

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIAS, GERÊNCIA E  
SEGURANÇA DE REDES DE COMPUTADORES

FÁBIO TORRES BALSEMÃO

**Gerência e Monitoramento de Redes Através  
de Dispositivos Móveis**

Trabalho de Conclusão apresentado como  
requisito parcial para a obtenção do grau de  
Especialista

Prof. ME. Henrique J. Brodbeck  
Orientador

Prof. Dr. Sérgio Luis Cechin  
Prof. Dr. Luciano Paschoal Gasparly  
Coordenadores do Curso

Porto Alegre, dezembro de 2008.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenadores do Curso: Profs. Sérgio Luis Cechin e Luciano Paschoal Gaspary

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## **AGRADECIMENTOS**

Agradeço a todos meus colegas de curso pela troca de experiências, pelo espírito de equipe e pelo sempre presente bom humor, fazendo da elaboração deste trabalho uma tarefa bem mais prazerosa. Minha gratidão também a Ida Rossi e ao Henrique Costa por toda presteza e um agradecimento especial a minha esposa Marjoli por todo incentivo, compreensão e boas vibrações.

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>6</b>
<b>LISTA DE FIGURAS.....</b>	<b>8</b>
<b>RESUMO.....</b>	<b>9</b>
<b>ABSTRACT.....</b>	<b>10</b>
<b>1 INTRODUÇÃO.....</b>	<b>11</b>
<b>2 HISTÓRICO.....</b>	<b>13</b>
2.1 Gerações.....	13
<b>3 APLICAÇÕES PARA DISPOSITIVOS MÓVEIS.....</b>	<b>15</b>
3.1 Internet Móvel.....	15
3.2 Smart-Clients.....	16
3.3 Sistemas Operacionais.....	16
3.3.1 Palm OS.....	16
3.3.2 Symbian.....	17
3.3.3 Windows Mobile.....	17
3.3.4 BREW (Binary Runtime Environment for Wireless).....	17
<b>4 ANÁLISE DE FERRAMENTAS PARA GERÊNCIA DE REDES.....</b>	<b>19</b>
4.1 Nagios.....	19
4.1.1 Requisitos do Sistema.....	20
4.1.2 Plugins.....	21
4.1.3 CGIs.....	21
4.2 Zenoss.....	22
4.2.1 Zenoss Dashboard.....	24
4.2.2 Google Maps.....	25
4.2.3 Monitorando remotamente dispositivos via SSH.....	26
4.2.4 Monitorando Dispositivos Windows.....	26
4.3 Cacti.....	26
4.4 Considerações Finais.....	28
<b>5 ESPECIFICAÇÃO.....</b>	<b>30</b>
5.1 Análise de requisitos.....	30
5.1.1 Visão Geral.....	30
5.1.2 Requisitos.....	30

<b>5.2 Casos de uso.....</b>	<b>31</b>
<b>5.3 Fluxo de operação .....</b>	<b>31</b>
5.3.1 Fluxo principal .....	31
5.3.2 Fluxos dos comandos Netstat e Ping .....	32
<b>5.4 Diagrama de Classes .....</b>	<b>32</b>
5.4.1 Classes Netstat e Ping .....	32
<b>6 MODELAGEM .....</b>	<b>34</b>
6.1 Classes .....	35
6.2 Modelo da Interface.....	36
<b>7 CONCLUSÃO .....</b>	<b>38</b>
<b>REFERÊNCIAS.....</b>	<b>39</b>

## LISTA DE ABREVIATURAS E SIGLAS

AJAX	Asynchronous Javascript And XML
API	Application Programming Interface
BDS	Brew Distribution System
CDMA	Code Division Multiple Acces
CGI	Common Gateway Interface
CPU	Central Processing Unit
DCOM	Distributed Component Object Model
EDGE (GSM)	Enhanced Data rates for GSM Evolution
GB	Gigabytes
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HTML	HiperText Markup Language
HTTP	HyperText Transfer Protocol
ICMP	Internet Control Message Protocol
IDE	Integrated Development Environment
IP	Internet Protocol
J2ME	Java Plataform, Micro Edition
Kbps	Kilobytes por segundo
MP3	MPEG-1/2 Audio Layer 3
NNTP	Network News Transfer Protocol
NRPE	Nagios Remote Plugin Executor
OID	Object Identifier
PDA	Personal Digital Assistants
POP3	Post Office Protocol
RRD	Round Robin Database
SMS	Short Message Service

SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TCP/IP	Transmission Control Protocol - Internet Protocol
URL	Uniform Resource Locator
XML	eXtensible Markup Language
XMP-RPC	Remote Procedure Call
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
W-CDMA3	Wideband - Code Division Multiple Access
WMI	Windows Management Instrumentation

## LISTA DE FIGURAS

Figura 4.1: Agente Nagios para Máquinas Windows.....	20
Figura 4.2: Agente Nagios para Máquinas Linux .....	21
Figura 4.3: Arquitetura detalhada do zenoss .....	24
Figura 4.4: Interface Zenoss .....	25
Figura 4.5: Princípios de operação do Cacti.....	27
Figura 4.6: Interface do Cacti .....	28
Figura 5.1: Executar comandos .....	31
Figura 5.2: Fluxo principal .....	31
Figura 5.3: Fluxo Netstat .....	32
Figura 5.4: Fluxo Ping .....	32
Figura 5.5: Classe Netstat.....	32
Figura 5.6: Classe Ping.....	32
Figura 6.1: Tela Inicial .....	36
Figura 6.2: Menu de Operações.....	36
Figura 6.3: Interface da Classe MMPing.....	37
Figura 6.4: Interface da Classe MMNetstat.....	37



## RESUMO

À medida que a complexidade do gerenciamento de redes aumenta, se faz necessária a adoção de ferramentas automatizadas para monitoração e controle. Através da análise de ferramentas de monitoração de redes de código aberto, este trabalho busca constatar a carência de funcionalidades adaptadas a dispositivos móveis, negligenciando-se a utilização destes no cotidiano do gerente de redes.

Apresenta-se um estudo sobre a utilização de telefones celulares e *smartphones* como ferramentas de trabalho para profissionais responsáveis pela gerência e segurança de redes, tendo como objetivo a elaboração de uma biblioteca de programas utilitários que auxiliem, remotamente, estes profissionais nas suas tarefas.

**Palavras-Chave:** Gerência de redes, telefones celulares, smartphones, mobilidade.

## **Management and Network Monitoring Through Mobile Devices**

### **ABSTRACT**

As the complexity of network management increases, the adoption of automated tools for monitoring and control is required. Through the analysis of open source tools for network monitoring, this paper searches for evidences of lack of suitable features in mobile devices, with causes the neglect of the devices in the daily work of the network manager.

This paper presents a study about the use of mobile phones and smartphones as work tools for professionals responsible for management and security of networks, with the goal of developing a library of utility programs that can help, remotely, these professionals in their tasks.

**Keywords:** Management of nets, mobile phones, smartphones, mobility.

# 1 INTRODUÇÃO

O crescimento das redes de computadores alcançou a esfera global, aumentando tanto em tamanho quanto em complexidade. Hoje é possível acessar informações que fisicamente estão a distâncias continentais e até mesmo interplanetárias. Não é necessário nem mesmo um cabo de rede, pois com o desenvolvimento das redes sem-fio, basta estar perto de um ponto de acesso, para estabelecer uma conexão.

Com o evento da globalização, empresas espalharam-se pelo mundo, atuando em diferentes áreas. O avanço constante da tecnologia permite e retroalimenta este processo. Cada vez mais se faz necessária a presença de mão-de-obra qualificada capaz de compreender os novos panoramas que surgem e evoluem rapidamente, às vezes mais rápido que a própria capacidade do mercado de gerar este profissional.

