica

A conexão lógica utiliza padrões também denominados protocolos. O principal protocolo utilizado na Internet é chamado de TCP/IP (*Transmission Control Protocol / Internet Protocol*).

As aplicações são os *softwares* ou programas de computador que utilizam os protocolos e os dispositivos físicos para troca de dados. Dentre as principais aplicações utilizadas atualmente podem-se citar os navegadores (*browsers*) e os clientes de correio eletrônico. Dentre os principais softwares navegadores temos, por exemplo: Internet Explorer, Mozilla, Chrome da Google, entre outros.

2.2 Conceitos básicos de redes

Apesar da aplicação inicial ter sido para fins militares, foi com a evolução dos aplicativos empresariais que as redes de dados tornaram-se populares. A necessidade de compartilhar dados e recursos em vários computadores de maneira eficiente e econômica tornou-se diferencial competitivo entre as empresas.

No início dos anos 1980 as tecnologias de rede surgiram usando diferentes implementações de *hardware* e *software*, sendo que cada fornecedor de soluções de conectividade criava seu *hardware* e *software*. Ao optar por determinado fabricante, o cliente ficava dependente daquela tecnologia específica.

Com a necessidade de garantir o investimento e possibilitar a evolução das redes de computadores, uma das primeiras soluções foi criar padrões com um conjunto aberto de diretrizes ao desenvolvimento de dispositivos de rede e os *softwares* utilizados.

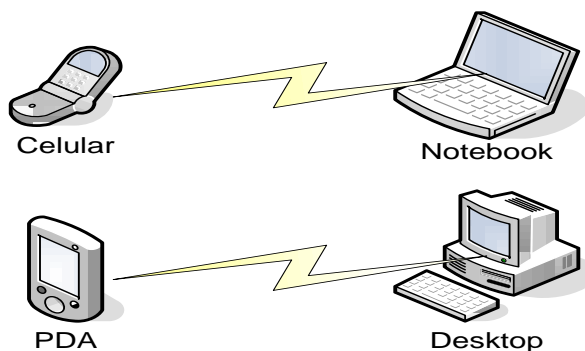
Durante o processo de padronização houve a necessidade de definir as redes de dados quanto à abrangência geográfica. Atualmente os principais conceitos são:

- PAN – *Personal Area Network*;
- LAN – *Local Area Network*;
- MAN - *Metropolitan Area Network*;
- WAN – *Wide Area Network*;
- SAN – *Storage Area Network*;

- VPN – *Virtual Private Network*.

As redes denominadas PAN ou *Personal Area Network* caracterizam-se por um pequeno raio de atuação e número limitado de dispositivos interconectados. Como exemplo de uma PAN pode-se citar uma rede utilizando tecnologia Bluetooth ou um cabo USB, exibido na Figura 1.

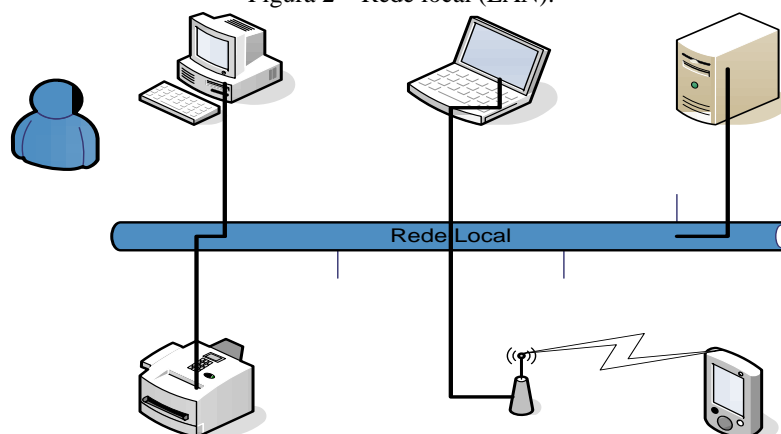
Figura 1 – Rede de abrangência pessoal (PAN).



Fonte: (SPOZITO, 2011)

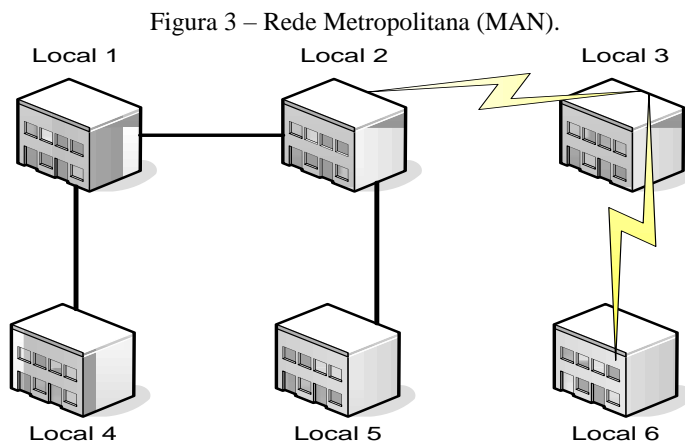
Redes locais (Figura 2) ou *Local Area Network* – LAN são utilizadas no compartilhamento de dados e recursos em um ambiente geográfico delimitado, como por exemplo, um escritório ou várias salas. Com as tecnologias atuais até mesmo um edifício com vários andares pode ser classificado com uma rede LAN.

Figura 2 – Rede local (LAN).



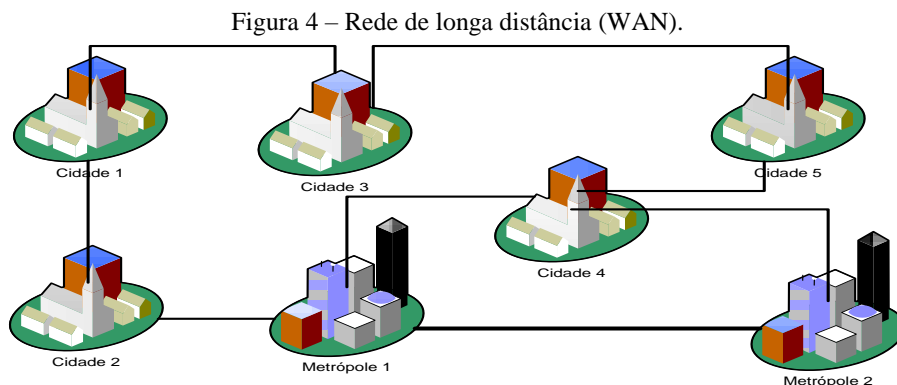
Fonte: (SPOZITO, 2011)

Uma rede local metropolitana (*Metropolitan Area Network – MAN*) trabalha com o fundamento de metrópole ou cidade (Figura 3). Pode-se dizer que uma rede MAN está situada entre uma rede LAN e uma rede WAN. As universidades ou centros universitários são exemplos de redes MAN, sendo que nessas situações há necessidade de conectar vários prédios para compartilhamento de dados e recursos.



Fonte: (SPOZITO, 2011)

Do conceito de *Wide Area Network – WAN* (rede alargada ou longa) surgiu a Internet. Mesmo com a possibilidade de interligar computadores e prédios com as redes LAN e MAN, o compartilhamento de informações entre cidades (Figura 4) e até mesmo países era iminente. Foi a partir dessa necessidade e das restrições tecnológicas quanto ao transporte de dados a longas distâncias que surgiram novos protocolos de comunicação. As redes WAN são o principal foco deste trabalho e serão abordadas de modo mais profundo.



Fonte: (SPOZITO, 2011)

Os conceitos de rede SAN (*Storage Area Network*) e VPN (*Virtual Private Network*) são relativamente novos e estão relacionados aos conceitos de compartilhamento de recursos e segurança. O conceito de rede SAN consiste em acoplar um dispositivo de armazenamento como um disco rígido (*Hard Disk – HD*) ou fita DDS – *Digital Data Storage* que utiliza o mesmo princípio da fita DAT (*Digital Audio Storage*) em uma rede local, possibilitando salvar e recuperar informações sem a necessidade de um microcomputador como intermediário. Numa rede SAN há também critérios de segurança e replicação de dados objetivando garantir a integridade das informações.

Numa VPN temos o conceito de rede privada ou particular. A rede trabalha com um protocolo de codificação de dados sobre uma rede TCP/IP e visa garantir que a troca de informações entre diferentes pontos ocorra de forma segura. A VPN surgiu da necessidade de garantir a segurança das informações que as redes WAN disponibilizaram através da Internet. Com o compartilhamento dos meios há também uma fragilidade quanto à integridade das informações trafegadas, que podem ser acessadas e modificadas por terceiros.

2.3 Meios físicos para redes

Todos os componentes pertencentes a uma rede, desde uma PAN até uma WAN são dotados de dispositivos eletrônicos e *softwares* desenvolvidos, normalmente conforme padrões internacionais de acesso público, descritos pelo *Institute of Electrical and Electronics Engineers - IEEE* ou pela *International Organization for Standardization – ISO*.

O conceito de *host* em informática é definido como qualquer máquina ou computador que possa ser conectado a uma rede, operando em camada 3 do modelo TCP/IP.

Para realizar a comunicação entre dois *hosts* é necessária a utilização de apenas um cabo ou vários equipamentos, dependendo fundamentalmente de sua localização geográfica e do protocolo utilizado.

Os principais equipamentos utilizados atualmente nas redes de dados são:

- Placas de rede (*Network Interface Card - NIC*);
- HUB;
- *Switch*;
- Roteador ;
- Repetidor;

- Conversor de Mídia;
- Pontos de acesso;
- Regeneradores.

Os principais meios na qual as informações trafegam são:

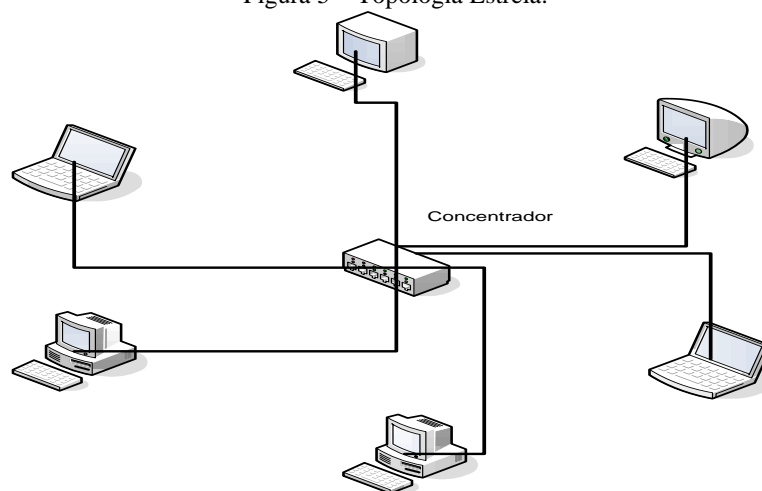
- Cabos metálicos;
- Cabos óticos;
- Rádio Frequência.

As placas de rede são responsáveis por converter as informações binárias de um microcomputador em sinais elétricos no caso da utilização de cabos metálicos; luminosos quando a ligação entre dois pontos é feito por fibra ótica ou rádio frequência – RF em placas de rede sem fio (*wireless*).

Para conectar três ou mais *hosts* em uma LAN, utiliza-se um equipamento que concentre em um ponto a comunicação de todas as placas de rede conforme o meio e topologia adotados.

Apesar da existência de vários formatos para a interligação dos pontos, atualmente a topologia aplicada na maioria das redes é denominada do tipo estrela (Figura 5). Nesse formato, todos os dispositivos estão conectados a um ponto central.

Figura 5 – Topologia Estrela.



Fonte: (SPOZITO, 2011)

Os equipamentos utilizados para a conexão da rede local cabeada são chamados de *hub* e *switch*, independente do meio utilizado. Isto é, podem ser utilizados cabos metálicos ou fibra ótica para conectar os *hosts* ao concentrador. Para isso deve ser respeitado apenas o tipo de cabo suportado pela placa de rede e concentrador, pois existem placas de rede e *switch* para cabos do tipo par metálico ou fibra ótica.

A escolha da tecnologia depende da aplicação, condições ambientais e principalmente do orçamento disponível. Portanto há situações em que são utilizados cabos óticos e cabos metálicos, nesse modelo de infraestrutura mista são instalados conversores de mídia, responsáveis por converter sinais elétricos utilizados pelas placas de rede comuns para as placas de rede óticas.

Outra tecnologia recente são as placas de rede sem fio ou *wireless*, que permitem a comunicação entre dois ou mais *hosts* sem que haja necessidade de realizar conexões físicas entre os dispositivos, pois o sinal utiliza tecnologia de Rádio Frequência – RF para troca de informações.

Diante dos diferentes formatos de comunicação, seja utilizando sinais elétricos, fibra ótica ou canal de RF, há de se considerar as peculiaridades de cada meio ou tecnologia utilizada.

Os cabos metálicos são regidos pelos fundamentos estudados nas áreas de eletrotécnica, eletrônica e campos magnéticos. As peculiaridades surgem quando são projetados e montados circuitos que trabalham com variações da DDP – Diferença de Potencial, corrente elétrica (alternada ou contínua) e também do campo magnético. Tais grandezas geram comportamentos diferentes em um mesmo circuito ou cabo.

Considerando, por exemplo, a geometria de um cabo de rede, composto por quatro pares de fios paralelos, separados por um isolante, tem-se um capacitor intrínseco ao meio.

A variação da frequência nessa situação pode tornar a admitância desfavorável. No caso de computadores e redes de dados, os circuitos operam em altas frequências. Sendo a admitância dependente da frequência, haveria distorções no resultado.

Em se tratando de cabos óticos, o principio fundamental para a transmissão da luz é a refração. O fenômeno da refração depende da espessura, permissividade e qualidade do material que a luz atravessa. O composto que reveste o núcleo também tem papel fundamental no funcionamento da fibra ótica. Qualquer alteração impacta nos ângulos de reflexão e também na atenuação ou perda do sinal luminoso que percorre o cabo ótico. Ao utilizar cabeamento ótico, há menos suscetibilidade a interferências eletromagnéticas.

Há também a possibilidade de transmitir, em um mesmo cabo ótico, vários sinais diferentes mudando apenas o comprimento de onda, isto é, alterando a cor da luz utilizada.

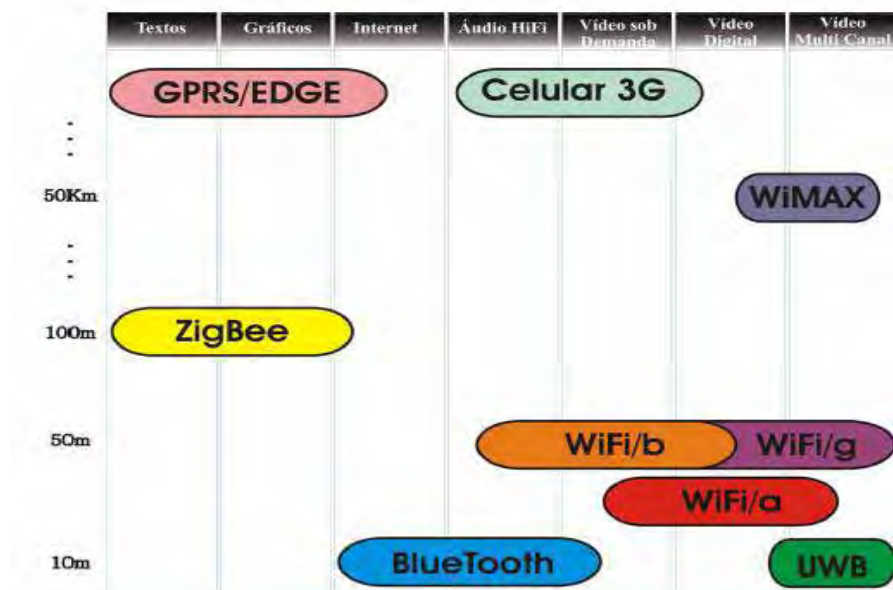
No caso das redes sem fio ou *wireless*, devido à grande quantidade de equipamentos eletrônicos hoje em funcionamento a interferência sofrida por esse tipo de rede é elevada, além dos equipamentos sofrerem naturalmente a interferências causadas por intempéries.

A saturação ambiental em se tratando do uso de rádio frequência pode ser facilmente justificada dada a idade da criação do primeiro sistema que utilizou ondas eletromagnéticas, nos anos 1890, feito este atribuído a Guglielmo Marconi.

Apesar do exposto, a necessidade de se manter conectado, aliada à mobilidade propiciada pelas redes sem fio, gerou uma gama de tecnologias que permitem a comunicação de forma eficiente (ZUCATO, 2009). Atribui-se à rede de telefonia móvel celular as principais mudanças nos contextos sociais e tecnológicos associados às vantagens na utilização de comunicação móvel (BALSEMÃO, 2008).

Aplicações que vão desde a transmissão de vídeo ou simplesmente arquivos de texto são atendidas, mas há necessidade de maior investimento e pesquisa na escolha da tecnologia em conformidade com as necessidades. A Figura 6 ilustra um panorama geral das pesquisas nessa área.

Figura 6 - Tecnologias *wireless*.



Fonte: (ZUCATO, 2009)

As redes de cabo metálico são atualmente as mais utilizadas, dados os seguintes fatores: O custo do cabo metálico é menor que o da fibra ótica e a instalação mais simples. São menos suscetíveis a interferências eletromagnéticas se comparados aos meios sem fio. Há mão de obra qualificada no mercado para instalar e manter o cabeamento a custo baixo e os equipamentos utilizados na montagem e manutenção da rede são de fácil manuseio.

2.4 Redes WAN e largura de banda

As redes de longa distância ou redes WAN são o principal modelo de rede tratado neste trabalho.

Esse tipo de rede interconecta as redes locais através de uma ampla área geográfica, permitindo que empresas, universidades e usuários domésticos se comuniquem a longas distâncias. Toda a Internet está baseada na utilização de redes WAN e suas tecnologias.

As redes WAN são projetadas para executar as seguintes ações:

- Operar em grandes áreas separadas geograficamente;
- Permitir a comunicação entre usuários com latência relativamente pequena, passando a impressão de operar em tempo real, isto é, com tempo de resposta menor que 100 ms;
- Garantir a conexão permanente de recursos em qualquer lugar do planeta.

Como as redes WAN operam a longas distâncias e interligam redes locais distintas devem-se considerar alguns fatores.

A taxa de transferência de dados é inferior à das redes locais. Essa diferença é justificada pelas limitações impostas por condições de infraestrutura física e tecnologias adotadas. Atualmente os principais meios de ativação das redes WAN nas instituições dá-se por meio de cabos metálicos via rede do serviço de telefonia pública comutada - STPC. Com as grandes distâncias as condições dos cabos não são homogêneas. Mesmo assim, devido à grande capilaridade, atualmente esse é o principal meio utilizado na ativação de redes WAN.

Como o cabeamento instalado nas cidades foi projetado para a transmissão de sinais analógicos, são utilizados Modems, equipamentos responsáveis pela conversão dos dados digitais em analógicos e vice-versa, ou seja, modular e demodular sinais.

Dada a limitação física imposta, as empresas de telecomunicações optaram por desenvolver novas formas de modular e codificar os sinais digitais. Surgiram então as principais tecnologias utilizadas em redes WAN:

- *Integrated Services Digital Network* (ISDN);
- *Digital Subscriber Line* (DSL);
- *Frame Relay*;
- Hierarquias Digitais T (EUA) e E (Europa): T1, E1, T3, E3;
- *Synchronous Optical Network* (SONET) utilizada em cabos ópticos.

Como o volume de informações transmitidas pelas redes WAN durante determinado período de tempo é menor se comparada às redes locais, um fator importante a se considerar é a largura de banda, definida exatamente como a quantidade de informações que flui na rede por um período T. Assim sendo, é possível afirmar:

- a largura de banda pode ser associada à velocidade da rede e, como depende da infraestrutura e da tecnologia adotada, é finita;
- para aumentar a largura de banda em uma rede WAN é preciso alterar o meio físico, mudar a tecnologia ou agregar várias conexões que utilizam uma mesma tecnologia. Portanto, aumentar a largura de banda gera custos adicionais;
- largura de banda é fator fundamental na análise do desempenho da rede e precisa ser dimensionada conforme a aplicação prevista. Como exemplo pode-se citar o uso de um *link* de dados para rede WAN utilizado para interligar servidores de *e-mail* ou sistemas de videoconferência;
- historicamente tem-se registrado um crescimento da demanda por largura de banda.

Além das aplicações comerciais e pesquisa, a possibilidade de adquirir microcomputadores pessoais criou uma demanda expressiva do tráfego na Internet.

Como dado importante no crescimento do uso de computadores no ambiente doméstico, no ano de 2006, no Brasil, dois milhões e duzentas mil pessoas adquiriram o primeiro computador, justificando o percentual de crescimento de 20% do parque nacional dos equipamentos de uso privado.

Segundo publicação realizada pela empresa DMR *Consulting*, exibido no site Everis¹ e relatório do Tribunal de Contas da União – TCU² as Tecnologias da Informação e as Comunicações – TIC mantiveram a posição, devido principalmente ao fato de o número de telefones móveis ter atingido 517 unidades para cada 1.000 habitantes, o que significa 24,5% a mais do que um ano antes e a manutenção do papel de maior impulsionador das TIC. Também foi relevante a contribuição do número de computadores, que somam 130 para cada 1.000 habitantes, e um aumento institucional de 16%.

A Tabela 1 mostra o perfil da inclusão digital no país a partir do censo do ano 2000. Considerando que atualmente o principal meio de acesso à Internet é o microcomputador, a possibilidade de acesso a máquina torna-se relevante no contexto.

Tabela 1 - Unidades da federação brasileira com maior e menor grau de inclusão digital.

	Moradores em domicílios particulares permanentes	Acesso a computador	(%) moradores com acesso
Brasil	168.450.489	17.328.185	10,29
Unidades da Federação Maior Inclusão Digital			
Distrito Federal	2.035.459	485.820	23,87
São Paulo	36.719.202	6.603.586	17,89
Rio de Janeiro	14.289.735	2.217.769	15,51
Santa Catarina	5.319.120	654.177	12,30
Paraná	9.471.919	1.097.529	11,59
Unidades da Federação Menor Inclusão digital			
Maranhão	5.621.913	115.211	2,05
Piauí	2.832.095	78.811	2,78
Tocantins	1.143.283	31.533	2,76
Acre	552.016	18.881	3,42
Alagoas	2.797.246	100.664	3,60

Fonte: TRIBUNAL DE CONTAS DA UNIÃO – TCU

¹ Disponível em: <http://www.everis.com.br/imprensa-e-publicacoes/notas-de-imprensa/2007/isi-julho-06.asp>

² Disponível em: <http://portal2.tcu.gov.br/portal/pls/portal/docs/669944.PDF>

Confome dados do obtidos pelo Instituto Brasileiro de Geografia e Estatística – IBGE (2000), o Brasil possui 168.450.489 habitantes em domicílios particulares permanentes, dos quais 17.328.185 têm acesso a computadores, o que representa 10,29% do total.

O estado de São Paulo posiciona-se em primeiro lugar, com 36.719.202 habitantes, dos quais 6.603.586 têm acesso a computadores – 17,89% do total.

Algumas comunidades têm se destacado na implantação e manutenção de provedores de acesso gratuito à Internet contemplando usuários dos mais variados perfis.

Em 2005, o Brasil sediou, pela primeira vez, a conferência mundial "Oportunidade da Internet sem fio para as comunidades carentes na América Latina: uma perspectiva brasileira". Na ocasião, destacou-se o programa Município Digital, arquitetado pela prefeitura da cidade de Piraí, Rio de Janeiro, como parte da proposta para impulsionar o desenvolvimento local de Piraí, que contava, então com aproximadamente 22.500 habitantes.

O programa que visa à democratização do acesso aos meios de informação e comunicação, gerando oportunidades de desenvolvimento econômico e social a partir da instalação de um provedor de acesso gratuito à Internet, via rádio, tornou-se exemplo de otimização do uso dos recursos tecnológicos da informação e da comunicação.

Outro exemplo de inclusão digital encontra-se na cidade de Sud Mennucci, interior do Estado de São Paulo e no ano de 2009, o mesmo projeto foi implantado no município de Ilha Solteira, estado de SP.

Como exemplo do crescimento na demanda de largura de banda, a Prefeitura Municipal de Ilha Solteira instalou um *link* de dados conectado ao *Backbone* Internet via cabo ótico com largura de banda de 10 Mbps, com tecnologia para ampliar até 32 Mbps.

Se comparada as velocidades hoje disponibilizadas pelas redes locais, é possível notar uma discrepância. As primeiras redes locais utilizavam cabos metálicos coaxiais e permitiam numa distância de até 300 m uma largura de banda de 10 Mbps.

Atualmente as redes locais projetadas utilizam o padrão regido pelas normas EIA/TIA-568-A/B sob protocolo *Ethernet* categoria cinco (CAT5) que opera em frequências de até 100 MHz e largura de banda de 100 Mbps. Há uma tendência em projetar redes locais com o padrão EIA/TIA 568-A2/B2 ou categoria seis (CAT 6) operando a 250 MHz com largura de banda de até 1000 Mbps.

Taxas de transferência adotadas em redes locais com 100 Mbps chegando à faixa de Gbps são aplicadas em redes WAN apenas por operadoras de telecomunicações, sendo que essas agregam vários *links* de dados comercializados separadamente a pequenas, médias e grandes empresas, justificando o investimento necessário.

2.5 Modelo TCP/IP

O modelo TCP/IP – (*Transmission Control Protocol / Internet Protocol*) surgiu com objetivo de garantir as comunicações em aplicações militares.

O Departamento de Defesa dos Estados Unidos (*Department of Defense - DoD*) desenvolveu o modelo pois precisava de uma rede que pudesse continuar operando em qualquer condição, mesmo na ocorrência de uma guerra nuclear. Deste modo já havia no mundo diferentes tipos de meios de comunicação como fios de cobre, satélites, sistemas de microondas utilizados pelas operadoras de telefonia. Os pontos remotos eram interconectados por linhas privadas de 56 kbps cedidas pelas operadoras (TANEMBAUM, 1997).

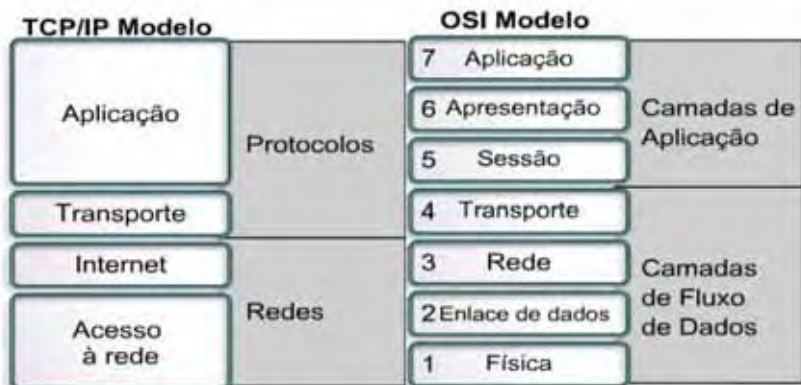
A adoção de um sistema que trabalha com comutação por pacotes e não por circuitos possibilitou implementar o projeto. Caso a comunicação entre dois pontos fosse interrompida, a informação seria direcionada para outro caminho, chegando ao destino previsto, diferentemente da comutação por circuitos, na qual a comunicação é interrompida.

O modelo TCP/IP é dividido em quatro camadas: (ODOM; KNOT, 2006)

- Camada de Aplicação;
- Camada de Transporte;
- Camada de Rede (ou Internet);
- Camada de Acesso à Rede;

A Figura 7 ilustra a disposição das camadas do modelo TCP/IP.

Figura 7 – Modelo OSI e TCP/IP.



Fonte: (TANEMBAUM, 1997).

Trabalhar com camadas simplifica a tarefa dos desenvolvedores na criação de novos sistemas e produtos e também na manutenção das redes em operação.

Na Figura 7 é possível observar também outro modelo de camadas denominado OSI. O modelo OSI foi a primeira tentativa dos organismos internacionais em criar um padrão que possibilitasse aos desenvolvedores de tecnologia de redes de dados trabalhar com os conceitos de compatibilidade. Tal iniciativa foi fundamental em se tratando de manutenção do parque tecnológico instalado, preservando o investimento dos usuários de redes de dados tanto na integração dos sistemas projetados e produzidos por diferentes fornecedores, como na compatibilidade das novas tecnologias com as tecnologias de legado. A exemplo disso, atualmente é possível conectar em uma rede moderna que trafega a velocidade de Gigabits por segundo (Gbps), um dispositivo produzido há dez anos, como uma placa de rede que trafega a velocidade de dez Megabits por segundo (Mbps).

Retomando o conceito de camadas no protocolo TCP/IP, quando no desenvolvimento de um novo aplicativo que demande acesso a uma LAN ou à Internet o programador utiliza bibliotecas prontas que operam na camada de Aplicação.

Cabe aos projetistas e administradores da rede de dados pensarem nas camadas de Transporte e Aplicação, configurando o sistema operacional e o protocolo TCP/IP com as características pertinentes às necessidades específicas dos ambientes. Nessa etapa, deve-se considerar a arquitetura da rede, número de *hosts*, aplicativos utilizados, largura de banda e distância geográfica entre os pontos.

Aos engenheiros e fabricantes de equipamentos de conectividade fica a responsabilidade de projetar os equipamentos utilizados nas conexões físicas entre os *hosts*. Esses equipamentos são responsáveis pela camada de acesso à rede. Nessa camada são escolhidos os tipos de componentes eletrônicos conforme o meio escolhido (óptico, elétrico, RF), a modulação do sinal, temporização, etc.

O modelo TCP/IP permitiu a disseminação da Internet em âmbito mundial, dado principalmente aos fatos de utilizar dispositivos como placas de rede e cabos de diversos fabricantes além da possibilidade de comunicação entre *softwares* aplicativos desenvolvidos com várias finalidades por empresas diversificadas.

2.6 Conjunto de protocolos TCP/IP e endereçamento IP

O modelo TCP/IP na versão quatro, denominado IPv4, opera no sistema numérico Base 2 ou binário, compatível com a arquitetura dos microcomputadores. Todas as operações básicas são realizadas respeitando as regras de álgebra Booleana, em especial o AND (e) e OR (ou) Booleano.

Para que haja uma comunicação eficaz entre dois pontos distintos, não pode haver na rede dois endereços idênticos. Quando se fala de rede, o parâmetro é a Internet, portanto, não deve haver no mundo dois endereços idênticos.

Cada *host* na Internet é representado por um número decimal pontuado em quatro octetos. Somados os octetos, há 32 bits representando cada ponto (Figura 8).

A esse conjunto de octeto dá-se o nome de endereçamento IP. Portanto cada *host* possui seu próprio IP.

Um octeto possui oito bits e pode representar até 256 números decimais diferentes. Para facilitar a visualização do endereço IP utiliza-se o seguinte formato:

Figura 8 – Representação de um endereço IPv4.

