UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

FERRAMENTA PARA O CONTROLE DE E-MAILS E ANTI-*SPAM*

GUILHERME LUIS EBERHARDT

BLUMENAU
2007

2007/2-08

GUILHERME LUIS EBERHARDT

FERRAMENTA PARA O CONTROLE DE E-MAILS E ANTI-*SPAM*

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Sistemas de Informação — Bacharelado.

Prof. Francisco Adell Péricas, Mestre

BLUMENAU
2007

2007/2-08

FERRAMENTA PARA O CONTROLE DE E-MAILS E ANTI-*SPAM*

Por

GUILHERME LUIS EBERHARDT

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente:	Prof. Francisco Adell Péricas, Mestre – Orientador, FURB
Membro:	Prof. Sérgio Stringari
Membro:	Prof. Paulo Fernando da Silva

Blumenau, 11 de dezembro de 2007

Dedico este trabalho a todos os amigos, especialmente aqueles que me ajudaram diretamente na realização deste. Ao meu orientador pelo apoio neste trabalho, a minha família ao me proporcionar grande parte desta faculdade, à minha namorada pelo incentivo e ajuda nesta jornada.

AGRADECIMENTOS

À Deus, por sempre mostrar um caminho nas horas mais difíceis da vida.

À minha família, que sempre me proporcionou condições para um excelente estudo.

À minha namorada, Joice Deglmann, pelo incentivo, força e compreensão nas horas difíceis encontradas desta jornada.

Aos meus amigos, pela cobrança e motivação.

Aos colegas de trabalho que me incentivaram e me auxiliaram na conclusão deste trabalho.

Ao meu orientador, Francisco Adell Péricas, por ter acreditado na conclusão deste trabalho.

Os bons livros fazem “sacar” para fora o que a pessoa tem de melhor dentro dela.

Lina Sotis Francesco Moratti

RESUMO

Com o acentuado crescimento de mensagens em massa não solicitadas pelos destinatários, a tarefa de administração de *e-mails* tornou-se muito importante e ao mesmo tempo complexa. Este trabalho apresenta o desenvolvimento de uma ferramenta para controle de *e-mails* e anti-*spam* via *web*. Utilizou–se o Postfix como sendo o servidor de *e-mail* do projeto utilizando a linguagem Perl para seu desenvolvimento e o banco de dados MySQL para armazenamento de regras de mensagens e *e-mails*.

Palavras-chave: Postfix. Linux. *E-mail.* Perl. MySQL. *SPAM*.

ABSTRACT

With the accentuated growth of messages in mass not requested by the addressees, the task of administration of e-mails became very important and at the same time complex. This work presents the development of a tool for control of e-mails and anti-spam through web. Postfix was used as being the e-mail server of the project using the language Perl for its development and the database MySQL for storage of rules of messages and e-mails.

Key-words: Postfix. Linux. *E-mail.* Perl. MySQL. *SPAM*.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

CTSS - *Compatible Time-Sharing System*

DBI - *Database Interface*

HD - *hard disk*

HTML - *HyperText Markup Language*

IMAP - *Internet Message Access Protocol*

IP - *Internet Protocol*

ISP - *Internet Service Provider*

MTA - *Mail Transfer Agent*

MIT - *Massachusetts Institute of Technology*

POP3 - *Post Office Protocol[*

SDC - *System Development Corporation*

SMTP - *Simple Mail Transfer Protocol*

TCP - *Transmission Control Protocol*

UCE - *Unsolicited Commercial E-mail*

UML - *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO

Atualmente na administração da área de correio eletrônico, ou seja *e-mails*, há deficiências tanto no controle de *spams* quanto no uso indevido deste serviço por determinados funcionários e/ou colaboradores das empresas. *Spam* é uma mensagem eletrônica não solicitada enviada em massa.

Em forma mais comum, o *spam* consiste em uma mensagem eletrônica para fins publicitários, geralmente com a característica de uma mensagem apelativa. Os tipos mais comuns de *spams* são conhecidos como boatos, correntes, vírus, propagandas, golpes, entre outros.

> "No ambiente da Internet, *spam* é considerado um abuso e se refere ao envio de um grande volume de mensagens não solicitadas, ou seja, o envio de mensagens indiscriminadamente a vários usuários, sem que estes tenham requisitado tal informação. O conteúdo do *spam* pode ser: propaganda de produtos e serviços, pedido de doações para obras assistenciais, correntes da sorte, propostas de ganho de dinheiro fácil, boatos desacreditando o serviço prestado por determinada empresa, dentre outros." (TEIXEIRA, 2004)

O controle de uso de correio eletrônico dentro das empresas também é um assunto polêmico, já que se trata de uma correspondência. Muitos defendem a privacidade do funcionário e/ou colaborador, já outros defendem o livre controle de conteúdo do mesmo já que julgam como uma ferramenta fornecida pela empresa.

Segundo Dalazen (2005), o e-mail corporativo é considerado uma ferramenta de trabalho, no qual em princípio é considerado de uso profissional e de controle da empresa.

A partir destes conceitos de controle de uso de e-*mail* integrado com um anti-*spam* percebeu-se a necessidade de desenvolvimento de uma ferramenta que integre essas duas funções e facilite a administração de um servidor de e-*mail*. Atualmente esta administração de *spam* é feita por softwares já existentes no mercado, mas não integrados a um software de auditoria de e-*mail*.

Deste modo, com a presente proposta objetiva-se o desenvolvimento de uma ferramenta de anti-*spam* e controle de e-*mails* por usuário integrada, facilitando assim o gerenciamento dos mesmos.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi desenvolver uma ferramenta que gerencie o uso de correio eletrônico e o controle de *spam*.

Os objetivos específicos do trabalho são:

a) coletar as informações, como destinatário, remetente, assunto, data, hora, entre outras, dos e-mails que entram e saem de uma organização armazenando-os em uma base de dados, integrada no gerenciador de caixa postal do *Mail Transfer Agent* (MTA) Postfix;
b) anti-*spam* com base nos e-mails coletados operando integrado ao Postfix;
c) gerenciar e-mails que entram e saem de uma organização, podendo assim ser feita uma possível auditoria.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está organizado em quatro capítulos. A estrutura do trabalho foi definida seguindo os padrões propostos pela coordenação. No primeiro capítulo tem-se a introdução, a justificativa do trabalho, o objetivo geral e objetivos específicos. No segundo capítulo é exposta a fundamentação teórica, onde são abordados temas relevantes como E-MAIL, POSTFIX e SPAM. Além também de trabalhos correlatos. No terceiro é abordado o desenvolvimento do trabalho. Para um melhor entendimento esta seção foi dividida em requisitos principais, especificação, implementação, técnicas e ferramentas utilizadas e resultados. E como quarto capítulo é exposto a conclusão final e as possíveis extensões, este último com sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo são apresentadas algumas funcionalidades existentes no agente de transporte de e-mail, *Mail Transfer Agent* (MTA), *Postfix,* que será utilizado como base para o desenvolvimento da ferramenta e os conceitos sobre e-mail e *spam*.

2.1 E-MAILS

Como define Levine (1997), o e-mail é uma ferramenta de trabalho e, principalmente, um meio de comunicação. Apela a uma linguagem diferente, uma mistura de livre oralidade e de escrita, de abreviaturas, símbolos e meias palavras - como *smileys* ou acrônimos. A utilização de regras de etiqueta on-line é recomendada, embora não obrigatória.

Mas utilizar o correio eletrônico implica mais do que clicar o botão “enviar". De início, além de um computador, um *modem*, uma linha telefônica, uma conta de correio eletrônico (ou várias) e software adequado, conta-se com uma mensagem e o endereço do destinatário. A conta de e-mail, gratuita ou não, pode estar associada a um *Internet Service Provider* (ISP), a outras entidades, ou simplesmente residir na *Web* acessível a partir de qualquer computador com ligação à Internet.

> “Normalmente chamado de e-mail, o correio eletrônico é o serviço disponibilizado na rede mundial de computadores, a Internet, para troca de mensagens, a princípio constituídas apenas de texto (cadeia de caracteres). O sistema de correio eletrônico utiliza os *hostnames*, para endereçamento dos usuários. Assim, as mensagens que passeiam pela rede, se destinam a um determinado usuário em determinado sistema. São os endereços eletrônicos que se vêem hoje em dia, em qualquer meio de comunicação: um nome de usuário e um *hostname* separados pelo caractere @. Na rede mundial, para cada nome de sistema que existe, geralmente existe um número IP identificando o servidor que cuida das mensagens eletrônicas. Normalmente as mensagens de um usuário são entregues ao servidor destinatário pelo servidor da rede local do usuário.” (CARNEIRO, 2000)

2.1.1 História

Segundo Levine (1997), o surgimento do correio eletrônico, e-mail, é anterior ao da Internet. O surgimento deste sistema, e-mail, foi uma ferramenta crucial para a criação da rede internacional de computadores.

Em 1965 tem-se notícia do primeiro sistema criado de troca de mensagens entre computadores, e possibilitava a comunicação entre os múltiplos usuários de um computador do tipo *mainframe*. Mesmo a história sendo um tanto obscura, acredita-se que os primeiros sistemas criados com tal funcionalidade foram o Q32 da *System Development Corporation* (SDC) e o CTSS do *Massachusetts Institute of Technology* (MIT).

Este sistema eletrônico de mensagens acabou se transformando rapidamente em um e-mail em rede, permitindo desta forma que usuários situados em diferentes computadores trocassem mensagens. Também não é muito claro qual foi o primeiro sistema que suportou o e-mail em rede. O sistema AUTODIN, em 1966, parece ter sido o primeiro a permitir que mensagens eletrônicas fossem transferidas entre computadores diferentes, mas é possível que o sistema SAGE tivesse a mesma funcionalidade algum tempo antes. (LEVINE, 1997)

Para a evolução do e-mail, a rede de computadores ARPANET fez uma grande contribuição, aumentando a popularidade dos e-mails. Há um relato que indica a transferência de mensagens eletrônicas entre diferentes sistemas situados nesta rede logo após a sua criação, em 1969. O programador Ray Tomlinson iniciou o uso do sinal @ (arroba) para separar os nomes do usuário e da máquina no endereço de correio eletrônico em 1971. Pode-se considerar que ele foi o inventor do e-mail devido à importância dos seus programas de email: SNDMSG e READMAIL. Tem-se conhecimento que a primeira mensagem enviada por Ray Tomlinson não foi preservada; era uma mensagem anunciando a disponibilidade de um e-mail em rede. (LEVINE, 1997)

2.1.2 Terminologias utilizadas

Nesta sessão serão apresentadas algumas terminologias utilizadas quando se aborda o tema e-mail segundo Levine (1997).

- *auto-responders* (resposta automática) — O software do receptor responde

automaticamente após receber a mensagem;

- *bulk, bulking* ("baciada"): Sinônimo de *spam*, utilizado principalmente pelos *spammers;*
- *commercial e-mail* (e-mail comercial): e-mail enviado com finalidade comercial;
- *express consent* (consentimento expresso): O receptor concorda ativamente em receber e-mails selecionando uma opção em um formulário na *web* ou qualquer outra forma. Se por exemplo essa opção já estiver selecionada e o receptor não desativar a seleção, esse consentimento não é expresso;
- *false positives* (positivo falso): e-mails identificados erroneamente como *spam* pelo filtro do receptor;
- *format* (formatos): e-mails podem ser enviados em texto, HTML, ou *rich text format*.;
- *hard bounce*: e-mail retornado por nunca ter atingido seu destino porque o endereço de e-mail não existe;
- *list broker* (revendedor de listas): revendedor de listas de endereços de e-mails, conhecido também como *spammer*;
- *spam* ou UCE (*Unsolicited Commercial E-mail*): e-mail encaminhado sem o consentimento do receptor;
- *spam filter*: software utilizado para filtrar e-mails, evitando ou anunciando a presença de *spam*;
- *subject line* (assunto): campo destinado a dizer qual a finalidade da correspondência.

2.1.3 Protocolos

Nesta sessão serão abordados os protocolos mais comuns de transporte e leituras de e-mails compreendidos entre: POP3, SMTP e IMAP. Levine (1997), conceitua que os protocolos para Internet formam o grupo de protocolos de comunicação que implementam a pilha de protocolos sobre a qual a internet e a maioria das redes comerciais funcionam.

2.1.3.1 POP3

Post Office Protocol (POP3) é o protocolo que é utilizado para acesso remoto a uma caixa de correio eletrônico. Este protocolo, POP3, permite que as mensagens contidas em uma caixa de e-mail, correio eletrônico, possam ser transferidas de forma seqüencial para um computador local. Desta forma o usuário, utilizador do e-mail, poderá como exemplo: ler estas mensagens recebidas, apagá-las, respondê-las e/ou armazená-las.