Um gerente de redes tem uma vasta quantidade de ferramentas que auxiliam sua atividade, seja no projeto de uma rede, seja na análise dos diferentes tipos de equipamentos e suas configurações ou até mesmo na cotidiana função de análise de tráfego de uma rede. Entretanto, um simples cabo com defeito, uma fibra quebrada ou um servidor mal configurado pode causar um problema de grandes proporções.

O fator tempo é outro elemento preocupante, pois quanto mais rápido for detectado um problema, maiores são as chances de este problema ser resolvido sem se colocar em risco a qualidade do serviço. É importante manter um bom grau de resiliência, principalmente em funções básicas de gerência.

Embora o acesso remoto tenha facilitado bastante a vida dos profissionais da área de redes, ainda assim é necessário todo um contexto que permita este artifício ser utilizado: um computador, software de acesso remoto, um ponto de acesso, etc. Este trabalho busca fornecer uma alternativa simples de auxílio ao gerenciamento de redes à distância, utilizando o telefone celular como ferramenta tecnológica de auxílio ao profissional, permitindo, através de uma biblioteca de funções programadas especialmente para esta mídia.

Percebe-se nitidamente que o telefone celular trouxe importantes mudanças tecnológicas e culturais. “Sua capacidade (do telefone celular) de permitir conexões em todo e qualquer lugar estendendo a voz e ouvido humano, representa uma facilidade que se incorporou ao nosso cotidiano, exercendo uma influência insuspeitada no comportamento cultural dos indivíduos” (LIMA, 2000, p.41).

O celular não pode mais ser visto como um telefone móvel, mas como um instrumento de interação móvel, que nos proporciona uma infinidade de informações e serviços.

Este trabalho procura então, agregar funcionalidades ao telefone celular através do desenvolvimento de um software voltado ao profissional de informática, na área de gerenciamento de redes. Para tanto, fez-se necessário conhecer, desde as origens destes dispositivos móveis, até os diferentes tipos de sistemas operacionais que neles rodam e suas diferentes características e propostas.

Além disso, dentro do universo de ferramentas de análise e gerenciamento de redes, escolheu-se para este estudo um grupo de software, que hoje, estão contidos no estado da arte do que se faz em matéria de gerenciamento de redes. Buscou-se através do estudo destes software, filtrar algumas tarefas básicas necessárias à qualquer produto que pretenda acrescentar facilidades ao cotidiano da gerência de redes. Procurou-se também verificar a capacidade destas ferramentas de se adaptarem ao ambiente de dispositivos móveis, analisando-se pontos positivos e negativos no que tange a simplicidade e facilidade de uso.

## 2 HISTÓRICO

Quando, em 1876, Graham Bell fez com que a voz humana trafegasse por fios, talvez jamais imaginasse que sua invenção fosse parte de uma revolução cultural e social tão grande como a que hoje acontece com o que chamamos de telefone. Este aparelho vem se reinventando desde então, e que principalmente no século 21 acelera a passos largos, agregando um sem número de funções, ferramentas e utilidades muito mais complexas.

A partir do advento da internet e da telefonia móvel o universo digital confunde-se com o real, uma vez que, para acessar todo o conteúdo disponível na rede, o indivíduo não precisa mais estar ligado a um ponto fixo de conexão.

Atualmente os aparelhos celulares reúnem diversas tecnologias (câmera fotográfica, filmadora, viva-voz, agenda, gravador de voz, MP3 *player*, comunicação sem-fio). Oferecem, ainda, diversos tipos de serviços, como por exemplo, sistemas de posicionamento, envio e recebimento de mensagens de texto, voz, multimídia e e-mails, acesso à Internet, download de sons e imagens, acesso a contas bancárias, jogos, bate-papo, entre outros. Segundo (FIGUEIREDO; NAKAMURA, 2003) “Devemos estar atentos à explosão das tecnologias para computação móvel que deve acontecer nos próximos anos. Muitas oportunidades devem surgir, seja no desenvolvimento dessas tecnologias ou na utilização delas, tanto no setor comercial quanto no setor acadêmico”.

### 2.1 Gerações

A primeira geração de redes celulares utilizava sistemas analógicos e cada país implantava o seu sistema e, pela escassez de padronização, os sistemas eram, em geral, incompatíveis entre si. A segunda geração (2G), já digital, buscou a padronização, a nível continental, que permitisse a mobilidade entre várias redes. Atualmente, as redes

celulares estão em transição da segunda para a terceira geração. A terceira geração (3G) é caracterizada por uma comunicação celular de alcance mundial.

- Primeira geração: Baseada em protocolos de comunicação que permitiam somente a comunicação por voz através de celulares analógicos.
- Segunda geração: Muito utilizado no país, os celulares (2G) permitem, além da comunicação por voz, transmissão de dados digitais a baixa velocidade (14,4 kbps), acesso a internet (WAP) e envio de mensagens SMS.
- Geração 2,5: É representada pelas novas tecnologias de transmissão por pacotes, como GPRS e CDMA 1x e, principalmente, pelos serviços diferenciados, como o acesso permanente à web, possíveis pelo aumento das velocidades (50kbps) e tarifação por volume de dados. A cada ligação de voz há um canal de dados à disposição do usuário, basta acessá-lo.
- Terceira geração ou 3G: Com taxas de transmissão ainda maiores (ordem de centenas de kbps) podemos destacar a tecnologia EDGE (GSM), FOMA (NTT DOCOMO), W-CDMA3 (*Wideband - Code Division Multiple Access*). Este novo serviço oferece qualidade na transmissão de voz, com mínima interferência e ruído, além de, suporte a diversos conteúdos de multimídia, como transmissão de vídeos, imagens, música e jogos, além de grande capacidade de comunicação de dados.

## 3 APLICAÇÕES PARA DISPOSITIVOS MÓVEIS

### 3.1 Internet Móvel

Com a explosão do uso da internet, a maioria das empresas migrou suas aplicações de uma arquitetura cliente/servidor para aplicações web, possibilitando o acesso de todos via um navegador (*browser*) como o Internet Explorer. Com o objetivo de dar mobilidade a estas aplicações buscou-se desenvolver a internet móvel ou internet sem-fio. No caso do celular, a navegação é feita através de um *microbrowser* de forma que ele se adapte às condições limitadas do celular. Assim como o navegador normal, o *microbrowser* utiliza um endereço (URL) para contatar um servidor específico. A resposta processada no servidor é traduzida pelo navegador que a apresenta ao usuário.

Dentre os *microbrowsers*, existem o OpenWave, Pocket Internet Explorer, Avantgo, Blazer, Wapaka e Opera, sendo o WAP o protocolo mais utilizado em celulares. A vantagem desse tipo de arquitetura está no fato de as aplicações serem construídas em cima de aplicações existentes na internet sem precisarem ser implementadas nos aparelhos dos clientes e, quando necessária, sua atualização pode ser feita a qualquer momento, afetando apenas o servidor. O grande problema dessa arquitetura está na qualidade da internet móvel, pois ela necessita de uma rede efetiva e de qualidade durante todo o tempo e se a taxa de transmissão for baixa, a aplicação será afetada. Mesmo assim, pode ser utilizada em e-commerce e entretenimento, bem como em outros aplicativos que levem em conta a conectividade oferecida pela rede existente.

A maioria dos navegadores presentes nos celulares modernos usa XHTML em vez de WML. O mesmo ocorre com os sites desenhados para celulares. Na época em que foi desenvolvido, o WAP pretendia ser a versão do WWW para tecnologias móveis. Entretanto, o que aconteceu foi um distanciamento da web HTML / HTTP, o que deixou os usuários apenas com o conteúdo nativo WAP e Web-to-WAP. Enumeram-se várias e efetivas vantagens de se usar XHTML nos novos documentos web produzidos ou

migrar os documentos atuais escritos em HTML. Dentre elas destaca-se em primeiro plano a compatibilidade da linguagem XHTML com as futuras aplicações de usuários, garantindo desde já que as criações XHTML conservar-se-ão estáveis por longos anos. Soma-se a isso o fato de ser uma linguagem estável, oficialmente especificada pelo W3C, tendo sido projetada e revisada pelos seus membros e é um padrão da web.