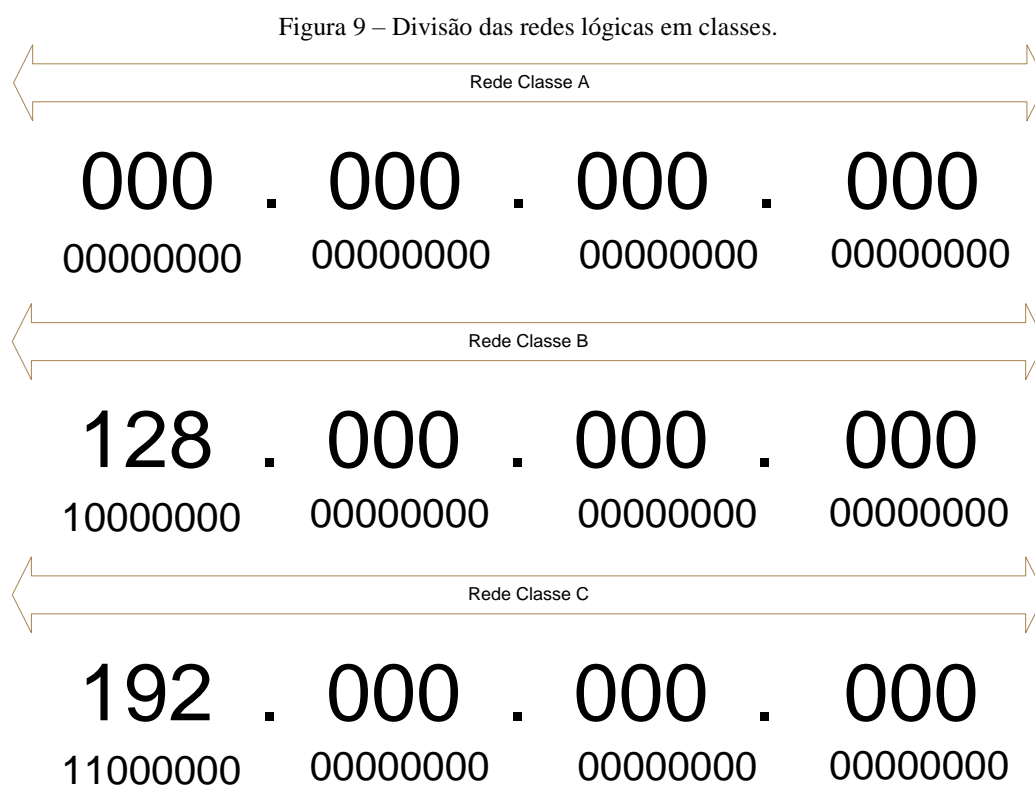
XXX	.	XXX	.	XXX	.	XXX
00110011		00110011		00110011		00110011

Fonte: (SPOZITO, 2011)

Para melhor aproveitamento dos endereços IPs, devido à abrangência da Internet, houve uma divisão das faixas de endereços em classes. O objetivo era possibilitar a divisão da rede em subredes, facilitando a distribuição e administração.

Para identificar a que classe de rede um determinado número IP pertence, utilizam-se os primeiros bits do primeiro octeto. Deste modo, são definidas faixas de endereços conforme a ativação desses bits.

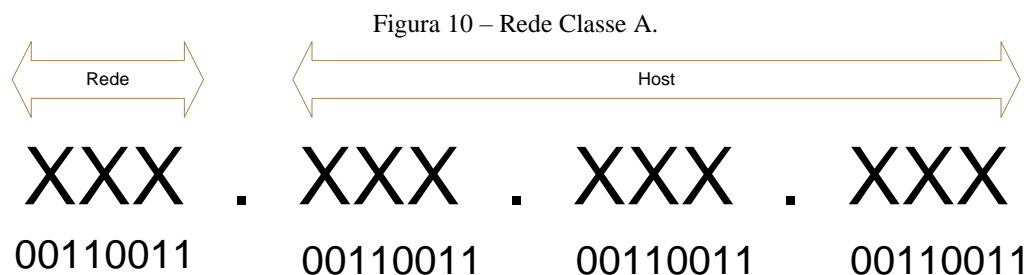
As redes mascaradas são classificadas em classes A, B e C, e a ativação dos bits para divisão pode ser observada na Figura 9.



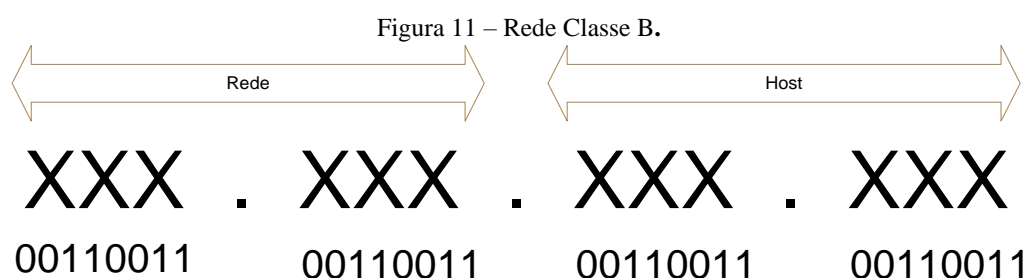
Fonte: (SPOZITO, 2011)

Como aplicação prática, a utilização de um segundo número composto por quatro octetos binários, denominado máscara de sub-rede possibilitou um melhor aproveitamento do endereçamento IP, facilitando a distribuição e gerência. Associando os dois conceitos, observa-se a divisão das classes e sub-redes nas figuras 10, 11 e 12.

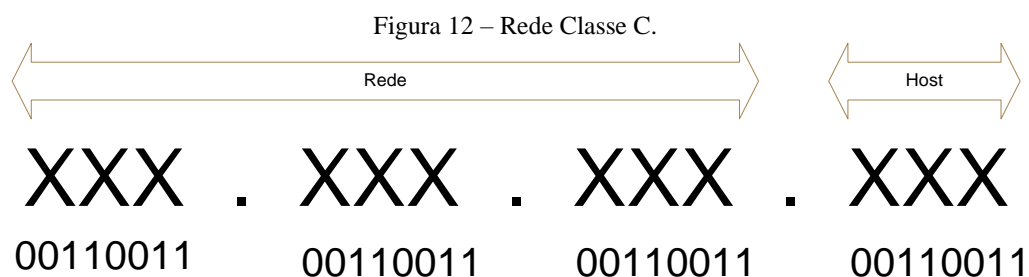
Máscaras definidas decimalmente como: 255.0.0.0 estão associadas a redes Classe A. Às redes Classe B, convencionam-se máscaras 255.255.0.0 e redes pertencentes à Classe C utilizam máscaras 255.255.255.0.



Fonte: (SPOZITO, 2011)



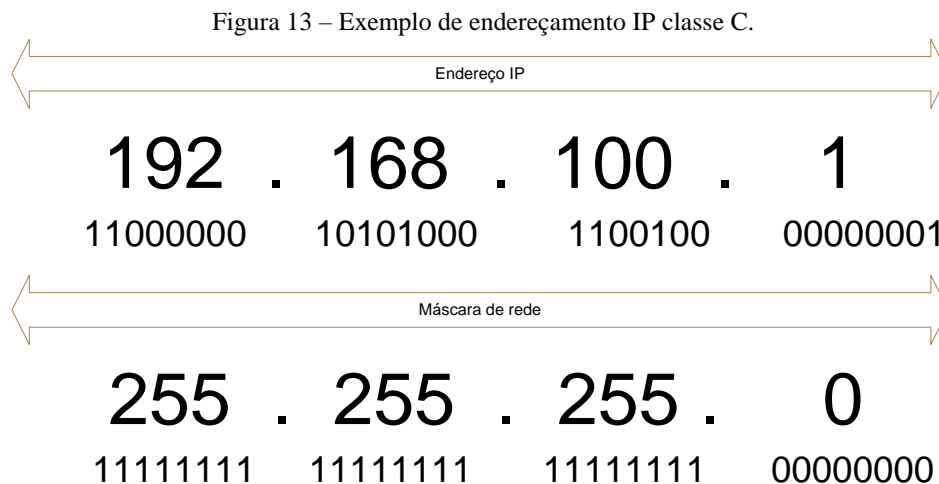
Fonte: (SPOZITO, 2011)



Fonte: (SPOZITO, 2011)

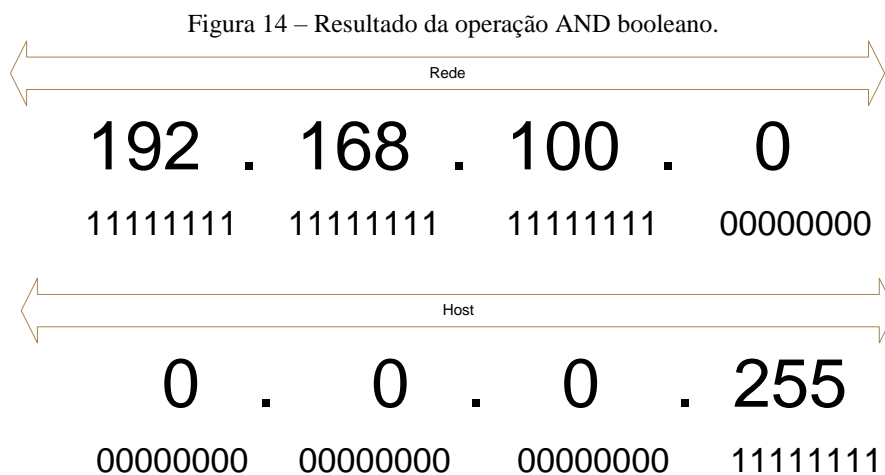
Não existem controles nem bloqueios nos arquivos de configuração dos sistemas operacionais que restrinjam a utilização e funcionamento de redes TCP/IP dentro desses critérios. Com excessão da Internet, desde que utilizando o mesmo critério, qualquer rede configurada isoladamente opera normalmente o protocolo IPv4 utilizando máscaras de sub-rede em conformidade com a padronização descrita ou não.

Ao realizar a operação AND entre o endereço IP e a Máscara de rede obtém-se como resultado a qual rede o *host* pertence. Na Figura 13 é ilustrado um endereço IP pertencente a uma rede Classe C.



Fonte: (SPOZITO, 2011)

Após a operação AND booleano do endereço IP com a Máscara de subrede obtém a rede à qual pertence o *host* e o número do *host*, como ilustrado na Figura 14.



Fonte: (SPOZITO, 2011)

Nesse cenário cabe aos roteadores encaminharem os pacotes com os dados à rede correta. Esta é uma breve ilustração do efeito que uma máscara de rede tem sobre um endereço IP.

Além do conceito de redes e subredes IP, outro elemento fundamental é o tipo de informação que será enviada de determinado ponto A para um ponto remoto B.

Para tratar diferentes tipos de informações e as peculiaridades da aplicação de cada uma, foram desenvolvidos “pacotes” específicos que em conjunto formam a gama de protocolos pertencentes ao padrão TCP/IP.

Os protocolos estão associados às diferentes camadas do modelo TCP/IP e cada um é projetado para tratar o formato que os dados são transmitidos, em conformidade com a aplicação.

Os projetistas do TCP/IP decidiram que os protocolos de mais alto nível deviam incluir os detalhes da camada de sessão e de apresentação. A camada de transporte lida com questões de qualidade, confiabilidade, controle de fluxo e correção de erro. Portanto, o protocolo TCP é orientado a conexões.

O protocolo IP governa o caminho que os pacotes devem seguir.

Alguns dos principais protocolos e suas aplicações especificadas no modelo de referência TCP/IP são:

- *File Transfer Protocol* – FTP;
- *Hypertext Transfer Protocol* – HTTP;
- *Simple Mail Transfer Protocol* – SMTP;
- *Domain Name System* – DNS;
- *Trivial File Transfer Protocol* – TFTP.

Os protocolos mais comuns na camada de transporte incluem:

- *Transmission Control Protocol* – TCP;
- *User Datagram Protocol* – UDP.

Já o principal protocolo da camada Internet é:

- *Internet Protocol* – IP.

É possível visualizar que o termo TCP/IP implica numa série de conceitos e diferentes protocolos que possibilitam a operação dos diversos tipos de *softwares* operando hoje em redes WAN.

Portanto, dimensionar e planejar uma rede WAN envolve desde a aplicação prevista, número de usuários e aplicativos até a abrangência geográfica pretendida.

2.7 Qualidade em redes TCP/IP

Qualidade em redes pode ser definida sob dois pontos de vista (SANTANA, 2006). O primeiro consiste no resultado experimentado pelo usuário por intermédio de um equipamento para determinada aplicação como um telefone ou um microcomputador com acesso à Internet. Num segundo parâmetro, qualidade está relacionada ao comportamento ou funcionamento de sistemas, independente da percepção do usuário.

Duas aplicações práticas podem subjetivamente distinguir qualidade dentro de um mesmo sistema de comunicação. Utilizando um *link* de dados com taxas de transferência de 56 kbps entre dois pontos, seja na troca de mensagens eletrônicas como e-mails e para comunicação de voz sobre ip (VoiP) pode-se subjetivamente afirmar que o sistema oferece boa qualidade para o primeiro exemplo e qualidade ruim para conversas de voz, dado o fato que haveriam atrasos e perda de informações durante o diálogo.

Dentro desse contexto, algumas métricas são adotadas como parâmetro para medição de qualidade em redes TCP/IP. (FERREIRA, 2005; SANTANA, 2006; ZARPELÃO, 2010). Aqui são listados os parâmetros mais comuns:

a) Vazão / capacidade:

Refere-se à quantidade de informações que podem ser transferidas através da rede em certo período de tempo, normalmente expressa em bits por segundo.

b) Atraso:

O atraso refere-se ao tempo necessário para que um pacote chegue ao destino. Os atrasos são afetados diretamente pelas soluções tecnológicas adotadas e também pelas condições ambientais ou condições do meio que o sinal trafega.

c) Variação do atraso (*jitter*):

O *jitter* pode ser definido como a diferença entre os atrasos na recepção de pacotes consecutivos.

d) *Skew* (obliquidade):

Parâmetro utilizado para medir a diferença entre a chegada das informações que em conformidade com a aplicação devem estar sincronizadas.

e) Perda de pacotes:

Consiste na relação entre a quantidade de pacotes ou bits transmitidos e entregues corretamente ao destino. Em redes de alta disponibilidade a perda de pacotes deve ser zero, em casos de congestionamento, os sistemas devem ser tratados para considerar e contornar o problema.

2.8 PING

O comando PING (*Packet Internet Grouper* ou Procurador de Pacotes da Internet) utiliza o protocolo ICMP (*Internet Control Message Protocol*) (ODOM; KNOT, 2006).

A idéia da ferramenta consiste em enviar um pacote de dados a determinado destino e caso a operação tenha sucesso, é retornado um pacote confirmando o recebimento da requisição agregando ao pacote informações como número de saltos, bytes, entre outras.

O tempo entre o envio e retorno da informação é calculado com base no relógio de tempo real do equipamento que originou a requisição.

Como o comando PING utiliza o protocolo ICMP e esse trabalha na camada de rede, ao receber a confirmação de uma requisição é possível afirmar que até a terceira camada (rede) os *hosts* estão configurados corretamente.

O comando PING será a principal ferramenta utilizada nesse trabalho. Com base nas informações que o pacote carrega será possível verificar se determinado ponto remoto está ativo e os roteadores corretamente configurados.

Para confirmar a operação de camadas superiores, há necessidade de utilizar aplicativos com protocolos HTTP, FTP ou mesmo o comando telnet, validando então todo o processo de comunicação entre dois ou mais pontos.

Um exemplo de utilização do comando PING pode ser observado na Figura 15, na qual é disparado um pacote de 56 bytes a o domínio “wikipedia.com”. Após a descoberta do IP (130.94.122.195), o microcomputador dispara o pacote para o destino que recebendo a requisição

retorna à origem uma informação de confirmação. Calculando a diferença entre o horário de disparo e chegada obtem-se o tempo de ida e volta da informação.

Figura 15 – Disparo de um comando PING para wikipedia.com.

```
$ ping -c 5 wikipedia.com
PING wikipedia.com (130.94.122.195): 56 data bytes
64 bytes from 130.94.122.195: icmp_seq=0 ttl=235 time=284.3 ms
64 bytes from 130.94.122.195: icmp_seq=1 ttl=235 time=292.9 ms
64 bytes from 130.94.122.195: icmp_seq=2 ttl=235 time=289.7 ms
64 bytes from 130.94.122.195: icmp_seq=3 ttl=235 time=282.4 ms
64 bytes from 130.94.122.195: icmp_seq=4 ttl=235 time=272.0 ms

--- wikipedia.com ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 272.0/284.2/292.9 ms
```

Fonte: (SPOZITO, 2011)

Apesar de simples, o comando PING retorna diretamente dois dos principais parâmetros utilizados na métrica de qualidade de redes.

A primeira métrica pode ser obtida pela quantidade de pacotes que são enviados e recebidos, determinando a perda de pacotes.

Um segundo critério consiste no tempo de envio e retorno do pacote, definindo o atraso no sistema.

É possível, também, com uma série de disparos, calcular a diferença entre os atrasos ou o *jitter*, outra métrica de qualidade.

Apesar de diretamente considerar neste trabalho duas métricas como parâmetro de qualidade, deve-se considerar fatores que podem mascarar alguns resultados. O tempo exibido pelo comando PING, por exemplo, não distingue se o atraso ocorre simetricamente na ida e volta ou se há diferenças consideráveis.

CAPÍTULO III

3 Sistemas Georreferenciados

Neste capítulo são apresentados os conceitos fundamentais de sistemas georreferenciados e alguns exemplos de aplicações.

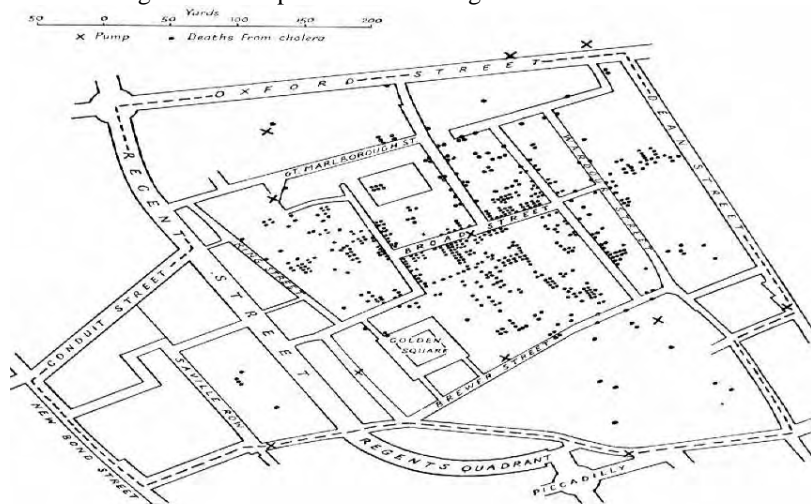
3.1 Conceitos básicos

Com o crescente uso das ferramentas computacionais para armazenar dados, sejam eles sociais, econômicos, políticos ou de qualquer domínio, tanto na iniciativa pública como privada, surge a necessidade de aprimorar ferramentas que possibilitem a geração de informações para análise e tomada de decisões.

Dependendo da informação obtida, relacioná-la a uma figura geométrica ou mapa pode facilitar a visualização e antever tendências, sejam espaciais ou temporais.

Um exemplo clássico na história da análise de informações geográficas ocorreu em 1854 no método utilizado pelo Dr. John Snow. Dados colhidos sobre a localização dos casos de cólera foram relacionados a pontos geográficos em um mapa da cidade de Londres (Figura 16), ou seja, georreferenciados. Durante a análise, o Dr. Snow percebeu que os casos de cólera se concentravam no entorno de determinados poços de coleta de água utilizada no consumo humano. Após o fechamento de um determinado poço houve redução significativa nos casos de cólera no entorno. Tal informação foi fundamental para implantar medidas no controle da doença, dado principalmente ao fato de na época não existirem registros sobre a forma de contaminação de cólera pelas autoridades sanitárias. (WIKIPEDIA, 2010)

Figura 16 – Mapa de Londres e registros de casos de cólera.



Fonte: (WIKIPEDIA, 2010)

3.2 Sistemas de Informação Geográfica - SIG

Os sistemas de informação geográfica – SIG são sistemas computacionais que tomam como base os conceitos de geoprocessamento.

O geoprocessamento envolve um conjunto de técnicas, métodos e teorias sobre a representação computacional no espaço. Nestes termos é definido todo o conjunto de tecnologias integradas voltadas à coleta de dados para uma aplicação específica. A correta definição dos tipos de dados e associação dos mesmos a uma referência espacial e/ou temporal pode com a aplicação das técnicas corretas gerarem informações que possibilitam visualizar de modo direto um acontecimento ou uma tendência.

O termo SIG é aplicado para sistemas que realizam o tratamento computacional de dados geográficos.

Deste modo os dados geográficos possuem uma dimensão espacial (ou localização), que deve refletir o mundo geográfico real como, por exemplo: imagens de satélite, mapas no formato raster (matriciais) ou mapas vetorizados.

Em uma visão global, os SIG possuem os seguintes componentes:

- Interface com o usuário;
- Entrada e integração de dados;
- Funções de processamento gráfico e de imagens;

- Visualização e plotagem;
- Armazenamento e recuperação de dados.

A interface com o usuário deve ser a forma pelo qual o operador se comunica com o sistema. Essa deve ser desenhada e programada de forma que o operador não necessite de conhecimentos técnicos computacionais avançados. O formato ideal seria dentro da aplicação desejada prover meios de atualização dos dados constantemente. Isso é possível criando processos cadastrais planejados no domínio que se deseja atuar, como por exemplo, a atualização cadastral de estudantes.

A entrada e a integração de dados devem possuir meios de relacionar os dados atualizados em uma base de dados geograficamente referenciada, isto é, deve ser inserida a informação correta que corresponda a um local geográfico cadastrado no sistema. Nesse momento também deve ser previsto o armazenamento temporal das informações do local, caso seja necessário o levantamento de um histórico pontual ou análise de tendências.

Para o processamento gráfico o sistema tratará os dados conforme os parâmetros definidos pelo operador ou programador do sistema. A modelagem e relacionamento dos dados são fundamentais na obtenção das informações. Nesse momento a informação requisitada é gerada e vinculada a um ponto referenciado geograficamente.

Na visualização e plotagem a informação obtida na etapa anterior é plotada em uma tela ou impressora. O requisito fundamental deste componente consiste em gerar a visão espacial da informação de forma a facilitar a interpretação por parte do usuário do sistema.

O armazenamento e recuperação de dados consistem em registrar um histórico dos dados alimentados e suas posições geográficas. Utilizados de modo correto, os dados armazenados além de possibilitarem a visualização de informações conforme a periodicidade de atualização dos dados, com a utilização de técnicas estatísticas e matemáticas é possível visualizar tendências como, por exemplo, uma epidemia regional ou determinado ponto de uma cidade que ocorram acidentes de trânsito constantes.

3.3 Generalização cartográfica

Os processos geográficos são dependentes da escala, isto é, a percepção do espaço analisado muda conforme a escala em que um elemento é visualizado. Com a alteração das escalas, um mesmo objeto pode ser interpretado de forma diferente.

Na generalização cartográfica, a geometria do objeto pode ser simplificada ou alterada.

Dependendo da escala, uma cidade pode tornar-se um ponto ou um grupo de objetos pode tornar-se um objeto, assim como um conjunto de lotes pode tornar-se uma quadra.

Para tanto, é necessário definir as primitivas utilizadas nos sistemas georreferenciados (DAVIS; FONSECA, 2001).

3.3.1 Ponto

Um ponto é um par ordenado (x,y) de coordenadas espaciais. É o modo mais simples de representar um geo-objeto. Todas as características geométricas do mundo real são ignoradas. Neste trabalho defini-se ponto em um único plano, desprezando o eixo ordenado z .

Postes de iluminação pública são exemplos de elementos que podem ser georreferenciados utilizando esse tipo de primitiva.

3.3.2 Linha

Define-se linha como um conjunto consecutivo de pontos em um plano, formando um segmento de reta. Esta representação é utilizada para representar objetos na qual o comprimento é muito superior à largura, como por exemplo, rios, estradas, redes de água, esgoto e atualmente redes de telecomunicações como telefonia e cabos óticos.

3.3.3 Polígono

Polígonos são utilizados para representar quaisquer objetos individualizados do mundo real. São montados com segmentos de reta interconectados, formando uma área fechada, menor que a área total do plano em que se está atuando.

3.4 Dados geográficos

Entidades geográficas são elementos que devem possuir elementos representativos do mundo real.

O espaço geográfico é o meio onde as entidades geográficas estão inseridas; pode ser considerado o hiperespaço em que as entidades serão inseridas.

Dado espacial georeferenciado é qualquer tipo de dado que indica fenômenos associados a algum atributo espacial, isto é, tenha relação direta à sua localização na superfície da terra, podendo também estar associado a um período ou data (CÂMARA et al., 1996).

Neste trabalho, os polígonos compõem a base georreferenciada onde são relacionados os pontos monitorados.

3.5 Representação vetorial/matricial

A representação matricial, também conhecida como *raster* é caracterizada por uma matriz de elementos com tamanhos regulares onde cada célula da matriz é associada a um conjunto de valores que representam as características da região. As células podem possuir formatos triangulares, hexagonais ou retangulares e são denominadas *pixels*. Os relacionamentos topológicos de cada *pixel* são determinados a partir das orientações geográficas (longitude, latitude) correspondendo implicitamente à coordenada plana de um ponto (x,y). Imagens de satélite e modelos digitais são representadas no formato *raster*.

O formato vetorial utiliza pontos, linhas e polígonos para representar a geometria das entidades geográficas. Pontos contêm apenas duas informações de coordenadas (x,y). Linhas podem ser definidas com uma seqüência de pontos e polígonos uma seqüência de linhas.

Nas figuras 17 e 18 é possível visualizar e comparar um mesmo local representado de forma vetorial ou matricial.

Importante ressaltar que a armazenagem de informações no formato vetorial implica na redução significativa da quantidade de informações que devem ser armazenadas nos sistemas computacionais.

Um mapa vetorizado utiliza informações primárias como pontos, linhas ou polígonos e “gera” no momento da visualização, com auxílio computacional, uma imagem.

O sistema matricial ou *raster* armazena a informação (*pixel a pixel*) na íntegra da imagem toda.

Se comparados, o resultado final dos dois sistemas é uma imagem na tela de um computador ou impressão em papel. Nas duas situações, quando tratada a mesma informação ou os mesmos dados georreferenciados, na mesma escala, a quantidade de pontos ou *pixels* exibidos será o mesmo.

Em termos computacionais, a necessidade de armazenar informações primárias ocupa uma quantidade de bits inferior se comparada à situação em que são armazenados todos os pontos.

Em termos mais simples, pode ser comparada a situação em que seja preciso gravar digitalmente uma folha de texto. No formato matricial serão salvos os bits correspondentes a cada ponto da folha, independentemente de existir alguma informação relevante. Há redução significativa da quantidade de informações salvas quando são registrados apenas os bits que representam as letras do texto.

A qualidade de informação armazenada também está diretamente relacionada à resolução da imagem. Deste modo, quanto maior e mais precisa a imagem, maior o tamanho do arquivo e mais espaço de armazenamento secundário o sistema computacional deverá possuir.

Os modelos matriciais e vetorizados podem ser comparados nas figuras abaixo.

Figura 17 – Mapa de São José do Rio Preto no formato vetorial.



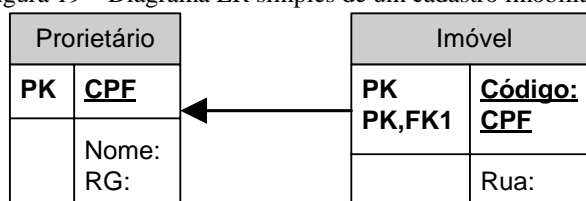
Fonte: <http://maps.google.com.br/>

memória e dispositivo secundário quando geradas. Apesar de formato seqüencial parecer mais prático, há o problema de fragmentação de memória e arquivos, pois como um sistema computacional trabalha com diversos arquivos, lendo e gravando informações, sempre que necessária a busca e registro de informações, o sistema deverá “pular” trechos ocupados por outros arquivos, o que torna o processo lento.

Com o advento dos Sistemas Gerenciadores de Banco de Dados – SGBD, não cabe mais ao programador e nem ao operador determinar como as informações são arquivadas. O próprio SGBD se encarrega de indexar e utilizar métodos computacionais otimizados de salvamento, busca e recuperação de dados. É preciso considerar também a evolução dos sistemas computacionais (*Hardware e Software*), com maior capacidade de processamento e aumento da capacidade de armazenamento dos dispositivos secundários.

Um conceito clássico utilizado nos SGBDs é o modelo de entidades e relacionamentos – ER. Esse modelo busca sistematizar o entendimento a respeito dos objetos e fenômenos que serão representados em um sistema computacional. Deste modo antes de programar a base de dados dentro de um SGBD, cabe aos desenvolvedores criarem um modelo conceitual abstrato que consiga representar da melhor forma a grandeza ou fenômeno que pretendem representar. Para isso são utilizados os conceitos de entidades, atributos e relacionamentos. Um imóvel pode ser representado pelo modelo ER conforme demonstrado na Figura 19.

Figura 19 – Diagrama ER simples de um cadastro imobiliário.



Fonte: (SPOZITO, 2011)

Além do conceito ER amplamente utilizado no desenvolvimento de sistemas, um novo formato de modelagem denominado Orientado a Objetos – OO vem sendo aplicado nos projetos de geoprocessamento largamente.

Conforme Câmara et al. (1996) a modelagem orientada a objetos não obriga o armazenamento em um SGBD orientado a objetos, mas simplesmente visa dar ao usuário maior flexibilidade na modelagem incremental da realidade.

Deste modo, os objetos geográficos se adequam bem ao modelo orientado a objetos, tal conceito pode ser notado quando é observado o fato da representação *Unified Modeling Language* - UML possibilitar um nível de abstração elevado, não vinculando o desenvolvimento a uma tecnologia específica.

As representações utilizadas na UML podem ser aplicadas tanto na representação de objetos geográficos para aplicação em SGBD, como no desenvolvimento de *softwares* que não necessariamente utilizam um SGBD.

Apesar da tendência em utilizar conceitos de orientação a objetos, grande parte dos SIG utilizados partiu de uma base cadastral existente o que implica no uso de SGBD baseados no formato ER.

Nesse trabalho, por se tratar da pesquisa para uma aplicação específica, não haverá necessidade de utilizar conceitos de orientação a objetos no SGBD, dado o fato que o projeto prevê o armazenamento de informações formato texto vinculados a uma base georreferenciada contendo apenas pontos, linhas e polígonos. Conceitos de orientação a objetos para sistemas georreferenciados, em especial os SGBDs geográficos ou OMT-G (*Object Modeling Technique for Geographic Applications*), não são aplicáveis.

No capítulo seguinte serão delimitados, claramente, os conceitos de redes TCP/IP em redes WAN e geoprocessamento aplicados neste trabalho.

CAPÍTULO IV

4 Monitoramento TCP/IP georreferenciado

Nesse capítulo será abordada a utilização dos conceitos de SIG em uma rede de dados WAN utilizando protocolo TCP/IP.

4.1 Gerência em redes IP

A gerência de redes compreende vários aspectos, entre eles a monitoração e medição dos padrões de qualidade da rede (FERREIRA, 2005).

Considerando a afirmação anterior, monitorar uma rede TCP/IP consiste em acompanhar o estado da rede e principalmente se a mesma está ativa.

Como uma rede implica na existência de vários *hosts*, via de regra, numa rede WAN eles estão situados a uma distância relativamente grande, há de se considerar alguns aspectos durante o processo:

- O *host* monitorado é um servidor?
- O equipamento fica em operação 24 horas, sete dias por semana?
- Há sistema de *backup* na alimentação elétrica do *host*?

Caso alguma das situações indicadas acima não seja atendida, há grande chance de confundir a situação de anormalidade com uma provável falha na rede de comunicação de dados.

Outro aspecto abordado nas redes WAN e também de grande importância é a medição dos padrões de qualidade. Medir uma rede de dados envolve o fornecimento de dados relacionados ao negócio ou atividade fim que o *link* de dados deve atender e estes estão indicados no tipo de contrato firmado com a operadora de telecomunicações responsável pela prestação do serviço.

Objetivando apenas identificar se determinado ponto remoto está ativo, determinar um ponto remoto para medição que atenda aos critérios mínimos e independam das atividades dos usuários e/ou operadores é fundamental.