A principal característica do protocolo POP3 é o funcionamento que se diz *off-line* uma vez que o processo suportado se baseia nas etapas de:

1. é estabelecida uma ligação TCP entre o cliente e servidor;
2. todas as mensagens contidas na caixa de correio são transferidas seqüencialmente para o computador;
3. as mensagens são apagadas da caixa de correio quando o protocolo for configurado para tal função;
4. a ligação com o servidor é terminada.

Neste caso, o utilizador pode agora ler e processar as suas mensagens de maneira *off-line*, ou seja, sem estar conectada a Internet.

A vantagem do protocolo POP3 é particularmente para utilizadores que se ligam à Internet através de redes públicas comutadas, em que o custo da ligação é proporcional ao tempo de ligação ou ainda quando o cliente deseja arquivar as mensagens recebidas.

2.1.3.2 SMTP

Simple Mail Transfer Protocol (SMTP) é um protocolo relativamente simples, baseado em texto simples, onde um ou vários destinatários de uma mensagem são especificados e, no caso do *Mail Transfer Agent* (MTA) Postfix validando estes destinatários sendo, depois, a mensagem transferida.

O teste de funcionamento de um servidor STMP é simples, realizado apenas utilizando um programa de *telnet*. Utiliza-se da porta 25 em uma rede *Transmission Control Protocol* (TCP) e com a resolução *Domain Name Server* (DNS) de um servidor SMTP de um determinado domínio, é possível através da sua entrada *Mail Exchange* (MX). É um protocolo apenas de envio de e-mail, ou seja, permite apenas que o usuário descarregue as mensagens de

um servidor.

O MTA Sendmail foi um dos primeiros agente de transporte de e-mail a implementar SMTP, e em 2001 já existiam cerca de 50 programas que implementam o STMP como cliente e/ou servidor de e-mails, entre eles o Postfix que foi utilizado para o desenvolvimento deste trabalho.

2.1.3.3 IMAP

Internet Message Access Protocol (IMAP) é um protocolo de gerenciamento de correio eletrônico no qual as mensagens ficam armazenadas no servidor e o usuário pode ter acesso a suas pastas e mensagens em qualquer computador, tanto por *webmail* como por cliente de correio eletrônico. Uma das vantagens deste protocolo é o compartilhamento de caixas postais entre usuários membros de um grupo de trabalho, facilitando a pesquisa de mensagens no servidor através de palavras chave. Outra vantagem é a realização de *backup*, pois as mensagens estando armazenadas no servidor centraliza o backup dos e-mails em um local único.

Mas o protocolo IMAP também tem suas desvantagens. Se o cliente de e-mail estiver em uma estação remota, utilizando a Internet, e esta por sua vez estiver com alguma falha ou sem comunicação, o usuário fica impossibilitado de verificar seus e-mails. Outra desvantagens é que o número de mensagens é delimitada ao espaço em *hard disk* (HD) disponível no servidor ou a cota definida deste espaço por usuário.

2.1.4 Problemas

Segundo Cert (2007), algumas das desvantagens do uso de e-mail encontram-se na falta de conhecimento da grande maioria dos internautas e, ainda, os *spammers* ou geradores de *spam*, grandes remetentes de vírus. Como citados em seguida:

- *Spam* – o *spam* é o termo usado para se referir aos e-mails não solicitados, que geralmente são enviados para um grande número de pessoas;
- Vírus - os e-mails tornam-se um excelente veículo de propagação de vírus,

sobretudo través dos arquivos anexos, recomendando-se nunca baixar um arquivo com extensão do tipo “exe”, que se trata de um arquivo executável ou outras extensões suspeitas.

Aconselha-se jamais abrir um e-mail desconhecido ou contendo arquivo não solicitado, exceto se for de um remetente confiável, mas nunca antes de observar os procedimentos de segurança.

2.2 POSTFIX

Segundo Marcelo (2004), o Postfix é um MTA, responsável pelo envio e recebimento das mensagens entre servidores, e é configurado para responder por um domínio, porém não entrega as mensagens aos clientes de e-mail (POP3), não baixa as mensagens de outros servidores, recebe mensagens quando enviadas diretamente a ele mas não vai buscar em outros servidores, não tem anti-vírus e nem outros filtros utilizados em servidores de e-mail, precisando de outros pacotes que se integram ao Postfix para configurar o que se chama de um servidor de e-mail.

O Postifix será utilizado como o MTA do sistema desenvolvido, justamente por não possuir um anti-*spam* ou qualquer filtro de e-mails próprio.

Segundo Michellis (2004), o funcionamento da arquitetura interna do sistema não possui um conceito muito rígido sobre processos "pai-filho", mas com a idéia de processos cooperativos. Basicamente, as tarefas que o Postfix roda são independentes, e provêem serviços umas as outras como exemplo: existe uma tarefa que prove reescrita/tradução de endereços para todos os outros processos do Postfix.

Há um processo principal, conhecido como *master*, que não faz nada a não ser administrar os outros *daemons* do Postfix, como exemplo:

- carregar o *smtpd* quando existe necessidade de atender a porta *Simple Mail Transfer Protocol* (SMTP);
- manter o *trivial-rewrite* rodando para fazer a manipulação dos endereços de correio;
- chamar o *qmgr* para gerenciar a fila de e-mails ainda não entregues;
- chamar o *pickup* que pega novas mensagens e as joga na fila para entrega.

O *master* também se encarrega de manter uma espécie de *cache* dos *daemons* que já foram iniciados, reutilizando processos ou removendo-os depois de um tempo específico em ociosidade. Isso diminui bastante o tempo e esforço despendido na maneira mais tradicional, chamando o processo e removendo-o logo que este acabe sua tarefa. Há sempre um tempo que leva para carregar um processo e todas as suas bibliotecas do disco, alocar memória. Da maneira como o *master* gerencia isso, esse tempo é praticamente nulo, uma vez que o processo já está no ar; ao mesmo tempo, o *master* mantém o sistema sob controle, não deixando processos inúteis ocupando recursos da máquina.

O *master* é o único processo que tem privilégios de *root*, ou seja, super usuário, uma vez que apenas o *root* tem poderes para mudar de usuário para chamar um agente de entrega que roda como o usuário XYZ, por exemplo, ou para ouvir a porta 25. Na maioria dos sistemas, portas abaixo de 1024 são privilegiadas, e apenas o *root* pode ouvi-las.

É interessante notar que o processo *master* não tem contato direto com nenhum outro processo ou com o mundo exterior; seu único objetivo é descartar privilégios e chamar o processo necessário, já em um ambiente desprivilegiado. (MICHELLIS, 2004)

Segundo Michellis (2004), existem duas formas de armazenamento de e-mails no Postfix: *MailBox* e *MailDir*. Este trabalho foi desenvolvido utilizando a forma de armazenamento *MailBox* abordada a seguir.

2.2.1 Sub-programas e parâmetros

Segundo Marcelo (2004), o Postfix se compõe de uma porção de pequenos programas e parâmetros, cada um com um tarefa específica para auxiliar o processo *master*. Neste item serão listados abaixo alguns deste sub-programas e parâmetros utilizados pelo Postfix:

- *postfix*: programa que inicia ou interrompe o processo *master* e os *daemons* que auxiliam no seu funcionamento, que pode utilizar os seguintes parâmetros:
    - *start*: inicia o serviço do *postfix*;
    - *stop*: interrompe o serviço do *postfix*;
    - *reload*: relê as configurações do *postfix*;
    - *abort*: para o *postfix* no ato;
    - *check*: verifica sintaxe dos arquivos;

  - *flush*: parâmetro para o programa *postfix* no qual move todas as mensagens contidas nos diretórios “HOLD” dos usuários e coloca na pasta *incoming* e ainda força o reenvio de todos os e-mails na fila;
- *postcat*: programa parecido com o *cat* do *shell*, exibe o conteúdo de um arquivo de e-mail;
- *postsuper*: ferramenta de manutenção de fila do Postfix que utiliza os seguintes parâmetros:
  - –H <mensagem>: marca a mensagem especificada como apta a ser movida para o diretório *incoming* do usuário;
  - –d <mensagem>: deleta a mensagem especificada;
  - –d ALL: deleta todas as mensagens;
- *postconf*: ferramenta de configuração do Postfix;

2.2.2 MailDir

Segundo Michellis (2004), o formato *Maildir* trata cada mensagem como um arquivo independente dentro de um diretório, por isso MAIL+DIR. Esse formato é bastante interessante porque agrega muitas vantagens:

- dispensa o uso de *LockFiles*, travamento de arquivos, ou seja, permite que o *MailBox* pode estar sendo escrito por vários programas ao mesmo tempo;
- não existe problemas com indexação do mailbox;
- torna-se desnecessário abrir e indexar o mailbox inteiro para extrair uma única mensagem, pois os e-mail são separados por arquivos;
- não existe a corrupção do *MailBox* pois é um diretório, ao máximo pode-se corromper uma única mensagem.

Devido a maior facilidade de escrita e leitura dos arquivos de mensagens de e-mail, foi optado utilizar o MailDir aqui citado para o desenvolvimento deste trabalho, podendo-se assim, ler e verificar cada mensagem de forma separada e segura.

2.3 SPAM

Segundo Cert (2006), o *spam* é o termo usado para se referir aos e-mails não solicitados, que geralmente são enviados para um grande número de pessoas. Quando o conteúdo é exclusivamente comercial, este tipo de mensagem também é referenciada como *Unsolicited Commercial E-mail* (UCE).

Em plena era de Internet comercial, o *spam* é uma das principais perturbações para internautas, administradores de redes e provedores, de tal forma que o abuso desta prática já se tornou um problema de segurança de sistemas. Além disso, é também um problema financeiro, pois vem trazendo perdas econômicas para uma boa parte dos internautas e lucro para um pequeno e obscuro grupo. (CERT, 2006)

Segundo Teixeira (2004), o primeiro *spam* via e-mail documentado foi enviado em 3 de maio de 1978. Já o uso do termo *spam* na Usenet completou 14 anos em março de 2007. A palavra *spam* começou a ser realmente difundida a partir de abril de 1994, quando Laurence Canter e Martha Siegel, dois advogados da cidade norte-americana de Phoenix, que trabalhavam em casos de imigração, enviaram uma mensagem anunciando serviços que teoricamente ajudavam as pessoas a ganhar vistos de permanência (*Green Card*) nos EUA. Por causa disso, a mensagem é hoje conhecida como *Green Card Spam* e, já na época, imediatamente gerou as mesmas reações que o *spam* atual, com questionamentos sobre ética e legalidade da prática. Não era uma mensagem nova, mas no dia 12 de abril eles usaram uma tática inovadora: contrataram um programador para criar um script simples e enviar o anúncio da dupla para todos os milhares de grupos de notícias da Usenet. O esquema deu certo e todos receberam o primeiro *spam* em larga escala da história, o que contribuiu para difundir o termo.

A partir daí, várias outras mensagens receberam o rótulo de spam, na maioria anúncios pessoais ou de empresas. Logo depois, as pessoas começaram a usar os programas de envio em massa de e-mails para enviar lixo eletrônico para grandes massas de usuários da rede.

Conforme Teixeira (2004), os tipos mais comuns de *spam* são:

Boatos e correntes:

Os boatos e as correntes na Internet têm algo em comum: pedem para serem enviados a todas as pessoas que você conhece. Tais e-mails se apresentam com diversos tipos de conteúdo, sendo na maioria das vezes histórias falsas ou antigas. Para atingir seus objetivos de propagação, os boatos e correntes apelam para diversos métodos de engenharia social.

Os boatos (*hoaxes*) são textos que contam estórias alarmantes e falsas, que instigam o leitor a continuar sua divulgação. Geralmente, o texto começa com frases apelativas do tipo: "envie este e-mail a todos os seus amigos...". Algumas classes comuns de boatos são os que apelam para a necessidade que o ser humano possui de ajudar o próximo. Como exemplos têm os casos de crianças com doenças graves, o caso do roubo de rins, entre outros.

Outros tipos de boatos são aqueles que difamam empresas ou produtos, prometem brindes ou ganho de dinheiro fácil. Continuando com os exemplos, há e-mails sobre a existência de certa substância cancerígena em determinado produto, o caso do e-mail que tratava da distribuição gratuita de telefones celulares, de viagens gratuitas a *Disneyworld*, entre outros.

