

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC**

**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**RAFAEL SPILERE MARANGONI**

**FILTERING MAIL: FERRAMENTA PARA FILTRAGEM DE E-MAIL  
CAPTURADO ATRAVÉS DO PROTOCOLO DE COMUNICAÇÃO IMAP4**

**CRICIÚMA  
2012**

**RAFAEL SPILERE MARANGONI**

**FILTERING MAIL: FERRAMENTA PARA FILTRAGEM DE E-MAIL  
CAPTURADO ATRAVÉS DO PROTOCOLO DE COMUNICAÇÃO IMAP4**

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de graduação do curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: M.Sc. Rogério Antônio Casagrande

**CRICIÚMA  
2012**

**RAFAEL SPILERE MARANGONI**

**FILTERING MAIL: FERRAMENTA PARA FILTRAGEM DE E-MAIL  
CAPTURADO ATRAVÉS DO PROTOCOLO DE COMUNICAÇÃO IMAP4**

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Redes de Computadores.

Criciúma, 26 de junho de 2012.

**BANCA EXAMINADORA**

Prof. M.Sc. Rogério Antônio Casagrande (UNESC) - Orientador

Prof. M.Sc. Paulo João Martins (UNESC)

Prof. Esp. Fabrício Giordani (UNESC)

**Dedico este trabalho aos meus pais, por me  
incentivarem a nunca desistir. A eles o meu  
Muito Obrigado!**

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por estar sempre vigiando os meus passos em quaisquer circunstâncias da vida.

Aos meus pais Aldenir e Teresa, que sempre me apoiaram em todas as decisões da vida, mostrando o melhor caminho a ser seguido. Agradeço imensamente o modo como fui educado e os ensinamentos necessários para obter sucesso em todos os obstáculos que vieram e ainda estão por vir. A minha irmã Vitória, pelas suas demonstrações de amor e carinho.

A minha namorada Luize, por me incentivar a não desistir, por entender os momentos de cansaço e principalmente por estar ao meu lado todo este tempo.

As pessoas especiais que fazem parte da minha rotina diária, por estarem sempre ao meu lado me ajudando a ser uma pessoa melhor a cada dia, principalmente ao meu colega de trabalho Matheus Leandro Ferreira que não mediu esforços para me auxiliar no desenvolvimento deste protótipo.

Aos meus colegas de curso, pelos momentos de convívio e amizade durante este período de troca de conhecimentos.

Aos professores e coordenadores do curso de Ciência da Computação, pelo ensinamento e calma durante estes anos, em especial ao meu professor e orientador Rogério Antônio Casagrande, sempre a disposição para repassar o seu conhecimento de grande educador.

**“Seja você quem for, seja qual for a posição social que você tenha na vida, a mais alta ou a mais baixa, tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá, de alguma maneira você chega lá.”**

**Ayrton Senna**

## RESUMO

Servidores Proxy são computadores intermediários que se posicionam entre o computador do usuário e a Internet. Geralmente chamados de *gateway*, podem ser disponibilizados no computador local do usuário ou em pontos estratégicos entre o usuário e o servidor de destino. São utilizados como bloqueadores de informações ou como armazenadores de memória cache para melhor desempenho na busca de informações já resgatadas. Um servidor proxy possibilita, dentre suas diversas funcionalidades, o bloqueio de e-mails recuperados em uma caixa de correio eletrônico, permitindo maior segurança e confiança nos dados recuperados por parte do usuário. Para se comportar desta forma, um servidor proxy precisa saber quais informações serão bloqueadas, permitindo que o usuário realize estes cadastrados conforme sua necessidade. Este trabalho descreve o desenvolvimento de um protótipo de filtragem de e-mail usando o servidor de e-mail IMAP. Com a pretensão de auxiliar o usuário na segurança das informações recebidas, o servidor proxy pode ser executado juntamente com o cliente de e-mail Thunderbird, trabalhando de forma consistente e dinâmica, conforme disponibilização do protocolo IMAP. O protótipo também provê fácil operabilidade na configuração dos filtros, usando uma *interface* simples e objetiva, a fim de facilitar a operação do usuário. O protótipo trabalha com múltiplos usuários conectados simultaneamente devido às conexões feitas pelo Thunderbird, onde cria uma nova conexão para cada pasta contida na caixa de correio do usuário, garantindo uma troca de mensagens automática para cada uma destas pastas. Todos os comandos enviados e recebidos pelo cliente de e-mail e servidor de e-mail podem ser vistos no *Servidor* da aplicação, permitindo ao administrador de uma rede local ou até mesmo a um usuário comum que esteja utilizando o protótipo, uma depuração caso ocorra qualquer problema durante a comunicação estabelecida entre ambos. Este protótipo foi desenvolvido com base em diversos testes relacionados à troca de informações entre cliente-servidor.

**Palavras-Chave:** Servidor Proxy; Correio Eletrônico; Protocolo de comunicação IMAP; Thunderbird.

## ABSTRACT

Proxy servers are computers that are positioned intermediate between the user's computer and the Internet. Often called the gateway can be made available on the user's local computer or at strategic points between the user and the destination server. Blockers are used as information or as hosts of cache for better performance in search of information already redeemed. A proxy server provides, among its many features, blocking e-mails recovered in an e-mail box, allowing for greater security and confidence in the data retrieved by the user. To behave in this way, a proxy server needs to know what information will be blocked, allowing the user to perform these registrations as needed. This paper describes the development of an e-mail filtering prototype using the IMAP e-mail server. With the pretension of help the user with security of received information, the proxy server can be executed together the e-mail client Thunderbird, working in a consistent and dinamic way, according to IMAP protocol availability. The prototype also provides easy operability in the filter configuration, using a simple and objective interface to facilitate user operation. The prototype works with multiple connected users simultaneously due to connections made by Thunderbird, where creates a new connection to each folder contained in the user mail box, guaranteeing an automatic message exchange to each single folder. All commands sent and received by e-mail client and e-mail server can be seen in the Application Server, allowing the local network administrator or even a common user that is using the prototype, a depuration if any problem occurs during the communication established between both. This prototype was developed with base on various tests related to information exchange between client-server.

**Keywords:** Proxy Server; E-mail; Communication Protocol IMAP; Thunderbird.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Distribuição das camadas do modelo TCP/IP. ....	20
Figura 2 – Exemplo de troca de mensagens na arquitetura cliente e servidor.....	23
Figura 3 – Servidor de correio corporativo .....	28
Figura 4 – Recuperando um e-mail através do protocolo IMAP4.....	34
Figura 5 – Tela principal do Microsoft Outlook Express.....	39
Figura 6 – Tela principal do Mozilla Thunderbird. ....	40
Figura 7 – Diagrama de casos de uso do protótipo.....	44
Figura 8 – Diagrama de atividades do protótipo. ....	45
Figura 9 – Criação do <i>ServerSocket</i> para o servidor proxy. ....	48
Figura 10 – Criação do <i>Socket</i> para conexão no servidor de e-mail IMAP Terra. ....	48
Figura 11 – Código que identifica se o e-mail está na BlackList ou WhiteList.....	49
Figura 12 - Tela principal de configuração da aplicação .....	52
Figura 13 – Configuração de um domínio na BlackList. ....	53
Figura 14 – Troca de mensagens entre cliente de e-mail Thunderbird e servidor IMAP, passando pelo servidor proxy. ....	54
Figura 15 – E-mail bloqueado pela BlackList do servidor proxy.....	55
Figura 16 – Utilização de um Servidor Proxy em uma rede local.....	56
Figura 17 – Base de testes da aplicação. ....	57
Figura 18 – Ferramenta JVisualVM em execução. ....	60

## LISTA DE TABELAS

Tabela 1 - Primitivas de <i>Socket</i> para o protocolo TCP.....	25
Tabela 2 - Serviços relacionados com sua porta e transporte.....	27
Tabela 3 - Campos do cabeçalho na RFC 822 ligados ao transporte de mensagens.....	30
Tabela 4 - Alguns campos utilizados no cabeçalho da mensagem RFC 822.....	30
Tabela 5 - Alguns comandos utilizados pelo servidor SMTP.....	31

## LISTA DE ABREVIATURAS E SIGLAS

ASCII	American Standard Code for Information Interchange
DNS	Domain Name System
E-MAIL	Eletronic Mail
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
IMAP	Internet Message Access Protocol
LAN	Local Area Network
MAN	Metropolitan Area Network
OSI	Open Systems Interconnection
POP3	Post Office Protocol Release 3
RFC	Request For Comments
SMTP	Simple Mail Transfer Protocol
SPAM	Spiced Ham
TCP	Transport Control Protocol
UDP	User Datagram Protocol
WAN	Wide Area Network
WWW	World Wide Web

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>13</b>
1.1 OBJETIVO GERAL.....	14
1.2 OBJETIVOS ESPECÍFICOS .....	14
1.3 JUSTIFICATIVA .....	14
1.4 ESTRUTURA DO TRABALHO .....	16
<b>2 REDES DE COMPUTADORES.....</b>	<b>17</b>
2.1 TCP/IP .....	18
<b>2.1.1 Internet .....</b>	<b>19</b>
<b>2.1.2 Arquitetura do TCP/IP .....</b>	<b>20</b>
<b>2.1.3 Transmission Control Protocol .....</b>	<b>21</b>
<b>2.1.4 Internet Protocol.....</b>	<b>21</b>
2.2 MODELO CLIENTE-SERVIDOR .....	22
2.3 SOCKETS .....	23
<b>2.3.1 Primitivas .....</b>	<b>24</b>
2.4 SERVIÇOS DE REDE.....	25
<b>3 CORREIO ELETRÔNICO .....</b>	<b>28</b>
3.1 MAIL USER AGENT, MAIL TRANSPORT AGENT e MAIL DELIVERY AGENT ...	28
3.2 PROTOCOLOS E DEFINIÇÕES .....	29
<b>3.2.1 Request For Comments 822.....</b>	<b>29</b>
<b>3.2.2 Simple Mail Transfer Protocol.....</b>	<b>30</b>
<b>3.2.3 Post Office Protocol Release 3 .....</b>	<b>32</b>
<b>3.2.4 Internet Message Access Protocol Version 4.....</b>	<b>33</b>
3.2.4.1 Exemplos de Problemas e as Soluções IMAP4.....	34
3.2.4.2 Respostas e Comandos do Protocolo IMAP4.....	35
<b>4 TRABALHOS CORRELATOS .....</b>	<b>37</b>
4.1 ENVIANDO E RECEBENDO MENSAGENS ATRAVÉS DOS PROTOCOLOS SMTP E POP3 .....	37
4.2 PROTÓTIPO DE SOFTWARE PARA NOTIFICAÇÃO DE RECEBIMENTO DE NOVOS E-MAIL'S, BASEADO NO PROTOCOLO IMAP .....	37
4.3 ADVANCED MAIL: FERRAMENTA DE ACESSO AVANÇADO A CONTAS DE E- MAIL BASEADO NO PROTOCOLO POP3 .....	37
4.4 FERRAMENTAS EXISTENTES .....	38
<b>4.4.1 Microsoft Outlook Express.....</b>	<b>38</b>

<b>4.4.2 Mozilla Thunderbird.....</b>	<b>39</b>
<b>5 SISTEMA PROPOSTO .....</b>	<b>41</b>
5.1 METODOLOGIA.....	41
5.2 RECURSOS .....	41
<b>6 TRABALHO DESENVOLVIDO .....</b>	<b>43</b>
6.1 MODELAGEM .....	43
<b>6.1.1 Fases do Desenvolvimento de um Sistema em UML .....</b>	<b>43</b>
6.1.1.1 Diagrama de Casos de Uso .....	44
6.1.1.2 Diagrama de Atividades .....	45
6.2 AMBIENTE DE DESENVOLVIMENTO.....	45
6.3 DESENVOLVIMENTO DO PROTÓTIPO .....	47
6.4 UTILIZAÇÃO DO PROTÓTIPO .....	50
<b>6.4.1 Configurando o Sistema.....</b>	<b>50</b>
<b>6.4.2 Identificando as Mensagens Apresentadas no Servidor Proxy .....</b>	<b>53</b>
<b>6.4.3 Como Utilizar o Servidor Proxy em uma Rede Local.....</b>	<b>56</b>
6.5 EXECUÇÃO DOS TESTES E RESULTADOS.....	57
<b>6.5.1 Ferramentas para Depuração de Problemas no Desenvolvimento do Protótipo.....</b>	<b>59</b>
6.5.1.1 JVisualVM.....	60
6.5.1.2 Console Depurador da IDE Eclipse.....	60
<b>7 CONCLUSÃO.....</b>	<b>62</b>
<b>REFERÊNCIAS .....</b>	<b>64</b>
<b>APÊNDICE A – ARTIGO .....</b>	<b>68</b>

## 1 INTRODUÇÃO

Os computadores, inicialmente, eram máquinas caríssimas que centralizavam o processamento de dados em um único ponto. Após a redução de custos do hardware e inclusão dos microcomputadores no cenário da informática, a estrutura passou a ser distribuída e não mais centralizada (SOARES, 1995). Com a distribuição destes recursos aumentou-se os equipamentos, gerando duplicação desnecessária de recursos de hardware e software, melhorando o processamento de dados, mas ocupando maior espaço físico. Foi nesta deficiência que surgiram as redes de computadores, interligando estes equipamentos de processamento de dados com a finalidade de compartilhamento de recursos. Hoje o maior exemplo de conectividade através de rede de computadores é a Internet que é um amplo sistema de comunicação que conecta muitas redes de computadores, nomeada núcleo da comunicação moderna (KUROSE, 2003).

O sistema de acesso a banda larga tem sido bastante popularizado no Brasil, fazendo com que a maioria dos cidadãos brasileiros já tenha acesso à Internet. Juntamente com a Internet, caminham diversos programas e sistemas que facilitam a vida dos usuários. Dentre eles temos o correio eletrônico, muito difundido antes mesmo do surgimento desta tecnologia. Entretanto, foi após a criação desta ferramenta que os correios eletrônicos começaram a ser utilizados, devido a sua rapidez se comparado com o correio tradicional e ao seu custo reduzido se comparado a uma chamada telefônica nacional ou internacional (MCFEDRIES, 2008).

O usuário que utiliza um cliente de e-mail sob o protocolo Internet Message Access Protocol Version 4 (IMAP4) tem algumas vantagens perante os demais protocolos de recebimento de e-mail, porém não tem o controle específico sobre determinadas funções, como por exemplo, não receber e-mails de determinado *domínio*<sup>1</sup>.

Neste trabalho foi desenvolvida uma ferramenta que permite ao usuário manipular e gerenciar as suas mensagens, fazendo um filtro para que identifique quais os e-mails que o usuário deseja realmente receber, descartando os demais.

---

<sup>1</sup> Realiza a função de relacionar nomes a endereços IP (Internet Protocol).

## 1.1 OBJETIVO GERAL

Desenvolver uma ferramenta para filtragem de e-mail capturado através do protocolo de comunicação IMAP4.

## 1.2 OBJETIVOS ESPECÍFICOS

Os objetivos deste trabalho são:

- a) estudar a funcionalidade das Redes de Computadores;
- b) estudar o protocolo de comunicação IMAP4 a fim de utilizá-lo como base na comunicação entre o cliente x servidor Proxy x servidor de e-mail;
- c) estudar o funcionamento e arquiteturas do protocolo TCP/IP;
- d) estudar e utilizar a comunicação de redes de computadores utilizando *Sockets* da linguagem de programação Java;
- e) compreender e estudar o funcionamento das *Thread's* na linguagem de programação Java;
- f) desenvolver o estudo de como a ferramenta filtrará os e-mails contidos no correio-eletrônico do usuário;
- g) implementar a ferramenta.

## 1.3 JUSTIFICATIVA

A Internet evoluiu tão rapidamente que os e-mails tornaram-se a identidade eletrônica do usuário, e não somente o seu correio eletrônico (COMER, 1998). No início, o e-mail teve função de trocar mensagens simples entre usuários, sem arquivos anexados ou codificações na mensagem. Conforme a sua evolução, o e-mail começou a ser visto, de fato, como um correio eletrônico.

Fala-se muito hoje na extinção do e-mail, devido à grande quantidade de redes sociais e formas de comunicação na Internet (GERAÇÃO..., 2008). Porém deve-se lembrar que para se criar um perfil em uma rede social ou para conseguir trocar mensagens simultâneas no seu *MSN Messenger (Microsoft Service Network Messenger)* é preciso de um e-mail. Uma pesquisa recente apontou que o e-mail é ainda a ferramenta mais utilizada na Internet, sendo 77% dos usuários da Internet, utilizadores deste serviço (E-MAIL..., 2011).

O correio eletrônico se tornou muito utilizado devido a sua rapidez se comparado ao correio tradicional e devido também, ao seu custo reduzido se comparado a uma chamada telefônica. Ou seja, o custo de uma mensagem de correio eletrônico para uma pessoa que está em outro país é o mesmo de uma mensagem para uma pessoa que está na mesma cidade do remetente.

O sistema de acesso em banda larga tem sido bastante popularizado no Brasil. Como exemplo disto, existe hoje um projeto do Ministério das Comunicações em conjunto com algumas operadoras de telefonia fixa, chamado Plano Nacional de Banda Larga (PNBL), que fará uma oferta de banda larga à R\$35,00 mensais, com velocidade de 1 Mbps (megabit por segundo). As primeiras localidades a receber o plano, serão no centro-oeste do Brasil, expandindo-se para todo o país até 2014 (VENDA..., 2011).

Para utilização do correio eletrônico, foi estudada e implementada a troca de mensagens conforme o protocolo IMAP4, no qual a principal vantagem é a economia de tempo para baixar os e-mails, pois baixa apenas os cabeçalhos das mensagens de maneira que possa escolher, por exemplo, quais mensagens deseja ler mais tarde e quais deseja exibir na sua totalidade. A sua sincronização permite que várias pessoas utilizem uma caixa de correio compartilhada.

A Internet traz muitos benefícios, mas podem trazer também problemas devido ao mau uso desta importante ferramenta. Uma das vantagens que esta pesquisa proporcionou é a organização do recebimento dos e-mails, fazendo um filtro para cada e-mail recebido, a fim de evitar mensagens indesejadas e lixos eletrônicos. Para a filtragem, configurar-se-á na ferramenta uma lista branca ou negra a fim de permitir ou bloquear, respectivamente, os e-mails recebidos do servidor. A permissão ou bloqueio poderá ser feito por usuário ou por domínio.

Para implementação da ferramenta utilizar-se-á a linguagem de programação Java, através do mecanismo de comunicação entre aplicações chamado *socket* (MITCHELL, 2001). O modo de utilização do *Socket* será orientado a conexão, que funciona sobre o protocolo TCP/IP, de forma a criar um *cliente* para conexão com o servidor IMAP4 dos provedores de e-mail.

Quase tudo na vida funciona como uma faca: “um lado alisa e o outro corta”. Porém, o progresso e a modernidade tecnológica têm mais vantagens do que desvantagens. Cabe ao usuário a escolha de como utilizar os aspectos positivos da Internet, que chegou para mudar tudo. O correio eletrônico é um exemplo desta mudança devido à sua facilidade de uso

e ao amplo poder de troca de informações que permite através de uma simples troca de mensagens.