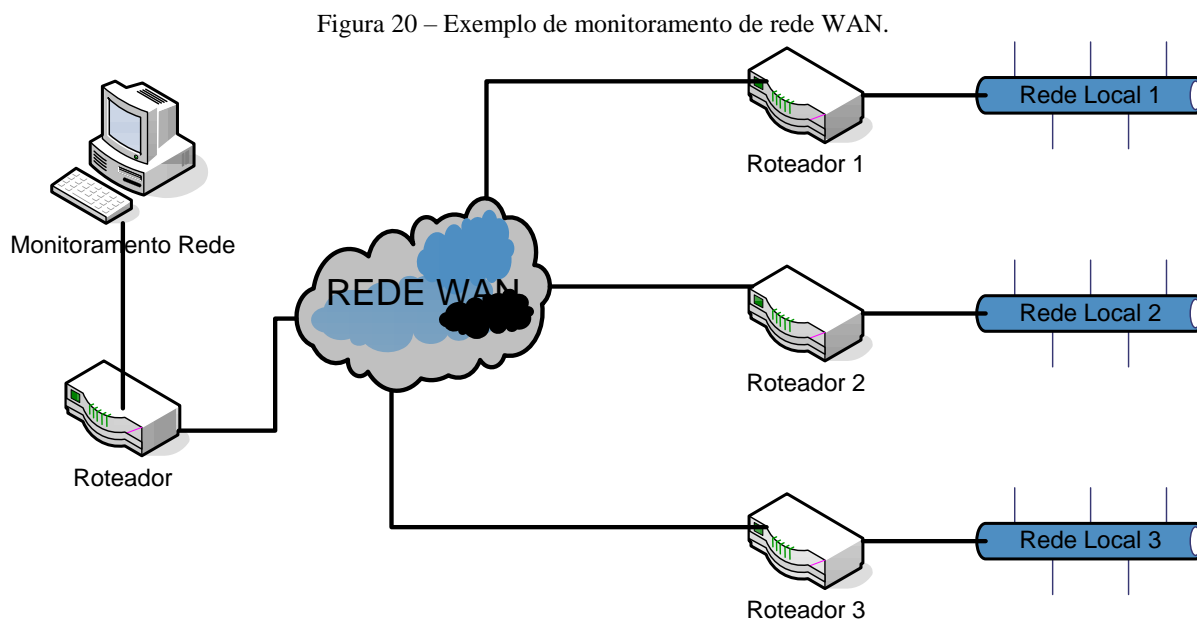
No ato da instalação de um *link* de dados, instalar e configurar o roteador são atividades integrantes do processo. Esse equipamento é responsável por tratar o endereçamento IP da rede WAN e da rede LAN. Fisicamente o roteador é um tipo de ponte responsável por transferir os dados entre as redes.

Dentre as boas práticas, deve ser considerada a possibilidade de instalar nos roteadores um equipamento *no-break* que garanta o fornecimento de energia elétrica por determinado período caso haja interrupção por parte das concessionárias de energia elétrica.

Como o roteador é o primeiro ponto da rede e possui todas as regras para interligar a rede externa à rede interna, o mesmo deve ficar operante em regime 24x7, isto é, 24 horas, sete dias por semana, independentemente da existência de *hosts* na rede interna gerando ou consumindo tráfego na rede.

Partindo dessa condição, o roteador é o candidato natural para verificar se o ponto está em operação e assim iniciar o monitoramento da rede.

Na Figura 20 é possível observar o exemplo de uma situação em que um microcomputador é utilizado para verificar o status dos pontos remotos nos roteadores um, dois e três. Conforme a topologia indicada, em todos os pontos o roteador é o primeiro elemento da rede de dados.



Fonte: (SPOZITO, 2011)

4.2 Georreferenciamento de equipamentos de rede

Habitualmente são utilizados arquivos de texto ou log, onde são armazenadas informações periodicamente quanto ao estado de um ponto remoto (FERREIRA, 2005). A partir da análise periódica desses arquivos um *software* indica na tela de um computador qual equipamento *link* está inoperante.

A contribuição deste trabalho consiste em discorrer sobre as vantagens em relacionar cada roteador a seu ponto geográfico em um mapa, plotando em uma tela de grandes dimensões e alta resolução os pontos cadastrados.

Durante os estudos, duas soluções para representação dos pontos monitorados em uma base georreferenciada foram propostas.

A primeira solução consistia em utilizar a API do *Google Maps* para gerar as imagens. (GOMES, 2008). Utilizando serviços de Internet (*webservices*), as coordenadas são transmitidas aos servidores da Google©, que retornam uma página em HTML com todas as informações impressas no mapa. A solução foi descartada, dada à dependência da Internet na operação do sistema, considerando principalmente o fato que se o acesso à Internet sofrer interrupções, a operação do sistema é prejudicada. Outro ponto relevante na escolha da plataforma foi o tempo necessário para a criação do protótipo. Desenvolver aplicações *desktop*, via de regra, leva menos tempo e há menor dependência dos aplicativos servidores de páginas que suportam sistemas para plataforma WEB.

O modelo escolhido para desenvolver o protótipo, consiste em utilizar bibliotecas Java específicas para tratamento de imagens georreferenciadas. Propõe-se atualizar periodicamente o *status* de cada ponto remoto no mapa plotando na tela seu *status*, com auxílio de bibliotecas Java com acesso à camada de rede. Deste modo, exclui-se a dependência da Internet para operação do sistema.

Utilizando um SGBD com arquitetura ER, os registros obtidos com o monitoramento dos pontos remotos ficarão armazenados e, a partir dos dados será possível extrair informações do comportamento de cada ponto e o tempo médio para resolução dos problemas.

Há possibilidade, também, de gerar um registro em uma tabela auxiliar do que provavelmente causou a parada de determinado ponto remoto. Com o histórico de registros, há chances de conhecer com maior facilidade o fator gerador das paradas.

No capítulo VI é proposto e descrito um diagrama de entidades e relacionamentos com foco nos objetivos propostos.

CAPÍTULO V

5 Estudo de caso.

Nesse capítulo é apresentado o fato gerador para o trabalho desenvolvido.

A aplicação da técnica será de âmbito estadual, sendo testada inicialmente em algumas Escolas Técnicas Estaduais – Etecs e Faculdades de Tecnologia - Fatecs do Centro Paula Souza.

5.1 O Centro Estadual de Educação Tecnológica Paula Souza - CEETEPS

O Centro Paula Souza iniciou suas atividades em seis de outubro de 1969 quando surgiu a necessidade de formação profissional para acompanhar a expansão industrial paulista.

O Centro Paula Souza administra atualmente 186 Escolas Técnicas (Etecs) e 49 Faculdades de Tecnologia (Fatecs) estaduais em 148 municípios no Estado de São Paulo.

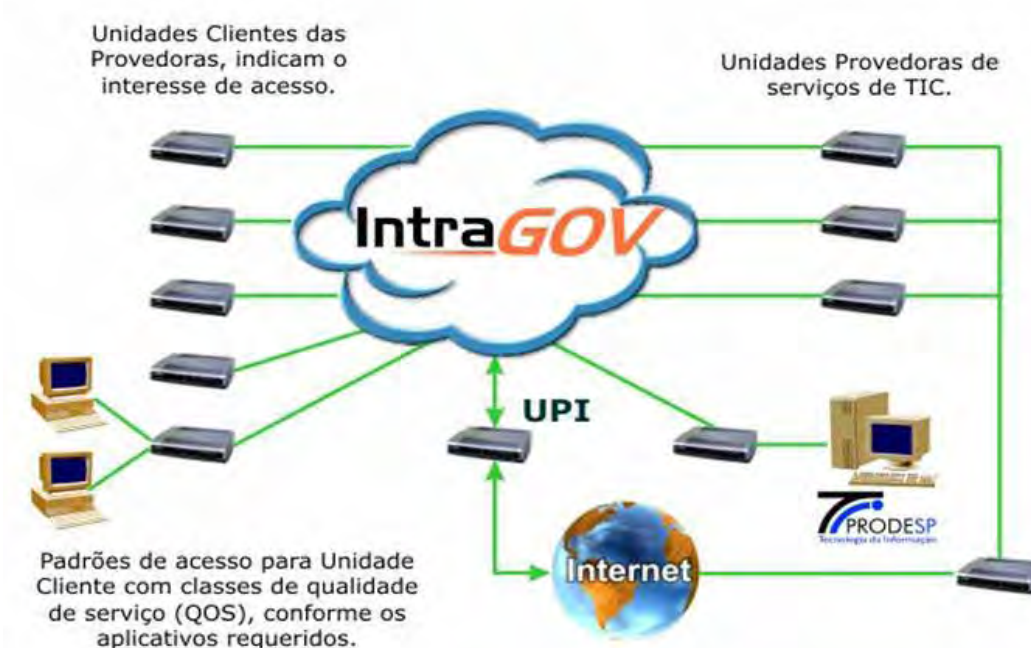
Todas as unidades do CEETEPS espalhadas no estado de SP estão conectadas via sistema Intragov utilizando tecnologia *Multiprotocol Label Switching* – MPLS.

O projeto Intragov surgiu a partir de uma iniciativa do governo do estado de São Paulo objetivando transmitir dados entre várias unidades geograficamente distribuídas. Deste modo foi realizada uma licitação em nível estadual para que uma única empresa pudesse oferecer os serviços com critérios de qualidade previamente estabelecidos independentemente da região do estado a ser interligada.

Com o sucesso do projeto, através da Casa Civil, o governo do estado de São Paulo disponibilizou também às prefeituras e outros órgãos estaduais um meio legal de contratação dentro dos melhores custos obtidos e com padrões de quantidade e qualidade definidos.

Tecnicamente, cabe à operadora de telecomunicações vencedora do processo licitatório buscar meios de interligar unidades públicas distribuídas no estado de São Paulo, sempre atendendo a padrões de quantidade e qualidade definidos em contrato, independentemente da região do estado. Na Figura 21 pode ser observada a estrutura básica proposta pelo projeto Intragov.

Figura 21 – Estrutura da rede Intragov.



Fonte: <http://www.intragov.sp.gov.br/menuprinc/estrutura.html>

Nesse contexto, o Centro Paula Souza optou também por utilizar o projeto Intragov para interligar suas unidades de ensino.

Apesar de exigir, dentro do processo licitatório, os parâmetros mínimos de comportamento da rede de dados, há ocorrência de falhas nos sistemas. Dentro dos requisitos de garantia de qualidade há um acordo de nível de serviço – SLA que a operadora deve cumprir. Devido à abrangência geográfica e ao grande número de unidades, via de regra, as falhas são detectadas quando as unidades clientes, ou as pontas, informam a parada no sistema à Divisão de Informática – DI do Centro Paula Souza.

Este trabalho propõe a modelagem e desenvolvimento de uma ferramenta que permitirá o monitoramento de modo ativo da rede de dados, reduzindo o tempo de parada das unidades, melhorando a qualidade dos serviços administrativos e do processo de ensino, dado o fato que todo o acesso à Internet dentro das unidades de ensino também utiliza o *link* Intragov.

A parada de um *link* ou acesso em determinada unidade ou região será detectada a partir da visualização em um mapa do estado de São Paulo, como na Figura 22. A partir dessa

informação os técnicos poderão acompanhar junto à operadora e à unidade de ensino o processo de resolução do problema.

Figura 22– *Link* Intragov Estado SP.



Fonte: <http://www.intragov.sp.gov.br>

5.2 Problemas e Hipóteses

A interligação das unidades escolares do Centro Paula Souza depende de uma rede única denominada Intragov. Toda a instalação, desde a parte física (cabos e fios), até a parte lógica (configuração de roteadores e regras de roteamento) depende de empresa terceirizada.

Do anteriormente exposto surgem os problemas:

- Como acompanhar de modo ativo os *links* de dados de aproximadamente 220 unidades?
- Qual a melhor maneira de fazer cumprir as exigências de SLA por parte da operadora contratada?

As hipóteses para o tratamento dos problemas são:

- A quantidade de colaboradores é pequena diante do tamanho e complexidade da rede e tecnologias envolvidas;

- Não há ferramenta de gestão no centro de informática para o monitoramento dos serviços contratados;

5.3 Tipo de pesquisa

A pesquisa proposta neste trabalho é do tipo experimental. Foi modelado e desenvolvido um *software* dotado de uma base de dados relacional e realizados ensaios em unidades escolares para verificar e comparar os resultados obtidos.

5.4 Materiais

- Microcomputadores e impressoras;
- Sistema Operacional Windows XP;
- Sistema Operacional Windows 7;
- IDE NetBeans versão 6.8;
- Pacote JDK Sun versão 1.6.0_11;
- SGBD MySQL versão 5.1.37;
- Ferramenta de gestão para SGBD MySQL Workbench versão 5.2.25 CE;
- Biblioteca GeoTools;
- Acesso a topologia lógica da rede Intragov do centro Paula Souza.

5.5 Metodologia proposta

A problemática inicial apresentou-se na escolha e definição das ferramentas e estudo dos conceitos de rede TCP/IP e SIG.

5.6 Descrição das ferramentas

Aqui são descritas as características que justificam a escolha das ferramentas utilizadas no trabalho.

Durante a escolha das ferramentas e com a premissa de utilizar, sempre que possível, *software* livre (sem a necessidade de pagamento de direitos de uso), foram selecionados os seguintes componentes:

- Sistema operacional Windows 7;
- Linguagem Java;
- Biblioteca GeoTools;
- Sistema Gerenciador de Banco de Dados – SGBD – MySQL.

Com exceção do *software* do sistema operacional Windows escolhido como plataforma, os outros componentes são distribuídos gratuitamente.

5.6.1 Sistemas Operacionais

No início do estudo e objetivando tornar a ferramenta disponível a todos os usuários, sem a necessidade de pagar por licenças de *softwares*, o sistema operacional Linux foi a primeira escolha. A distribuição Slackware versão 12.0 foi utilizada como modelo inicial.

Durante essa etapa houve dificuldade em configurar as outras ferramentas no ambiente.

Considerando que o desenvolvimento é realizado em modo gráfico, dado principalmente o foco da aplicação, foi necessário habilitar a interface gráfica KDE.

Apesar da instalação relativamente fácil da distribuição, instalar e configurar o SGBD gerou várias dificuldades dentro do sistema operacional Linux. Variáveis de ambiente devem ser configuradas em modo texto e arquivos de inicialização precisam ser editados de modo a permitir que o serviço inicialize automaticamente quando o computador é ligado.

Numa segunda etapa o ambiente de desenvolvimento (*Integrated Development Environment* – IDE, ou ambiente de desenvolvimento integrado) NetBeans foi configurado, gerando outros problemas. Há necessidade de apontar nas variáveis de ambiente a Máquina Virtual Java, responsável por interpretar e executar os *softwares* gerados no ambiente.

Por fim, foi carregada a biblioteca GeoTools que permite trabalhar os conceitos de georreferenciamento no desenvolvimento dos aplicativos Java.

Considerando os aspectos iniciais de tornar a ferramenta de monitoramento simples e de fácil acesso, foi reconsiderada a utilização dos sistemas operacionais Windows XP e Windows 7. Mesmo tratando-se de um sistema operacional comercial, o tempo despendido na configuração do

ambiente Linux e os conhecimentos técnicos necessários para essa tarefa, justifica o investimento.

Foi considerada a possibilidade de utilizar outra distribuição LINUX, como o Ubuntu, que possui interface gráfica e instalação mais amigáveis, mas o fator decisivo na escolha da plataforma Windows deu-se na fato de a grande maioria dos usuários de computador estar habituado ao ambiente gráfico utilizado no sistema da Microsoft®, o que facilita o uso da ferramenta.

No ambiente Windows todas as etapas de configuração do SGBD MySQL, IDE NetBeans e carga da biblioteca de geoprocessamento ocorreram de modo simples, sem a necessidade de acessar linhas de comando para ajustes. Ao reinicializar o microcomputador, o SGDB foi automaticamente carregado, a IDE estava com acesso ao diretório padrão da máquina virtual Java e a biblioteca de desenvolvimento foi carregada sem dificuldade.

Diante do exposto, optou-se por utilizar o Windows 7 como base no desenvolvimento e Windows XP no ambiente de monitoramento, mantendo todas as outras ferramentas dentro dos conceitos de *software* livre. Em testes realizados o monitoramento ocorreu sem problemas também na plataforma Windows 7 (64 bits), o que amplia o tempo de vida do sistema.

5.6.2 A Linguagem Java.

A linguagem Java, representada pelo ícone da Figura 24, foi desenvolvida por uma equipe da *Sun Microsystems* (DEITEL; DEITEL, 2003) e inicialmente visava projetos de *software* para produtos eletrônicos. As exigências geradas pela rápida atualização dos produtos requisitaram a criação de uma linguagem específica para uso em processadores de aparelhos domésticos que se caracterizasse pela simplicidade, reuso, e possibilidade multi-plataforma.

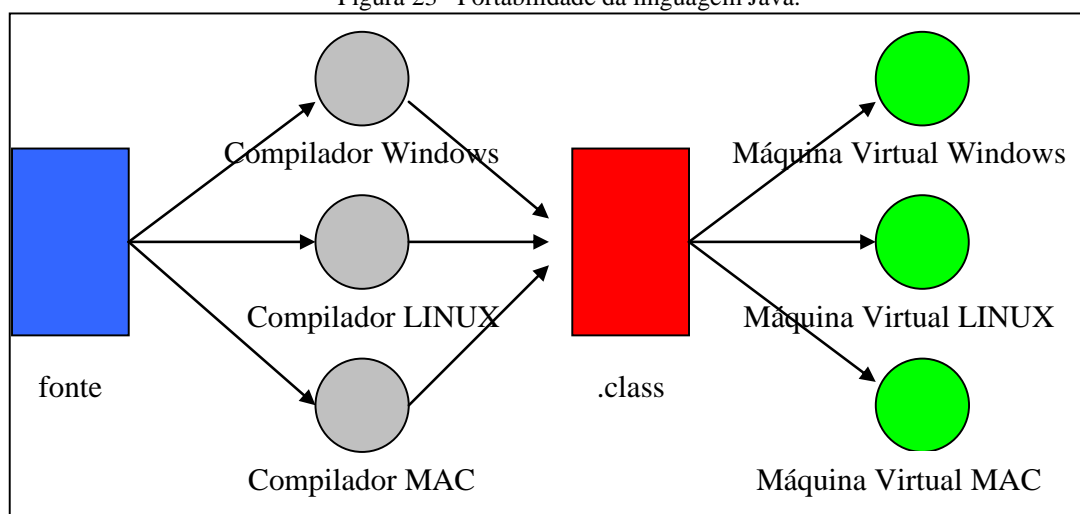
No início de 1990, a *Sun Microsystems*, financiou uma pesquisa corporativa interna com o codinome Green (DEITEL; DEITEL, 2003). James Gosling definiu as bases para o projeto dessa nova linguagem de programação para suprir esses requisitos. Segundo (DEITEL; DEITEL, 2003) a especificação da linguagem terminou em agosto de 1991, e a ela deu-se o nome de "Oak" (Carvalho). Por problemas de copyright (já existia uma linguagem chamada Oak) o nome foi mudado em 1995 para Java, em homenagem à ilha de Java, de onde vinha o café consumido pela equipe da *Sun Microsystems* em uma cafeteria local.

Java é uma linguagem de alto nível que carrega consigo diversas características herdadas de outras linguagens. É simples, muito parecida com C++, é orientada a objetos, pois a maior parte dos elementos de um programa Java são objetos. Seu código é organizado em classes. (DEITEL; DEITEL, 2003).

Caracteriza-se, também, por suportar:

- *Multithreading* – Conceito multitarefa embutido na linguagem: um programa Java pode possuir mais de uma linha em execução (*thread*).
- Exceções – Erros em programas Java são tratados também em tempo de execução, já que a máquina virtual Java faz uma verificação quanto aos acessos de memória, abertura de arquivos e uma série de eventos que podem gerar problemas computacionais graves em outras linguagens, mas que geram exceções em programas Java.
- *Garbage collector* - Conceito no qual se define verdadeira coleta de lixo, varrendo a memória de tempos em tempos, liberando automaticamente os blocos que não estão sendo utilizados.
- Portabilidade - Os programas são compilados independentes da plataforma (Figura 23). Um sistema desenvolvido para ambiente Windows poderá ser executado em ambientes LINUX e MACINTOSH. Isso ocorre devido ao código compilado resultar em arquivos com extensão *.class* (*dot class*) que são interpretados e executados por uma máquina virtual.

Figura 23– Portabilidade da linguagem Java.



Fonte: (SPOZITO, 2011)

- Gratuidade - Há uma ampla biblioteca de componentes e recursos disponíveis gratuitamente.

Figura 24– Ícone representativo da linguagem Java.



Fonte: <http://www.sun.com>

- Praticidade - O ambiente de desenvolvimento disponibilizado pela Sun, o NetBeans é uma ferramenta integrada de desenvolvimento – IDE, extremamente eficiente, confiável e gratuita.
- Orientada a objetos – É organizada em classes, isto é, possui um padrão definido para criação de variáveis de modo estruturado que permite a utilização da mesma em vários pontos do *software*, evitando o retrabalho, atendendo aos padrões atuais de engenharia de *software* (PRESSMAN, 1995; BOOCH; RUMBAUGH; JACOBSON, 2000) como a UML (Figura 25).

Figura 25– Simbologia universal da notação UML.



Fonte: <http://www.uml.org>

5.6.3 A IDE NetBeans

O Ambiente de Desenvolvimento Integrado (IDE - *integrated development environment*) utilizado para codificação do protótipo em linguagem Java pode ser utilizado gratuitamente,

bastando apenas realizar o *download* do arquivo de instalação no sítio da Internet do desenvolvedor.

A tela de inicialização da IDE pode ser observada na Figura 26. O uso de um ambiente de desenvolvimento facilita e agiliza o processo de codificação de sistemas computacionais. Há outras ferramentas de desenvolvimento para a linguagem Java, algumas com exigências de pagamento pelo direito de uso.

Há, também, dentro do ambiente de desenvolvimento, funções como engenharia reversa ou automatização na criação da documentação e diagramas de classes dentro dos padrões UML (GONÇALVEZ, 2006).

Figura 26– Tela de abertura da IDE NetBeans.



Fonte: <http://www.netbeans.org>

5.6.4 O SGBD MySQL

O Sistema de Gerenciamento de Banco de Dados – SGBD, MySQL foi escolhido, por possuir algumas características semelhantes à linguagem Java:

- Portabilidade - suporta praticamente qualquer plataforma atual;
- Disponível gratuitamente na versão CE;
- Ótimo desempenho e estabilidade;

- Facilidade de uso e implementação.

O ícone padrão que representa o SGBD MySQL pode ser visualizado na Figura 27.

A Agência Espacial Norte Americana – NASA³ optou por migrar parte de seus sistemas do SGBD Oracle para MySQL justificando, para isso, benefícios como redução do custo efetivo total, facilidade no suporte dada pela grande comunidade de desenvolvedores da plataforma e compatibilidade com os sistemas que utilizam padrão Java.

Outro fator que tornou relevante a escolha do SGBD para a modelagem do banco de dados proposto, foi a disponibilização gratuita da ferramenta de modelagem e engenharia MySQLWorkbench (Figura 28)

Utilizando o Workbench, é possível realizar toda a modelagem do banco de dados em ambiente gráfico, utilizando conectores e linhas para tratar os relacionamentos e gerar automaticamente um *script* SQL que, ao ser executado, cria todas as tabelas dentro do SGBD.

Figura 27– Logotipo do SGBD MySQL.



Fonte: <http://www.mysql.org>

³Disponível em <http://www.mysql.com/news-and-events/generate-article.php?type=ss&id=nasa>

Figura 28 – Logotipo MySQL Workbench.



Fonte: <http://www.mysql.org>

5.6.5 A biblioteca GeoTools

A biblioteca GeoTools (Figura 29) consiste em um grupo de programas ou Classes desenvolvidas para tratamentos de dados geoespaciais. A biblioteca pode ser obtida gratuitamente através do sitio da Internet <http://www.geotools.org>.

Figura 29 – Logotipo da biblioteca GeoTools (*Open Source Geospatial Foundation*).

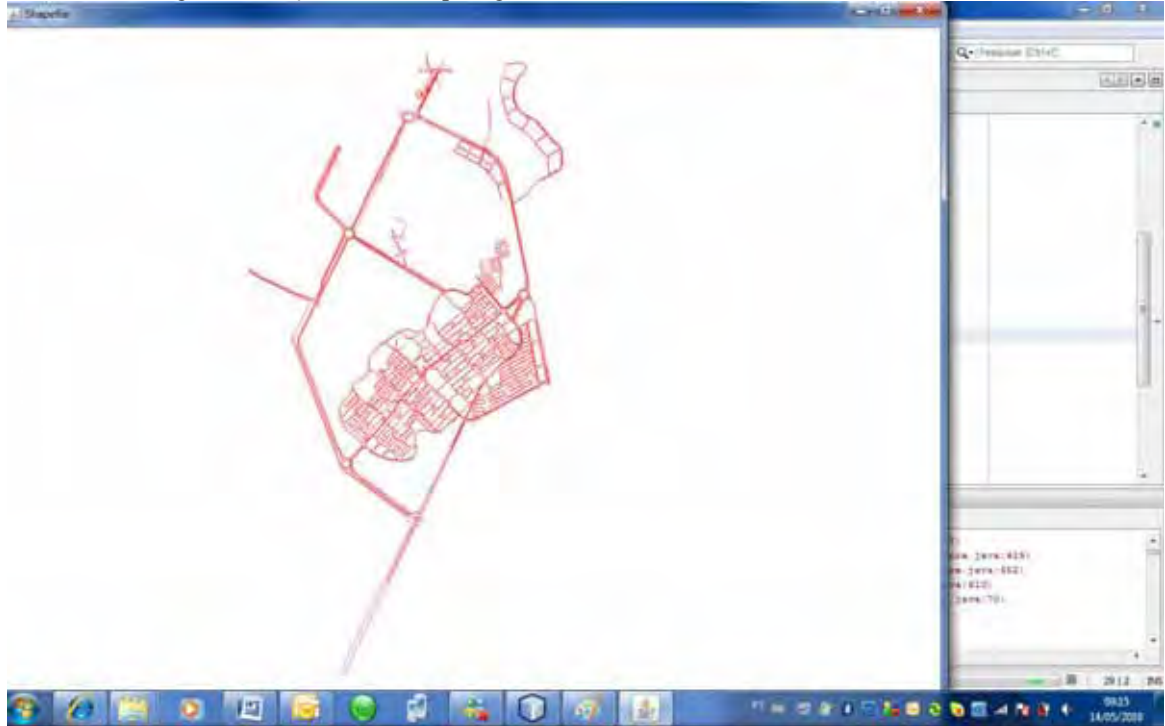


Fonte: <http://geotools.org/about.html>

Integrantes da fundação OSGeo codificam e disponibilizam gratuitamente classes em linguagem Java utilizando as melhores técnicas de manipulação de dados para SIG.

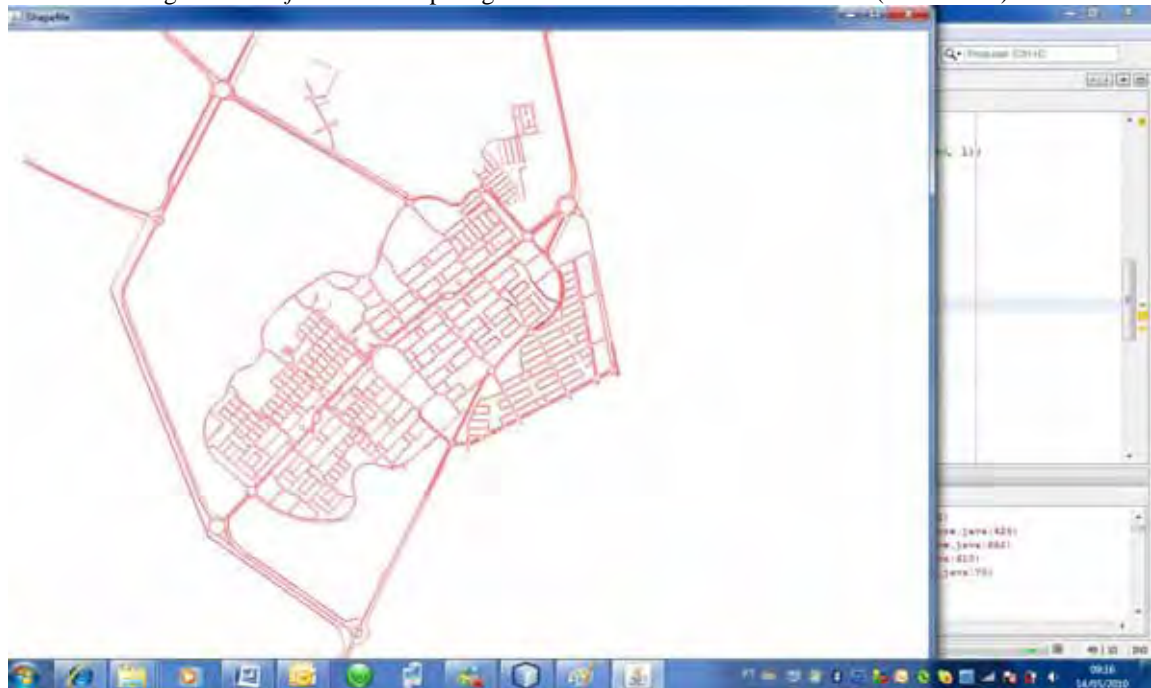
O comportamento da biblioteca foi analisado com o desenvolvimento de uma aplicação em Java (ANEXO – I) utilizando um mapa da cidade de Ilha Solteira em formato vetorizado.

Figura 31– *Software* com a plotagem da malha urbana de Ilha Solteira (ZOOM=5).



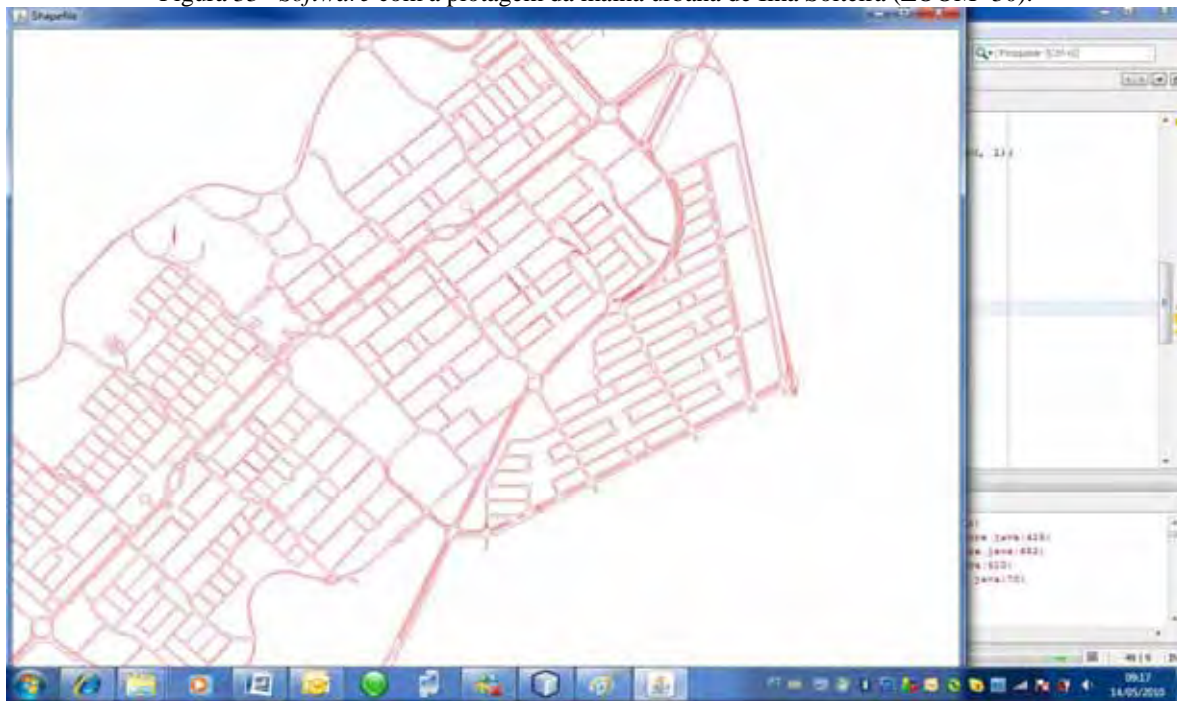
Fonte: (SPOZITO, 2011)

Figura 32– *Software* com a plotagem da malha urbana de Ilha Solteira (ZOOM=10).