Ainda dentre os boatos mais comuns na rede, pode-se citar aqueles que tratam de código malicioso, como vírus ou cavalos de tróia. Neste caso, a mensagem sempre fala de vírus poderosíssimos, capazes de destruir seu computador e assim por diante. Um dos mais famosos é o *Good Times*, que circulou pela rede durante anos e, de vez em quando, ainda aparece um remanescente enviado por internautas desavisados.

No Brasil, os boatos mais recentes foram sobre o roubo da Amazônia e a fiscalização de software em aeroportos.

As correntes, *chain letters*, são textos que estimulam o leitor a enviar várias cópias a outras pessoas, gerando um processo contínuo de propagação. São muito semelhantes aos boatos, mas o mecanismo usado para incentivar a propagação é um pouco diferente, pois a maioria das correntes promete sorte e riqueza aos que não as interrompem e anos de má sorte e desgraça aos que se recusam a enviar N cópias do e-mail para Y pessoas nas próximas X horas! Como exemplos têm a corrente dos índios da sorte, dentre tantas outras;

Propagandas:

Os *spams* com o intuito de divulgar produtos, serviços, novos sites, enfim, propagandas em geral têm ganhado cada vez mais espaço nas caixas postais dos internautas. Vale ressaltar que, seguindo o próprio conceito de *spam*, se recebemos um e-mail que não solicitamos, estamos sim sendo vítimas de *spam*, mesmo que seja um e-mail de uma super-promoção que muito nos interessa. O maior problema com a propaganda por *spam* é que a Internet se mostra como um meio fértil para divulgação de produtos, atinge um grande número de pessoas e a baixo custo, sendo que na verdade, quem paga a conta é quem recebe a propaganda;

Ameaças e/ou brincadeiras:

Alguns *spams* são enviados com o intuito de fazer ameaças, brincadeiras de mau gosto ou apenas por diversão. Ainda assim são considerados *spam*. Casos de ex-namorados difamando ex-namoradas, e-mails forjados assumindo identidade alheia e aqueles que dizem: "olá, estou testando uma nova ferramenta *spammer* e por isto você está recebendo este e-mail", constituem alguns exemplos. Vale lembrar que não há legislação específica para casos de *spam*. No entanto, podem-se enquadrar certos casos nas leis vigentes no atual Código Penal Brasileiro, tais como: calúnia e difamação, falsidade ideológica e estelionato.

2.3.1 Como atuam os Spammers

Segundo Teixeira (2004), em uma realidade em que cada vez mais pessoas consideram os spams uma praga a ser combatida a todo custo, os *spammers* utilizam técnicas no mínimo duvidosas para colocar seu lixo eletrônico na caixa postal de suas vítimas.

A primeira delas é a obtenção de listas de endereços de suas vítimas. Os endereços constantes em tais listas, apesar de serem apregoados como de pessoas que autorizaram o recebimento de e-mails comerciais, são coletados utilizando-se de técnicas nem sempre éticas, às vezes ilegais.

Caso nenhum endereço constante nessas listas autorizou o envio de mensagens, a propaganda pode ser pornográfica ou até mesmo uma oferta de drogas.

Muitos provedores procuram diminuir o recebimento de *spams* implementando filtros nas mensagens que chegam. Para evitar esses filtros os *spammers* enviam seus e-mails através de servidores de terceiros, mal configurados, que permitem o redirecionamento de mensagens para qualquer destinatário. Esses servidores são chamados de *open relay*, algo como servidores de retransmissão de e-mails liberada. (TEIXEIRA, 2004).

> Os "open relays" já foram úteis quando a Internet estava em seu início e havia poucos servidores no mundo. Como um usuário não precisa de autenticação para utilizar a conexão desta máquina, poderia usá-la em caso de necessidade ou sobrecarga da rede. Mas hoje, com o desenvolvimento comercial da Internet, eles perderam seu sentido original e servem praticamente apenas para que se abuse deles em várias atividades ilícitas ou antiéticas, como spam. (Aguilera, 2006)

Segundo Aguilera (2006), outro tipo de servidor mal configurado que pode ser utilizado para fins maliciosos é o *open proxy*, servidor que, consultado, busca páginas, figuras

e arquivos e os retransmite a quem fez a consulta. Um servidor desses, se mal configurado, também pode ser usado por *spammers*, escondendo a verdadeira origem da mensagem.

Mais uma técnica usada pelos *spammers* é o envio de mensagens usando falsos remetentes, procurando evitar filtros implantados pelo provedor ou pela vítima. Do mesmo modo, enviam mensagens com *subject* (assunto) totalmente diverso do conteúdo da mensagem, tentando fazer com que, curiosa, sua vítima leia o *spam*.

Segundo Teixeira (2004), algumas ferramentas utilizadas pelos *spammers* são aquelas capazes de enviar e-mails em grandes quantidades e outras que permitem falsificar o cabeçalho e outros campos de e-mail.

A maior fonte de ferramentas tanto para spammers quanto para hackers é a própria internet. Assim, é possível encontrar na rede os anonimizadores para o envio de e-mail e os programas para envio de e-mail em massa.

3 DESENVOLVIMENTO DO TRABALHO

Esta seção aborda a metodologia de desenvolvimento do trabalho. O tópico inicial descreve os requisitos atendidos pelo software. Os próximos tópicos referem-se à especificação da ferramenta, onde são apresentados alguns diagramas *Unified Modeling Language* (UML) visando uma melhor compreensão do software. O tópico seguinte refere-se à implementação do software, onde está descrito seu funcionamento, tecnologias utilizadas, operacionalidade e resultados obtidos.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

No quadro 1 são apresentados os requisitos funcionais previstos para a ferramenta e sua rastreabilidade, ou seja, vinculação com o(s) caso(s) de uso associado(s).

Requisitos Funcionais	**Caso de Uso**
RF01: A ferramenta deverá interpretar todos os e-mails que entram ou saem de um determinado servidor.	UC01
RF02: A ferramenta deverá gerar uma lista com os assuntos mais utilizados para uma lista de *spam*.	UC02
RF03: A ferramenta deverá emitir um relatório contendo os e-mails que entraram e saíram do servidor.	UC03
RF04: A ferramenta deverá permitir que o administrador cadastre palavras chaves que determinem se um e-mail é *spam* ou não.	UC04
RF05: A ferramenta deverá bloquear todos os e-mails que identifique como *spam*.	UC05
RF06: A ferramenta deverá listar todos os e-mails bloqueados como SPAM para liberação dos usuários se necessário.	UC06
RF07: A ferramenta deverá permitir o cadastro de usuários.	UC07
RF08: A ferramenta deverá permitir que o administrador cadastre palavras	UC08

chaves que determinem se um e-mail é um e-mail bloqueado ou não.	

Quadro 1: Requisitos funcionais

O Quadro 2 lista os requisitos não funcionais previstos para a ferramenta.

Requisitos Não Funcionais
RNF01: A ferramenta deverá utilizar como linguagem de desenvolvimento o Perl.
RNF02: A ferramenta deverá utilizar como base de dados o MySQL.
RNF03: A ferramenta deverá utilizar como servidor de transporte de e-mail o Postfix.
RNF04: A ferramenta deverá ter como sistema operacional base qualquer distribuição Linux.

Quadro 2: Requisitos não funcionais

3.2 ESPECIFICAÇÃO

Esta seção descreve os modelos e diagramas desenvolvidos durante o trabalho. Os primeiros tópicos tratam, respectivamente, os diagramas UML de caso de uso e de atividades. Ambos foram desenvolvidos utilizando o *Enterprise Architect 7*.

3.2.1 Diagrama de caso de uso

A figura 01 representa o diagrama de casos de uso. Este diagrama faz referência aos requisitos funcionais do software. Cada pacote representa as funcionalidades exclusivas de um ator. A descrição dos cenários encontra-se na seção Apêndice A.

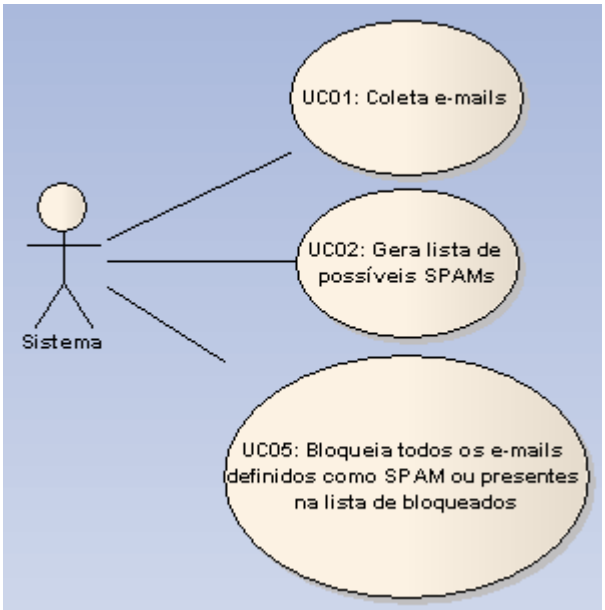

Figura 01: Diagrama de caso de uso da ferramenta.

Na Figura 02, o caso de uso do administrador.

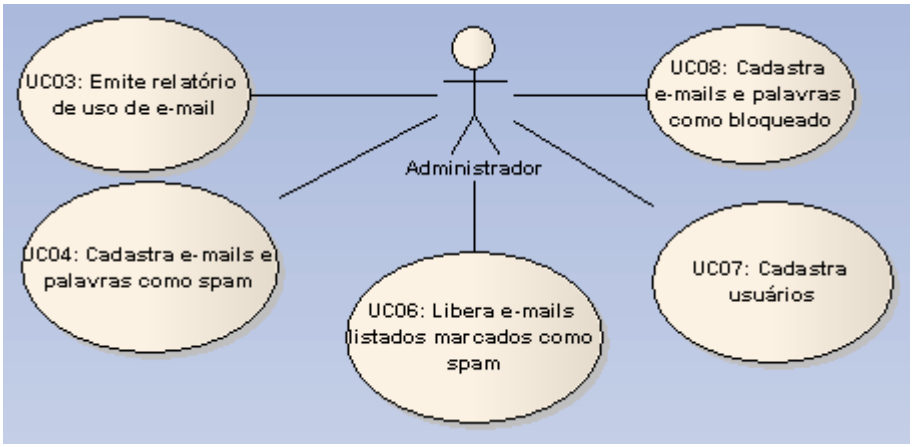

Figura 02: Diagrama de caso de uso do administrador.

3.2.2 Diagrama de atividades

A figura 03 representa o diagrama de atividades. Este diagrama faz referência às ações tomadas pelo servidor ao chegar um e-mail e pelo usuário ao analisar um e-mail para chegar ao cenário final, sendo o processo de verificar se um e-mail é definido como bloqueado ou spam pelo servidor e spam pelo usuário.