## 3.2 Smart-Clients

Esse tipo de arquitetura é utilizado na maioria dos aplicativos desenvolvidos para celular, pois eles podem ser executados a qualquer momento, mesmo quando o acesso à internet não está disponível. Eles dispõem de uma forma permanente de armazenamento de dados e não apenas cachê. Dessa forma, quando existe conexão, o acesso móvel sincroniza as informações entre cliente e servidor permitindo que as aplicações tenham uma interface sofisticada e um melhor desempenho. Exemplo: Aplicação de coleta de dados.

Esta arquitetura apresenta maior complexidade no desenvolvimento das aplicações, passando a exigir dispositivos móveis que disponham de um sistema operacional para suporte a essas aplicações.

## 3.3 Sistemas Operacionais

Existem vários sistemas operacionais para celulares, alguns dos mais utilizados são o Palm OS, o Symbian OS e o Windows Mobile. Além destes sistemas operacionais, devem ser citadas também as plataformas BREW, Java e Linux.

### 3.3.1 Palm OS

Pioneiro nos PDAs, o Palm OS (PALM, 2008) não teve no mercado corporativo o mesmo sucesso obtido no mercado de consumidores em geral, chegou a ser o sistema operacional mais usado em *palmtops*. Limitações iniciais das versões anteriores à 5.0 dificultaram a implantação de aplicações complexas abrindo campo para a adoção de



dispositivos com outros sistemas operacionais como o Windows Mobile da Microsoft.

### 3.3.2 Symbian

O Symbian OS (SYMBIAN, 2008) é um sistema operacional desenvolvido por um consórcio de empresas de telefonia para dispositivos móveis. Seu sistema é modular, permitindo que cada empresa crie sua própria interface. Abrindo, assim, a possibilidade de desenvolver sistemas desde textos em telas monocromáticas até aperfeiçoados sistemas gráficos encontrados nos *SmartPhones* da Nokia e Ericsson, sendo a interface Nokia Series 60 a mais famosa delas.

### 3.3.3 Windows Mobile

Sistema operacional desenvolvido pela Microsoft para dispositivos móveis como os Pocket PCs. O Windows Mobile (2008) permite uma adaptação mais rápida de aplicações desenvolvidas originalmente para o Windows, facilidade para conexões sem fio e a última versão do Windows Mobile possui suporte nativo às tecnologias Wi-Fi e Bluetooth sem a necessidade de baixar drivers adicionais para acessá-las. Possui integração com o Microsoft Exchange Server 2003 e 2007, permitindo sincronizar e-mails e agendas pessoais. Permite o desenvolvimento de aplicativos e tem ao seu favor todo o *know-how* para desenvolvimento e fácil utilização da Microsoft, além de contar com um dos melhores ambientes de produção de *software*, o Visual Studio.

### 3.3.4 BREW (Binary Runtime Environment for Wireless)

Ambiente binário de tempo de execução para aplicativos sem fio, BREW (2008), é um sistema operacional, desenvolvido pela Qualcomm, semelhante ao Symbian. Possui: mensagens (SMS, EMS, email etc.); navegação pela internet (WAP e HTML); suporte à tecnologias de comunicação celular: CDMA e suas extensões (CDMA200 1xe WCDMA); desenvolvimento de aplicativos em C++ e Java (J2ME, SDK).

O BREW foi projetado para permitir que o desenvolvedor escolha a linguagem para a codificação dos programas. Oferece suporte nativo à linguagem C/C++, mas aceita a integração de navegadores e aplicações em outras linguagens, inclusive Java e XML.

O BREW inclui plataforma de aplicativos e ferramentas de portagem para fabricantes de dispositivos, a saber: o Brew SDK, Software Development Kit e o Brew Distribution System, BDS. Este sistema de distribuição é controlado e gerenciado pelas operadoras, o que lhes permite obter aplicativos dos desenvolvedores, comercializá-los

e coordenar os processos de faturamento e pagamento.

Este sistema apresenta alguns atrativos que o diferencia de sistemas como o J2ME, como a capacidade de processamento *off-line*, recursos multimídia e grande riqueza gráfica. Ele não define limites às aplicações que por sua vez podem acessar áreas de memória de qualquer parte do celular, compartilhar dados com outros aplicativos e alterar áreas de código. O fato do C/C++ permitir acesso via apontadores à memória, o que o Java não permite, ajuda o BREW a ser mais flexível, porém exige que o programador saiba o que está fazendo. Por esse motivo, o custo de desenvolvimento no BREW é mais alto, pois toda aplicação tem que ser testada e certificada pela Qualcomm para garantir que não se introduza na rede aplicações mal intencionadas ou simplesmente mal feitas e com isso se coloque em risco toda a credibilidade da plataforma BREW. Exemplos de aparelhos que usam a tecnologia: Motorola T720, Audiovox e Toshiba

## Análise de Ferramentas Para Gerência de Redes

Este capítulo busca examinar o estado da arte em ferramentas de gerência de redes através da análise de alguns dos mais relevantes software de apoio existentes hoje no mercado, esmiuçando suas características e diferentes visões, além de buscar uma plausível justificativa para o desenvolvimento deste trabalho.

Foram selecionados para a análise apenas *software* de código aberto sendo relegados os sistemas proprietários. Um dos principais critérios de análise é a adequação destas ferramentas ao uso via dispositivos móveis, assim como a possibilidade de rodar em diferentes sistemas operacionais, a existência de interfaces amigáveis e a simplicidade de utilização.

### 3.4 Nagios

Nagios (2008) é um sistema de monitoramento de rede que gerencia máquinas e serviços alertando quando alguma coisa não vai bem conforme critérios pré-definidos. Algumas das suas características:

- a) Monitoramento de serviços:
  - SMTP
  - POP3
  - HTTP
  - NNTP
  - PING
  - Outros.
- b) Monitoramento de recurso de computadores (processamento, uso de disco, etc.)
- c) Permite que os usuários desenvolvam seus próprios monitoramentos.
- d) Habilidade de definir hierarquia de rede, além de detecção e distinção entre máquinas que estão *down* e aquelas que não estão acessíveis.
- e) Notificação quando algum problema ocorre ou é resolvido (via email, *pager*, ou algum método definido pelo usuário).
- f) Rotação automática de arquivos de log.
- g) Opcional interface web para verificação de status, notificações e históricos de problemas, arquivos de log, etc.

### 3.4.1 Requisitos do Sistema

Nagios requer uma máquina rodando Linux e um compilador C, além de ter o TCP/IP configurado. Caso o uso das CGIs inclusas no Nagios seja necessário, é preciso ter um servidor web (preferencialmente Apache) instalado e também a versão 1.6.3 (ou maior) da gd library.

Esta ferramenta nos permite monitorar, tanto máquinas que rodam o sistema operacional Windows como máquinas que rodam Linux. Entretanto, para o monitoramento de uma máquina Windows é necessária a instalação de um agente, que irá atuar como um *proxy* entre um *plugin* Nagios e o real serviço na máquina Windows. Não é possível o monitoramento via Nagios sem este agente.

