FACULDADES INTEGRADAS DE TAQUARA
CURSO DE SISTEMAS DE INFORMAÇÃO

SISTEMA DE MONITORAMENTO DA UTILIZAÇÃO DE SOFTWARES EM ESTAÇÕES DE TRABALHO PARA REDES DE COMPUTADORES

NAIRA KAIESKI

Taquara
2009

NAIRA KAIESKI

SISTEMA DE MONITORAMENTO DA UTILIZAÇÃO DE SOFTWARES EM ESTAÇÕES DE TRABALHO PARA REDES DE COMPUTADORES

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação das Faculdades Integradas de Taquara, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação, sob orientação do Prof. M.Eng. Alexandre Timm Vieira.

Taquara

2009

Dedico este trabalho a Deus, em quem confio e a quem agradeço por tudo que tem me proporcionado.
Dedico à minha mãe que muito me ensinou sobre a vida.
Em especial ao meu companheiro e esposo Jacques, a quem amo muito e que tem estado comigo, me apoiando.

AGRADECIMENTOS

Gostaria de agradecer a Deus pela oportunidade de estudar e de crescer como profissional e também como pessoa.

Com todo carinho ao meu esposo Jacques pela dedicação, carinho e paciência, principalmente neste período mais conturbado onde tem a pressão do TCC.

Aos professores do Curso de Sistemas de Informação da FACCAT, que mais que professores, tornaram-se amigos, pelos quais guardo enorme admiração.

Ao pessoal do Núcleo de Sistemas Administrativos e Núcleo de Atendimento aos Usuários da FACCAT por estarem sempre dispostos a ajudar.

Ao pessoal dos laboratórios de informática da FACCAT onde testei o SPY007.

Aos meus colegas de trabalho do Núcleo de Internet e Redes da FACCAT pela compreensão e ajuda, em especial à Márcia. Me sinto feliz por integrar esta equipe.

Aos colegas formandos do curso de Sistemas de Informação, somos vencedores.

Em especial ao colega Leandro Sorgetz, membro da comissão de formatura do curso de Sistemas de Informação, acredito que fizemos um bom trabalho para a nossa formatura.

A todos que torceram por mim e de alguma forma me ajudaram.

RESUMO

O uso de computadores e o acesso a Internet nas empresas é praticamente universal, nestas organizações grande parte dos funcionários faz uso da conexão disponível. Tendo em vista que a empresa é responsável pelos atos praticados pelos seus colaboradores dentro da instituição, torna-se indispensável ao administrador monitorar as atividades que são realizadas nos computadores da organização. Este trabalho tem por objetivo desenvolver um *software* de monitoramento das atividades dos usuários nas estações que compõem a rede de computadores da empresa. O aplicativo desenvolvido monitora os programas que realmente estão sendo utilizados, ou seja, que estão em primeiro plano no computador. O aplicativo não disponibiliza as telas que os usuários estão acessando, mas sim informações referentes à aplicação que está sendo utilizada e o *site* que está sendo acessado no momento. Os dados coletados são armazenados em banco de dados e posteriormente são exibidos em forma de gráficos, onde o administrador pode selecionar o período que deseja consultar bem como a estação e/ou o usuário. O *software* desenvolvido se intitula SPY007 e foi instalado em dois laboratórios das Faculdades Integradas de Taquara. Primeiramente o monitoramento foi realizado sem que os usuários soubessem e posteriormente foram informados do monitoramento. O presente trabalho também apresenta informações referentes ao comportamento dos usuários frente à navegação na Internet e o uso de *softwares* quando não estavam cientes do monitoramento e também quando foram notificados quanto à realização do mesmo.

Palavras-chave: Monitoramento, Gerência de rede, SNMP.

LISTA DE FIGURAS

LISTA DE SIGLAS

ASN.1 - *Abstract Syntax Notation One*

CETIC.br - Centro de Estudos sobre as Tecnologias de Informação e Comunicação

CGI.br - Comitê Gestor da Internet no Brasil

HTTP - *Hypertext Transfer Protocol*

HTTPS - *Hypertext Transfer Protocol Secure*

IP - *Internet Protocol*

ISO - *International Organization for Standardization*

MIB - *Management Information Base*

MVC - *Model View Controller*

OID - *Object Identifier*

OSI - *Open Systems Interconnection*

PHP - PHP: *Hypertext Preprocessor*

RFC - *Request for Comments*

RNP - Rede Nacional de Pesquisa

SMI - *Structure of Mangement Information*

SNMP - *Simple Network Management Protocol*

TCP - *Transmission Control Protocol*

TIC - Tecnologia da Informação e Comunicação

UML - *Unified Modeling Language*

WEB - *World Wide Web*

XHTML - *Extensible Hypertext Markup Language*

SUMÁRIO

1 INTRODUÇÃO

De acordo com os dados divulgados na pesquisa anual TIC Empresas 2008, que mede o uso das tecnologias de comunicação e informação, o percentual de empresas brasileiras que utilizam computador chega a 94% onde a grande maioria, 97%, possuem acesso a Internet. A pesquisa citada foi divulgada pelo CETIC.BR (2009).

Na Figura 1 são apresentados os percentuais de uso de computadores segundo o porte das empresas. É possível perceber que a utilização de computadores aumenta quando são analisadas empresas de maior porte.

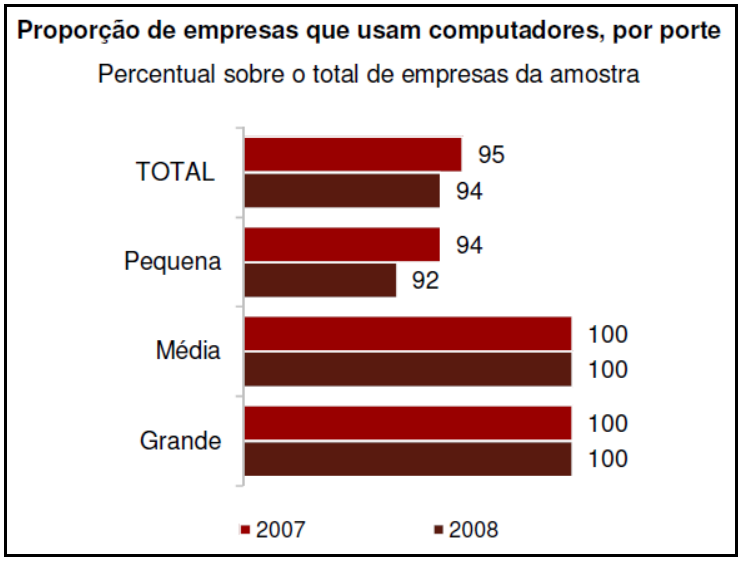

Figura 1 - Proporção de empresas que usam computadores de acordo com o porte
Fonte: CETIC.BR (2009, p. 3)

A pesquisa divulgada pelo CETIC.BR (2009) também apresenta informações sobre o uso de redes de computadores, onde as redes sem fio podem ser encontradas em 35% das empresas e a infra-estrutura de rede com fio está presente em 83% das organizações que possuem computador.

Nas empresas com acesso a Internet, a proporção média de funcionários que usam computadores conectados a WEB é de 43%. As principais atividades realizadas pela empresas na Internet são o envio e recebimento de correio eletrônico (99%), busca de informações sobre produtos e serviços (94%) e busca de informações e atividades de pesquisa (82%) (CETIC.BR, 2009).

Na Figura 2 é possível visualizar as principais atividades realizadas pelas empresas na Internet.

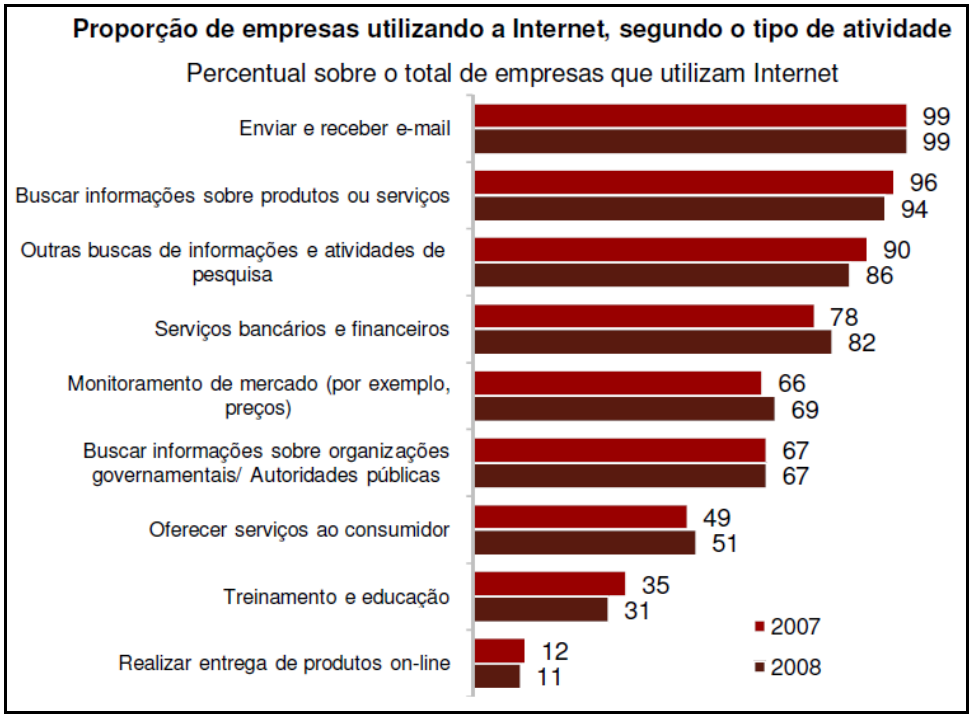

Figura 2 - Proporção de empresas utilizando a Internet por tipo de atividade
Fonte: CETIC.BR (2009, p. 10)

As grandes organizações apresentam uma preocupação significativa com a segurança no uso da Internet. Em média, 33% das empresas brasileiras possuem políticas de segurança e políticas de uso aceitável das tecnologias de informação e comunicação. Em 22% existem programas de treinamento em segurança da informação para os funcionários. Um dos grandes problemas enfrentados pelas empresas no ano de 2008 foram os vírus, sendo 55% das organizações a relatar este tipo de problema. O segundo maior foram os cavalos de tróia, 48%. Já os acessos externos indevidos foram relatados por 10% das empresas e os acessos internos não autorizados por 9% (CETIC.BR, 2009).

A revolução que é proporcionada pelos avanços tecnológicos em termos de comunicação e entretenimento, pode afetar diretamente as relações de trabalho entre colaboradores e organizações. O uso inadequado de uma tecnologia por parte de um funcionário é responsabilidade da empresa que o contratou, ficando esta sujeita a ações judiciais (BATISTELA, 2009).

> Pirataria de *software*, utilização indevida do e-mail da empresa, uso inadequado da Internet, confidencialidade de informações coorporativas, responsabilidade na utilização das ferramentas de informática, são apenas exemplos dos riscos aos quais estas empresas estão expostas e a confirmação de que a PREVENÇÃO é efetivamente necessária (BATISTELA, 2009, p. 1).

Os administradores de tecnologia se veem obrigados a adotar práticas cada vez mais elaboradas para impedir que os usuários façam uso indevido dos recursos computacionais da empresa, seja este intencional ou mesmo por falta de conhecimentos. No entanto, sempre surgem métodos para burlar qualquer restrição de acesso. Uma maneira de inibir os usuários é definir uma rígida política de acesso e uso dos recursos de informática, e também monitorar as atividades realizadas nas estações da rede, de tal forma que o gerente de tecnologia da informação possa agir a fim de proteger os interesses da organização.

O uso de uma ferramenta de gerência de rede com o objetivo de monitorar as ações dos usuários nas estações de trabalho, juntamente com uma política de segurança amplamente divulgada e clara, são medidas efetivas no controle dos recursos computacionais e podem auxiliar a eximir a organização de uma ação de responsabilidade por algum ato ilegal praticado por um funcionário.

O presente trabalho visa ao desenvolvimento de um *software open source*[1] de monitoramento dos aplicativos utilizados pelos usuários nos computadores que compõem a rede de uma empresa. No mercado existem algumas ferramentas proprietárias que entre suas funcionalidades apresentam um módulo para monitorar as atividades dos usuários, porém esses aplicativos exigem um investimento considerável em licenças de uso. O sistema desenvolvido utiliza tecnologias chamadas o*pen source*, reduzindo assim os custos de desenvolvimento e implantação do *software* em qualquer empresa.

O sistema é composto por um agente, responsável pela coleta dos dados, que deve ser instalado em cada uma das estações da rede que se deseja monitorar. Um gerenciador que irá processar os dados coletados e armazenar os mesmos em banco de dados. As informações geradas pelo sistema são disponibilizadas na forma de gráficos e relatórios acessíveis na WEB.

---

[1] *Open source* é o termo utilizado para os aplicativos que possuem seu código fonte disponível de forma gratuita para que outros programadores possam acessá-lo, alterá-lo ou simplesmente utilizá-lo.

1.1 Organização do trabalho

Este trabalho está dividido em oito capítulos, o segundo capítulo apresenta uma contextualização sobre gerência de rede. No terceiro capítulo são expostos aplicativos disponíveis no mercado, que possuem funções semelhantes ao SPY007. O quarto capítulo expõe a metodologia empregada nas diversas etapas da análise, projeto e desenvolvimento do sistema. No quinto capítulo são citadas as tecnologias empregadas no desenvolvimento do aplicativo. O sexto capítulo explana o funcionamento do SPY007. O sétimo capítulo apresenta os dados referentes ao estudo aplicado da implantação do sistema SPY007 em dois laboratórios das Faculdades Integradas de Taquara. E finalmente as conclusões do trabalho.

2 GERÊNCIA DE REDES

> Como a Internet pública e as Intranets[2] privadas cresceram e se transformaram de pequenas redes em grandes infra-estruturas globais, a necessidade de gerenciar mais sistematicamente a enorme quantidade de componentes de *hardware* e *software* dentro dessas redes também se tornou mais importante. (KUROSE e ROOS, 2006, p. 572).

Conforme descrito pela RNP (2009) a área de gerência de rede foi impulsionada pela necessidade de monitoração e controle do universo de dispositivos que compõem a rede de comunicação. A expansão das redes e da Internet alavancou o desenvolvimento de mecanismos de gerência, visto que a comunicação em rede é fundamental para o mundo atual onde tudo acontece tem tempo real e *online*[3] na Internet. Tendo em vista a necessidade de comunicação, a tolerância a falhas na rede é cada vez menor frente às perdas financeiras que podem estar associadas a uma indisponibilidade da rede.

A gerência está associada ao controle das atividades e ao monitoramento do uso de recursos da rede. As tarefas básicas de gerência de redes são obter informações da rede, e tratar estas informações de maneira que seja possível diagnosticar e encaminhar as soluções para os problemas. Para que isso seja possível, é necessário que funções de gerência sejam embutidas nos diversos componentes de uma rede, possibilitando descobrir, prever e reagir aos problemas.

Segundo Sztajnberg (2009), as informações que circulam em uma rede de computadores devem ser transportadas de modo confiável e rápido. Para que isso aconteça, é importante que os dados sejam monitorados de maneira que os problemas que porventura possam existir, sejam resolvidos na medida do possível. Uma rede sem mecanismos de gerência pode apresentar problemas como congestionamento do tráfego, recursos mal utilizados, sistemas sobrecarregados, problemas com segurança e outros.

Para resolver os problemas associados à gerência de rede, a ISO propôs três modelos de gerenciamento, a saber:

a) modelo organizacional: que estabelece a hierarquia entre sistemas de gerência, onde o ambiente a ser gerenciado é dividido em vários domínios.

---

[2] Intranet é uma rede de computadores privada que utiliza os mesmos protocolos da Internet.

[3] *Online* no contexto da Internet significa estar conectado e disponível para acesso imediato.

b) modelo informacional: define os objetos da gerência, as relações e as operações sobre esses objetos. Neste modelo, uma MIB é utilizada para armazenar as informações sobre os objetos gerenciados.
c) modelo funcional: descreve as funcionalidades de gerência.

O gerenciamento no modelo OSI da ISO baseia-se na teoria de orientação a objetos. O sistema representa os recursos gerenciados através de entidades lógicas chamadas de objetos gerenciados. Uma aplicação desenvolvida utilizando a teoria de objetos faz uso de processos distribuídos, que são os gerentes (que gerenciam) e os agentes (que realizam ações).

2.1 Monitoramento de redes e sistemas

De acordo com Teixeira Júnior *et al.* (1999), o monitoramento de sistemas é essencial para o planejamento das necessidades nos ambientes computacionais. No projeto de um sistema de monitoramento alguns aspectos devem ser avaliados cuidadosamente, como por exemplo, os dados que devem ser monitorados, o mecanismo de coleta destes dados e como as informações obtidas podem ser utilizadas. Devem ser definidos os dados mais relevantes a serem coletados e estratégias para obter as informações dos recursos gerenciados. Os dados resultantes do monitoramento devem permitir análise e diagnóstico de problemas nas várias áreas funcionais de gerenciamento.

2.2 Gerenciamento baseado no modelo funcional OSI da ISO

Este modelo de gerenciamento é útil para situar os cenários apresentados em um quadro mais estruturado, onde são definidas cinco áreas de gerenciamento de rede: configuração, desempenho, falhas, contabilidade e segurança.

O gerenciamento de configuração, de acordo com Kurose e Roos (2006), permite que um administrador de rede saiba quais dispositivos fazem parte da rede e quais são as suas configurações de *hardware* e *software*.

O gerenciamento de desempenho, segundo Leite (2004), permite avaliar o comportamento dos objetos e a eficiência das atividades de comunicação. Esta gerência se

divide em duas categorias: a monitoração e o controle. A monitoração verifica as atividades na rede, enquanto que o controle permite fazer ajustes a fim de melhorar o desempenho da rede.

No gerenciamento de falhas, o objetivo é registrar, detectar e reagir às condições de falha na rede. A divisão entre gerenciamento de falha e gerenciamento de desempenho é de difícil definição. O gerenciamento de falhas pode ser considerado como o tratamento imediato de falhas transitórias da rede, enquanto o gerenciamento de desempenho aborda o longo prazo em relação ao desempenho da rede em face das demandas variáveis de tráfego, e falhas ocasionais na rede.

A gerência de contabilidade, de acordo com Teixeira Júnior *et al.* (1999) está relacionada à cobrança pelo uso dos serviços de rede. O administrador deve poder monitorar o uso dos recursos da rede por um usuário ou mesmo um grupo de usuários. Este acompanhamento deve ser executado por várias razões, já que um usuário pode estar abusando de seus privilégios de acesso e atrapalhando o uso da rede por outros usuários, alguém pode estar fazendo um uso ineficiente da rede, ou mesmo uso indevido dos recursos disponíveis. Com a utilização de ferramentas de monitoramento de rede e dos vários sistemas que compõem a rede, o administrador estará em uma posição mais confortável para planejar o crescimento da rede visto que terá informações que o permita conhecer o nível de atividades dos usuários em detalhes.

Segundo Leite (2004), a gerência de segurança tem por meta manter os dados seguros e para isso controlar o acesso aos recursos da rede de acordo com alguma política de acesso definida.

2.3 Arquitetura de um sistema de gerenciamento de rede

O gerenciamento de rede possui uma terminologia específica, onde merecem destaque três componentes principais: a entidade gerenciadora, os dispositivos gerenciados e o protocolo de gerenciamento de rede.

A entidade gerenciadora é uma aplicação que controla a coleta, o processamento e a análise ou apresentação das informações de gerenciamento de rede. É nela que são iniciadas ações para controlar o comportamento da rede. Através da entidade gerenciadora o administrador pode interagir com os dispositivos da rede e, se necessário, alterar a

configuração remotamente. A entidade gerenciadora se comunica com dispositivos gerenciados espalhados pela rede emitindo comandos e obtendo respostas.

Um dispositivo gerenciado é um equipamento de rede qualquer, incluindo seu *software* (computador, impressora, switch, roteador), que faz parte de uma rede gerenciada. Nestes dispositivos pode haver diversos objetos gerenciados. Estes objetos são peças de *hardware*, propriamente ditas, que estão dentro dos dispositivos gerenciados (por exemplo, uma *interface* de rede). Estes objetos gerenciados possuem informações associadas a eles armazenadas dentro de uma base de informações de gerenciamento. Estas informações estão disponíveis para a entidade gerenciadora que, em muitos casos, pode efetuar ajustes nestes dados.

Em cada dispositivo gerenciado reside um agente de gerenciamento de rede. Este agente é um processo que se comunica com a entidade gerenciadora e que executa ações locais aos dispositivos gerenciados sob o comando e o controle da entidade gerenciadora.

O protocolo de gerenciamento de rede é executado entre a entidade gerenciadora e o agente de gerenciamento. O protocolo permite que a entidade gerenciadora consulte o estado dos dispositivos gerenciados e execute ações sobre ele mediante seus agentes. Os agentes podem usar o protocolo de gerenciamento de rede para informar à entidade gerenciadora à ocorrência de eventos excepcionais, como por exemplo, uma falha. O protocolo de gerenciamento de rede não gerencia a rede propriamente, ele fornece uma ferramenta com a qual o administrador pode gerenciar a rede.

Na Figura 3 é possível visualizar os principais componentes de uma arquitetura de gerenciamento de rede.

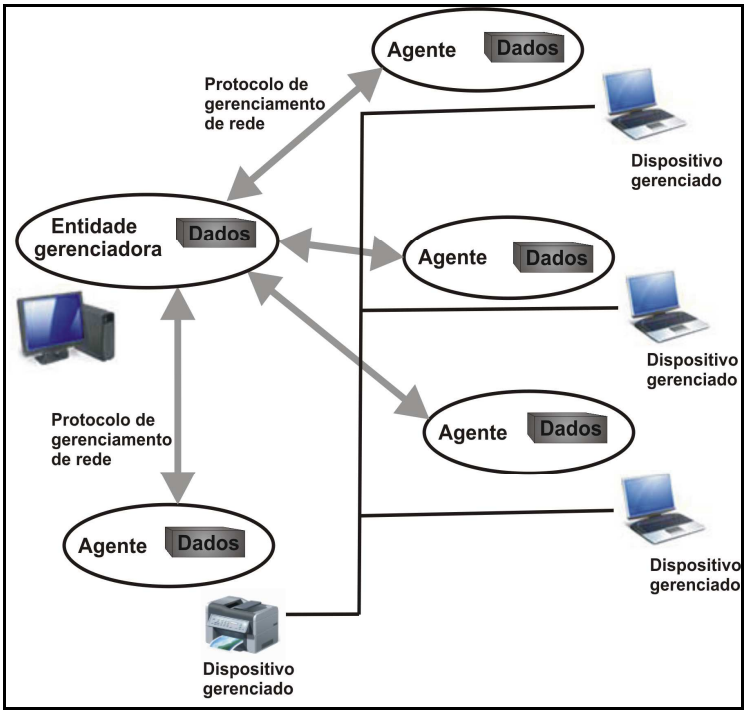

Figura 3 - Principais componentes de uma arquitetura de gerenciamento de rede
Fonte: Adaptado de Kurose e Roos (2006)

2.4 Protocolo SNMP

A origem do SNMP está relacionada com o crescimento do uso da Internet e a ausência de meios de gerência para essa grande rede. Nos primórdios do desenvolvimento da Internet não existia um protocolo de gerência específico para rede, então era utilizado o ICMP (*Internet Control Message Protocol*), que é um protocolo de comunicação entre roteadores, utilizado para identificar equipamentos inoperantes (KROLOW, 2000).

> [...] Como o SNMP foi projetado e oferecido rapidamente em uma época em que a necessidade de gerenciamento de rede começava a ficar premente, ele encontrou uma ampla aceitação. Hoje, esse protocolo é a estrutura de gerenciamento de rede mais amplamente usada e disseminada [...] (KUROSE e ROOS, 2006, p. 577).

Conforme citado anteriormente o SNMP é o protocolo de gerenciamento de rede mais difundido da atualidade, sendo que este se tornou amplamente utilizado no início da década de

1990. O SNMP auxilia o administrador de rede na tarefa de localizar e corrigir anomalias em uma interligação de rede TCP/IP. O administrador executa um aplicativo cliente SNMP (também chamado gerenciador SNMP) em sua estação local e utiliza o cliente para contatar um ou mais servidores SNMP (denominados agentes SNMP) executados em equipamentos remotos.

De acordo com o trabalho de Comer e Stevens (1999), o SNMP emprega um paradigma de busca e armazenamento (*fetch-store paradigm*) no qual cada servidor mantém um conjunto de variáveis conceituais que incluem estatísticas simples, como um contagem de pacotes recebidos, e variáveis complexas que correspondem a estruturas de dados do TCP/IP (tabelas de roteamento IP). As mensagens do SNMP especificam que o servidor deve buscar ou armazenar valores em variáveis e o servidor converte as solicitações para operações equivalentes sobre estrutura de dados locais. Como o protocolo não abrange outras operações, todo o controle precisa ser realizado por intermédio do paradigma de busca e armazenamento. Além do protocolo SNMP, um padrão à parte, referente a uma MIB, define o conjunto de variáveis que os servidores SNMP mantêm e a semântica de cada variável. Variáveis da MIB registram o estado de cada rede ou dispositivo conectado, estatísticas de tráfego, contagens de erros encontrados e os conteúdos correntes de estruturas de dados internas, utilização de CPU, processos ativos, entre outros. As especificações do protocolo SNMP podem ser consultadas na RFC1157.