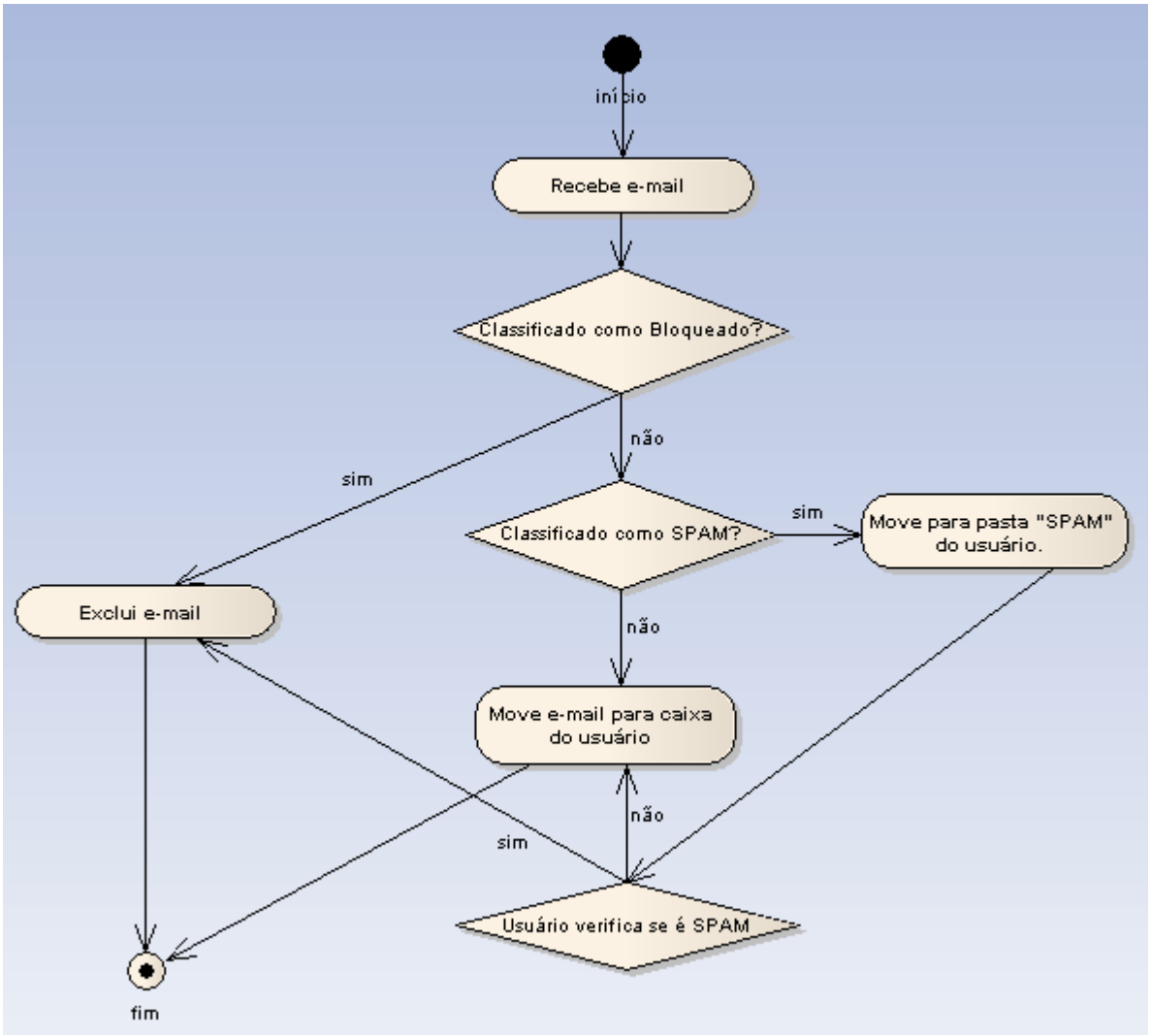

Figura 03: Diagrama de atividades.

3.2.3 Modelo de dados

O modelo de dados deste projeto é composto, inicialmente, por quatro tabelas, como pode ser visto na figura 04.

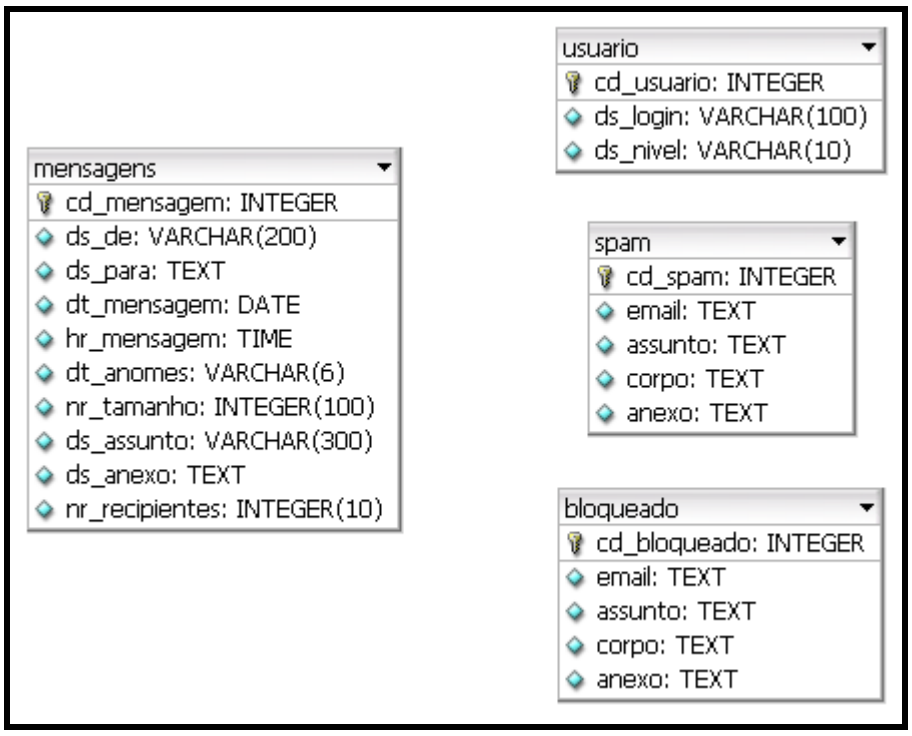

Figura 04: Modelo de dados.

Os propósitos das tabelas são os seguintes:

a) mensagens: armazena as informações de todos os e-mails que entram e saem do servidor;

b) usuario: armazena as informações de usuários que utilizam o serviço de e-mail;

c) spam: armazena os e-mails, assuntos, corpo e anexos que serão considerados nos e-mails como spam;

d) bloqueados: armazena os e-mails, assuntos, corpo e anexos que serão considerados nos e-mails como bloqueados.

3.3 IMPLEMENTAÇÃO

Esta seção contém o detalhamento sobre a implementação da ferramenta. O tópico inicial identifica as técnicas e ferramentas utilizadas. O tópico seguinte apresenta um estudo de caso do ponto de vista do usuário, destacando a funcionalidade ou operacionalidade do software. O último tópico descreve os resultados obtidos.

3.3.1 Técnicas e ferramentas utilizadas

O software implementado faz uso da ferramenta Perl para o desenvolvimento do sistema *web*, de coleta de e-mails e verificação de *spam*, o HTML, Javascript para a interface e navegação do sistema. Além, também, da base de dados MySQL. Nota-se que todas as tecnologias utilizadas no desenvolvimento deste software são do tipo *Freeware* (Software Livre) ou *Open Source* (Código Aberto).

Perl é uma linguagem de programação estável e multiplataforma, usada em aplicações de missão crítica em todos os setores, sendo destacado o seu uso no desenvolvimento de aplicações web de todos os tipos. Foi criada por Larry Wall em dezembro de 1987. A origem do Perl remonta ao *shell scripting*, *Awk* e linguagem C, estando disponível para praticamente todos os sistemas operacionais, embora seja usado mais comumente em sistemas Unix e compatíveis (WALL, 2001).

As figuras 05, 06, 07 e 08 apresentam o código Perl utilizado pelo software que coleta os e-mails que entram e saem do servidor.

Na figura 05 é utilizada a função para conexão ao banco de dados através do módulo Perl: *Database Interface* (DBI). Também demonstra a configuração para o módulo Perl *Mail::IMAPClient* que realiza a coleta e leitura de todos os e-mails que entram e saem do servidor.

```
#!/usr/bin/perl

use lib '/var/www/cgi-bin/mailreport';
use CGI qw/:standard/;
use Time::Local;
use DBI();

@mySQLConnect=("DBI:mysql:database=mailadm", "mailadm", "mail..adm");

&VerificaHoraSistema;

my $dbh = DBI->connect(@mySQLConnect);

##########################################################################
## Conectando na conta imap

use Mail::IMAPClient;

my $imap = Mail::IMAPClient->new;

### Aqui vao os dados do servidor!!!
#
# User copymail eh aonde sao copiados todos os e-mails
#

my $imap = Mail::IMAPClient->new(
                                  Server   => "127.0.0.1",
                                  User     => "copymail",
                                  Port     => "143",
                                  Password => "copy..mail",
                                  Peek     => 1,
                                 ) or die "Cannot connect: $@";

$imap->select("INBOX");
my @msg_list = $imap->search('UNDELETED');
```

Figura 05: Arquivo Perl de coleta de e-mails.

Na próxima figura inicia-se a leitura de todos os e-mails marcados como "*UNDELETED*" do servidor da caixa de e-mail *copymail*.

```
#########################################################################
## Lendo mensagens

foreach my $message (@msg_list)
{

  print "$message";

  my $data = $imap->parse_headers($message,"Subject","From","To","Date");
  my $EmailDe = $data->{From}->[0];
  my $EmailsPara = $data->{To}->[0];
  my $DataEmail = $data->{Date}->[0];
  my $Assunto = $data->{Subject}->[0];

  ($DiaS,$Dia,$Mes,$Ano,$HorarioMsg) = split(/ /,$DataEmail);

  if ($Mes eq "Jan") { $Mes = "01"; }
  elsif ($Mes eq "Feb") { $Mes = "02"; }
  elsif ($Mes eq "Mar") { $Mes = "03"; }
  elsif ($Mes eq "Apr") { $Mes = "04"; }
  elsif ($Mes eq "May") { $Mes = "05"; }
  elsif ($Mes eq "Jun") { $Mes = "06"; }
  elsif ($Mes eq "Jul") { $Mes = "07"; }
  elsif ($Mes eq "Aug") { $Mes = "08"; }
  elsif ($Mes eq "Sep") { $Mes = "09"; }
  elsif ($Mes eq "Oct") { $Mes = "10"; }
  elsif ($Mes eq "Nov") { $Mes = "11"; }
  elsif ($Mes eq "Dec") { $Mes = "12"; }

  $DataMsg = $Ano.'-'.$Mes.'-'.$Dia;
  $DataAnoMes = "$Ano$Mes";

  $EmailDe = $1
  if ($EmailDe =~ m/[<"]?([^\s@]+@[^\s@>"]+)"?>?/);

  if ($EmailDe=~/\</i) {
    /\</;
    $EmailDe = $';
  }
```

Figura 06: Arquivo Perl de coleta de e-mails.

```
my $size = $imap->size($message)
           or die "Could not find size of message $msgId: $@\n";

$EmailDe=~s/\'/\&\#039\;/g;
$DominioDe=~s/\'/\&\#039\;/g;
$DataMsg=~s/\'/\&\#039\;/g;
$HorarioMsg=~s/\'/\&\#039\;/g;
$size=~s/\'/\&\#039\;/g;
$Assunto=~s/\'/\&\#039\;/g;
$AnoMes=~s/\'/\&\#039\;/g;

(@wDestinatarios) = split(/\,/,$EmailsPara);

foreach $destinatarios (@wDestinatarios) {

  $i++;

  $destinatarios = $1
  if ($destinatarios =~ m/[<"]?([^\s@]+@[^\s@>"]+)"?>?/);
  $destinatarios=~s/\'/\&\#039\;/g;
  $wDest.= $destinatarios.'##';

}

my $CorpoMensagem = $imap->body_string($message);

while($CorpoMensagem=~s/filename=([^\s]+)//i) {

  $wAnexo = $1;
  $wAnexo=~s/\'/\&\#039\;/g;

  $DsAnexo.= $1.'##';

}
```

Figura 07: Arquivo Perl de coleta de e-mails.

Na figura 08 é mostrada a gravação dos e-mails lidos e seus dados necessários para gravação no banco de dados. Em seguida é realizada a exclusão de todos os e-mails lidos e gravados no banco de dados.

```
  $wInsert ='insert into mensagens ';

  $wInsert.='(ds_de,dt_mensagem,hr_mensagem,nr_tamanho,ds_assunto,dt_anomes,';
  $wInsert.='ds_para,ds_anexo,nr_recipientes) ';
  $wInsert.='values (\''.$EmailDe.'\',\''.$DataMsg.'\',\'';
  $wInsert.= $HorarioMsg.'\',\''.$size.'\',\''.$Assunto.'\',\''.$DataAnoMes.'\',\'';
  $wInsert.= $wDest.'\',\''.$DsAnexo.'\',\''.$i.'\')';

  $sth = $dbh->prepare("$wInsert");
  $sth->execute();

  $wInsertId = $sth->{mysql_insertid};

  $i = 0;


  $imap->delete_message($message) or die "Could not delete_message: $@\n";

}

## Deletando mensagens ################################################

$imap->expunge("INBOX") or die "Could not expunge: $@\n";

######################################################################

exit 0;

######################################################################

sub VerificaHoraSistema() {

   ($Ssec,$Smin,$Shour,$Smday,$Smon,$Syear,$Swday) = (localtime(time))[0,1,2,3,4,5,6];
   $time = sprintf("%02d:%02d:%02d",$Shour,$Smin,$Ssec);
   $Smon = $Smon + 1;
   if ($Smday < 10) { $Smday = "0$Smday"};
   if ($Smon < 10) { $Smon = "0$Smon"};
   $Syear += 1900;
}
```

Figura 08: Arquivo Perl de coleta de e-mails.