#### 1.4 ESTRUTURA DO TRABALHO

Este trabalho é composto por seis capítulos e a Conclusão: Introdução, Redes de Computadores, Correio Eletrônico, Trabalhos Relacionados, Sistema Proposto e Trabalho Desenvolvido. Cada capítulo faz uma abordagem objetiva dos temas relacionados a pesquisa e desenvolvimento do protótipo.

O primeiro capítulo deste trabalho é constituído de objetivo geral, objetivos específicos, justificativa e estrutura do trabalho, onde são mostrados de forma resumida os pontos principais da pesquisa e uma breve descrição do funcionamento do protótipo.

No capítulo seguinte é feita a abordagem sobre as redes de computadores, sua classificação de acordo com o seu tamanho, o estudo feito sobre a família de protocolos TCP/IP, o modelo de organização de rede conhecido como cliente-servidor, a abstração de *sockets* e os serviços de rede em torno da Internet global.

No terceiro capítulo são mostrados os protocolos de comunicação utilizados pelo correio eletrônico, tais como suas funcionalidades, facilidades e acessibilidade. Também são apresentadas as tag's que são utilizadas para o recebimento de um e-mail, incluindo sua utilização e descrição.

No quarto capítulo são apresentados trabalhos utilizados como base de pesquisa e que estão relacionados ao tema proposto por este projeto.

No quinto capítulo, explica-se como a ferramenta pretende ser desenvolvida e elaborada a partir dos conceitos estudados.

No capítulo relacionado ao trabalho desenvolvido, serão apontadas ferramentas que auxiliaram no desenvolvimento do protótipo, partes do código-fonte que são relevantes para o entendimento das principais funcionalidades da ferramenta desenvolvida, telas do protótipo mostrando a configuração do sistema e fluxo das trocas de mensagem entre cliente de e-mail, servidor proxy e servidor de e-mail e por fim, os resultados obtidos e testes relativos à aplicação.

No último capítulo foram descritos os principais objetivos alcançados e a conclusão do trabalho de pesquisa, citando os passos para o sucesso do protótipo.

## 2 REDES DE COMPUTADORES

Uma rede de computadores é um grupo de computadores com funcionamentos independentes um do outro e interconectados por cabos de rede. A rede permite o compartilhamento de softwares, informações, arquivos e demais serviços.

A rede pode atuar em computadores, periféricos (impressoras, scanners) e em máquinas de linha de produção. Reduz o custo com a aquisição de softwares, minimiza processos de envio de projetos, significa economia de tempo e melhor desempenho operacional.

Conceitualmente se define uma rede como computadores interligados entre si e que possam compartilhar seus dados. Toda e qualquer rede, independentemente de tamanho e sofisticação, descende desse modelo básico e não restrito somente a microcomputadores (TANENBAUM, 1994).

As redes são classificadas conforme o seu tamanho. As mais comuns são:

- a) Rede Local – *Local Area Network* (LAN) ocupa uma área limitada de no máximo um edifício ou alguns edifícios próximos, podendo estar até em uma única sala;
- b) Rede Metropolitana – *Metropolitan Area Network* (MAN) pode assegurar a ligação de redes locais, sendo a versão amplificada de uma LAN. Uma MAN chega a atingir uma cidade inteira e pode ser pública ou privada;
- c) Rede de Longa Distância – *Wide Area Network* (WAN) têm a dimensão correspondente a países, continentes ou vários continentes. São constituídas por inúmeras redes interligadas, por exemplo, LANs e WANs. O exemplo mais conhecido é a Internet;
- d) Rede Local Sem Fio – *Wireless Area Local Network* (WLAN) é adequado a situações em que é necessária mobilidade. São flexíveis e de fácil instalação;
- e) Rede Pessoal – *Personal Area Network* (PAN) está relacionada a sua reduzida dimensão e utilização da comunicação sem fio. O alcance é limitado a algumas dezenas de metros;
- f) Rede Local Virtual – *Virtual Local Area Network* (VLAN) é definida sobre redes locais que estão equipadas com dispositivos apropriados (dispositivos que suportem VLANs). Trata-se de definir até que zonas da LAN se propagam as emissões de broadcast que tem origem noutra zona. As VLANs não proporcionam qualquer segurança para a rede;

- g) Rede Ethernet – as redes Ethernet dominam as redes locais (LAN) e tem uma expressão nas redes metropolitanas (MAN). Atualmente a taxa de transmissão mais utilizada é de 100 Mbps (Fast Ethernet) e especialmente em MANs utiliza-se 1 Gbps (Gigabit Ethernet) (TANENBAUM, 2003).

Para que haja troca de informações entre as redes é necessária uma lógica de comunicação entre estas redes. Esta lógica é chamada de protocolo de rede. O conjunto de protocolos mais conhecido e difundido é o TCP/IP, que será apresentado no próximo sub-capítulo.

## 2.1 TCP/IP

Nome dado a todo o conjunto de protocolos utilizados pela Internet. Este conjunto de protocolos foi desenvolvido para permitir aos computadores compartilharem recursos dentro de uma rede. Estes protocolos incluem um conjunto de padrões que especificam os detalhes de como comunicar computadores, assim como também convenções para interconectar redes e rotear o tráfego (MURHAMMER, 2000, tradução nossa).

O desenvolvimento das diferentes arquiteturas de redes começou bem antes do que se imagina e, como a maioria das grandes invenções, o propósito inicial era o uso militar, ainda na época da Guerra Fria. Uma das principais prioridades dentro de uma força militar é a comunicação.

No final da década de 60, esta era uma grande preocupação do Departamento de Defesa do Exército Americano (DOD): como interligar computadores de arquiteturas completamente diferentes e que ainda por cima estavam muito distantes um do outro, ou mesmo em alto-mar, dentro de um porta-aviões ou submarino.

Após alguns anos de pesquisa, surgiu o TCP/IP, abreviação de *Transmission Control Protocol/Internet Protocol*, ou protocolo de controle de transmissão/protocolo Internet. O TPC/IP permitiu que as várias pequenas redes de computadores do exército Americano fossem interligadas, formando uma grande rede, embrião do que hoje conhecemos como Internet. O TCP/IP é composto de dois protocolos: o IP cuida do endereçamento, enquanto o TCP cuida da transmissão dos dados e correção de erros. O segredo do TCP/IP é dividir a grande rede em pequenas redes independentes, interligadas por roteadores. Como (apesar de interligadas) cada rede é independente da outra, caso uma das redes perca a conexão, apenas aquele segmento fica fora do ar, sem afetar a rede como um todo (TANENBAUM, 2000).

Apesar de inicialmente o uso ter sido restrito a aplicações militares, com o passar do tempo o protocolo acabou tornando-se de domínio público, o que permitiu aos fabricantes de software adicionar suporte ao TCP/IP aos seus sistemas operacionais de rede. Atualmente, este protocolo é suportado por todos os principais sistemas operacionais, não apenas os destinados a PCs, mas a praticamente todas as arquiteturas, incluindo até mesmo celulares e handhelds. Qualquer sistema com um mínimo de poder de processamento pode conectar-se à Internet, desde que alguém desenvolva uma implementação do TCP/IP para ele, juntamente com alguns aplicativos.

### 2.1.1 Internet

A Internet é uma gigantesca rede mundial que interliga computadores de todos os tipos e capacidades através de linhas de comunicação (telefone, canais de satélite, cabo submarino) utilizando um conjunto de protocolos específicos para computadores.

Para o mundo globalizado das redes não existem fronteiras, assim como também não há distinção de raça, sexo, cor ou nacionalidade entre pessoas.

Da década de setenta até o início da década de noventa ela era um reduto de universidades e órgãos governamentais, em que quatro aplicações detinham a sua atenção: Correio Eletrônico, *Newsgroups*, *Logon Remoto* e Transferência de Arquivos (TANENBAUM, 2003).

Não levou muito tempo para a Internet se difundir no mundo capitalista e tornar-se um meio comercial. Com o surgimento de novos navegadores, cada vez com mais recursos e mais amigáveis e somado ao surgimento dos provedores de acesso (Internet Service Providers - ISPs) em que era possível acessar a Internet até mesmo de casa, o número de usuários em pouco chegava a milhões de pessoas (TANENBAUM, 2003).

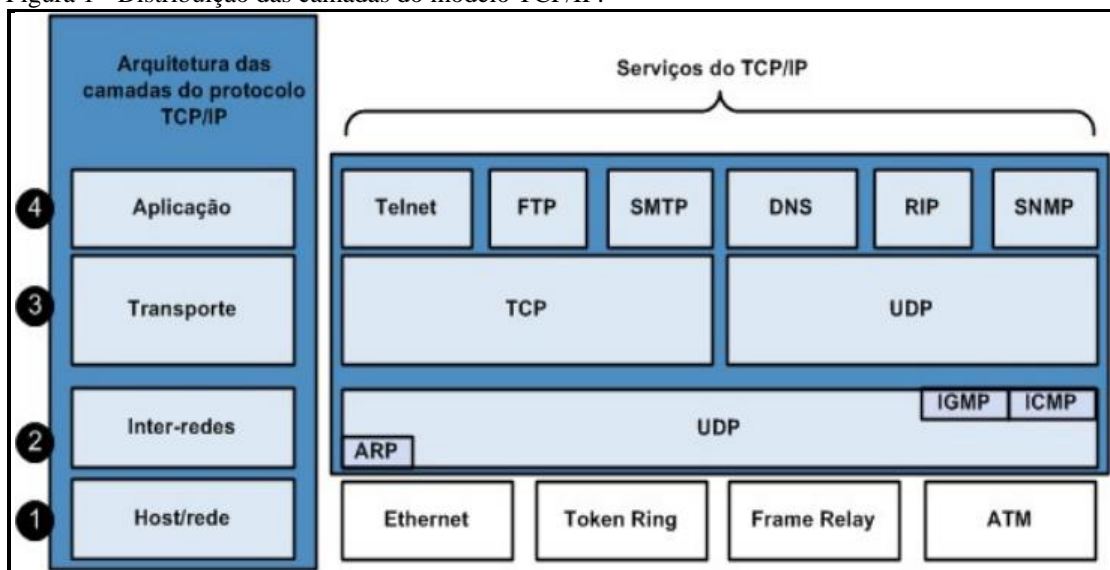
No mundo da Internet encontramos serviços e facilidades, notícias e atualidades, ou se preferir como é o caso de muitas pessoas, um excelente local para encontrar amigos, jogos, bate papo e muito mais, como lojas virtuais, filmes, quadros de museus, fotografias, pesquisas escolares e outros programas e arquivos, úteis ou não no seu dia-a-dia.

Não é de admirar que esse espaço virtual atraia desde homens de negócio em busca de bons lucros até curiosos à procura de diversão.

## 2.1.2 Arquitetura do TCP/IP

Todos os softwares de redes são baseados em alguma arquitetura de camadas. A arquitetura TCP/IP visualiza a rede de computadores em quatro camadas, incluindo nesta Figura 1 a camada de segurança.

Figura 1 - Distribuição das camadas do modelo TCP/IP.



Fonte: TANENBAUM (2003).

Segue uma descrição da função de cada camada do TCP/IP segundo Tanenbaum (2003):

- a) os protocolos físicos e de enlace têm a função de fazer com que informações sejam transmitidas de um computador para outro em uma mesma mídia de acesso compartilhado (também chamada de rede local) ou em uma ligação ponto-a-ponto (ex: modem). A preocupação destes protocolos é permitir o uso do meio físico que conecta os computadores na rede e fazer com que os bytes enviados por um computador cheguem a um outro computador diretamente desde que haja uma conexão direta entre eles;
- b) o protocolo de rede, Internet Protocol (IP), é responsável por fazer com que as informações enviadas por um computador cheguem a outros computadores mesmo que eles estejam em redes fisicamente distintas, ou seja, não existe conexão direta entre eles um destaque é a capacidade da

- rede TCP/IP se re-configurar quando uma parte da rede está fora do ar, procurando um caminho (rota) alternativo para a comunicação;
- c) os protocolos de transporte mudam o objetivo, que era conectar dois equipamentos, para conectar dois programas. Pode-se ter em um mesmo computador vários programas trabalhando com a rede simultaneamente, por exemplo, um browser Web e um leitor de e-mail. Da mesma forma, um mesmo computador pode estar executando ao mesmo tempo um servidor Web e um servidor POP3. Os protocolos de transporte (UDP e TCP) atribuem a cada programa um número de porta, que é anexado a cada pacote de modo que o TCP/IP saiba para qual programa entregar cada mensagem recebida pela rede.
  - d) Os protocolos de aplicação são específicos para cada programa que faz uso da rede. Desta forma existe um protocolo para a conversação entre um servidor web e um navegador web (HTTP), um protocolo para a conversação entre um cliente e servidor Telnet, e assim por diante.

### **2.1.3 Transmission Control Protocol**

O protocolo Transmission Control Protocol (TCP) foi criado com o intuito de realizar a intercomunicação de computadores. A configuração desse protocolo tem como função controlar como a informação é passada de uma rede a outra, e como manipular o endereçamento contido nos pacotes, a fragmentação dos dados e a checagem de erros.

O TCP é o protocolo da camada de transporte orientado à conexão, que oferece um serviço confiável. Frequentemente aparece como parte da pilha TCP/IP da arquitetura Internet, mas é um protocolo de propósito geral que pode ser adaptado para ser usado com uma variedade de sistemas.

### **2.1.4 Internet Protocol**

O Protocolo de Internet (IP) é um protocolo de comunicação usado entre duas ou mais máquinas de uma rede para encaminhamento dos dados.

Segundo Murhammer (2000) as funções básicas do IP são:

- a) fornecer o endereçamento para os hosts da rede;
- b) roteamento, escolhendo um caminho sobre o qual os dados serão enviados;

c) a unidade básica de dados a ser transferida na Internet.

O IP dá uma visão virtual da rede, sendo um protocolo de entrega de pacotes não confiável, ou seja, sem conexão, onde os pacotes enviados por ele podem ser perdidos ou chegar fora de ordem ao destinatário (TANENBAUM, 2003).

## 2.2 MODELO CLIENTE-SERVIDOR

A arquitetura cliente-servidor é hoje uma das arquiteturas mais utilizadas em ambientes corporativos, substituindo a arquitetura muito rígida que eram os sistemas envolvendo mainframes<sup>2</sup>.

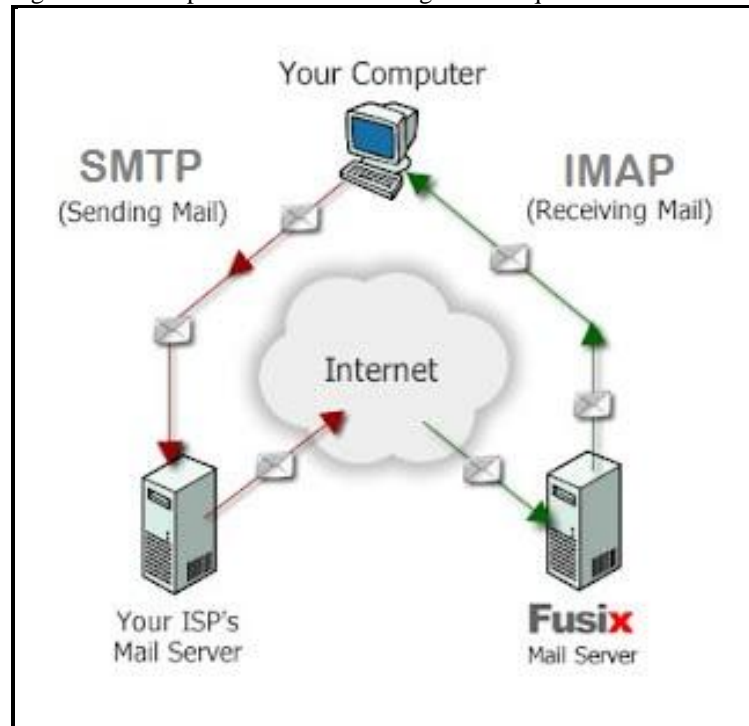
Esta tecnologia tem uma arquitetura na qual o processamento da informação é dividido em módulos ou processos distintos. Um processo é responsável pela manutenção da informação (servidores) e o outro responsável pela obtenção dos dados (clientes). Os processos *clientes* enviam pedidos para o processo *servidor* e este por sua vez processa e envia os resultados dos pedidos (BATTISTI, 2001).

Geralmente, os serviços oferecidos pelos servidores dependem de processamento específico que só eles podem fazer. O processo cliente, por sua vez, fica livre para realizar outros trabalhos. A interação entre os processos é uma troca cooperativa, como exemplificada na Figura 2, em que o cliente é o ativo e o servidor reativo, ou seja, o cliente requisita uma operação, o servidor processa e responde ao cliente.

---

<sup>2</sup> Computador de grande porte dedicado normalmente ao processamento de um grande volume de informações, capaz de oferecer serviços de processamento a milhares de usuários (CASAD; WILLSEY, 1999).

Figura 2 – Exemplo de troca de mensagens na arquitetura cliente e servidor.



Fonte: VASKEVITCH (1995).

Havendo um cliente, um servidor com serviços e um meio de comunicação entre eles, está estabelecida uma arquitetura Cliente/Servidor. O padrão desta arquitetura é utilizado em serviços de bancos de dados, aplicações de *groupware*<sup>3</sup>, Internet e outros.

### 2.3 SOCKETS

Pode ser usado em ligações de redes de computadores para um fim de um elo bidirecional de comunicação entre dois processos (MITCHELL, 2001, tradução nossa).

A interface padronizada de *Socket's* surgiu originalmente no sistema operacional Unix Berkeley Software Distribution (BSD), portanto, eles são muitas vezes chamados de Berkeley *Sockets*. É também uma abstração computacional que mapeia diretamente a uma porta de transporte (TCP ou UDP) e mais um endereço de rede. Com esse conceito é possível identificar unicamente um aplicativo ou servidor na rede de comunicação IP (HOLLINGWORTH, 2000, tradução nossa).

<sup>3</sup> Conjunto de ferramentas cooperativas que possibilitam a interação entre múltiplos participantes.

Em documentos de Request for Comments (RFC) relacionado à TCP ou UDP, esta comunicação é definida como a combinação de um endereço IP, um protocolo e o número da porta do protocolo.

*Sockets* e RFCs têm a função de implementar numa aplicação as funções de rede, mas fazem isto de maneiras diferentes.

Quando este objeto é criado, devem ser especificados pelo menos três parâmetros: tipo de comunicação, espaço de nomes e protocolo (COMER, 1998).

O tipo de comunicação controla como o ele fará a transmissão dos dados e especifica o número de participantes na comunicação. Quando um fluxo de dados é transmitido, estes são chamados de pacotes. O tipo de comunicação determina como esses pacotes são tratados e como serão endereçados do remetente ao destinatário. Existem dois tipos:

- a) tipo de conexão: forma utilizada pelo TCP para enviar dados, ou seja, antes de enviar os dados deve ser estabelecida uma conexão. Desta forma consegue-se garantir a entrega confiável do fluxo de dados;
- b) tipo de datagrama: forma utilizada pelo UDP enviar dados. Este modelo não garante a entrega correta dos dados.

O espaço de nomes especifica como os endereços são escritos. Como exemplo, um endereço de *Socket* em espaço de nomes local, fará a escrita em arquivos locais. No espaço de nomes Internet, é composto de um endereço Internet (IP) e o número de uma porta. Uma porta distingue múltiplos *Socket's* em um mesmo *host*.

Um protocolo especifica como o dado é transmitido. Como exemplo de protocolo, existe o próprio TCP/IP; o *Apple Talk Network Protocol*, entre outros.