2.4.1 ASN.1

A essência do modelo SNMP é o conjunto de objetos que é gerenciado pelos agentes e que é lido e gravado pela entidade gerenciadora. Para tornar possível a comunicação entre equipamentos produzidos por diferentes empresas é imprescindível que esses objetos sejam definidos de uma forma padronizada e neutra (no que se refere à empresa fabricante). Esta padronização também é necessária para que os objetos sejam codificados para transferência através da rede. A padronização dos objetos depende de uma linguagem de definição de objetos associada a regras de codificação (TANENBAUM, 1997).

“A ASN.1 é um padrão originado na ISO, usado em uma série de protocolos relacionados à Internet, particularmente na área de gerenciamento de rede.” (KUROSE e ROOS, 2006, p. 589).

O SNMP define a sintaxe e o significado das mensagens que a entidade gerenciadora e os dispositivos gerenciados trocam, e utiliza a ASN.1 para especificar tanto o formato de mensagens como nomes de variáveis da MIB. Assim as mensagens SNMP não possuem campos fixos e não podem ser definidas com estruturas fixas (COMER e STEVENS, 1999).

A ANS.1 é uma notação que permite definir tipos de dados simples e complexos e especificar valores que estes tipos podem assumir. Os valores que são transmitidos podem ser de diversos tipos e cada um recebe uma denominação que o distingue, de forma inequívoca de todos os demais tipos.

2.5 SMI

A SMI é a linguagem utilizada para definir a estrutura de dados de gerenciamento que residem em uma entidade gerenciada de rede (agente). Essa linguagem de definição é necessária para assegurar que a sintaxe e a semântica dos dados de gerenciamento de rede sejam definidas de forma rígida e clara e assim não apresentem ambiguidades. A SMI não define uma instância específica para os dados em uma entidade gerenciada de rede, mas a linguagem na qual a informação está especificada (KUROSE e ROSS, 2006).

2.6 Organização do agente SNMP

Um agente SNMP precisa aceitar uma solicitação que chega até ele, executar a operação especificada e retornar a resposta.

O fluxo de dados de uma mensagem SNMP através de um agente pode ser visualizado na Figura 4.

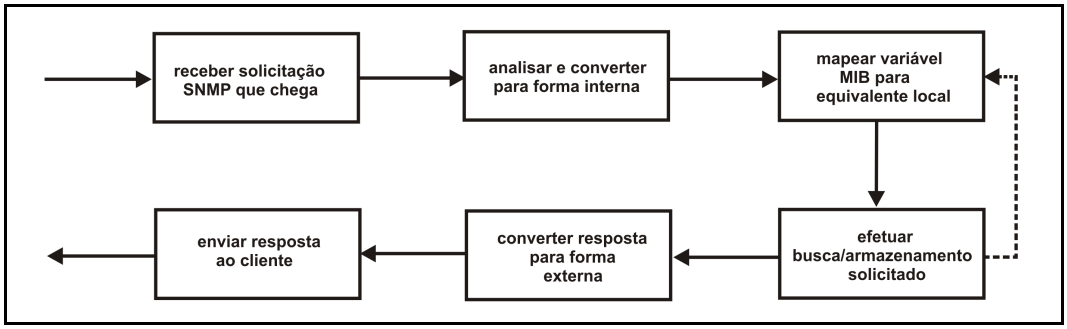

Figura 4 - A circulação de uma mensagem do SNMP por um servidor
Fonte: Adaptado de Comer e Stevens (1999)

O agente primeiramente analisa a mensagem e efetua a conversão para a forma interna. Depois, mapeia a especificação da variável da MIB para o item de dados local responsável pelo armazenamento das informações necessárias e efetua a operação de busca e armazenamento. Nas operações de busca, é feito o preenchimento da área de dados da mensagem SNMP com o valor que foi consultado no objeto respectivo. Caso a mensagem especifique diversas variáveis, o agente executa o mapeamento da variável MIB e efetua a busca e armazenamento correspondente a cada variável. Depois que todas as operações foram concluídas o agente converte a resposta da forma interna para a forma externa e retorna a mesma ao servidor.

2.7 Hierarquia do SNMP

Os dados do SNMP são organizados em uma hierarquia padronizada. Essa organização imposta permite que o espaço de dados permaneça universal e extensível. Grandes porções são separadas para expansão futura e inclusões específicas do fornecedor são localizadas para evitar conflitos. A hierarquia de atribuições de nome é composta por MIBs, arquivos-texto estruturados que descrevem os dados acessíveis via SNMP. As MIBS contêm descrições específicas das variáveis de dados, as quais são referidas por nomes conhecidos como OID (NEMETH, 2002).

Os tipos de dados básicos que uma variável SNMP pode conter são inteiro, conjunto de caracteres (*string*) e nulo. Eles podem ser combinados em sequências de tipos básicos, onde uma sequência pode ser instanciada repetidamente a fim de formar uma tabela.

A hierarquia de SNMP é muito semelhante a um sistema de arquivos. Entretanto, um ponto é utilizado como caractere separador e cada nó recebe um número em vez de um nome. Por convenção, os nós também recebem nomes de texto para facilitar a referência.

Conforme exposto na Figura 5, os primeiros níveis da hierarquia do SNMP são artefatos políticos e geralmente não contém dados úteis. Os dados mais relevantes podem ser encontrados sob o OID:

iso.org.dod.internet.mgmt → numericamente 1.3.6.1.2

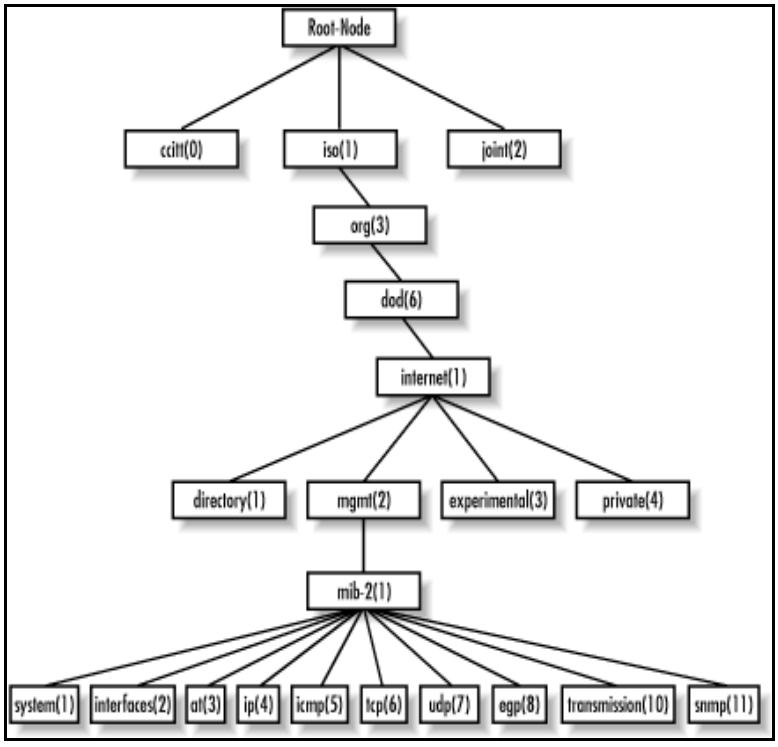

Figura 5 - Hierarquia SNMP
Fonte: O'Reilly (2009, p. 1)

2.8 MIB

> A MIB define variáveis conceituais que nem sempre correspondem diretamente às estruturas de dados que um *gateway* utiliza. O *software* do SNMP pode efetuar cálculos para simular algumas das variáveis conceituais, mas o sistema remoto não tomará conhecimento da ocorrência dos cálculos. (COMER e STEVENS, 1999, p. 407).

A MIB define as variáveis que um servidor SNMP deve manter. Precisamente, a MIB define um conjunto de variáveis conceituais que um servidor SNMP precisa ter condições de acessar. Em muitos casos, é possível usar variáveis convencionais para armazenar os itens que a MIB exige. Em outros casos, as estruturas de dados internas usadas por protocolos TCP/IP podem não corresponder exatamente às variáveis exigidas pela MIB. Em tais casos, o SNMP precisa ter condições de calcular os valores MIB necessários com base em estruturas de dados disponíveis.

As variáveis contidas na MIB podem ser divididas em duas classes: variáveis simples e tabelas. Variáveis simples abrangem tipos como inteiros com ou sem sinal, strings e estruturas conhecidas como registros. Tabelas correspondem a vetores unidimensionais. Uma só tabela pode conter diversas instâncias de uma variável. Embora o tamanho de variáveis simples seja conhecido a priori, o tamanho de uma tabela pode mudar com a passagem do tempo.

2.8.1 Nomes de variáveis da MIB

A MIB usa a ASN.1 para atribuir nomes a todas as variáveis. A ASN.1 define um espaço de nome hierárquico, de modo que o nome de cada variável reflita sua posição hierárquica. A organização da estrutura de variáveis MIB garante que embora atualmente muitas organizações atribuam nomes, os nomes resultantes sejam seguramente únicos e absolutos. A hierarquia que conduz a nomes MIB começa com o ISO, continua pela sub-árvore de organizações, departamento de defesa dos Estados Unidos (DoD), Internet, sub-árvore de gerenciamento (*management*) e finalmente a sub-árvore MIB-2. A cada parte da hierarquia foi atribuído um rótulo. Um nome é escrito como uma sequência de rótulos que denotam sub-árvores, com pontos separando os rótulos. O rótulo referente à hierarquia mais significativa aparece à esquerda. Desse modo, a variável MIB da sub-árvore IP que conta datagramas IP que chegam, *ipInReceives*, recebe o seguinte nome:

iso.org.dod.internet.mgmt.mib.ip.ipInReceives

Como pode ser visto no exemplo acima, os nomes MIB podem ser bastante longos e por conseqüência nomes de itens contidos em tabelas serão ainda mais longos do que nomes de variáveis simples, pois contêm rótulos adicionais que indicam o índice da entrada da tabela e o campo desejado dessa entrada (COMER e STEVENS, 1999).

2.8.2 Representação numérica de nomes

Ao enviar e receber mensagens, o SNMP não armazena nomes de variáveis como *strings* de texto, ele usa a forma numérica da ASN.1 para representar cada nome. Por ser mais compacta do que a representação em texto, a representação numérica economiza espaço em pacotes.

A forma numérica da ASN.1 atribui um só inteiro (geralmente pequeno) a cada rótulo contido em um nome e representa o nome como uma sequência de inteiros.

Para exemplificar a sequência de rótulos numéricos da variável *ipInReceives* é:

1.3.6.1.2.1.4.3

Quando a representação numérica de nomes de variáveis simples é utilizada em uma mensagem SNMP, esta apresenta um zero acrescentado ao final para especificar que o nome representa a única instância dessa variável MIB. Portanto, a forma exata se torna:

1.3.6.1.2.1.4.3.0

2.9 Operações do protocolo SNMP

Existem somente quatro operações básicas do SNMP: *get*, *get-next*, *set* e *trap*.

Get e *set* são operações de leitura e gravação de dados para um nó identificado por um OID específico.

Get-next é utilizado para percorrer passo a passo uma hierarquia MIB e ler o conteúdo de suas tabelas.

Uma *trap* (interrupção) é uma notificação assíncrona não-solicitada de uma entidade gerenciada (agente) para uma entidade gerenciadora (gerenciador) que informa a ocorrência de uma condição ou evento que pode ser importante para o funcionamento da rede. Várias *traps* padrão são definidas, incluindo notificações de falha ou recuperação de um *link* de rede. As *traps* comumente são utilizadas para controlar os valores de outras variáveis SNMP e disparar mensagens quando um intervalo determinado é excedido.

Na Figura 6 é possível visualizar as operações do SNMP.

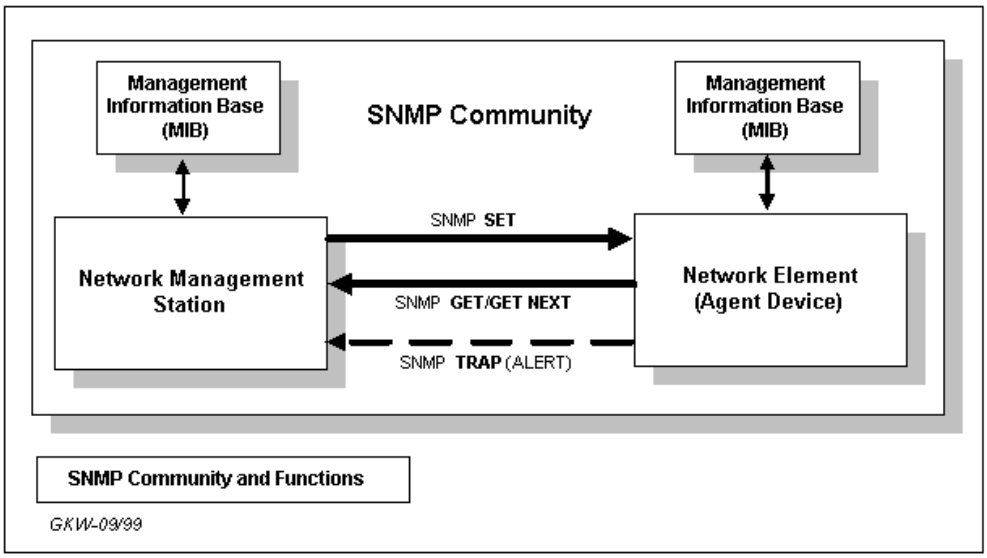

Figura 6 - Operações do protocolo SNMP
Fonte: WTCS (2009, p. 1)

Visto que as mensagens SNMP podem ser utilizadas para modificar configurações, é necessário um mecanismo de segurança para evitar alterações indevidas. A versão mais simples de segurança SNMP baseia-se no conceito de um "nome de comunidade" (*Community*) de SNMP, que efetivamente tem o mesmo papel de uma senha. Geralmente são definidas *communities* para acesso somente leitura e outra para acesso que permite gravação de dados (NEMETH, 2002).

O SNMP versão 3 (SNMPv3) inclui três importantes serviços: autenticação (*authentication*), privacidade (*privacy*) e controle de acesso (*access control*). O SNMPv3 é definido modularmente, onde cada entidade (*entity*) SNMP inclui um mecanismo (*engine*) SNMP. Uma *engine* implementa funções de envio e recebimento de mensagens, autenticação, encriptação e controle de acesso aos objetos geridos. Essas funções são consideradas serviços para um ou mais aplicativos que estão configurados no mecanismo SNMP para formar uma entidade SNMP. O SNMPv3 utiliza o modelo de segurança baseado no usuário (*User-Based Security Model*), que usa o conceito de mecanismo de autorização. Na transmissão de uma mensagem, uma das duas entidades é designada como autorizador (*authoritative engine*) da transmissão.

3 ESTADO DA ARTE

A necessidade de monitorar as atividades dos usuários a fim de estimar a sua produtividade junto à organização é uma realidade empresarial. Segundo Loureiro (2009), monitorar a Internet é o melhor remédio para prevenir e gerenciar crises nas empresas.

No mercado são comercializados alguns aplicativos que possuem um módulo específico para controle da produtividade dos colaboradores da organização. No projeto desenvolvido, o objetivo é a implementação de uma ferramenta de *software* livre, intuitiva, cujos relatórios possam ser acessados via Internet e que tenha por prioridade fornecer informações sobre as atividades dos usuários na rede sem a necessidade de outros módulos ou sistemas.

Ferramentas *open source* tradicionalmente empregadas no monitoramento de rede como MRTG, Nagios, Cacti e Zabbix não são apropriadas para o tipo de monitoramento que é realizado com o SPY007, visto que os dados a serem exibidos nos gráficos estão em constante mudança, o objeto do monitoramento são os programas que estão sendo utilizados e os sites que estão sendo acessados. Os aplicativos citados possuem em comum com o SPY007 a geração de gráficos, porém são comumente utilizados para monitorar dados como a utilização de processador, tráfego de rede em uma *interface*, utilização de memória, entre outros. Todos os monitoramentos citados possuem dados fixos, como por exemplo, quantidade de memória utilizada e quantidade de memória livre, tráfego de entrada e tráfego de saída de uma interface de rede. Os gráficos gerados pelos SPY007 contabilizam a utilização dos aplicativos mais utilizados e sites mais acessados sendo que os dados a serem exibidos podem variar de forma drástica.

Nas seções seguintes são apresentadas algumas ferramentas comerciais disponíveis no mercado. Porém ambas possibilitam ao administrador visualizar a tela do usuário com seus dados pessoais sem que ele saiba. Este tipo de acesso pode ser interpretado como violação de privacidade, sendo necessária a implantação de políticas de segurança e conscientização dos colaboradores quanto ao monitoramento que está sendo realizado. O SPY007, além de ser um *software* livre que não requer licenças de uso, não possibilita ao administrador visualizar a tela das estações monitoradas. O aplicativo desenvolvido apenas coleta dados que permitem identificar o aplicativo que está sendo utilizado e o *site* que está sendo acessado. Todos os dados referentes às coletas trafegam na rede através do protocolo SNMP enquanto que os

aplicativos citados abaixo implementam técnicas de *WebService* para transferir os dados das estações monitoradas ao servidor.

3.1 NetEye

O aplicativo NetEye é desenvolvido pela empresa que recebe o mesmo nome do aplicativo. A empresa NetEye surgiu no ano de 2000, a partir de uma ferramenta para gerenciamento de computadores em rede.

A ferramenta NetEye é uma solução comercial, para gerenciamento de computadores em rede, que realiza auditorias nos computadores, gerando estatísticas que permitem o aumento da produtividade, já que o gerente sabe exatamente o que o usuário está fazendo na estação. Possibilita a racionalização de investimentos e a otimização no uso dos equipamentos. O aplicativo permite que sejam enviados alertas sobre modificações na configuração de *hardware* ou *software* dos equipamentos que compõe a rede. O NetEye possui uma divisão em módulos, sendo eles: inventário, produtividade, monitoramento com controle remoto, desempenho e segurança.

No módulo de produtividade temos o monitoramento das atividades dos usuários, onde o administrador pode gerar relatórios a fim de estimar o tempo real de trabalho de um colaborador da organização com base nos aplicativos dos quais o mesmo fez uso (NETEYE, 2009).

Para utilizar o NetEye é necessário adquirir uma licença de uso para cada estação de trabalho que se deseja monitorar. A ferramenta é comercializada através de módulos, sendo possível selecionar apenas alguns módulos para aquisição. O NetEye funciona como cliente-servidor onde um computador da rede deve ser definido como servidor do aplicativo. Os dados obtidos pelo NetEye são armazenados em banco de dados para posterior consulta. Os relatórios do NetEye não podem ser acessados através da WEB. Apenas estações com sistema operacional Windows podem ser controladas/monitoradas com o NetEye. O NetEye pode ser utilizado sem custos para monitorar até 5 computadores.

Na Figura 7 é apresentada uma tela de monitoramento do NetEye.

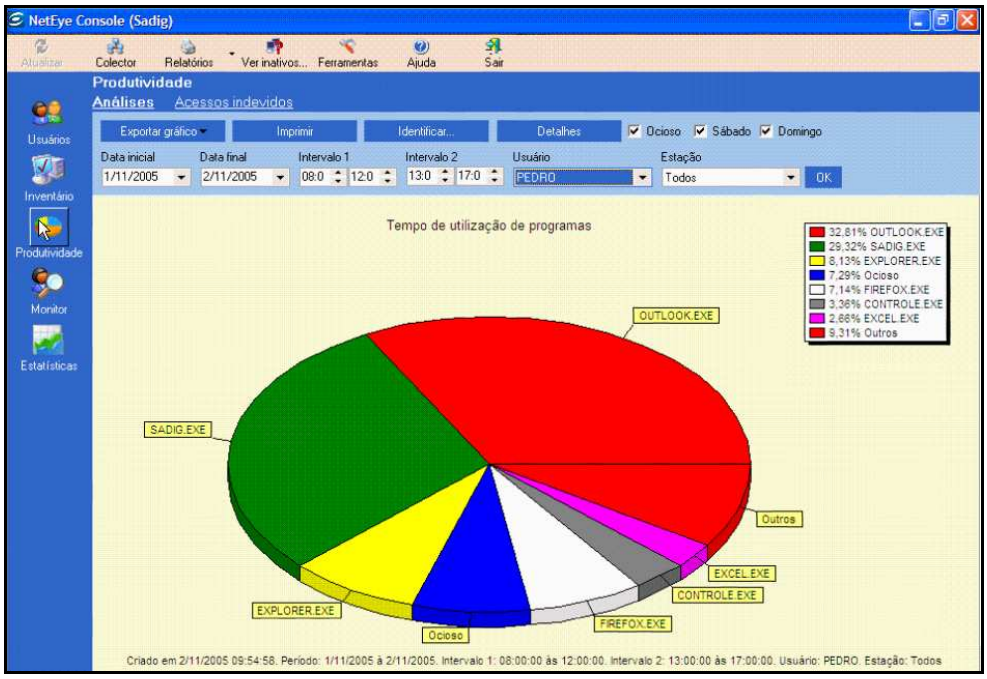

Figura 7 - Relatório de produtividade disponível no NetEye
Fonte: NetEye (2009, p. 1)

3.2 TraumaZer0

A suíte de aplicativos Trauma Zer0 (Tz0) é desenvolvida pela empresa iVirtua Solutions que foi fundada em 2001 na cidade no Montenegro no Rio Grande do Sul. A iVirtua é uma empresa brasileira que desenvolve soluções voltadas para o gerenciamento de serviços de tecnologia de informação.

O aplicativo Tz0 é uma ferramenta comercial dividida em diversos módulos que visam o gerenciamento centralizado do ciclo de vida da TI cobrindo a infra-estrutura, segurança e relatórios.

O Tz0 possui diversos módulos, são eles: controle de produtividade (Tz0 Productivity), inventário de *hardware* e *software* (Tz0 Asset Inventory), controle remoto (Tz0 Remote Control), distribuição de *software* (Tz0 Software Delivery and Deploy), controle e aplicação de diretrizes de segurança (Tz0 Network Security), monitor de performance (Tz0 Performance Monitor), análise e reconstrução de pacotes (Tz0 Sniffer Rescue), compactação

de anexos de mensagens de correio eletrônico (Tz0 Email Warp) e controle da informação (HelpDesk/ServiceDesk/Workflow - Tz0 Support Cycle).

O Tz0 é divido em um servidor e agentes que são instalados nos computadores que se deseja administrar/monitorar. Os agentes são multiplaforma, enquanto que o servidor deve ser instalado sobre o sistema operacional Windows.

O Trauma Zero pode ser adquirido separadamente por módulo, sendo que para cada computador que possui o agente instalado deve ser adquirida uma licença para cada um dos módulos desejados.

O módulo Tz0 Productivity and Software Metering é responsável pelo monitoramento das atividades dos usuários da rede bem como a utilização de cada um dos softwares disponíveis (IVIRTUA, 2009).

Na Figura 8 é apresentado um exemplo de relatório de produtividade gerado pelo Tz0 Productivity and Software Metering.

Figura 8 - Módulo Tz0 Productivity and Software Metering
Fonte: Ivirtua (2009, p. 1)

4 METODOLOGIA

No sistema proposto foi utilizada uma abordagem de desenvolvimento evolucionário, que segundo Sommerville (2003) tem como proposta desenvolver uma implementação inicial, expor o resultado ao comentário do usuário e fazer seu aprimoramento por meio de várias versões, até que o sistema adequado tenha sido desenvolvido. Ao invés de separar as atividades de especificação, desenvolvimento e validação, todo esse trabalho é realizado concomitantemente com um rápido retorno por meio dessas atividades. A vantagem de um processo de *software* com base na abordagem evolucionária é que a especificação pode ser desenvolvida gradativamente. À medida que os usuários desenvolvem uma maior compreensão das suas necessidades frente ao sistema que está sendo desenvolvido, todo o processo tende a ser melhorado, e refletir-se-á no *software* propriamente.

A Figura 9 demonstra as etapas do desenvolvimento baseado na abordagem evolucionária.

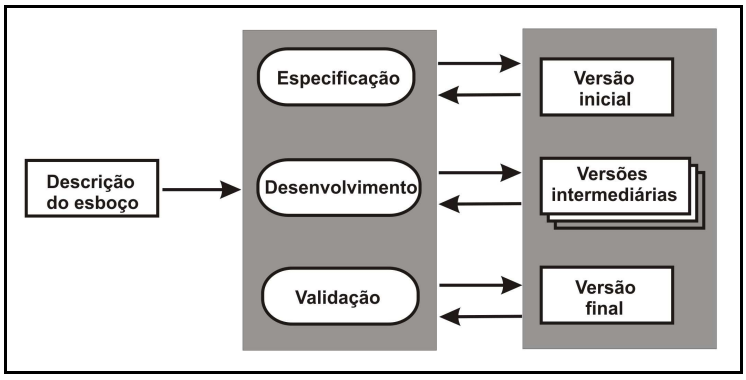

Figura 9 - Desenvolvimento evolucionário
Fonte: Adaptado de Sommerville (2003)

O modelo evolucionário adotado neste trabalho foi o modelo de desenvolvimento incremental, que de acordo com Pressman (2002), objetiva a elaboração de um produto funcional a cada incremento. Os primeiros incrementos são versões simplificadas do produto final, mas oferecem suficientes funcionalidades que permitem ao usuário realizar uma avaliação prévia do produto.

O modelo incremental aplica sequências lineares de uma forma racional à medida que o tempo passa, onde cada sequência produz um “incremento” factível do *software*. Quando este modelo é utilizado, o primeiro incremento é frequentemente chamado núcleo do produto, ou seja, os requisitos básicos são satisfeitos, mas muitas características suplementares deixam

de ser elaboradas. O núcleo do produto é usado pelo cliente ou passa por uma revisão detalhada. Um plano é desenvolvido para o próximo incremento, como resultado do uso ou avaliação. O plano visa à modificação do núcleo do produto para melhor satisfazer as necessidades do cliente e a elaboração de características e funcionalidades adicionais (PRESSMAN, 2002).