As figuras 09, 10, 11 e 12 apresentam o código Perl utilizado pelo software que verifica se o e-mail está bloqueado ou se é um *spam*.

```
#!/usr/bin/perl

use CGI qw/:standard/;
use Time::Local;
use DBI();

@mySQLConnect=("DBI:mysql:database=mailadm", "mailadm", "mail..adm");

my $dbh = DBI->connect(@mySQLConnect);
```

Figura 09: Função para conexão ao banco de dados através do módulo Perl: DBI.

```
#########################################################################
## Verificando lista de e-mails

$Dir_Emails = "/var/spool/postfix/hold/";
chdir ($Dir_Emails) or die "$1";

$y = 1;
while ($y) {

  @wLsHold = `ls $Dir_Emails`;

  foreach $email (@wLsHold) {
    $wPostsuper = `/usr/sbin/postsuper -H $email`;
  }

  $wFlush = `/usr/sbin/postfix flush`;

  @wBuscaUsuarios = `ls /home/`;

  foreach $usuario (@wBuscaUsuarios) {

    $usuario=~s/\n//;
    $usuario=~s/\r\n//;
    $usuario=~s/ //;
    $usuario=~s/\'/\&\#039\;/g;
    $usuario=~s/\</\&lt\;/g;
    $usuario=~s/\>/\&gt\;/g;
    $usuario=~s/\"/\&quot\;/g;
    $usuario=~s/\\/\&\#092\;/g;
    $usuario=~s/\$/\&\#036\;/g;

    next if ($usuario=~m/bloqueados/g);
    next if ($usuario=~m/copymail/g);

    $Dir_Usuario = '/home/'.$usuario.'/Maildir/new';

    chdir ($Dir_Usuario) or $Cria_Dir_Usuario = 1;

    if ($Cria_Dir_Usuario) {
      system "mkdir /home/$usuario/Maildir";
      system "mkdir /home/$usuario/Maildir/new";
    }
```

Figura 10: Programa busca todas as mensagens e cria os diretórios se necessário.

```
@wLs = `ls $Dir_Usuario`;

foreach $email (@wLs) {

  $BLOQUEADO = 0;
  $SPAM = 0;

  open EMAIL, "/home/$usuario/Maildir/new/$email";
  @CorpoEmail = <EMAIL>;
  close EMAIL;

  $wBloq = $dbh->prepare("select email,assunto,corpo,anexo from bloqueado");
  $wBloq->execute();

  while(($Email_Bloq,$Assunto_Bloq,$Corpo_Bloq,
         $Anexo_Bloq)=$wBloq->fetchrow_array) {
    foreach $linha_email (@CorpoEmail) {
      if ($linha_email=~m/$Email_Bloq/ig) {
        $BLOQUEADO = 1;
      }
      if ($linha_email=~m/$Assunto_Bloq/ig) {
        $BLOQUEADO = 1;
      }
      if ($linha_email=~m/$Corpo_Bloq/ig) {
        $BLOQUEADO = 1;
      }
      if ($linha_email=~m/$Anexo_Bloq/ig) {
        $BLOQUEADO = 1;
      }
    }
  }
  $wSpam = $dbh->prepare("select email,assunto,corpo,anexo from spam");
  $wSpam->execute();

  while(($Email_Spam,$Assunto_Spam,$Corpo_Spam,
         $Anexo_Spam)=$wSpam->fetchrow_array) {
    foreach $linha_email (@CorpoEmail) {
      if ($linha_email=~m/$Email_Spam/ig) {
        $SPAM = 1;
      }
      if ($linha_email=~m/$Assunto_Spam/ig) {
        $SPAM = 1;
      }
      if ($linha_email=~m/$Corpo_Spam/ig) {
        $SPAM = 1;
      }
      if ($linha_email=~m/$Anexo_Spam/ig) {
        $SPAM = 1;
      }
    }
  }
```

Figura 11: Programa de verifica se o e-mail é *spam* ou deverá ser bloqueado.

```
      if ($BLOQUEADO == 1) {
        print "\n BLOQ";
        open BLOQUEADO, ">\/home\/bloqueados\/Maildir\/new\/$email";
        foreach $linha_email (@CorpoEmail) {
          print BLOQUEADO "$linha_email\n";
        }
        close BLOQUEADO;
        system "chmod 777 \/home\/$usuario\/Maildir\/new\/$email";
        system "rm -rf \/home\/$usuario\/Maildir\/new\/$email";
      } elsif ($SPAM == 1) {
        print "\n SPAM";
        $Dir_Usuario = '/home/'.$usuario.'/Maildir/spam';
        chdir ($Dir_Usuario) or $Cria_Dir_Usuario = 1;
        if ($Cria_Dir_Usuario) {
          system "mkdir /home/$usuario/Maildir/spam/";
          system "chown $usuario\:$usuario /home/$usuario/Maildir/spam/";
          $Cria_Dir_Usuario = "";
        }
        open SPAM, ">\/home\/$usuario\/Maildir\/spam\/$email" or print "\nkkk";
        foreach $linha_email (@CorpoEmail) {
          print SPAM "$linha_email\n";
          print "$linha_email\n"
        }
        close SPAM;
        system "chmod 777 \/home\/$usuario\/Maildir\/new\/$email";
        system "rm -rf \/home\/$usuario\/Maildir\/new\/$email";
        system "chmod 755 \/home\/$usuario\/Maildir\/spam\/$email";
      }

    }
  }
}

#########################################################################

exit 0;

#########################################################################
```

Figura 12: Programa remove a mensagem ou move para a pasta *spam*.

3.3.2 Operacionalidade da implementação

Esta seção apresenta um estudo de caso, do ponto de vista do usuário, objetivando mostrar a funcionalidade e operacionalidade do software.

Primeiramente na figura 13 é demonstrada a tela inicial do software, no qual é efetuada a entrada do sistema com usuário cadastrado no banco de dados e a senha do mesmo usuário previamente cadastrado no servidor.

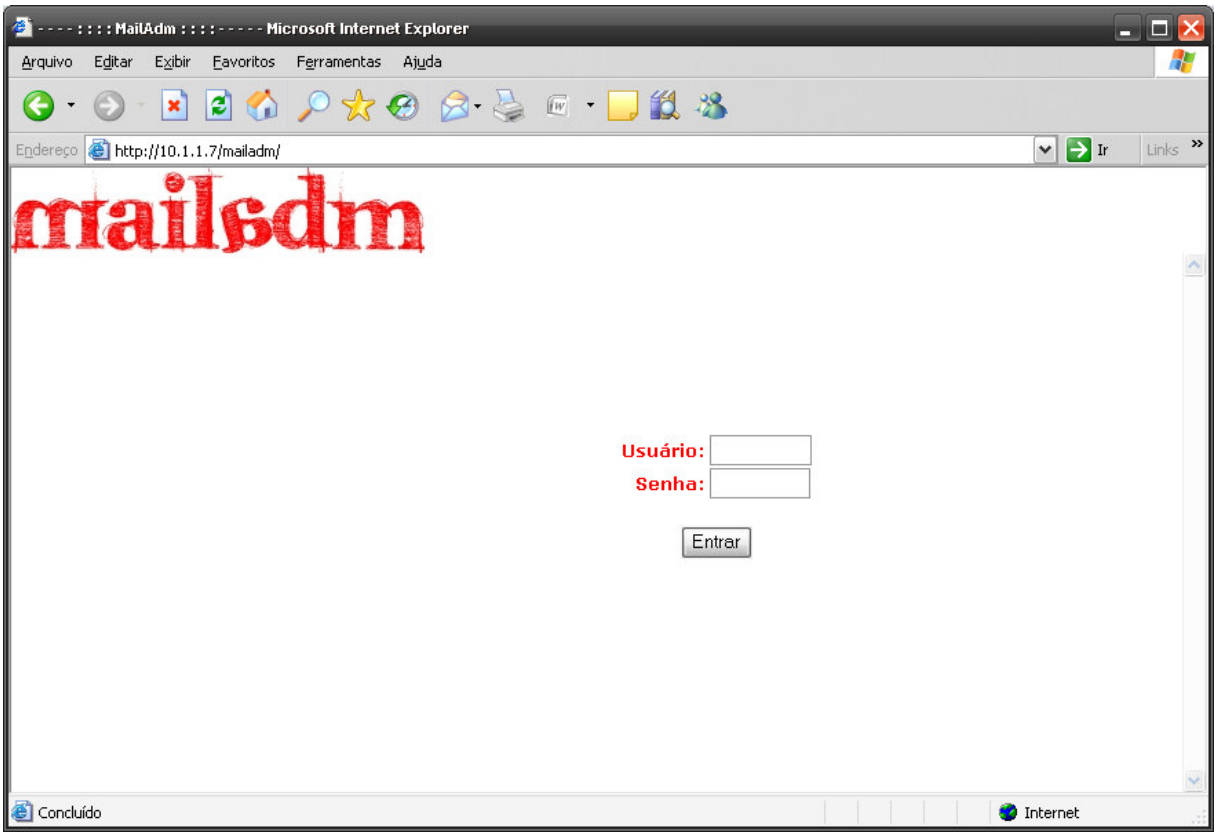

Figura 13: Tela de login do sistema.

Após a entrada do sistema, o mesmo apresenta um menu, apresentados nas figuras 14 e 15, de acordo com o nível do usuário, definidos como “usr” ou “adm”. O nível “usr” é definido para os usuários que poderão visualizar somente seus e-mails, já o nível “adm” é definido para os usuários que poderão visualizar os seus e-mails, cadastrar *spams* e bloqueios de e-mails.

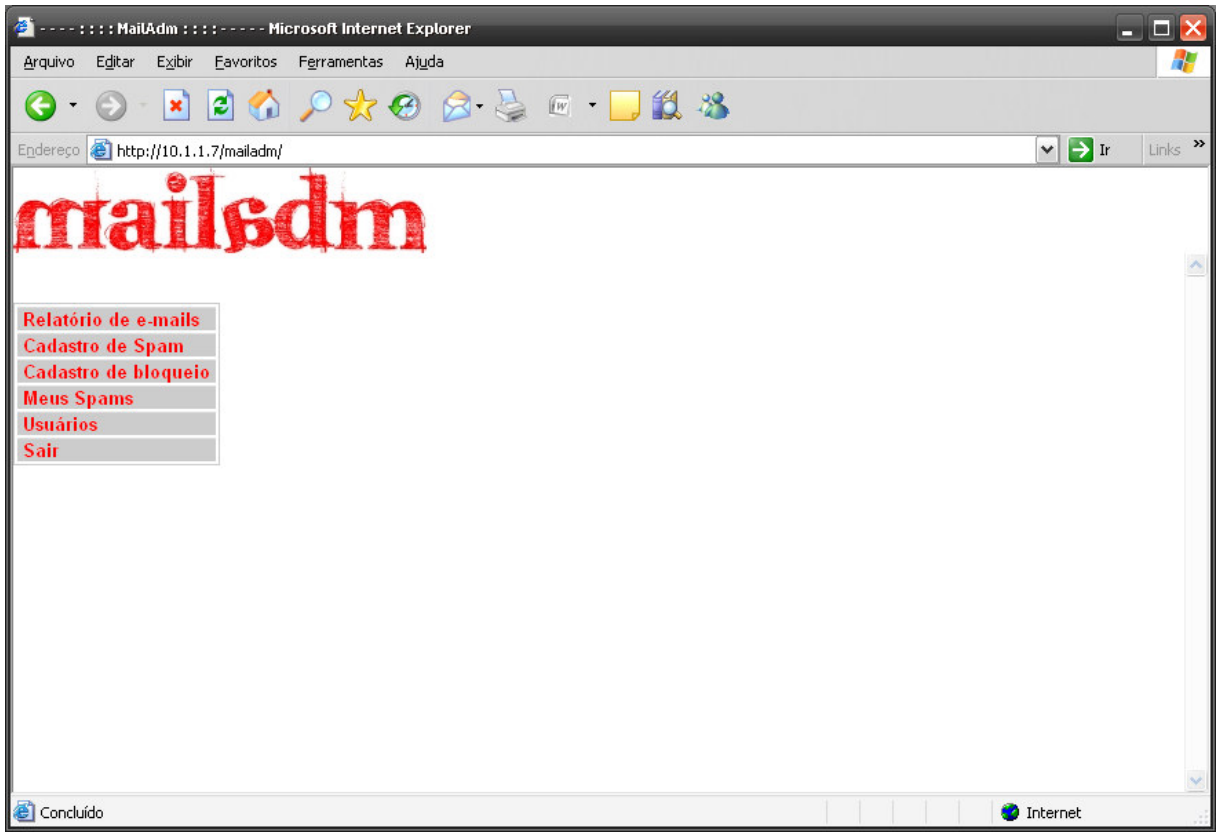

Figura 14: Menu do sistema do nível “adm”.