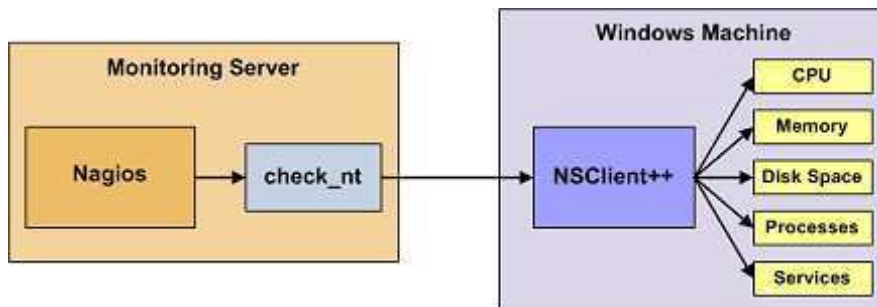


Figura 4.1: Agente Nagios para Máquinas Windows

Por outro lado, temos vários caminhos diferentes para monitorar servidores Linux/Unix. Um deles utiliza conexões seguras SSH, o que embora seguro pode resultar num esforço grande para o servidor de monitoramento, principalmente se este está monitorando centenas ou milhares de serviços, causado pelo *overhead* das conexões SSH.

Outro método comum é o uso do *addon* NRPE, que permite a execução de *plugins* em máquinas Linux remotas. É útil para monitorar recursos locais como o uso de memória, CPU ou disco.

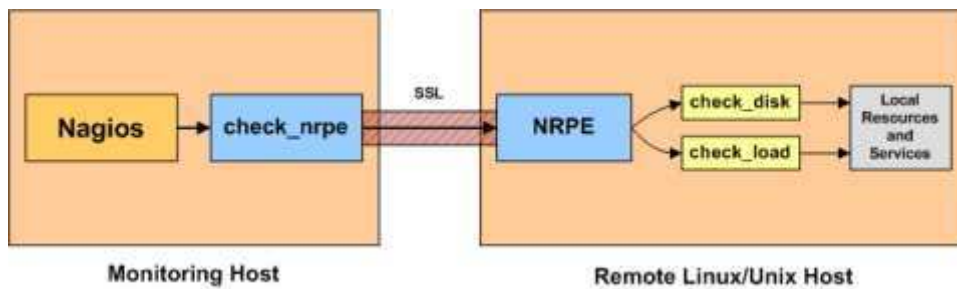


Figura 4.2: Agente Nagios para Máquinas Linux

### 3.4.2 Plugins

Diferente de outras ferramentas de monitoramento, Nagios não inclui nenhum mecanismo interno de verificação de status de máquinas ou serviços na rede, o sistema utiliza-se de programas externos chamados *plugins*.

*Plugins* são executáveis compilados ou scripts (Perl, shell, etc.) que podem ser rodados por linha de comando. Esses *plugins* serão executados sempre que o Nagios precisar fazer uma checagem no serviço ou máquina, retornando os resultados para o Nagios que irá processá-los e tomar as ações necessárias, rodando algum aplicativo, mandando notificações, etc.

### 3.4.3 CGIs

Existem várias CGIs distribuídas com Nagios, algumas necessitam autorização para serem usadas, essas normalmente requerem que o usuário autentique-se em um servidor web e esteja autorizado para ver as informações que estejam sendo requisitadas. A seguir são descritas algumas delas.

#### 3.4.3.1 WAP Interface CGI (*statuswml.cgi*)

Esta CGI serve como uma interface WAP para informações do status da rede como detalhes de serviços, checagem de problemas, etc. Esta CGI necessita uma série de autorizações e configurações para poder funcionar.

#### 3.4.3.2 Configuration CGI (*config.cgi*)

Permite visualizar objetos (dispositivos, grupos de dispositivos, contatos, períodos de tempo) previamente definidos em arquivos de configuração.

#### 3.4.3.3 *Command CGI (cmd.cgi)*

Permite enviar comandos ao processo do Nagios. Possui vários argumentos, mas deve ser utilizada com cuidado, pois seu comportamento varia nas diferentes versões do Nagios.

#### 3.4.3.4 *Event Log CGI (showlog.cgi)*

Esta CGI permite a visualização de arquivos de log do sistema.

Nagios é uma ferramenta poderosa e flexível, mas consome um bom tempo para ser configurado e para rodar exatamente como o usuário deseja, além de ser trabalhoso e oferecer um leque enorme de configurações, exige do usuário que conheça muito bem a sua rede e saiba exatamente o que deseja monitorar e como.

Existem vários e diferentes arquivos de configuração que precisam ser criados ou modificados antes de começar a monitorar qualquer coisa, exigindo bastante paciência e atenção do usuário.

### 3.5 **Zenoss**

Zenoss (ZENOSS, 2008) é um sistema que traz muitas informações de gerenciamento e monitoramento baseadas em um modelo padrão que contém a descrição detalhada de todos os dispositivos gerenciados e a relação existente entre eles ou entre outros grupos de dispositivos definidos pelo usuário. O acesso ao sistema é feito através de um navegador web.

O sistema foi escrito na linguagem de programação Python (PYTHON, 2008) e baseado no servidor de aplicações web Zope (ZOPE, 2008). Zenoss possui várias versões, algumas das quais são comerciais.

Para rodar o Zenoss, algumas dependências devem ser satisfeitas, a saber:

- a) MySQL;
- b) Python e python-dev;
- c) Twisted Network Library (python);

- d) rrdtool;
- e) NET-SNMP;
- d) ssh.

Devido ao alto nível de detalhamento das informações deste modelo existem várias maneiras de exibir estas informações. A forma básica é através de um processo denominado pelo Zenoss de “*auto-discovery*” que descobre os serviços, interfaces, etc. de um dispositivo. Com essas informações é construído um modelo do dispositivo no sistema. Este modelo também pode ser complementado acrescentando-se e manipulando-se dados através da web ou APIs externas ao Zenoss.

O monitoramento dos dispositivos feito pelo sistema consiste numa série de testes sobre a infra-estrutura para determinar se tudo está funcionando como deveria. Por exemplo: testes de ping, de processamento, e de serviços.

A ferramenta possui um sistema de gerenciamento de eventos que é a consolidação do *status* das informações de todas as partes do sistema Zenoss. Quando alguma parte do sistema detecta uma falha, uma série de eventos é gerada, muito similar à outras ferramentas de monitoramento. Além de unir todos estes eventos em um único repositório, eles são submetidos a um conjunto de regras que aumentam essas informações integrando-as ao modelo.

Seu sistema de gerenciamento de desempenho analisa as mudanças ocorridas em importantes recursos, como o espaço disponível em um disco, a utilização da CPU ou quanto tempo demora o download de uma determinada página web. Zenoss pode coletar informações usando SNMP, scripts (ZenCommands) ou XML-RPC.

Para preencher o seu banco de dados com as informações dos dispositivos o Zenoss pode utilizar do “*auto-discovery*”, um de cada vez através da interface web, ou ambos os métodos via um arquivo XML.