Fonte: (SPOZITO, 2011)

Figura 33– Software com a plotagem da malha urbana de Ilha Solteira (ZOOM=30).



Fonte: (SPOZITO, 2011)

Como pode ser observado nas figuras anteriores, a biblioteca GeoTools possui classes JAVA e métodos desenvolvidos em conformidade com os padrões de arquivos *shapefile* (.shp), tornando mais rápido o desenvolvimento das aplicações GIS. O reaproveitamento e padronização de códigos é uma dos principais objetivos do padrão UML, que possibilita utilizar bibliotecas não nativas ao pacote disponibilizado pela SUN©, de modo prático e de entendimento relativamente simples.

Além das bibliotecas JAVA, é possível obter gratuitamente na Internet a outras referências da biblioteca GeoTools, como toda a documentação, o guia de usuário e os arquivos “javadocs”.

No capítulo seguinte, são ilustradas partes dos códigos fonte desenvolvidos na aplicação e a utilização de um pacote para tratar dados geográficos utilizados no sistema é descrito em detalhes.

CAPÍTULO VI

6 Sistema GeoMon_Ipv4

Neste capítulo, é apresentado o sistema desenvolvido para tratar o problema apresentado.

Desenvolvido utilizando conceitos de monitoramento de redes TCP/IP em ambientes geográficos, o *software* foi batizado de GeoMon_Ipv4, sendo “Geo” associado a geográfico, “Mon” referente a monitoramento e Ipv4 sigla relacionada a padronização versão 4 do protocolo TCP/IP.

Durante a programação, optou-se por utilizar um campo do tipo *String* para armazenar os endereços de rede, o que a princípio permite tratar endereços IPv4 e IPv6. Considerando que a rede Intragov no Centro Paula Souza utiliza o IPv4, nos locais de ensaio não foram executados testes no novo padrão TCP/IP, o que impossibilitou avaliar o comportamento do sistema no padrão IPv6.

6.1 Modelagem do banco de dados

A primeira etapa no desenvolvimento do sistema foi a modelagem do banco de dados responsável por armazenar os dados que serão tratados pelo sistema.

O banco de dados foi chamado de ‘intragov’.

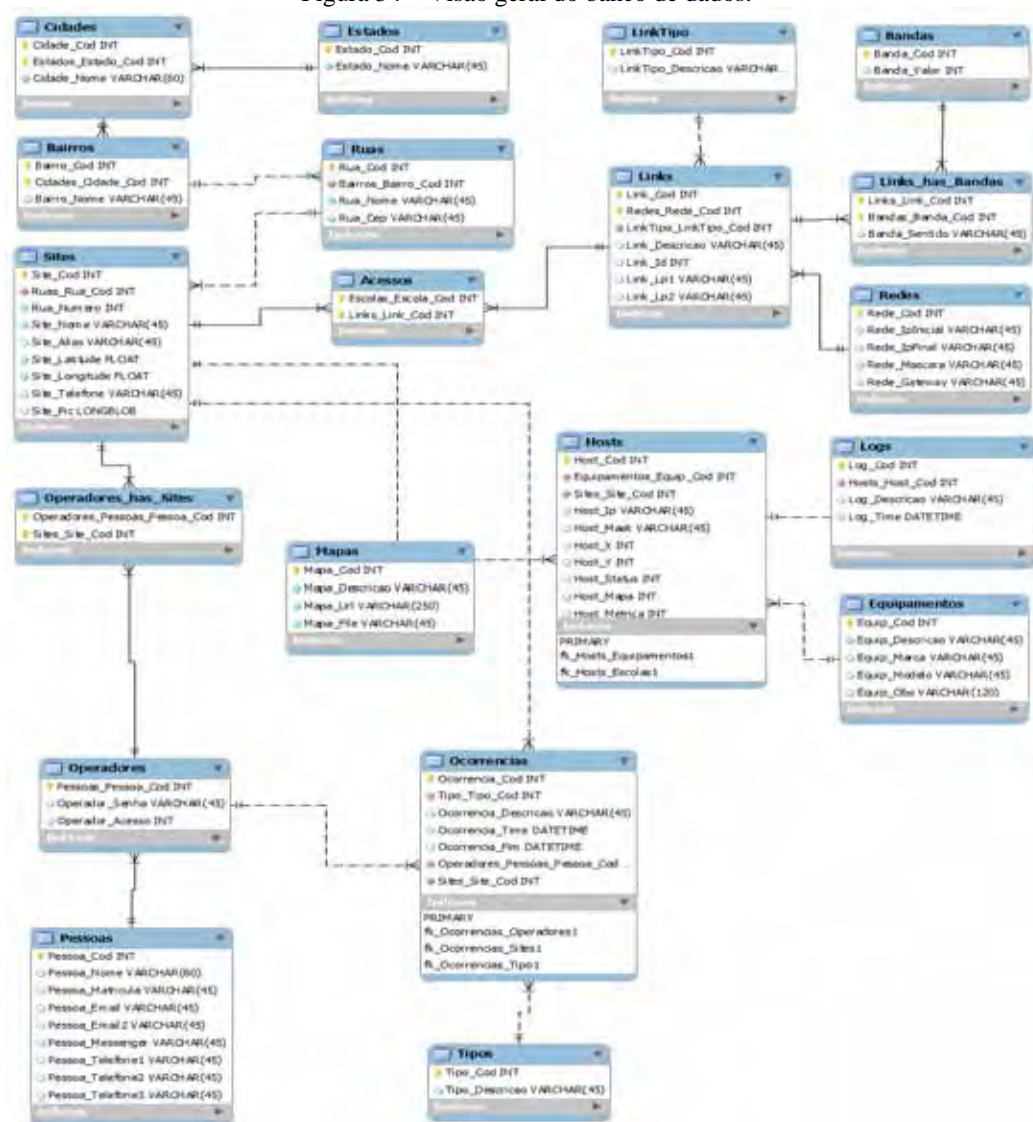
Foram criadas 20 tabelas considerando quatro blocos:

1. Bloco de georreferenciamento;
2. Bloco de caracterização de tecnologias e topologias das redes TCP/IP ;
3. Bloco de equipamentos de rede (*hosts*) e histórico de estado (logs);
4. Bloco de operadores e ocorrências.

Na Figura 34 é possível observar, de forma geral, as tabelas e os relacionamentos entre as mesmas.

Durante o processo de modelagem, foram realizadas visitas e coletadas opiniões dos gestores do sistema Intragov. Foi sugerida a inserção de mais um bloco para listar em cada local os cursos ministrados, tipos de equipamentos, quantidade de equipamentos e usuários. Essa informação possibilitaria comparar e justificar a banda instalada no local e a importância da qualidade de serviço quanto ao tempo de disponibilidade do *link*. A sugestão será indicada como trabalho futuro.

Figura 34 – Visão geral do banco de dados.



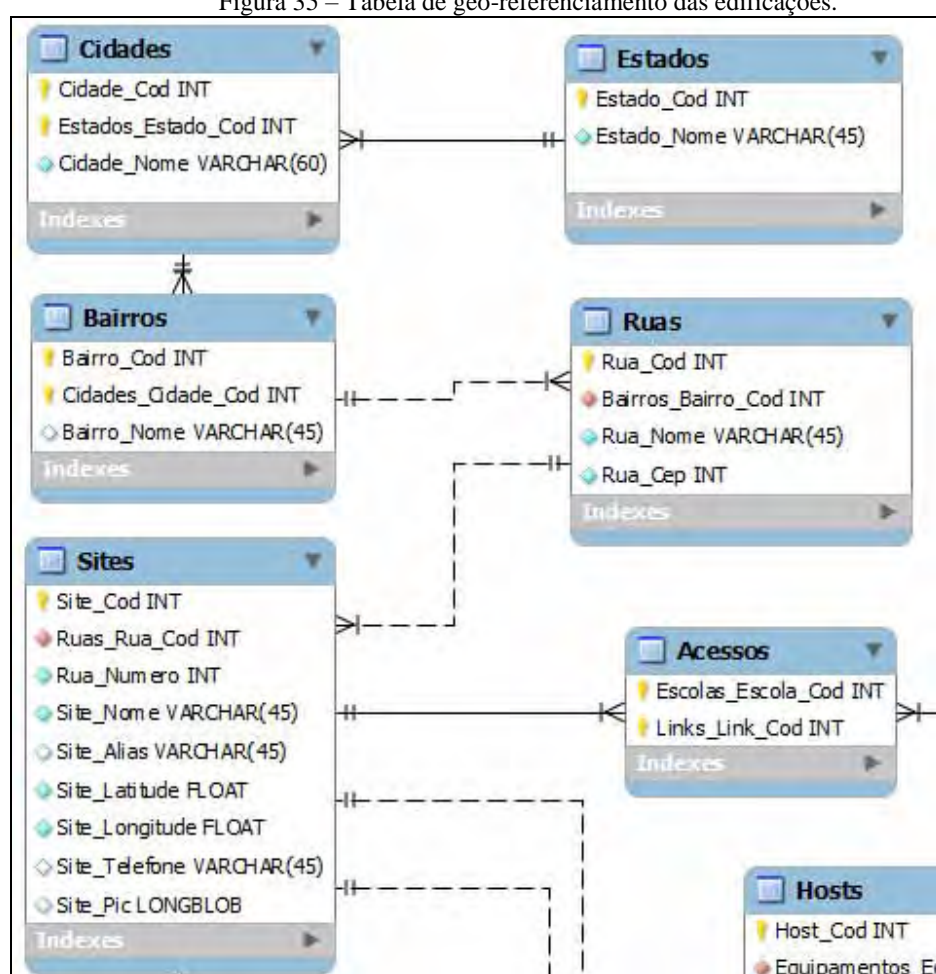
Fonte: (SPOZITO, 2011)

6.1.1 Bloco de georreferenciamento

As tabelas utilizadas neste bloco têm como objetivo associar uma ou mais edificações a uma coordenada geográfica.

Como indicado na Figura 35, na tabela “Sites” foram reservados campos do tipo *Float* para armazenar os valores de latitude e longitude para cada ponto.

Figura 35 – Tabela de geo-referenciamento das edificações.



Fonte: (SPOZITO, 2011)

Foram criadas e relacionadas, também, as tabelas Estados, Cidades, Bairros e Ruas.

Observa-se a dependência criada entre as tabelas com a utilização de chaves estrangeiras, isto é, a inserção de um novo registro na tabela Cidades, depende obrigatoriamente de um registro

da tabela Estados. Essa técnica foi utilizada, pois em termos geográficos, uma cidade obrigatoriamente pertence a algum estado da federação.

A mesma técnica foi aplicada para o cadastramento de bairros e ruas. Todo bairro pertence a uma cidade e toda rua está contida em determinado bairro.

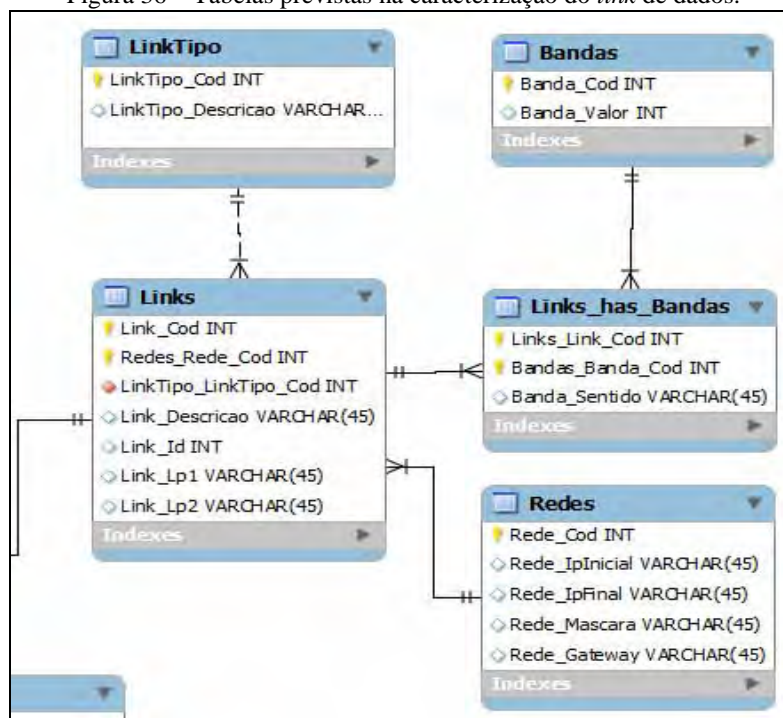
As tabelas foram criadas para facilitar e agilizar a busca de informações pelos operadores. Apesar de existirem os campos latitude e longitude para cada *site*, a busca de informações específicas, mas de uso cotidiano, exigiria a utilização de uma base de dados georreferenciada de grande porte com a necessidade constante de atualização, o que inviabilizaria o sistema projetado para aplicações gerais.

6.1.2 Bloco de caracterização de tecnologias e topologias das redes TCP/IP.

Este conjunto de tabelas tem como objetivo definir a tecnologia e parâmetros utilizados para conectar a edificação à rede WAN da instituição.

Na Figura 36 é possível visualizar as tabelas previstas.

Figura 36 – Tabelas previstas na caracterização do *link* de dados.



Fonte: (SPOZITO, 2011)

A tabela Links foi o ponto de partida para a criação de outras tabelas auxiliares, considerando os estudos realizados nos capítulos anteriores.

Considera-se o fato que, para disponibilizar acesso a um ponto remoto, há necessidade da instalação de um ou mais *links* de acesso.

Como o projeto prevê a distribuição de redes WAN, isto é, com abrangência geográfica relativamente grande, chegando a centenas de quilômetros, nem todos os locais possuem os mesmos recursos disponibilizados pelas operadoras de telecomunicações.

Locais urbanos com maior densidade demográfica e maior renda possuem tecnologias mais novas disponíveis. Áreas com pequenas populações ou distantes dos grandes centros comerciais e industriais demandam um período maior para instalação de cabos ou sistemas de comunicação de alta tecnologia. Tal característica deve-se ao alto custo dos equipamentos e mão de obra envolvida, na qual a demanda por consumo deve justificar o investimento.

Deste modo o sistema foi projetado para qualificar e quantificar os *links* que serão utilizados pelos pontos remotos:

A tabela LinkTipo, tem como objetivo armazenar a tecnologia ou meio utilizado pelo *link*. Nesse campo pode ser indicada a utilização de um meio óptico com tecnologia FDDI ou MPLS sobre par metálico.

Quando é contratado um *link* de dados, é indicado à operadora de telecomunicações, a banda ou velocidade que esse *link* de dados deve possuir. Há no mercado bandas ou “velocidades” padrão, geralmente associadas à tecnologia envolvida. Na tabela Bandas são inseridos os registros com as larguras de banda padrão.

A tabela Links_has_Bandas associa a um *link* de dados qual a largura de banda contratada e em qual sentido. Há possibilidade de contratar junto às operadoras velocidades diferentes para os fluxos de dados nos dois sentidos, saída e entrada ou mais conhecidos como *upload* (carga ou fluxo de subida) e *download* (carga ou fluxo de descida).

Como o sistema trata a interconexão de pontos remotos dentro de uma única rede, há necessidade de indicar as características de dados para cada *link*. Sempre que um *link* de dados é ativado, a operadora de telecomunicações fornece ao usuário uma faixa específica em uma rede do tipo IPv4. Para uma rede com acesso a Internet, cabe ao comitê gestor da Internet no Brasil – CGI.br distribuir e gerenciar os IP's disponíveis às operadoras e as mesmas distribuírem entre seus clientes conforme a necessidade. Há possibilidade também de configurar uma rede com as

características exigidas pelo cliente, nessa situação o contratante indicará para a operadora as características de cada *link* em cada local.

Uma tabela com características importantes pode ser observada na Figura 37. A tabela Acessos tem como objetivo associar uma unidade a um *link*. A mesma é formada por uma chave composta. Essa metodologia garante a integridade do relacionamento possibilitando associar um *link* contratado a apenas um local, dado o fato de nunca ocorrer a situação de um mesmo *link* atender a mais de um ponto.

Figura 37 – Tabela Acessos.



Fonte: (SPOZITO, 2011)

6.1.3 Bloco de equipamentos de rede (*hosts*) e histórico de *status* (*logs*);

Considerando que nos blocos anteriores relacionaram os locais físicos (com suas características geográficas) e os recursos de comunicação que atenderão cada ponto, há necessidade de um mecanismo para monitorar e registrar o status dos recursos contratados das operadoras que interligam as unidades.

As redes geograficamente distribuídas, embora haja exceções, utilizam sistemas ponto a ponto (TANEMBAUM, 1997). Dentro desta topologia o primeiro equipamento da rede, responsável por conectar a rede local à rede WAN é o roteador. Os roteadores são considerados *hosts* e operam na camada três do modelo TCP/IP.

Na Figura 38 podem ser observadas as três tabelas principais projetadas para cadastro e monitoramento contínuo.

A tabela Equipamentos registra informações como a marca, modelo e outros detalhes dos equipamentos que podem ser *hosts* de início da rede, ou *gateways*.

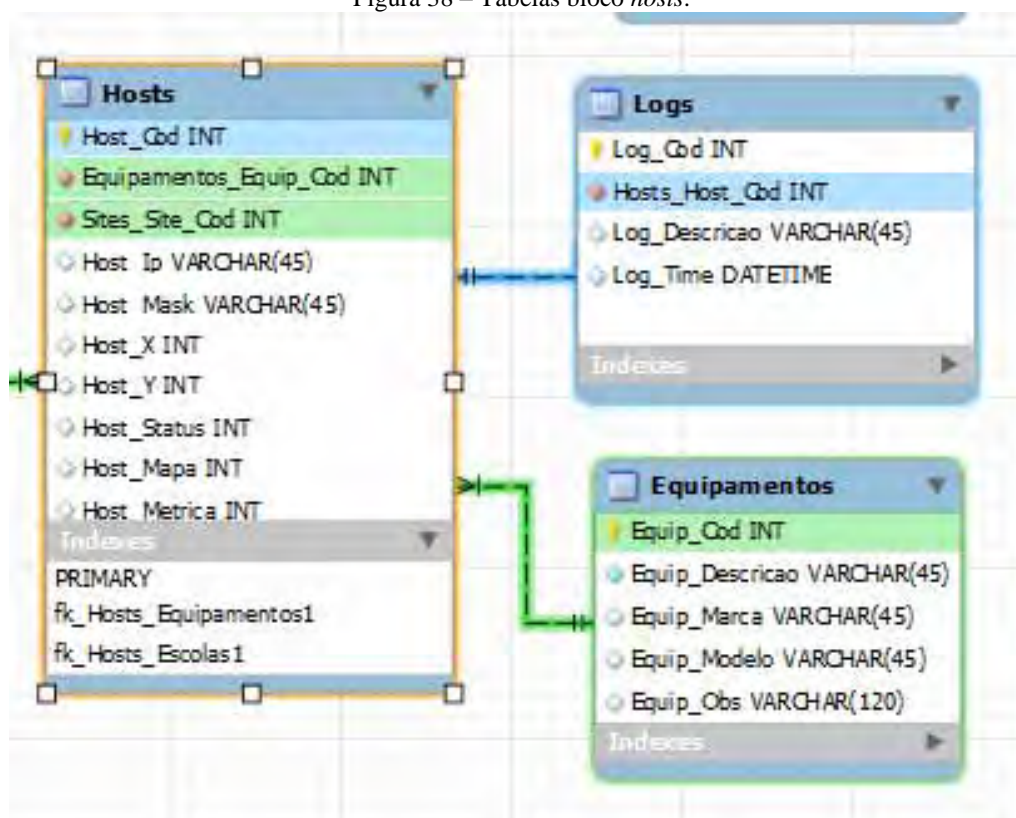
O equipamento que fica instalado no ponto atendido é registrado na tabela Hosts e informações como endereçamento IP também são inseridos nessa tabela. Há uma diversidade de

equipamentos que atuam como roteadores, deste modo o tipo de equipamento é relacionado à tabela Equipamentos.

Outra tabela fundamental na operação do sistema é chamada Logs. Periodicamente o *software* de monitoramento faz uma “varredura” em todos os *hosts* cadastrados na tabela Hosts, verifica sua disponibilidade, coleta informações referentes ao tempo de resposta e salva esse dados na tabela Logs, gerando um histórico do *status* de cada ponto.

Durante o processamento e consulta do status de cada *host*, o sistema atualiza a variável Host_Status INT com o tempo de resposta em milissegundos, obtido de um comando PING disparado. Esse dado é a principal métrica utilizada para determinar a qualidade da comunicação entre os pontos.

Figura 38 – Tabelas bloco *hosts*.



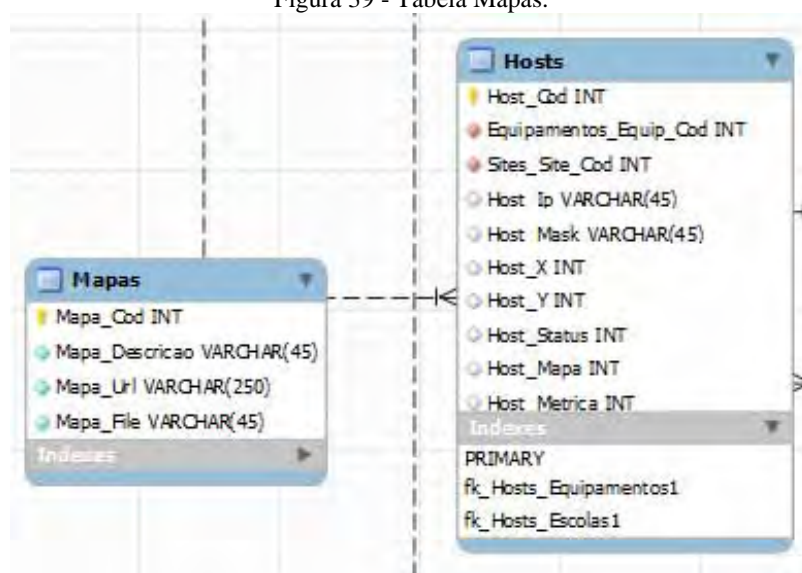
Fonte: (SPOZITO, 2011)

Com o propósito de monitorar vários pontos em diferentes áreas simultaneamente, há necessidade de associar um *host* ao mapa em que o mesmo deve ser plotado. Essa técnica permite atender às condições de generalização cartográfica expostas anteriormente.

Relacionar individualmente cada *host* a um mapa específico, possibilitou durante o desenvolvimento do sistema, plotar a condição específica de cada ponto relativo (*host*), sobre a base georreferenciada.

A Figura 39 ilustra os campos de relacionamento entre os *hosts* e os mapas disponibilizados. O arquivo no formato *shape* é armazenado em um repositório, na qual durante o processo de inserção, o operador aponta o caminho do arquivo, ficando essa informação armazenada no campo Mapa_Url, bem como o nome do arquivo fica armazenado em Mapa_File. Ambos os campos do tipo VARCHAR.

Figura 39 - Tabela Mapas.



Fonte: (SPOZITO, 2011)

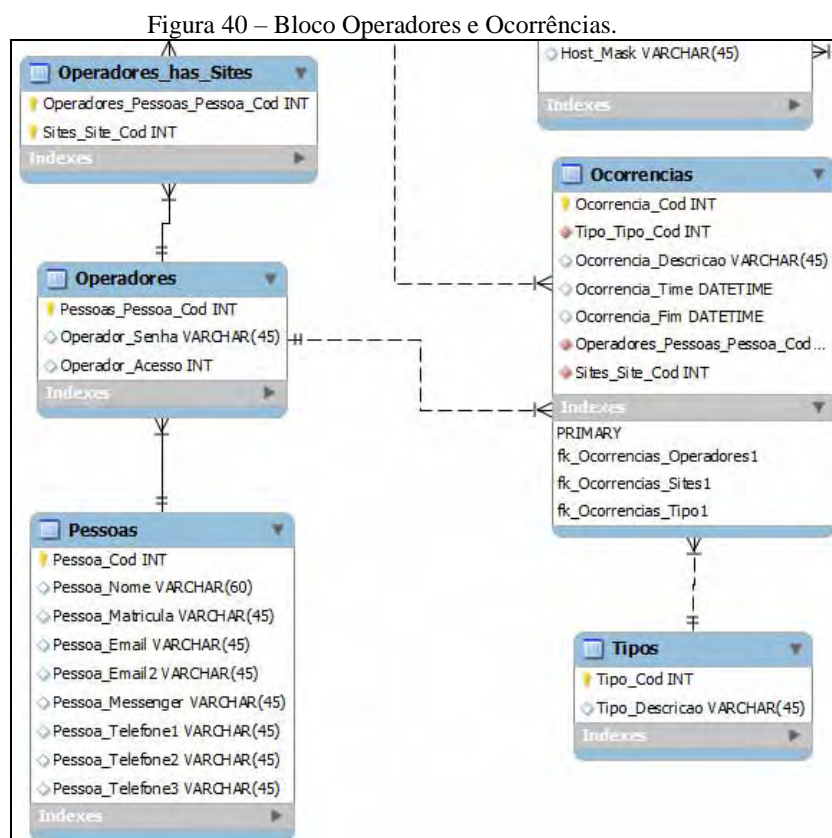
6.1.4 Bloco de operadores e ocorrências.

No processo de análise das necessidades de controle e ativação de *links* de dados, foi constatada a necessidade de controlar as ordens de serviço expedidas às operadoras de telecomunicações.

Por tratar-se de um processo dinâmico, há também pedidos de ampliação de banda e habilitação de novos postos de comunicação.

Diante da necessidade de controlar as situações descritas anteriormente, a tabela Ocorrencias registrará informações relevantes relacionadas a um ponto geográfico específico.

Relacionada à tabela Ocorrências foi implementada a tabela Tipos, objetivando facilitar a classificação da ocorrência, como por exemplo: ativação, manutenção, ampliação ou cancelamento. A Figura 40 exibe a disposição das tabelas tratadas neste bloco.



Fonte: (SPOZITO, 2011)

Outro fator previsto a ser tratado pelo sistema é o controle da operação. A tabela Pessoas registrará todos os dados pessoais dos usuários. Todos os registros inseridos na tabela Pessoas são candidatos a operadores do sistema. Para eleger um operador é necessário inserir na tabela Operadores uma senha e nível de acesso. As garantias de acesso ao sistema serão tratadas no código de programação da interface gráfica.

A tabela Operadores_has_Sites relaciona os locais físicos que os operadores têm acesso.

6.2 Modelagem do sistema e codificação Java.

Na terceira etapa do trabalho foi planejado e codificado um sistema para operacionalização da proposta.

Partindo do planejamento do banco de dados relacional e dos estudos realizados, uma interface de tela foi projetada dividindo as tarefas necessárias à execução do sistema em blocos do seguinte modo:

1. Bloco Principal.
2. Bloco tabelas geográficas.
3. Bloco tabelas redes.
4. Bloco tabelas equipamentos.
5. Bloco tabelas pessoas.
6. Bloco monitoramento.

Ao executar o sistema, a tela principal (Figura 41) é apresentada ao operador, disponibilizando as operações no formato de menus e submenus as operações.

Os blocos tabelas geográficas, tabelas redes, tabelas equipamentos e tabelas pessoas referem-se aos dados que devem ser cadastrados no sistema possibilitando a operação.

Para visualizar em mapa o status dos diferentes pontos distribuídos geograficamente é preciso utilizar o bloco principal.

Objetivando registrar o histórico dos diferentes pontos foi implementado o bloco monitoramento. As opções disponibilizadas referem-se a atualização constante e análise do histórico.

Figura 41 - Tela Principal do Sistema GeoMonIpv4.



Fonte: (SPOZITO, 2011)

A descrição dos blocos pode ser observada nos tópicos abaixo. Essas foram organizadas em ordem diferente da exibida na Figura 41, possibilitando melhor visualização da construção do sistema a partir das bases de dados.

6.2.1 Bloco tabelas geográficas.

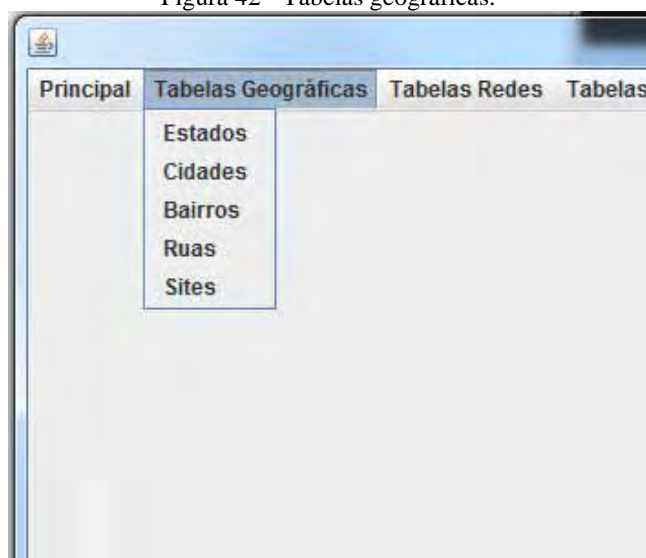
Os primeiros dados que devem ser inseridos no sistema pelo operador são as tabelas geográficas. As inserções dos pontos que serão monitorados dependem de várias tabelas auxiliares.

Todo o processo de cadastramento segue uma ordem hierárquica, como pode ser observado na Figura 42.

O sistema foi projetado para operar em âmbito nacional, sendo o primeiro nível de cadastramento os estados da federação.

As operações previstas nos módulos são inserção, edição e exclusão. A utilização de um SGBD para armazenar os dados, permite garantir a integridade das informações, impedindo operações que causem inconsistência de informações.

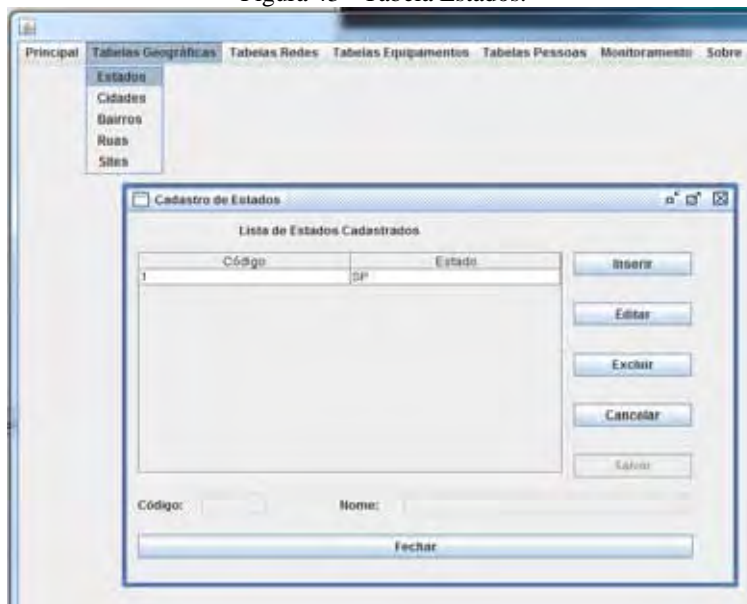
Figura 42 - Tabelas geográficas.



Fonte: (SPOZITO, 2011)

Utilizando a opção Estados, será apresentada ao usuário a tela da Figura 43.

Figura 43 - Tabela Estados.