De acordo com os requisitos analisados, foram elaborados os seguintes incrementos para o desenvolvimento do SPY007:

a) desenvolver as funções básicas de comunicação entre o gerenciador do SPY007 e os agentes instalados nas estações. Desenvolver o ambiente gráfico de gerenciamento do SPY007 simplificado;
b) desenvolver as funções de coletas de dados completa nos agentes (nome da máquina, usuário, aplicativo e endereço da página WEB que está sendo acessada no momento);
c) desenvolver os gráficos detalhados referentes aos dados coletados e exibir estatísticas adicionais referentes à utilização dos aplicativos.

4.1 Análise de requisitos

A análise de requisitos é um processo de descobrimento, refinamento, modelagem e especificação. Os requisitos do sistema e o papel atribuído ao *software* são refinados em detalhes. Modelo de dados, informação e fluxo de controle, bem como comportamentos operacionais necessários, são definidos. Soluções alternativas são avaliadas e um modelo de análise completo é gerado (PRESSMAN, 2002).

> As atividades da engenharia de requisitos resultam na especificação das características operacionais do *software* (função, dados e comportamento), indicam a interface do *software* com outros elementos do sistema e estabelecem restrições que o *software* deve satisfazer (PRESSMAN, 2002, p. 266).

Foram analisados os requisitos operacionais do sistema que deve coletar os dados nas estações da rede, filtrar estes dados e exibir as informações através de gráficos disponíveis na WEB de tal forma que sua interpretação seja intuitiva.

4.2 Descrição dos requisitos

O SPY007 é composto por três módulos e assim a análise foi desenvolvida focando cada módulo do sistema individualmente.

Abaixo são descritos os módulos do SPY007.

a) agente de monitoramento: disponibiliza informações que permitem identificar o aplicativo que está sendo utilizado e o endereço do *site* acessado, se estiver sendo utilizado um navegador.
b) módulo de coleta de dados: é composto por duas rotinas, o Collector e o Discovery. O Collector executa coletas nos agentes instalados nas estações da rede. O Discovery identifica quais estações estão ativas na rede.
c) módulo de gerenciamento: é a interface gráfica do sistema. Permite ao administrador da rede inserir parâmetros no sistema, visualizar os registros das operações executadas e exibe as informações coletadas na forma de gráficos.

4.2.1 Requisitos do agente de monitoramento

O agente é instalado nas estações gerenciadas e captura o título e a classe da janela ativa, o nome do usuário que está utilizando a estação, o identificador da estação e o endereço do *site* que está sendo acessado, através de rotinas previamente implementadas. Estes dados serão consultados pelo gerenciador através do protocolo SNMP.

4.2.2 Requisitos do módulo de coleta de dados

4.2.2.1 Collector

Esta rotina efetua a coleta de dados em todas as estações da rede definidas como ativas, e classifica estes dados de acordo com os aplicativos e os endereços de domínios WEB (*URL*[4]) cadastrados no SPY007.

4.2.2.2 Requisitos do Discovery

O Discovery deve consultar, com o protocolo SNMP, todos os endereços IP das redes cadastradas. De acordo com a resposta obtida de cada estação cliente, deve alterar o estado desta estação para ativa ou inativa junto ao SPY007.

4.2.3 Requisitos do módulo de gerenciamento

A interface gráfica possui acesso restrito onde o utilizador deve identificar-se mediante usuário e a senha. Através desta interface são cadastrados os aplicativos, as categorias de aplicativos e cadastrado o relacionamento entre um aplicativo e uma categoria de aplicativo. Os aplicativos devem ser únicos, onde para cada aplicativo deve ser definido um nome, o título e a classe da janela do aplicativo. A categoria de aplicativo também deve ser única sendo necessário especificar apenas um nome para a categoria. Um aplicativo pode estar relacionado com uma categoria de aplicativo, não é obrigatória a definição desta relação. Uma categoria pode ter relacionados a ela vários aplicativos. Também podem ser cadastrados

---

[4] O termo URL (*Uniform Resource Identifier* - Identificador de Recursos Uniforme) está sendo utilizado para definir o endereço de uma página qualquer disponível na Internet.

*URL*s e endereços de redes através da interface gráfica, bem como visualizar os *logs* de funcionamento do SPY007. Os *logs* registram dados referentes as coletas efetuadas nas estações, buscas por estações ativas efetuadas nos endereços de rede cadastradas e registros de operação geral do sistema como cadastro, edição e remoção de aplicativos.

Através da interface gráfica, podem ser visualizados os gráficos de utilização dos *softwares* classificados por aplicativo, categoria de aplicativo e URL. Para refinar a consulta, pode-se utilizar filtros de período de monitoramento (data e hora inicial e data e hora final), estação e/ou usuário.

4.3 Diagramas UML

A UML é uma linguagem de modelagem, não proprietária, que disponibiliza diagramas padronizados e foi projetada para auxiliar aqueles que participam da atividade de desenvolvimento de *software* a definir modelos que permitam visualizar o sistema, especificar a estrutura e o comportamento deste, construí-lo e documentar as decisões tomadas durante o processo (SCOTT, 2003).

Nas seções seguintes são apresentados os diagramas elaborados durante a análise do SPY007. Como o funcionamento do aplicativo e sua estrutura são otimizados, optou-se por elaborar os diagramas de caso de uso, classes e entidade relacionamento.

4.3.1 Diagramas de caso de uso

A utilização dos diagramas de caso de uso é uma técnica baseada em cenários para a obtenção dos requisitos do sistema. Esses diagramas tornaram-se uma característica fundamental da notação UML para descrever modelos de sistemas orientados a objetos. Estes diagramas identificam os agentes envolvidos em uma interação e especifica o tipo da interação (SOMMERVILLE, 2003). É importante ressaltar que no diagrama de caso de uso o cliente deve ser capaz de identificar facilmente as principais funcionalidades de seu sistema.

O ator, ou agente, do caso de uso é um usuário do sistema que pode ser um usuário humano ou um sistema computacional, enquanto que o caso de uso, ou interação, define uma

grande função do sistema, porém esta função pode ser estruturada em outras funções, assim um caso de uso pode ser estruturado em outros.

O SPY007 possui como atores o usuário administrador do sistema, o agente instalado nas estações, e duas rotinas internas: o Collector e o Discovery.

Abaixo são listados os quatro principais diagramas de caso de uso do SPY007:

a) agente

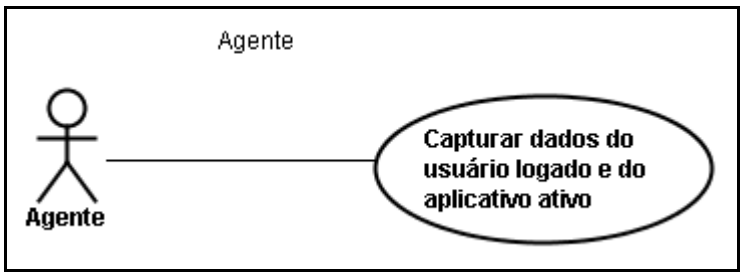

Figura 10 - Diagrama de caso de uso do agente

Capturar dados do usuário e do aplicativo ativo

- Atores: agente.
- Descrição: capturar o título e a classe da janela do aplicativo que está em primeiro plano na estação, URL acessada, e os dados do usuário que está utilizando a estação.
- Dados: título da janela, classe da janela, endereço do *site* acessado, usuário logado.
- Resposta: disponibilizar os dados para o gerenciador.

b) Collector

Figura 11 - Diagrama de caso de uso do Collector

Coletar dados nas estações monitoradas

- Atores: Collector.
- Descrição: buscar nas estações monitoradas o identificador da estação, o usuário logado, dados sobre a aplicação em primeiro plano e endereço do *site* WEB que está sendo acessado no momento. O Collector faz a identificação do aplicativo que está sendo executado com base nos dados dos aplicativos cadastrados. No momento da coleta também é realizada a identificação do domínio WEB que está sendo acessado (*URL*), se o domínio não estiver cadastrado, o Collector faz o cadastro da *URL* no banco de dados.
- Dados: identificador da estação, usuário logado, título e classe da janela em primeiro plano e endereço *URL* que está sendo acessado no momento.
- Resposta: se a estação está devidamente cadastra, e os dados foram obtidos com sucesso, inserção dos dados coletados no banco de dados, caso contrário, apenas é feito o registro do erro na tabela de *logs* do SPY007.

c) Discovery

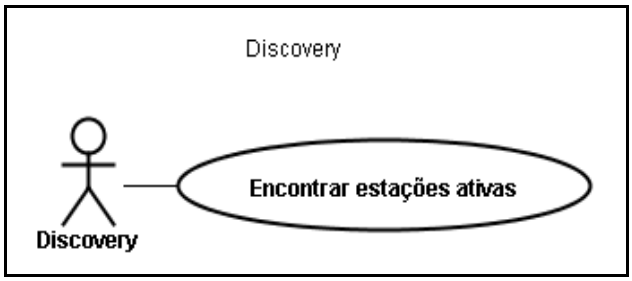

Figura 12 - Diagrama de caso de uso do Discovery

Encontrar estações ativas

- Atores: Discovery.
- Descrição: consultar todos os endereços IP de todas as redes cadastradas.
- Dados: consultar o identificador das estações.
- Resposta: se obtiver o identificador da estação, alterar o estado da estação para ativa, caso contrário, apenas é feito o registro do erro na tabela de *logs* do SPY007.

d) interface gráfica

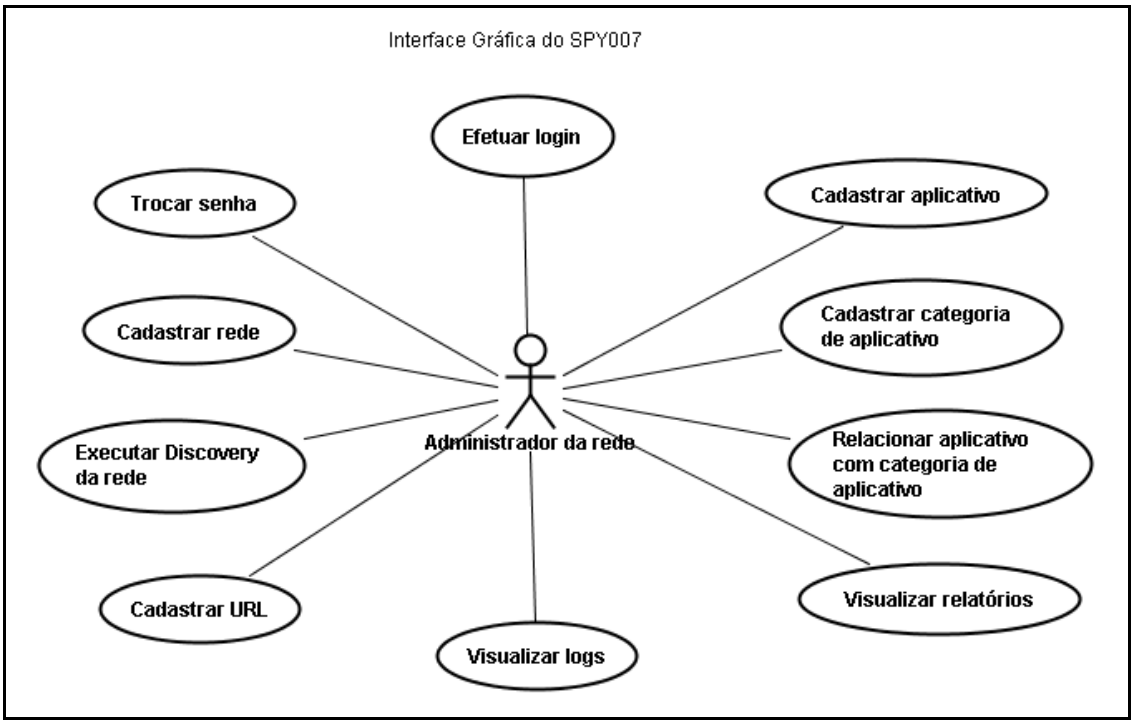

Figura 13 - Diagrama de caso de uso da interface gráfica

Efetuar login

- Atores: administrador do sistema.
- Descrição: para acessar o sistema é preciso que o usuário se identifique através de um *login* e uma senha. Existe apenas um tipo de usuário, o administrador que possui acesso a todos os recursos do sistema.
- Dados: usuário e senha.
- Resposta: acesso ao sistema ou mensagem de erro.

Cadastrar aplicativo

- Atores: administrador do sistema.
- Descrição: o administrador do sistema deve cadastrar os aplicativos. Baseado nas informações cadastradas dos aplicativos o Collector fará a identificação do aplicativo quando coletar os dados das estações.
- Dados: nome do aplicativo, título e classe da janela.
- Resposta: mensagem de sucesso ou erro e registro da operação em banco de dados.

Cadastrar categoria de aplicativo

- Atores: administrador do sistema.
- Descrição: o administrador do sistema deve cadastrar as categorias de aplicativos.
- Dados: nome da categoria de aplicativo.
- Resposta: mensagem de sucesso ou erro e registro da operação em banco de dados.

Relacionar aplicativo com categoria de aplicativo

- Atores: administrador do sistema.
- Descrição: é exibida uma lista com os aplicativos já cadastrados no sistema e ainda não relacionados com nenhuma categoria, e uma lista com as categorias de aplicativos cadastrados. O administrador deve estabelecer a qual categoria de aplicativo cada aplicativo pertence.
- Dados: aplicativo e categoria de aplicativo.
- Resposta: mensagem de sucesso ou erro e registro da operação em banco de dados.

Visualizar relatórios

- Atores: administrador do sistema.
- Descrição: é exibida uma lista com todas as estações cadastradas, uma lista com os usuários cadastrados e campos onde o usuário pode selecionar o período de início e período de fim.
- Dados: início do período, fim do período, usuário, estação.
- Resposta: gráficos de utilização por categoria de aplicativo, aplicativo e domínio WEB acessado.

Visualizar *logs*

- Atores: administrador do sistema.
- Descrição: é exibida uma lista com os módulos do sistema, *status* da operação, e campos onde o usuário pode selecionar o período de início e período de fim. O administrador deve selecionar o período e o módulo não sendo obrigatório selecionar um *status*.

- Dados: início do período, fim do período, módulo e *status*.
- Resposta: *logs* referentes ao filtro definido.

Cadastrar URL

- Atores: administrador do sistema.
- Descrição: o administrador pode cadastrar, excluir ou substituir uma URL através da interface gráfica.
- Dados: URL.
- Resposta: mensagem de sucesso ou erro na interface gráfica e registro da operação no banco de dados.

Executar Discovery da rede

- Atores: administrador do sistema.
- Descrição: é exibida uma lista com as redes cadastradas onde o administrador deve selecionar uma rede na qual deve ser feita a consulta a todos os IP's a fim de identificar se é possível obter o identificador da máquina. Se não for possível obter o identificador da máquina, é enviado um pacote *ping* (ICMP) para o endereço IP em questão a fim de analisar a resposta ao mesmo.
- Dados: rede.
- Resposta: lista com todos os endereços IP da rede seleciona e respectiva resposta do SNMP e do *ping*.

Cadastrar rede

- Atores: administrador do sistema.
- Descrição: o administrador deve informar a uma rede classe C a ser cadastrada. Estas redes cadastradas são utilizadas pelo Discovery para identificar as estações ativas em cada rede.
- Dados: rede.
- Resposta: mensagem de sucesso ou erro e registro da operação em banco de dados.

Trocar senha

- Atores: administrador do sistema.
- Descrição: o administrador pode alterar a senha de acesso ao sistema. É exibida uma tela com campos de senha e confirmação de senha.
- Dados: senha e confirmação da senha.
- Resposta: mensagem de sucesso ou erro.

4.3.2 Diagrama de classes

Uma classe, em linguagem orientada a objetos, representa a possibilidade de combinar em um único registro, campos de dados e campos de funções. O diagrama de classes é um dos mais importantes da documentação do sistema, nele são definidas informações sobre métodos, atributos, nome das funções e como estas serão integradas. Descreve os vários tipos de objetos no sistema e o relacionamento entre eles.

Na Figura 14 é possível visualizar o diagrama de classes do SPY007.

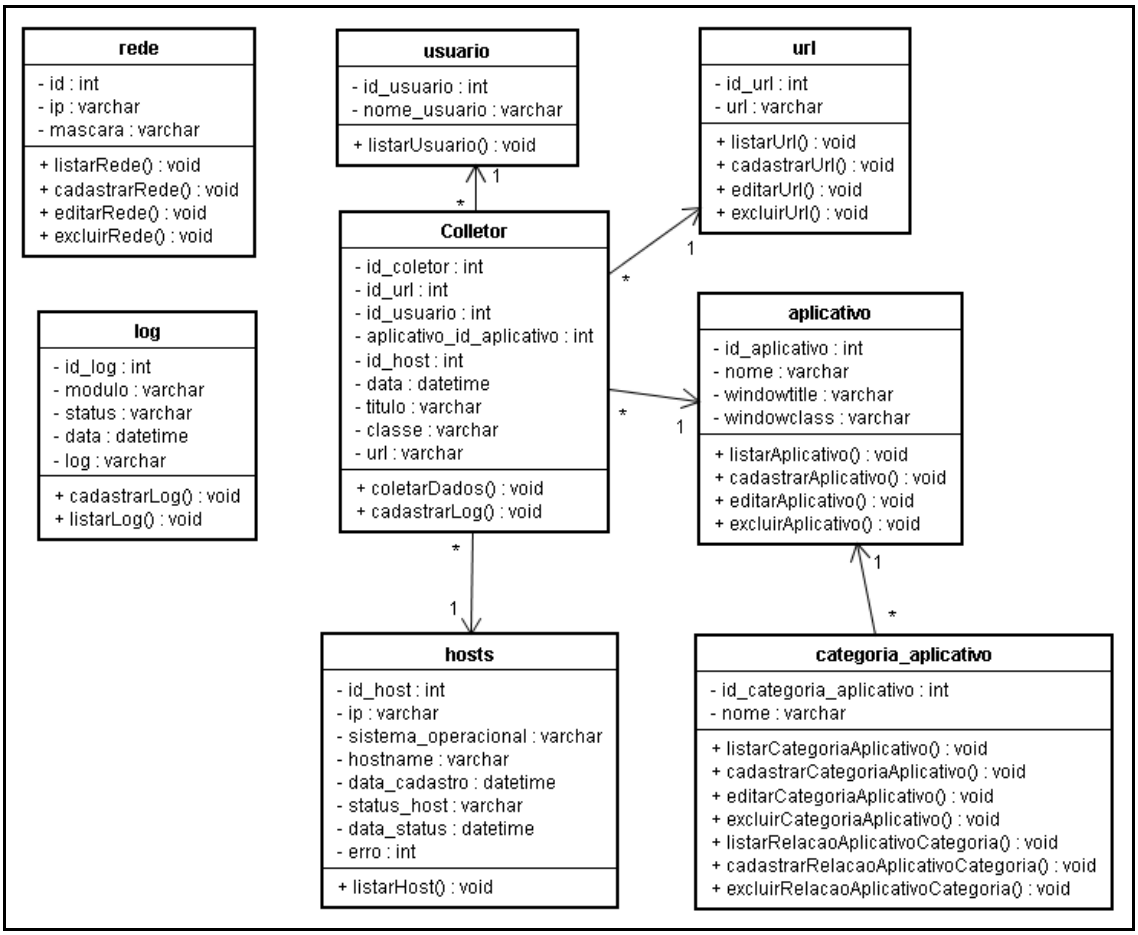

Figura 14 - Diagrama de classes

4.3.3 Diagrama entidade relacionamento

Os diagramas de entidade relacionamento também chamados diagramas ER, mostram o desenho conceitual dos aplicativos de banco de dados. Eles definem as várias entidades (conceitos) no sistema de informação e as relações e restrições entre eles. Uma entidade é considerada qualquer conceito no mundo real com uma existência independente. Poderá ser um objeto com uma existência física ou conceitual. Uma associação (ou relação) interliga várias entidades.

Na Figura 15 é apresentado o diagrama ER do sistema SPY007.

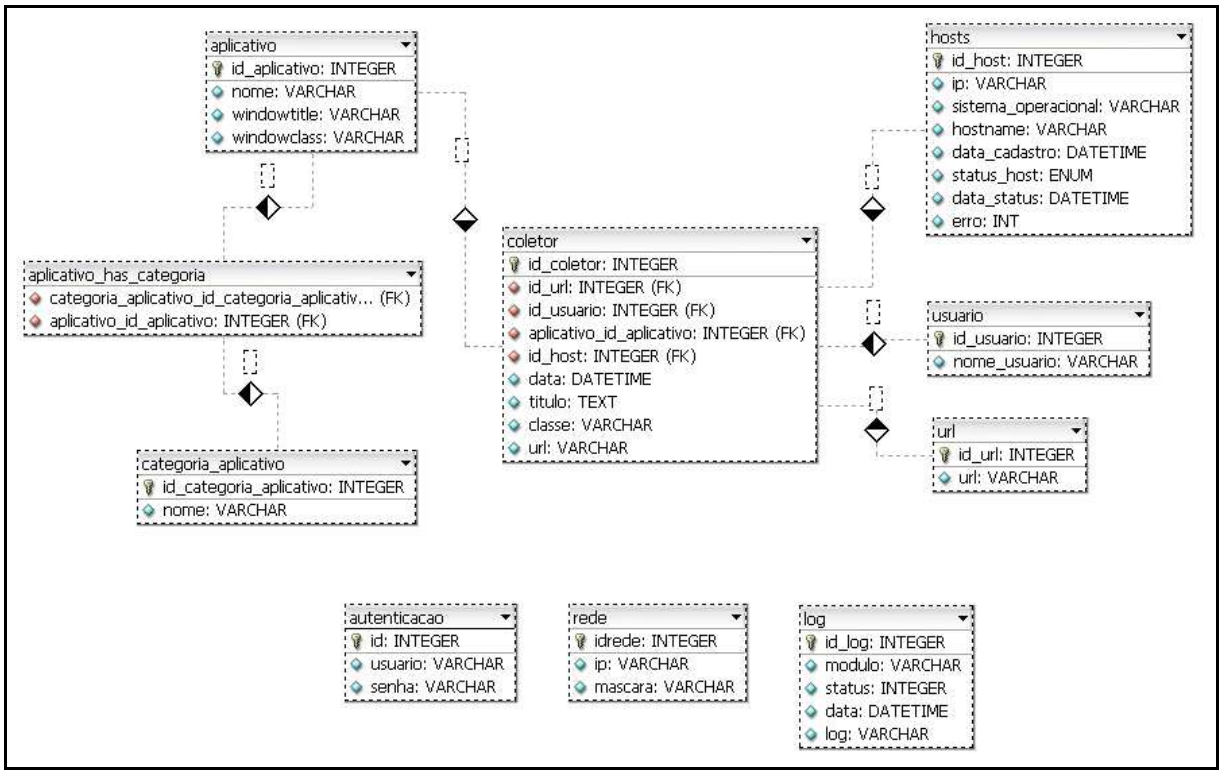

Figura 15 - Diagrama ER

4.4 Padrão de projeto

O padrão de projeto utilizado no desenvolvimento do SPY007 é MVC que de acordo com Fowler (2006) é atualmente um dos padrões mais citados. O MVC começou como um *framework*[5] desenvolvido por Trygve Reenskaug para a plataforma *SmallTalk* no final dos anos 70. Desde então, ele tem exercido um papel de influência na maioria dos *frameworks* voltados para a interface com o usuário.

O MVC considera o sistema baseado em três papéis: *model*, *view*, *controller*. O *model* é um objeto que representa alguma informação sobre o domínio. É um objeto não-visual, contendo todos os dados e comportamentos que não os utilizados pela interface do usuário. A *view* representa a exibição do modelo na interface com o usuário. A *view* diz respeito apenas à

[5] Framework: em desenvolvimento do *software* é uma abstração que une códigos comuns entre vários projetos de *software* provendo funcionalidades genéricas.

apresentação das informações, qualquer alteração nessas informações deve ser manipulada pelo *controller*. O *controller* recebe a entrada do usuário, manipula o *model* e faz com que a *view* seja atualizada apropriadamente. Assim a interface do usuário é uma combinação de *view* e *controller* (FOWLER, 2006).

4.5 Padrão de desenvolvimento

O padrão de desenvolvimento utilizado na codificação do SPY007 foi o orientado a objetos (OO). De acordo com Sommerville (2003), a programação orientada a objetos se ocupa de realizar um projeto de *software* utilizando uma linguagem de programação OO, como por exemplo, a linguagem PHP, que aceita a implementação direta de objetos e fornece recursos para definir as classes de objetos.

Orientação a objetos é uma estrutura que implica em organizar o *software* como uma coleção de objetos discretos que incorporam tanto estrutura de dados como comportamento. Objetos de *software* são modelados de forma semelhante aos objetos do mundo real com estados e comportamentos que são comumente chamados de atributos e métodos, respectivamente. Um objeto de *software* mantém o seu estado em uma ou mais variáveis enquanto que o seu comportamento é definido por funções. A OO, quando utilizada corretamente, oferece ganhos em termos de rapidez, custo, confiabilidade, flexibilidade e facilidade de manutenção.

5 TECNOLOGIAS

Neste capítulo serão listadas e explicadas cada uma das tecnologias empregadas no desenvolvimento do SPY007 desde o ambiente de programação, às linguagens e demais serviços associados.