### 2.3.1 Primitivas

As primitivas de transporte *Socket* contêm uma estrutura básica, mas permite que os programas e aplicativos se conectem, utilizem e encerrem uma conexão, o que é o suficiente para muitas aplicações (TANENBAUM, 1997).

As principais primitivas utilizadas são mostradas na Tabela 1.

Tabela 1 - Primitivas de *Socket* para o protocolo TCP

Primitiva	Significado
SOCKET	Criar um novo ponto final de comunicação
BIND	Anexar um endereço local a um <i>socket</i>
LISTEN	Anunciar a disposição para aceitar conexões; mostra o tamanho da fila
ACCEPT	Bloquear até uma tentativa de conexão ser recebida
CONNECT	Tentar estabelecer uma conexão ativamente
SEND	Enviar dados através da conexão
RECEIVE	Receber dados da conexão
CLOSE	Encerrar a conexão

Fonte: Camada de Transporte (VASKEVITCH, 1995).

No subcapítulo seguinte serão abordados alguns dos serviços que são muito utilizados no contexto da Internet global e posteriormente o sistema de correio eletrônico.

## 2.4 SERVIÇOS DE REDE

A Internet oferece uma grande quantidade de recursos e possibilidades de uso que vão do e-mail e do acesso a páginas Web ao vídeo em tempo real e ao compartilhamento de arquivos em sistemas *peer-to-peer*<sup>4</sup>. Todas essas possibilidades de uso são construídas a partir de um conceito relativamente simples: o de serviço de rede. A seguir são listados alguns destes serviços e facilidades:

- a) *HyperText Transfer Protocol* (HTTP): é o protocolo usado na *World Wide Web* (WWW) para a distribuição e recuperação de informação. A troca de informações entre um browser e um servidor Web é toda feita através desse protocolo, que foi criado especificamente para o *World Wide Web*;
- b) *Domain Name System* (DNS): são servidores que convertem URLs nos endereços IP dos servidores. Ao digitar `http://www.unesc.net/` no Browser, por exemplo, as partes envolvidas na comunicação entre computador e rede enviarão a requisição ao servidor DNS do seu provedor que retornará o endereço IP do servidor do site para que seu computador possa finalmente acessá-lo. Este serviço é apenas um facilitador, mas é essencial para a

existência da Web como a conhecemos. Sem os servidores DNS seria necessário decorar os endereços IP de todos os sites que desejasse visitar;

- c) *Simple Mail Transfer Protocol* (SMTP): é o protocolo padrão para o envio de e-mails. Ele é usado tanto para o envio da mensagem original, do computador até o servidor SMTP do provedor, quanto para transferir a mensagem para outros servidores, até que ela chegue ao servidor destino;
- d) *Post Office Protocol* (POP3): é um protocolo utilizado no acesso remoto à uma caixa de correio eletrônico. O POP3 permite que todas as mensagens contidas numa caixa de correio eletrônico possam ser transferidas sequencialmente para um computador local;
- e) *Internet Message Access Protocol* (IMAP4): é também um protocolo para recebimento de e-mails, assim como o POP3. A diferença entre os dois é que, ao receber os e-mails via POP3, eles são apagados do servidor assim que baixados, liberando o espaço usado na caixa postal. No IMAP4, os e-mails continuam no servidor até serem removidos manualmente;
- f) mensagens instantâneas: há diversos softwares que implementam esta funcionalidade em tempo real. O mais conhecido hoje é o MSN Messenger;
- g) correio eletrônico: é atualmente um dos mais importantes meios de comunicação entre os usuários. Uma mensagem eletrônica é armazenada em um arquivo, formatada de maneira especial e programas de correio específicos são utilizados para ler (POP3 e IMAP4) e enviar mensagens (SMTP). O sistema de correio eletrônico será abordado com mais detalhes no próximo sub-capítulo.

Na Tabela 2, serão mostrados alguns serviços descritos acima, informando a porta de utilização e transporte referente a cada um destes:

---

<sup>4</sup> Arquitetura de sistemas distribuídos caracterizada pela descentralização das funções na rede, onde cada nodo realiza tanto funções de servidor quanto de cliente (MENDES, 2002).

Tabela 2 - Serviços relacionados com sua porta e transporte.

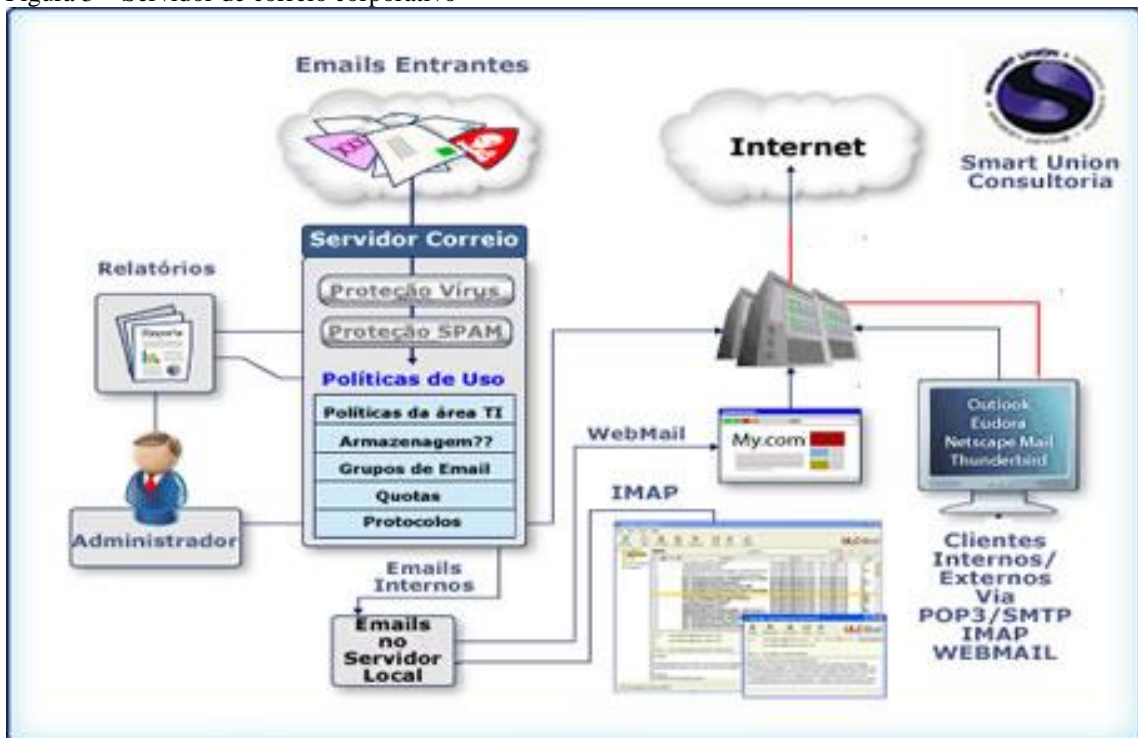
<b>Serviços</b>	<b>Porta</b>	<b>Transporte</b>
HTTP	80	TCP
DNS	53	UDP
SMTP	25	TCP
POP3	110	TCP
IMAP4	143	TCP

Fonte: Portas TCP e UDP (VASKEVITCH, 1995).

### 3 CORREIO ELETRÔNICO

O correio eletrônico ou e-mail, como é mais comumente conhecido, tornou-se popular devido às vantagens oferecidas em termos de permitir comunicação além das barreiras do tempo e do espaço. Hoje em dia, desde que o usuário possua um microcomputador e linha telefônica, este usuário pode, fazendo uso de um provedor de Internet pago ou gratuito, enviar e receber mensagens vinte e quatro horas por dia, sete dias da semana para qualquer parte do mundo. É tão rápido e fácil mandar um e-mail para qualquer parte do mundo quanto para o seu colega da sala ao lado. Você pode enviar um e-mail sempre que tiver vontade e ele estará ao alcance do destinatário sempre que ele quiser lê-lo.

Figura 3 – Servidor de correio corporativo



Fonte: Servidor de correio corporativo (INCREMAIL, 2011).

#### 3.1 MAIL USER AGENT, MAIL TRANSPORT AGENT e MAIL DELIVERY AGENT

Os sistemas de correio eletrônico normalmente estão organizados em três subsistemas: os agentes de correio de usuário (MUA), os agentes de transporte de correio (MTA) e os agentes de entrega de correio (MDA).

O agente de correio de usuário é o software utilizado pelo cliente para gerenciar seus e-mails. É responsável por receber da caixa de entrada e repassar ao MTA para envio.

O agente de transporte de correio realiza a transferência das mensagens. Recebem elas do MUA ou de outro MTA, e com base no cabeçalho define a forma que entregará a mensagem ao MDA.

O agente de entrega de correio promove a entrega das mensagens. As recebe do MTA e realiza a entrega na caixa de mensagens do destinatário.

A transferência da mensagem entre o MUA e o MTA se efetua utilizando-se um protocolo chamado SMTP que será descrito mais adiante. Este protocolo será utilizado também entre o MTA do remetente e o MTA do destinatário.

### 3.2 PROTOCOLOS E DEFINIÇÕES

O correio eletrônico é padronizado por meio de protocolos e definições. Os mais utilizados e conhecidos são:

- a) RFC 822: esta RFC<sup>5</sup> define o formato das mensagens que circulam pela Internet (Cabeçalho e Corpo da mensagem);
- b) SMTP: protocolo simples de envio de mensagens;
- c) POP3: protocolo para recebimento de mensagens completadas, incluindo cabeçalho e anexo(s);
- d) IMAP4: protocolo para recebimento de mensagens parciais, recebendo somente o cabeçalho com informações sobre a mensagem, dando a opção de o usuário recuperá-la.

#### 3.2.1 Request For Comments 822

A Request For Comments (RFC) 822 define que as mensagens têm duas partes: um cabeçalho (header) e um corpo (body). Neste padrão está definido qual é o conjunto mínimo de cabeçalhos que devem ser reconhecidos e sua funcionalidade. Cada campo do cabeçalho tem uma linha de texto ASCII, formatado da seguinte maneira: Nome do campo, um sinal de dois-pontos e o valor do campo (NOME\_DO\_CAMPO:VALOR\_DO\_CAMPO).

Os principais campos do cabeçalho de uma mensagem de correio eletrônico estão descritos na Tabela 3:

---

<sup>5</sup> Request For Comments (RFC) – Série de relatórios técnicos numerados em ordem cronológica e de acesso livre. Estão disponíveis em: [www.ietf.org/rfc](http://www.ietf.org/rfc).

Tabela 3 - Campos do cabeçalho na RFC 822 ligados ao transporte de mensagens.

<b>Cabeçalho</b>	<b>Significado</b>
To:	O endereço de e-mail do destinatário principal
Cc:	O endereço de e-mail do destinatário secundário
Bcc:	O endereço de e-mail para cópia carbono ocultas
From:	A pessoa que criou a mensagem
Sender:	O endereço de e-mail do remetente
Received:	A linha incluída por cada agente de transferência ao longo da rota
Return-Path:	Pode ser usada para identificar um caminho de volta ao remetente

Fonte: TANENBAUM (2000).

Além dos campos informados acima, as mensagens baseadas na RFC 822 também possuem outros campos de cabeçalho. Alguns deles estão sendo mostrados e descritos na Tabela 4:

Tabela 4 - Alguns campos utilizados no cabeçalho da mensagem RFC 822.

<b>Cabeçalho</b>	<b>Significado</b>
Date:	A data e hora em que a mensagem foi enviada
Reply-To:	Endereço de correio eletrônico para onde a resposta deve ser enviada
Message-Id:	Número exclusivo que será usado para fazer referência a essa mensagem posteriormente
In-Reply-To:	Message-Id da mensagem original correspondente à resposta
References:	Outras Messages-Id relevantes
Keywords:	Palavras-chaves do usuário
Subject:	Pequeno resumo da mensagem apresentado em apenas uma linha

Fonte: TANENBAUM (2000).

Após os cabeçalhos, segue uma linha em branco e então o corpo da mensagem em que o usuário poderá incluir o texto que desejar.

### 3.2.2 Simple Mail Transfer Protocol

A maneira mais simples de se transferir mensagens de um remetente para um destinatário é estabelecer uma conexão de transporte entre a máquina de origem e de destino.

O Simple Mail Transfer Protocol (SMTP) é o protocolo de aplicação mais importante para o funcionamento do correio eletrônico. Estabelecendo uma conexão com a porta 25, ele utiliza o serviço de transferência de dados confiável do TCP para transferir a mensagem desde o remetente até a caixa postal do destinatário. As mensagens trocadas pelo protocolo SMTP são em caracteres ASCII (VASKEVITCH, 1995, tradução nossa).

A troca de mensagens entre servidor e cliente é muito simples, começando com a conexão do *host* de transmissão (cliente) com o *host* de recepção (servidor). Quando o cliente envia o primeiro pacote de dados, o servidor envia uma mensagem informando se as informações contidas no pacote são válidas. Se lido com sucesso, o servidor “avisa” o cliente que está apto a receber mensagens, caso contrário, o cliente encerra a conexão com o servidor (BOCHENSKI, 1995).

O protocolo SMTP usa comandos simples para estabelecer uma conexão e para transferir informação e dados entre servidores. Os comandos consistem de palavras únicas com informações adicionais seguindo o comando. Após cada comando, o servidor remoto deve enviar um código de resposta para o servidor de origem de forma a indicar se o comando teve sucesso.

Tabela 5 - Alguns comandos utilizados pelo servidor SMTP.

Comando	Descrição
HELO	Abre uma sessão com o servidor
MAIL	Identifica remetente da mensagem
RCPT	Identifica receptor da mensagem
DATA	Identifica o início da mensagem. O fim é identificado por um ponto e retorno de linha
RSET	Zera a conexão SMTP
HELP	Requisita lista de comandos
QUIT	Para a sessão SMTP

Fonte: RUBIN (2001).

Sempre que o cliente estiver conectado na Internet, este sistema de troca de informações entre servidor e cliente irá funcionar corretamente. Porém há a possibilidade de o cliente não estar conectado à Internet. Desta forma, foram criados os sistemas de armazenamento local de e-mails. São programas que se conectam automaticamente ao servidor de e-mail cadastrado nele, atualizando assim os e-mails contidos na sua caixa de

correio. Para prover esta facilidade, existem os protocolos de recebimento de e-mails POP3, que será apresentado no próximo sub-capítulo, e IMAP4, que será apresentado posteriormente (MCFEDRIES, 2008, tradução nossa). Estes protocolos de recebimento de mensagens podem trabalhar com três formas possíveis, conforme informado no documento RFC 1733. São elas:

- a) online: as mensagens ficam armazenadas no servidor de e-mail e o usuário as manipula remotamente através de um programa de e-mail, com a possibilidade de criar, remover, alterar e mover pastas e mensagens;
- b) offline: é o mais conhecido e utilizado modo de acesso remoto a mensagens no servidor de e-mail. Esta forma de acesso é utilizada pelo protocolo POP3. Neste modo, são transferidos todos os pacotes de novas mensagens para o computador local do usuário e as apaga da sua origem.
- c) disconnected: comumente confundido com o modo offline, o disconnected também transfere todos os pacotes para o computador local, porém mantém as mensagens originais no servidor de e-mail. Como os pacotes poderão ser alterados na máquina local do usuário, neste modo há a possibilidade de fazer uma sincronização para atualizar no servidor de e-mail, as alterações feitas na máquina local do usuário.

### 3.2.3 Post Office Protocol Release 3

O protocolo Post Office Protocol Release 3 (POP3) foi projetado para oferecer suporte ao processamento de e-mails off-line e a especificação e regulamentação deste protocolo são dadas pela RFC 1939. As mensagens de e-mail são removidas do servidor e armazenadas no cliente local, a menos que o cliente tenha sido configurado para deixar os e-mails no servidor. Isso coloca o gerenciamento dos dados e a responsabilidade pela segurança nas mãos do usuário. Este protocolo não oferece recursos de colaboração avançados, como calendário, contatos e tarefas (BATTISTI, 2001).

Para abrir uma conexão com este protocolo, o cliente deve conectar no servidor e na porta definida para o serviço, que é a 110 quando não utilizar conexão Security Socket Layer (SSL) ou Transport Layer Security (TLS) e a porta 995 quando utilizar conexão SSL ou TLS.

Após fazer a conexão com o servidor, o servidor envia uma mensagem de congratulação e fica a espera de mensagens enviadas pelo cliente. Uma conexão correta e estável deve passar por três estados:

- a) autenticação: neste estado, o cliente de e-mail que estiver conectado ao servidor deve ser autenticado antes que os usuários possam recuperar seus e-mails. Se o nome de usuário e senha que são fornecidos pelo cliente corresponder àqueles contidos no servidor, o usuário será autenticado e continuará com o estado de *transação*. Caso contrário, o usuário receberá uma mensagem de erro;
- b) transação: neste estado, o cliente envia comandos e o servidor recebe e responde a eles de acordo com o protocolo. Qualquer solicitação do cliente recebida pelo servidor que não esteja de conformidade com o protocolo é ignorada e uma mensagem de erro é enviada pelo servidor;
- c) atualização: encerra-se a conexão entre o servidor e cliente. É o último comando transmitido pelo cliente.

Para evitar qualquer dano ao armazenamento de e-mail depois que o cliente é autenticado, o serviço bloqueia a caixa de correio do usuário. Os novos e-mails entregues na caixa de correio depois que o usuário foi autenticado (e que a caixa de correio foi bloqueada) não estarão disponíveis para download até que a conexão tenha sido encerrada. Além disso, somente um cliente pode conectar-se a uma caixa de correio de cada vez (MYERS, 1996, tradução nossa).

### **3.2.4 Internet Message Access Protocol Version 4**

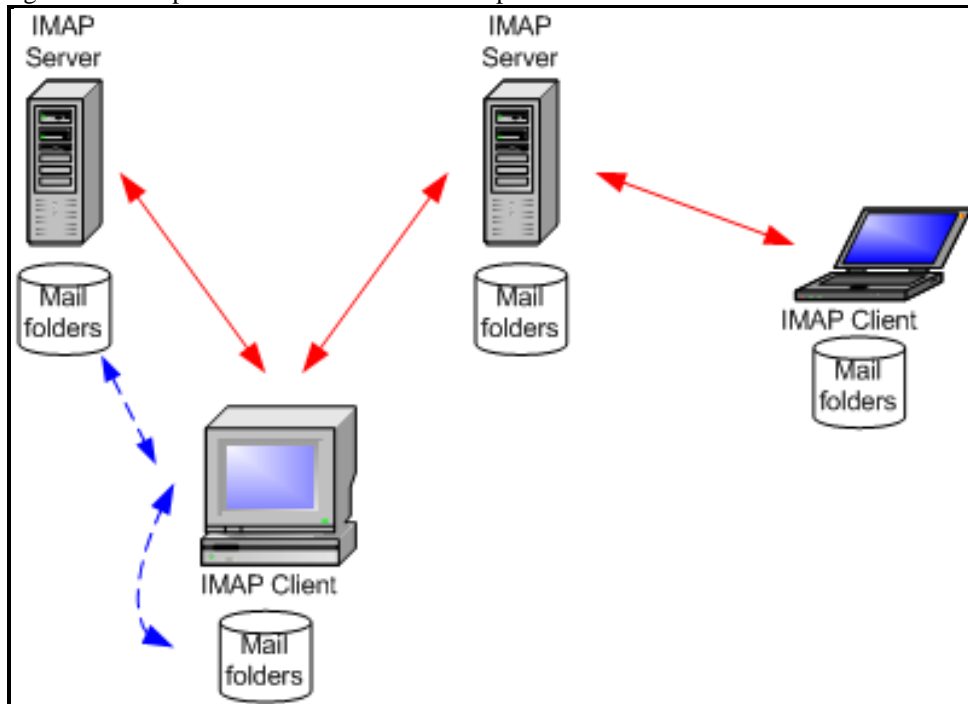
O protocolo Internet Message Access Protocol Version 4 (IMAP4) trata-se de um método de acesso a mensagens eletrônicas armazenadas em um servidor remoto.