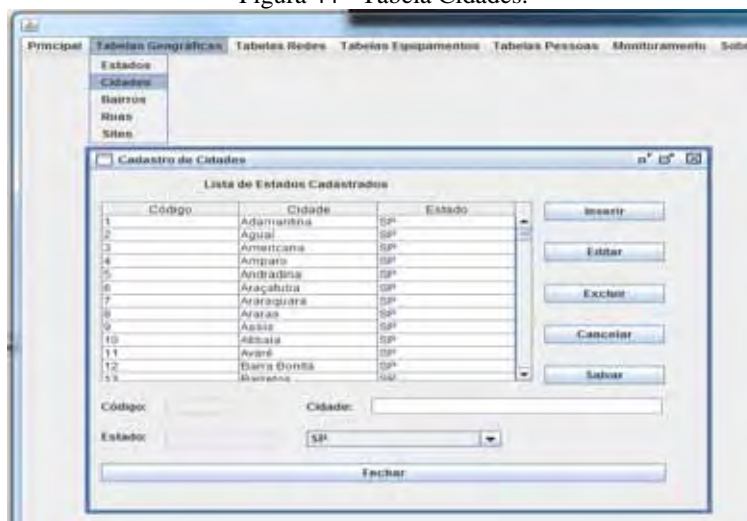


Fonte: (SPOZITO, 2011)

A interface gráfica permite ao operador tratar as informações inseridas no banco de dados de modo simples, sem recorrer a linguagens de programação e linguagens de manipulação de dados.

Na Figura 44 são exibidas as informações referentes às cidades cadastradas no sistema. Como pode ser observado, há dependência de informação anterior; nesse caso deve ser informado obrigatoriamente o estado a que pertence a cidade.

Figura 44 - Tabela Cidades.



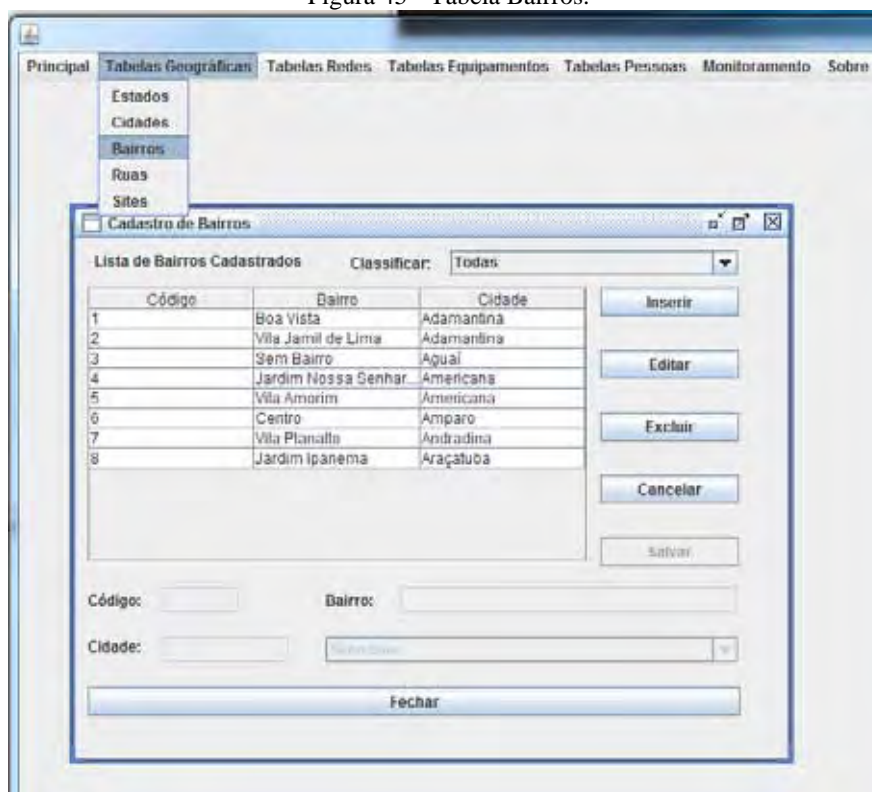
Fonte: (SPOZITO, 2011)

Numa próxima etapa os bairros devem ser informados no sistema. Cada bairro cadastrado é relacionado a apenas uma cidade. Essa regra garante que mesmo havendo bairros homônimos, é possível identificar a qual cidade o mesmo pertence.

Considerando que durante o procedimento de cadastramento da cidade, o estado deve ser informado, tal informação pode ser suprimida, tornando a interface gráfica mais simples sob a ótica do operador.

Durante o processo de cadastramento de um bairro, é obrigatório informar a qual cidade o mesmo pertence. A tela de operações referentes à tabela de bairros, pode ser observada na Figura 45.

Figura 45 - Tabela Bairros.



Fonte: (SPOZITO, 2011)

Os últimos dados de referência político geográfica que devem ser inseridos no sistema são os nomes das ruas.

Considerando que o sistema foi projetado para aceitar cadastros em âmbito nacional, realizar operações referentes a nomes de ruas, pode tornar-se tarefa complexa por causa do

volume de dados. Deste modo, foram acrescentados filtros de exibição que permitem ao operador classificar as informações exibidas na tela por cidades (Figura 46) e após por bairros (Figura 47).

A inserção de uma nova rua no sistema implica em relacioná-la a uma cidade e bairro, obrigatoriamente cadastrados no sistema, como pode ser observado na Figura 48.

Figura 46 - Tabela Ruas (classificação cidades).

Código	Rua	Cidade	Bairro
1	Adamantina	Adamantina	
2	Aguaí	Adamantina	
3	Americana	Adamantina	
4	Amparo	Americana	
5	Andradina	Americana	
6	Araçatuba	Amparo	
7	Araçatuba	Andradina	
8	Avenida Pre. 16007	Jardim Ipanema	Araçatuba

Fonte: (SPOZITO, 2011)

Figura 47 - Tabela Ruas (classificação bairros).

Código	Rua	CEP	Bairro
4	Avenida Nossa Senhora de Fátima	13478008	Jardim Ipanema
	Via Anorim		

Fonte: (SPOZITO, 2011)

Figura 48 - Cadastro de Ruas.

Código	Rua	CEP	Bairro	Cidade
1	Estrada Seis	17800000	Boa Vista	Adamantina
2	Rua Libero ...	17800000	Boa Vista	Adamantina
3	Rua Joaqui...	0	Vila Jamil de...	Adamantina
4	Avenida Nos...	13478000	Jardim Nos...	Americana
5	Rua Emilio ...	13469111	Vila Amorim	Americana
6	Rua Sete de...	13900372	Centro	Amparo
7	Estr.Vic. Seb...	16900970	Vila Planalto	Andradina
8	Avenida Pre...	16007	Jardim Ipan...	Araçatuba

Fonte: (SPOZITO, 2011)

Inseridas as informações referentes a critérios político geográficos, deve o operador inserir no sistema o local que será monitorado.

O nome atribuído a esta informação vem do inglês *Site*, que traduzido refere-se a um local, podendo este ser um local físico ou virtual.

Durante o processo de cadastramento o operador deverá informar dados específicos que auxiliem a identificação nos mapas. Foram disponibilizados campos para o nome do local, apelido se o mesmo possuir, telefone e nesse momento é realizado o referenciamento geográfico do ponto informando a latitude e longitude. Para facilitar o processo de cadastramento, foram implementados *combos* ou botões que listam hierarquicamente a rua do novo local informado.

Um filtro também foi programado, possibilitando classificar por cidade os locais cadastrados, facilitando o trabalho do operador.

Na Figura 49 pode ser observada a interface para as operações referentes aos *sites* no sistema.

Figura 49 - Cadastro de sites.

Código	Site	Apelido	Rua	Número	Bairro	Cidade	Latitude	Longitude	Fone
1	ETEC Eng. Herval Bellussi	apelido	Estrada	0	Boa Vista	Adama	1.0E-4	2.0E-4	(18) 35
2	ETEC Professor Eudélio L.	apelido	Rua Lib	500	Boa Vista	Adama	1.0E-4	2.0E-4	(18) 35
3	ETEC de Agual	apelido	Rua Jo	500	Vila Jp.	Adama	1.0E-4	2.0E-4	(00) 00
4	ETEC Polivalente de Americ	apelido	Avenida	500	Jardim	Americ	1.0E-4	2.0E-4	(00) 00
5	FATEC Americana	apelido	Rua E.	500	Vila Am	Americ	1.0E-4	2.0E-4	(00) 00
6	ETEC João Belarmino	apelido	Rua Se	500	Centro	Amparo	1.0E-4	2.0E-4	(19) 38
7	ETEC Sebastiana Aug. de M.	apelido	Est. Vic.	500	Vila Pta	Andrad.	1.0E-4	2.0E-4	(18) 37
8	ETEC Araçatuba	apelido	Avenida	1754	Jardim	Araçat.	1.0E-4	2.0E-4	(18) 38
9	FATEC Araçatuba	apelido	Avenida	1754	Jardim	Araçat.	1.0E-4	2.0E-4	(18) 38

Fonte: (SPOZITO, 2011)

6.2.2 Bloco tabelas equipamentos.

Como observado no banco de dados, na Figura 34 e na Figura 38, há relacionamento entre o ponto monitorado, denominado *host*, o local ou *site* e o tipo de equipamento instalado. Deste modo o cadastramento de um ponto a ser monitorado, depende do cadastramento prévio dessas informações.

O processo de inserção de um equipamento é realizado com auxílio da interface gráfica disponibilizada no sub-menu equipamentos, inscrito item do menu tabelas equipamentos (Figura 50).

Um ponto chave no processo de cadastramento é a inserção de um *host* no sistema (Figura 51). Nessa etapa as informações de rede específicas do local a ser monitorado são informadas e devem estar em consonância com as características técnicas da rede de dados, isto é, o número IP do *host* deve pertencer a rede em conformidade com a versão 4 do protocolo IP (IPv4). Relacionar o *host* ao local ou *site* correto, também implicará na correta exibição no mapa da localização geográfica do ponto.

Figura 50 – Cadastro equipamentos.

Principal Tabelas Geográficas Tabelas Redes Tabelas Equipamentos Tabelas Pessoas Monitoramento Sobre

Cadastro de Equipamentos

Lista de Estados Cadastrados

Código	Descrição	Marca	Modelo	Obs
1	roteador padrão	marca padrao	modelo padrao	

Inserir
Editar
Excluir
Cancelar
Salvar

Código: Descrição:
 Marca: Modelo:
 Observação:

Fechar

Fonte: (SPOZITO, 2011)

Figura 51 - Cadastro de hosts.

Principal Tabelas Geográficas Tabelas Redes Tabelas Equipamentos Tabelas Pessoas Monitoramento Sobre

Cadastro de Hosts

Lista de Hosts Cadastrados

Host	Equipam.	Escola	IP	Mascara
1	1	1	10.66.162.1	255.255.255.255
2	1	2	10.66.168.1	255.255.255.255
3	1	3	10.67.222.1	255.255.255.255
4	1	4	10.66.98.1	255.255.255.255
5	1	5	10.66.10.1	255.255.255.255
6	1	6	10.66.186.1	255.255.255.255
7	1	7	10.66.122.1	255.255.255.255
8	1	8	10.67.88.1	255.255.255.255
9	1	9	10.67.202.1	255.255.255.255

Inserir
Editar
Excluir
Cancelar
Salvar

Descrição:
 IP: Máscara:
 Localização:

Fechar

Fonte: (SPOZITO, 2011)

6.2.3 Bloco monitoramento.

O sistema GeoMon_IPv4 foi desenvolvido para auxiliar o trabalho de gestores de redes WAN, ou redes geograficamente distribuídas.

Após inseridas as informações que podem ser classificadas genericamente como:

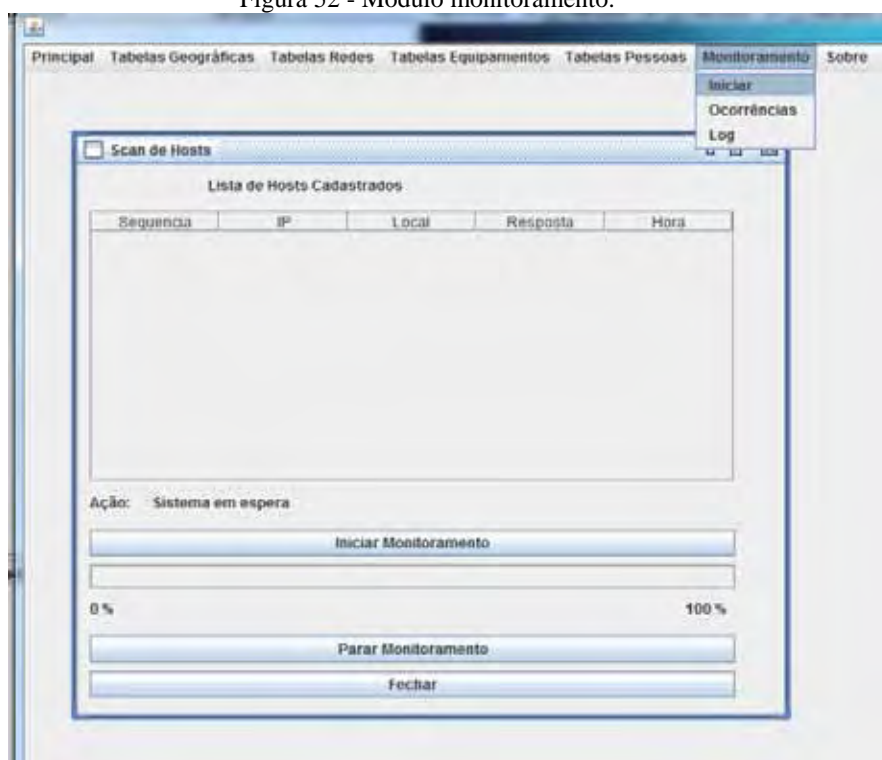
- Objeto monitorado;
- Local do objeto.

Deve-se iniciar a coleta de informações sobre o *status* desses objetos, ou *hosts*.

Foi desenvolvido um código de programação com características diferenciadas nesse bloco. Há necessidade de verificar continuamente se determinado ponto remoto está em funcionamento, sem bloquear outras operações no sistema.

O código de programação desenvolvido para a operação do sistema de monitoramento ou *scan* (Figura 52) foi desenvolvido utilizando recursos de programação denominados *threads*, que consistem em uma forma de um processo dividir a si mesmo em duas ou mais tarefas executadas concorrentemente (DEITEL; DEITEL, 2003).

Figura 52 - Módulo monitoramento.

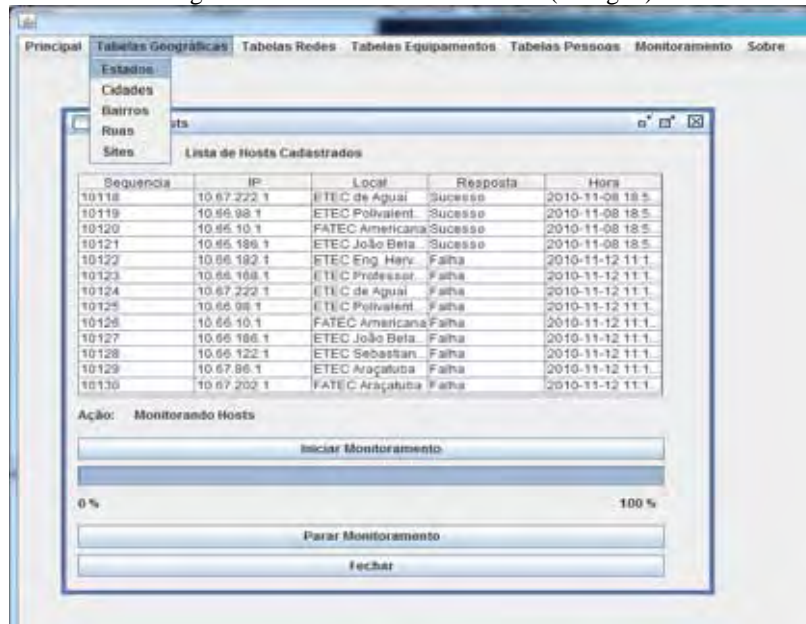


Fonte: (SPOZITO, 2011)

Quando o monitoramento é iniciado, duas novas *threads* concorrentes são criadas pelo *software*, liberando o programa principal para outras tarefas.

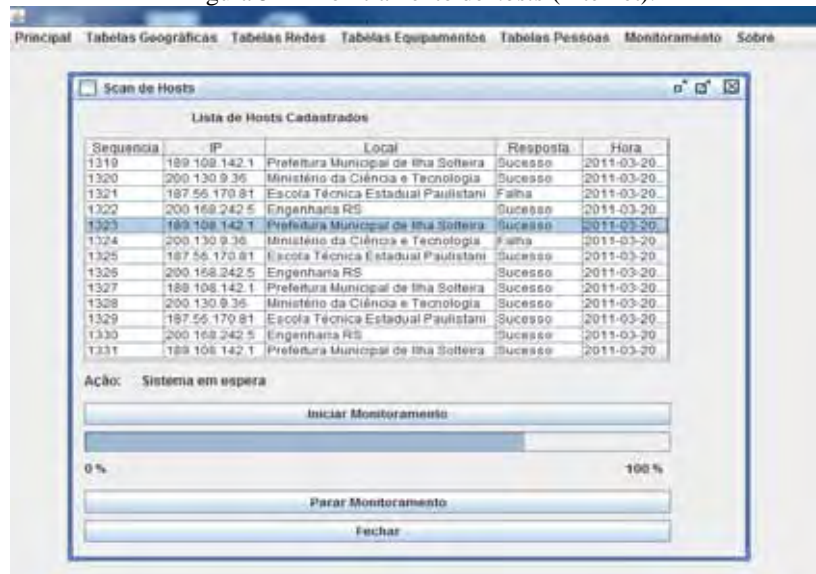
Na Figura 53 e Figura 54 é possível observar o processo de monitoramento e acesso ao menu de tabelas geográficas simultaneamente.

Figura 53 - Monitoramento de *hosts* (Intragov).



Fonte: (SPOZITO, 2011)

Figura 54 - Monitoramento de *hosts* (Internet).



Fonte: (SPOZITO, 2011)

Durante o processo de monitoramento, o sistema armazena no banco de dados o histórico do *status* dos pontos remotos.

A consulta dos registros facilita ao operador conhecer exatamente a data e horário de interrupção da comunicação entre o ponto de origem e destino.

Na Figura 55 pode ser observada a interface gráfica que exhibe todo o histórico do *status* dos pontos monitorados. Foram implementados filtros que facilitam a classificação do local (*site*) e *status*.

O bloco de monitoramento pode ser considerado o núcleo do sistema. Quando ativado, todas as informações exibidas em tela são geradas a partir das rotinas desenvolvidas dentro desse módulo.

Figura 55 - Logs de monitoramento.

The screenshot shows a software window titled 'Registro de Logs' with a menu bar containing 'Principal', 'Tabelas Geográficas', 'Tabelas Redes', 'Tabelas Equipamentos', 'Tabelas Pessoas', 'Monitoramento', and 'Sobre'. The main area displays a table with the following data:

Sequencia	IP	Site	Resposta	Equipamento	Data
1	10.66.182.1	ETEC Eng. Herval ...	Sucesso	1	2010-11-08 11:14:...
2	10.66.168.1	ETEC Professor Eu...	Sucesso	2	2010-11-08 11:14:...
3	10.67.222.1	ETEC de Aguaí	Sucesso	3	2010-11-08 11:14:...
4	10.66.98.1	ETEC Polivalente d...	Sucesso	4	2010-11-08 11:14:...
5	10.66.10.1	FATEC Americana	Sucesso	5	2010-11-08 11:14:...
6	10.66.186.1	ETEC João Belarm...	Sucesso	6	2010-11-08 11:14:...
7	10.66.122.1	ETEC Sebastiana ...	Sucesso	7	2010-11-08 11:14:...
8	10.67.86.1	ETEC Araçatuba	Sucesso	8	2010-11-08 11:14:...
9	10.67.202.1	FATEC Araçatuba	Sucesso	9	2010-11-08 11:14:...
10	10.66.182.1	ETEC Eng. Herval ...	Sucesso	1	2010-11-08 11:15:...
11	10.66.168.1	ETEC Professor Eu...	Sucesso	2	2010-11-08 11:15:...
12	10.67.222.1	ETEC de Aguaí	Sucesso	3	2010-11-08 11:15:...
13	10.66.98.1	ETEC Polivalente d...	Sucesso	4	2010-11-08 11:15:...
14	10.66.10.1	FATEC Americana	Sucesso	5	2010-11-08 11:15:...
15	10.66.102.1	ETEC Eng. Herval ...	Sucesso	4	2010-11-08 11:15:...

Below the table, there is a 'Nova Ocorrência' button and two dropdown menus. The 'Site' dropdown is open, showing a list of sites: Todos, ETEC Eng. Herval Bellusci, ETEC Professor Eudecio Luiz Vicente, ETEC de Aguaí, ETEC Polivalente de Americana, FATEC Americana, ETEC João Belarmino, and ETEC Sebastiana Aug. de Moraes. The 'Status' dropdown is set to 'Todos'.

Fonte: (SPOZITO, 2011)

6.2.4 Bloco principal.

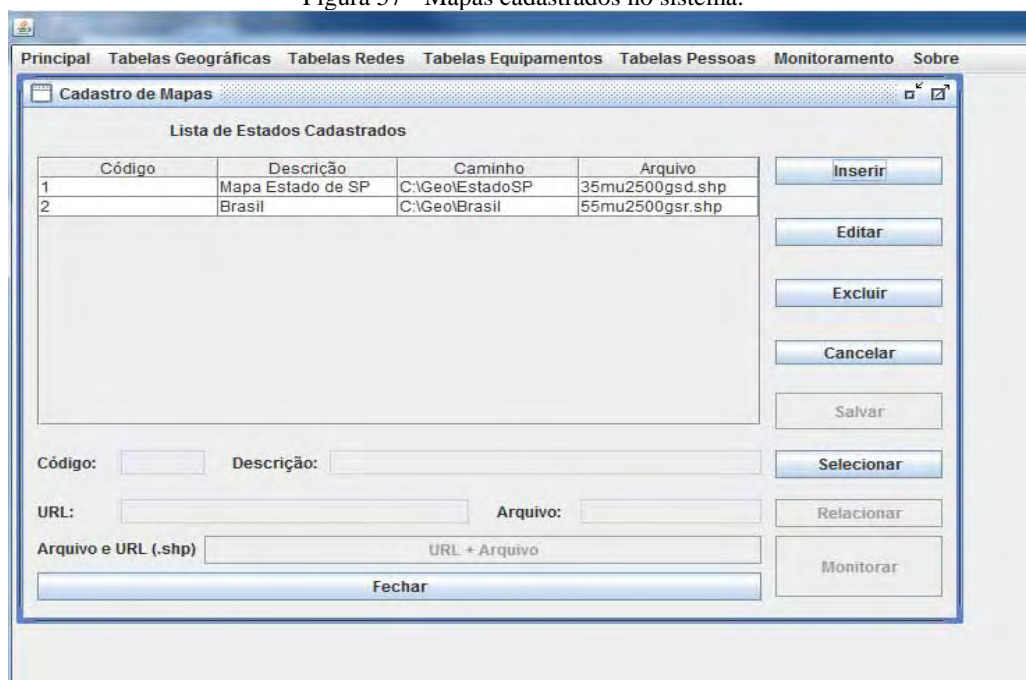
O bloco principal ativa o suporte a ferramenta de geoprocessamento. Ao ativar no menu principal, clicando sobre a opção GIS (Figura 56), são indicadas as bases georreferenciadas cadastradas no sistema (Figura 57).

Figura 56 - Visualizar mapa.



Fonte: (SPOZITO, 2011)

Figura 57 - Mapas cadastrados no sistema.

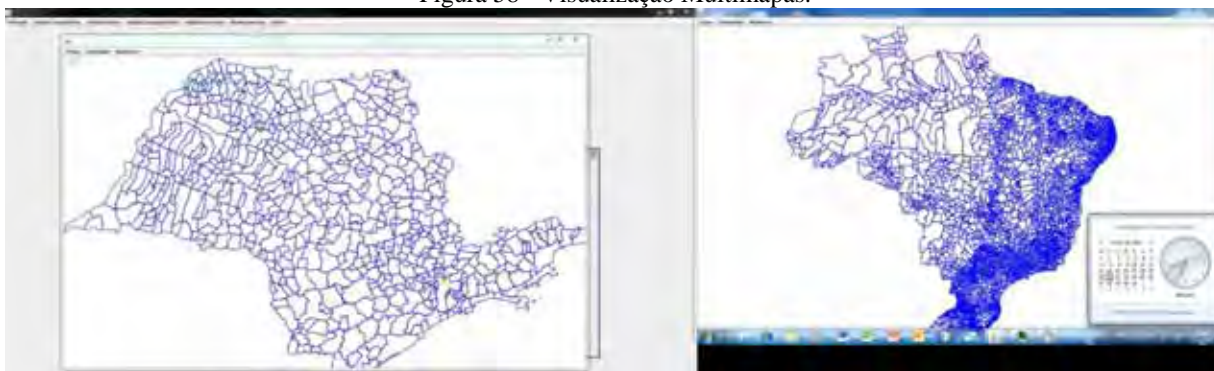


Fonte: (SPOZITO, 2011)

Utilizando o conceito de *threads*, o *software* foi desenvolvido de modo a permitir o monitoramento de vários pontos, respeitando os conceitos de generalização cartográfica, sem prejudicar a visualização pelo operador.

Na Figura 58 é possível, com a utilização do recurso de área de trabalho estendida do Windows, visualizar o sistema operando em tempo real com dois mapas simultaneamente, sendo exibidas as divisões políticos geográficas dos municípios no Estado de São Paulo e também em todo o Brasil.

Figura 58 - Visualização Multimapas.



Fonte: (SPOZITO, 2011)

A Figura 59 corresponde a uma ampliação do mapa do Estado de São Paulo exibido na Figura 58.

Figura 59 - Mapa do Estado de São Paulo.

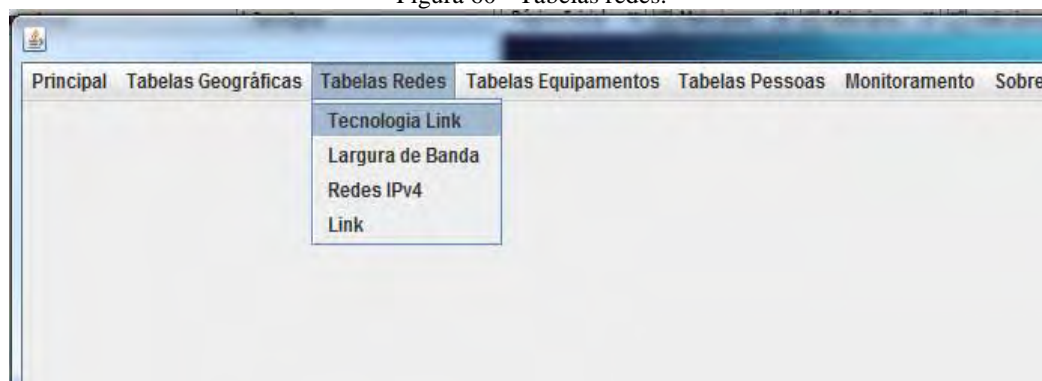


Fonte: (SPOZITO, 2011)

6.2.5 Blocos auxiliares.

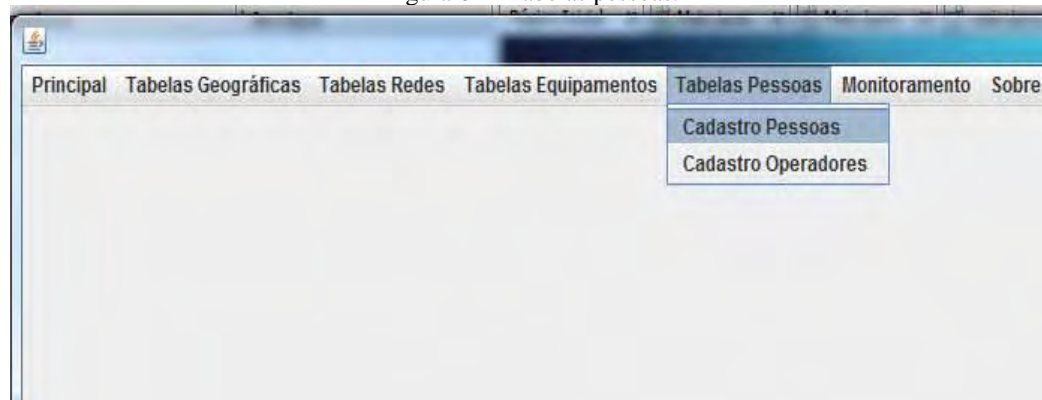
Os blocos: tabelas redes (Figura 60) e tabelas pessoas (Figura 61) são utilizados para cadastro de informações adicionais objetivando auxiliar os gestores de rede quanto às informações técnicas (topologia de rede lógica e tecnologia utilizada) e responsáveis pelos locais.

Figura 60 - Tabelas redes.



Fonte: (SPOZITO, 2011)

Figura 61 - Tabelas pessoas.



Fonte: (SPOZITO, 2011)

6.3 Organização do código fonte.

A linguagem Java possui por concepção características que permitem aos programadores aplicar durante o desenvolvimento de um *software*, técnicas que agilizam o processo e possibilitam uma melhor compreensão por terceiros das rotinas de desenvolvidas.

Durante o desenvolvimento foi estudada e utilizada a biblioteca GeoTools, disponibilizada por terceiros para aplicações SIG. A documentação e exemplos disponibilizados no sítio da Internet pelo grupo, bem como o simples processo de importação das bibliotecas, possibilitou o desenvolvimento do sistema em tempo menor se comparado à situação em que há o desenvolvimento de todas as rotinas do sistema.

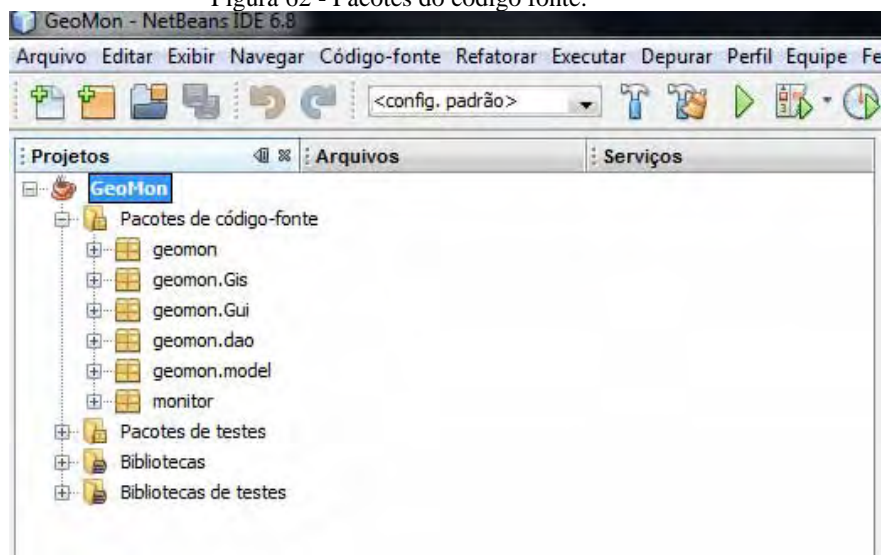
O modelo de desenvolvimento de padrão de *softwares* denominado MVC – *Model-View-Controller* (PRESSMAN, 1995), que visa separar a lógica de exibição, lógica de negócios e a lógica de persistência foi utilizado desde o início da codificação do sistema.

Na Figura 62, é possível observar dentro do aplicativo NetBeans os grupos de código fonte subdividido em pacotes.

Cada pacote possui um conjunto de programas que interagem baseados nos conceitos de orientação a objetos e desenvolvimento estruturado.

Considerando que as classes mais complexas possuem até 500 linhas, apenas partes dos códigos fonte serão colocadas como apêndice no trabalho.

Figura 62 - Pacotes do código fonte.