Como o SPY007 é composto por um agente SNMP instalado nas estações monitoradas, um gerente (coleta as informações dos agentes) e uma interface gráfica para administração do sistema, optou-se por explicar as tecnologias de acordo com a sua utilização no desenvolvimento dos módulos do SPY007.

5.1 Interface gráfica, Collector e Discovery

O Collector e o Discovery são rotinas desenvolvidas utilizando a linguagem PHP, onde o Collector consulta todas as estações cadastradas e ativas a fim de obter as dados de monitoramento. O Discovery consulta todos os endereços IP de todas as rede cadastradas no SPY007 e, baseado nas respostas obtidas, define quais estações estão ativas e que por consequência serão consultadas, posteriormente, pelo Collector.

A interface gráfica, além de disponibilizar funções de administração do SPY007, possibilita ao administrador da rede visualizar as informações coletadas nas estações em forma de gráficos.

Nas seções seguintes são detalhadas as tecnologias empregadas para o desenvolvimento da interface gráfica, Collector e Discovery.

5.1.1 APACHE

O Apache é um servidor HTTP, que tem por função publicar documentos na Internet que serão acessados pelos usuários utilizando um navegador. O Apache HTTP *Server* é um projeto colaborativo de desenvolvimento de *software* que visa o aperfeiçoamento do servidor Apache. Este projeto é desenvolvido por um grupo de pessoas voluntárias que estão espalhadas ao redor do mundo e que utilizam a Internet como meio de comunicação, planejamento e desenvolvimento do servidor e da sua documentação. Este projeto é parte do Apache *Software Fundation* que desenvolve diversos outros projetos.

A primeira versão do Apache foi lançada em abril de 1995 como uma melhoria do NCSA (*National Center for Supercomputing Applications*). Segundo dados publicados no *site* Netcraft[6] o Apache é o servidor HTTP mais utilizado, sendo empregado para hospedar praticamente 50% dos domínios existentes, ver Figura 16.

Developer	September 2009	Percent	October 2009	Percent	Change
Apache	105,416,925	46.62%	108,078,535	46.90%	0.28
Microsoft	49,615,010	21.94%	49,723,999	21.58%	-0.37
qq.com	30,069,048	13.30%	30,069,136	13.05%	-0.25
Google	13,767,338	6.09%	13,819,947	6.00%	-0.09
nginx	12,676,238	5.61%	13,813,997	5.99%	0.39

Figura 16 - Utilização de servidores HTTP em relação aos *sites* ativos.
Fonte: Netcraft (2009, p. 1)

O servidor Apache tem seu código fonte disponível na Internet e a sua utilização não requer uma licença, caracterizando-se assim como um aplicativo *open source*. É um servidor multiplaforma, que pode ser instalado sob a plataforma Windows, GNU/Linux ou outra suportada.

O servidor Apache está sendo utilizado no SPY007 para publicar a interface gráfica na WEB para os administradores da rede consultarem os dados nela contidos.

---

[6] Netcraft é um *site* que publica informações sobre serviços na Internet. Entre eles, serviço de segurança na Internet, anti-fraude, testes de aplicações, revisão de códigos. Provê ainda informações e análises sobre diversos aspectos da Internet.

5.1.2 MySQL

O MySQL é um dos SGBD's (Sistema Gerenciador de Banco de Dados) *open source* mais populares da atualidade. É desenvolvido e distribuído pela empresa MySQL AB. Possui suporte a banco de dados relacionais e linguagem SQL (*Structered Query Language* – Linguagem Estrutural de Consultas). O MySQL é um sistema cliente/servidor que consiste de um servidor SQL multitarefa, multiplataforma. Este SGBD possui diversos programas clientes e bibliotecas, ferramentas administrativas e um grande número de interfaces de programação (MYSQLAB, 2009).

Atualmente, o MySQL pertence à Oracle Inc. que comprou recentemente a Sun Microsystems que, por sua vez, havia adquirido a MySQLAB.

5.1.3 NetBeans

O NetBeans foi utilizado no SPY007 como ambiente de desenvolvimento para os códigos escritos utilizando a linguagem PHP. É um ambiente de desenvolvimento integrado (IDE – *Integrated Development Environment*) multiplataforma, de código aberto que suporta diversas linguagens de programação, entre elas o PHP.

Um ambiente de desenvolvimento integrado é um aplicativo que reúne características e ferramentas de apoio ao desenvolvimento de *software* com o objetivo de agilizar o processo de desenvolvimento. As IDE possuem algumas características comuns, como um editor de código-fonte nas linguagens suportadas pelo ambiente: um compilador que transforma o código escrito em uma linguagem de programação determinada para a linguagem de máquina tornando o arquivo executável, um *linker* que interliga os vários códigos compilados em linguagem de máquina formando assim um programa executável; um depurador que é uma ferramenta que auxilia no processo de encontrar e corrigir erros no código-fonte do programa que está sendo desenvolvido; uma ferramenta de modelagem que possibilita a criação de modelos de classes, objetos, interfaces, associações e interações dos artefatos envolvidos no *software*. Algumas IDE's ainda disponibilizam ferramentas de geração de código e automatização de testes.

No mercado existem diversos ambientes de desenvolvimento integrado para diversas linguagens e com inúmeros recursos. A equipe de desenvolvimento deve analisar as necessidades para o desenvolvimento do projeto e, baseado nas funcionalidades oferecidas, escolher uma IDE que melhor se adapte ao projeto a ser desenvolvido.

O projeto NetBeans pertence à Sun Microsystems, que disponibilizou o código fonte do NetBeans no ano 2000 tornando-o uma plataforma *open source*.

A IDE em questão é arquitetada em uma forma de estrutura reutilizável, que visa simplificar o desenvolvimento e aumentar a produtividade, pois reúne em um único ambiente diversas funcionalidades. O NetBeans é totalmente escrito em Java[7]. Também suporta linguagens de marcação como XHTML e o XML que serão detalhados nas seções posteriores.

O NetBeans fornece um ambiente sólido para a criação de projetos e módulos, possui um grande conjunto de bibliotecas, módulos e API's (*Application Program Interface*) além de uma vasta documentação em diversos idiomas. Foi escolhido como ambiente de desenvolvimento do SPY007 por possuir diversas facilidades para desenvolvimento orientado a objetos, suporte a linguagem PHP e experiência prévia dos programadores envolvidos no desenvolvimento do sistema.

5.1.4 Linguagem PHP

PHP é uma linguagem de script interpretada, o*pen source*, muito difundida e especialmente utilizada para o desenvolvimento de aplicações WEB em associação com o HTML (PHP.NET, 2009).

Surgiu por volta de 1995, desenvolvida por Rasmus Lerdof, inicialmente como simples *scripts* Perl como estatística de acesso para seu currículo *online* com o nome de "*Personal Home Page Tools*". Como mais funcionalidades foram requeridas, Rasmus escreveu uma implementação em C que era capaz de comunicar-se com uma base de dados, e possibilitava aos usuários desenvolver simples aplicativos dinâmicos para WEB. Atualmente a versão oficial do PHP é a 5, porém a versão 6 está em desenvolvimento.

---

[7] Java é uma linguagem de programação independente da plataforma, com a qual é possível desenvolver uma variedade enorme de aplicativos.

O PHP é uma combinação de linguagem de programação e servidor de aplicação deixando a parte do cliente mais leve, não causando demora no processamento das páginas. É portável, podendo ser executado em diversos sistemas operacionais, possui código nativo para interagir com diversos bancos de dados (SOARES, 2000).

A linguagem PHP foi utilizada para desenvolver o Collector, o Discovery e a interface gráfica do SPY007. Cabe aqui ressaltar, que o PHP está sendo utilizado para fazer todo o processamento da informação, porém, os dados são exibidos para o usuário através da biblioteca gerenciadora de *templates* Smarty que será detalhada abaixo.

5.1.5 Smarty

O Smarty é uma biblioteca PHP para gerenciar *templates*. Com ele é possível controlar facilmente a separação da aplicação lógica (código PHP) do conteúdo de sua apresentação (*tags* XHTML). Assim uma mudança na parte lógica do programa PHP não afeta em nada a maneira como os dados são exibidos.

Um aspecto interessante do Smarty é o seu sistema de compilação de *templates*. O Smarty lê os arquivos de *templates* e cria scripts PHP a partir deles. Uma vez criados, eles são executados sem a necessidade de uma nova compilação. Com isso, os arquivos de *templates* não são analisados toda vez que uma página é solicitada. Cada *template* tem a vantagem de utilizar a solução de *cache*[8] do compilador PHP (SMARTY, 2007).

[8] Cache é um dispositivo de acesso rápido, interno a um sistema. O termo cache está sendo utilizado para representar a situação onde um arquivo é interpretado e fica disponível para o sistema sem a necessidade de ser interpretado novamente quando for acessado.

Na Figura 17 é representado o funcionamento da biblioteca Smarty.

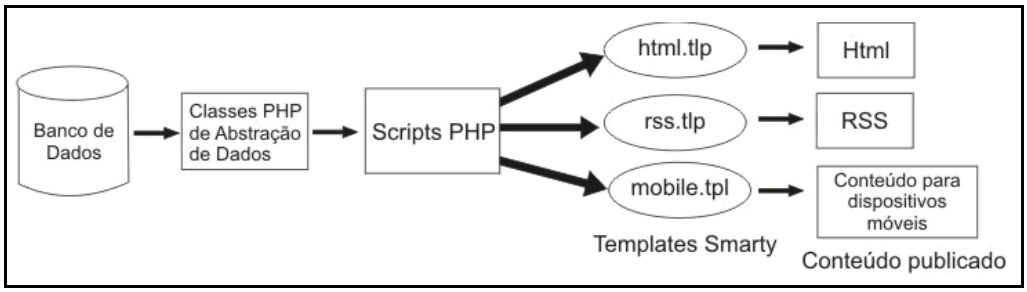

Figura 17 - Funcionamento Smarty
Fonte: Alencar (2009, p. 1)

A biblioteca Smarty está sendo utilizada, no SPY007, para exibir aos usuários os dados processados pelo PHP. A linguagem de exibição dos dados que está sendo utilizada é o XHTML com folhas de estilo CSS, como se verá nos próximos tópicos.

5.1.6 JpGraph

A JpGraph é uma biblioteca gráfica orientada a objetos, desenvolvida em PHP, que pode ser utilizada para criar numerosos tipos de gráficos. É liberada sob uma licença dupla, uma não comercial para uso educacional ou *open source*, e a licença profissional para uso comercial. Esta biblioteca foi utilizada para gerar os gráficos em forma de pizza e os gráficos em linha disponíveis no SPY007.

5.1.7 XHTML

XHTML é uma reformulação da linguagem HTML, porém mais restrita. Um documento XHTML é construído com as *tags* HTML no formato de um arquivo XML (*Extensible Markup Language*) (W3SCHOOLS, 2009).

> O padrão XHTML requer que os documentos se adaptem a rígidas regras estruturais de modo que os aplicativos possam lê-los. (LEMAY, 2002, p. 338)

De acordo com Maujor (2009) todas as linguagens de marcação da WEB são baseadas em SGML, uma metalinguagem complexa, projetada para máquinas com a finalidade de servir de base para criação de outras linguagens. O SGML foi usado para criar a XML (*Extensible Markup Language*), também uma metalinguagem, porém bem mais simples. Com o XML é possível definir *tags* e atributos para escrever documentos WEB. O XHTML foi desenvolvido baseado neste conceito e por isso é uma aplicação XML. As *tags* e atributos do XHTML foram criados aproveitando as já conhecidas *tags* e atributos do HTML 4.01 e suas regras.

Assim, quando se utiliza o XHTML, efetivamente está sendo escrito um código XML, onde as *tags* e atributos já estão definidas, e isto proporciona todos os benefícios do XML sem as complexidades do SGML. O XHTML foi escrito para substituir o HTML, e nada mais é do que um HTML mais “puro, claro e limpo”.

Uma das grandes vantagens do XHTML é a compatibilidade com as futuras aplicações de usuários, garantindo, desde já, que os códigos escritos em XHTML serão estáveis por um longo período. A tendência é a de que futuras versões de navegadores de Internet deixem de suportar elementos e atributos já em desuso (*deprecated*). O XHTML é um código consistente que dispensa uso de *hacks* para contornar *bugs* e é totalmente compatível com todas as aplicações de usuários escritas em HTML.

O XHTML foi utilizado no SPY007 para estruturar os conteúdos e exibi-los para os usuários enquanto que, a formatação da exibição do conteúdo foi feita utilizando CSS.

5.1.8 CSS (*Cascading Style Sheets*)

O CSS (Folhas de Estilo em Cascata) é um padrão de formatação para documentos HTML/XHTML, que permite ao *designer* um controle maior sobre os atributos tipográficos de um *site*, como tamanho e cor da fonte, espaçamento entre linhas e caracteres, e margem do texto. Segundo Macedo (2006) este padrão introduziu também a utilização de camadas, permitindo a sobreposição de texto sobre texto ou sobre imagens. Com a utilização das folhas de estilo, é possível separar o estilo do conteúdo em um documento HTML/XHTML. Sendo o CSS responsável pelo *design* (posicionamento, cores, fontes). Já a separação da estrutura do documento em blocos de informação (títulos, cabeçalhos, parágrafos) é definida pelas *tags* HTML/XHTML.

5.1.9 JavaScript

O JavaScript foi desenvolvido originalmente por Brendan Eich da Netscape Communications em meados de 1995, sob o nome Mocha, depois renomeado para Livescript e finalmente JavaScript. É uma linguagem de programação do lado cliente (é executado no navegador do cliente) amplamente utilizada por desenvolvedores de *websites*[9] ou aplicações WEB. Geralmente o JavaScript é utilizado para criar efeitos especiais nas páginas e definir interatividade com o usuário. O navegador do cliente é encarregado de interpretar as instruções JavaScript e executá-las para realizar os efeitos e as funções de interatividades. Assim, o maior recurso com que conta esta linguagem é o próprio navegador do cliente.

JavaScript é uma linguagem com muitas possibilidades, permite a programação de pequenas rotinas, mas também de programas maiores, orientados a objetos, com funções e estruturas de dados complexas. Apesar de a palavra JavaScript se parecer muito com Java, que é outra linguagem de programação, ambas não compartilham muitas semelhanças entre si. Basicamente, possuem métodos parecidos, porém a sintaxe do JavaScript se parece mais com a da linguagem PHP e C++.

A linguagem JavaScript foi utilizada para fazer algumas validações de informações digitas pelo usuário. Cabe ressaltar que outras validações foram efetuadas, já que o JavaScript é executado no lado do cliente. O JavaScript também é utilizado como parte da tecnologia Ajax, que será explicada abaixo e que foi utilizada para inserir alguns efeitos e funcionalidades no SPY007.

5.1.10 Ajax

Ajax (*Asynchronous JavaScript And* XML) é um conceito para o desenvolvimento de aplicações *online*. O Ajax não é uma nova tecnologia mas uma nova forma de utilizar tecnologias já existentes, como *JavaScript*, CSS e XML, de uma maneira totalmente revolucionária, a qual permite a total interação entre o usuário e um *site* WEB.

---

[9] *Website* possui o mesmo significado que *site*, um conjunto de páginas na *WEB*.

> [...] a Internet evoluiu desde suas primeiras versões quando era uma maneira dos pesquisadores se conectarem e compartilharem informações. Ela começou com simples navegadores textuais e páginas estáticas, mas agora é difícil encontrar uma empresa que não tenha um *site* sofisticado. [...] Desenvolvedores cansados da dificuldade de implantar aplicativos clientes pesados para milhares de usuários se voltaram para a WEB para facilitar seu trabalho. (ASLESON e SCHUTTA , 2006, p. 14).

A utilização do Ajax visa tornar as aplicações WEB mais interativas e dinâmicas. Um exemplo típico de aplicações WEB é o usuário digitar um endereço e receber uma página com formulários, imagens, *links* etc. Qualquer ação que usuário promover na página, fará com que seja feita uma requisição que irá carregar novamente toda a página. Com a utilização de técnicas de Ajax, será recarregado apenas o conteúdo que o usuário solicitou e com muito mais interatividade. Neste caso, o aplicativo é executado diretamente no navegador do cliente que solicita ao servidor apenas os dados desejados pelo usuário.

> O conceito básico do Ajax é permitir que o usuário navegue pelo *site* de forma assíncrona, ou seja, acesse diferentes conteúdos ao mesmo tempo, faça ações diversas sem a necessidade de esperar enquanto o *site* processa as informações. A cada nova ação ou solicitação do usuário, nem sempre será preciso carregar a mesma página ou outra diferente (CLOPER, 2006, p. 1).

5.2 Agente

O agente do SPY007 será detalhado no próximo capítulo. Em uma breve explanação sobre o agente SPY007, deve-se ressaltar que o agente armazena as informações relevantes ao SPY007 e quando consultado pelo gerente fornece tais dados. Efetivamente, o agente do SPY007 utiliza as funções do agente SNMP, o NetSNMP, e através de scripts escritos na linguagem C, consulta os dados relevantes ao monitoramento.

O agente SPY007 é multiplataforma, podendo ser utilizado no sistema operacional Windows e GNU/Linux. Nas seções que se sucedem, serão apresentadas as tecnologias empregadas no desenvolvimento do agente do SPY007.

5.2.1 Borland Delphi

O ambiente Delphi engloba um compilador, uma IDE e uma linguagem de programação, a Object Pascal. O Delphi é desenvolvido pela Borland e possui inúmeras bibliotecas de componentes visuais que proporcionam um desenvolvimento rápido das aplicações. É uma linguagem orientada a objetos e eventos que é compilada. Esta linguagem é direcionada para a plataforma Windows, no entanto chegou a se utilizada para o desenvolvimento de aplicações nativas Linux e MacOS através do Kylix (uma IDE para linguagens C++ e Object Pascal).

É amplamente utilizada no desenvolvimento de aplicações desktop, aplicações multicamadas e cliente/servidor, compatível com os bancos de dados mais conhecidos do mercado. Pelas facilidades já mencionadas, o Delphi foi utilizado no SPY007 para criar o serviço Windows, que é executado nas estações monitoradas, que utilizam o sistema operacional Windows. Serviço este que identifica a aplicação que está sendo utilizada e grava informações sobre a aplicação no registro do Windows. Posteriormente estas informações serão consultadas pelo agente NetSNMP através de rotinas específicas implementadas para tal.

5.2.2 Linguagem C++

A linguagem de programação C++ foi desenvolvida por Bjarne Stroustrup na empresa AT&T a partir da linguagem C. O objetivo inicial era agregar o conceito de classes e orientação a objetos à linguagem C. As linguagens C e C++ não são totalmente compatíveis, uma vez que a sintaxe e a semântica de algumas construções são diferentes. Todo programa escrito em C++ pode ser visto como um conjunto de funções. O C++ é uma linguagem de alto nível com algumas facilidades de baixo nível, multi-paradigma e de uso geral.

A linguagem C++ foi utilizada para desenvolver as rotinas que acessam o registro do Windows em busca das informações de monitoramento do SPY007. Estas rotinas são necessárias apenas no agente que é instalado no sistema operacional Windows, que será explicado nas seções seguintes.

5.2.3 Net-SNMP

O Net-SNMP é um conjunto de utilitários e aplicações usados para implementar SNMPv1, SNMPv2, SNMPv3. O Net-SNMP é um aplicativo multiplataforma *open source*, cujo objetivo principal é o monitoramento e configuração de dispositivos e serviços de rede (NET-SNMP, 2009). Com o Net-SNMP é possível configurar rotinas externas que devem ser executadas para monitorar algo que não seja monitorado nativamente. Foi exatamente esta a funcionalidade utilizada para obter informações sobre os aplicativos que estão sendo executados em primeiro plano nas estações. A versão do SNMP utilizada para efetuar as consultas nas estações foi a dois (SNMPv2). Nesta versão do SNMP os dados trafegam na rede sem criptografia e o controle de acesso é feito através de *communities* que funcionam como uma senha de acesso.

5.2.4 GNU/Linux

Usualmente o sistema operacional (SO) GNU/Linux é denominado apenas Linux, porém Linux é o *kernel*[10] do sistema. As demais ferramentas que compõem o SO são desenvolvidas pelo projeto GNU, por isso a nomenclatura correta é GNU/Linux (Ferramentas/*Kernel*). O GNU/Linux é um sistema operacional multitarefa, multiusuário e *open source*. Devido às características mencionadas, e principalmente ao código fonte aberto, existem diversas distribuições GNU/Linux com peculiaridades e objetivos próprios.

O *kernel* Linux foi desenvolvido por estudante da Universidade de Helsinki, na Finlândia. O nome deste universitário é Linus Torvalds que nos dias atuais continua ativo no projeto de desenvolvimento do *kernel* Linux juntamente com outros colaboradores (DANESH, 2000).

O projeto GNU possui grupo de programadores que trabalham voluntariamente a fim de aprimorar e desenvolver novas ferramentas para o SO GNU/Linux. Todos os resultados dos esforços do projeto GNU são disponibilizados gratuitamente na Internet juntamente com

---

[10] *Kernel* é denominado o núcleo de um sistema operacional. No *kernel* temos a camada de *software* mais próxima do *hardware* onde são definidas as interações entre a parte lógica e a física do computador.

o código fonte, permitindo assim que qualquer pessoa modifique as ferramentas ou faça adaptações e melhorias, de acordo com as suas necessidades. No mundo GNU/Linux o sistema operacional é gratuito e existem empresas especializadas em prestar serviços. A base da renda obtida através do GNU/Linux está justamente nos serviços que são prestados.

O agente do SPY007 foi testado utilizando a distribuição GNU/Linux Ubuntu na sua versão 9.04 com ambiente gráfico Gnome.

5.2.5 Microsoft Windows

Microsoft Windows é uma família de sistemas operacionais criados pela Microsoft, empresa fundada em 1975 por Bill Gates e Paul Allen. Atualmente a Microsoft é uma das empresas desenvolvedoras de *software* de maior notoriedade no cenário mundial, sendo seus sistemas operacionais os mais utilizados nos computadores pessoais. O Windows é um *software* proprietário, sendo necessário adquirir uma licença de uso para sua utilização.

O Windows é caracterizado pela sua *interface* com estrutura de janelas, daí o nome Windows que significa justamente janela.

O agente do SPY007 foi testado utilizando o sistema operacional da Microsoft Windows XP Professional.

6 RESULTADO: SISTEMA DE MONITORAMENTO SPY007

Como resultado deste trabalho, foi desenvolvido um sistema de monitoramento da utilização de *softwares* nas estações de trabalho que compõem uma rede de computadores. O SPY007, como é chamado o *software*, monitora exclusivamente as aplicações que estão sendo executadas em primeiro plano na estação, ou seja, aquelas que estão de fato sendo utilizadas pelo usuário. O sistema foi desenvolvido utilizando conceitos e ferramentas de gerência de rede, onde a interface de administração pode ser acessada através da WEB.

A Figura 18 demonstra a organização do SPY007, que é divido em três módulos:

a) agente de monitoramento: é instalado nas estações a serem monitoradas;
b) módulo de coleta de dados: é composto por duas rotinas, o Collector e o Discovery, as quais devem ser instaladas no servidor do SPY007;
c) módulo de gerenciamento: é a interface gráfica do SPY007, onde são visualizados os gráficos com os dados coletados nas estações monitoradas e feito o gerenciamento do sistema em geral do sistema.

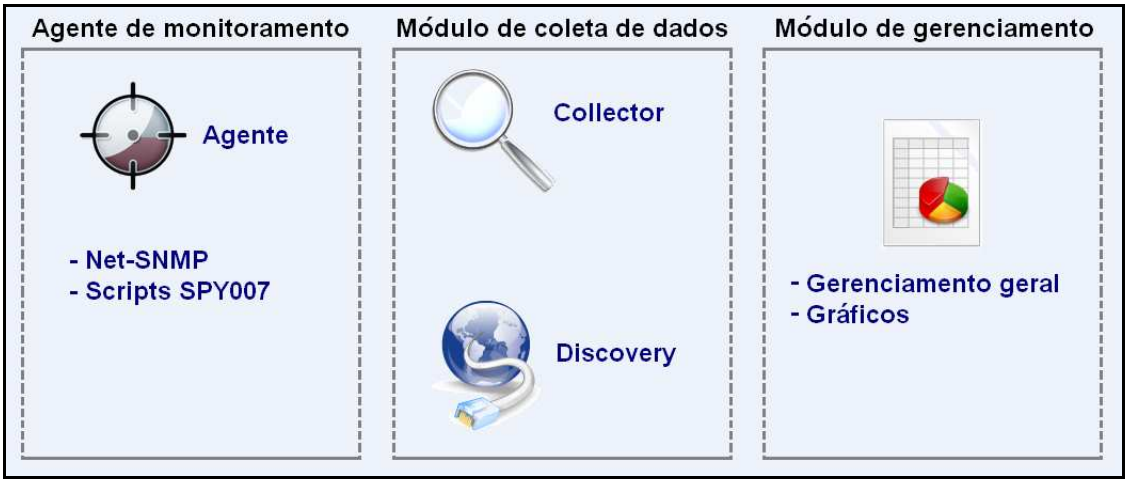

Figura 18 - Módulos do SPY007

Nas seções seguintes será explicado em detalhes o funcionamento de cada um dos módulos do SPY007.

6.1 Funcionamento do SPY007

O SPY007 é um aplicativo de gerência de rede que tem seu funcionamento baseado no modelo agente/gerenciador, nas estações que se deseja monitorar, deve ser instalado o agente, e o gerenciador envia consultas SNMP aos dispositivos gerenciados, neste caso as estações que devem ser monitoradas.

Na Figura 19 é possível visualizar o diagrama de funcionamento do SPY007.