Através de um programa cliente que envia comandos ao servidor de correio eletrônico que suporta o protocolo IMAP4, o usuário pode manipular suas mensagens e pastas a partir de computadores diferentes em diversas localidades sem que seja necessária a transferência das mesmas do servidor para o computador de onde se está fazendo o acesso. Assim, as mensagens podem ser acessadas em um notebook durante uma viagem, no computador de casa ou do trabalho (PETERSON, 2000, tradução nossa).

Uma característica importante do IMAP é permitir que a manipulação de mensagens e pastas fosse feita como se estivessem no computador local.

Na Figura 4, mostra-se o caminho percorrido por uma mensagem quando ela é recuperada do servidor IMAP4 e a demonstração de mais de um cliente de e-mail utilizando o mesmo servidor.

Figura 4 – Recuperando um e-mail através do protocolo IMAP4.



Fonte: TANENBAUM (2003)

#### 3.2.4.1 Exemplos de Problemas e as Soluções IMAP4

Muitos usuários não têm a paciência ou tempo para ficar aguardando todos os seus e-mails serem baixados do seu correio eletrônico. Porém o protocolo trabalha desta forma, fazendo a conexão e trazendo os e-mails completos para a sua caixa de entrada, inclusive os anexos.

Segundo Martins (2011), esta é uma grande vantagem do protocolo IMAP4, no qual oferece duas soluções para este problema:

- a) ao acessar a sua caixa postal o usuário pode receber, inicialmente, uma listagem com os cabeçalhos das mensagens presentes na mesma. Desta forma, o usuário pode selecionar aquelas que ele deseja que o servidor IMAP4 envie para o seu computador local, baseado em seu interesse e bom-senso;
- b) permitir ao usuário escolher que parte da mensagem ele deseja receber. Se assim desejar, o usuário pode pedir para receber somente a parte textual da mensagem, e não receber o(s) arquivo(s) anexado(s).

### 3.2.4.2 Respostas e Comandos do Protocolo IMAP4

Como todo protocolo de comunicação, o IMAP4 também necessita da troca de mensagens para alcançar o seu objetivo.

A comunicação é feita através do protocolo Transmission Control Protocol (TCP) podendo ser utilizado com ou sem segurança, usando o protocolo Security Socket Layer (SSL) (SOARES, 1995).

Para começar a troca de mensagens com o servidor de e-mail, deve-se estar ciente de que todos os comandos começam com um caractere alfanumérico, chamado *rótulo* seguido de um espaço em branco. O cliente deve fornecer um novo *rótulo* para cada comando enviado ao servidor. Quando o servidor necessita informações adicionais ou complementares, ele envia ao cliente o caractere “+”, solicitando o restante dos dados pertinentes ao comando.

Caso o comando esteja correto, o servidor irá encaminhar uma resposta contendo uma ou mais linhas. Quando contiver mais de uma linha, para cada linha que pertencer aquele comando, será adicionado o caractere “\*”.

Quando o cliente envia um comando ao servidor, ele responde informando se o comando foi executado com sucesso ou não. Há três possibilidades de resposta por parte do servidor: “OK” quando o comando foi executado com sucesso; “NO” quando o comando existe, porém o seu complemento está errado e “BAD” quando o comando enviado não existir ou for inválido.

Abaixo serão listados os principais comandos utilizados pelo protocolo IMAP4 e uma breve explicação sobre cada comando:

- a) LOGIN [User] [Password]
  - argumentos: usuário e senha para autenticação;
  - descrição: este comando inicia a comunicação por fazer logon na caixa de correio;
- b) LIST “” “\*”
  - descrição: este comando fornece uma lista de pastas disponíveis do usuário;
- c) SELECT [Folder]
  - argumentos: seleciona uma pasta contida no seu correio;
  - descrição: este comando seleciona a caixa de correio apropriada;
- d) FETCH [Number]
  - argumentos: número referente a mensagem;

- descrição: este comando recupera mensagens específicas, conforme o número informado;

e) LOGOUT

- descrição: este comando finaliza a conexão do usuário com o servidor de e-mail;

As respostas enviadas pelo servidor serão sempre no formato “<rotulo>  
<comando> <complemento>”, exemplo:

**1 OK LOGIN Ok**

Onde: “1” corresponde ao rótulo; “OK” corresponde ao status do comando e “LOGIN Ok” corresponde ao complemento.

## **4 TRABALHOS CORRELATOS**

Com base em informações que foram relevantes para o entendimento do assunto, abaixo são apresentados resumidamente alguns trabalhos que estão relacionados a esta pesquisa. As observações apontadas nos trabalhos relatam e demonstram diversas formas de interação entre correios eletrônicos, à montagem de pacotes a serem enviados e exemplos práticos da funcionalidade da troca de mensagens entre um cliente de e-mail e um servidor de e-mail.

### **4.1 ENVIANDO E RECEBENDO MENSAGENS ATRAVÉS DOS PROTOCOLOS SMTP E POP3**

Trabalho de conclusão de curso desenvolvido por João Gilberto Magalhães em 2007, acadêmico da Universidade Estadual de Ponta Grossa.

O objetivo desse trabalho foi demonstrar como é possível enviar e receber e-mail através da utilização dos protocolos de comunicação SMTP e POP3 existentes na camada de software. Também teve como objetivo ilustrar rapidamente como funciona o mecanismo para definição de padrão (RFC) e suas aplicações práticas.

### **4.2 PROTÓTIPO DE SOFTWARE PARA NOTIFICAÇÃO DE RECEBIMENTO DE NOVOS E-MAIL'S, BASEADO NO PROTOCOLO IMAP**

Trabalho de conclusão de curso do desenvolvido por Alexandre Eduardo Knorst, tendo como objetivo o estudo dos principais protocolos de correio eletrônico, enfatizando o protocolo IMAP. Além dos estudos, implementou-se também um protótipo para notificação do recebimento de novos e-mail's, baseado no protocolo IMAP e algumas informações adicionais sobre estes e-mail's, tendo o usuário, a possibilidade de configurar a verificação de cada novo e-mail recebido.

### **4.3 ADVANCED MAIL: FERRAMENTA DE ACESSO AVANÇADO A CONTAS DE E-MAIL BASEADO NO PROTOCOLO POP3**

Trabalho de conclusão de curso desenvolvido por Fernando Sainovich em 2004, acadêmico da Universidade do Extremo Sul Catarinense.

No desenvolvimento desta ferramenta utilizou-se a infraestrutura da empresa Unimed de Criciúma, que possui um sistema de recuperação de mensagens eletrônicas baseado no protocolo POP3, tendo em vista que este não pode ser substituído por um protocolo mais avançado como o IMAP4, pois a empresa possui uma ferramenta de Workflow que utiliza as funcionalidades do POP3 para seu funcionamento.

#### 4.4 FERRAMENTAS EXISTENTES

Apesar do grande e explosivo crescimento da Internet e suas facilidades, os clientes de e-mail ainda são bastante utilizados por permitir enviar, receber e personalizar mensagens de e-mail (GHIZZO, 2011). Em um cliente de e-mail podem ser configuradas diversas contas de correio eletrônico ao mesmo tempo, manuseando-as e configurando-as independentemente. Disponibilizam também a possibilidade de criar listas de contato detalhadas e trabalhar com mensagens criptografadas dando maior segurança ao usuário no momento do envio ou recebimento de uma mensagem.

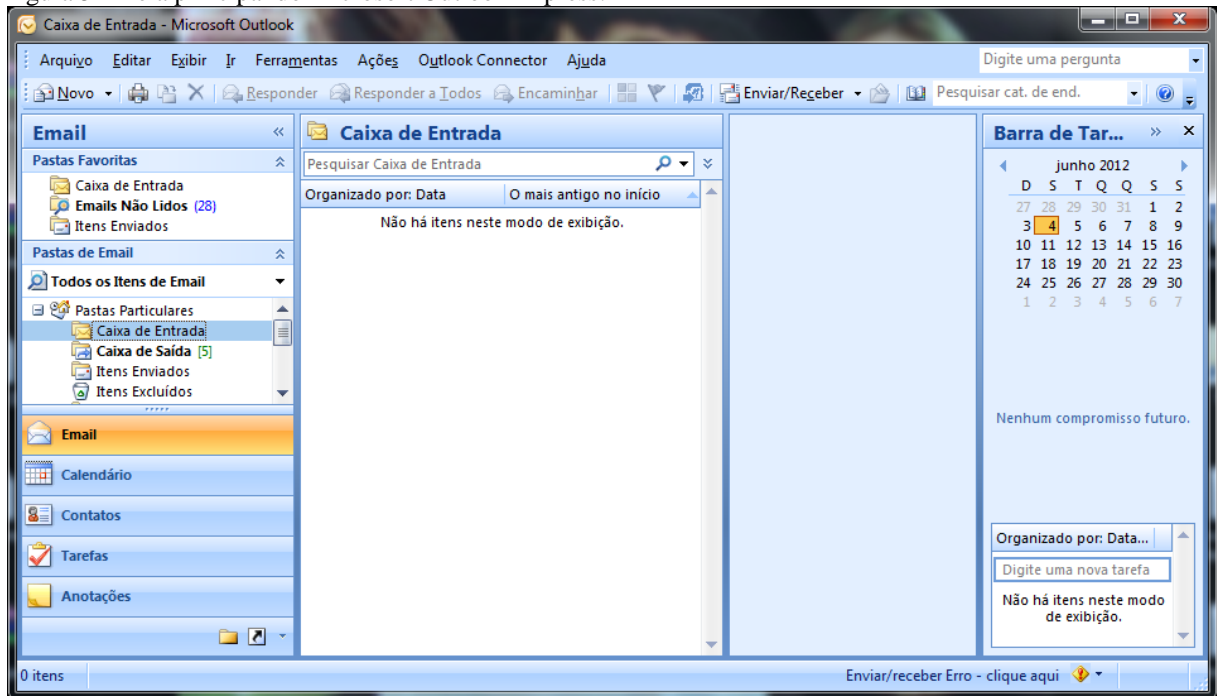
Algumas ferramentas foram bastante utilizadas quando o cliente de e-mail foi elaborado e desenvolvido. Uma das pioneiras foi a *Lotus Organizer* que era utilizado principalmente nos computadores que utilizam a plataforma Windows, onde trabalhava com uma interface não muito rica, levando ao seu declínio e dando lugar a outra ferramenta que é utilizada pela maioria dos computadores com sistema operacional Windows, o Microsoft Outlook Express que já vem incluído no próprio Windows.

##### 4.4.1 Microsoft Outlook Express

Aplicativo que permite gerenciar uma ou mais contas de e-mail através de um layout simples que pode ser personalizado pelo usuário (MICROSOFT, 2004). Além de ser um cliente de e-mail, também pode ser utilizado como calendário, onde se pode agendar compromissos diários, semanais ou mensais. Sua interface gráfica é bastante rica e dispõe de várias opções de configurações ajustáveis de acordo com a necessidade de cada usuário.

O Microsoft Outlook Express foi por muito tempo o cliente de e-mail mais utilizado pelos usuários, realizando uma comunicação estável com os protocolos de recebimento de e-mail POP3 e IMAP4, e o protocolo de envio de e-mail SMTP. Na Figura 5 é apresentada a tela principal da ferramenta, podendo identificar as funcionalidades expostas acima.

Figura 5 – Tela principal do Microsoft Outlook Express.



Fonte: MICROSOFT, 2004.

Além do Microsoft Outlook Express, há também outro cliente de e-mail muito utilizado chamado Mozilla Thunderbird que logo na sua estréia na versão 1.0 atingiu mais de um milhão de downloads em apenas dez dias (DOTZLER, 2004, tradução nossa).

#### 4.4.2 Mozilla Thunderbird

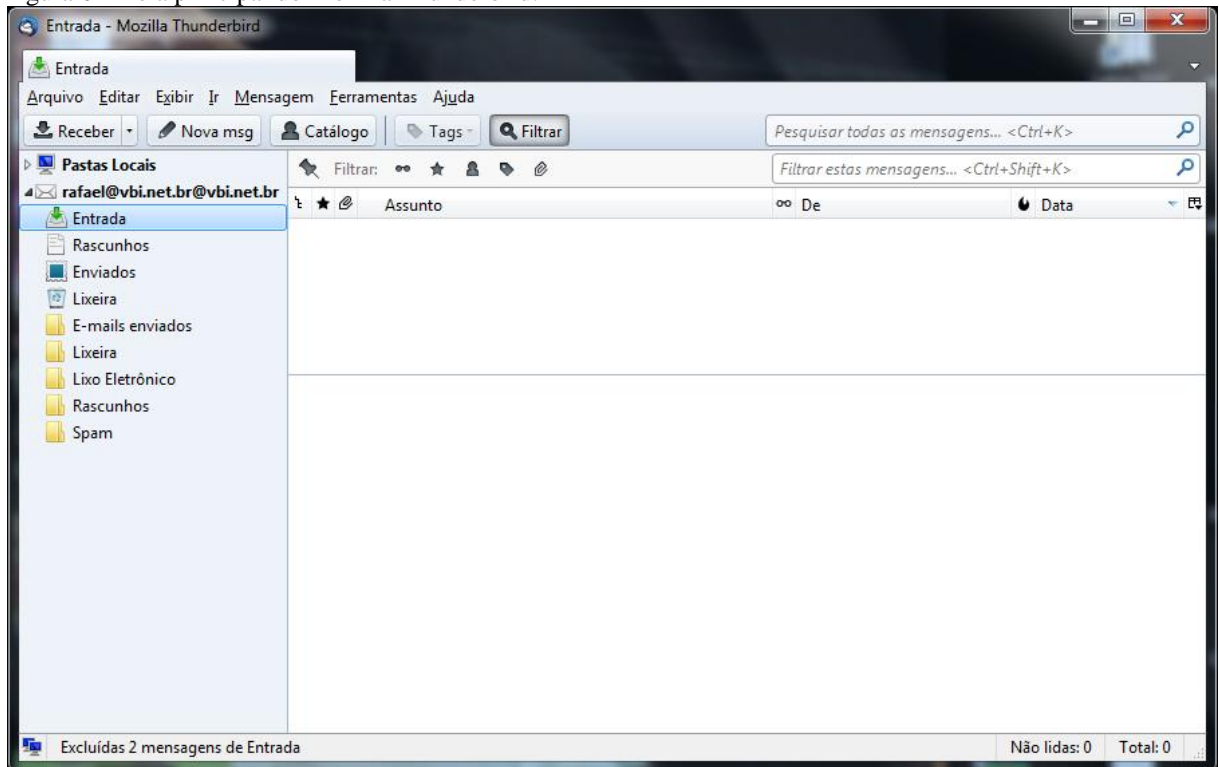
Software de correio eletrônico gratuito que trabalha de forma segura no bloqueio de imagens, filtro anti-spam e utiliza mecanismos que previnem golpes através de mensagens eletrônicas (MOZILLA, 2011). Além de sua solidez e segurança, o Thunderbird também oferece recursos como assinaturas digitais, criptografia de mensagens, suporte a certificados e dispositivos de segurança.

Por ser um software livre, conhecido como programas de código aberto, a sua utilização é irrestrita mesmo para uso comercial tornando-o um produto bastante desejado e explorado. Sua interface mantém-se praticamente estável desde sua primeira versão, apesar de embutir cada vez mais novas funcionalidades.

Com sua expansão, o Microsoft Outlook Express foi deixando de ser o principal cliente de e-mail, tendo muitas reclamações de usuários informando que este trabalhava lentamente quando havia um alto número de e-mails na sua caixa de entrada. Neste momento

o a companhia Mozilla entrou em cena com um produto legível e de fácil manuseio. Na Figura 6 é apresentada a tela principal do Mozilla Thunderbird.

Figura 6 – Tela principal do Mozilla Thunderbird.



Fonte: MOZILLA..., 2011.

## 5 SISTEMA PROPOSTO

Este projeto teve como objetivo principal montar um protótipo de um servidor Proxy que faça a filtragem de e-mails de um correio eletrônico, com a finalidade de recuperar do servidor de e-mail, apenas os e-mails que interessem ao usuário. Esta filtragem será configurada pela ferramenta, contendo uma lista negra (BlackList) e uma lista branca (WhiteList), onde os e-mails são restringidos ou permitidos, respectivamente, conforme necessidade do usuário, podendo realizar o cadastro dos filtros que o interessam. As funcionalidades serão utilizadas conforme a disponibilidade do protocolo IMAP4. O presente trabalho foi realizado a partir das etapas metodológicas abaixo.

### 5.1 METODOLOGIA

A primeira etapa metodológica consiste no levantamento bibliográfico realizado por meio de busca em bases de dados, artigos, publicações em periódicos, livros.

De posse do material bibliográfico, deu-se sequencia com a descrição dos principais conceitos sobre o tema, nos quais constam os protocolos de comunicação e suas ferramentas, compreendendo e, posteriormente, desenvolvendo um protótipo com base nestes estudos.

A terceira etapa consiste na amostragem das tecnologias utilizadas para o desenvolvimento da ferramenta, fazendo uma abordagem dos serviços utilizados por estas tecnologias.

Na quarta etapa são mostrados os principais protocolos de comunicação utilizados por um correio eletrônico, apresentando exemplos das trocas de mensagens utilizadas pelos protocolos.

A partir da quinta etapa metodológica, escreveu-se o trabalho desenvolvido, dando ênfase ao protótipo desenvolvido, mostrando telas e a sua funcionalidade perante o objetivo geral de desenvolvimento da ferramenta.

### 5.2 RECURSOS

Para a pesquisa será utilizado um computador com as configurações básicas para o acesso a Internet que esteja instalado o sistema operacional Windows, a ferramenta Eclipse para o desenvolvimento da ferramenta na linguagem de programação Java e o aplicativo

Mozilla Thunderbird que é um cliente de e-mail simples para envio e recebimento de e-mails em um computador desktop, podendo gerenciar múltiplas contas de e-mail.

Como recurso humano haverá o auxílio de um professor do curso de Ciência da Computação com conhecimento na área de Redes de Computadores e serão utilizados recursos gratuitos ou fornecidos pela universidade para os devidos fins.

## 6 TRABALHO DESENVOLVIDO

O presente trabalho refere-se ao desenvolvimento de um protótipo de servidor Proxy onde os e-mails são filtrados através do protocolo de comunicação IMAP4, disponibilizando ao usuário a filtragem necessária para que somente sejam recebidos os e-mails que o interessam, descartando os demais. Os e-mails que não forem recebidos pelo usuário, continuarão dispostos no servidor de e-mail, não sendo removidos.

Neste capítulo foram descritas as ferramentas utilizadas para o auxílio da implementação do protótipo, formas como a ferramenta foi montada e amostragem de algumas telas e partes de código-fonte que são relevantes para o funcionamento do mesmo.