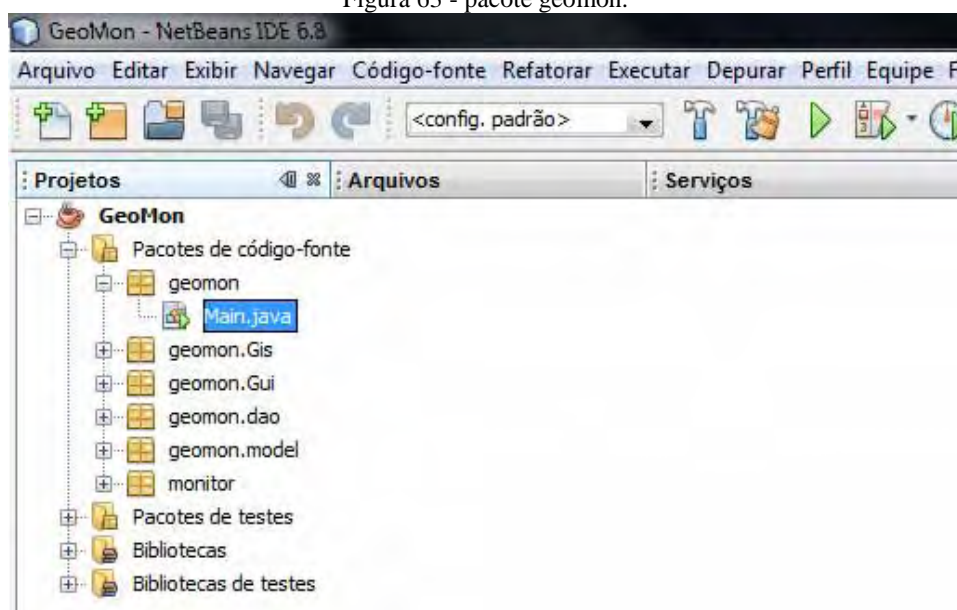


Fonte: (SPOZITO, 2011)

6.3.1 Pacote geomon

O pacote geomon (Figura 63) possui apenas uma classe ou programa chamado Main.java. Sempre que o programa é executado, o interpretador busca nessa classe a primeira linha de comando.

Figura 63 - pacote geomon.



Fonte: (SPOZITO, 2011)

A classe Main.java possui o código mais simples do *software* desenvolvido com apenas 32 linhas.

No Apêndice II pode ser observado o código fonte.

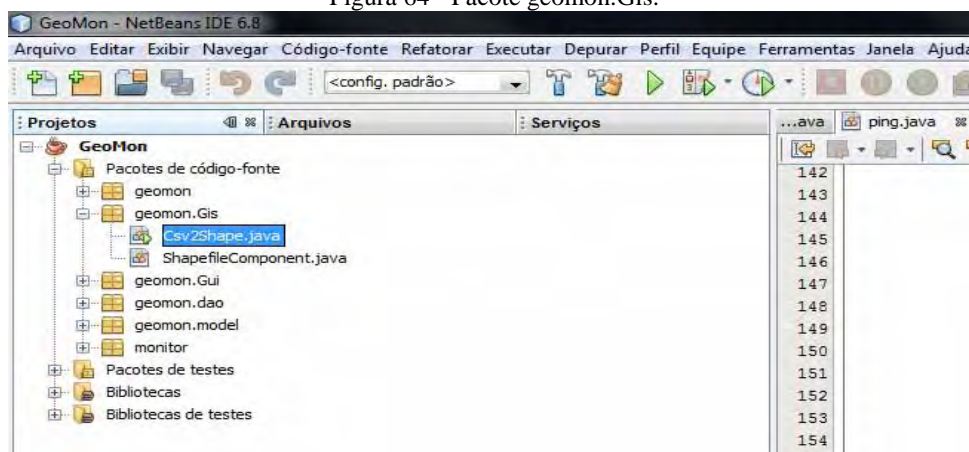
6.3.2 Pacote geomon.Gis

Em geomon.Gis (Figura 64) foram desenvolvidas as classes cuja função é tratar os dados obtidos através da consulta ao banco de dados e periodicamente exibir em um mapa os resultados conforme as regras codificadas no corpo do programa.

No Apêndice III o código fonte da classe ShapefileComponent.java pertencente a esse pacote pode ser analisada.

A classe `ShapefileComponent.java`, através das informações obtidas em um arquivo denominado *shapefile*, (.shp) retorna a figura de todos os objetos geográficos descritos e compilados.

Figura 64 - Pacote `geomon.Gis`.



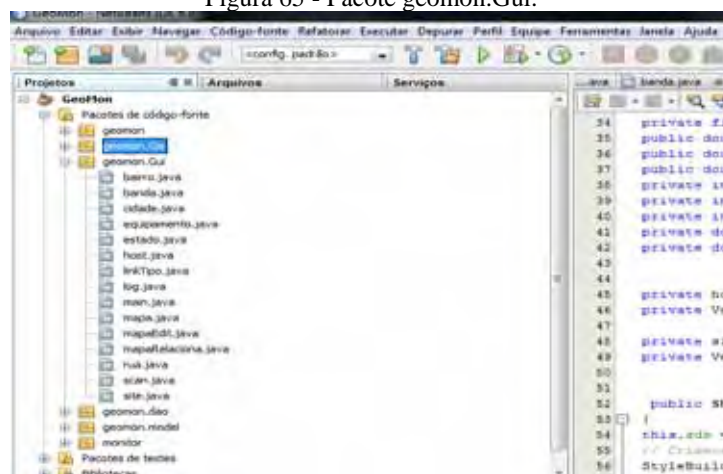
Fonte: (SPOZITO, 2011)

6.3.3 Pacote `geomon.Gui`

Nesse conjunto de classes (Figura 65) são desenhadas as interfaces gráficas e parte das regras de negócios do sistema. Dada a importância e considerado o núcleo do sistema, a classe `scan.java` pode ser analisada no Apêndice IV.

Nessa classe também estão codificadas as *threads* responsáveis por criar os processos concorrentes que monitoram e salvam no banco de dados as informações dos *hosts*, que servirão de base para visualização e gestão dos sistemas remotos.

Figura 65 - Pacote `geomon.Gui`.



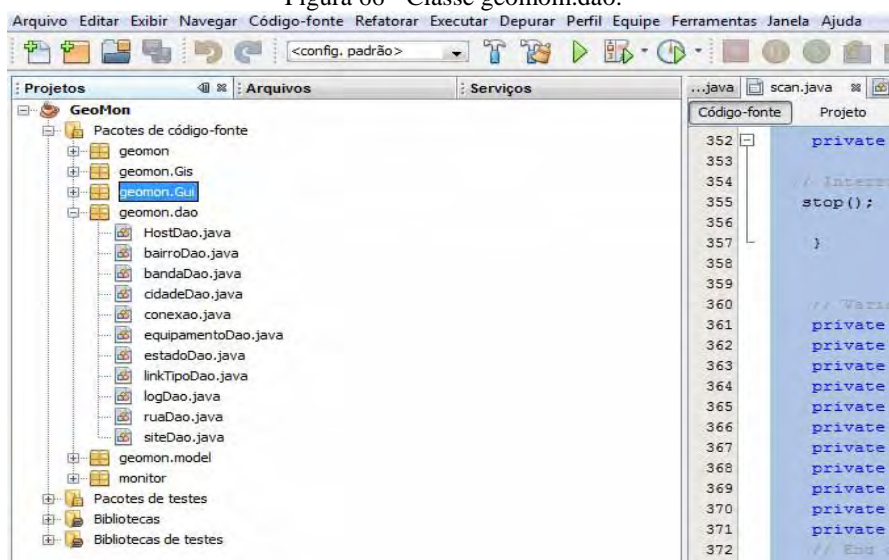
Fonte: (SPOZITO, 2011)

6.3.4 Pacote geomon.dao

Projetado para monitorar e registrar um histórico de informações no banco de dados, este pacote tem como função salvar e recuperar informações de um SGDB em tempo de execução, consumindo o mínimo de processamento e portas de acesso do SGDB.

As classes do pacote geomon.dao (Figura 66) possuem unicamente a função de manipular os dados da base de dados conforme a solicitação do usuário ou operação do sistema.

Figura 66 - Classe geomom.dao.



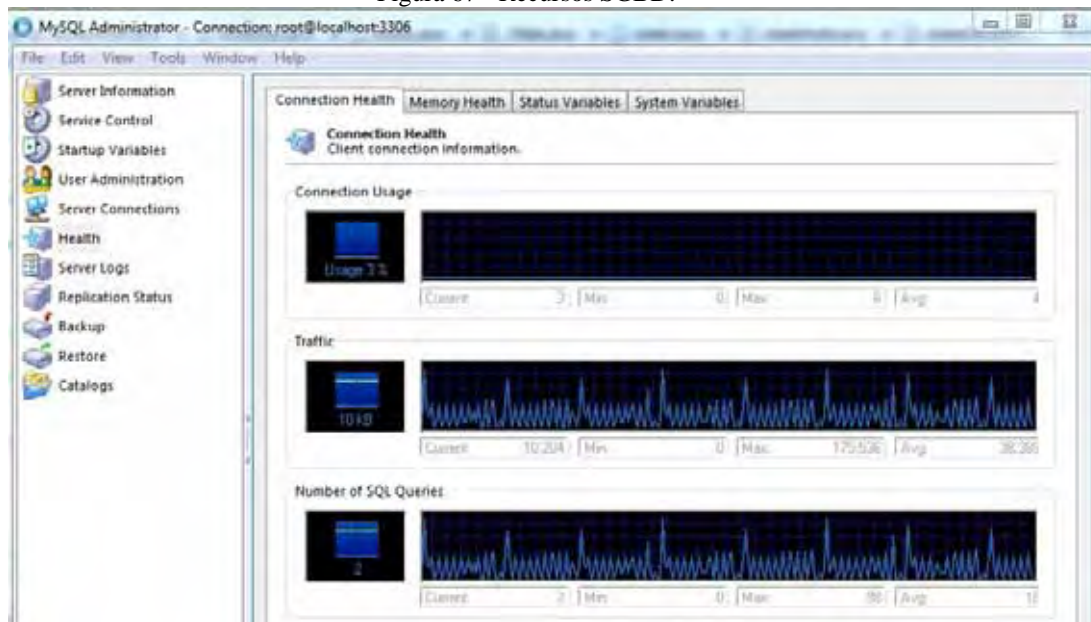
Fonte: (SPOZITO, 2011)

Deste modo, durante o planejamento e escrita das classes responsáveis por essa tarefa, as rotinas foram testadas e adequadas de modo a exigir em cada operação o mínimo consumo de recursos computacionais.

Na Figura 67 pode ser observado o consumo dos recursos durante a operação do sistema, utilizando uma ferramenta disponibilizada pelo desenvolvedor do SGDB.

Durante o desenvolvimento de todas as classes de manipulação do banco de dados foram monitorados os recursos e na fase de monitoramento a classe chamada HostDao.java é acessada em intervalos menores que um segundo. Garantir que todas as conexões abertas sejam fechadas e reduzir o tráfego de dados trouxe estabilidade ao sistema.

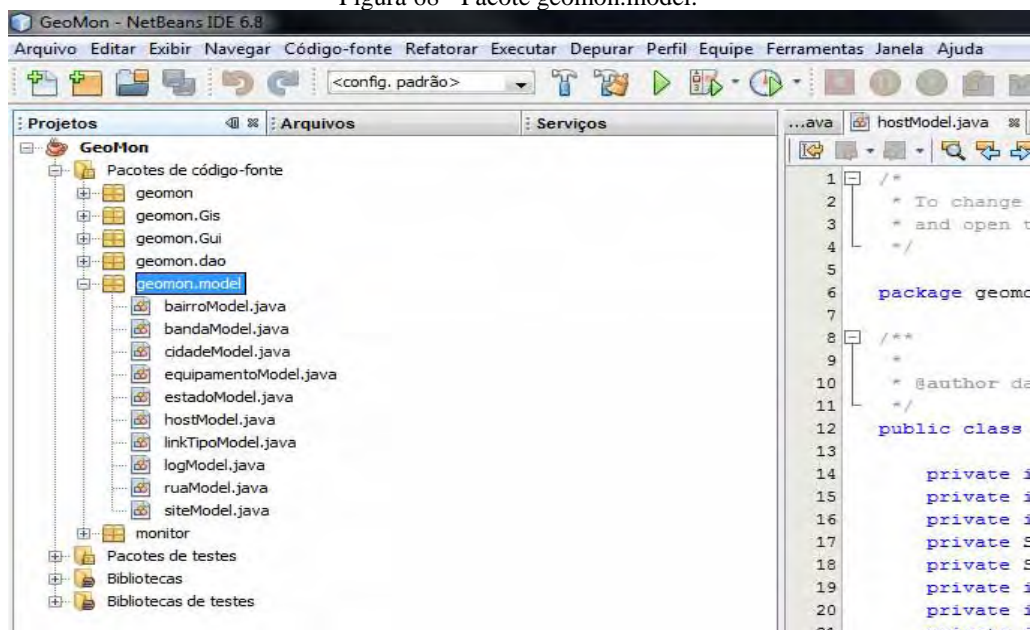
Figura 67 - Recursos SGBD.



Fonte: (SPOZITO, 2011)

6.3.5 Pacote geomon.model

O pacote `geomon.model` (Figura 68) contém as classes auxiliares exigidas na programação orientada a objetos para o modelo MVC.

Figura 68 - Pacote `geomon.model`.

Fonte: (SPOZITO, 2011)

No Apêndice V, está descrita a classe `hostModel.java` com os atributos e métodos. Todas as classes desse pacote refletem os campos programados nas tabelas do banco de dados.

6.3.6 Pacote `geomon.monitor`

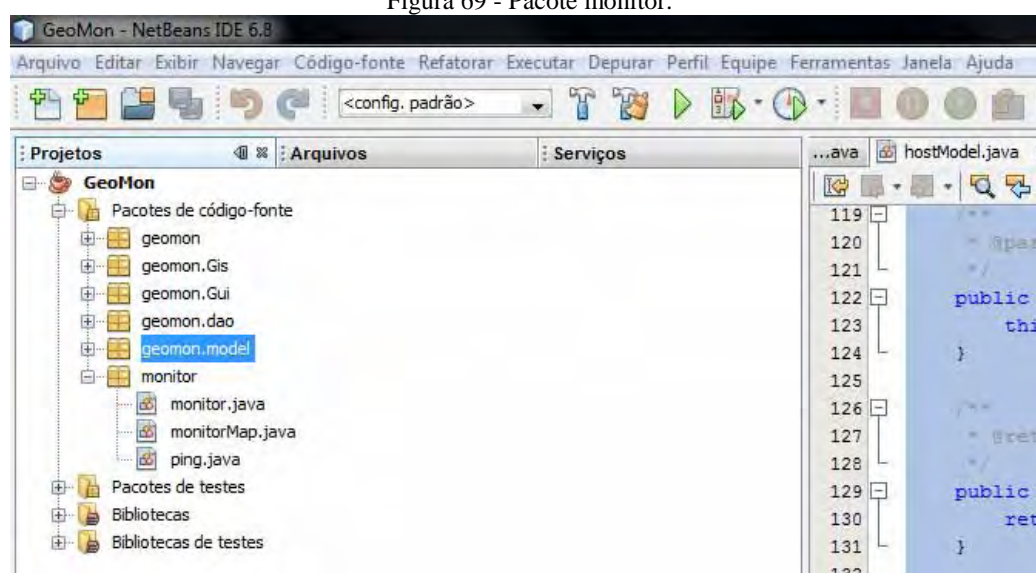
Considerando a necessidade de executar processos concorrentes para monitoramento e atualização da base de dados, foi desenvolvido um pacote com algumas classes que são executadas simultaneamente.

Dentro do pacote `monitor` (Figura 69) as classes `monitor.java`, `monitorMap.java` e `ping.java` executam atividades fundamentais quando o sistema está operando.

Ao iniciar o monitoramento dos pontos remotos, são abertos processos concorrentes da classe `monitor.java` como *threads*.

Para consultar um ponto remoto a classe `monitor.java`, invoca a classe `ping.java` passando os parâmetros de rede do ponto que será consultado. A classe `ping.java` foi desenvolvida para acessar a pilha do protocolo TCP/IP com o comando PING em sistema operacional. Considerados os parâmetros temporais programados na classe `ping.java`, é retornada uma sinalização informando se o ponto remoto está operante ou não. São então disparadas informações às outras classes em execução paralela para atualização do banco de dados.

Figura 69 - Pacote `monitor`.



Fonte: (SPOZITO, 2011)

Dentro do pacote monitor a classe monitorMap.java atualiza as informações no mapa em tempo de execução.

As classes monitor.java e monitorMap.java são classificadas como extensões de *thread*, pois devem ser executadas concorrentemente.

6.3.7 Coleta das métricas

Dentro do conjunto de classes que monitoram o sistema, foi desenvolvida uma rotina para coletar o valor instantâneo dos parâmetros retornados pelo comando PING e armazenar no SGBD. Os valores servem de referência ao sistema para exibir ao operador a qualidade do *link* em tempo de execução. Cabe a classe geomon.Gis tratar as informações no momento de plotar as referências nos mapas.

No apêndice VI pode ser observada parte do código fonte e os parâmetros definidos dentro do tempo de resposta para plotar os pontos no mapa em conformidade com os critérios estabelecidos.

Os valores foram definidos dentro do código fonte e classificados conforme os parâmetros indicados na Tabela 2.

Tabela 2 - Limites para cálculo dos parâmetros de qualidade.

Tempo de resposta	Cor plotada	Parâmetro
0 a 100 ms	Verde	Ótimo
101 a 200 ms	Amarela	Bom
201 a 300 ms	Laranja	Ruim
301 a 500 ms	Vermelho	Péssimo ou sem comunicação
<i>Out of time</i>	Preto	Péssimo ou sem comunicação

Fonte: (SPOZITO, 2011)

6.4 Testes do sistema GeoMonIPv4

Para analisar o comportamento e resposta do sistema desenvolvido, foram cadastradas algumas unidades pertencentes ao Centro Paula Souza. Foi elaborado um *script* de inserção com

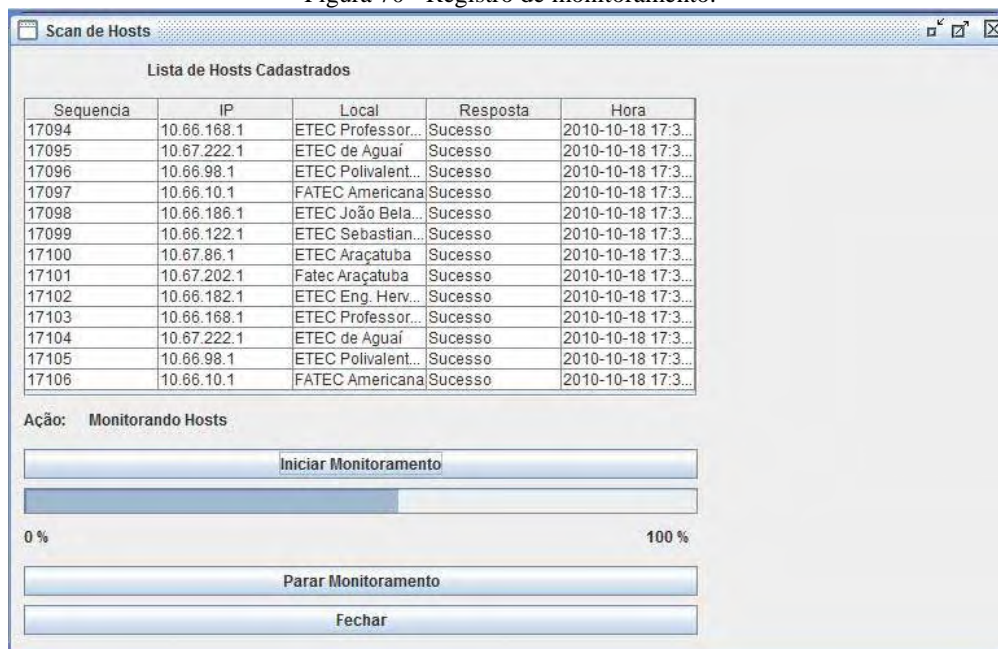
com as informações fornecidas pela equipe de informática da instituição. O *script* de inserção dos dados pode ser analisado no Apêndice VII.

As bases cartográficas ou mapas georreferenciados foram obtidos a partir do sítio da Internet do IBGE.

Foram realizadas duas simulações. Na primeira delas, foram gerados dados e armazenados na tabela Logs no banco de dados.

Na Figura 70 pode ser observado o sistema com 17.106 registros ininterruptos de monitoramento. A periodicidade foi de dois segundos para visualizar o status de cada ponto remoto.

Figura 70 - Registro de monitoramento.



Fonte: (SPOZITO, 2011)

Numa segunda etapa, foram ajustadas as temporizações, aumentando o período de dois para três segundos na consulta. O intervalo de atualização do mapa foi ajustado para cinco segundos.

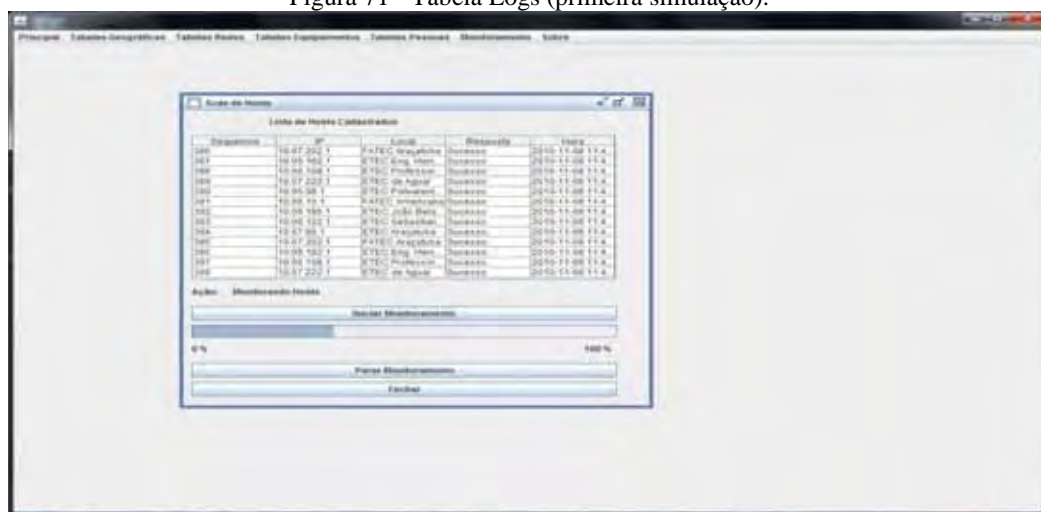
Foram coletados aproximadamente 400 registros (Figura 71) em nove pontos distribuídos no estado de São Paulo. O tempo total de coleta foi de 60 minutos.

Durante o processo de coleta, o sistema se mostrou estável e não causou sobrecargas de processamento ou uso excessivo de memória. Toda a carga no sistema operacional pode ser

observada na Figura 72. Foi utilizado o gerenciador de tarefas do Windows para obter as informações.

Não houve problemas no SGBD, sendo que o mesmo apresentou comportamento muito próximo ao representado na Figura 67.

Figura 71 - Tabela Logs (primeira simulação).

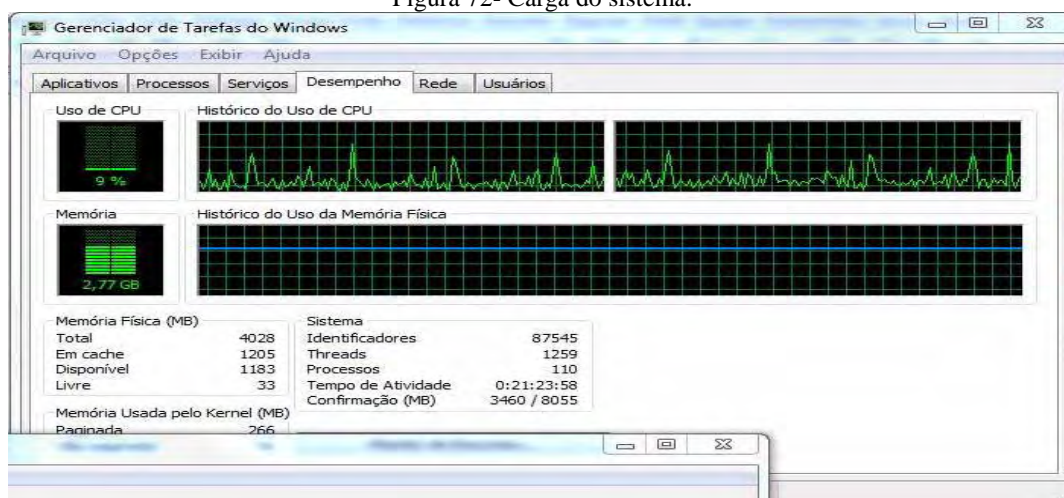


ID	IP	Local	Remote	Data
100	10.0.2.15	192.168.1.1	192.168.1.1	11/08/11 11:00
101	10.0.2.15	192.168.1.1	192.168.1.1	11/08/11 11:00
102	10.0.2.15	192.168.1.1	192.168.1.1	11/08/11 11:00
103	10.0.2.15	192.168.1.1	192.168.1.1	11/08/11 11:00
104	10.0.2.15	192.168.1.1	192.168.1.1	11/08/11 11:00
105	10.0.2.15	192.168.1.1	192.168.1.1	11/08/11 11:00
106	10.0.2.15	192.168.1.1	192.168.1.1	11/08/11 11:00
107	10.0.2.15	192.168.1.1	192.168.1.1	11/08/11 11:00
108	10.0.2.15	192.168.1.1	192.168.1.1	11/08/11 11:00
109	10.0.2.15	192.168.1.1	192.168.1.1	11/08/11 11:00
110	10.0.2.15	192.168.1.1	192.168.1.1	11/08/11 11:00

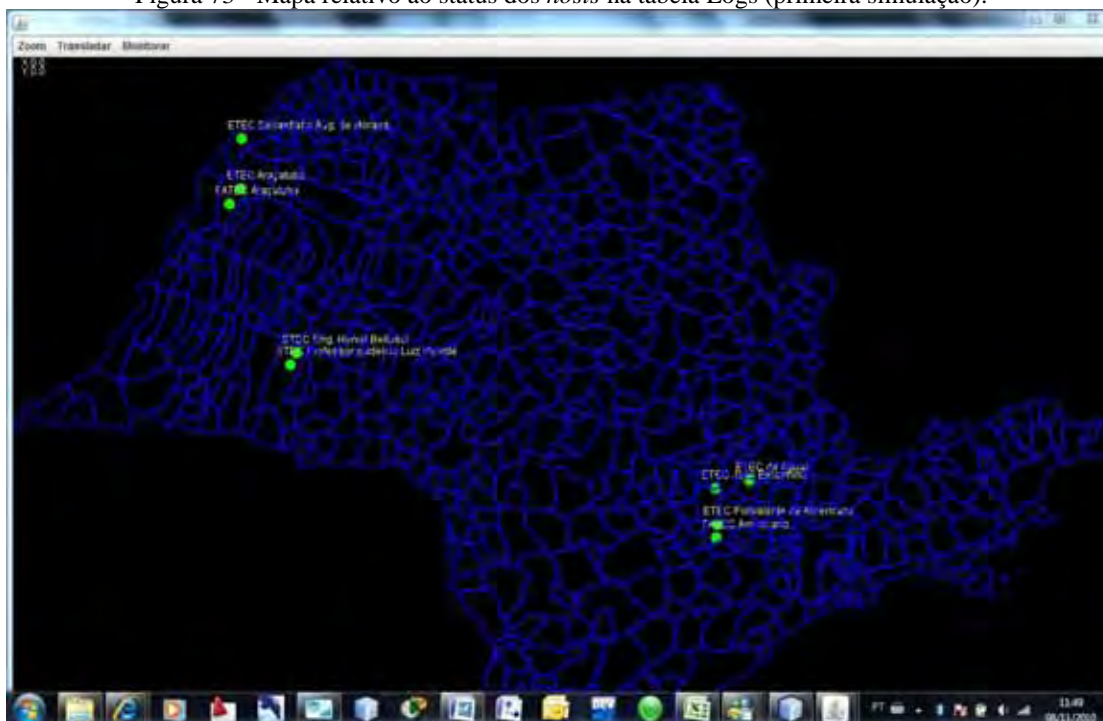
Fonte: (SPOZITO, 2011)

Na Figura 73 foi plotado no mapa o status dos pontos em conformidade com os dados obtidos a partir do banco de dados. Apesar de executados em *threads* com processos concorrentes, as informações exibidas no mapa refletem com precisão se o ponto está operante ou não.

Figura 72- Carga do sistema.



Fonte: (SPOZITO, 2011)

Figura 73 - Mapa relativo ao status dos *hosts* na tabela Logs (primeira simulação).

Fonte: (SPOZITO, 2011)

Considerando a distância geográfica dos pontos monitorados e a inviabilidade de realizar paradas nas redes das unidades remotas, foi preciso determinar uma forma de interromper a comunicação entre o sistema e os pontos remotos.

O sistema de monitoramento estava ativo em microcomputador dentro da cidade de São Paulo, deste modo toda a comunicação era realizada dentro da rede Intragov. Considerando que o envio de pacotes ICMP ocorre no formato ponto-a-ponto, isto é, a requisição parte da origem diretamente ao destino e a confirmação; caso o local esteja operante, retorna à origem, optou-se por interromper apenas o local em que o sistema estava operando.

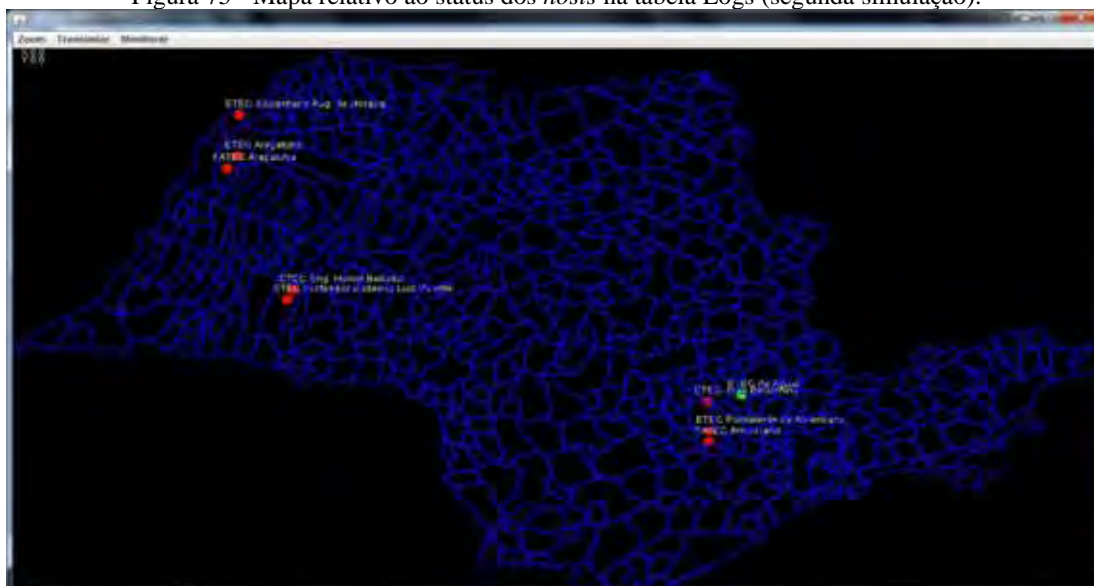
Com interrupção da comunicação na origem, o sistema não teria acesso a nenhum ponto remoto cadastrado. Na Figura 74, é possível observar o sistema respondendo à parada da rede e consequentemente interrupção na comunicação.

Figura 74 - Tabela Logs (segunda simulação).