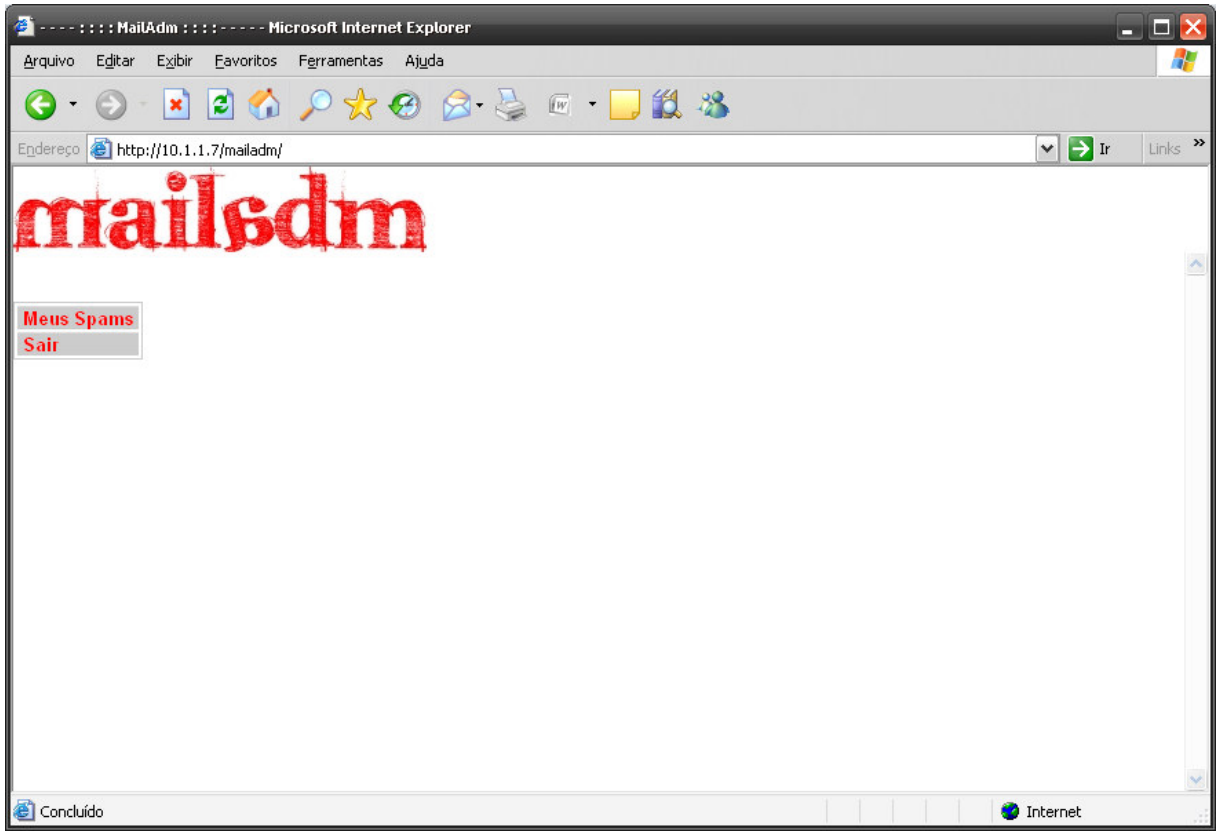

Figura 15: Menu do sistema do nível “usr”.

No item do menu “Relatório de e-mails “ do nível “adm”, encontramos um formulário para emissão de relatório de uso de e-mails, conforme a figura 16.

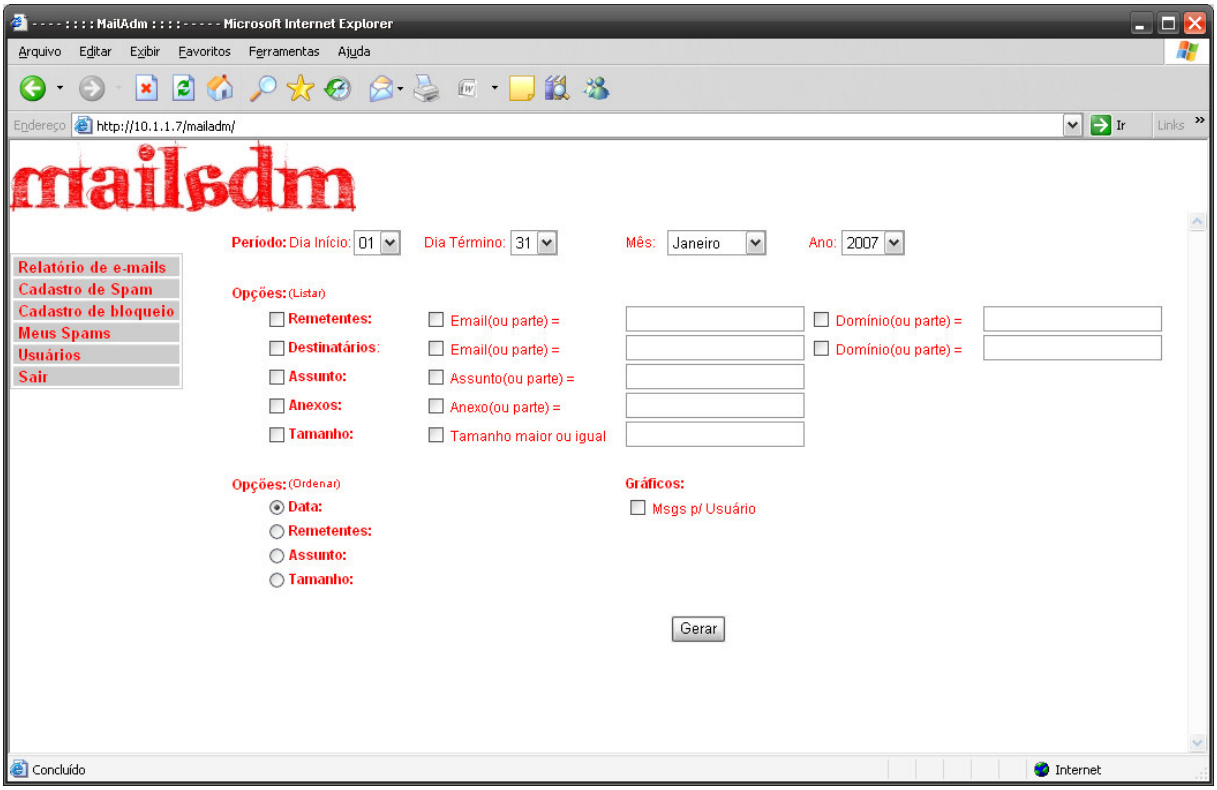

Figura 16: Formulário de emissão de relatórios.

Nos relatórios emitidos pelo sistema, os e-mails são separados por cores de fundo e existe ainda um gráfico de uso de e-mails por usuários e uma tabela que, em ordem decrescente, lista o uso em percentual dos e-mails por usuário do domínio pré-cadastrado, conforme mostrado na figura 17.

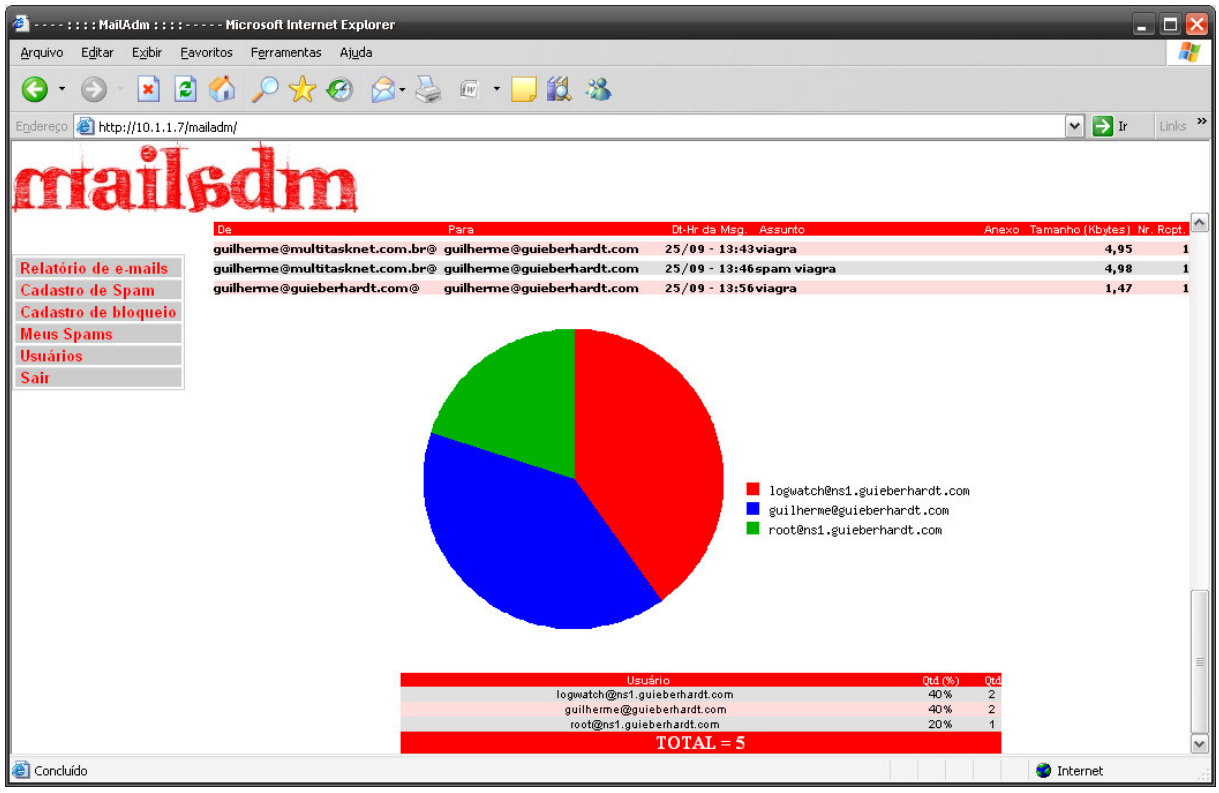

Figura 17: Exemplo de emissão de relatórios.

O software permite o cadastro de e-mails, assuntos, anexos e corpo definidos como *spam* e que devem ser bloqueados. Conforme é mostrado nas figuras 18 e 19 os formulários de cadastro de *spam* e bloqueios.

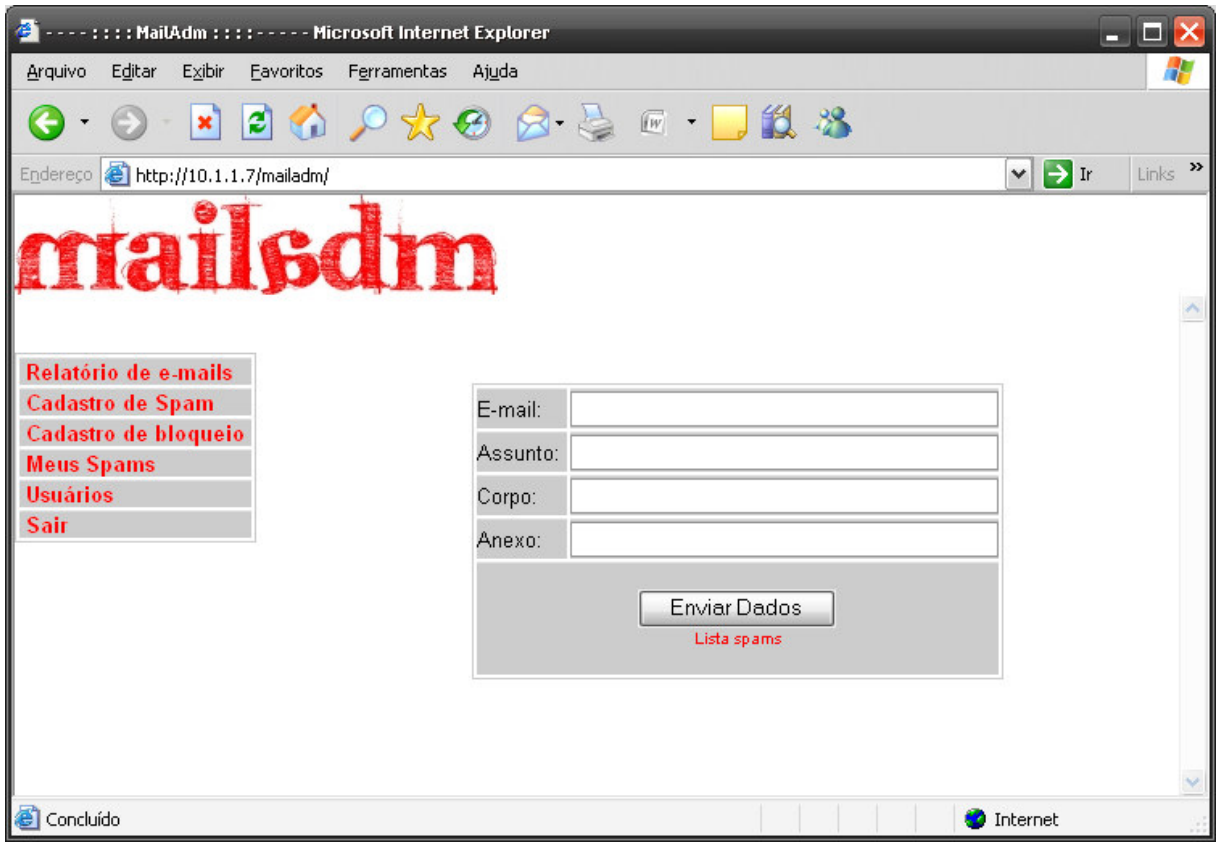

Figura 18: Formulário de cadastro de *spam*.