### 6.1 MODELAGEM

Para o desenvolvimento de um sistema devem-se construir modelos que expliquem o comportamento do software em questão. Nestes modelos devem conter dados que demonstrem a forma como o sistema funcionará e que facilite a construção do mesmo.

Uma modelagem de software é uma simplificação da realidade tanto na visualização, arquitetura e melhor compreensão do sistema, podendo trabalhar com planos detalhados ou planos mais gerais, com uma visão panorâmica da aplicação.

Para a modelagem deste protótipo foi utilizada a linguagem *Unified Modeling Language* (UML) que é utilizada nos mais diversos tipos de sistemas, podendo ser aplicada desde a especificação da análise de requisitos até a finalização com a fase de testes (BOOCH, 2000).

#### 6.1.1 Fases do Desenvolvimento de um Sistema em UML

Há cinco fases no desenvolvimento de sistemas de software que não devem ser executados na ordem descrita abaixo, mas concomitantemente de forma que problemas detectados em certa fase modifiquem e melhorem as fases desenvolvidas anteriormente, gerando um resultado de alta qualidade e desempenho.

- a) análise de requisitos: captura as necessidades e intenções dos usuários perante a utilização do sistema a ser desenvolvido;
- b) análise: são as primeiras abstrações (classes e objetos) e mecanismos que estarão presentes no domínio do problema;

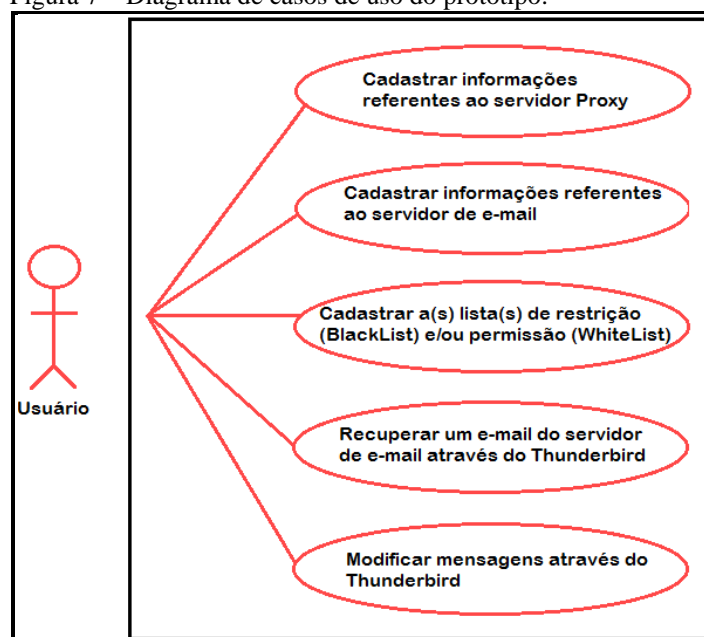
- c) design ou projeto: o resultado da análise é expandido em soluções técnicas. Novas classes serão criadas para uma infraestrutura mais robusta e compreensível. Entre estas classes estão a interface do usuário, gerenciamento de banco de dados, comunicação com outros sistemas, dentre outros;
- d) programação: nesta fase, as classes provenientes do design serão convertidas para o código-fonte da linguagem escolhida para o desenvolvimento do sistema;
- e) testes: geralmente são divididos em três subfases: testes de unidade, onde o sistema é geralmente testado pelo programador, teste de integração que confirmam se as classes estão cooperando umas com as outras e o teste de aceitação que verifica se o sistema está funcionando conforme a sua especificação.

A ferramenta em questão foi modelada com os diagramas de Casos de Uso e de Atividades.

#### 6.1.1.1 Diagrama de Casos de Uso

Neste momento descreve-se um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário (BOOCH, 2000).

Figura 7 – Diagrama de casos de uso do protótipo.



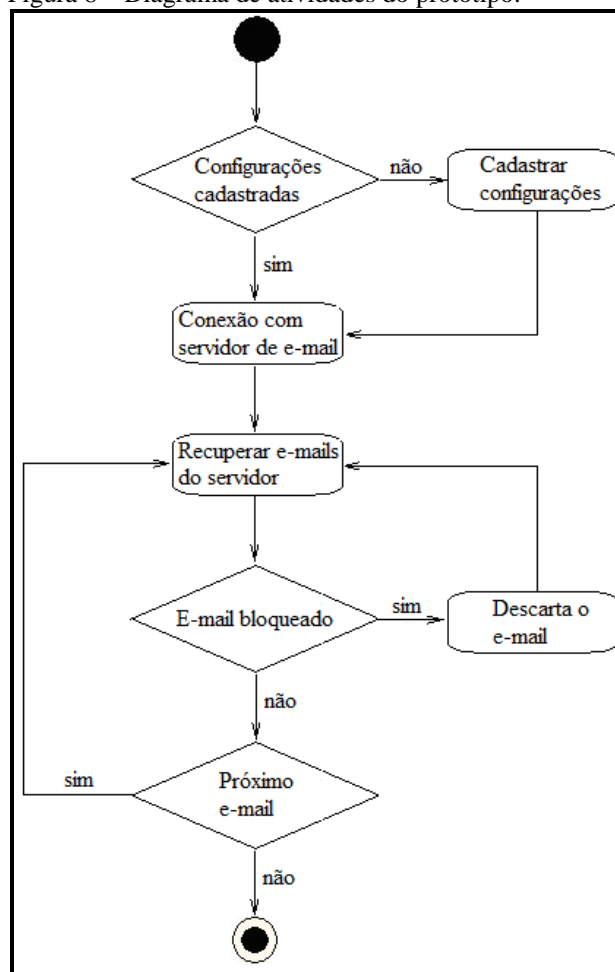
Fonte: Do Autor.

Além do diagrama de casos de uso, faz-se importante a demonstração do diagrama de atividades, descrito no próximo subcapítulo.

### 6.1.1.2 Diagrama de Atividades

O diagrama de atividades descreve basicamente a sequência de atividades através de um fluxo sequencial, mostrando o controle de um sistema e permitindo uma radiografia de uma lógica de código, onde esta lógica é apresentada em um fluxograma (BOOCH, 2000).

Figura 8 – Diagrama de atividades do protótipo.



Fonte: Do autor.

## 6.2 AMBIENTE DE DESENVOLVIMENTO

A linguagem escolhida para o desenvolvimento do protótipo foi a linguagem Java, devido ao suporte a múltiplas plataformas e orientação a objetos. Uma grande vantagem da linguagem Java é a orientação a objetos possibilitou grande facilidade devido à reutilização de

código, trabalhando com dados na memória e facilitando a implementação e entendimento do código fonte.

Para o cadastramento das informações para conexão e filtragem, foi escolhido o banco de dados SQLite, implementado na linguagem de programação C, não precisando executar um Sistema Gerenciador de Banco de dados (SGBD) separado. Este banco de dados é mais utilizado por aplicativos móveis onde as informações não são guardadas em grandes quantidades.

Como primeira análise, verificou-se qual o cliente de e-mail seria utilizado, chegando assim ao Mozilla Thunderbird pelo fato de trabalhar com pacotes de recebimentos mais elaborados que os demais clientes de e-mails e por ser multi-plataforma.

O Ambiente de Desenvolvimento Integrado (Integrated Development Environment (IDE)) utilizado foi o Eclipse, devido às facilidades de configuração e depuração do código-fonte.

Na escolha do servidor de e-mail IMAP, foram priorizados àqueles onde não utilizavam Security Socket Layer (SSL), na qual oferece um canal de comunicação segura entre o cliente de e-mail e o servidor de e-mail. O SSL torna-se mais complexo devido à necessidade de uma chave pública e privada para a troca de mensagens entre cliente e servidor, sendo a chave pública reconhecida por todos os clientes e a chave privada apenas pelo destinatário que neste caso é o servidor de e-mail. Com isso, optou-se por utilizar o servidor de e-mail, onde implementa esta funcionalidade, não dependendo exclusivamente de uma conexão SSL, podendo assim, explorá-lo seguindo as definições do protocolo IMAP.

Antes de começar efetivamente a programação da ferramenta, alguns testes referentes à troca de informações entre cliente de e-mail e servidor de e-mail, foram necessários. Para estes testes, foi utilizada a aplicação Wireshark, um analisador de protocolos de rede que, além de capturar todas as informações enviadas e recebidas através de dispositivos de comunicação, tais como, placa de rede, placa de fax, modem, entre outros, também os organiza por protocolo. A ferramenta foi bastante utilizada para depuração dos problemas encontrados durante o desenvolvimento da ferramenta de filtragem de e-mail.

Com os primeiros testes realizados e o entendimento do conceito principal da troca de mensagens entre cliente de e-mail e servidor de e-mail, começou-se a procurar a melhor forma de desenvolvimento do protótipo, tendo como objetivo o funcionamento e posteriores resultados satisfatórios.

### 6.3 DESENVOLVIMENTO DO PROTÓTIPO

Com as ferramentas auxiliares em mãos, iniciou-se uma pesquisa para identificar a melhor forma de implementar o protótipo, baseando-se nos objetos fornecidos pela linguagem Java que possibilita a inserção de diversas bibliotecas externas documentadas e consolidadas, facilitando o estudo e uso de cada uma destas bibliotecas.

Antes de escrever a primeira linha de código foram extraídas as ideias de como o protótipo iria se comportar perante a comunicação com o servidor de e-mail, passando-as para o papel a fim de facilitar a visualização da aplicação como um todo. Esta etapa foi a principal segurança de que aquilo que se tinha em mente, poderia efetivamente virar uma aplicação real.

O protótipo baseou-se no modelo Cliente-Servidor onde a troca de mensagens entre ambos necessita obrigatoriamente de uma conexão com a Internet ativa e as configurações TCP/IP funcionando conforme estabelece o sistema operacional utilizado, que neste caso foi o *Windows*. Como a linguagem Java e o cliente de e-mail Thunderbird são multiplataforma, ou seja, podem ser executados em qualquer sistema operacional sem a necessidade de alterações no código fonte, nada impede que este protótipo seja utilizado em um computador que utilize a plataforma *Linux*.

Após o desenho de como funcionaria o protótipo, começaram os primeiros experimentos relativos à comunicação utilizando a classe *Socket* na qual estabelece conexões entre máquinas, envia e recebe pacotes de dados e aguarda conexões em uma porta específica da máquina, resumindo, é um elemento de software que provê uma interface de rede para a aplicação. Com o *Socket* começaram os primeiros testes de comunicação com o servidor de e-mail IMAP, através do protocolo de comunicação IMAP4. Com a conexão estabelecida, iniciou-se a troca de mensagens com o servidor através das classes de leitura e escrita da linguagem, que deram total suporte à ferramenta em questão.

Com a instalação do banco de dados SQLite e inserção das classes obrigatórias para o funcionamento do mesmo em Java, as telas começaram a ser desenvolvidas com o intuito de simplificar o entendimento para o usuário final. Tendo em vista que o principal objetivo do protótipo é a filtragem correta de e-mails, as telas incluíam apenas o cadastro dos filtros em um primeiro momento, sendo melhoradas conforme as necessidades eram identificadas. A conexão entre aplicação Java e banco de dados SQLite foi estabelecida sem maiores problemas, sendo testada e explorada com total segurança.

A criação de conexões lógicas foi feita através de *Socket's*, abrindo uma conexão na porta 2012 para o servidor proxy, como visto na Figura 9. Esta conexão aguarda a chegada dos clientes de e-mail e são ilimitadas, podendo trabalhar com diversos clientes de e-mail simultaneamente. Isto é possível devido as *Threads* da linguagem Java e a aceitação de *multithreading* nos sistemas operacionais *Windows* e *Linux*. *Thread* nada mais é que um fluxo sequencial de execução, onde é permitido que o programa leia mais de uma linha de execução simultaneamente. Para a conexão com o servidor IMAP de testes (provedor Terra), também se utiliza *Socket*, conectando no endereço *imap.terra.com.br* e porta *143*, conforme código apresentado na Figura 10.

Figura 9 – Criação do *ServerSocket* para o servidor proxy.

```
// Servidor que ficará aguardando conexões dos clientes de e-mail.
ServerSocket
io_sst_proxy          =      new ServerSocket (2012);
```

Fonte: Do Autor.

Figura 10 – Criação do *Socket* para conexão no servidor de e-mail IMAP Terra.

```
// Cria a conexão e os objetos de entrada e saída do servidor de e-mail IMAP Terra.
Socket
io_sck_server        =      new Socket ("imap.terra.com.br", 143);
io_in_server         =      new BufferedReader (new InputStreamReader (io_sck_server.getInputStream()));
io_out_server        =      io_sck_server.getOutputStream();
```

Fonte: Do Autor.

Com as conexões estabelecidas, começou-se a se trabalhar com a implementação do filtro de e-mails. Primeiramente foi identificado o comando que o Thunderbird enviava para receber o cabeçalho do e-mail. Sabendo este comando, o sistema o identificou no código e montou uma estrutura com as informações vindas no cabeçalho. As informações que são trazidas no cabeçalho, são montadas pelo cliente de e-mail, ou seja, ele pode solicitar apenas uma ou diversas informações no comando. Estas informações são separadas em variáveis para posterior comparação no filtro. O comando utilizado para capturar o cabeçalho de um e-mail é montado da seguinte forma: `XXX UID fetch YYYYY (UID RFC822.SIZE FLAGS BODY.PEEK[HEADER.FIELDS (From To Cc Bcc Subject Date Message-ID Priority X-Priority References Newsgroups In-Reply-To Content-Type)])`, onde:

- a) XXX é um número único e sequencial para cada conexão;
- b) UID: único identificador (unique identifier);
- c) fetch é o comando utilizado quando se deseja recuperar dados associados à uma mensagem na caixa de correio;

- d) YYYYYY: número da mensagem que se deseja recuperar, podendo estar entre ponto e vírgula (;) quando for recuperar uma sequencia de mensagens;
- e) RFC822.SIZE FLAGS: identifica que será retornado do servidor de e-mail apenas o cabeçalho da mensagem e não as mensagens com seu conteúdo completo;
- f) BODY.PEEK[HEADER.FIELDS: dentro deste parâmetro serão incluídas as tag's que se deseja recuperar;
- g) From To Cc Bcc Subject Date Message-ID Priority X-Priority References Newsgroups In-Reply-To Content-Type: tag's que serão recuperadas da(s) mensagem(ns).

Este foi o principal comando utilizado para identificação dos cabeçalhos das mensagens e filtros relacionados ao servidor proxy. Assim sendo, o sistema desenvolvido recebia todas as mensagens do servidor de e-mail antes de enviá-las para o Thunderbird. Com isso, pôde ser feito o filtro e criar uma lista somente com os e-mails que não foram bloqueados.

Na intervenção dos e-mails não foi alterada nenhuma linha recebida do servidor de e-mail, deixando seu conteúdo intacto. Apenas retirou-se da lista àqueles e-mails bloqueados conforme o cabeçalho daquela mensagem.

Os dois filtros foram tratados de forma diferente, porém utilizam os mesmos dados para consistência, que são as informações cadastradas na base de dados. Um e-mail será bloqueado pela BlackList quando há uma alguma informação cadastrada nesta lista que também esteja presente no cabeçalho do e-mail. Já para a WhiteList o e-mail será bloqueado quando a informação cadastrada não estiver no cabeçalho do e-mail. Para melhor entendimento da explicação acima, estará apresentada na Figura 11 as linhas de código que fazem esta verificação.

Figura 11 – Código que identifica se o e-mail está na BlackList ou WhiteList.

```
// Se o e-mail não estiver na BlackList ou estiver na WhiteList...
if (
    !isBlackListed()
    &&
    isWhiteListed()
)
{
    // Insere este e-mail na lista para retornar ao Thunderbird.
    lo_arl_enviar_email.add(ls_cada_email.toString());
}
```

Fonte: Do Autor.

Apesar da utilização da condição “&&” no código, a verificação é sempre feita em apenas uma das listas porque sempre que for habilitada uma lista, a outra automaticamente será desabilitada.

## 6.4 UTILIZAÇÃO DO PROTÓTIPO

Quando uma aplicação é desenvolvida para uso generalizado onde os usuários não possuem grande entendimento de como tudo aquilo foi desenvolvido, deve-se tomar cuidado para não dificultar o entendimento desta aplicação ao usuário final. A utilização de uma *interface* simples e objetiva facilita o seu uso e mantém uma relação amigável entre aplicação e usuário.

### 6.4.1 Configurando o Sistema

Um protótipo de um servidor Proxy não necessita uma *interface* rica, porque seu principal trabalho será feito nas camadas abaixo da camada de aplicação, conforme o modelo TCP/IP. Ao abrir o protótipo será apresentado na barra de tarefas (onde se encontra o relógio na plataforma *Windows*), o logotipo da aplicação, tendo três funções básicas: *Sair*, *Configuração* e *Servidor*.

Ao selecionar a função *Configuração*, será aberta uma janela com quatro abas onde estarão todas as configurações para utilização do protótipo, sendo:

- a) *Servidor Proxy*: conterà a configuração relativa ao servidor Proxy, indicando qual porta será utilizada para aguardar conexões dos clientes de e-mail, neste caso o Thunderbird, e a conexão utilizada entre cliente de e-mail e servidor Proxy;
- b) *Servidor de E-mail*: configuração do servidor de e-mail, contendo o endereço, porta, protocolo e conexão utilizados pelo servidor de e-mail. Como foi definido para este protótipo o servidor de e-mail Terra, estas informações já estão pré-definidas, podendo ser alteradas caso haja alguma alteração por parte do servidor de e-mail;
- c) *BlackList*: cadastro, alteração e remoção dos filtros utilizados para a lista negra. Esta lista será configurada sempre que o usuário desejar bloquear e-mails indesejados, através de algumas informações como: Assunto, Cópia, Destinatário, Domínio, Encaminhado e Remetente. Não há uma restrição

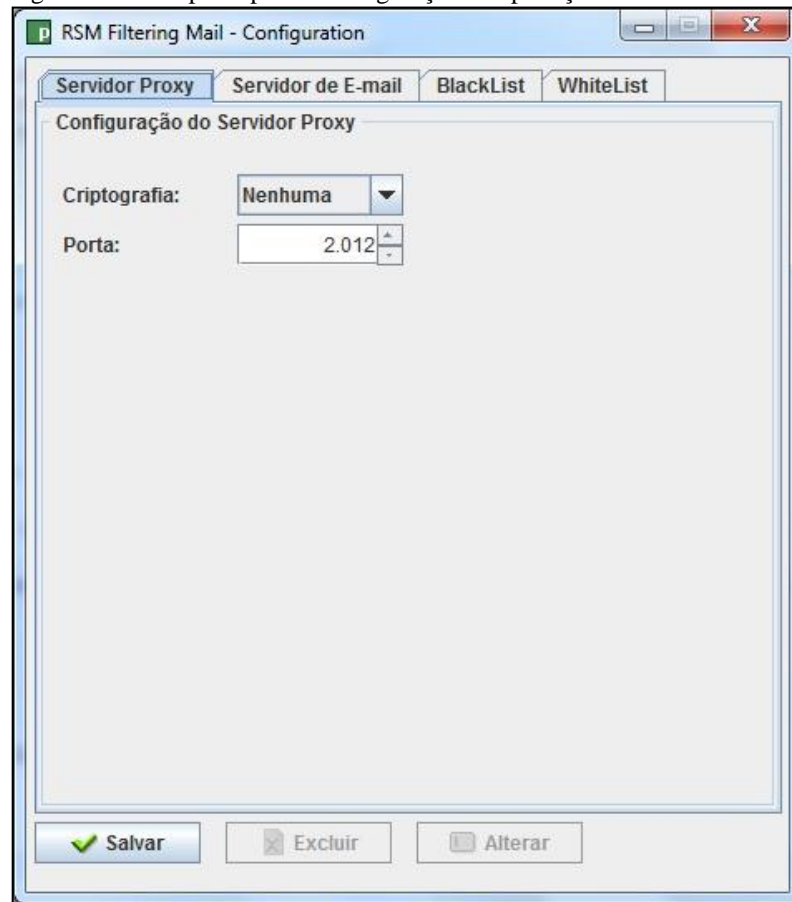
de número máximo de filtros, sendo assim, independente de quantas informações forem cadastrados, o servidor Proxy irá verificar cada uma delas. Caso algum e-mail que esteja sendo recebido se encaixar em algum destes bloqueios, ele será bloqueado pelo servidor Proxy.;

- d) WhiteList: cadastro, alteração e remoção dos filtros utilizados para a lista branca. Diferente da BlackList, esta lista será utilizada sempre que o usuário desejar a liberação de e-mails, ou seja, só irão ser recebidos aqueles e-mails onde os dados do cabeçalho deste e-mail se encaixarem em pelo menos uma destas informações cadastradas.