Seqüencia	IP	Local	Resposta	Hora
398	19.07.222.1	E-TEC de Aguaré	Success	2010-11-08 11:4
399	19.08.98.1	E-TEC Pujolant	Success	2010-11-08 11:5
400	19.08.10.1	FATEC Americana	Success	2010-11-08 11:6
401	19.08.195.1	E-TEC João Beto	Success	2010-11-08 11:5
402	19.08.152.1	E-TEC Sabastian	Success	2010-11-08 11:5
403	19.07.88.1	E-TEC Aracaju	Failure	2010-11-08 11:5
404	19.07.203.1	FATEC Aracaju	Success	2010-11-08 11:5
405	19.08.182.1	E-TEC Eng. Hen.	Success	2010-11-08 11:5
406	19.08.158.1	E-TEC Professor	Success	2010-11-08 11:5
407	19.07.222.1	E-TEC de Aguaré	Success	2010-11-08 11:5
408	19.08.98.1	E-TEC Pujolant	Failure	2010-11-08 11:5
409	19.08.10.1	FATEC Americana	Failure	2010-11-08 11:5
410	19.08.195.1	E-TEC João Beto	Failure	2010-11-08 11:5

Fonte: (SPOZITO, 2011)

Após alguns segundos, foi possível visualizar no mapa a mudança na cor dos pontos monitorados de verde para vermelho (Figura 75).

Figura 75 - Mapa relativo ao status dos *hosts* na tabela Logs (segunda simulação).

Fonte: (SPOZITO, 2011)

Para concluir o teste de resposta do sistema, a rede do ponto de origem foi reestabelecida. Após a renovação por parte do sistema operacional da conexão à rede Intragov, o sistema voltou a obter respostas positivas quanto a operação dos pontos remotos.

O sistema atualizou as informações de todos os locais monitorados em apenas alguns segundos refletindo no mapa com tempo de resposta menor que cinco segundos.

Na Figura 78, são exibidos os dados obtidos da tabela Logs após o reestabelecimento da comunicação. Apesar de exibidas informações de falha na tabela, o tempo necessário para coletar a imagem do mapa (Figura 79) foi superior à resposta do sistema, que já exibia todos os pontos na cor verde.

O sistema ficou operante por 24 horas exibindo o mapa com o status dos pontos e armazenando os registros no banco de dados. Não houve, no período, evidências quanto a paradas nos pontos remotos ou problemas relacionados ao sistema ou *hardware* utilizado no ensaio.

Figura 78 - Tabela Logs (quarta simulação).

ID	IP	Local	Resposta	Hora
422	10.67.202.1	FATEC Araçatuba	Falha	2010-11-08 11:5
423	10.66.182.1	ETEC Eng. Henr.	Falha	2010-11-08 11:5
424	10.66.158.1	ETEC Professor	Falha	2010-11-08 11:5
425	10.67.222.1	ETEC de Agual	Falha	2010-11-08 11:5
426	10.66.88.1	ETEC Pinoberti	Falha	2010-11-08 11:5
427	10.66.15.1	FATEC Aracatuba	Falha	2010-11-08 11:5
428	10.66.186.1	ETEC João Bela	Falha	2010-11-08 11:5
429	10.66.122.1	ETEC Sebastião	Falha	2010-11-08 11:5
430	10.67.86.1	ETEC Araçatuba	Falha	2010-11-08 11:5
431	10.67.202.1	FATEC Araçatuba	Falha	2010-11-08 11:5
432	10.66.182.1	ETEC Eng. Henr.	Sucesso	2010-11-08 11:5
433	10.66.158.1	ETEC Professor	Sucesso	2010-11-08 11:5
434	10.67.202.1	ETEC de Agual	Sucesso	2010-11-08 11:5

Ação: Monitorando Hosts

0% 100%

Fechar

Fonte: (SPOZITO, 2011)

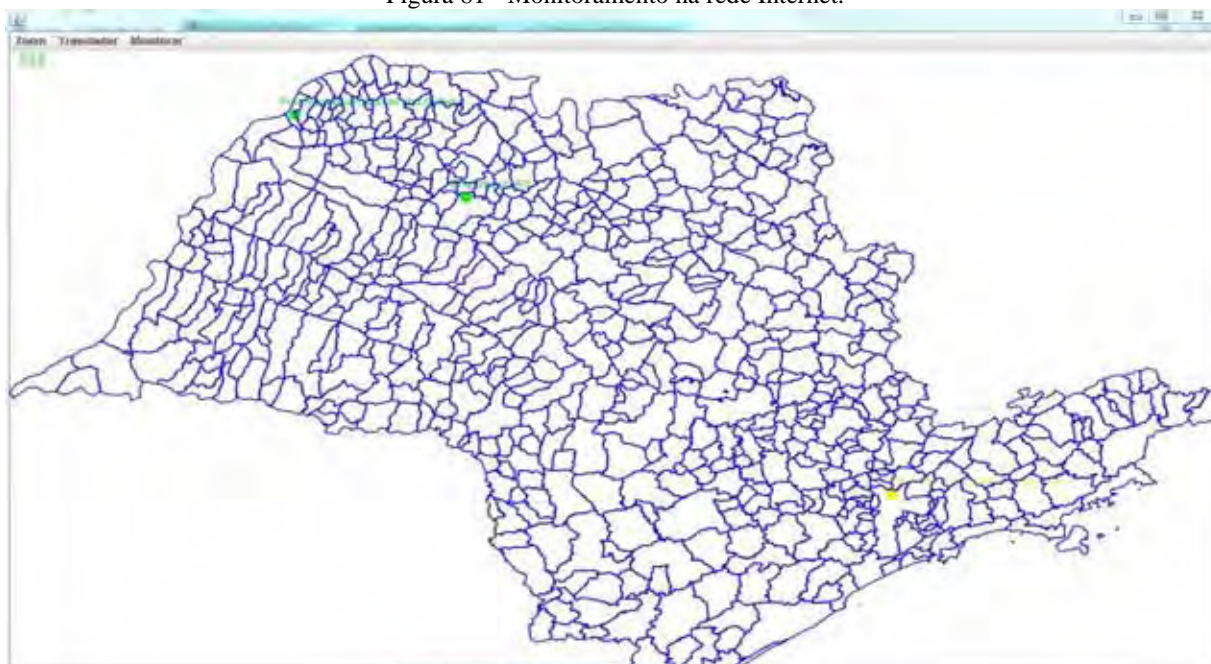
possível observar a alteração das cores entre verde e amarelo nos mapas, resultado obtido a partir da resposta em tempo de execução.

Figura 80 - Monitoramento na rede Internet.



Fonte: (SPOZITO, 2011)

Figura 81 - Monitoramento na rede Internet.



Fonte: (SPOZITO, 2011)

CONCLUSÕES.

Neste trabalho, foi desenvolvido um *software* em linguagem Java que permitiu o monitoramento de redes de comunicação de dados que utilizam os protocolos do padrão TCP/IP.

As informações referentes ao *status* da rede foram coletadas de modo automático e projetadas sobre uma base cartográfica, facilitando a localização dos pontos monitorados.

Como fundamento foram estudados conceitos de redes de comunicação de dados e sistemas de informação geográfica.

Os resultados apresentados nas simulações do *software* se mostraram satisfatórios, considerando que o protótipo desenvolvido respondeu como esperado.

Consideram-se diferenciais importantes, além da utilização de linguagem de programação e ferramentas gratuitas, a visualização dos pontos monitorados sobre uma base cartográfica, isto é, um mapa com a divisão político-regional da área atendida; diferentemente dos sistemas comerciais utilizados para a atividade de supervisão, que utilizam figuras geométricas com ícones representando os equipamentos remotos (roteadores ou microcomputadores) monitorados.

A versatilidade obtida em alterar a base cartográfica, cadastrar os pontos monitorados e relacioná-los aos parâmetros de rede do protocolo IPv4, permite concluir que a ferramenta pode ser utilizada por usuários, sem a necessidade de estudo aprofundado dos conceitos expostos. Operar sobre o sistema operacional Windows 7 © (64 bits), com interface gráfica conhecida pela maioria dos usuários facilita a operação e amplia o ciclo de vida do produto.

Durante as visitas realizadas ao Centro Paula Souza, considerando que há apenas dois profissionais para gerir aproximadamente duzentos e vinte unidades espalhadas no estado de São Paulo, desprovidos de ferramentas computacionais que permitam o monitoramento de modo ativo do *status* de cada ponto, confirma-se a hipótese de que na impossibilidade momentânea de ampliar o quadro de colaboradores, o *software* pode proporcionar uma mudança no modo que as atividades são realizadas, já que possibilitará o acompanhamento ativo do comportamento da rede.

Com base no histórico dos dados de monitoramento armazenado no SGBD, será possível, se necessário, sugerir a contratação e implantação de sistemas secundários de comunicação, como por exemplo, o sistema comercializado pela Telefonica denominado Interdados, que utiliza

tecnologia ADSL, a mesma aplicada no serviço SPEEDY, diferenciando-se apenas por possuir uma faixa de rede IP não pública. Tecnologias como 3G e cable modem poderão ser utilizadas como backup, desde que criadas as regras de proteção necessárias, como a configuração de VPNs, dado o fato das mesmas utilizarem endereços IP públicos na Internet.

Durante o processo, além das referências bibliográficas, a Internet mostrou-se fundamental na obtenção de informações referentes às bibliotecas Java aplicadas.

Como trabalhos futuros sugerem-se os itens listados a seguir:

- Melhorar a representação na base cartográfica dos pontos monitorados, utilizando classes Java para atualização constante de arquivos do formato *shape* (.shp) dos pontos georreferenciados, considerados os parâmetros de generalização cartográfica para que não ocorram confusões por parte dos operadores entre os pontos exibidos na tela com elementos que poderiam pertencer nativamente à base, como por exemplo, confundir um ponto de monitoramento representado por um círculo com uma cidade quando exibido um mapa de um estado da federação.
- Inserir no sistema rotinas que possam coletar e armazenar um histórico dos dados relativos aos parâmetros de qualidade, definidos anteriormente, dos pontos monitorados. Sugere-se também calcular saltos, carga do *link* e informações dos roteadores utilizando protocolo SNMP, exibindo na tela informações mais precisas das cores que representam o *status* dos *links*.
- Estudar e implementar uma interface gráfica com acesso via *browser*, permitindo a operação sem necessidade de instalação de todo o aplicativo, isto é, desenvolver uma rotina de acesso remoto via HTTP de funções específicas do sistema.
- Utilizar servidores de geomonitoramento para gerar mapas mais amigáveis e precisos, como por exemplo o GeoServer© ou API's do Google Maps©.
- Como sugestão da equipe de informática do Centro Paula Souza, criar tabelas e rotinas auxiliares no sistema que possibilitem identificar a quantidade de equipamentos instalados nos pontos monitorados e as atividades exercidas, possibilitando facilitar a gerência dos *links* e planejamento na ampliação dos serviços oferecidos, considerando parâmetros de qualidade e quantidade.

REFERÊNCIAS

BALSEMÃO, F. T. **Gerência e monitoramento de redes através de dispositivos móveis**. 2008. 40 f. Trabalho de Conclusão de Curso (Especialização em Tecnologias, Gerência e Segurança de Redes de Computadores) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.

BHOGENAHALLY, L. A. B. **Applications of GIS for mapping and tracking underground infrastructure**. 2009. 85 f. Dissertation (Master of Science) - The University of Texas, Arlington, 2009.

BOOCH, D.; RAUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. 6. ed. Rio de Janeiro: Campus, 2000. 471 p.

BORGES, KARLA A. V. **Modelagem de dados geográficos**. Belo Horizonte: Universidade Federal de Minas Gerais, 2002. 66p. Apostila do curso de especialização em geoprocessamento.

CÂMARA, G. et al. **Anatomia de sistemas de informação geográfica**. Campinas: UNICAMP/Instituto de Computação, 1996. 197p.

DAVIS, C.; FONSECA, F. **Introdução aos sistemas de informação geográficos**. Belo Horizonte: Universidade Federal de Minas Gerais, 2001. 261p. Apostila do curso de especialização em geoprocessamento.

DEITEL, H. M; DEITEL, P J. **Java: como programar**. 4. ed. Porto Alegre: Bookman, 2003. 1386 p.

FERREIRA, A. E. L. A. **Engenharia de tráfego em redes IP: estudo e aplicação de ferramentas de medição**. 2005. 191 f. Dissertação (Mestrado em Computação) – Instituto de Computação, Universidade Federal Fluminense, Niterói, 2005.

GARCEZ, O. L. **Avaliação de desempenho e dimensionamento de redes de teleinformática centralizada para tráfego de dados corporativos**. 2007. 99 f. Dissertação (Mestrado Profissional em Gestão de Redes de Telecomunicações) – Centro de Ciências Exatas, Ambientais e de Tecnologias, Pontifícia Universidade Católica – PUC, Campinas, 2007.

GONCALVES, E. **Dominando NetBeans**. São Paulo: Ciência Moderna, 2006. 384 p.

GOMES, J. E. **Protótipo de localizador de notebook, utilizando GPS e o API do Google Maps**. 2008. 110 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Centro Universitário do Sul de Minas, Varginha, 2008.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE. **Arquivos de divisão territorial**. [S.l.: s.n., 2010?] 1p. Disponível em: <ftp://geoftp.ibge.gov.br/Organizacao/Divisao_Territorial>. Acesso em: 30 out. 2010.

NAGIOS. **Software de monitoramento**. [S.l.: s.n., 2011?] 1p. Disponível em: <<http://www.nagios.org>>. Acesso em: 10 jan. 2011.

NASSIF, L.; NORONHA, B.; ALEX, V. et al. Gerenciamento georreferenciado de falhas para redes ópticas metropolitanas. **Informática Pública**, Belo Horizonte, v.2, n.10, 2008. Disponível em: <http://www.ip.pbh.gov.br/ANO10_N2_PDF/gerenciamento_georreferenciado.pdf>. Acesso em: 15 out. 2010.

MRTG. **Software de monitoramento**. [S.l.: s.n., 2011?] 1p. Disponível em: <<http://www.networkview.com>>. Acesso em: 25 jan. 2011.

ODOM, W.; MCDONALD, R. **Networking basics CCNA 1 companion guide**. Indianapolis: Cisco Press, 2006. 624 f.

ODOM, W.; MCDONALD, R. **Routers and routing basics CCNA 2 companion guide**. Indianapolis: Cisco Press, 2006. 473 f.

OPMANAGER. **Software de monitoramento** [S.l.: s.n., 2011?] 1p. Disponível em: <<http://www.manageengine.com>>. Acesso em: 15 jan. 2011.

PRESSMAN, R. S. **Engenharia de software**. São Paulo: Makron Books, 1995. 1056 p.

SANTANA, S. F. L.. **Proposta de referência para projetos de qualidade de serviço (QoS) em redes corporativas**. 2006. 185 f. Dissertação (Mestrado em Redes de Computadores) - Universidade Salvador - UNIFACS, Salvador, 2006.

SANTOS, G. H. A.. **Cadastro da rede de abastecimento do município de Iha Solteira com a utilização de geoprocessamento**. 2008. 42 f. Trabalho de Conclusão de Curso (Relatório final de estágio) - Faculdade de Engenharia, Universidade Estadual Paulista-UNESP, Ilha Solteira, 2008.

STALLINGS, W. **Arquitetura e organização de computadores**. 5. ed. São Paulo: Makron Books, 2002. 792 p.

TANEMBAUM, A. S. **Redes de computadores**. Rio De Janeiro: Campus, 1997.

TRIBUNAL DE CONTAS DA UNIÃO – TCU. **Mapa da exclusão digital**. [S.l.: s.n., 2003?]. 143p. Disponível em: <<http://portal2.tcu.gov.br/portal/pls/portal/docs/669944.PDF>>. Acesso em: 12 nov. 2010.

ZARPELÃO, B. B. **Deteção de anomalias em redes de computadores**. 2010. 159 f. Tese (Doutorado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas – UNICAMP, Campinas, 2010.

ZUZATO, F. L. **Rede ZigBee gerenciada por sistema de monitoramento remoto utilizando TCP/IP e GPRS**. 2009. 138 f. Dissertação (Mestrado em Engenharia Elétrica) - Escola de Engenharia de São Carlos, Universidade de São Paulo - USP, São Carlos, 2009.

WIKIPEDIA. **Snow cholera map.** [S.l.: s.n., 2006?]. Disponível em: <http://pt.wikipedia.org/wiki/John_Snow>. Acesso em: 15 mar. 2010.

ANEXOS

Anexo I

Caracteres não imprimíveis

Tabela 3 – Caracteres de controle da padronização ASCII

Binário	Decimal	Hexa	Controle	Abreviação	Descrição
0000 0000	00	00	^@	NUL	Null - Nulo
0000 0001	01	01	^A	SOH	Start of Header - Início do cabeçalho
0000 0010	02	02	^B	STX	Start of Text - Início do texto
0000 0011	03	03	^C	ETX	End of Text - Fim do texto
0000 0100	04	04	^D	EOT	End of Tape - Fim de fita
0000 0101	05	05	^E	ENQ	Enquire - Interroga identidade do terminal
0000 0110	06	06	^F	ACK	Acknowledge - Reconhecimento
0000 0111	07	07	^G	BEL	Bell - Campanha
0000 1000	08	08	^H	BS	Back-space - Espaço atrás
0000 1001	09	09	^I	HT	Horizontal Tabulation - Tabulação horizontal
0000 1010	10	0A	^J	LF	Line-Feed - Alimenta linha
0000 1011	11	0B	^K	VT	Vertical Tabulation - Tabulação vertical
0000 1100	12	0C	^L	FF	Form-Feed - Alimenta formulário
0000 1101	13	0D	^M	CR	Carriage-Return - Retorno do carro (enter)
0000 1110	14	0E	^N	SO	Shift-Out - Saída do shift (passa a usar caracteres de baixo da tecla - minúsculas, etc.)
0000 1111	15	0F	^O	SI	Shift-In - Entrada no shift (passa a usar caracteres de cima da tecla: maiúsculas, caracteres especiais, etc.)
0001 0000	16	10	^P	DLE	Data-Link Escape
0001 0001	17	11	^Q	DC1	Device-Control 1
0001 0010	18	12	^R	DC2	Device-Control 2
0001 0011	19	13	^S	DC3	Device-Control 3
0001 0100	20	14	^T	DC4	Device-Control 4
0001 0101	21	15	^U	NAK	Neg-Acknowledge - Não-reconhecimento
0001 0110	22	16	^V	SYN	Synchronous Idle
0001 0111	23	17	^W	ETB	End-of-Transmission Block
0001 1000	24	18	^X	CAN	Cancel
0001 1001	25	19	^Y	EM	End-Of-Medium
0001 1010	26	1A	^Z	SUB	Substitute
0001 1011	27	1B	^[ESC	Escape
0001 1100	28	1C	^\ ^	FS	File Separator
0001 1101	29	1D	^] ^	GS	Group Separator
0001 1110	30	1E	^^ ^	RS	Record Separator
0001 1111	31	1F	^_ ^	US	Unit Separator
0111 1111	127	7F	^? ^	DEL	Delete

Fonte: <http://pt.wikipedia.org/wiki/ASCII>

Caracteres imprimíveis

Binário	Decimal	Hexa	Glifo
0010 0000	32	20	
0010 0001	33	21	!
0010 0010	34	22	"
0010 0011	35	23	#
0010 0100	36	24	\$
0010 0101	37	25	%
0010 0110	38	26	&
0010 0111	39	27	'
0010 1000	40	28	(
0010 1001	41	29)
0010 1010	42	2A	*
0010 1011	43	2B	±
0010 1100	44	2C	ˆ
0010 1101	45	2D	-
0010 1110	46	2E	˘
0010 1111	47	2F	/
0011 0000	48	30	0
0011 0001	49	31	1
0011 0010	50	32	2
0011 0011	51	33	3
0011 0100	52	34	4
0011 0101	53	35	5
0011 0110	54	36	6
0011 0111	55	37	7
0011 1000	56	38	8
0011 1001	57	39	9
0011 1010	58	3A	:
0011 1011	59	3B	;
0011 1100	60	3C	≤
0011 1101	61	3D	≡
0011 1110	62	3E	≥
0011 1111	63	3F	?

Tabela 4 – Caracteres Imprimíveis da Tabela ASCII

Binário	Decimal	Hexa	Glifo
0100 0000	64	40	@
0100 0001	65	41	A
0100 0010	66	42	B
0100 0011	67	43	C
0100 0100	68	44	D
0100 0101	69	45	E
0100 0110	70	46	F
0100 0111	71	47	G
0100 1000	72	48	H
0100 1001	73	49	I
0100 1010	74	4A	J
0100 1011	75	4B	K
0100 1100	76	4C	L
0100 1101	77	4D	M
0100 1110	78	4E	N
0100 1111	79	4F	O
0101 0000	80	50	P
0101 0001	81	51	Q
0101 0010	82	52	R
0101 0011	83	53	S
0101 0100	84	54	T
0101 0101	85	55	U
0101 0110	86	56	V
0101 0111	87	57	W
0101 1000	88	58	X
0101 1001	89	59	Y
0101 1010	90	5A	Z
0101 1011	91	5B	[
0101 1100	92	5C	\
0101 1101	93	5D]
0101 1110	94	5E	^
0101 1111	95	5F	_

Binário	Decimal	Hexa	Glifo
0110 0000	96	60	`
0110 0001	97	61	a
0110 0010	98	62	b
0110 0011	99	63	c
0110 0100	100	64	d
0110 0101	101	65	e
0110 0110	102	66	f
0110 0111	103	67	g
0110 1000	104	68	h
0110 1001	105	69	i
0110 1010	106	6A	j
0110 1011	107	6B	k
0110 1100	108	6C	l
0110 1101	109	6D	m
0110 1110	110	6E	n
0110 1111	111	6F	o
0111 0000	112	70	p
0111 0001	113	71	q
0111 0010	114	72	r
0111 0011	115	73	s
0111 0100	116	74	t
0111 0101	117	75	u
0111 0110	118	76	v
0111 0111	119	77	w
0111 1000	120	78	x
0111 1001	121	79	y
0111 1010	122	7A	z
0111 1011	123	7B	{
0111 1100	124	7C	
0111 1101	125	7D	}
0111 1110			

Fonte: <http://pt.wikipedia.org/wiki/ASCII>

APÊNDICES

Apêndice I

Descrição: Script de criação do banco de dados “intragov” dentro do SGBD MySQL, com as tabelas, atributos e relacionamentos.

```

SET
@OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

DROP SCHEMA IF EXISTS `mydb` ;
CREATE SCHEMA IF NOT EXISTS `mydb`
DEFAULT CHARACTER SET latin1 ;
SHOW WARNINGS;
DROP SCHEMA IF EXISTS `intragov` ;
CREATE SCHEMA IF NOT EXISTS `intragov` ;
SHOW WARNINGS;
USE `mydb` ;
USE `intragov` ;

-----
-- Table `Pessoas`
-----
DROP TABLE IF EXISTS `Pessoas` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Pessoas` (
  `Pessoa_Cod` INT NOT NULL ,
  `Pessoa_Nome` VARCHAR(60) NULL ,
  `Pessoa_Matricula` VARCHAR(45) NULL ,
  `Pessoa_Email` VARCHAR(45) NULL ,
  `Pessoa_Email2` VARCHAR(45) NULL ,
  `Pessoa_Messenger` VARCHAR(45) NULL ,
  `Pessoa_Telefone1` VARCHAR(45) NULL ,
  `Pessoa_Telefone2` VARCHAR(45) NULL ,
  `Pessoa_Telefone3` VARCHAR(45) NULL ,
  PRIMARY KEY (`Pessoa_Cod`))
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `Estados`
-----
DROP TABLE IF EXISTS `Estados` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Estados` (
  `Estado_Cod` INT NOT NULL
  AUTO_INCREMENT ,
  `Estado_Nome` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`Estado_Cod`))
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `Cidades`
-----
DROP TABLE IF EXISTS `Cidades` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Cidades` (
  `Cidade_Cod` INT NOT NULL
  AUTO_INCREMENT ,
  `Estados_Estado_Cod` INT NOT NULL ,
  `Cidade_Nome` VARCHAR(60) NOT NULL ,
  PRIMARY KEY (`Cidade_Cod`,
  `Estados_Estado_Cod`))
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `Bairros`
-----
DROP TABLE IF EXISTS `Bairros` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Bairros` (
  `Bairro_Cod` INT NOT NULL
  AUTO_INCREMENT ,
  `Cidades_Cidade_Cod` INT NOT NULL ,
  `Bairro_Nome` VARCHAR(45) NULL ,
  PRIMARY KEY (`Bairro_Cod`,
  `Cidades_Cidade_Cod`))
ENGINE = InnoDB;

SHOW WARNINGS;

```

```

-----
-- Table `Ruas`
-----
DROP TABLE IF EXISTS `Ruas` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Ruas` (
  `Rua_Cod` INT NOT NULL AUTO_INCREMENT
,
  `Bairros_Bairro_Cod` INT NOT NULL ,
  `Rua_Nome` VARCHAR(45) NOT NULL ,
  `Rua_Cep` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`Rua_Cod`))
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `Sites`
-----
DROP TABLE IF EXISTS `Sites` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Sites` (
  `Site_Cod` INT NOT NULL AUTO_INCREMENT
,
  `Ruas_Rua_Cod` INT NOT NULL ,
  `Rua_Numero` INT NOT NULL ,
  `Site_Nome` VARCHAR(45) NOT NULL ,
  `Site_Alias` VARCHAR(45) NULL ,
  `Site_Latitude` FLOAT NOT NULL ,
  `Site_Longitude` FLOAT NOT NULL ,
  `Site_Telefone` VARCHAR(45) NULL ,
  `Site_Pic` LONGBLOB NULL ,
  PRIMARY KEY (`Site_Cod`))
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `Bandas`
-----
DROP TABLE IF EXISTS `Bandas` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Bandas` (
  `Banda_Cod` INT NOT NULL
  AUTO_INCREMENT ,
  `Banda_Valor` INT NULL ,
  PRIMARY KEY (`Banda_Cod`))
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `Redes`
-----
DROP TABLE IF EXISTS `Redes` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Redes` (
  `Rede_Cod` INT NOT NULL
  AUTO_INCREMENT ,
  `Rede_IpInicial` VARCHAR(45) NULL ,
  `Rede_IpFinal` VARCHAR(45) NULL ,
  `Rede_Mascara` VARCHAR(45) NULL ,
  `Rede_Gateway` VARCHAR(45) NULL ,
  PRIMARY KEY (`Rede_Cod`))
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `LinkTipo`
-----
DROP TABLE IF EXISTS `LinkTipo` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `LinkTipo` (
  `LinkTipo_Cod` INT NOT NULL
  AUTO_INCREMENT ,
  `LinkTipo_Descricao` VARCHAR(45) NULL ,
  PRIMARY KEY (`LinkTipo_Cod`))
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `Links`
-----
DROP TABLE IF EXISTS `Links` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Links` (
  `Link_Cod` INT NOT NULL AUTO_INCREMENT
,
  `Redes_Rede_Cod` INT NOT NULL ,
  `LinkTipo_LinkTipo_Cod` INT NOT NULL ,
  `Link_Descricao` VARCHAR(45) NULL ,
  `Link_Id` INT NULL ,
  `Link_Lp1` VARCHAR(45) NULL ,
  `Link_Lp2` VARCHAR(45) NULL ,
  PRIMARY KEY (`Link_Cod`, `Redes_Rede_Cod`))
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `Equipamentos`

```

```
-----
DROP TABLE IF EXISTS `Equipamentos` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Equipamentos` (
  `Equip_Cod` INT NOT NULL
  AUTO_INCREMENT ,
  `Equip_Descricao` VARCHAR(45) NOT NULL ,
  `Equip_Marca` VARCHAR(45) NULL ,
  `Equip_Modelo` VARCHAR(45) NULL ,
  `Equip_Obs` VARCHAR(120) NULL ,
  PRIMARY KEY (`Equip_Cod`))
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-----
-- Table `Hosts`
-----
```

```
DROP TABLE IF EXISTS `Hosts` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Hosts` (
  `Host_Cod` INT NOT NULL AUTO_INCREMENT
  ,
  `Equipamentos_Equip_Cod` INT NOT NULL ,
  `Sites_Site_Cod` INT NOT NULL ,
  `Host_Ip` VARCHAR(45) NULL ,
  `Host_Mask` VARCHAR(45) NULL ,
  `Host_X` INT NULL ,
  `Host_Y` INT NULL ,
  `Host_Status` INT NULL ,
  PRIMARY KEY (`Host_Cod`))
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-----
-- Table `Acessos`
-----
```

```
DROP TABLE IF EXISTS `Acessos` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Acessos` (
  `Escolas_Escola_Cod` INT NOT NULL ,
  `Links_Link_Cod` INT NOT NULL ,
  PRIMARY KEY (`Escolas_Escola_Cod`,
  `Links_Link_Cod`))
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-----
-- Table `Links_has_Bandas`
-----
```

```
-----
DROP TABLE IF EXISTS `Links_has_Bandas` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS
`Links_has_Bandas` (
  `Links_Link_Cod` INT NOT NULL ,
  `Bandas_Banda_Cod` INT NOT NULL ,
  `Banda_Sentido` VARCHAR(45) NULL ,
  PRIMARY KEY (`Links_Link_Cod`,
  `Bandas_Banda_Cod`))
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-----
-- Table `Operadores`
-----
```

```
DROP TABLE IF EXISTS `Operadores` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Operadores` (
  `Pessoas_Pessoa_Cod` INT NOT NULL ,
  `Operador_Senha` VARCHAR(45) NULL ,
  `Operador_Acesso` INT NULL ,
  PRIMARY KEY (`Pessoas_Pessoa_Cod`))
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-----
-- Table `Logs`
-----
```

```
DROP TABLE IF EXISTS `Logs` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Logs` (
  `Log_Cod` INT NOT NULL AUTO_INCREMENT
  ,
  `Hosts_Host_Cod` INT NOT NULL ,
  `Log_Descricao` VARCHAR(45) NULL ,
  `Log_Time` DATETIME NULL ,
  PRIMARY KEY (`Log_Cod`))
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-----
-- Table `Tipos`
-----
```

```
DROP TABLE IF EXISTS `Tipos` ;
```

```
SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Tipos` (
```

```

`Tipo_Cod` INT NOT NULL AUTO_INCREMENT
,
`Tipo_Descricao` VARCHAR(45) NULL ,
PRIMARY KEY (`Tipo_Cod`))
ENGINE = InnoDB;

```

```
SHOW WARNINGS;
```

```
-----
-- Table `Ocorrencias`
-----
```

```
DROP TABLE IF EXISTS `Ocorrencias` ;
```

```
SHOW WARNINGS;
```

```
CREATE TABLE IF NOT EXISTS `Ocorrencias` (
  `Ocorrencia_Cod` INT NOT NULL
  AUTO_INCREMENT ,
  `Tipo_Tipo_Cod` INT NOT NULL ,
  `Ocorrencia_Descricao` VARCHAR(45) NULL ,
  `Ocorrencia_Time` DATETIME NULL ,
  `Ocorrencia_Fim` DATETIME NULL ,
  `Operadores_Pessoas_Pessoa_Cod` INT NOT
  NULL ,
  `Sites_Site_Cod` INT NOT NULL ,
  PRIMARY KEY (`Ocorrencia_Cod`))
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-----
-- Table `Operadores_has_Sites`
-----
```

```
DROP TABLE IF EXISTS `Operadores_has_Sites` ;
```

```
SHOW WARNINGS;
```

```
CREATE TABLE IF NOT EXISTS
```

```
`Operadores_has_Sites` (
  `Operadores_Pessoas_Pessoa_Cod` INT NOT
  NULL ,
  `Sites_Site_Cod` INT NOT NULL ,
  PRIMARY KEY
  (`Operadores_Pessoas_Pessoa_Cod`,
  `Sites_Site_Cod`))
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
```

```
SET
```

```
FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_
CHECKS;
```

```
SET
```

```
UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;U
NIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Apêndice II

Descrição: Classe principal (Main), invocada na abertura do sistema GEOMON_IPv4.

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package geomon;

import geomon.Gui.main;
import java.io.IOException;
import javax.swing.JDesktopPane;

/**
 *
 * @author daniel
 */
public class Main
{

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException {

        // Abre Interface Gráfica
        main geo=new main ();
        JDesktopPane deskPanel = new JDesktopPane();
        geo.add(deskPanel);
        geo.setVisible(true);

    }

}
```

Apêndice III

Descrição: Classe `shapefileComponent.java`, invocada na plotagem dos mapas, tratando características de um mapa específico. Parâmetros de métricas de qualidade são definidos nesta classe, imprimindo as cores dos pontos em conformidade com os padrões de qualidade definidos.