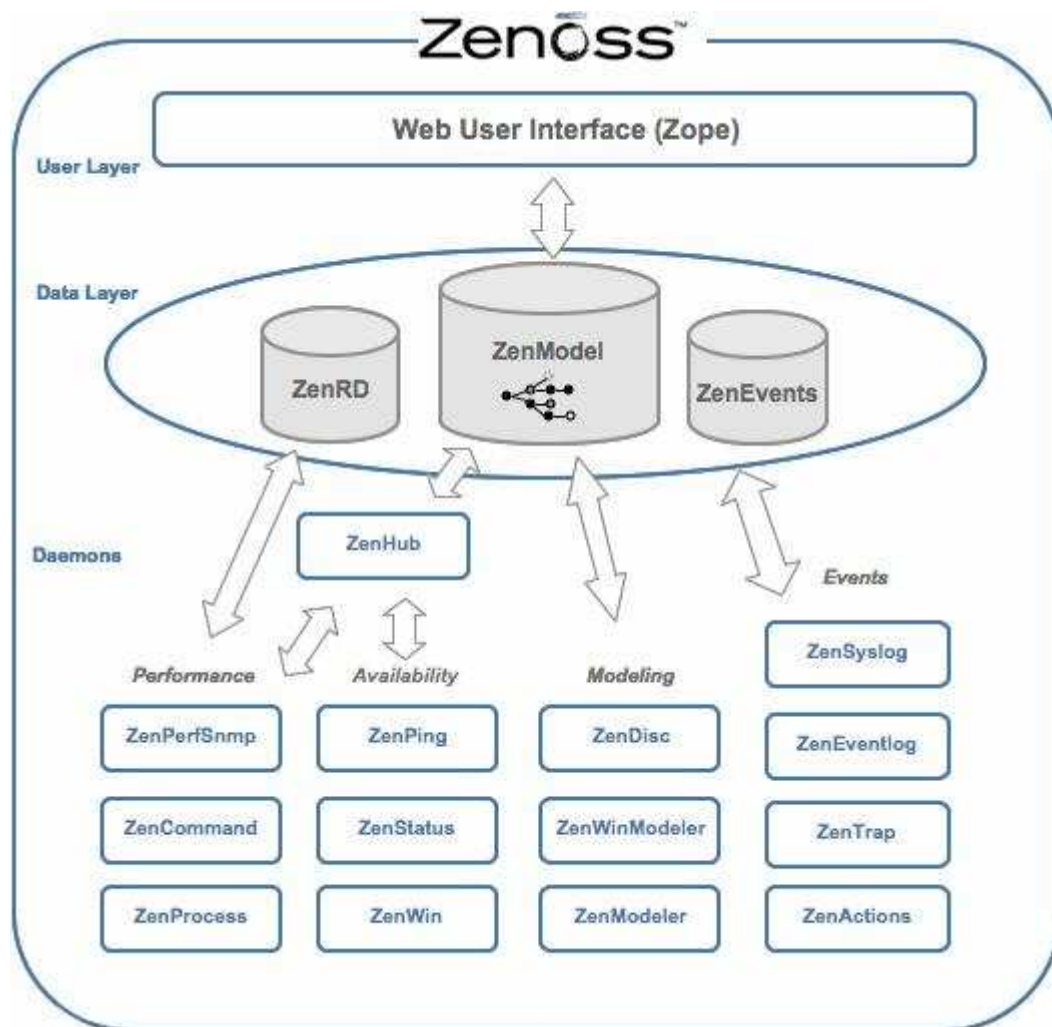


Figura 4.3: Arquitetura detalhada do zenoss

Os dispositivos podem ser modelados por SNMP, SSH ou Telnet. Cada técnica produz diferentes tipos de informações. Geralmente, através de SNMP obtêm-se um conjunto mais completo de informações, SSH/Telnet comumente são usados para aumentar o modelo quando agentes SNMP não conseguem obter uma específica parte de uma informação crítica.

### 3.5.1 Zenoss Dashboard

Uma vez instalado o sistema, através de uma URL têm-se a interface do usuário, uma janela inicial com os dispositivos, eventos e atividades disponibilizados pelo sistema. Existe um campo de busca onde o usuário pode entrar com o nome ou nome parcial de uma máquina ou ainda com o endereço IP. A cada 60 segundos a página é atualizada através de AJAX. Embora bastante detalhada, esta interface não está adaptada, nem parece ser a mais adequada para ser acessada via dispositivos móveis como um aparelho celular.



The screenshot displays the Zenoss CORE web interface. At the top, the logo 'Zenoss CORE' is visible, along with a search bar for 'Device/IP Search' and navigation links for 'docs', 'Preferences', 'Logout', and 'Help'. The server time is shown as 'Zenoss server time: 11:29:18'. A sidebar on the left contains a 'Main Views' menu with options like 'Dashboard', 'Event Console', 'Device List', and 'Network Map', as well as 'Classes', 'Events', 'Browse By', and 'Management' sections.

The main content area features a 'Locations' map of the United States with several green and red markers. Below the map is an 'Object Watch List' table:

Object	Events
/Devices/Discovered	1

To the right of the map are three panels:

- Zenoss Issues:** A table with columns 'Device', 'Daemon', and 'Seconds'. It currently shows 'No records found.'
- Root Organizers:** A table listing various system objects and their event counts:

Object	Events
/Devices/Network	1
/Devices/Server	1
/Devices/Ping	1
/Devices/Discovered	1
/Devices/Printer	0
/Devices/Power	0
/Devices/Web	0
/Devices/Storage	0

- Device Issues:** A table listing specific devices and their event counts:

Device	Events
ubuntu804t-64.zenoss.loc	1
comcast.zenoss.loc	1
fc8t.zenoss.loc	2
ubuntu606t.zenoss.loc	1
talk.google.com	1

Figura 4.4: Interface Zenoss

### 3.5.2 Google Maps

A ferramenta utiliza de um sistema externo, para mapear localizações, disponibilizado pelo Google chamado Google Maps. Atribuindo a dispositivos “endereço” considerados válidos pelo Google Maps estes aparecerão em um mapa como um ponto. Pode-se atribuir diferentes cores para representar diferentes dispositivos ou eventos.

### 3.5.3 Monitorando remotamente dispositivos via SSH

Para monitorar um dispositivo remotamente é preciso instalar um *plugin* do Zenoss em cada dispositivo remoto que se deseja monitorar. Os *plugins* são pacotes que podem ser encontrados em dois formatos: RPM ou “*source distribution*”. RPM é recomendado para sistemas baseados em Red Hat, que permitem aos administradores atualizar estes pacotes facilmente quando novas versões são lançadas. “*Source distribution*” é utilizado, geralmente, por desenvolvedores e tem como benefício a não necessidade do usuário ter privilégios de *root* para instalar o *plugin*.

### 3.5.4 Monitorando Dispositivos Windows

Zenoss requer alguma preparação para monitorar dispositivos Windows. É preciso ter um agente SNMP habilitado, assim como DCOM precisa estar habilitado para receber conexões WMI. Opcionalmente, é possível utilizar o SNMP para obter estatísticas de CPU, memória e dados sobre leitura e escrita de disco.

O agente SNMP obtém as informações do Windows via WMI no servidor onde ele foi instalado e converte, estados e dados operacionais em SNMO OIDs. Zenoss então processa essas informações e gera eventos e alertas baseado nelas. A não utilização deste agente pode limitar a quantidade de informações que o Zenoss pode processar.

## 3.6 Cacti

Cacti (CACTI, 2008) é uma ferramenta de monitoramento, escrita em PHP, que pode ser dividida em três diferentes tarefas, captar, guardar e apresentar informações, podendo monitorar diversos tipos de dispositivos, como servidores, equipamentos de rede, etc.

Para rodar essa ferramenta em uma máquina Linux, é necessário uma série de pacotes tais como: httpd, php, php-mysql, php-snmp, entre outros. Além disso é preciso configurar o PHP, o servidor apache e o MySQL, que já devem estar instalados. Cacti também roda sobre Windows.

Para receber dados de pontos remotos, Cacti utiliza-se do Simple Network Management Protocol SNMP. Para armazenar dados, o sistema utiliza rrdtool. RRD é o acrônimo para Round Robin Database, que é capaz de armazenar dados de forma muito compacta sem gastar muito tempo, além de permitir a criação de gráficos a partir das informações armazenadas. Esta característica pode ser bastante útil quando combinada

com um servidor web, permitindo o acesso aos gráficos em qualquer navegador em qualquer plataforma.