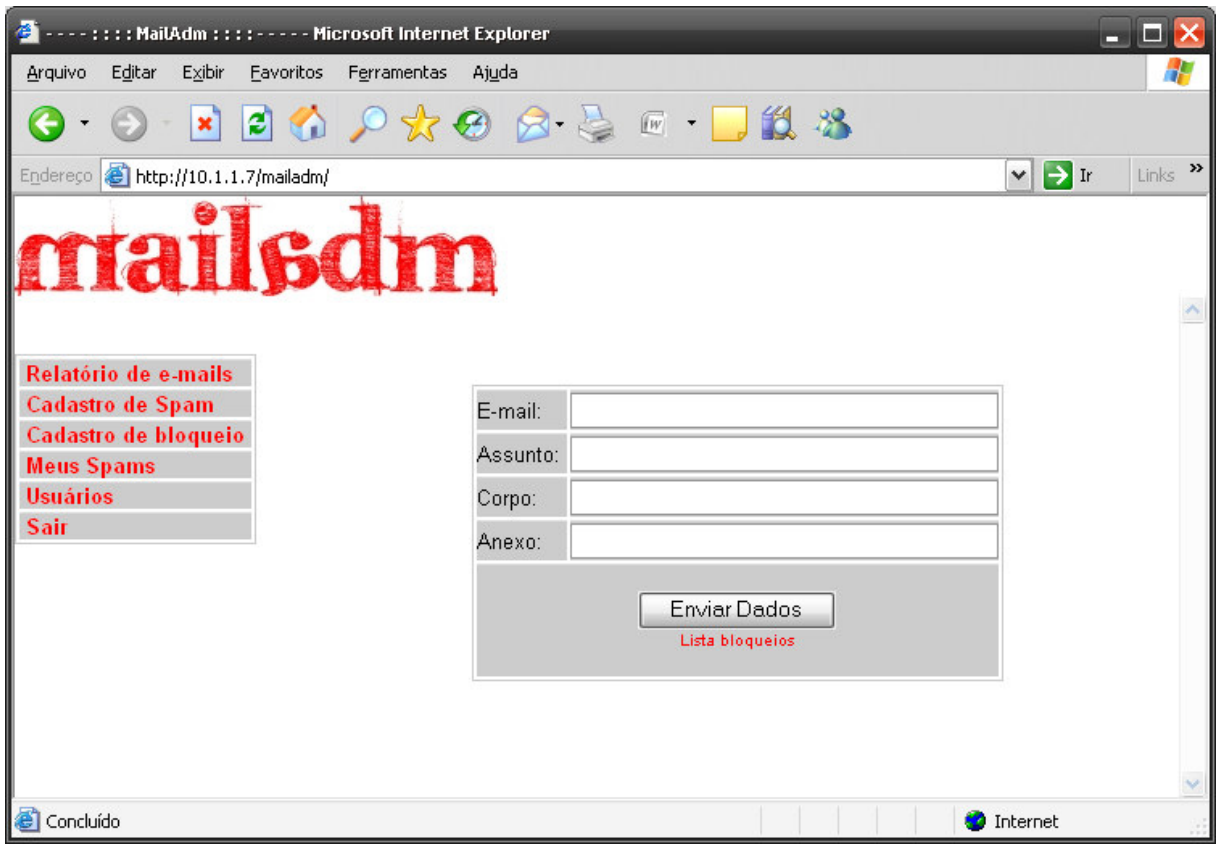

Figura 19: Formulário de cadastro de bloqueios.

Após o cadastro de e-mails, assuntos, corpos e anexos que serão classificados como spam ou bloqueados pelo sistema, o usuário de nível “adm” poderá listar estas definições cadastras, conforme apresentadas nas figuras 20 e 21.

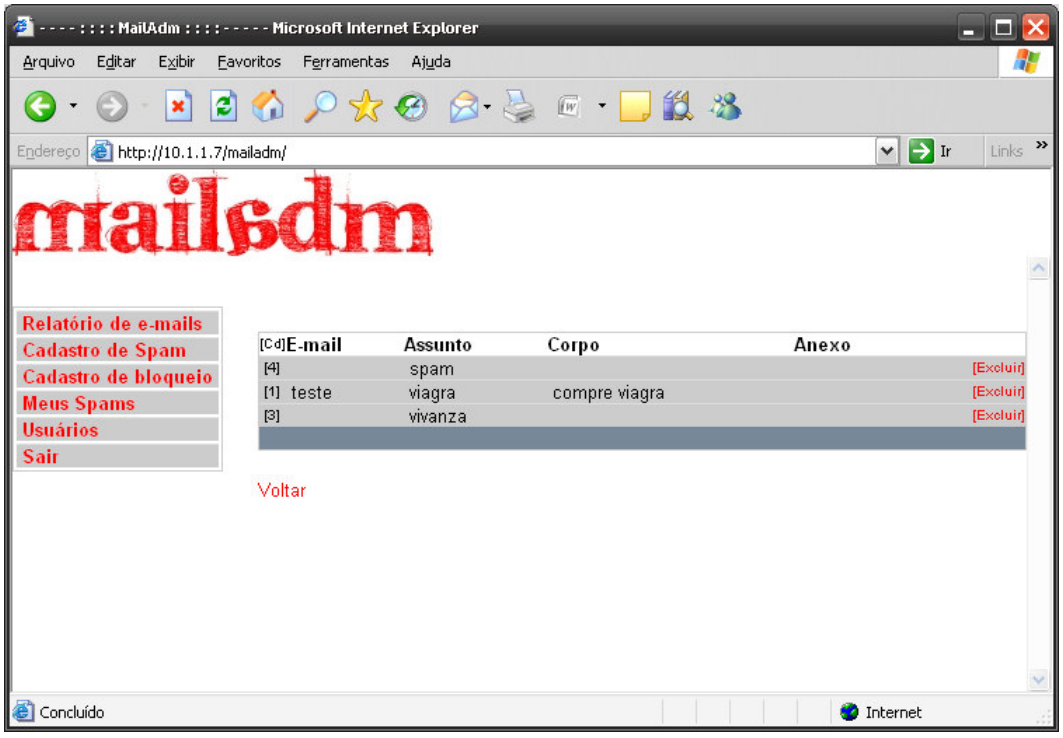

Figura 20: Lista de *spams* cadastrados no sistema.

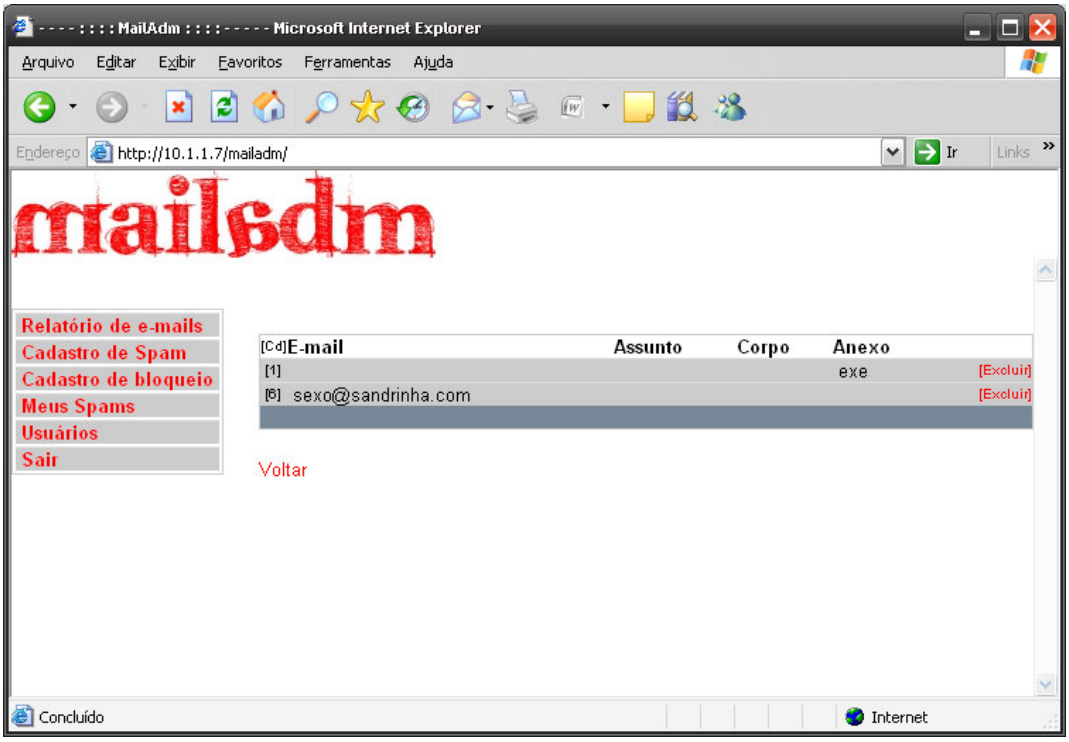

Figura 21: Lista de bloqueios cadastrados no sistema.

No software também deverão ser cadastrados os usuários que deverão operá-lo, tanto para administrar quanto apenas para a verificação de *spams* de cada usuário. Os usuários necessariamente deverão ser pré-cadastrados no servidor de e-mail e com o mesmo usuário no sistema de anti-*spam*. Se o usuário possui o e-mail como exemplo. "guilherme@guieberhardt.com", este deverá ser cadastrado no sistema como o usuário "guilherme". Na figura 22 é mostrada a tela de cadastro de usuários e na figura 23 é mostrada a lista de usuários do sistema.

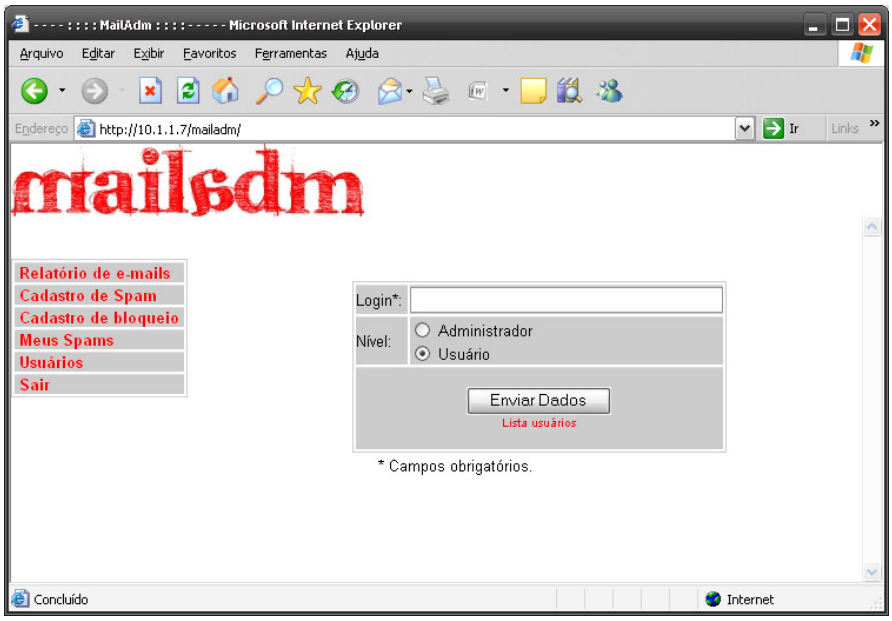

Figura 22: Formulário de cadastro de usuários.