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package geomon.Gis;

import java.awt.*;
import java.awt.Dimension;
import java.awt.geom.AffineTransform;
import java.io.IOException;
import javax.swing.JComponent;
import org.geotools.data.shapefile.ShapefileDataStore;
import org.geotools.map.*;
import org.geotools.renderer.shape.ShapefileRenderer;
import org.geotools.styling.*;
import com.vividsolutions.jts.geom.*;
import geomon.dao.HostDao;
import geomon.dao.siteDao;
import geomon.model.hostModel;
import geomon.model.siteModel;
import java.util.Vector;
import javax.swing.JOptionPane;

/**
 *
 * @author Daniel
 */
public class ShapefileComponent extends JComponent {

    private ShapefileDataStore sds;
    private ShapefileRenderer renderer;
    private Envelope envelope;
    private final Coordinate center;
    public double zoom;
    public double deltaX;
    public double deltaY;
    private int oper;

```

```

private int width = 800;
private int height = 600;
private double mousex;
private double mousey;

private hostModel host;
private Vector hostLista;
private siteModel site;
private Vector siteLista;

public ShapefileComponent(ShapefileDataStore sds) throws IOException
{
this.sds = sds;
// Criamos um Style
StyleBuilder sb = new StyleBuilder();
// Criamos símbolos para a linha
LineSymbolizer lineSymb = (LineSymbolizer) sb.createLineSymbolizer(Color.BLUE, 1);
Style style = sb.createStyle(lineSymb);
// Criamos um MapContext com os dados e um estilo
MapContext mc = new DefaultMapContext();
mc.addLayer(sds.getFeatureSource(),style);
// Agora com o contexto criamos um ShapefileRenderer
renderer = new ShapefileRenderer(mc);
envelope = mc.getLayerBounds();
center = new Coordinate(((envelope.getMinX()+envelope.getMaxX())/2),
((envelope.getMinY()+envelope.getMaxY())/2));
zoom = 1;
}
public Dimension getMaximumSize() { return getPreferredSize(); }
public Dimension getMinimumSize() { return getPreferredSize(); }
public Dimension getPreferredSize() { return new Dimension(getWidth(), getHeight()); }

protected void paintComponent(Graphics g)
{
Graphics2D g2d = (Graphics2D)g;
g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

g2d.setColor(Color.BLACK);
g2d.fillRect(0,0, getWidth(), getHeight());

// Plota no gráfico os hosts cadastrados com coordenadas relativas (Apenas se oper=1)

```

```

if (oper==1)
{
siteLista = new Vector();
siteLista = siteDao.listar();

hostLista = new Vector();
hostLista = HostDao.listar();

for (int i=0;i<(int)hostLista.size();i++)
{
if
(((hostModel)hostLista.get(i)).getHost_X()!=0&&((hostModel)hostLista.get(i)).getHost_Y()!=0)
{
if (((hostModel)hostLista.get(i)).getHost_Status()==0) g2d.setColor(Color.RED);
else g2d.setColor(Color.GREEN);

g2d.fillOval(((hostModel)hostLista.get(i)).getHost_X(),((hostModel)hostLista.get(i)).getHost_Y(
), 12,12);
g2d.setColor(Color.white);
for (int ii=0;ii<(int)siteLista.size();ii++)
{
if
((((hostModel)hostLista.get(ii)).getEscola_Codigo()==((siteModel)siteLista.get(ii)).getSite_cod(
)))
{
g2d.drawString(((siteModel)siteLista.get(ii)).getSite_Nome(),
(((hostModel)hostLista.get(i)).getHost_X()-10, (((hostModel)hostLista.get(i)).getHost_Y()-5);
}
}
}
}

if (oper==0)
{
g2d.setColor(Color.RED);
g2d.fillOval((int)mousex-10,(int)mousey-60, 15,15);
g2d.setColor(Color.white);
g2d.drawString("Selecione o Ponto Relativo",(int)mousex-30,(int)mousey-80);
}

// Uma AffineTransform que mapeie dados no shapefile com a área para plotagem
AffineTransform at = new AffineTransform();
// Calculamos a escala

```

```

double escala = (Math.min(getWidth()/envelope.getWidth(),
getHeight()/envelope.getHeight())
*zoom);
// Fazemos a translação para o centro do componente.
at.translate (getWidth()/2,getHeight()/2);
// Mudamos a escala vertical para corrigir a orientação.
at.scale(escala,-escala);
// Fazemos a translação para o centro da geometria.
at.translate((-center.x)+deltaX,(-center.y)+deltaY);

g2d.drawString("X:"+getMouseX(), 10, 10);
g2d.drawString("Y:"+getMouseY(), 10, 20);
// Pintamos a geometria no componente.
renderer.paint(g2d,getBounds(),at);
}

public void zoomIn()
{
if (zoom <4.0) zoom=zoom+.1;
else JOptionPane.showMessageDialog(this,"Tamanho Máximo");
}

public void zoomOut()
{
if (zoom >0.1) zoom=zoom-.1;
else JOptionPane.showMessageDialog(this,"Tamanho MÍmimo");
}

public void zoomDefault()
{
zoom=1;
}

public void deltaXInc()
{
if (deltaX<5) deltaX=deltaX+.1;
else JOptionPane.showMessageDialog(this,"Fim da translação");
}

public void deltaXDec()
{
if (deltaX>-5)deltaX=deltaX-.1;
else JOptionPane.showMessageDialog(this,"Fim da translação");
}

public void deltaYInc()

```

```
{
if (deltaY<5) deltaY=deltaY+.1;
else JOptionPane.showMessageDialog(this,"Fim da translação");
}

public void deltaYDec()
{
if (deltaY>-5)deltaY=deltaY-.1;
else JOptionPane.showMessageDialog(this,"Fim da translação");
}

public void deltaCenter()

{
    deltaX=0;
    deltaY=0;
}

/**
 * @return the width
 */
public int getWidth() {
    return width;
}

/**
 * @param width the width to set
 */
public void setWidth(int width) {
    this.width = width;
}

/**
 * @return the height
 */
public int getHeight() {
    return height;
}

/**
 * @param height the height to set
 */
public void setHeight(int height) {
    this.height = height;
}
```

```
/**
 * @return the mousex
 */
public double getMousex() {
    return mousex;
}
/**
 * @param mousex the mousex to set
 */
public void setMousex(double mousex) {
    this.mousex = mousex;
}

/**
 * @return the mousey
 */
public double getMousey() {
    return mousey;
}

/**
 * @param mousey the mousey to set
 */
public void setMousey(double mousey) {
    this.mousey = mousey;
}

/**
 * @return the oper
 */
public int getOper() {
    return oper;
}

/**
 * @param oper the oper to set
 */
public void setOper(int oper) {
    this.oper = oper;
}
}
```

Apêndice IV

Descrição: Classe Java responsável pela Interface gráfica do módulo Scan. Ao pressionar o botão monitoramento, são ativadas outras rotinas concorrentes dentro do sistema. Processos de acesso à rede, coleta de métricas e inserção no banco de dados ocorrem simultaneamente.

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * NewJInternalFrame.java
 *
 * Created on 20/08/2010, 10:58:26
 */

package geomon.Gui;

import geomon.dao.HostDao;
import geomon.dao.logDao;
import geomon.dao.siteDao;
import geomon.model.hostModel;
import geomon.model.logModel;
import geomon.model.siteModel;
import java.awt.Toolkit;
import java.util.Vector;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import monitor.monitor;

/**
 *
 * @author daniel
 */
public class scan extends javax.swing.JInternalFrame {

    public main principal;
    private Vector listaTabela;
    private monitor mon;
    public int controle;
    private DefaultTableModel model;

```

```

/** Creates new form NewJInternalFrame */
public scan(JFrame objeto)
{

    super("Scan de Hosts",true,true,true,true);

    // Carrega tela principal junto
    this.principal=(main) objeto;

    // Inicializa componentes
    initComponents();

    // ajusta frame interno no frame principal
    System.out.println((int)Toolkit.getDefaultToolkit().getScreenSize().getHeight()+"
"+(int)Toolkit.getDefaultToolkit().getScreenSize().getWidth());
    this.setSize((int)Toolkit.getDefaultToolkit().getScreenSize().getWidth()-
700,(int)Toolkit.getDefaultToolkit().getScreenSize().getHeight()-300);

    // Limpa Avisos
    jLabel3.setText("Sistema em espera");

    // Carrega Tabela
    lista();
    mon=new monitor();

}

public void lista()
{

    // Formatar Tabela
    model = new DefaultTableModel();
    model.addColumn("Sequencia");
    model.addColumn("IP");
    model.addColumn("Local");
    model.addColumn("Resposta");
    model.addColumn("Hora");
    jTable1.setModel(model);

}

private void action()
{

```

```
// Verifica se há monitoramento em andamento
if (mon.isAlive())
{
    controle=1;
    if (mon.isAlive()) jLabel3.setText("Monitorando Hosts");
}
else
{
    controle=0;
}

// inicia um novo monitoramento, apenas se não houver ativo
if (controle==0)
{
    controle=1;
    Vector listaHosts=new Vector();

    // Carrega Vetor listaHosts
    jLabel3.setText("Carregando lista de hosts");
    listaHosts = HostDao.listar();

    if (listaHosts!=null)
    {
        // Publica ação Usuário
        jLabel3.setText("Lista criada com sucesso");
    }

    // Inicia Thread Monitoramento
    mon=new monitor();
    mon.start();

    // Publica ação Usuário
    if (mon.isAlive()) jLabel3.setText("Monitorando Hosts");

    // inicia Thread Atualização Painele looping infinito
    new Thread (new Runnable()
    {
        public void run()
        {
            try
            {
                while (controle==1)
                {
                    if (mon.isAlive())
```

```

        {
            // Atualiza jbar e tabela (enquanto Thread está ativa)
            checkUpdates();
        }
        else
        {
            // Se chegou ao fim da lista de hosts, inicia novo monitoramento
            mon=new monitor();
            mon.start();
        }
        Thread.sleep( 1000 * 4 ); // dorme 3 segundos antes da próxima atualização da barra e
tabela
    }
}
catch( Exception e )
{
    e.printStackTrace();
}
}
}).start();

}

else
{
    JOptionPane.showMessageDialog(this,"Monitoramento já inicializado");
}

}

private void stop()
{

mon.stop();
controle=0;
jLabel3.setText("Sistema em espera");

}

private void checkUpdates()
{

// Carrega os logs do SGBD (Apenas Ultimos 15)
Vector lista=new Vector();
int logSize = logDao.logSize();
lista = logDao.listarLast();

```

```

// Carrega os hosts cadastrados no SGBD
Vector hosts=new Vector();
hosts = HostDao.listar();

// Carrega os sites cadastrados no SGBD
Vector sites=new Vector();
sites = siteDao.listar();

// trabalha a jprogress bar
jProgressBar1.setMaximum(hosts.size());
jProgressBar1.setMinimum(0);

// trabalha a tabela
lista();
for (int i=14; i>1; i--)
{
Vector linha = new Vector();
jProgressBar1.setValue(((logModel)lista.get(i)).getLog_host());
linha.add(((logModel)lista.get(i)).getLog_cod());

// busca dados específicos do host relacionados ao registro no vetor LOG
for (int j=0; j<hosts.size(); j++)
{
if (((hostModel)hosts.get(j)).getHost_Cod()==((logModel)lista.get(i)).getLog_host())
{
linha.add(((hostModel)hosts.get(j)).getHost_Ip());

// busca dados específicos do site relacionados ao registro no vetor sites
for (int k=0; k<sites.size(); k++)
{
if
(((siteModel)sites.get(k)).getSite_cod()==((hostModel)hosts.get(j)).getEscola_Codigo())
{
linha.add(((siteModel)sites.get(k)).getSite_Nome());
}
}
}
}

// busca dados específicos do host relacionados ao registro no vetor sites
linha.add(((logModel)lista.get(i)).getLog_Descricao());
linha.add(((logModel)lista.get(i)).getLog_Time());
model.addRow(linha);
}

```

```

jTable1.setModel(model);

}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jButton1 = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jButton2 = new javax.swing.JButton();
    jButton3 = new javax.swing.JButton();
    progressBar1 = new javax.swing.JProgressBar();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();

    addContainerListener(new java.awt.event.ContainerAdapter() {
        public void componentRemoved(java.awt.event.ContainerEvent evt) {
            formComponentRemoved(evt);
        }
    });
    getContentPane().setLayout(null);

    jButton1.setText("Fechar");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });
    getContentPane().add(jButton1);
    jButton1.setBounds(10, 430, 520, 23);

    jLabel1.setText("Lista de Hosts Cadastrados");
    getContentPane().add(jLabel1);
    jLabel1.setBounds(105, 11, 320, 14);

```

```

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jTable1.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_ALL_COLUMNS);
jScrollPane1.setViewportView(jTable1);

getContentPane().add(jScrollPane1);
jScrollPane1.setBounds(10, 40, 520, 230);

jLabel2.setText("Ação:");
getContentPane().add(jLabel2);
jLabel2.setBounds(10, 280, 40, 14);

jLabel3.setText("Ação");
getContentPane().add(jLabel3);
jLabel3.setBounds(60, 280, 460, 14);

jButton2.setText("Iniciar Monitoramento");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
getContentPane().add(jButton2);
jButton2.setBounds(10, 310, 520, 23);

jButton3.setText("Parar Monitoramento");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
getContentPane().add(jButton3);
jButton3.setBounds(10, 400, 520, 23);
getContentPane().add(jProgressBar1);
jProgressBar1.setBounds(10, 340, 520, 20);

```

```

jLabel4.setText("0 %");
getContentPane().add(jLabel4);
jLabel4.setBounds(10, 370, 30, 14);

jLabel5.setText("100 %");
getContentPane().add(jLabel5);
jLabel5.setBounds(490, 370, 40, 14);

pack();
} // </editor-fold>

private void formComponentRemoved(java.awt.event.ContainerEvent evt) {
principal.setSubFrame(null);
this.dispose();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

if (mon.isAlive())
{
JOptionPane.showMessageDialog(this,"Aguarde ou interrompa o monitoramento");
}
else
{
principal.setSubFrame(null);
this.dispose();
}
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// Inicia Varredura
controle=1;
action();

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// Interrompe Varredura
stop();

}

// Variables declaration - do not modify

```

```
private javax.swing.JButton jButton1;  
private javax.swing.JButton jButton2;  
private javax.swing.JButton jButton3;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JProgressBar jProgressBar1;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTable jTable1;  
// End of variables declaration  
  
}
```

Apêndice V

Descrição: Exemplo de classe model (modelo). Considerando a utilização de modelagem (MVC), a troca de objetos dentro do sistema facilita o trabalho em camadas, mas os objetos devem refletir os atributos tratados no SGBD.

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package geomon.model;
/**
 *
 * @author daniel
 */
public class hostModel {

    private int Host_Cod;
    private int Equipamento_Codigo;
    private int Escola_Codigo;
    private String Host_Ip;
    private String Host_Mask;
    private int Host_X;
    private int Host_Y;
    private int Host_Status;

    public hostModel()
    {
    }
    /**
     * @return the Host_Cod
     */
    public int getHost_Cod() {
        return Host_Cod;
    }
    /**
     * @param Host_Cod the Host_Cod to set
     */
    public void setHost_Cod(int Host_Cod) {
        this.Host_Cod = Host_Cod;
    }
    /**
     * @return the Equipamento_Codigo
     */

```

```
public int getEquipamento_Codigo() {
    return Equipamento_Codigo;
}

/**
 * @param Equipamento_Codigo the Equipamento_Codigo to set
 */
public void setEquipamento_Codigo(int Equipamento_Codigo) {
    this.Equipamento_Codigo = Equipamento_Codigo;
}

/**
 * @return the Escola_Codigo
 */
public int getEscola_Codigo() {
    return Escola_Codigo;
}

/**
 * @param Escola_Codigo the Escola_Codigo to set
 */
public void setEscola_Codigo(int Escola_Codigo) {
    this.Escola_Codigo = Escola_Codigo;
}

/**
 * @return the Host_Ip
 */
public String getHost_Ip() {
    return Host_Ip;
}

/**
 * @param Host_Ip the Host_Ip to set
 */
public void setHost_Ip(String Host_Ip) {
    this.Host_Ip = Host_Ip;
}

/**
 * @return the Host_Mask
 */
public String getHost_Mask() {
    return Host_Mask;
}

/**
 * @param Host_Mask the Host_Mask to set
 */
```

```
public void setHost_Mask(String Host_Mask) {
    this.Host_Mask = Host_Mask;
}

/**
 * @return the Host_X
 */
public int getHost_X() {
    return Host_X;
}

/**
 * @param Host_X the Host_X to set
 */
public void setHost_X(int Host_X) {
    this.Host_X = Host_X;
}

/**
 * @return the Host_Y
 */
public int getHost_Y() {
    return Host_Y;
}

/**
 * @param Host_Y the Host_Y to set
 */
public void setHost_Y(int Host_Y) {
    this.Host_Y = Host_Y;
}

/**
 * @return the Host_Status
 */
public int getHost_Status() {
    return Host_Status;
}

/**
 * @param Host_Status the Host_Status to set
 */
public void setHost_Status(int Host_Status) {
    this.Host_Status = Host_Status;
}
}
```

Apêndice VI

Descrição: Parte do código fonte da classe ShapeFile.java, responsável por tratar as cores que serão plotadas em objetos do tipo g2d (círculos) no mapa.

```

for (int i=0;i<(int)hostLista.size();i++)
{
    // plota apenas o ponto correspondente ao mapa
    if (((hostModel)hostLista.get(i)).getHost_Mapa()==tipo)
    {
        if
(((hostModel)hostLista.get(i)).getHost_X()!=0&&((hostModel)hostLista.get(i)).getHost_Y()!=0)
        {
            // plota cor correspondente a qualidade de sinal
            // determina cor
            if (((hostModel)hostLista.get(i)).getHost_Status()<0)g2d.setColor(Color.RED);
            if
(((hostModel)hostLista.get(i)).getHost_Status())>=0&&((hostModel)hostLista.get(i)).getHost_Sta
tus()<100)g2d.setColor(Color.GREEN);
            if
(((hostModel)hostLista.get(i)).getHost_Status())>=100&&((hostModel)hostLista.get(i)).getHost_
Status()<200)g2d.setColor(Color.YELLOW);
            if
(((hostModel)hostLista.get(i)).getHost_Status())>=200&&((hostModel)hostLista.get(i)).getHost_
Status()<=300)g2d.setColor(Color.ORANGE);
            if
(((hostModel)hostLista.get(i)).getHost_Status())>=300&&((hostModel)hostLista.get(i)).getHost_
Status()<500)g2d.setColor(Color.RED);

g2d.fillOval(((hostModel)hostLista.get(i)).getHost_X()),((hostModel)hostLista.get(i)).getHost_Y(
), 12,12);

```


Apêndice VII

Descrição: Script de inserção no banco de dados dos principais dados da rede Intragov do Centro Paula Souza.

```
/* Insere dados na tabela Estados */
```

```
insert into estados values (null,'SP');
```

```
/* Insere cidades atendidas pelo CEETEPS */
```

```
/* Cidades do estado de SP */
```

```
insert into cidades values (null,1,'Adamantina');
insert into cidades values (null,1,'Aguai');
insert into cidades values (null,1,'Americana');
insert into cidades values (null,1,'Amparo');
insert into cidades values (null,1,'Andradina');
insert into cidades values (null,1,'Araçatuba');
insert into cidades values (null,1,'Araraquara');
insert into cidades values (null,1,'Araras');
insert into cidades values (null,1,'Assis');
insert into cidades values (null,1,'Atibaia');
insert into cidades values (null,1,'Avaré');
insert into cidades values (null,1,'Barra Bonita');
insert into cidades values (null,1,'Barretos');
insert into cidades values (null,1,'Barueri');
insert into cidades values (null,1,'Batatais');
insert into cidades values (null,1,'Bauru');
insert into cidades values (null,1,'Bebedouro');
insert into cidades values (null,1,'Birigui');
insert into cidades values (null,1,'Botucatu');
insert into cidades values (null,1,'Bragança Paulista');
insert into cidades values (null,1,'Cabrália Paulista');
insert into cidades values (null,1,'Caçapava');
insert into cidades values (null,1,'Cachoeira Paulista');
insert into cidades values (null,1,'Cafelândia');
insert into cidades values (null,1,'Cajamar');
insert into cidades values (null,1,'Campinas');
insert into cidades values (null,1,'Campo Limpo Paulista');
insert into cidades values (null,1,'Cândido Mota');
insert into cidades values (null,1,'Capão Bonito');
insert into cidades values (null,1,'Caraguatatuba');
insert into cidades values (null,1,'Carapicuíba');
insert into cidades values (null,1,'Casa Branca');
insert into cidades values (null,1,'Catanduva');
insert into cidades values (null,1,'Cerqueira cesar');
insert into cidades values (null,1,'Cotia');
insert into cidades values (null,1,'Cruzeiro');
insert into cidades values (null,1,'Cubatão');
insert into cidades values (null,1,'Diadema');
insert into cidades values (null,1,'Dracena');
insert into cidades values (null,1,'Espírito Santo do Pinhal');
insert into cidades values (null,1,'Fernandópolis');
```

```
insert into cidades values (null,1,'Ferraz de Vasconcelos');
insert into cidades values (null,1,'Franca');
insert into cidades values (null,1,'Franco da Rocha');
insert into cidades values (null,1,'Garça');
insert into cidades values (null,1,'Guaratingueta');
insert into cidades values (null,1,'Guarujá');
insert into cidades values (null,1,'Guarulhos');
insert into cidades values (null,1,'Hortolândia');
insert into cidades values (null,1,'Ibitinga');
insert into cidades values (null,1,'Igarapava');
insert into cidades values (null,1,'Iguape');
insert into cidades values (null,1,'Ilha Solteira');
insert into cidades values (null,1,'Indaiatuba');
insert into cidades values (null,1,'Ipaussu');
insert into cidades values (null,1,'Itanhaém');
insert into cidades values (null,1,'Itapetininga');
insert into cidades values (null,1,'Itapeva');
insert into cidades values (null,1,'Itapira');
insert into cidades values (null,1,'Itaquaquecetuba');
insert into cidades values (null,1,'Itatiba');
insert into cidades values (null,1,'Itu');
insert into cidades values (null,1,'Jaboticabal');
insert into cidades values (null,1,'Jacarei');
insert into cidades values (null,1,'Jales');
insert into cidades values (null,1,'Jau');
insert into cidades values (null,1,'Jundiaí');
insert into cidades values (null,1,'Leme');
insert into cidades values (null,1,'Limeira');
insert into cidades values (null,1,'Lins');
insert into cidades values (null,1,'Marília');
insert into cidades values (null,1,'Matão');
insert into cidades values (null,1,'Mauá');
insert into cidades values (null,1,'Miguelópolis');
insert into cidades values (null,1,'Mirassol');
insert into cidades values (null,1,'Mococa');
insert into cidades values (null,1,'Mogi das Cruzes');
insert into cidades values (null,1,'Mogi Guaçu');
insert into cidades values (null,1,'Mogi Mirim');
insert into cidades values (null,1,'Monguagá');
insert into cidades values (null,1,'Monte Aprazível');
insert into cidades values (null,1,'Monte Mor');
insert into cidades values (null,1,'Novo Horizonte');
insert into cidades values (null,1,'Orlandia');
insert into cidades values (null,1,'Osasco');
insert into cidades values (null,1,'Osvaldo Cruz');
insert into cidades values (null,1,'Ourinhos');
insert into cidades values (null,1,'Palmital');
insert into cidades values (null,1,'Paraguaçu Paulista');
insert into cidades values (null,1,'Penápolis');
insert into cidades values (null,1,'Piedade');
insert into cidades values (null,1,'Pindamonhangaba');
insert into cidades values (null,1,'Piracicaba');
insert into cidades values (null,1,'Pirajú');
insert into cidades values (null,1,'Pirassununga');
insert into cidades values (null,1,'Poá');
```

```

insert into cidades values (null,1,'Porto Ferreira');
insert into cidades values (null,1,'Praia Grande');
insert into cidades values (null,1,'Presidente Prudente');
insert into cidades values (null,1,'Presidente Venceslau');
insert into cidades values (null,1,'Quatá');
insert into cidades values (null,1,'Rancharia');
insert into cidades values (null,1,'Ribeirão Pires');
insert into cidades values (null,1,'Rio Claro');
insert into cidades values (null,1,'Rio das Pedras');
insert into cidades values (null,1,'Santa Bárbara D'Oeste');
insert into cidades values (null,1,'Santa Cruz do Rio Pardo');
insert into cidades values (null,1,'Santa Isabel');
insert into cidades values (null,1,'Santa Rita do Passa Quatro');
insert into cidades values (null,1,'Santana do Parnaíba');
insert into cidades values (null,1,'Santo André');
insert into cidades values (null,1,'Santos');
insert into cidades values (null,1,'São Bernardo do Campo');
insert into cidades values (null,1,'São Caetano do Sul');
insert into cidades values (null,1,'São Carlos');
insert into cidades values (null,1,'São Joaquim da Barra');
insert into cidades values (null,1,'São José do Rio Pardo');
insert into cidades values (null,1,'São José do Rio Preto');
insert into cidades values (null,1,'São José dos Campos');
insert into cidades values (null,1,'São Manuel');
insert into cidades values (null,1,'São Paulo');
insert into cidades values (null,1,'São Roque');
insert into cidades values (null,1,'São Sebastião');
insert into cidades values (null,1,'São Simão');
insert into cidades values (null,1,'São Vicente');
insert into cidades values (null,1,'Serrana');
insert into cidades values (null,1,'Sertãozinho');
insert into cidades values (null,1,'Sorocaba');
insert into cidades values (null,1,'Suzano');
insert into cidades values (null,1,'Taquaritinga');
insert into cidades values (null,1,'Taquarivaí');
insert into cidades values (null,1,'Tatuí');
insert into cidades values (null,1,'Taubaté');
insert into cidades values (null,1,'Teodoro Sambaio');
insert into cidades values (null,1,'Tupã');
insert into cidades values (null,1,'Vargem Grande do Sul');
insert into cidades values (null,1,'Vera Cruz');
insert into cidades values (null,1,'Votorantim');
insert into cidades values (null,1,'Votuporanga');

```

/ Insere Bairros Modelos atendidas pelo CEETEPS */*

/ Cidade de Adamantina - Bairros */*

```

insert into bairros values (null,1, 'Boa Vista');
insert into bairros values (null,1, 'Vila Jamil de Lima');

```

/ Cidade de Aguai - Bairros */*

```

insert into bairros values (null,2, 'Sem Bairro');

```

```

/* Cidade de Americana - Bairros */
insert into bairros values (null,3, 'Jardim Nossa Senhora de Fátima');
insert into bairros values (null,3, 'Vila Amorim');

/* Cidade de Amparo - Bairros */
insert into bairros values (null,4, 'Centro');

/* Cidade de Andradina - Bairros */
insert into bairros values (null,5, 'Vila Planalto');

/* Cidade de Araçatuba - Bairros */
insert into bairros values (null,6, 'Jardim Ipanema');

/* Insere Ruas Modelos atendidas pelo CEETEPS */

/* Cidade de Adamantina - Ruas */

insert into ruas values (null,1, 'Estrada Seis',17800000);
insert into ruas values (null,1, 'Rua Libero Badaró',17800000);

/* Cidade de Aguaí - Ruas */
insert into ruas values (null,2, 'Rua Joaquim José, esquina Carlos Gomes',00000000);

/* Cidade de Americana - Ruas */
insert into ruas values (null,4, 'Avenida Nossa Senhora de Fátima',13478000);
insert into ruas values (null,5, 'Rua Emilio de Menezes, sn - Gleba B',13469111);

/* Cidade de Amparo - Ruas */
insert into ruas values (null,6, 'Rua Sete de Setembro',13900372);

/* Cidade de Andradina - Ruas */
insert into ruas values (null,7,'Estr. Vic. Sebastião Lourenço da Silva, Km 11',16900970);

/* Cidade de Araçatuba - Ruas */
insert into ruas values (null,8,'Avenida Prestes Maia',16052-045);

/* Insere Sites (Escolas) do CEETEPS */

/* Cidade de Adamantina - ETECs */

insert into sites values (null,1,0,'ETEC Eng. Herval Bellusci','apelido',0.0001,0.0002,'(18) 3521-2494',null);
insert into sites values (null,2,600,'ETEC Professor Eudeicio Luiz Vicente','apelido',0.0001,0.0002,'(18) 3521-2565',null);

/* Cidade de Aguaí - ETECs */

insert into sites values (null,3,600,'ETEC de Aguaí','apelido',0.0001,0.0002,'(00) 0000-0000',null);

/* Cidade de Americana - ETECs e FATEC */

```

```
insert into sites values (null,4,600,'ETEC Polivalente de Americana','apelido',0.0001,0.0002,'(00) 0000-0000',null);
insert into sites values (null,5,600,'FATEC Americana','apelido',0.0001,0.0002,'(00) 0000-0000',null);

/* Cidade de Amparo - ETECs e FATEC */

insert into sites values (null,6,600,'ETEC João Belarmino','apelido',0.0001,0.0002,'(19) 3807-2288',null);

/* Cidade de Andradina - ETECs e FATEC */

insert into sites values (null,7,600,'ETEC Sebastiana Aug. de Moraes','apelido',0.0001,0.0002,'(18) 3722-3302',null);

/* Cidade de Araçatuba - ETECs e FATEC */

insert into sites values (null,8,1764,'ETEC Araçatuba','apelido',0.0001,0.0002,'(18) 3625-8677',null);
insert into sites values (null,8,1764,'FATEC Araçatuba','apelido',0.0001,0.0002,'(18) 3625-9914',null);

/* Insere Equipamentos */

/* Roteador Modelo */

insert into equipamentos values (null,'roteador padrão','marca padrao','modelo padrao',null);

/* Insere host */

/* Host 1 no local 1 (Adamantina), escola 1 e 2*/

insert into hosts values (null,1,1,'10.66.182.1','255.255.255.255',0,0,0);
insert into hosts values (null,1,2,'10.66.168.1','255.255.255.255',0,0,0);

/* Host 1 no local 2 (Aguai), escola 1 */

insert into hosts values (null,1,3,'10.67.222.1','255.255.255.255',0,0,0);

/* Host 1 no local 2 (Americana), escola 1 e 2 */

insert into hosts values (null,1,4,'10.66.98.1','255.255.255.255',0,0,0);
insert into hosts values (null,1,5,'10.66.10.1','255.255.255.255',0,0,0);

/* Host 1 no local 2 (Amparo), escola 1 e 2 */

insert into hosts values (null,1,6,'10.66.186.1','255.255.255.255',0,0,0);

/* Host 1 no local 2 (Andradina), escola 1 e 2 */

insert into hosts values (null,1,7,'10.66.122.1','255.255.255.255',0,0,0);

/* Host 1 no local 2 (Araçatuba), escola 1 e 2 */

insert into hosts values (null,1,8,'10.67.86.1','255.255.255.255',0,0,0);
insert into hosts values (null,1,9,'10.67.202.1','255.255.255.255',0,0,0);
```