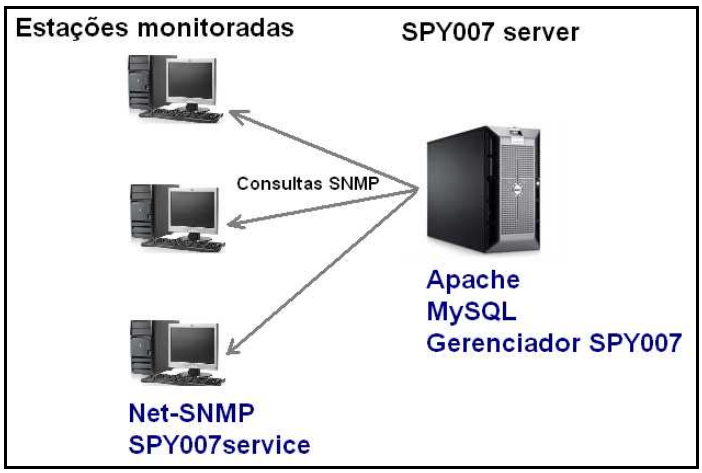

Figura 19 - Funcionamento do SPY007

6.1.1 Agente de monitoramento

O agente de monitoramento tem um funcionamento diferenciado de acordo com o sistema operacional onde é instalado.

O agente de monitoramento para estações com sistema operacional Windows é composto por um serviço (SPY007service) que é executado na estação monitorada, sem apresentar nenhum tipo de ícone ou mensagem de funcionamento ao usuário. Assim o usuário não pode finalizar o SPY007service facilmente, já os usuários mais avançados serão capazes de identificar os processos que estão sendo executados e então finalizar o SPY007service ou mesmo o agente NET-SNMP. É o agente Net-SNMP que irá “responder” as consultas SNMP originadas pelo gerenciador do SPY007.

O SPY007service grava no registro do Windows as informações referentes ao aplicativo que está sendo executado (título e classe da janela). E caso o usuário esteja fazendo uso de um navegador, a URL do *site* que está sendo acessada. Estas informações são gravadas em uma chave de registro própria para o SPY007:

HKEY_LOCAL_MACHINE/SOFTWARE/spy007

O agente Net-SNMP não é capaz de consultar o registro do Windows, mas permite executar uma rotina externa que faça essa consulta e repasse estes dados para o Net-SNMP. As rotinas externas implementadas para esta função são spy007GetUser.exe, spy007GetWindowTitle.exe, spy007GetWindowClass.exe e spy007GetWindowUrl.exe. Optou-se por armazenar os dados referentes ao aplicativo que está em primeiro plano no registro do Windows porque o registro é utilizado para armazenar informações referentes aos aplicativos instalados e em funcionamento. Também no registro são consultados os dados referentes ao usuário que está utilizando na estação assim todos os dados referentes ao monitoramento são consultados em um único local.

De forma sintética, o Collector dispara consultas SNMP para as estações monitoradas, estas consultas são recebidas pelo Net-SNMP que, de acordo com a consulta recebida, executa uma rotina externa que busca no registro do Windows a informação solicitada na consulta. Estes dados armazenados no registro são inseridos pelo serviço SPY007service que mantém atualizados no registro as informações quanto à aplicação que está sendo executada em primeiro plano na estação.

Figura 20 - Funcionamento do agente Windows

Quando o sistema operacional a ser monitorado for o GNU/Linux não é necessário utilizar o SPY007service, que é uma aplicação exclusivamente utilizada para o sistema operacional Windows. No entanto, o funcionamento é semelhante, é necessário instalar o agente Net-SNMP que irá receber e responder às consultas SNMP vindas do Collector. O Net-SNMP, novamente fará uso de rotinas externas para adquirir os dados referentes à aplicação em primeiro plano e o nome do usuário que está utilizando a estação. Para o sistema operacional GNU/Linux estas rotinas são escritas na linguagem Shell Script[11]. Esta linguagem é própria ao sistema GNU/Linux. O Windows e o GNU/Linux são sistemas totalmente distintos, com funcionamento de recursos diferenciados. Por este motivo, as rotinas externas ao Net-SNMP foram escritas em linguagens diferentes. Em nenhum dos SO citados o agente Net-SNMP é capaz de fornecer, de forma nativa, os dados necessários para o monitoramento que é objetivo do SPY007, por isso a necessidade de implementar rotinas específicas para estas tarefas.

6.1.2 Módulo de coleta de dados

O módulo de coleta de dados é composto por duas rotinas escritas com a linguagem PHP que consultam as estações monitoradas e gravam os dados obtidos no banco de dados MySQL. Essas rotinas são o Discovery e o Collector.

Quando uma ferramenta de gerência de rede, como é o caso do SPY007, é projetada, um fator de relevância que deve ser analisado cuidadosamente, é o tempo de coleta de dados. O SPY007 busca, nas estações monitoradas, dados referentes ao usuário que está utilizando a máquina, informações que possibilitem identificar qual o aplicativo está sendo executado em primeiro plano na estação e ainda se estiver sendo utilizado um navegador de Internet, o sistema identifica qual o endereço do *site* que está sendo acessado. São efetivamente 5 consultas por estação. Caso o número de estações a serem monitoradas seja muito grande, este tempo de coleta de dados em todas as estações pode ser alto, em vista disso o processo de coleta de dados foi divido em dois estágios. O primeiro estágio visa identificar as estações que estão ativas de acordo com as redes cadastradas. A segunda etapa deve executar a coleta de

[11] Shell é o prompt de linha de comando do Unix e Linux. Um Shell Script é arquivo que guarda um conjunto de instruções Shell que pode ser executado se preciso.

dados propriamente dita. Na coleta de dados são consultadas apenas as estações que foram previamente identificadas como ativas.

O Discovery é a rotina que consulta todos os endereços IP de todas as redes cadastradas no SPY007 visando identificar quais estações estão efetivamente ativas. Uma estação está ativa, se responder a uma consulta SNMP que solicita o identificador da estação.

O Collector faz a coleta de dados em cada uma das estações identificadas como ativas pelo Discovery. Constam abaixo os dados consultados em cada estação são:

a) identificador do host (é definido no momento do cadastro da estação junto ao SPY007).
b) nome do usuário que está utilizando a estação.
c) título da janela do aplicativo em primeiro plano na estação.
d) classe da janela do aplicativo em primeiro plano na estação.
e) URL que está sendo acessada.

Com os dados listados acima, o Collector faz a identificação do aplicativo comparando o título e a classe da janela coletados com os dados de aplicativos cadastrados previamente no SPY007. Também executa a identificação da URL, sendo apenas o domínio da URL é extraído e comparado com os demais domínios já cadastrados. Caso ainda não esteja cadastrado o Collector cadastra automaticamente todos os domínios que foram acessados e ainda são desconhecidos ao sistema. Ao identificar o usuário, caso este ainda não estija cadastrado no sistema, o Collector insere-o na base de dados automaticamente.

Tanto o Collector quanto o Discovery têm sua execução agendada no gerenciador de tarefas do sistema operacional onde o SPY007 está instalado, possibilitando assim ao administrador da rede determinar o intervalo de tempo para a execução destes. Esta coleta de dados na rede que é executada com um intervalo fixo de tempo, é denominada de amostragem aleatória sistemática.

Na Figura 21 é representado o funcionamento geral do Collector do SPY007.

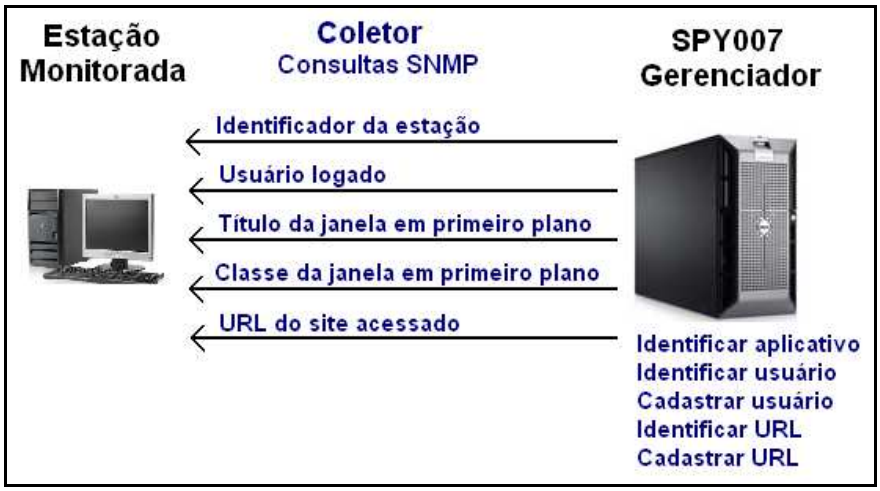

Figura 21 - Funcionamento do Collector

O Collector executa as consultas SNMP nas estações ativas da rede, no entanto se uma estação não responder à 3 consultas, esta estação é definida como inativa e será ativada novamente quando responder a uma consulta realizada pelo Discovery.

6.1.3 Módulo de gerenciamento

No módulo de gerenciamento além das funções de administração do sistema permite também visualizar, na forma de gráficos e em listas detalhadas, os dados coletados pelo sistema. Algumas telas da interface de gerenciamento serão detalhadas abaixo.

O módulo de gerenciamento possui acesso restrito. Ver na Figura 22 a tela de identificação de acesso do SPY007.

Figura 22 - SPY007 tela de identificação de acesso

Quando o usuário acessa o sistema, a primeira tela mostra um resumo com o número total de estações cadastradas e, destas quantas estão ativas e inativas no momento. O total de usuários cadastrados, número de aplicativos cadastrados, total de categorias de aplicativos cadastrados total de URL's cadastradas. Outra informação relevante que pode ser visualizada é o número de coletas efetuadas até o momento.

Na Figura 23 é apresentada a tela exibida após o acesso ser efetuado no SPY007.

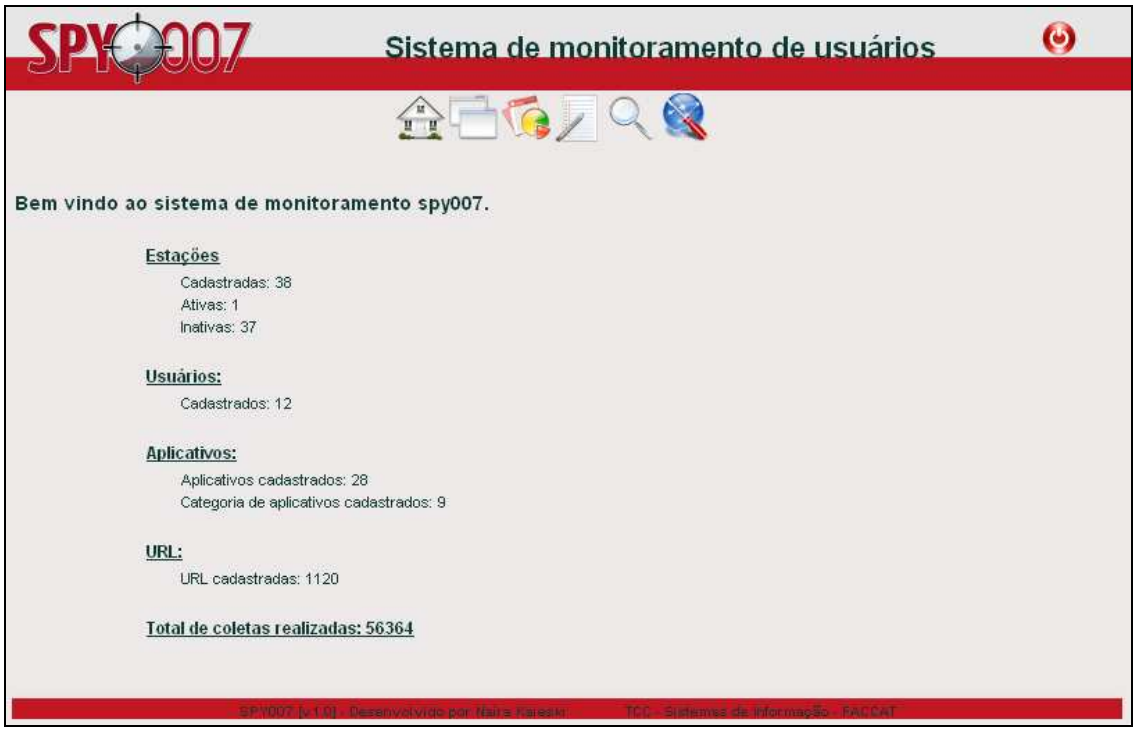

Figura 23 - SPY007 home

O SPY007 possui um menu de opções principal que direciona o usuário às funcionalidades disponíveis no sistema. Entre as funcionalidades, o administrador da rede pode gerenciar aplicativos, categorias de aplicativos e a relação entre os aplicativos com as categorias. Uma categoria de aplicativo pode ter vários aplicativos cadastrados, como por exemplo, a categoria, navegador, possui os aplicativos Mozilla Firefox e Internet Explorer. No entanto um aplicativo pode estar relacionado com apenas uma categoria de aplicativo ou nenhuma. O processo de gerenciamento de aplicativos e categorias inclui as funções de cadastro, edição e exclusão.

A Figura 24 apresenta a tela de gerenciamento de aplicativos do SPY007.

Figura 24 - SPY007 gerenciamento de aplicativos

A informação foco do SPY007, é o gráfico de utilização de aplicativos e acesso a Internet nas estações da rede. Estes gráficos são exibidos na interface administrativa. Os gráficos de produtividade dos *softwares* estão divididos em: categoria de aplicativos e aplicativos.

Na Figura 25 é apresentado um exemplo de gráfico em forma de pizza de utilização de *software*, este separado por categoria de aplicativo.

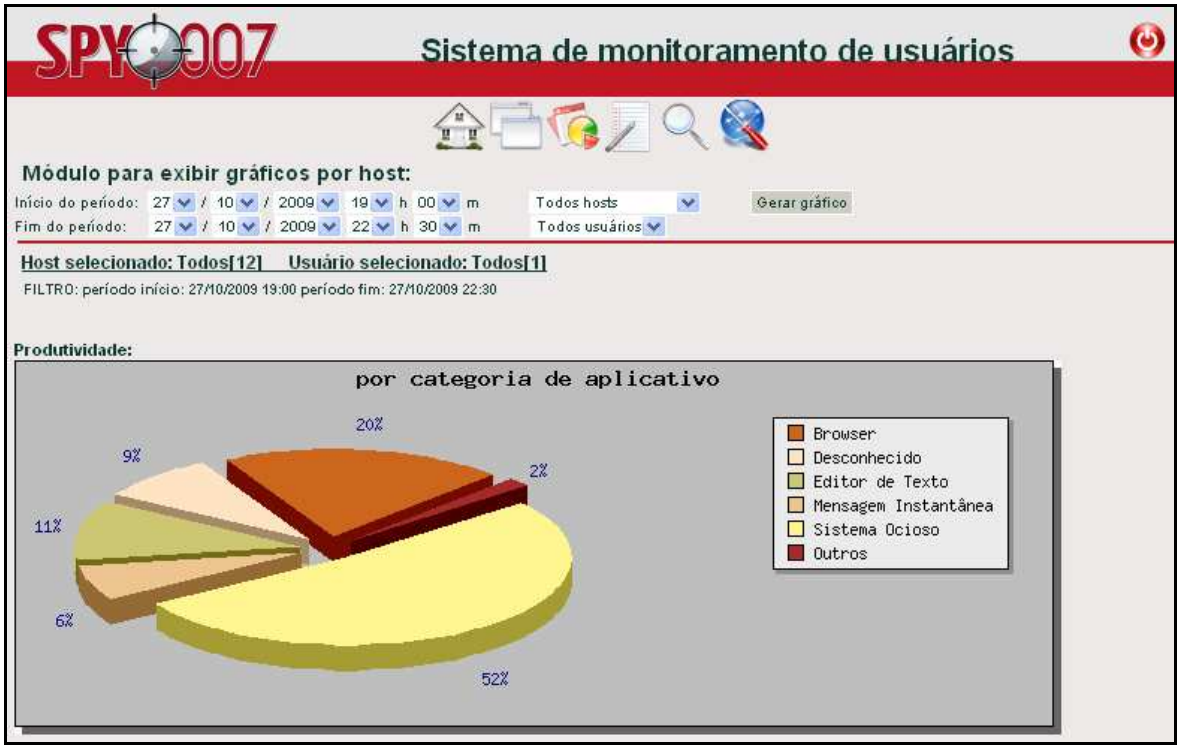

Figura 25 - SPY007 gráfico por categoria de aplicativo utilizado

O administrador deve selecionar o período (de início e fim) para gerar os gráficos. É possível também, adicionar filtros mais específicos, combinando usuário e/ou estação (*host*). Outro gráfico disponível pode ser visualizado na Figura 26, o gráfico de utilização de *softwares* separado por aplicativo.

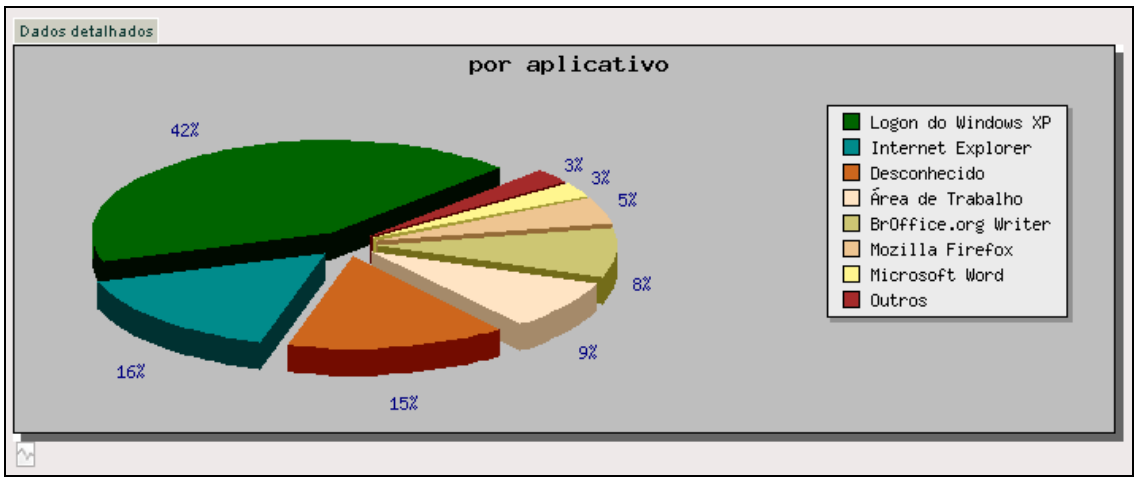

Figura 26 - SPY007 gráfico por aplicativo utilizado

Todos os gráficos em forma de pizza apresentados no sistema, exibem os dados referentes aos 10 itens mais utilizados ou mais acessados, desde que estes representem percentuais superiores a 2%. Este filtro foi realizado para que os gráficos fossem facilmente legíveis, já que, muitos aplicativos são utilizados e vários possuem um percentual utilização baixo isso dividiria muito o gráfico. Todos os percentuais que não estiverem de acordo com as regras citadas acima serão classificados como “Outros” no gráfico.

Na Figura 27 é possível visualizar os gráficos de utilização de aplicativos de forma detalhada. É exibida uma lista com todos os aplicativos acessados e o respectivo percentual de utilização.

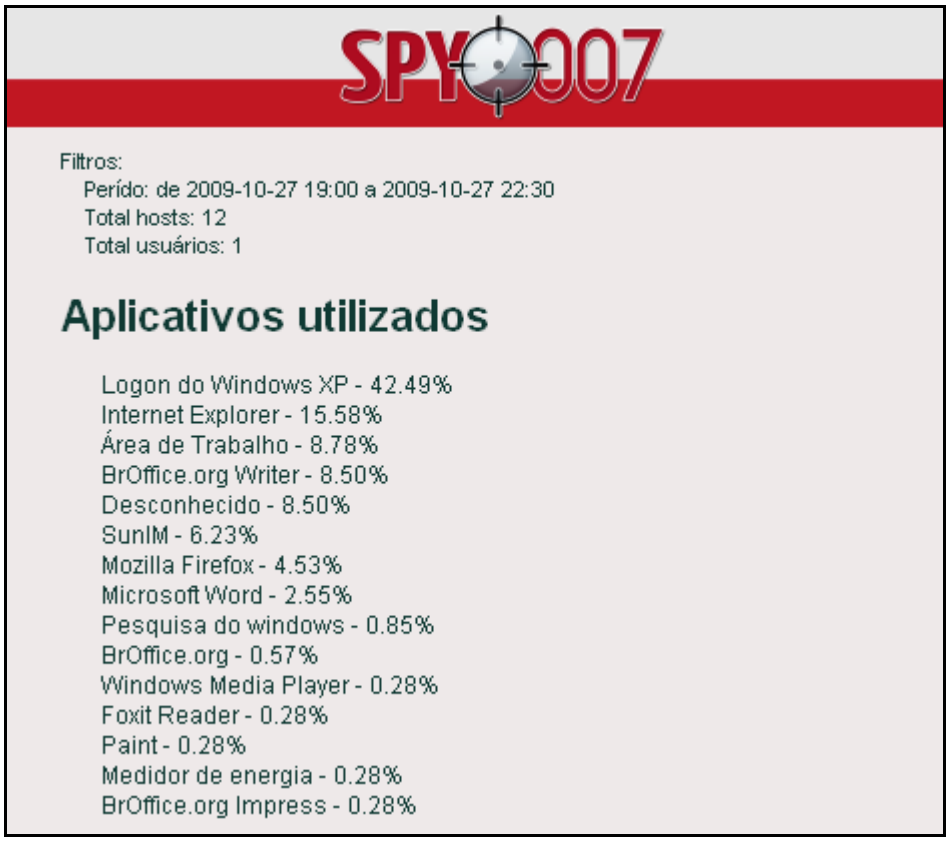

Figura 27 - SPY007 aplicativos utilizados detalhado

Em outro gráfico do SPY007 pode-se analisar a navegação na Internet realizada dentro da rede, pelos usuários. Estes dados podem ser visualizados tanto através de gráficos, quanto de forma detalhadas através de uma lista.

Um exemplo de gráfico de navegação na Internet pode ser apreciado na Figura 28.

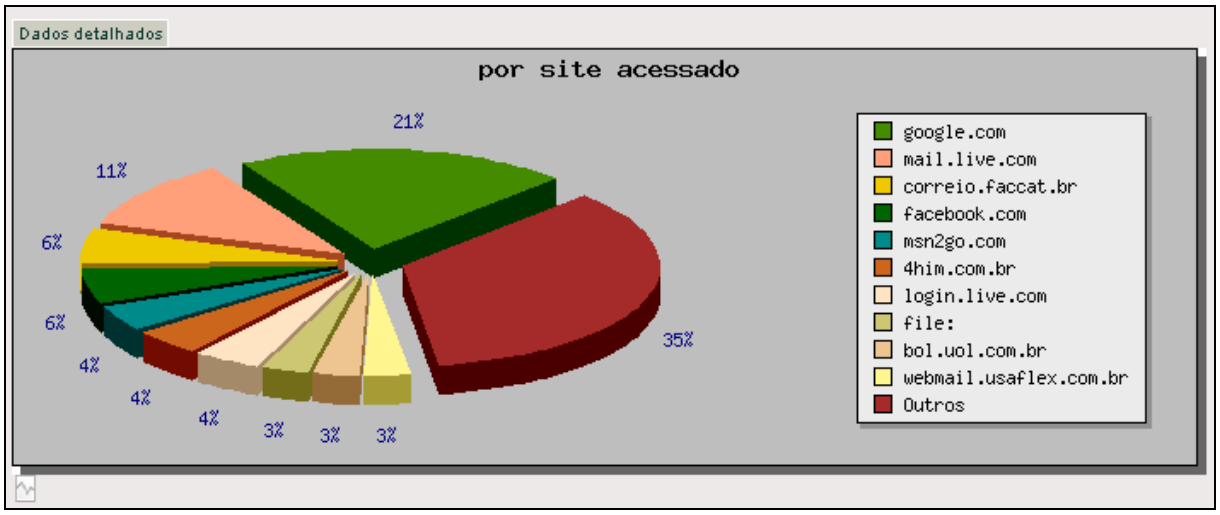

Figura 28 - SPY007 gráfico de navegação

O sistema disponibiliza gráficos em forma de pizza para visualizar a utilização de aplicativos e acesso a *sites* e dados detalhados dos gráficos conforme demonstrado acima. Porém, SPY007 também disponibiliza ao administrador, gráficos em linha que demonstram a utilização de aplicativos e navegação na Internet (URL acessadas). Nestes gráficos o administrador pode exibir os dados por período de um dia, um mês ou um ano e pode selecionar um usuário e/ou uma estação. Para facilitar a compreensão do gráfico são exibidos apenas os 5 itens com percentuais mais significativos.

Na Figura 29 é apresentado um gráfico em linha de acessos a Internet em determinado mês.