O programa desabilita automaticamente uma lista quando a outra for habilitada, levando em consideração que se a BlackList estiver habilitada, todos os e-mails que não forem bloqueados pelas informações configuradas nesta lista, serão recebidos pelo cliente de e-mail, tirando automaticamente a funcionalidade da WhiteList que seria de permitir o recebimento de e-mails, e não bloqueá-los.

Além das abas para configuração, os botões também são apresentados para interação de tudo o que está sendo cadastrado, alterado ou removido no banco de dados, salvando cada ação tomada em sua respectiva tabela, conforme demonstrado na Figura 12, onde está sendo apresentado somente os componentes da aba Servidores Proxy.

Figura 12 - Tela principal de configuração da aplicação



Fonte: Do Autor.

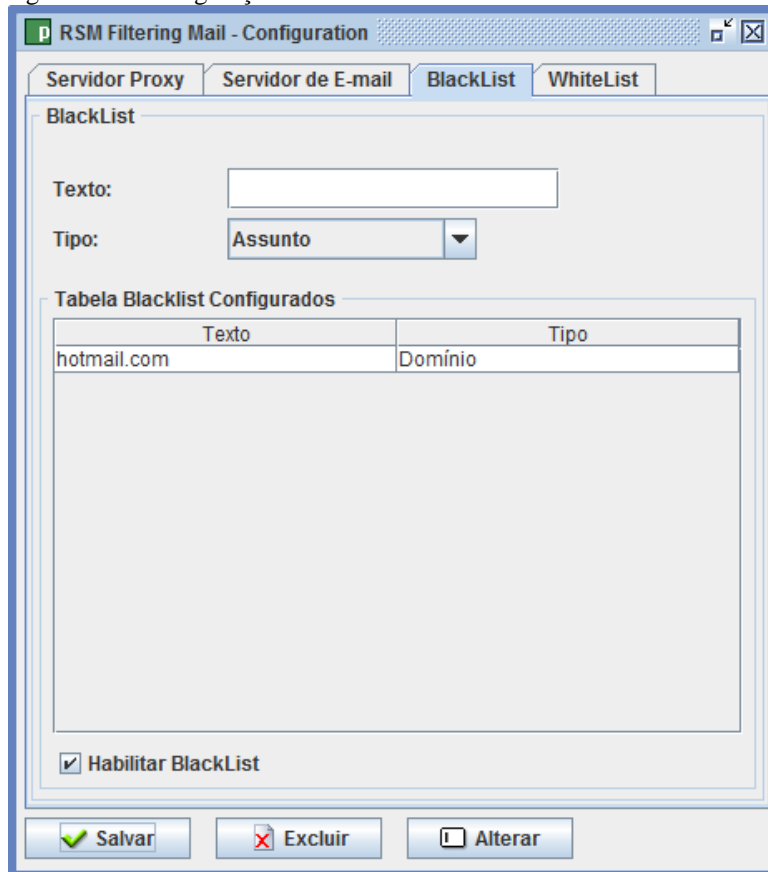
Após configurar a forma de conexão do servidor Proxy e do servidor de e-mail, é disponibilizada a forma de como as mensagens serão filtradas pela BlackList ou WhiteList. Para este filtro, dispõe-se de cinco chaves/valor:

- a) assunto: chave onde defini-se o assunto que será utilizado para filtro do e-mail;
- b) cópia: chave identificando o(s) usuário(s) e/ou domínio(s) que está(ão) sendo copiado(s) no e-mail;
- c) destinatário: chave identificando o usuário e/ou domínio que está sendo destinado o e-mail;
- d) domínio: chave identificando o domínio do usuário que está enviando o e-mail;
- e) remetente: chave identificando o usuário e/ou domínio que está enviando o e-mail.

Estes filtros servem para controle de e-mails indesejados que chegam a todo o momento nas caixas de correio eletrônico. Além de evitar mensagens com vírus, também

podem ser utilizados para restrição no recebimento de e-mails quando se usa, por exemplo, um e-mail de trabalho onde a sua caixa de correio deve conter somente e-mails relacionados ao trabalho. A configuração de um domínio na BlackList está sendo apresentado na Figura 13.

Figura 13 – Configuração de um domínio na BlackList.



Fonte: Do Autor.

Verificando a figura acima, o usuário decidiu bloquear todos os e-mails que chegam através do domínio *hotmail.com*.

#### 6.4.2 Identificando as Mensagens Apresentadas no Servidor Proxy

Pondo em prática o servidor proxy através da função *Servidor*, automaticamente é aberta a conexão local para que clientes de e-mail se conectem e a conexão com o servidor IMAP4 do provedor Terra, escolhida como base de testes para comunicação através do protocolo IMAP.

Com o servidor da aplicação aberto e ativo, conforme apresentado na Figura 14, deve-se configurar o Thunderbird para se conectar neste servidor. Para isto é inserido o nome do computador e porta onde está aberto o servidor proxy. Neste protótipo foram utilizados

dois computadores contidos na mesma rede local com IP's diferentes, mas conforme a modelagem do protocolo TCP/IP, nada impede que um computador que esteja localizado fora desta rede local possa também utilizá-lo, precisando apenas que o administrador da rede faça com que o servidor proxy seja visível em ambiente externo.

Figura 14 – Troca de mensagens entre cliente de e-mail Thunderbird e servidor IMAP, passando pelo servidor proxy.

```

RSM Filtering Mail - Pacotes recebidos e enviados
09/06/12 15:09:50.776 - Servidor Proxy aguardando conexões na porta "2012"
09/06/12 15:10:07.761 -
=====
Cliente conectado (0:0:0:0:0:0:1)
=====
09/06/12 15:10:08.651 - (S -> P -> C): * OK IMAP (version 3.7.27) at 12f.tpn.terra.com
09/06/12 15:10:08.688 - (C -> P -> S): 1 capability
09/06/12 15:10:08.976 - (S -> P -> C): * CAPABILITY IMAP4rev1 AUTH=trrproxy_v1 QUOTA
09/06/12 15:10:08.977 - (S -> P -> C): 1 OK CAPABILITY completed
09/06/12 15:10:13.364 - (C -> P -> S): 3 login "teste@terra.com.br" "teste"
09/06/12 15:10:13.727 - (S -> P -> C): 3 OK LOGIN Ok.
09/06/12 15:10:13.731 - (C -> P -> S): 4 lsub "" "*"
09/06/12 15:10:14.033 - (S -> P -> C): * LSUB () "." "Inbox"
09/06/12 15:10:14.034 - (S -> P -> C): * LSUB () "." "Drafts"
09/06/12 15:10:14.036 - (S -> P -> C): * LSUB () "." "E-mails enviados"
09/06/12 15:10:14.037 - (S -> P -> C): * LSUB () "." "Lixeira"
09/06/12 15:10:14.038 - (S -> P -> C): * LSUB () "." "Lixo Eletr&APQ-nico"
09/06/12 15:10:14.040 - (S -> P -> C): * LSUB () "." "Rascunhos"
09/06/12 15:10:14.042 - (S -> P -> C): * LSUB () "." "Sent"
09/06/12 15:10:14.044 - (S -> P -> C): * LSUB () "." "Spam"
09/06/12 15:10:14.046 - (S -> P -> C): * LSUB () "." "Trash"
09/06/12 15:10:14.049 - (S -> P -> C): 4 OK LSUB completed
09/06/12 15:10:14.051 - (C -> P -> S): 5 list "" "INBOX"
09/06/12 15:10:14.324 - (S -> P -> C): * LIST () "." "Inbox"

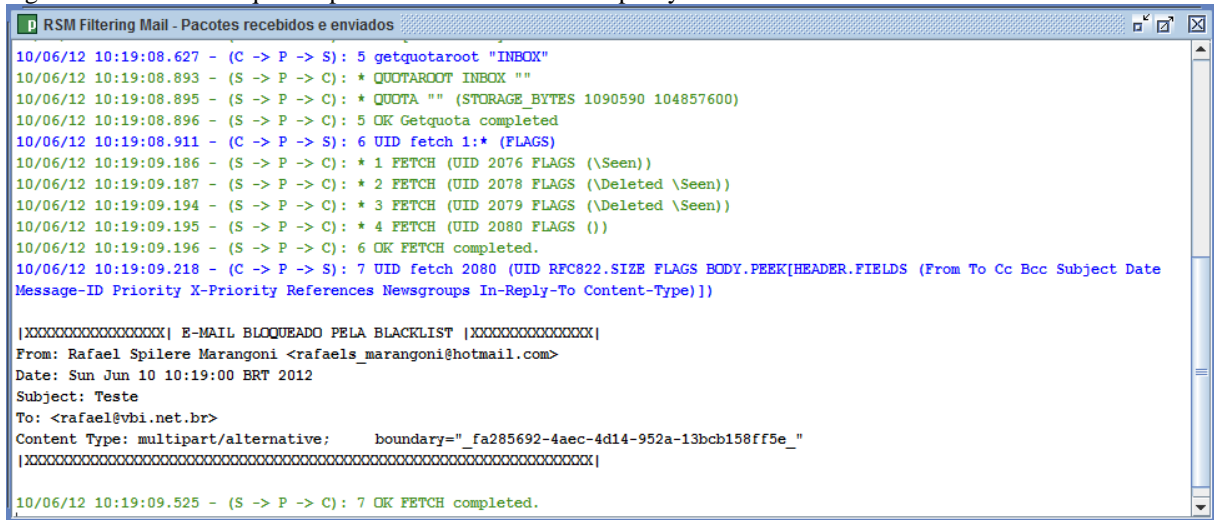
```

Fonte: Do Autor.

Conforme a figura acima, identifica-se primeiramente que um cliente de e-mail se conectou ao servidor através da mensagem “*Cliente conectado (X.X.X.X.X)*”, gerando assim uma mensagem automática do servidor IMAP informando “*\*OK IMAP...*”. Esta mensagem é enviada ao cliente sempre que ele realiza a primeira conexão com o servidor IMAP. A partir deste momento, com o cliente conectado, à troca de mensagens é feita seguinte o protocolo de comunicação IMAP4.

As ações tomadas pelo Thunderbird, ou seja, comandos enviados para o servidor de e-mail estão apontadas com a cor azul na indicação (C -> P -> S) onde o cliente de e-mail envia o comando ao servidor proxy que envia este comando ao servidor de e-mail. As respostas do servidor de e-mail estão apontadas com a cor verde, na indicação (S -> P -> C) onde o servidor de e-mail envia a resposta ao servidor proxy que encaminha ao cliente de e-mail. As mensagens bloqueadas serão apontadas com a cor preta como pode ser visto na Figura 15. Isto facilita na visualização e possível depuração de erros, podendo ajudar o desenvolvedor e também o responsável pela manutenção do servidor.

Figura 15 – E-mail bloqueado pela BlackList do servidor proxy.



```

RSM Filtering Mail - Pacotes recebidos e enviados
10/06/12 10:19:08.627 - (C -> P -> S): 5 getquotaroot "INBOX"
10/06/12 10:19:08.893 - (S -> P -> C): * QUOTAROOT INBOX ""
10/06/12 10:19:08.895 - (S -> P -> C): * QUOTA "" (STORAGE_BYTES 1090590 104857600)
10/06/12 10:19:08.896 - (S -> P -> C): 5 OK Getquota completed
10/06/12 10:19:08.911 - (C -> P -> S): 6 UID fetch 1:* (FLAGS)
10/06/12 10:19:09.186 - (S -> P -> C): * 1 FETCH (UID 2076 FLAGS (\Seen))
10/06/12 10:19:09.187 - (S -> P -> C): * 2 FETCH (UID 2078 FLAGS (\Deleted \Seen))
10/06/12 10:19:09.194 - (S -> P -> C): * 3 FETCH (UID 2079 FLAGS (\Deleted \Seen))
10/06/12 10:19:09.195 - (S -> P -> C): * 4 FETCH (UID 2080 FLAGS ())
10/06/12 10:19:09.196 - (S -> P -> C): 6 OK FETCH completed.
10/06/12 10:19:09.218 - (C -> P -> S): 7 UID fetch 2080 (UID RFC822.SIZE FLAGS BODY.PEEK[HEADER.FIELDS (From To Cc Bcc Subject Date
Message-ID Priority X-Priority References Newsgroups In-Reply-To Content-Type)])

|XXXXXXXXXXXXXXXXXXXX| E-MAIL BLOQUEADO PELA BLACKLIST |XXXXXXXXXXXXXXXXXXXX|
From: Rafael Spilere Marangoni <rafaels_marangoni@hotmail.com>
Date: Sun Jun 10 10:19:00 BRT 2012
Subject: Teste
To: <rafael@vbi.net.br>
Content Type: multipart/alternative;      boundary="_fa285692-4aec-4d14-952a-13bcb158ff5e_"
|XXXXXXXXXXXXXXXXXXXX|

10/06/12 10:19:09.525 - (S -> P -> C): 7 OK FETCH completed.

```

Fonte: Do Autor.

Conforme apresentado na figura acima, ao receber um e-mail do domínio *hotmail.com* o servidor proxy o bloqueou por estar configurado na BlackList. Subentendesse então que qualquer filtro configurado na BlackList ou WhiteList será consistido pela aplicação.

O Thunderbird cria uma nova conexão para cada pasta existente na conta do usuário. Se o usuário criou dez pastas para controle de e-mails, o Thunderbird abrirá dez conexões, uma para cada pasta. Isto é feito devido ao multiprocessamento das informações e agilidade na busca de informações. Desta forma, as pastas podem receber seus e-mails simultaneamente, ou seja, não precisam aguardar a finalização da comunicação de uma pasta. Apesar de cada pasta possuir sua conexão, a senha do usuário é solicitada apenas uma vez pelo fato do Thunderbird armazenar esta senha para as demais conexões.

Como já foi dito anteriormente, o servidor proxy foi projetado para aceitar conexões ilimitadas da mesma forma que o servidor de e-mail trabalha. Se não fosse feito desta maneira, não iria atender os clientes de e-mail existentes.

Pensando em uma rede local, um servidor proxy tem como objetivo bloquear o recebimento de e-mails de todas as máquinas contidas nesta rede. Com isso, deve-se montar uma infraestrutura de rede adequada para que estes filtros sejam válidos para qualquer máquina que esteja solicitando um e-mail de um servidor externo.

Quando se pensa em como embutir um servidor proxy dentro de uma infraestrutura de rede, deve-se levar em consideração alguns fatores importantes, nos quais estarão descritos no próximo subcapítulo.

### 6.4.3 Como Utilizar o Servidor Proxy em uma Rede Local

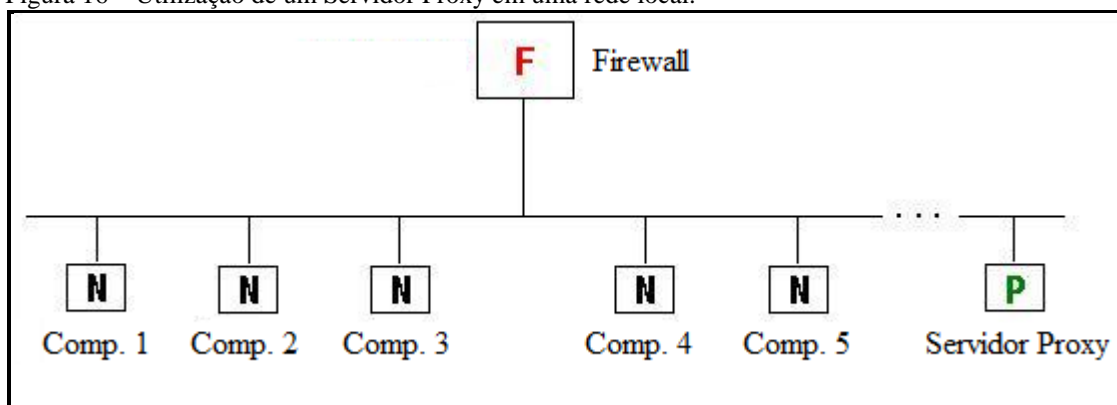
Devido à alta exposição de informações pessoais na Internet, qualquer pessoa que tenha um e-mail cadastrado pode ser vítima de mensagens não solicitadas, mais conhecidas como Spiced Ham (SPAM).

Com a popularização dos correios eletrônicos estas mensagens ficaram mais conhecidas e vulneráveis. Geralmente os *spammers*, nome dado aos criadores dos *spam's*, usam programas que facilitam ou automatizam a obtenção de e-mails e envio em grande quantidade.

Foi pensando nesta recepção indesejada de e-mails que os servidores proxy foram criados, para garantir ao máximo a segurança das informações que entram na caixa de correio dos usuários e podem atuar como um servidor que armazena dados em forma de cache que evita o acesso ao dispositivo de armazenamento, que pode ser demorado, armazenando os dados em meios de acesso mais rápidos.

Para utilização de um servidor proxy em uma rede local, é preciso saber onde e como implantar este servidor, afim de assegurar da melhor forma os dados que estão entrando na rede local. Desta forma, apresenta-se na Figura 16 uma das diversas formas de embutir um servidor proxy em uma rede local.

Figura 16 – Utilização de um Servidor Proxy em uma rede local.



Fonte: Do Autor.

Analisando a estrutura apresentada na figura acima, o servidor proxy estará em um *Switch* onde estão os demais computadores da rede. No *Firewall* será configurado para permitir a saída para a Internet quando utilizar as portas relacionadas ao protocolo IMAP, apenas o IP do computador onde está o servidor proxy, afim de garantir que as demais

máquinas da rede não tenham acesso aos correios eletrônicos se não estiverem configurados para se conectarem no servidor proxy.