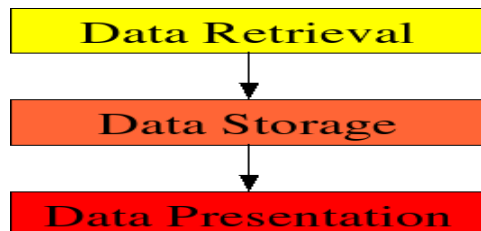


Figura 4.5: Princípios de operação do Cacti

Para não ter que definir todos os parâmetros do rddtool e do *layout* de seus gráficos o Cacti utiliza-se de *templates* para isso, perdendo um pouco de flexibilidade na tentativa de facilitar as coisas para o usuário. Existem *templates* para gráficos, dados e dispositivos.

Cacti permite a extensão de suas capacidades através de *scripts*, podendo ser utilizadas linguagens como PHP, Perl, shell/batch entre outras. Assim, proporcionando o uso de outros protocolos além do SNMP, como o ICMP para medir ping, telnet, ssh, ldap, etc...

A utilização de dispositivos móveis não é contemplada pelo Cacti, este pode ser acessado por um celular via navegador, mas sua interface não está preparada para tal, o que torna a sua utilização bem pouco produtiva devido à difícil visualização. O sistema permite o envio de mensagens de aviso para um celular, mas para isso é preciso desenvolver um script específico para este fim.

The screenshot displays the Cacti web interface for creating a new device. The interface is organized into a sidebar on the left and a main configuration area on the right. The sidebar contains a navigation menu with options such as 'Create', 'New Graphs', 'Management', 'Graph Management', 'Data Sources', 'Devices', 'Collector Methods', 'Data Queries', 'Data Input Methods', 'Templates', 'Graph Templates', 'Host Templates', 'Data Templates', 'Import/Export', 'Import Templates', 'Export Templates', 'Configuration', 'Settings', 'Utilities', 'System Utilities', 'User Management', and 'Logout User'. The main configuration area is titled 'Device [new]' and contains the following fields and options:

- Description:** Give this host a meaningful description. (Text input: New Host)
- Hostname:** Fully qualified hostname or IP address for this device. (Text input: hostname.domain.com)
- Host Template:** Choose what type of host, host template this is. The host template will govern what kinds of data should be gathered from this type of host. (Dropdown menu: ucd/net SNMP Host)
- Notes:** Enter notes for this host. (Text area)
- Disable Host:** Check this box to disable all checks for this host. (Checkbox:  Disable Host)
- Availability/Reachability Options:**
- Downed Device Detection:** The method Cacti will use to determine if a host is available for polling. NOTE: It is recommended that, at a minimum, SNMP always be selected. (Dropdown menu: Ping)
- Ping Method:** The type of ping packet to wait. NOTE: ICMP on Linux/UNIX requires net privileges. (Dropdown menu: UDP Ping)
- Ping Port:** TCP or UDP port to attempt connection. (Text input: 23)
- Ping Timeout Value:** The timeout value to use for host ICMP and UDP ping. This host SNMP timeout value applies for SNMP pings. (Text input: 400)
- Ping Retry Count:** The number of times Cacti will attempt to ping a host before failing. (Text input: 1)
- SNMP Options:**
- SNMP Version:** Choose the SNMP version for this device. (Dropdown menu: Version 3)
- SNMP Username (v3):** SNMP v3 username for this device. (Text input: snmpuser)
- SNMP Password (v3):** SNMP v3 password for this device. (Text input: \*\*\*\*\*)
- SNMP Auth Protocol (v3):** Choose the SNMPv3 Authorization Protocol. (Dropdown menu: MD5 (default))
- SNMP Privacy Passphrase (v3):** Choose the SNMPv3 Privacy Passphrase. (Text input: myprivpass)
- SNMP Privacy Protocol (v3):** Choose the SNMPv3 Privacy Protocol. (Dropdown menu: DES (default))
- SNMP Context:** Enter the SNMP Context to use for this device. (Text input: )
- SNMP Port:** Enter the UDP port number to use for SNMP (default is 161). (Text input: 161)
- SNMP Timeout:** The maximum number of milliseconds Cacti will wait for an SNMP response (does not work with php-snmp support). (Text input: 500)
- Maximum OIDs Per Get Request:** Specify the number of OIDs that can be obtained in a single SNMP Get request. NOTE: This feature only works when using Spine. (Text input: 10)

At the bottom right of the configuration area, there are two buttons: 'cancel' and 'create'.

Figura 4.6: Interface do Cacti

Cacti também suporta a utilização de *scripts* por linha de comando permitindo através destes comandos a criação de gráficos, a modificação de permissões, a inclusão de novos dispositivos para monitoramento, etc. Além de um bom entendimento da linguagem destes *scripts* é exigido do usuário um bom estudo prévio das possibilidades, variáveis e parâmetros para que o sistema não incorra em erros.

### 3.7 Considerações Finais

Percebe-se nas ferramentas descritas pelo trabalho a exigência de um profundo estudo e a exigência de pessoas qualificadas para a implantação e utilização dos referidos sistemas. O grande número de arquivos, suas respectivas configurações e o enorme leque de possibilidades, nem sempre vem a contribuir para a eficácia de resultados. Embora todas elas possam ser acessadas via telefones celulares, nenhuma

apresenta uma interface voltada diretamente para este tipo de dispositivo móvel. Pode-se destacar algumas características:

Nagios – Possui uma interface web, embora seja opcional; roda em Linux e Windows, mas para este último são necessários *plugins*; possui uma interface WAP disponibilizada via CGI; carece um bom tempo de estudo e conhecimento para poder se usufruir dos diversos recursos.

Zenoss – Disponibiliza um grande número de informações com alto nível de detalhamento; roda em Windows apenas via agente SNMP; roda em qualquer navegador via servidor web, mas não possui uma interface própria para celulares.

Cacti – Roda em qualquer navegador via servidor web, mas não possui uma interface voltada para o uso de dispositivos móveis; permite extensões através de *scripts*.

O próximo capítulo irá apresentar a modelagem de um sistema focado em dispositivos móveis, fácil utilização e interface amigável para profissionais da área de redes de computadores, em complemento ou substituição a estes sistemas consagrados, nas tarefas de administração a partir de um celular.

## **4 ESPECIFICAÇÃO**

### **4.1 Análise de requisitos**

#### **4.1.1 Visão Geral**

O sistema de gerência e monitoramento de redes através de dispositivos móveis tem como objetivo principal ser um sistema de fácil utilização e interface amigável para profissionais da área de redes de computadores, buscando tornar mais simples alguns processos cotidianos desta área da tecnologia da informação.

Os usuários deste sistema são, principalmente, gerentes de redes ou responsáveis pela área de informática de empresas que utilizam uma rede de computadores e que devem estar atentos aos problemas apresentados por esta.

#### **4.1.2 Requisitos**

1. O sistema deve prover uma lista de comandos de funções que auxiliem o monitoramento de redes de computadores
2. O sistema deve prover, minimamente, os comandos ping e netstat.
3. O sistema deve prover uma forma de visualização dos comandos, que permita ao usuário escolher qual comando ele deseja executar.
4. O sistema deve prover interfaces adequadas para cada comando.
5. O sistema deve permitir a entrada de parâmetros para os comandos escolhidos.
6. O sistema deve apresentar mensagens de saída para cada comando executado.
7. O sistema deve prover uma forma do usuário interromper a execução do mesmo e retornar para o uso normal do dispositivo.