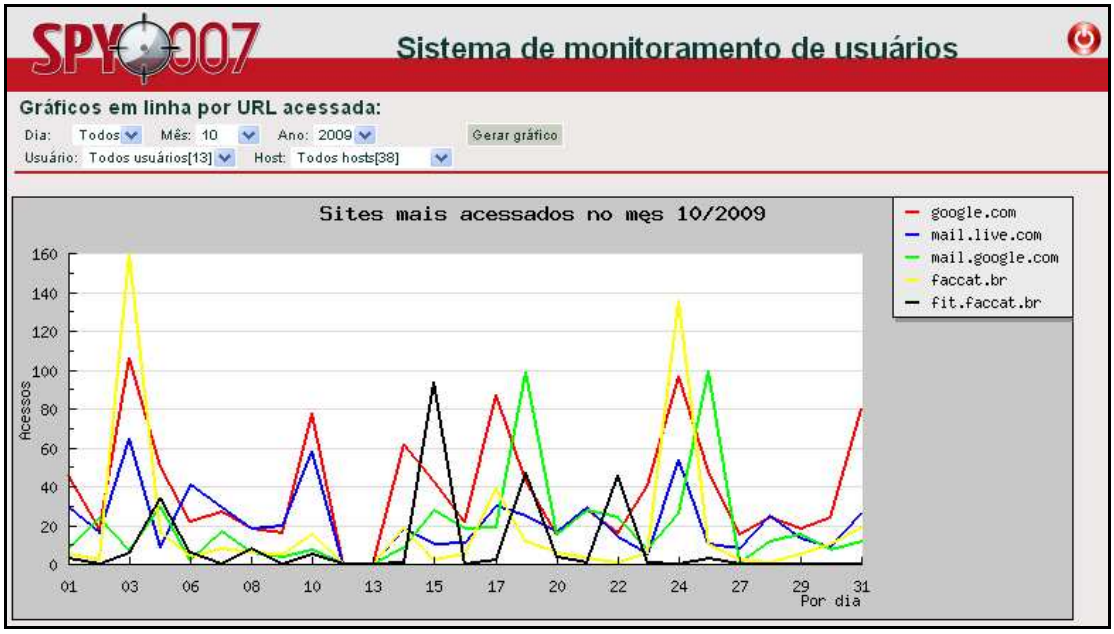

Figura 29 - SPY007 gráfico em linha

Na interface administrativa do SPY007 o administrador de rede pode visualizar os *logs* de funcionamento do sistema. Os *logs* do Collector estão separados dos demais por conterem os dados mais relevantes do sistema, que são as coletas de dados realizadas nas estações monitoradas. O administrador pode clicar sobre um *log* do Collector e detalhar a coleta de dados selecionada.

Na Figura 30 é apresentado um exemplo da tela onde é possível visualizar os *logs* do Collector.

Figura 30 - SPY007 *logs* Collector

No menu de configuração o administrador pode gerenciar as redes (cadastrar, excluir), trocar a senha de acesso ao sistema, gerenciar as URL (cadastrar, editar, excluir) e executar o Discovery da rede. Nesta opção de Discovery o administrador deve selecionar uma rede e o sistema irá verificar se os IP da rede selecionada respondem a consulta SNMP e a um pacote ICMP (conhecido ping).

Na Figura 31 é possível visualizar as opções de configuração disponíveis no menu “Configuração” do SPY007.

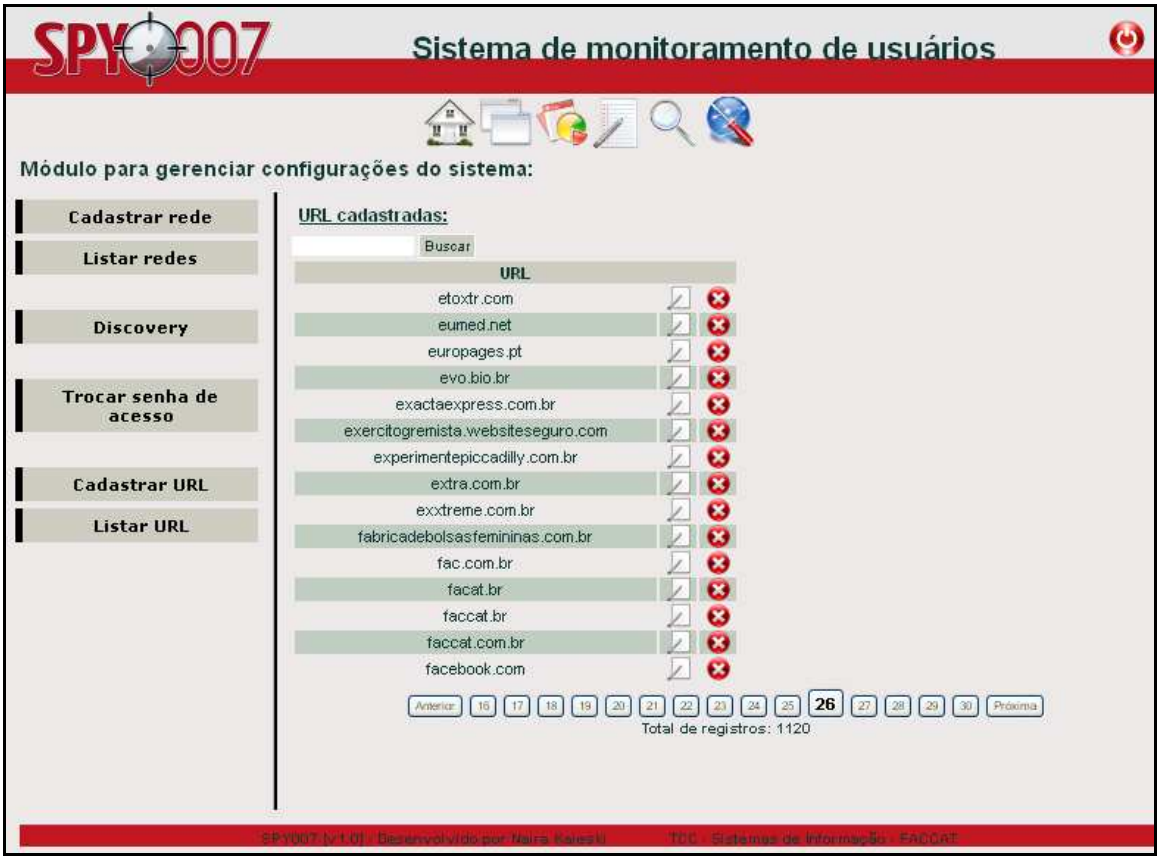

Figura 31 - SPY007 menu configuração

6.2 Implantação do SPY007

O SPY007 é um *software open source* e assim não é necessário adquirir uma licença de uso para a sua implantação em uma organização. Este aplicativo se enquadra na categoria de gerência de rede e tem por objetivo o monitoramento da utilização dos aplicativos e a navegação na Internet nas estações de trabalho da rede. Devido a este tipo de monitoramento é preciso que a organização defina as políticas de acesso e elabore um documento com a política de uso aceitável (AUP – *Acceptable Use Policy*) que deverá ser amplamente divulgada e de conhecimento de todos os usuários.

De acordo com a cartilha de segurança para Internet divulgada pelo CGI.BR (2006) a política de segurança atribui direitos e responsabilidades às pessoas que lidam com os

recursos computacionais da instituição e com as informações neles armazenados. Ela também define as atribuições de cada um em relação à segurança dos recursos com os quais trabalham. Uma política de segurança deve prever o que pode ser feito na rede da instituição e que será considerado inaceitável. Tudo que descumprir a política de segurança pode ser considerado um incidente de segurança. Na política de segurança devem ser definidas as penalidades as quais estão sujeitos aqueles que não cumprirem a política.

A política de uso aceitável é um documento que define como os recursos computacionais de uma empresa devem ser utilizados. Também é nela que é definido os direitos e responsabilidades de cada um dos usuários.

É importante que a empresa divulgue amplamente e esclareça as dúvidas dos seus colaboradores em relação as políticas de segurança. Todos os usuários devem estar cientes de que suas atividades estão sendo monitoradas pela organização.

7 ESTUDO APLICADO

Com o objetivo de proporcionar um teste em ambiente real da aplicação desenvolvida, foi efetuado um estudo aplicado do SPY007. Neste estudo o sistema foi implantando em dois laboratórios de informática das Faculdades Integradas de Taquara.

7.1 Cenário

As Faculdades Integradas de Taquara (FACCAT) estão localizadas na cidade de Taquara, estado do Rio Grande do Sul. A FACCAT possui mais de 15 cursos de graduação e pós-graduação e aproximadamente 4000 alunos regulares.

A FACCAT possui sete laboratórios de informática, sendo que o SPY007 foi implantado em dois destes. Os laboratórios selecionados para testes do SPY007 foram os que são mais utilizados. As aulas da faculdade ocorrem predominantemente no turno da noite, porém nos laboratórios também ocorrem cursos e são abertos a comunidade em geral durante todo o período do dia. Ambos laboratórios monitorados utilizam o sistema operacional Windows, sendo que um deles possui 22 computadores e o outro com 16.

7.2 Procedimento metodológico

O período de análise do monitoramento é de seis dias, de segunda a sábado, que correspondem aos dias letivos da instituição. Inicialmente o monitoramento dos laboratórios foi efetuado sem que os usuários soubessem e em um segundo momento, os usuários foram notificados quanto ao monitoramento através de cartazes colocados dentro e fora dos laboratórios.

Conforme mencionado anteriormente, as aulas da faculdade ocorrem em sua grande maioria no período da noite. Durante o dia os laboratórios são pouco utilizados, mas a grande maioria das máquinas fica ligada, já que o espaço é aberto à comunidade. Devido a esta característica, o monitoramento indica que os computadores ficam ociosos parte do dia.

Assim os gráficos referentes ao monitoramento dos laboratórios foram configurados para não contabilizar e exibir as telas de *logon* do Windows, e também dos aplicativos que o SPY007 classificou como desconhecido. O período diário de monitoramento nos gráficos apresentados nas sub-seções seguintes é das 8h às 23h.

Os gráficos apresentam o monitoramento separado por aplicativo e *site* acessado. São exibidos gráficos com o resumo das atividades nas semanas monitoradas e gráficos com as atividades por dia. O Collector foi configurado para realizar as coletas de dados a cada cinco minutos, já o Discovery foi executado a cada quinze minutos.

Em todas as estações são configuradas regras, junto ao sistema operacional, que não permitem ao usuário instalar novos aplicativos nas estações. O acesso à Internet passa por um servidor *proxy* que filtra os acessos, e de acordo com as regras estabelecidas permite ou não o acesso a determinado site.

7.2.1 Sistema atual

No sistema atual de funcionamento dos laboratórios, não há uma ferramenta de monitoramento que gere estatísticas, ou gráficos, de utilização dos *softwares* utilizados pelos usuários e tampouco do acesso à Internet. Os laboratórios são interconectados através de *switch*, um em cada laboratório. Estes equipamentos são conectados ao *switch* de *core* da instituição. Estes dispositivos estabelecem a comunicação entre os laboratórios e a sala de servidores da faculdade, ver Figura 32.

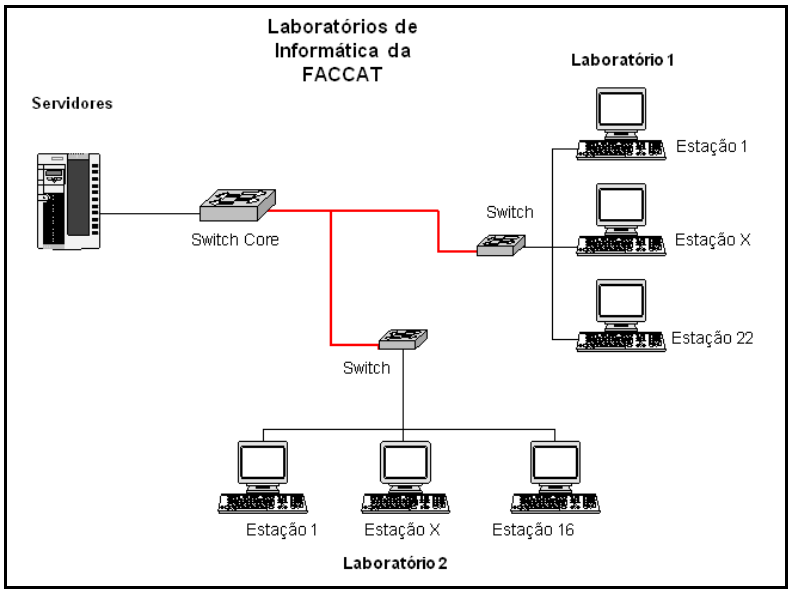

Figura 32 - Rede dos laboratórios

7.2.2 Sistema proposto

No sistema proposto, foi instalado o agente dos SPY007 em todas as estações dos laboratórios selecionados e em um servidor localizado na sala de máquinas da faculdade, foi instalado o gerenciador do SPY007, conforme pode ser visto na Figura 33. O gerenciador do SPY007 efetua as coletas de dados nas estações a cada cinco minutos.

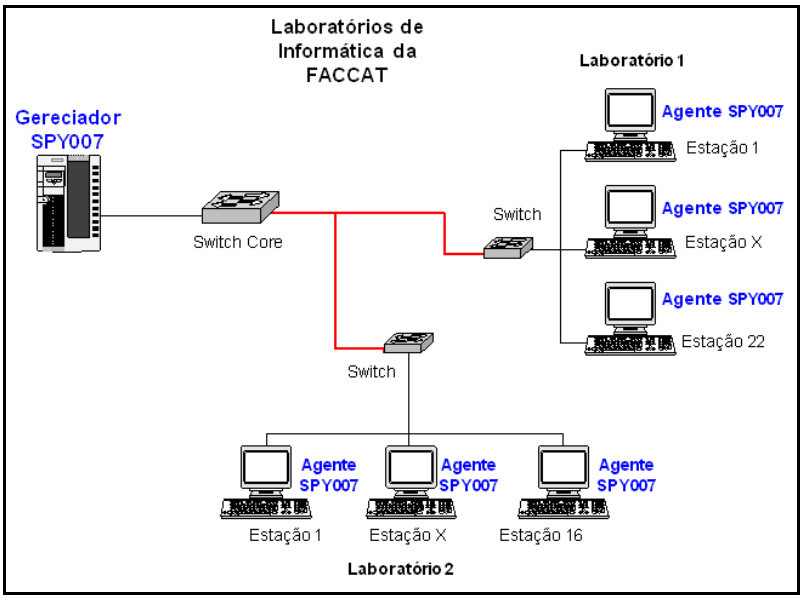

Figura 33 - Rede dos laboratórios monitorados

7.3 Resultados obtidos no primeiro período de monitoramento

Os gráficos abaixo expostos foram obtidos no período de 26 a 31 de outubro de 2009. Neste período os usuários não haviam sido informados sobre o monitoramento que estava sendo realizado.