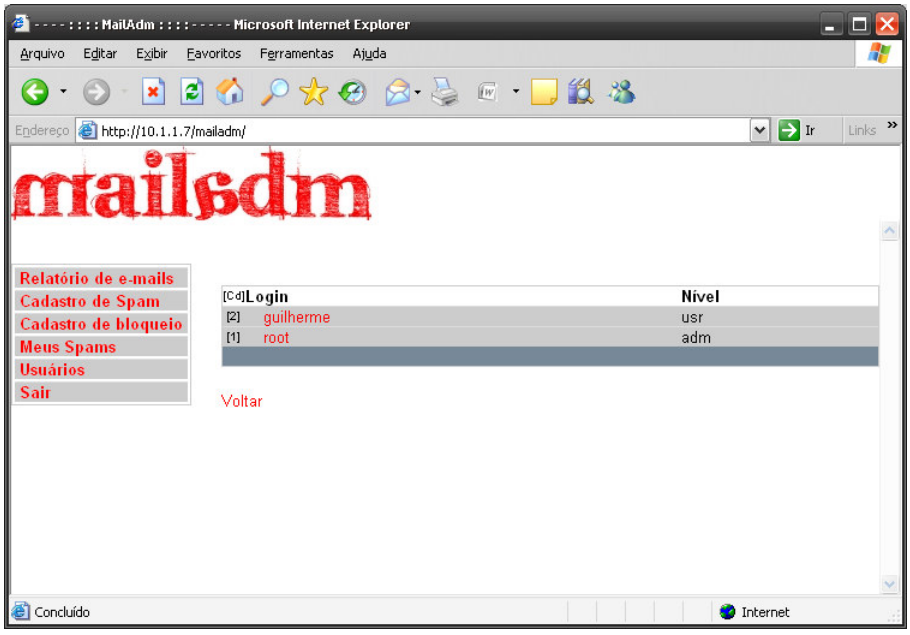

Figura 23: Lista de usuários cadastrados no sistema.

Cada usuário do sistema, sendo ele do nível “adm” ou “usr”, poderá visualizar seus e-mails que foram classificados como *spam* pelo servidor, e assim podendo liberar para ser recebido em sua caixa postal ou não. A lista de e-mails é mostrada por ordem de chegada do e-mail, conforme apresentada na figura 24, e clicando sobre a mensagem é exibido o seu conteúdo.

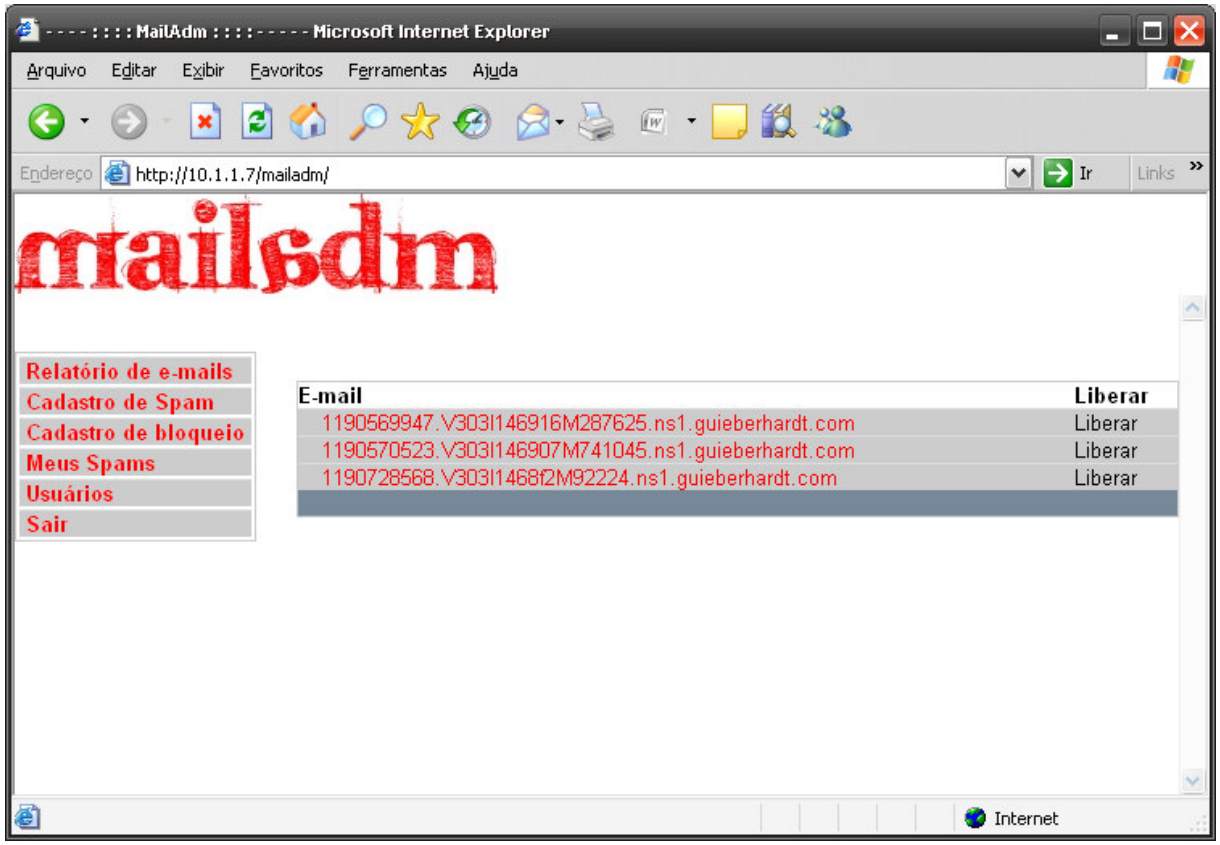

Figura 24: Lista de spams do usuário definidos pelo sistema.

3.4 RESULTADOS E DISCUSSÃO

Os resultados obtidos através do trabalho foram muito satisfatórios nos quesitos administração de um anti-*spam* e controle de e-mails em um servidor via *web*. Não foram encontrados muitos exemplos e trabalhos que pudessem auxiliar na confecção deste trabalho.

Santos (2006) elaborou sua monografia voltada a programas e técnicas contra *spam*. Diante do grande crescimento de envio de *e-mails* não autorizados pelo usuário, caso de roubo

de informação, envio de vírus e *trojans*, fez uma análise de mecanismos para combater *spam*, através de configuração, de filtros ou de programas de proteção.

Dalazen (2005), professor da UnB, publicou uma matéria referente ao controle de e-mail nas empresas, tendo como tema principal a dúvida se o empregador pode monitorar o e-mail do funcionário ou não.

O gerenciamento de e-mails está crescendo e se tornando cada vez mais complexo devido ao grande tráfego e o uso do mesmo. Além dos e-mails indesejados e não solicitados conhecidos como *spam*.

Os resultados obtidos conseguiram atingir todos os objetivos iniciais do trabalho, quer seja possibilitar o gerenciamento de e-mails, entrada e saída de um servidor, e ainda criar filtros de bloqueios de e-mails que não poderão ser encaminhados aos usuários e filtros de e-mails designados como *spam*.

Um dos resultados obtidos através do trabalho foi que o administrador do servidor de e-mail não tem mais a necessidade de cadastrar manualmente as regras de palavras de bloqueio dos e-mails e *spam* de forma manual, através de um terminal a caractere, podendo agora efetuar estes cadastros via *browser*, utilizando uma interface gráfica.

Outra facilidade foi a liberação de e-mails reconhecidos como *spam* pelo próprio usuário, poupando tempo do administrador que antes gerenciava estes *spams* de todos os usuários.

O relatório de todos os e-mails que entram e saem do servidor foi outra facilidade implantada para a administração dos e-mails. Assim, o administrador pode também visualizar via *browser* todos estes e-mails e acompanhar a quantidade de e-mails recebidos por usuário, e não mais via um terminal a caractere.

4 CONCLUSÕES

As atuais discussões sobre um melhor gerenciamento e um menor custo operacional de servidores *web* vêm gerando ótimos aplicativos para os administradores de servidores no ambiente Internet.

Os servidores de e-mails próprios estão se tornando cada vez mais comuns dentro das empresas, e por isso o seu o gerenciamento vem crescendo, gerando a necessidade de criação de ferramentas próprias para tal função. Um grande percentual dos servidores hoje são Linux devido ao custo de licença que na maioria dos casos é zero, embora a administração seja mais complexa, necessitando assim de ferramentas colaborativas para a tarefa.

O presente estudo pretende colaborar justamente com o conceito abordado acima, construindo uma ferramenta responsável pelo gerenciamento de e-mails, gerando lista de spams, e-mails bloqueados e ainda relatórios de e-mails de entrada e saída da empresa.

O trabalho desenvolvido pretende colaborar ainda mais com esse mercado de servidores Linux, que está em constante expansão. Atualmente o mercado tem a necessidade de melhoria na facilidade de administração, e facilidade de acesso aos e-mails.

O trabalho foi concluído com sucesso, atingindo todos os seus objetivos iniciais, com o desenvolvimento de uma ferramenta de coleta de e-mails integrada ao Postfix, desenvolvimento de uma ferramenta de anti-*spam* e bloqueio de mensagens não permitidas e desenvolvimento de uma ferramenta que gere relatórios de e-mails que entram e saem de uma organização. A sua implantação em um servidor Linux acabou acontecendo em um servidor de laboratório sendo adquirido um domínio de internet para testes, além de ambiente de produção para teste em maior escala de spams, e-mails bloqueados e na performance da emissão de relatório de e-mails que entram e saem da empresa.

4.1 EXTENSÕES

Devido às atualizações que irão sempre ocorrer nos servidores de e-mail utilizado neste trabalho, o Postfix, o trabalho terá como extensão a atualização e adaptação para estas atualizações, fazendo com que o software funcione no ambiente antigo e novo.

Outra sugestão para possível continuidade seria o desenvolvimento de módulos para o

software que administra regras de *firewall* via *browser*, facilitando ainda mais a administração de um servidor de Internet. Hoje esta administração de regras de firewall também é feita, na maioria dos casos, via terminal a caractere, o que torna mais difícil visualizar.

Outra extensão que pode ser aplicada seria a inteligência artificial no controle de spam. O software de anti-*spam* desenvolvido neste trabalho compara apenas palavras definidas pelo administrador e consultadas no banco de dados para definir se o e-mail é bloqueado e/ou *spam.* Desta forma ao aplicar a inteligência artificial, o sistema identificará sozinho os possíveis novos e-mails, assuntos, corpo e anexo que definirá como spam e/ou que deverá ser bloqueado pelo sistema para que não chegue ao usuário.

REFERÊNCIAS BIBLIOGRÁFICAS

AGUILERA, Ricardo. **Como atuam os spammers**. Outubro 2006,[]?. Disponível em: < http://informatica.terra.com.br/virusecia/spam/interna/0,,OI195563-EI2403,00.html> Acesso em: 20 Setembro 2007.

CARNEIRO, Márcio Rodrigo de Freitas. **Treinamento em UNIX(Linux/Solaris) para usuários do IME**. 2000. 76 p. Monografia – Instituto de Matemática e Estatística da Universidade de São Paulo, São Paulo, 2000.

CERT. **Cartilha de Segurança para internet**. Outubro 2006. Disponível em: <http://cartilha.cert.br> Acesso em: 12 Maio 2007.

DALAZEN, João Oreste. **E-mail: o empregador pode monitorar?** Folha de S. Paulo. Caderno A, 17/06/05, p.3, col.1-3.. Disponível em: <http://www.bc.furb.br/docs/JO/05/10/294252_1_1.pdf>. Acesso em: 16 abr. 2007.

LEVINE, John R. **E-mail para leigos**. São Paulo: Berkeley, 1997. 382 p.

MARCELO, Antônio. **Postfix**. Rio de Janeiro: Brasport, 2004. 108 p.

MICHELLIS, Deives. **Introducao ao Mundo do Postfix**. Maio 2004. Disponível em: < http://www.unitednerds.org/thefallen/docs/index.php?area=Postfix> Acesso em: 20 Novembro de 2007.

SANTOS, Cleiton Clóvis dos. **Programas e Técnicas de Proteção contra SPAM**. 2006. 111 p. Monografia - Universidade Regional de Blumenau, Blumenau, 2006.

TEIXEIRA, Renata Cicilini. **Combatendo o Spam**. São Paulo: Novatec, 2004. 170 p.

WALL, Larry. **Programação Perl**. Rio De Janeiro: Campus, 2001. 1084 p.