## 4.2 Casos de uso

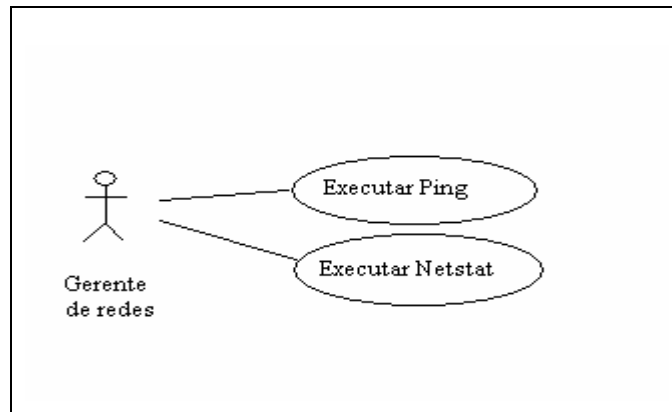


Figura 4.1: Executar comandos

## 4.3 Fluxo de operação

### 4.3.1 Fluxo principal

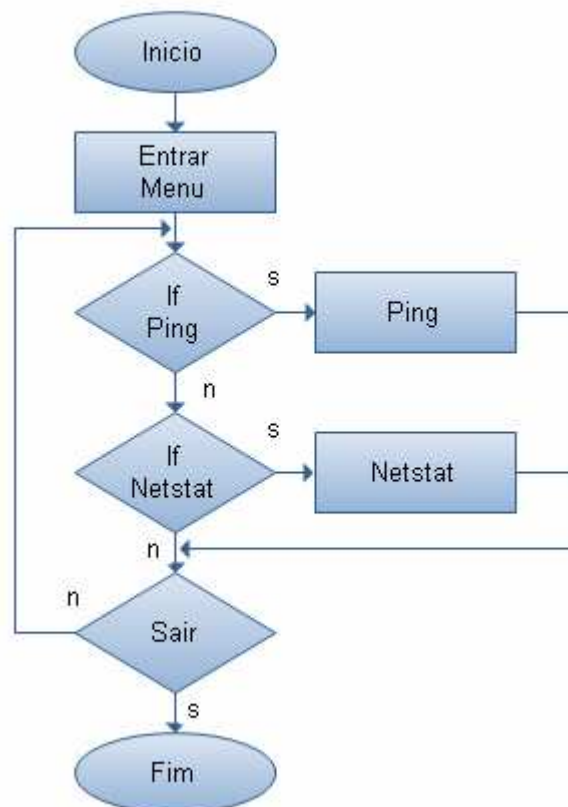


Figura 4.2: Fluxo principal

### 4.3.2 Fluxos dos comandos Netstat e Ping

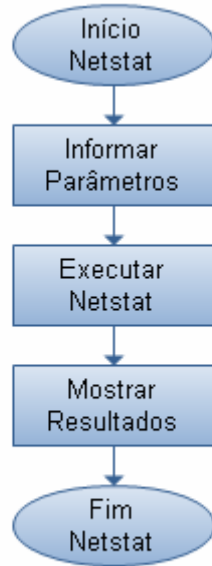


Figura 4.3: Fluxo Netstat



Figura 4.4: Fluxo Ping

## 4.4 Diagrama de Classes

### 4.4.1 Classes Netstat e Ping

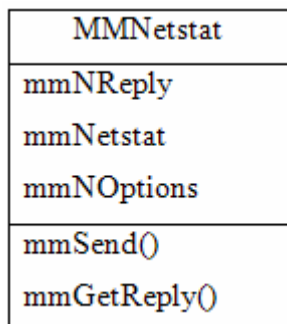


Figura 4.5: Classe Netstat

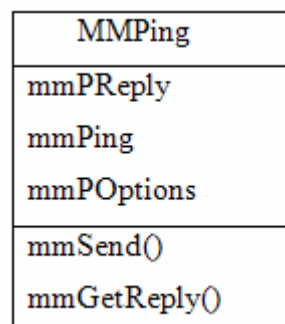


Figura 4.6: Classe Ping



No próximo capítulo será descrito o protótipo das classes do sistema e a interface da aplicação. São descritas as classes MMPing e MMNetstat, responsáveis, respectivamente, por implementar os comandos ping e netstat, ficando como sugestão para trabalhos futuros a ampliação desta biblioteca implementando-se novas classes que contemplem comandos como traceroute, whois, ipconfig, entre outros.

## 5 MODELAGEM

O Sistema foi desenvolvido utilizando a plataforma Pocket PC como interface ao usuário e escrito na linguagem C#, utilizando o conjunto de ferramentas integradas para desenvolvimento de software Visual Studio (Visual Studio) que possui uma IDE de fácil utilização, boa documentação e um grande número de exemplos.

O desenvolvimento é idêntico ao desenvolvimento desktop, uma vez que toda a suíte Visual Studio.net oferece as mesmas formas e sintaxes de desenvolvimento, por utilizar como plataforma de execução a .NET Framework e a .NET Compact Framework. Dentre as suas vantagens está a total compatibilidade com os aplicativos da Microsoft, como Internet Explorer, Word, Excel, Money, Outlook, sem a necessidade de aplicativos de terceiros. Outra vantagem, talvez a principal em relação aos seus concorrentes, é sua capacidade multimídia.

Objetivou-se ser o mais simples e portátil possível no desenvolvimento deste projeto, permitindo que qualquer programador entenda o código e possa transcrevê-lo pra outras plataformas, sem esquecer o usuário final desta aplicação. “Por vezes, desenvolvedores são atraídos pela complexidade, focando as mais complicadas áreas de um sistema, porque esta área representa um maior desafio e, portanto interessa mais a eles. Usuários, entretanto, não estão interessados em complexidade, pelo contrário, fogem dela, dando valor a simplicidade e facilidade de utilização” (McCONNEL, 1996, p.591).

A aplicação utiliza um serviço rodando num servidor que além de facilitar a execução de alguns comandos, no quesito permissões, facilita alterações no sistema e futuras atualizações. Serviços, assim como componentes, são considerados blocos de construção independentes, os quais coletivamente representam um ambiente de aplicação. No entanto, diferentemente de componentes tradicionais, serviços têm algumas características únicas que lhes permitem participar como parte de uma arquitetura orientada a serviços. Uma destas características é a completa autonomia em relação a outros serviços. Isto significa que cada serviço é responsável por seu próprio domínio, o que tipicamente significa limitar seu alcance para uma função de negócio específica (ou um grupo de funções relacionadas).

Devido à independência que esses serviços desfrutam dentro desta estrutura, a lógica de programação que encapsulam não tem necessidade de obedecer a nenhuma outra plataforma ou conjunto de tecnologias. Desta forma é utilizado um ambiente mais simples, independente do dispositivo que está sendo monitorado.

O estudo e a instalação de *web services* em servidores não fazem parte do escopo deste trabalho, sendo que muitas informações sobre este tópico podem ser encontradas em (SOTERO, 2008; PERKOVICH, 2008).

## 5.1 Classes

MMPing: Esta classe permite determinar se um computador remoto está acessível pela rede. Da mesma forma que o comando de rede ping, a classe envia um ICMP echo request para o host remoto e espera por um ICMP echo reply.

```
public class MMPing
{
    public MMPing()
    private PingReply mmPReply;
    private Ping mmPing;
    private PingOptions mmPOptions;

    public int mmSend(string host);
    public string mmGetReply();
}
```

MMNetstat: Esta classe mostra informações sobre o estado da rede, acessando estruturas de dados relacionadas na rede de dentro do *kernel*, fornecendo relatórios da tabela de roteamento, conexões TCP, escuta TCP e UDP, e protocolo de gerenciamento de memória.

```
public class MMNetstat
{
    public MMNetstat()
    private PingReply mmMReply;
    private Ping mmNetstat;
    private PingOptions mmNOptions;

    public int mmSend(string host);
    public string mmGetReply();
}
```

## 5.2 Modelo da Interface

A interface desta aplicação é baseada nas recomendações da W3C – *Mobile Web Best Practices*, cujo principal objetivo é “melhorar a experiência de usuários web quando acessando a rede de tais dispositivos” (RABIN; McCATHIENEVILE, 2008).