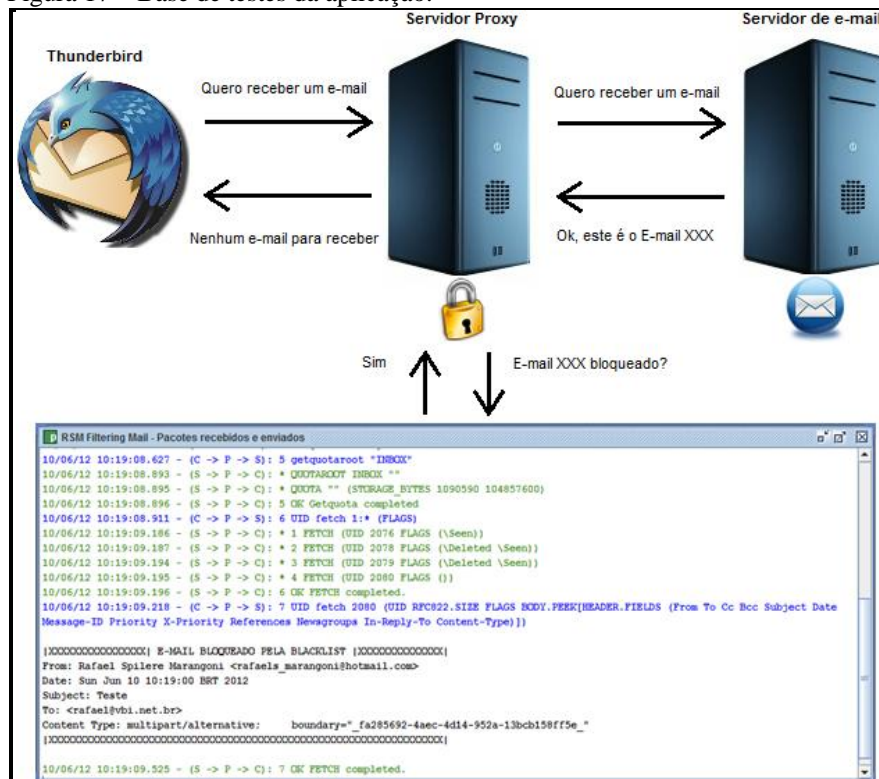
## 6.5 EXECUÇÃO DOS TESTES E RESULTADOS

A execução de testes em softwares é de suma importância e pode ser visto como uma parcela do processo de qualidade de software. Os problemas só serão encontrados se o produto for testado incansavelmente, podendo ainda assim estar sujeito a falhas, lembrando que não se pode garantir que todo software funcione perfeitamente, sem a presença de erros.

Alguns testes foram importantes para a solução de problemas relacionados ao próprio servidor de e-mail IMAP (sem a intervenção do servidor proxy) ainda nos testes iniciais. Quando é enviado um e-mail somente com o título, ou seja, sem conteúdo, ele nunca é recebido pelo servidor IMAP, como se não tivesse sido enviado. Este problema aconteceu nos servidores IMAP do provedor Terra e do provedor Gmail. Nos demais servidores de e-mail IMAP testados o e-mail foi enviado e recebido corretamente.

O cenário de testes da aplicação será apresentado na Figura 17, demonstrando como foram realizados os testes e de que forma a estrutura foi montada para a realização dos mesmos.

Figura 17 – Base de testes da aplicação.



Fonte: Do Autor.

Com base neste cenário, foram realizados os testes da aplicação. Como o principal objetivo do protótipo é bloquear o recebimento de e-mails indesejados, este cenário foi montado como primeiro objetivo a ser alcançado, sabendo que o não bloqueio do e-mail é o fluxo normal entre cliente de e-mail e servidor de e-mail, não precisando fazer nenhuma consistência no sistema.

Nos testes iniciais, quando o Thunderbird estabelecia a conexão com o servidor proxy e enviava o primeiro pacote de identificação, houve um problema na resposta do servidor proxy para o Thunderbird. Isto acontecia porque o servidor proxy aguardava o pacote completo do servidor IMAP para enviar ao Thunderbird. Após um longo período de análise, identificou-se que não estava correta esta forma de envio, ou seja, tudo o que o servidor proxy recebia ele deveria enviar ao Thunderbird, independente se o pacote estivesse ou não completo. Assim sendo, a troca de mensagens entre as aplicações foi corrigida e começou-se a explorar as melhores práticas de envio e recebimento de pacotes através do protocolo.

Após a conexão correta do Thunderbird com o servidor proxy, a aplicação apresentou um baixo rendimento quanto a recuperação de e-mails no servidor IMAP devido à criação de *Threads* de forma errada, ocupando muito processamento e afetando totalmente a performance do Thunderbird.

Com o ajuste nas *Threads* que melhoraram o desempenho do servidor proxy, começaram a ser realizados os testes para recebimento de e-mails simples através do proxy. Neste momento identificou-se um problema relacionado à perda de conexão que ocorria esporadicamente, afetando os e-mails que estavam sendo recebidos e inviabilizando a utilização do servidor proxy, uma vez que a conexão lógica fora quebrada. Posteriormente ajustou-se este problema com o desmembramento de classes no código-fonte, separando a classe de conexão do servidor e do cliente.

Quando se começou o recebimento de e-mails completos, explorou-se àqueles e-mails onde as informações eram mais elaboradas, por exemplo, e-mails que continham anexos. Apesar da preocupação com o desempenho nestes casos, esta etapa não consumiu maiores problemas, pois a perda de conexão que antes era frequente, agora não ocorria mais devido ao ajuste descrito no parágrafo acima.

Já nos testes de filtragem dos e-mails configurados na BlackList ou WhiteList, os problemas foram muitos por estarem relacionados a dados que não podem ser alterados de maneira alguma. Esta etapa de testes foi a mais complicada por ser o objetivo do protótipo.

Ao abstrair as informações para o filtro e usá-las nas comparações com os cabeçalhos das mensagens, houve uma rejeição do Thunderbird nos e-mails que haviam sido

separados para enviá-los, ou seja, e-mails não bloqueados. Para solucionar este problema, foi reescrita a classe de filtragem para identificar o erro. Por fim, analisando os log's no console da IDE Eclipse, que será mencionado nos subcapítulos posteriores, identificou-se a falta da mensagem “*XXX OK FETCH completed.*”, afetando significativamente a forma de tratamento dos e-mails recebidos pelo Thundebird. Este problema acontecia geralmente quando havia mais de um e-mail para ser recebido, sendo um bloqueado e um não bloqueado por exemplo.

Na primeira versão do servidor proxy, era enviada uma mensagem ao cliente de e-mail informando quantos e-mails haviam sido bloqueados pelo filtro. Porém isto foi retirado do código por uma simples razão que impedia o recebimento dos demais e-mails não bloqueados. Para o Thunderbird apresentar uma mensagem de erro, a resposta do servidor deve estar relacionada ao erro “*XXX NO Error in IMAP command received by server.*”, mas como neste caso em específico haviam e-mails que deveriam ser recebidos, se o servidor proxy enviasse uma mensagem de erro para o Thunderbird, ele estaria manipulando a resposta vinda do servidor de e-mail, não correspondendo com a realidade. Descobrimos esta falha, retirou-se do servidor proxy esta mensagem que era enviada. De qualquer forma, os e-mails que estão sendo bloqueados são mostrados na tela do *Servidor*, tornando aquela mensagem de erro redundante.

Como teste final, foi explorada a limitação de conexão dos clientes de e-mail no servidor proxy, tendo um bom desempenho por parte do proxy por trabalhar de forma consistente com as conexões lógicas estabelecidas.

Conforme observado, os testes foram os principais agentes causadores dos problemas encontrados, tornando-se uma atividade importante no desenvolvimento do software. Todos os testes realizados onde ocorriam respostas inesperadas, foram ajustados para que não ocorram novamente.

O principal resultado que deveria ser atingido, que era a conexão com um cliente de e-mail e filtro dos e-mails capturados através do protocolo IMAP, foi alcançado com êxito dando uma vasta ideia do funcionamento de um cliente de e-mail quando conectado a um servidor de e-mail, inclusive demonstrando todos os pacotes que trafegam neste protocolo.

### **6.5.1 Ferramentas para Depuração de Problemas no Desenvolvimento do Protótipo**

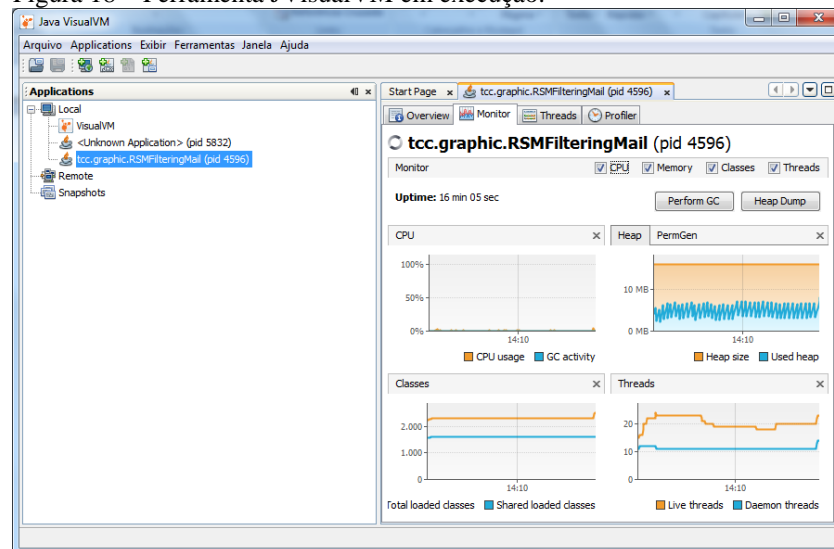
Qualquer ferramenta desenvolvida apresentará dificuldades e problemas que só serão corrigidos com testes e depurações. No desenvolvimento, as dificuldades foram sendo ajustadas e remodeladas perante sua importância no funcionamento da ferramenta. Neste

cenário, algumas ferramentas foram importantes para o bom entendimento dos problemas ocorridos. Estas ferramentas serão mostradas nos próximos subcapítulos juntamente com a descrição do problema e a forma como a ferramenta auxiliou na solução deste problema.

#### 6.5.1.1 JVisualVM

Esta ferramenta permite gerar e analisar dados contidos na pilha, rastrear vazamentos de memória, monitorar processos executados simultaneamente, monitorar o coletor de lixo do Java (Garbage Collector) e visualizar a performance da memória e CPU. Na Figura 18 a ilustração da ferramenta JVisualVM.

Figura 18 – Ferramenta JVisualVM em execução.



Fonte: Do Autor.

A aplicação em um determinado momento começou a operar utilizando 100% da memória disponível, travando as demais aplicações que estavam rodando simultaneamente com o servidor proxy. A ferramenta mostrou exatamente qual era a *Thread* que estava consumindo esta porcentagem total de CPU: era o processo que buscava os e-mails no servidor de e-mail. Ajustando este processo na aplicação, a obtenção de e-mails ficou mais rápida e sem travamentos, melhorando o desempenho do servidor proxy.

#### 6.5.1.2 Console Depurador da IDE Eclipse

Neste console podem ser apresentados quaisquer log's para depuração de problemas. Ao serem apresentados os log's, faz-se a análise dos dados impressos no console e

identifica-se possíveis problemas ou “gargalos” na aplicação que podem causar maiores problemas futuros.

A linguagem Java permite alterar a saída destes dados, trocando a saída do console para qualquer janela criada pelo programador. Isto foi feito para o protótipo, setando uma janela padrão para saída de dados. O comando utilizado para apresentar partes do código no console é *System.out.print()* e *System.err.print()*. No caso do protótipo, sempre que foram utilizados estes comandos a saída padrão era a janela do servidor proxy que mostra todo o fluxo de troca de mensagens entre cliente de e-mail, servidor proxy e servidor de e-mail, como pode ser visualizado nas Figuras 14 e 15 do subcapítulo *6.4.2 Identificando as Mensagens Apresentadas no Servidor Proxy*.

## 7 CONCLUSÃO

Este trabalho de pesquisa apresentou um estudo sobre uma aplicação já consolidada que são os Correios Eletrônicos, criada com a finalidade de facilitar a troca de mensagens entre pessoas e baixar os custos se comparado com um correio convencional. Foram abordados os temas relacionados à pesquisa detalhando principalmente os protocolos de comunicação SMTP, POP e IMAP, sendo este último o escolhido para o desenvolvimento do protótipo do servidor proxy. Além dos estudos apontados, foi desenvolvido um protótipo de um servidor proxy com filtragem de e-mails configurados através desta aplicação.

O uso do protocolo IMAP deu-se pela comunicação dinâmica que é estabelecida com o cliente de e-mail, atualizando em tempo real as informações no servidor de e-mail IMAP, ou seja, qualquer ação realizada no cliente de e-mail será automaticamente atualizada no servidor IMAP, garantindo de forma consistente que os dados sejam recebidos pelo servidor.

O avanço da Internet proporcionou um vasto leque de possibilidades no envio e recebimento de e-mails. Alguns usuários de correio eletrônico aproveitaram este crescimento para ampliar os seus serviços, explorando este avanço de forma positiva. Mas juntamente com os pontos positivos andam os negativos, que neste caso estão relacionados aos *hacker's*, usuários que usam as mensagens para invadir a privacidade ou enviar informações infectadas para os demais usuários dos correios eletrônicos.

Os filtros de e-mail podem ser muito utilizados na prevenção de mensagens com vírus, SPAM's, e-mails indesejados, possíveis domínios suspeitos, entre outras funcionalidades, descartando estas mensagens quando os filtros forem configurados corretamente.

O protótipo teve como principal objetivo o filtro de e-mails configurados na própria aplicação, incluindo sua utilização em um cliente de e-mail, chegando a um resultado satisfatório e de fácil compreensão quanto ao bloqueio dos e-mails, mostrando-os no *Servidor* da aplicação.

O cliente de e-mail Thunderbird, usado como base de testes do servidor proxy, foi explorado de ponta a ponta para que nenhum teste deixasse de ser realizado, fazendo com que o servidor proxy aguentasse as requisições de um cliente de e-mail da altura do Thunderbird.

Através dos testes pode-se entender a abrangência de um cliente de e-mail, desde sua *interface* até seus diversos recursos disponibilizados para facilitar a comunicação do usuário através de e-mails. No uso do servidor proxy foram analisados os diversos comandos

enviados pelo Thunderbird afim de identificar qual seria o comando correto para utilização no momento do filtro.

Em relação a trabalhos futuros, sugere-se a implementação da busca em um servidor BlackList externo, disposto na Internet. Há diversos destes servidores disponíveis, precisando somente fazer o estudo do seu protocolo de comunicação para recuperar a lista de domínios desta BlackList. Outra possibilidade é a implementação de um certificado SSL para o servidor proxy desenvolvido, garantindo a segurança das informações e confiança dos usuários de que seus dados pessoais estarão trafegando criptografados. Ainda, pode-se explorar a utilização do servidor proxy nos demais clientes de e-mail existentes, como por exemplo o Microsoft Outlook Express que obteve um desempenho significativo usando o proxy, porém com algumas inconsistências devido à forma de tratamento das mensagens recebidas por este cliente de e-mail. Com alguns ajustes no servidor proxy, este poderá ser utilizado sem nenhum problema também com o Outlook Express.

## REFERÊNCIAS

ABROAD DESIGN. **AD Mail Box Manager**. Disponível em: <<http://www.abroaddesign.com/mbm/mbm.zip>>. Acesso em: 02 nov. 2011.

Agência Brasil . **Venda de banda larga popular, a R\$ 35 mensais, deve começar em três meses**. Disponível em: <<http://www.rnp.br/newsgen/9710/n5-2.html>>. Acesso em: 03 set. 2011.

Base de conhecimento. **Mozilla Thunderbird**. Disponível em: <<http://unihostbrasil.com.br/apps/clientes/knowledgebase/6/Mozilla-Thunderbird.html>>. Acesso em: 05 jun. 2012.

DOTZLER, Asa. **Thunderbird 1.0 reaches 1.000.000 downloads in Just 10 days**. Disponível em: <<http://weblogs.mozillazine.org/asa/archives/007119.html>>. Acesso em: 05 maio 2012.

BATTISTI, Júlio. **SQL Server 2000: Administração e Desenvolvimento: Curso Completo**. 2. ed. Rio de Janeiro: Axcell Books, 2001.

BOCHENSKI, Bárbara. **Implementando sistemas cliente/servidor de qualidade**. São Paulo: Makron Books, 1995.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do usuário**. Rio de Janeiro: Campus, 2000.

CASAD, Joe; WILLSEY, Bob. **Aprenda em 24 horas TCP/IP**. Rio de Janeiro: Campus, 1999.

COMER, Douglas E. **Interligação em rede com TCP/IP: Princípios, protocolos e arquitetura**. Rio de Janeiro: Campus, 1998

COMER, D. E. **Redes de Computadores e Internet**., 2ª Edição, Bookman, 2001.

GHIZZO, Érico. **Internet**. Disponível em: <<http://www.atica.com.br/internet/index.asp>>. Acesso em: 03 nov. 2011.

HM SOFT. **HM NIS EDIT**. Disponível em:  
<<http://prdownloads.sourceforge.net/hmne/nisedit2.0.1.exe?download>>. Acesso em:  
01 nov. 2011.

HOLLINGWORTH, Jarrod. et al. **C++Builder5 Developer's Guide**. Indianapolis:  
Sams, 2000.

INCREDIMAIL. **Incredimail XE**. Disponível em:  
<<http://www.incredimail.com/download.asp?type=installer&setupid=22000007&langName=portuguese>>. Acesso em: 29 out. 2011.

INTERNET WORLD STATS. **Internet Growth History Statistics**. Disponível em:  
<<http://www.internetworldstats.com/index.html>>. Acesso em: 04 nov. 2011.

KUROSE, J.F. e ROSS, K. W. **Redes de Computadores e a Internet: Uma nova abordagem**, Addison Wesley, São Paulo, 2003.

MACEDO, Alvaro. **Geração Y e a extinção dos e-mails até 2015**. Disponível em:  
<<http://www.observatoriodaimprensa.com.br/news/view/geracao-y-e-a-extincao-dos-emails-ate-2015>>. Acesso em: 12 set. 2011.

MARTINS, Rodrigo. **E-mail ainda é mais popular que Twitter e Facebook, aponta estudo**. Disponível em: <<http://blogs.estadao.com.br/rodrigo-martins/2011/08/10/e-mail-ainda-e-mais-popular-que-twitter-e-facebook-aponta-estudo/>>. Acesso em: 09 set. 2011.

MCFEDRIES, Paul. **Guia incrível do Correio Eletrônico**. Editora: Makron Books, 2008.

MENDES, Antonio. **Arquitetura de Software: desenvolvimento orientado para arquitetura**. Editora Campus. Rio de Janeiro - RJ, 2002.

MICROSOFT. **Microsoft Help Workshop**. Disponível em:  
<<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=34D35502-4DE9-4676-952C-34CC7F64F098>>. Acesso em: 02 nov. 2011.

MICROSOFT. **Outlook Express 6**. Disponível em:  
<<http://www.microsoft.com/windows/ie/downloads/default.aspx>>. Acesso em: 04 jun. 2012.

MITCHELL, Mark; OLDHAM Jeffrey; SAMUEL, Alex. **Advanced Linux Programming**. First Edition. New Riders Publishing, 2001. Disponível em: <<http://www.advancedlinuxprogramming.com>>. Acesso em: 10 out. 2011.

MYERS, J.; ROSE, M. **Post Office Protocol - Version 3**. Request for Comments: 1939. 1996.

MURHAMMER, Martin W. et al. **TCP/IP Tutorial e Técnico**. São Paulo: Makron Books, 2000.