7.3.1 Resumo da utilização de aplicativos na semana de 26/10 a 31/10

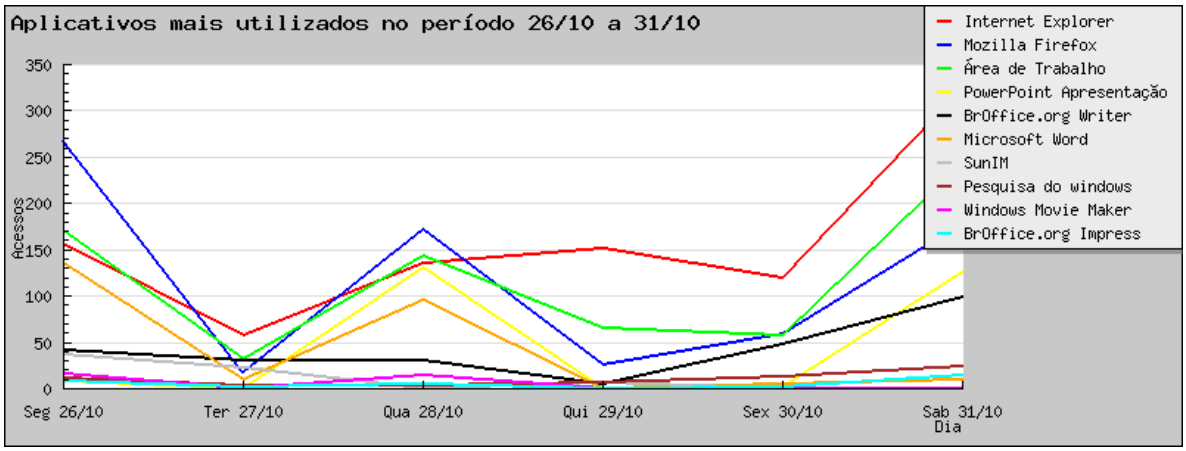

Figura 34 - Aplicativos mais utilizados na semana

7.3.2 Aplicativos mais utilizados no período de 26/10 a 31/10 por dia

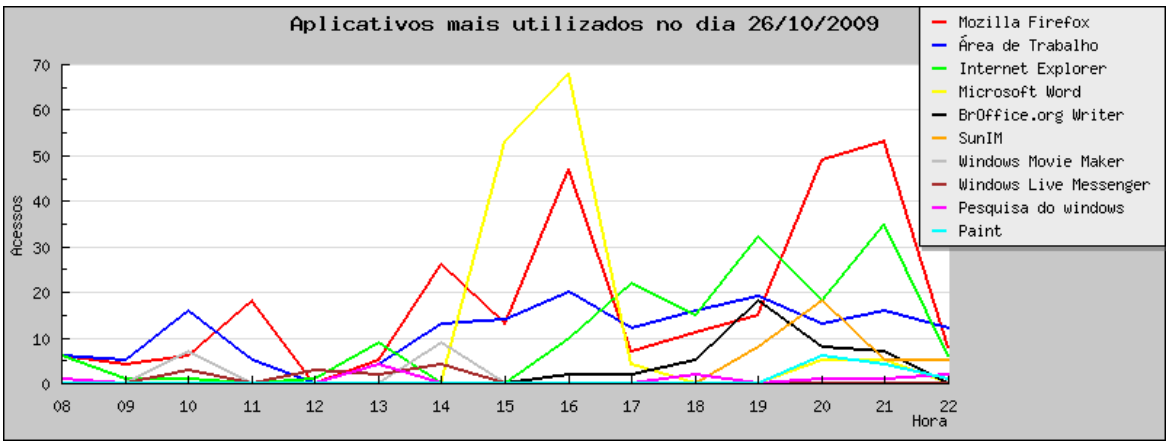

Figura 35 - Aplicativos mais utilizados no dia 26/10/2009

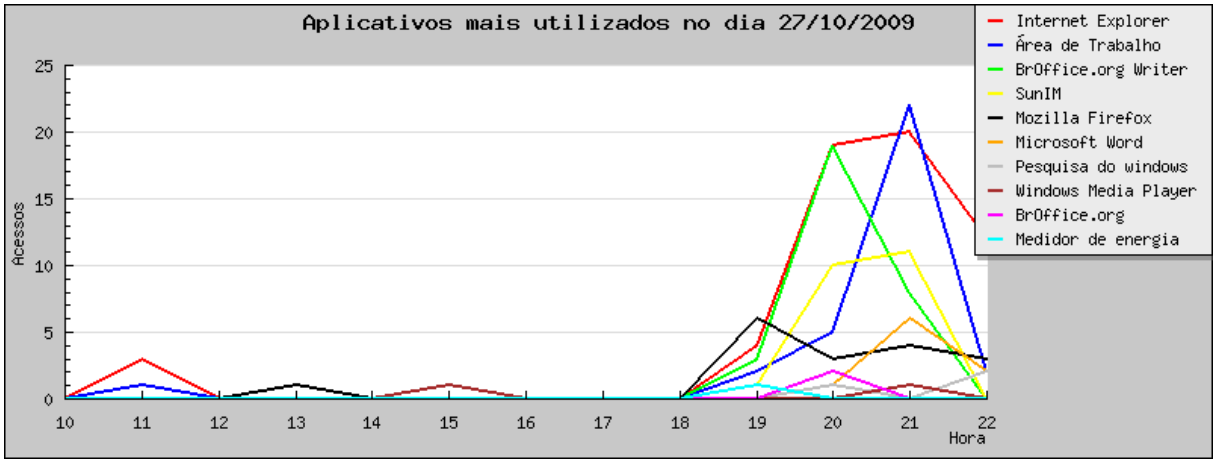

Figura 36 - Aplicativos mais utilizados no dia 27/10/2009

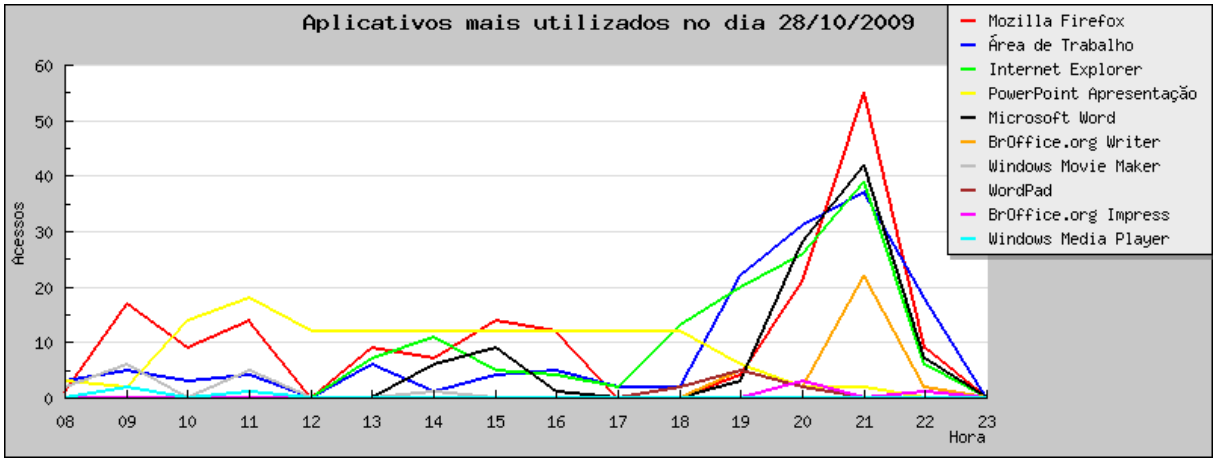

Figura 37 - Aplicativos mais utilizados no dia 28/10/2009

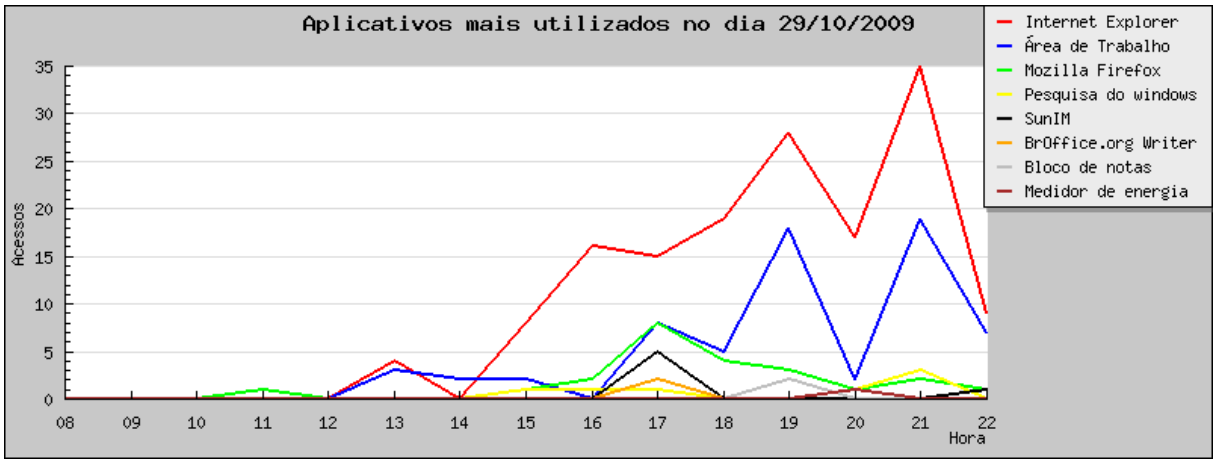

Figura 38 - Aplicativos mais utilizados no dia 29/10/2009

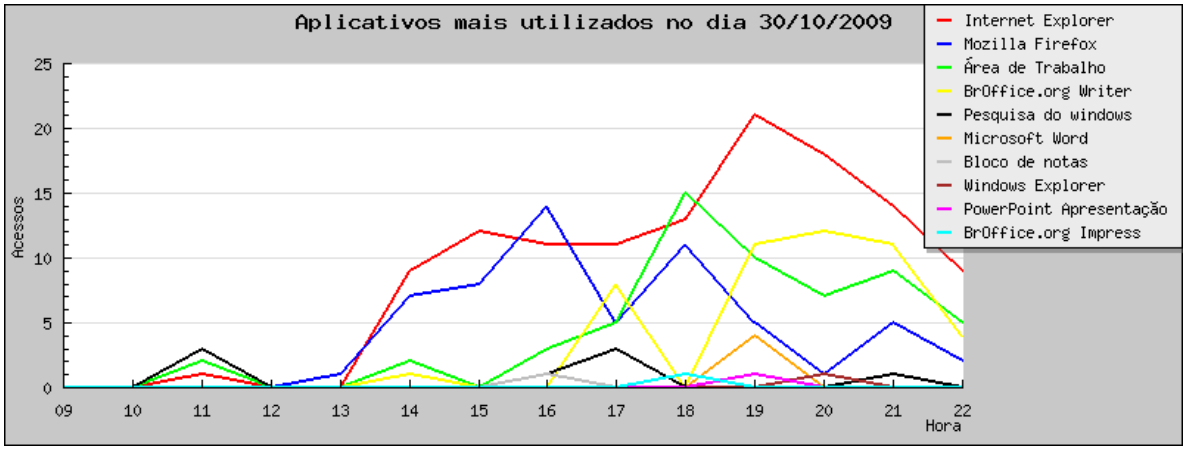

Figura 39 - Aplicativos mais utilizados no dia 30/10/2009

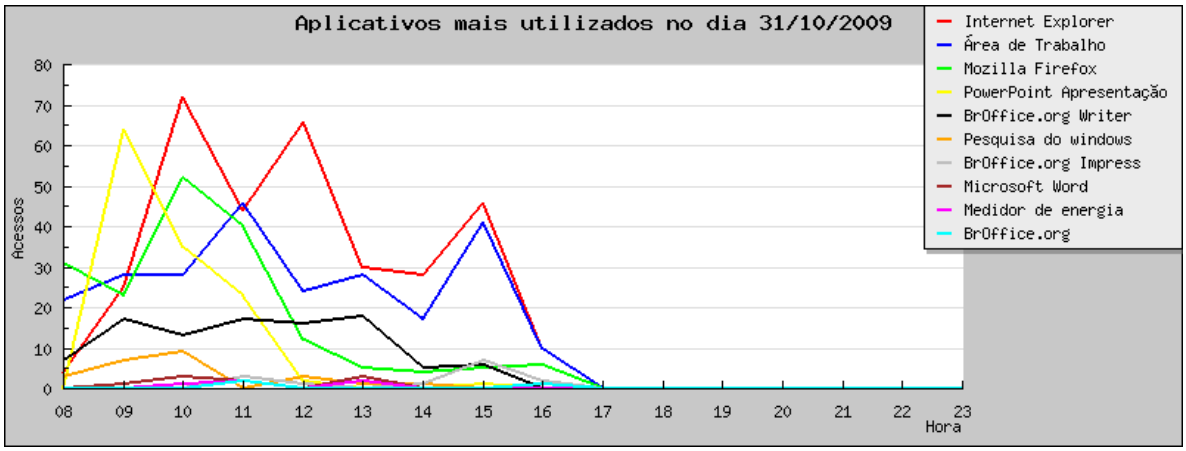

Figura 40 - Aplicativos mais utilizados no dia 31/10/2009

7.3.3 Resumo do acesso a Internet na semana de 26/10 a 31/10

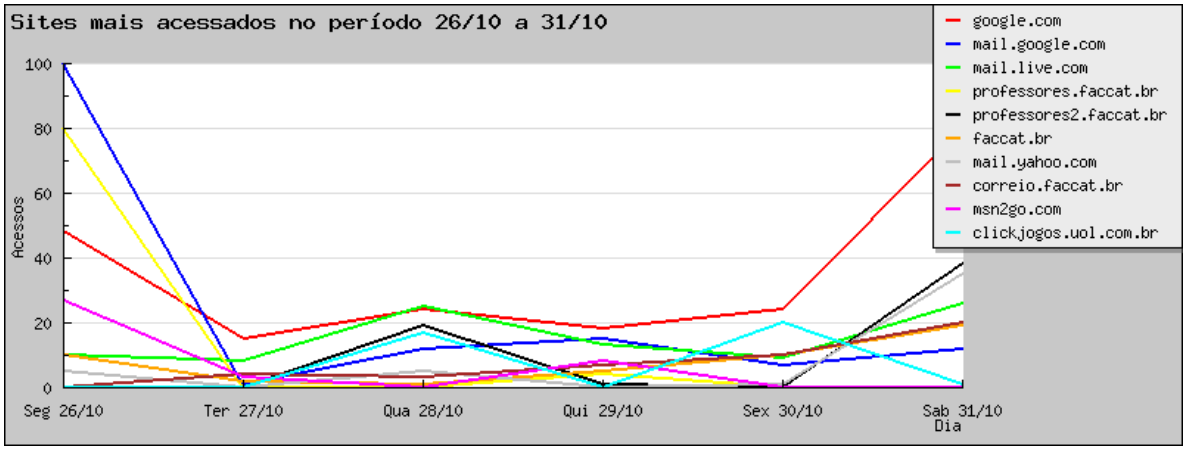

Figura 41 - *Sites* mais acessados na semana

7.3.4 *Sites* mais acessados no período de 26/10 a 31/10 por dia

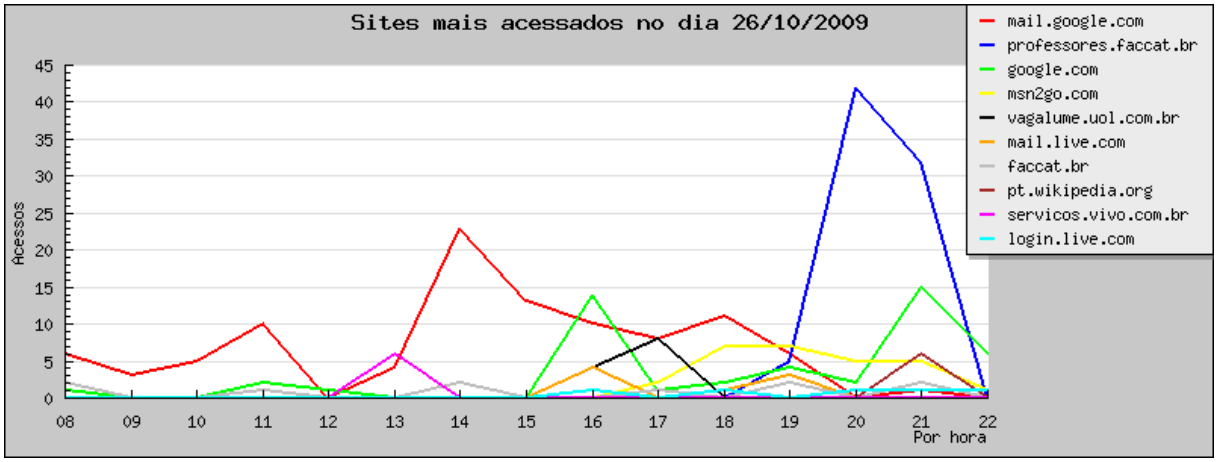

Figura 42 - *Sites* mais acessados no dia 26/10/2009

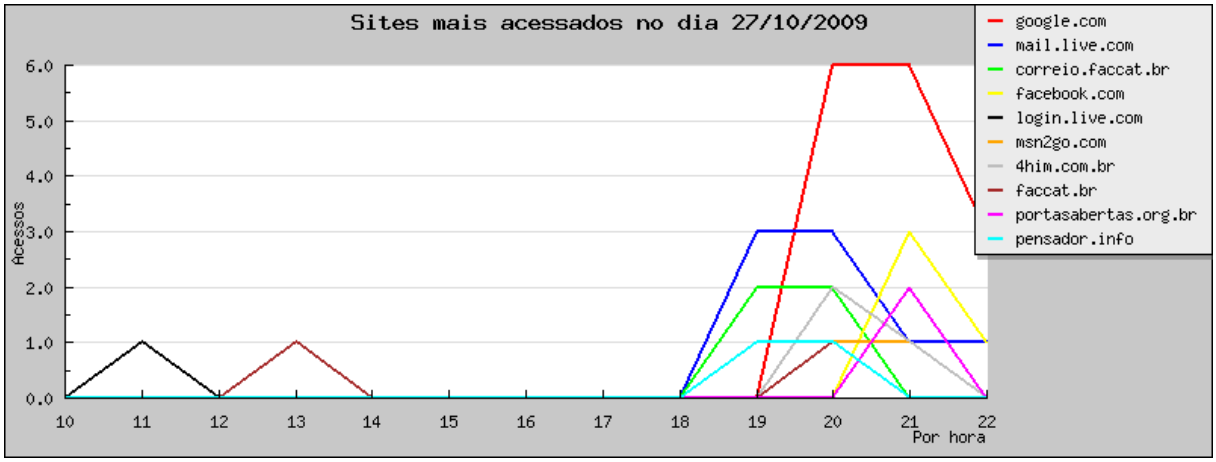

Figura 43 - *Sites* mais acessados no dia 27/10/2009

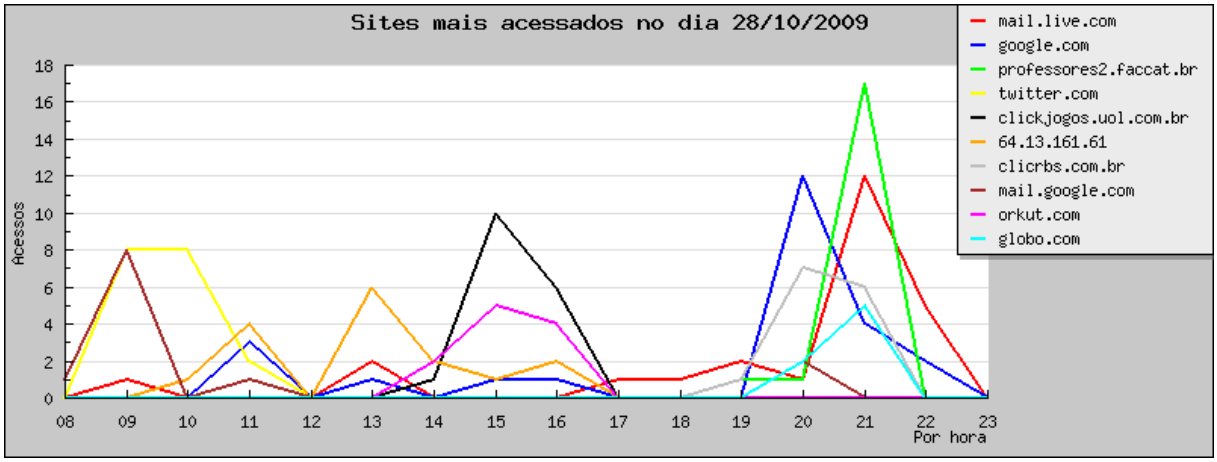

Figura 44 - *Sites* mais acessados no dia 28/10/2009

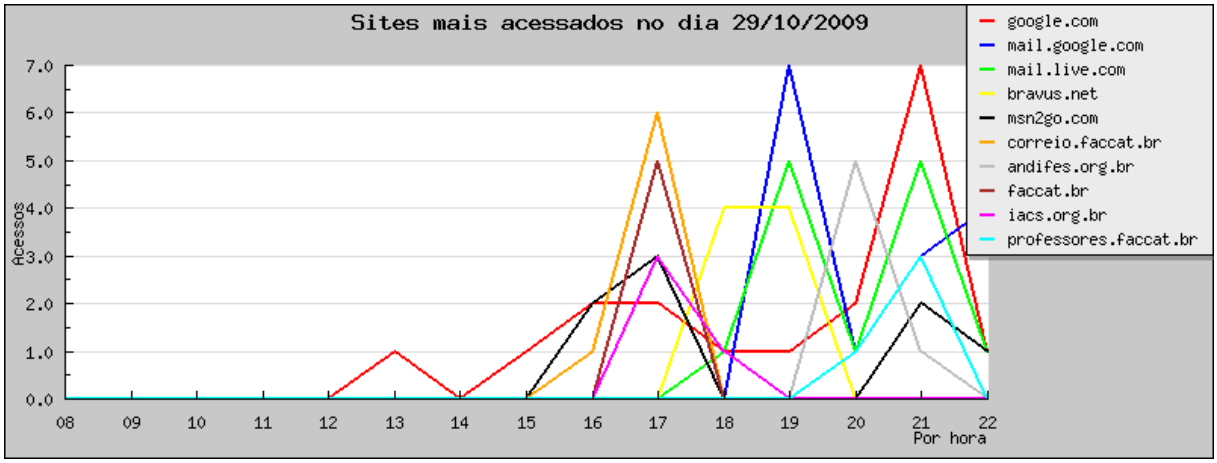

Figura 45 - *Sites* mais acessados no dia 29/10/2009

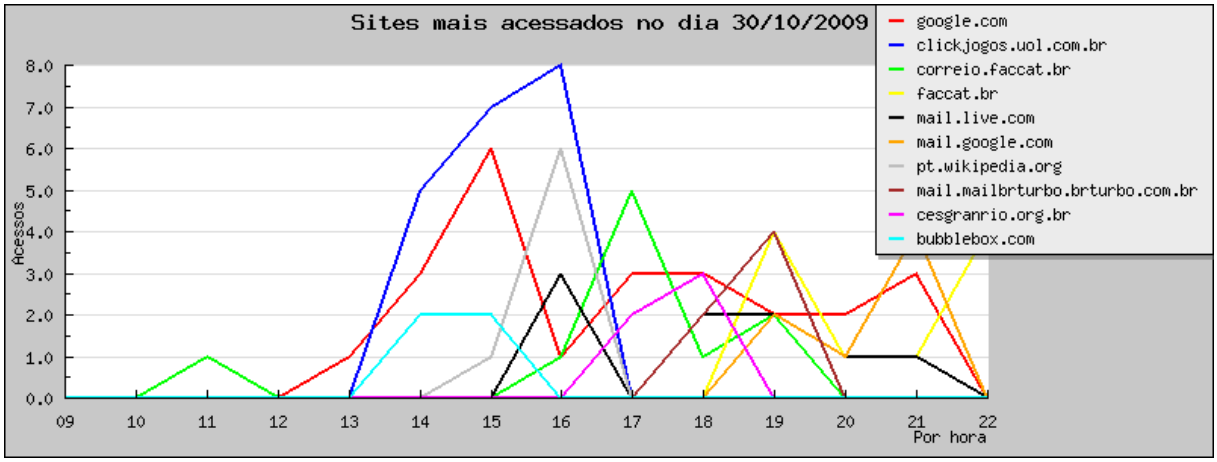

Figura 46 - *Sites* mais acessados no dia 30/10/2009

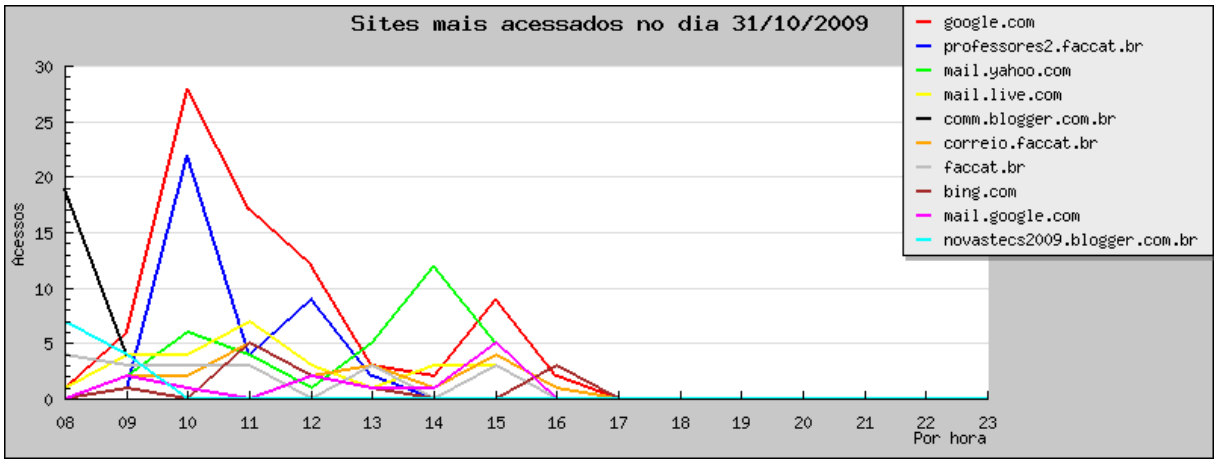

Figura 47 - *Sites* mais acessados no dia 31/10/2009

7.4 Resultados obtidos no segundo período de monitoramento

Os gráficos abaixo foram obtidos no período de 03 a 07 e 09 de novembro de 2009. Neste período os usuários foram informados sobre o monitoramento que estava sendo executado nos laboratórios.