Figura 5.1: Tela Inicial



Figura 5.2: Menu de Operações

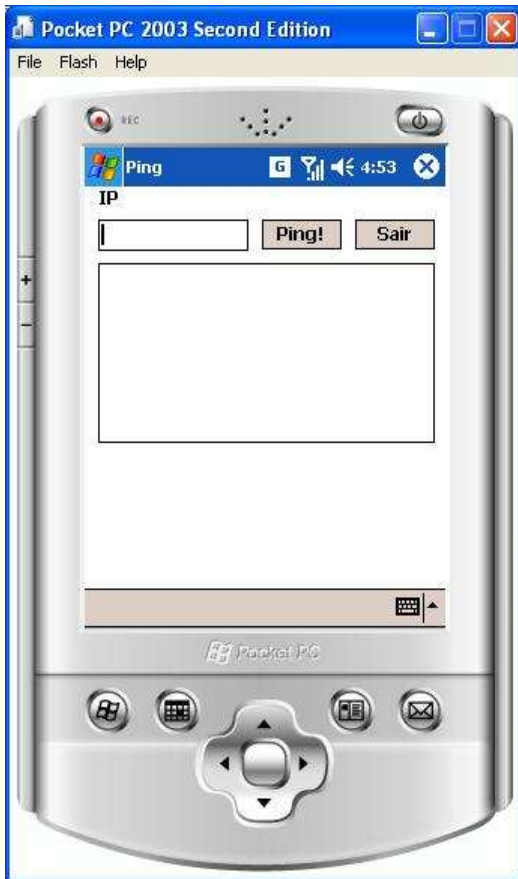


Figura 5.3: Interface da Classe MMPing

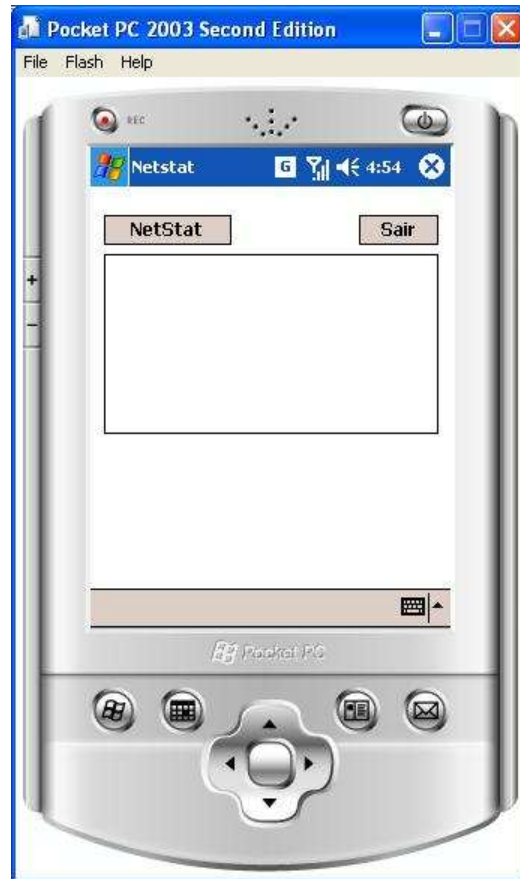


Figura 5.4: Interface da Classe MMNetstat

## 6 CONCLUSÃO

Após o estudo do advento da utilização de celulares nas mais variadas áreas, suas principais características, e a utilização destes dispositivos como ferramenta de trabalho dos profissionais de informática, mais precisamente, os envolvidos com o gerenciamento de redes, percebe-se, via análise de alguns software auxiliares destes, a necessidade do desenvolvimento de novas ferramentas específicas para utilização em dispositivos móveis. Visto que as ferramentas de análise e gerenciamento existentes ainda não se encaixam nos conceitos exigidos pelos celulares, como simplicidade, mobilidade e praticidade.

Através da análise de algumas ferramentas relevantes, de código aberto: Cacti, Nagios e Zenoss, percebeu-se, neste trabalho, que estas ferramentas exigem um apurado estudo e um conhecimento não superficial de seu funcionamento para um eficaz aproveitamento das suas potencialidades. Notou-se também a falta de uma interface que contemple as características peculiares de dispositivos móveis, como por exemplo, o tamanho exíguo das telas.

Procurou-se através deste trabalho, fornecer ao profissional de informática uma ferramenta simples e de rápido procedimento que proporcione, estando onde estiver, a tomada das primeiras decisões, minimizando assim o impacto e a duração de estados de falha, para depois tomar medidas secundárias através de sua equipe ou em um ambiente mais apropriado. Sugere-se, para trabalhos futuros, a expansão do software, agregando uma lista maior de comandos além dos aqui apresentados, tais como traceroute, whois, ipconfig, entre outros.

Muito ainda há para ser estudado e desenvolvido nesta área, assim como os dispositivos móveis cada vez mais tornam-se multifuncionais, além de expandirem sua memória e capacidade de processamento adquirindo contornos de microcomputadores. Dentro disso, maior a necessidade de software que contemplem suas específicas necessidades, e que permitam aos usuários uma utilização mais intuitiva, simples, eficiente e adaptada à necessidade destes.

## REFERÊNCIAS

BREW: Binary Runtime Environment for Wireless. Disponível em: <<http://brew.qualcomm.com>>. Acesso em: out. 2008.

THE CACTI Manual. Disponível em: <<http://www.cacti.net>>. Acesso em: ago. 2008.

FIGUEIREDO, C.; NAKAMURA, E. Computação Móvel: novas oportunidades e novos desafios. **T&C Amazônia**, ano 1, n.2, jun. 2003. Disponível em: <[https://portal.fucapi.br/tec/imagens/revistas/ed002\\_000\\_completa.pdf](https://portal.fucapi.br/tec/imagens/revistas/ed002_000_completa.pdf)>. Acesso em: set. 2008.

LIMA, F. O. **A sociedade digital**: o impacto da tecnologia na sociedade, na cultura, na educação e nas organizações. Rio de Janeiro: Qualitymark, 2000.

McCONNEL, S. **Rapid Developent**. Redmond, Washington: Microsoft Press, 1996.

MICROSOFT CORPORATION. **Visual Studio**. Disponível em: <<http://msdn.microsoft.com/en-us/vstudio/default.aspx>>. Acesso em: set. 2008.

NAGIOS Version 3.x Documentation. Disponível em: <<http://www.nagios.org/docs>>. Acesso em: ago. 2008.

PALM Corporate. Disponível em: <<http://www.palm.com/us/company>>. Acesso em: out.2008.

PERKOVICH, D. **Build Your First Web Service with Visual Studio .NET.**

JupiterMedia Corporation. Disponível em:

<<http://portals.devx.com/SummitDays/Article/6470/2213>>. Acesso em: set. 2008.

PYTHON Programming Language. Disponível em: <<http://www.python.org>>. Acesso em: out. 2008.

RABIN, J. ; McCATHIENEVILE, C. **Mobile Web Best Practices.** Disponível em:

<<http://www.w3.org/TR/mobile-bp/>>. Acesso em: set. 2008.

SOTERO, S. **Criando um Web Service com Visual Studio.NET.** Disponível em:

<[http://imasters.uol.com.br/artigo/1123/visual\\_studio/criando\\_um\\_web\\_service\\_com\\_o\\_visual\\_studionet/](http://imasters.uol.com.br/artigo/1123/visual_studio/criando_um_web_service_com_o_visual_studionet/)>. Acesso em: set. 2008.

SYMBIAN. **The open mobile operating system.** Disponível em:

<<http://www.symbian.com>>. Acesso em: out. 2008.

WINDOWS Mobile Home. Disponível em:

<<http://www.microsoft.com/windowsmobile>>. Acesso em: out. 2008.

ZENOSS Administration Guide. Disponível em:

<<http://www.zenoss.com/community/docs>>. Acesso em: ago. 2008.

ZOPE. **The Web Site for the Zope Community.** Disponível em:

<<http://www.zope.org>>. Acesso em: out. 2008.