7.4.1 Resumo da utilização de aplicativos na semana de 03/11 a 09/11

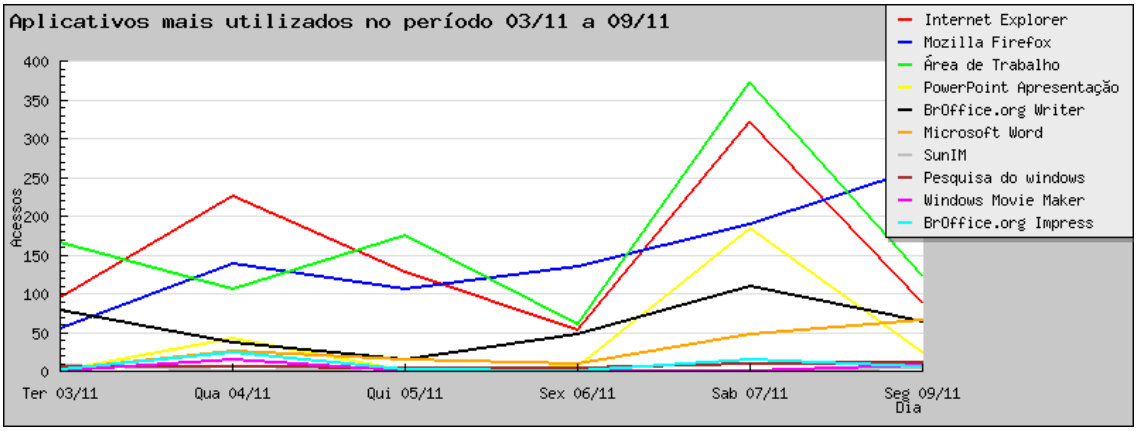

Figura 48 - Aplicativos mais utilizados na semana

7.4.2 Aplicativos mais utilizados no período de 03/11 a 09/11

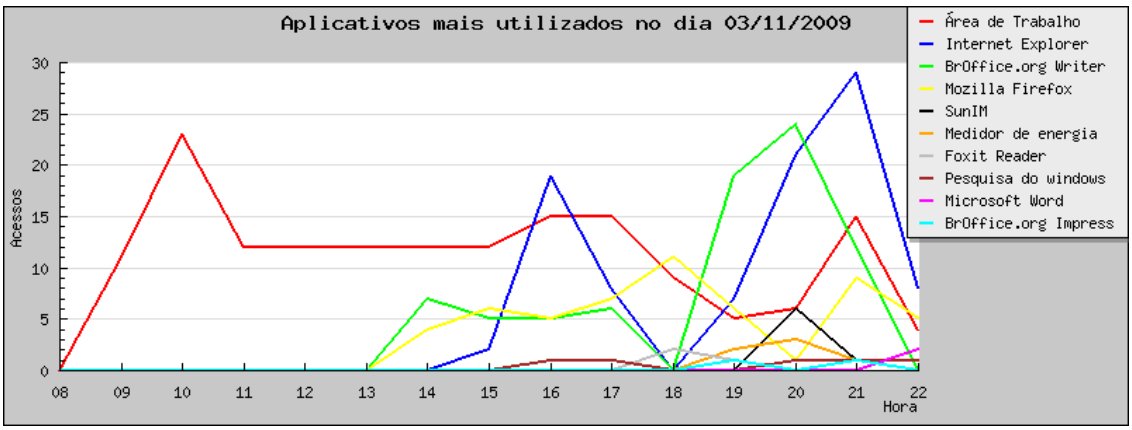

Figura 49 - Aplicativos mais utilizados no dia 03/11/2009

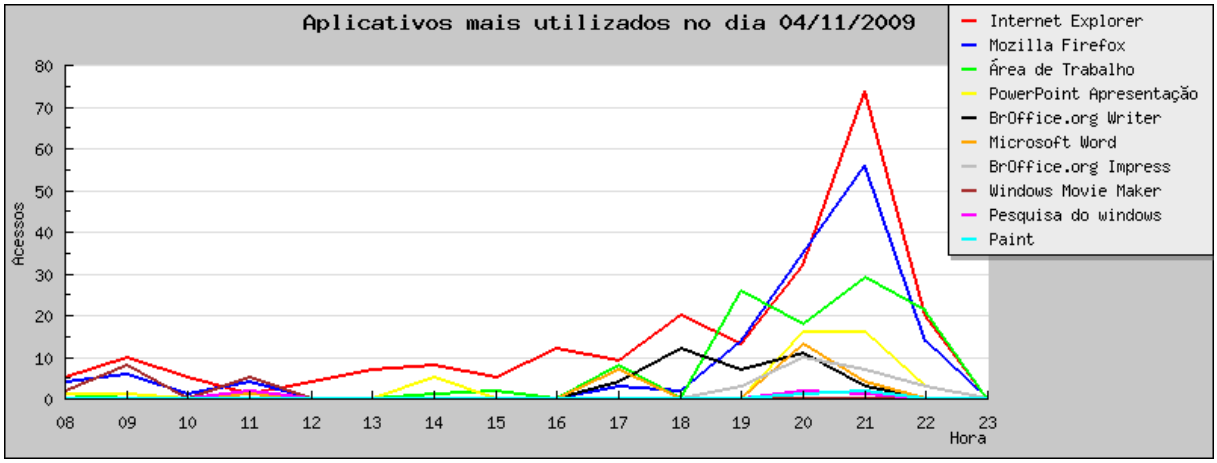

Figura 50 - Aplicativos mais utilizados no dia 04/11/2009

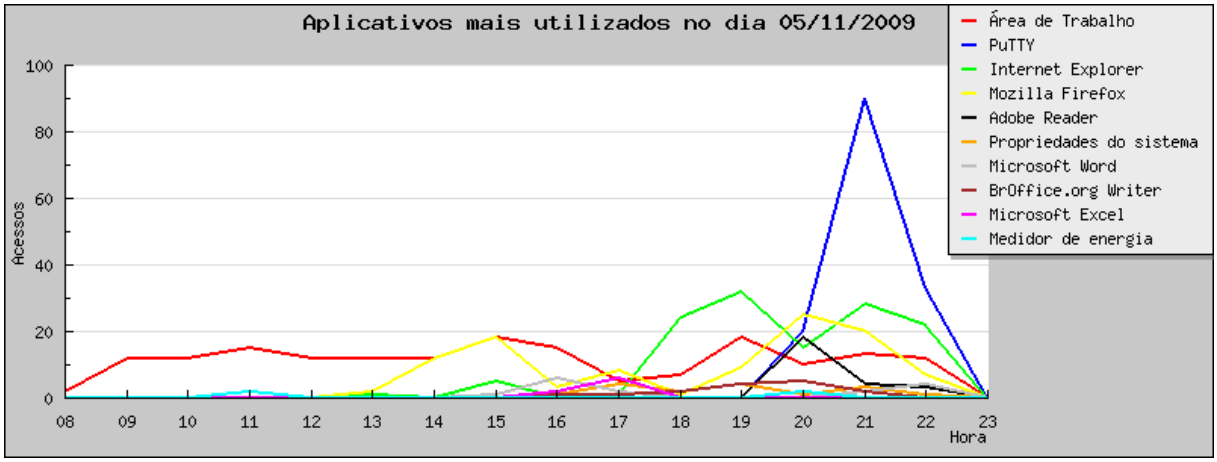

Figura 51 - Aplicativos mais utilizados no dia 05/11/2009

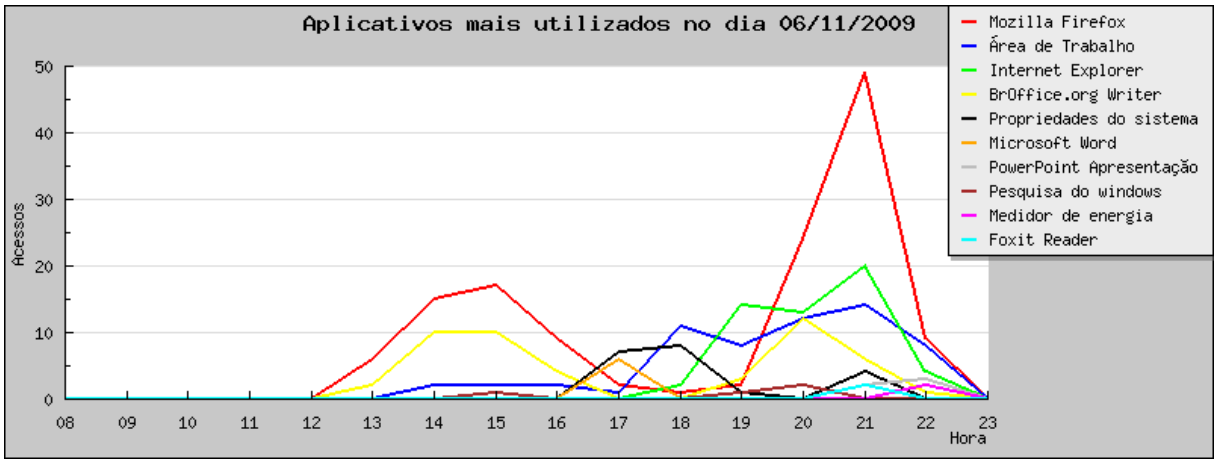

Figura 52 - Aplicativos mais utilizados no dia 06/11/2009

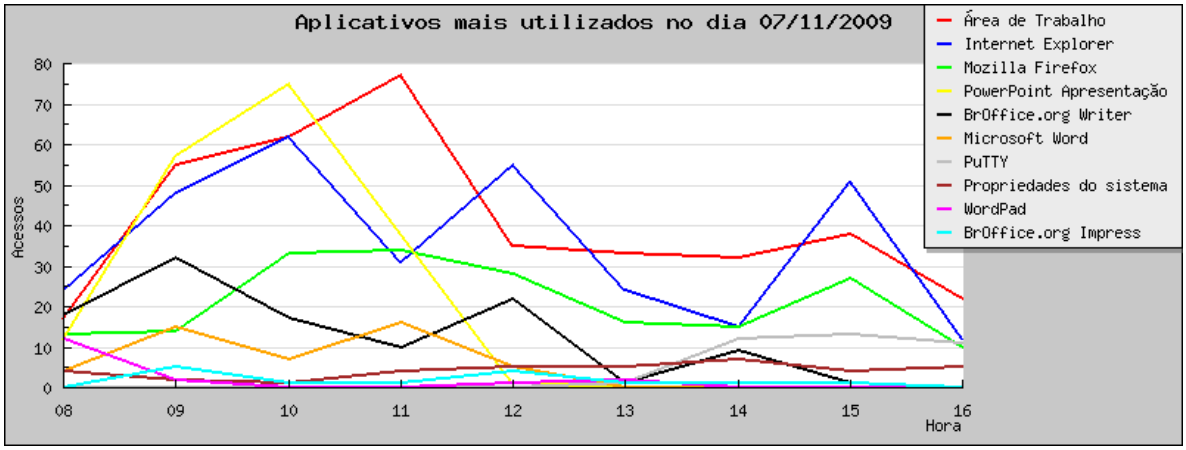

Figura 53 - Aplicativos mais utilizados no dia 07/11/2009

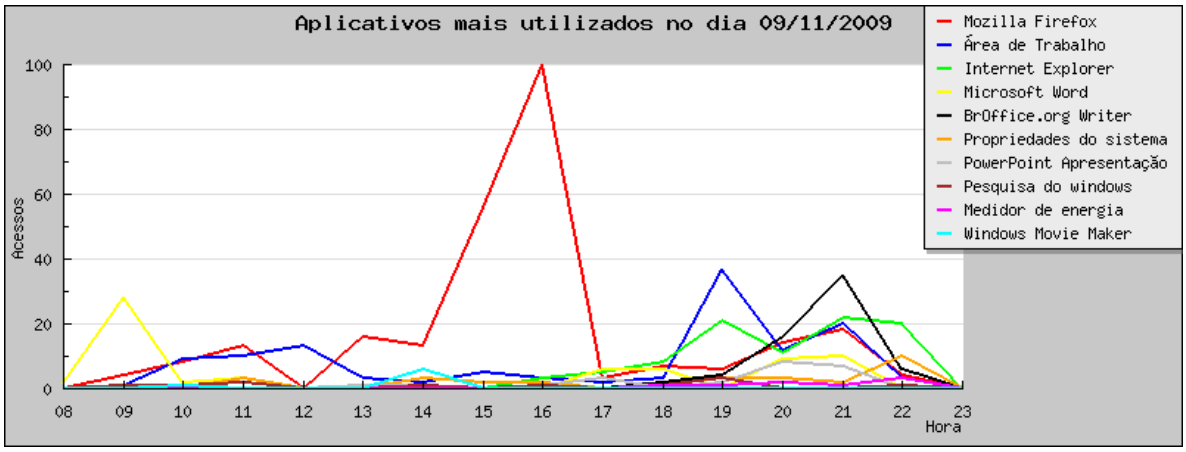

Figura 54 - Aplicativos mais utilizados no dia 09/11/2009

7.4.3 Resumo do acesso a Internet na semana de 03/11 a 09/11

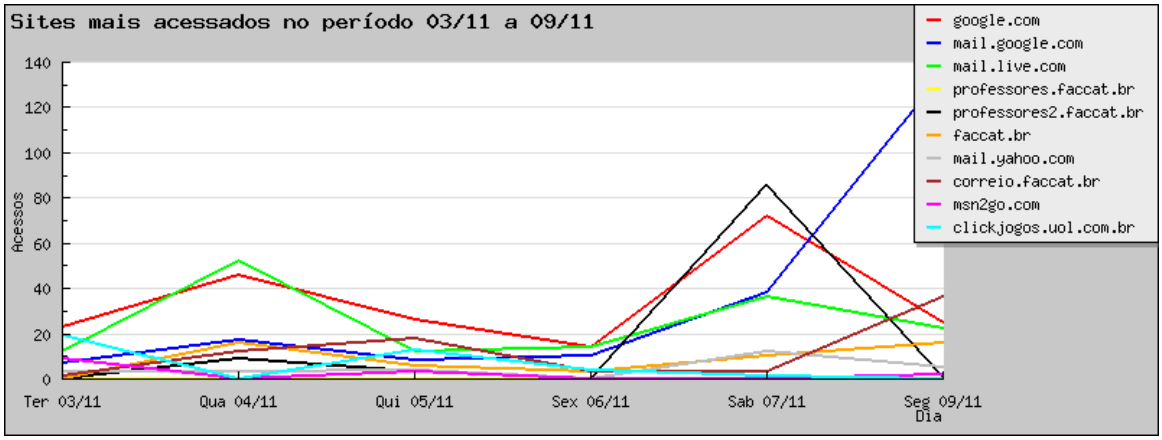

Figura 55 - *Sites* mais acessados na semana

7.4.4 *Sites* mais acessados no período de 03/11 a 09/11

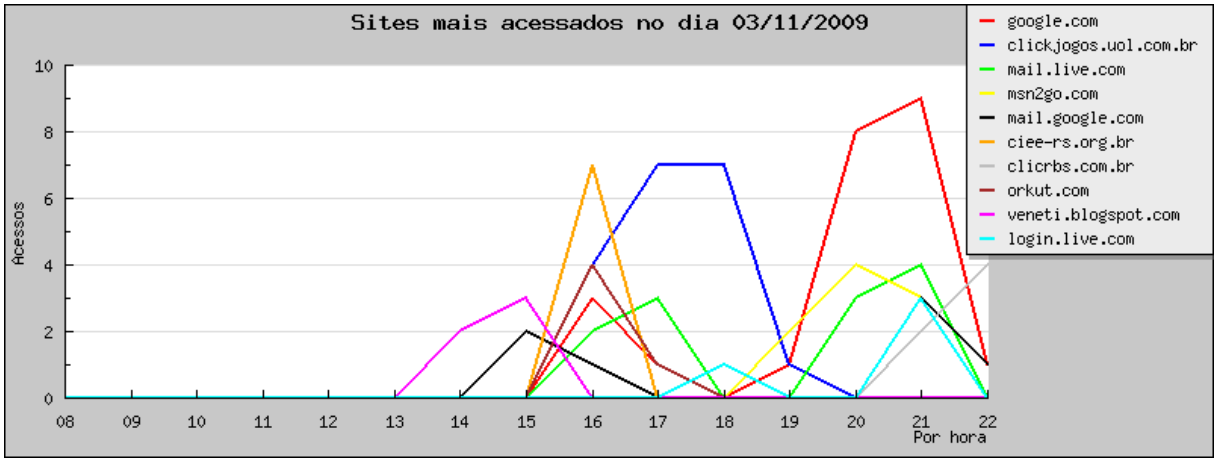

Figura 56 - *Sites* mais acessados no dia 03/11/2009

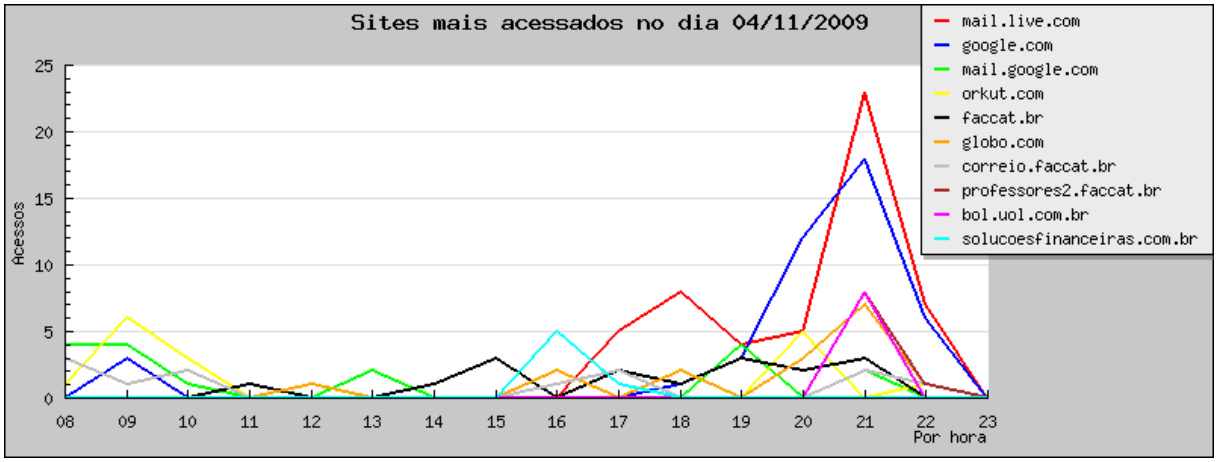

Figura 57 - *Sites* mais acessados no dia 04/11/2009

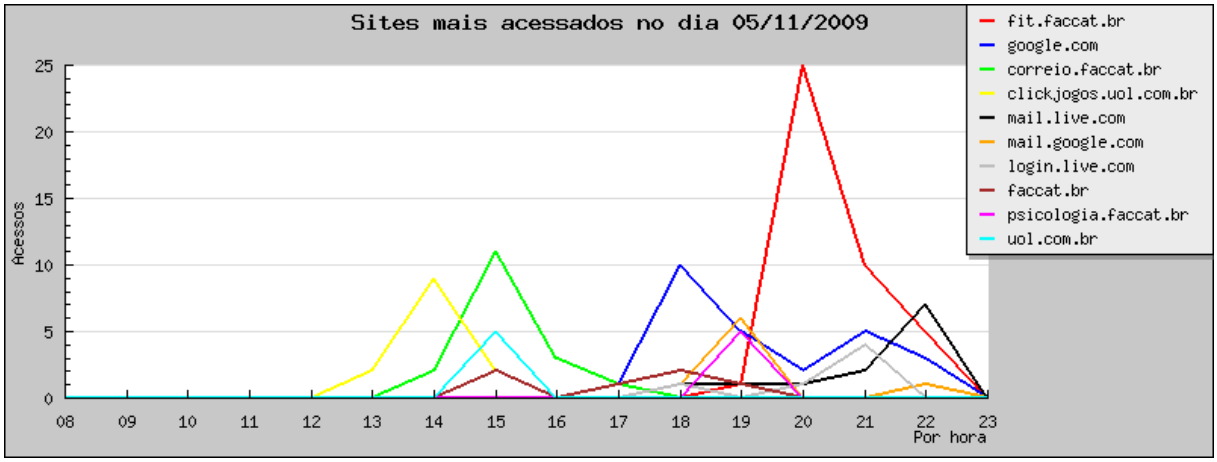

Figura 58 - *Sites* mais acessados no dia 05/11/2009

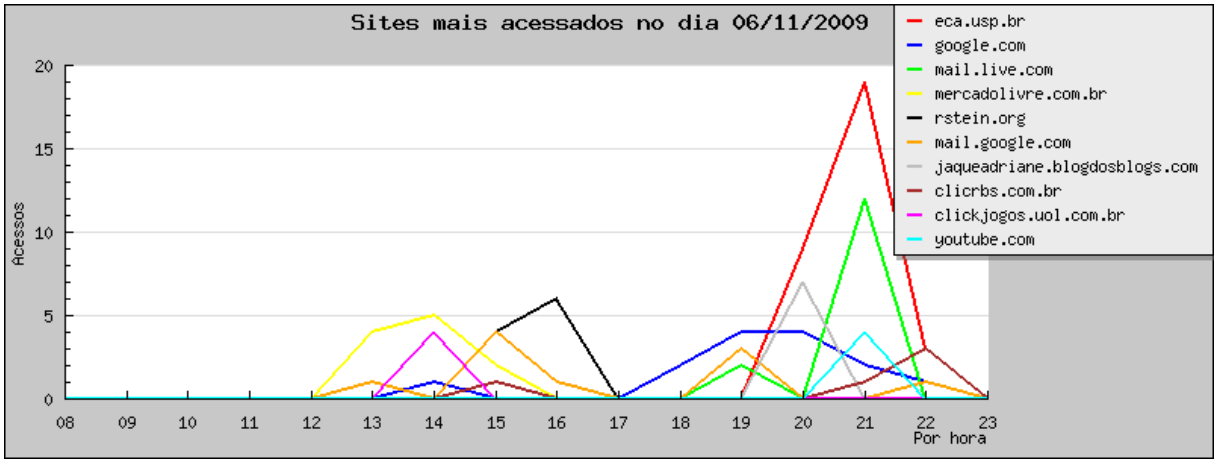

Figura 59 - *Sites* mais acessados no dia 06/11/2009

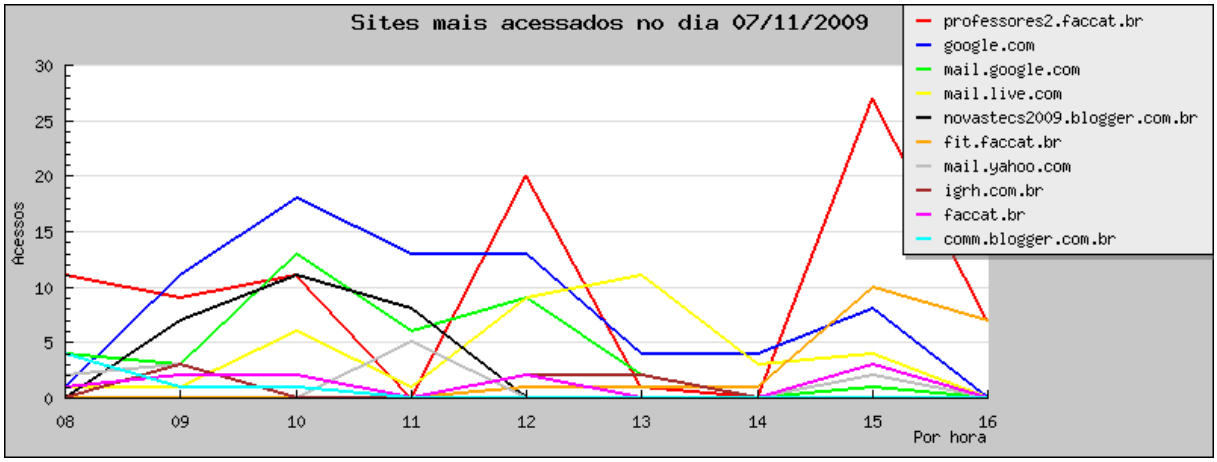

Figura 60 - *Sites* mais acessados no dia 07/11/2009

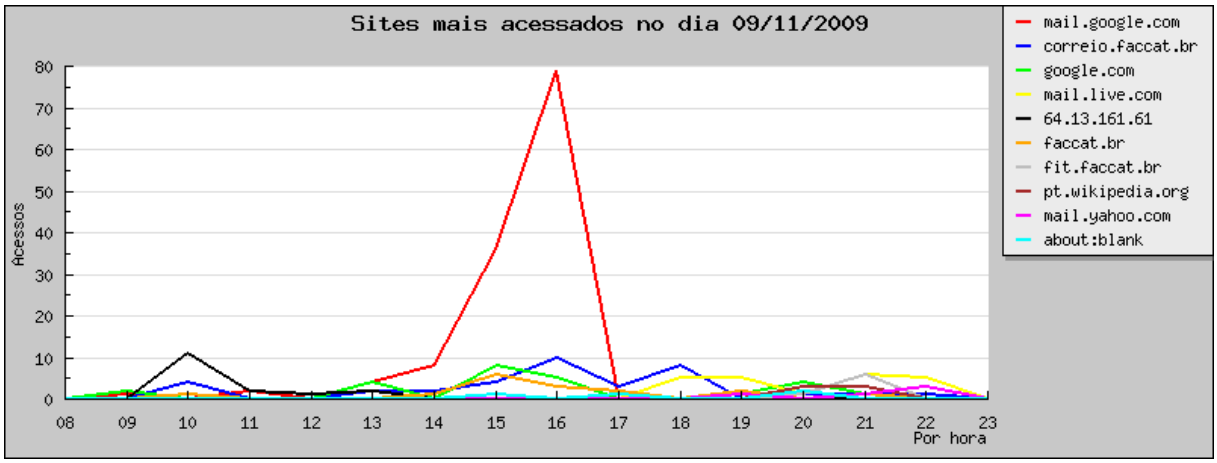

Figura 61- *Sites* mais acessados no dia 09/11/2009

7.5 Discussão

Os dados obtidos no monitoramento dos laboratórios demonstram que o período de maior utilização coincide com o período de aulas. É possível verificar também, que os aplicativos mais utilizados são os navegadores, seguido por aplicativos para visualizar apresentações e editores de texto.

Quanto ao acesso à Internet, os *sites* mais utilizados são os de pesquisa, correio eletrônico e o próprio domínio da FACCAT. De maneira geral, são poucos os acessos a *sites* considerados inapropriados. O uso de aplicativos se restringe aos disponíveis nas estações, já que os usuários não possuem permissão, junto ao sistema operacional, para instalar novos programas.

Não é possível perceber mudanças significativas na utilização dos aplicativos ou no acesso à Internet decorrente do fato dos usuários estarem, ou não, cientes do monitoramento

8 CONCLUSÃO

A grande quantidade de equipamentos, serviços e recursos que compõem uma rede de computadores faz com que o seu monitoramento e controle sejam imprescindíveis. A importância dos dados, que trafegam através da rede, e dos serviços instalados, torna a segurança um fator de grande relevância na administração e planejamento de TI. O monitoramento é uma das diversas práticas de segurança, através da qual, o administrador pode controlar a utilização dos recursos humanos e computacionais da organização. Monitorando as atividades realizadas pelos usuários nas estações, é possível estimar a sua produtividade e avaliar se os recursos tecnológicos estão alocados adequadamente.

O presente trabalho apresentou as etapas e estudos realizados para desenvolver um sistema *open source* de monitoramento das atividades realizadas pelos usuários nos computadores que integram uma rede. O código fonte do sistema está disponível no site do Portal da Inovação – Vale do Paranhana (PORTAL).

Neste estudo foram apresentados os conceitos de gerenciamento de rede e uma breve explicação sobre o funcionamento do protocolo de gerenciamento de rede SNMP. O agente Net-SNMP, empregado no desenvolvimento do agente SPY007, mostrou-se flexível e eficiente. Embora não forneça nativamente os dados necessários para o monitoramento planejado, possibilita integrar rotinas externas para esta função. A linguagem PHP, utilizada para o desenvolvimento do gerente do SPY007, possui funções SNMP nativas que simplificaram a sua implementação.

O sistema desenvolvido mostrou-se eficiente para as funções de monitoramento que foram planejadas, os gráficos que são disponibilizados sintetizam as informações coletadas de forma clara e precisa. Uma característica importante do SPY007, é que todas as informações provenientes de coleta de dados, trafegam através do protocolo SNMP, enquanto que, as ferramentas semelhantes citadas implementam um serviço específico para transferência de dados. O desenvolvimento da interface gráfica WEB possibilitou que o sistema seja acessado através de um navegador, facilitando o acesso remoto à ferramenta. O teste de aplicação do sistema demonstrou a efetividade da ferramenta desenvolvida, já que o monitoramento dos laboratórios, onde o agente foi instalado, foi realizado com sucesso.

Como projeto futuro estima-se a possibilidade de integrar o aplicativo desenvolvido neste estudo, com o *software* OCSinventory que é um aplicativo *open source*,

multiplataforma, que faz o inventário de *hardware* e *software* dos computadores que integram uma rede.

REFERÊNCIAS

ALENCAR, Rafael. **Smarty Template Engine – alguém já ouviu falar.** Disponível em: <http://www.artigosetutoriais.com/uploads/materias/diagrama_smarty.jpg>. Acesso em 31 de out. 2009.

ASLESON, Ryan; SCHUTTA, Nathaniel T. **Fundamentos do Ajax.** Alta Books. Rio de Janeiro, 2006.

BATISTELA, Letícia. **Uso da Internet e informática coorporativa.** Disponível em: <http://www.lbconsultoria.com.br/Arquivos/UsodaInterneteInformaticaCoorporativa.pdf>. Acesso em: 19 out. 2009.

COMER, Douglas E; STEVENS, David L. **Interligação em rede com TCP/IP.** (trad.) GUZ, Ana Maria Neto. 3 ed. Rio de Janeiro: Campus, 1999.

CETIC.BR. Centro de estudos sobre as tecnologias da informação e da comunicação. **TIC Empresas 2008.** Disponível em: <http://www.cetic.br/empresas/2008/analise-tic-empresas-2008.pdf>. Acesso em: 19 out. 2009.

CGI.BR. Comitê Gestor da Internet no Brasil. **Cartilha e Segurança para Internet**. São Paulo: Comitê Gestor da Internet no Brasil, 2006.

CLOPER, Luana. **Ajax: reinventar a Internet.** Disponível em: <http://www.timaster.com.br/revista/materias%5Cmain_materia.asp?codigo=1086>. Acesso em: 31 out. 2009.

CONCERINO, Arthur J. **Internet e Segurança são compatíveis?** In: LUCCA, Newton De; SIMÃO FILHO, Adalberto (coord.). Direito & Internet: Aspectos Jurídicos Relevantes. 2 ed. São Paulo: Quartier Latin, 2005.

DANESH, Arman. **Dominando o Linux**. São Paulo: Makron Boks, 2000.

FOWLER, Martin. **Padrões de arquitetura de aplicações corporativas.** (trad.) FERNANDES, Acuan. Porto Alegre: Bookman, 2006.

GALLO, Michael A.; HANCOCK, William M. **Comunicação entre computadores e tecnologias de rede.** (trad.) SILVA, Flávio Soares Corrêa da; CARNEIRO, Márcio Rodrigo de Freitas; Melo, Ana Cristina Vieira. São Paulo: Pioneira Thomson Learning, 2003.

GIL, Antonio de Loureiro. **Segurança em informática**. São Paulo: Editora Atlas SA, 1998.

IVIRTUA Solutions. **Tz0 Productivity and software metering,** Disponível em: <http://www.ivirtua.com.br/index.php?conteudo=solutions&pg=car2>. Acesso em: 28 out. 2009.

JÚNIOR, José Helvécio Teixeira. *et al.* **Redes de computadores: serviços, administração e segurança.** São Paulo: Makron Books, 1999.

KROLOW, Roger A. **Sistema de controle de consumo para redes de computadores.** 2000. 109f. Dissertação (Programa de pós-graduação em computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.

KUROSE, James F.; ROOS, Keith W. **Redes de computadores e a Internet: Uma abordagem top-down.** (trad.) MARQUES, Arlete Simille. 3 ed. São Paulo: Pearson Education, 2006.

LEITE, Silvio Luis. **Integrando ferramentas de software livre para gerenciamento e monitoração de redes locais.** 2004. 109f. Dissertação (Mestrado em informática) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.

LEMAY, Laura. **Aprenda a criar páginas Web com HTML e XHTML em 21 dias.** São Paulo: Prentice Education do Brasil, 2002.

LOUREIRO, Thiane. **Monitorar a Internet é a melhor forma de prevenir crises online.** Site IDG Now. Disponível em: <http://idgnow.uol.com.br/internet/2007/08/10/idgnoticia.2007-08-10.4912122363/>. Acesso em: 31 out. 2009.

MACEDO, Marcelo da Silva. **CSS (folhas de estilo): dicas & truques.** Rio de Janeiro: Ciência Moderna, 2006.

MAUJOR. Site do Maujor. **Tutorial XHTML**. Disponível em <http://maujor.com/tutorial/xhtml.php>. Acesso em 31 de out. 2009.

MySQLAB. **Visão geral do sistema de gerenciamento de banco de dados MySQL.** Disponível em: <http://dev.mysql.com/doc/refman/4.1/pt/what-is.html>. Acesso em: 31 out. 2009.

NEMETH, Evi. *et al.* **Manual do administrador do sistema Unix.** (trad.) FURMANKIEWICZ, Edson. 3 ed. Porto Alegre: Bookman, 2002.

NETCRAFT. **October 2009 Web Server survey.** Disponível em: <http://news.netcraft.com/archives/web_server_survey.html>. Acesso em: 31 out. 2009.

NETEYE. Disponível em: <http://www.neteye.com.br >. Acesso em: 28 out. 2009.

NET-SNMP. Disponível em: <http://www.net-snmp.org> Acesso em 01 nov. 2009.

OCSINVENTORY. Disponível em: <http://www.ocsinventory-ng.org> Acesso em 09 nov. 2009.

O'REILLY. **Essential SNMP.** Disponível em: <http://docstore.mik.ua/orelly/networking_2ndEd/snmp/index.htm>. Acesso em 31 out. 2009.

PORTAL DA INOVAÇÃO DO VALE DO PARANHANA. **Tecnologias disponíveis: Sistema de monitoramento da utilização de softwares em estações de trabalho para redes de computadores - código fonte.**
Disponível em: <http://portaldainovacao.faccat.br/moodle/course/view.php?id=7> Acesso em: 9 Dez 2009.

PRESSMAN, Roger S. **Engenharia de software**. (trad.) TRAVIESO, Mônica Maria G. 5 ed. Rio de Janeiro: McGraw-Hill, 2002.

RNP. **Introdução a Gerenciamento de Redes TCP/IP.** Disponível em <http://www.rnp.br/newsgen/9708/n3-2.html>. Acesso em 23 de out. 2009.

SCOTT, Kendall. **O processo unificado explicado.** (trad.) PRICE, Ana M. de Alencar. Porto Alegre: Bookman, 2003.

SOARES, Walace. **Programando em PHP: conceitos e aplicações.** São Paulo: Érica, 2000.

SOMMERVILE, Ian. **Engenharia de software.** (trad.) ANDRADE, Maurício de. 6 ed. São Paulo: Addison Wesley, 2003.

SZTAJNBERG, Alexandre. **Gerenciamento de Redes.** Disponível em <http://www.gta.ufrj.br/~alexszt/ger/snmpcmip.html#sec1>. Acesso em 23 de out. de 2009.

TANENBAUM, Andrew S. **Redes de computadores.** (trad.) Insight Serviços de Informática. 3 ed. Rio de Janeiro: Campus, 1997.

W3SCHOOLS. **Introdução ao XHTML.** Disponível em: <http://www.w3schools.com/xhtml/xhtml_intro.asp>. Acesso em: 31 out. 2009.

WTCS. Disponível em <http://www.wtcs.org>. Acesso em 31 out. 2009.