NULLSOFT. **NSIS**. Disponível em: <<http://prdownloads.sourceforge.net/nsis/nsis202.exe?download>>. Acesso em: 02 nov. 2011.

PETERSON, L. L. E DAVIE, B. S. **Computer Networks: A systems approach, Morgan Kaufmann, 2<sup>nd</sup> Edition**, 2000.

RNP – Rede Nacional de Ensino e Pesquisa. **É Vantajoso Utilizar o Protocolo IMAP**. Disponível em: <<http://www.rnp.br/newsgen/9710/n5-2.html>>. Acesso em: 07 set. 2011.

SOARES, Fernando. **Redes de Computadores: das Lans, Mans e Wans às Redes ATM**. 2 ed. Campus, 1995.

TANENBAUM, Andrew S. **Redes de computadores**. 3. ed. Rio de Janeiro: Campus, 1994.

TANENBAUM, Andrew S. **Redes de computadores**. 4 ed. Rio de Janeiro: Campus, 2003.

TANENBAUM, Andrew S.; WOODHULL, Albert S. **Sistemas Operacionais: Projeto e Implementação**. 2 ed. Porto Alegre: Bookman, 2000.

TANENBAUM, Andrew S. **Redes de computadores**. 3. ed. Rio de Janeiro: Campus, 1994.

TANENBAUM, Andrew S. **Redes de computadores**. 4 ed. Rio de Janeiro: Campus, 2003.

TANENBAUM, Andrew S.; WOODHULL, Albert S. **Sistemas Operacionais: Projeto e Implementação**. 2 ed. Porto Alegre: Bookman, 2000.

TEIXEIRA JUNIOR, José Helvécio. **Redes de computadores: serviços, administração e segurança**. São Paulo: Makron Books, 1999.

VASKEVITCH, David. **Estratégia Cliente/Servidor: um guia para a reengenharia da empresa**. São Paulo: Berkeley, 1995.

## APÊNDICE A – ARTIGO

# Filtering Mail: Ferramenta para Filtragem de E-mail Capturado Através do Protocolo de Comunicação IMAP4

Rafael Spilere Marangoni<sup>1</sup>, Rogério Antônio Casagrande<sup>1</sup>

<sup>1</sup>Curso de Ciência da Computação - Universidade do Extremo Sul Catarinense (UNESC)

Caixa Postal 3.167 – 88.806-000 – Criciúma – SC - Brasil

rafaels\_marangoni@hotmail.com, roc@unesc.net

***Abstract.** This article describes the development of an e-mail filtering prototype using the IMAP e-mail server. The prototype provides easy operability in the filter configuration, using a simple and objective interface to facilitate user operation. The prototype works with multiple users and all commands sent and received by e-mail client and e-mail server can be seen in the Application Server. This prototype was developed with base on various tests related to information exchange between client and server e-mail based on IMAP4 communication protocol.*

***Resumo.** Este artigo descreve o desenvolvimento de um protótipo de filtragem de e-mail usando o servidor de e-mail IMAP. O protótipo provê fácil operabilidade na configuração dos filtros, usando uma interface simples e objetiva, a fim de facilitar a operação do usuário. O protótipo trabalha com múltiplos usuários e todos os comandos enviados e recebidos pelo cliente de e-mail e servidor de e-mail podem ser vistos no Servidor da aplicação. Este protótipo fora desenvolvido com base em diversos testes relacionados à troca de informações entre cliente e servidor de e-mail baseado no protocolo de comunicação IMAP4.*

## 1. Introdução

Os computadores, inicialmente, eram máquinas caríssimas que centralizavam o processamento de dados em um único ponto. Após a redução de custos do hardware e inclusão dos microcomputadores no cenário da informática, a estrutura passou a ser distribuída e não mais centralizada (SOARES, 1995). Com a distribuição destes recursos aumentou-se os equipamentos, gerando duplicação desnecessária de recursos de hardware e software, melhorando o processamento de dados, mas ocupando maior espaço físico. Foi nesta deficiência que surgiram as redes de computadores, interligando estes equipamentos de processamento de dados com a finalidade de compartilhamento de recursos. Hoje o maior exemplo de conectividade através de rede de computadores é a Internet que é um amplo sistema de comunicação que conecta muitas redes de computadores, nomeada núcleo da comunicação moderna (KUROSE, 2003).

O usuário que utiliza um cliente de e-mail sob o protocolo Internet Message Access Protocol Version 4 (IMAP4) tem algumas vantagens perante os demais protocolos de recebimento de e-mail, porém não tem o controle específico sobre determinadas funções, como por exemplo, não receber e-mails de determinado domínio.

Neste trabalho será desenvolvida uma ferramenta que permite ao usuário manipular e gerenciar as suas mensagens, fazendo um filtro para que identifique quais os e-mails que o usuário deseja realmente receber, descartando os demais.

## 2. Redes de Computadores

Conceitualmente se define uma rede como computadores interligados entre si e que possam compartilhar seus dados. Toda e qualquer rede, independentemente de tamanho e sofisticação, descende desse modelo básico e não restrito somente a microcomputadores (TANENBAUM, 1994).

### 2.1. Modelo Cliente-Servidor

A arquitetura cliente-servidor é hoje uma das arquiteturas mais utilizadas em ambientes corporativos, substituindo a arquitetura muito rígida que eram os sistemas envolvendo mainframes1.

Esta tecnologia tem uma arquitetura na qual o processamento da informação é dividido em módulos ou processos distintos. Um processo é responsável pela manutenção da informação (servidores) e o outro responsável pela obtenção dos dados (clientes). Os processos clientes enviam pedidos para o processo servidor e este por sua vez processa e envia os resultados dos pedidos (BATTISTI, 2001).

Geralmente, os serviços oferecidos pelos servidores dependem de processamento específico que só eles podem fazer. O processo cliente, por sua vez, fica livre para realizar outros trabalhos. A interação entre os processos é uma troca cooperativa, como exemplificada na Figura 1, em que o cliente é o ativo e o servidor reativo, ou seja, o cliente requisita uma operação, o servidor processa e responde ao cliente.

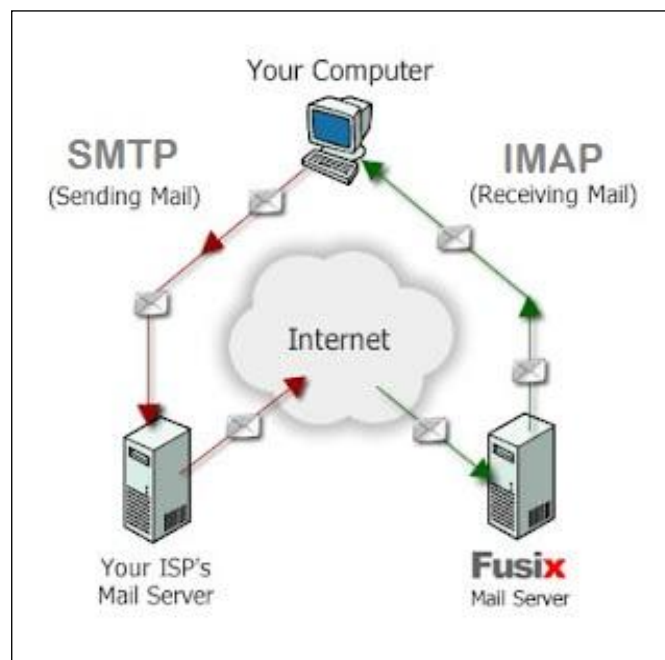


Figura 1. Exemplo de troca de mensagens na arquitetura cliente e servidor.

### 3. Correio Eletrônico

Tornou-se popular devido às vantagens oferecidas em termos de permitir comunicação além das barreiras do tempo e do espaço. Hoje em dia, desde que o usuário possua um microcomputador e linha telefônica, este usuário pode, fazendo uso de um provedor de Internet pago ou gratuito, enviar e receber mensagens vinte e quatro horas por dia, sete dias da semana para qualquer parte do mundo. É tão rápido e fácil mandar um e-mail para qualquer parte do mundo quanto para o seu colega da sala ao lado. Você pode enviar um e-mail sempre que tiver vontade e ele estará ao alcance do destinatário sempre que ele quiser lê-lo. Na Figura 2 é apresentado um servidor de correio eletrônico em uma rede corporativa.

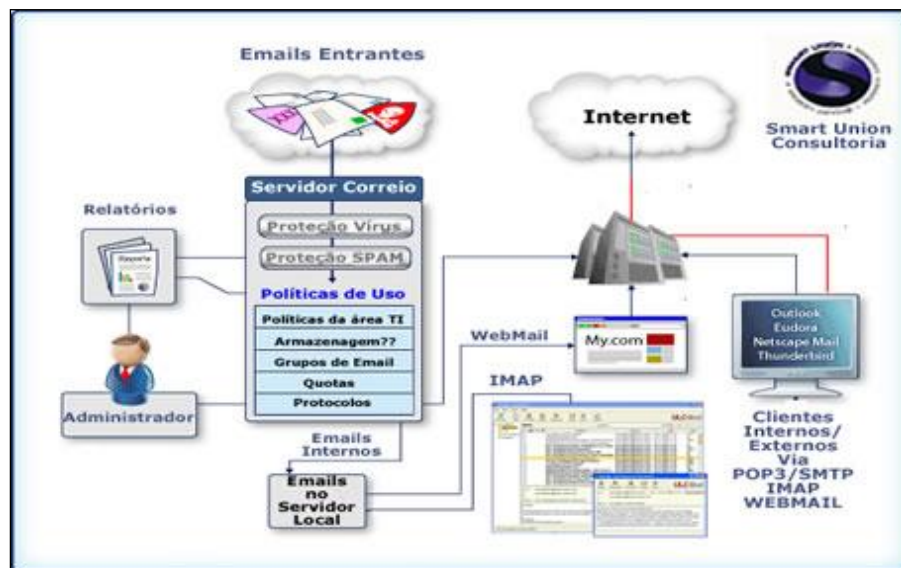


Figura 2. Servidor de correio corporativo.

### 4. Metodologia

A metodologia para o desenvolvimento do protótipo baseou-se nas seguintes etapas: seleção dos recursos de software; modelagem e desenvolvimento da ferramenta para filtragem de e-mails; execução de testes para análise dos resultados obtidos.

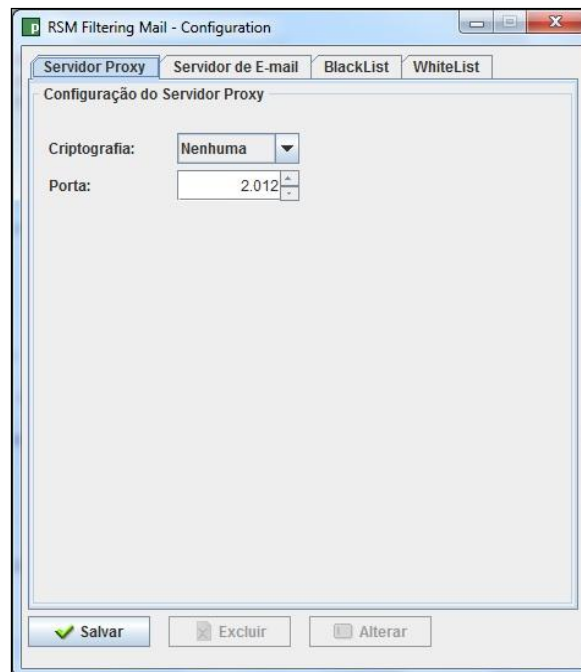
#### 4.1. Recursos Utilizados

Utilizou-se um computador com as configurações básicas para o acesso a Internet que esteja instalado o sistema operacional Windows, a ferramenta Eclipse para o desenvolvimento da ferramenta na linguagem de programação Java e o aplicativo Mozilla Thunderbird que é um cliente de e-mail para envio e recebimento de e-mails em um computador desktop, podendo gerenciar múltiplas contas de e-mail.

Como recurso humano houve o auxílio de um professor do curso de Ciência da Computação com conhecimento na área de Redes de Computadores e foram utilizados recursos gratuitos ou fornecidos pela universidade para os devidos fins.

## 5. Utilização do Protótipo

Quando uma aplicação é desenvolvida para uso generalizado onde os usuários não possuem grande entendimento de como tudo aquilo foi desenvolvido, deve-se tomar cuidado para não dificultar o entendimento desta aplicação ao usuário final. A utilização de uma *interface* simples e objetiva facilita o seu uso e mantém uma relação amigável entre aplicação e usuário. Na Figura 3 visualiza-se a tela principal de configuração do protótipo, fornecendo 4 abas de configuração: *Servidor Proxy* para configuração do servidor do protótipo, *Servidor de E-mail* para configuração do servidor de e-mail que será utilizado, *BlackList* para configuração da lista negra, ou seja, e-mails que serão bloqueados por constarem nesta lista e *WhiteList* para configuração da lista branca, ou seja, e-mails que serão bloqueados por não constarem nesta lista.



**Figura 3. Tela principal de configuração do protótipo.**

Além da tela de configuração há o Servidor do protótipo, que abre automaticamente uma conexão local para que clientes de e-mail se conectem e a conexão com o servidor IMAP4.

Com o servidor da aplicação aberto e ativo, conforme apresentado na Figura 4, deve-se configurar o cliente de e-mail (como base de estudos foi utilizado o Thunderbird) para se conectar neste servidor. Para isto é inserido o nome do computador e porta onde está aberto o servidor proxy. Neste protótipo foram utilizados dois computadores contidos na mesma rede local com IP's diferentes, mas conforme a modelagem do protocolo TCP/IP, nada impede que um computador que esteja localizado fora desta rede local possa também utilizá-lo, precisando apenas que o administrador da rede faça com que o servidor proxy esteja visível em ambiente externo.

```

10/06/12 10:19:08.627 - (C -> P -> S): 5 getquotaroot "INBOX"
10/06/12 10:19:08.893 - (S -> P -> C): * QUOTAROOT INBOX ""
10/06/12 10:19:08.895 - (S -> P -> C): * QUOTA "" (STORAGE_BYTES 1090590 104857600)
10/06/12 10:19:08.896 - (S -> P -> C): 5 OK Getquota completed
10/06/12 10:19:08.911 - (C -> P -> S): 6 UID fetch 1:* (FLAGS)
10/06/12 10:19:09.186 - (S -> P -> C): * 1 FETCH (UID 2076 FLAGS (\Seen))
10/06/12 10:19:09.187 - (S -> P -> C): * 2 FETCH (UID 2078 FLAGS (\Deleted \Seen))
10/06/12 10:19:09.194 - (S -> P -> C): * 3 FETCH (UID 2079 FLAGS (\Deleted \Seen))
10/06/12 10:19:09.195 - (S -> P -> C): * 4 FETCH (UID 2080 FLAGS ())
10/06/12 10:19:09.196 - (S -> P -> C): 6 OK FETCH completed.
10/06/12 10:19:09.218 - (C -> P -> S): 7 UID fetch 2080 (UID RFC822.SIZE FLAGS BODY.PEEK[HEADER.FIELDS (From To Cc Bcc Subject Date
Message-ID Priority X-Priority References Newsgroups In-Reply-To Content-Type)])

|XXXXXXXXXXXXXXXXXXXX| E-MAIL BLOQUEADO PELA BLACKLIST |XXXXXXXXXXXXXXXXXXXX|
From: Rafael Spilere Marangoni <rafaels_marangoni@hotmail.com>
Date: Sun Jun 10 10:19:00 BRT 2012
Subject: Teste
To: <rafael@vbi.net.br>
Content Type: multipart/alternative; boundary=" fa285692-4aec-4d14-952a-13bcb158ff5e_"
|XXXXXXXXXXXXXXXXXXXX|

10/06/12 10:19:09.525 - (S -> P -> C): 7 OK FETCH completed.

```

Figura 4. E-mail bloqueado pela BlackList no servidor proxy do protótipo.

## 5.1. Servidor Proxy em uma Rede Corporativa

Devido à alta exposição de informações pessoais na Internet, qualquer pessoa que tenha um e-mail cadastrado pode ser vítima de mensagens não solicitadas, mais conhecidas como Spiced Ham (SPAM).

Com a popularização dos correios eletrônicos estas mensagens ficaram mais conhecidas e vulneráveis. Geralmente os spammers, nome dado aos criadores dos spam's, usam programas que facilitam ou automatizam a obtenção de e-mails e envio em grande quantidade.

Foi pensando nesta recepção indesejada de e-mails que os servidores proxy foram criados, para garantir ao máximo a segurança das informações que entram na caixa de correio dos usuários e podem atuar como um servidor que armazena dados em forma de cache que evita o acesso ao dispositivo de armazenamento, que pode ser demorado, armazenando os dados em meios de acesso mais rápidos.

Para utilização de um servidor proxy em uma rede local, é preciso saber onde e como implantar este servidor, afim de assegurar da melhor forma os dados que estão entrando na rede local. Desta forma, apresenta-se na Figura 5 uma das diversas formas de embutir um servidor proxy em uma rede local.

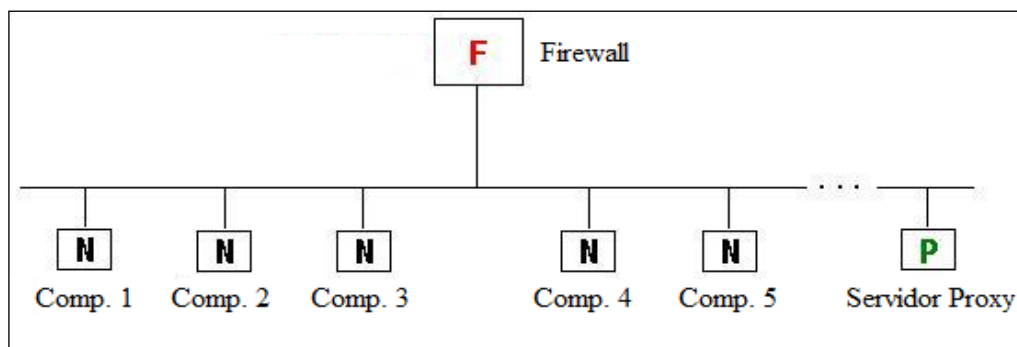


Figura 5. Utilização de um servidor proxy em uma rede local.

Analisando a estrutura apresentada na figura acima, o servidor proxy estará em um Switch onde estão os demais computadores da rede. No Firewall será configurado para permitir a saída para a Internet quando utilizar as portas relacionadas ao protocolo IMAP, apenas o IP do computador onde está o servidor proxy, afim de garantir que as demais máquinas da rede não tenham acesso aos correios eletrônicos se não estiverem configurados para se conectarem no servidor proxy.

## 7. Referências

KUROSE, J.F. e ROSS, K. W. **Redes de Computadores e a Internet**: Uma nova abordagem, Addison Wesley, São Paulo, 2003.

MCFEDRIES, Paul. Guia incrível do Correio Eletrônico. Editora: Makron Books, 2008.

RNP – Rede Nacional de Ensino e Pesquisa. É Vantajoso Utilizar o Protocolo IMAP. Disponível em: <<http://www.rnp.br/newsgen/9710/n5-2.html>>. Acesso em: 07 set. 2011.

TANENBAUM, Andrew S.; WOODHULL, Albert S. **Sistemas Operacionais: Projeto e Implementação**. 2 ed. Porto Alegre: Bookman, 2000.

VASKEVITCH, David. **Estratégia Cliente/Servidor: um guia para a reengenharia da empresa**. São Paulo: Berkeley, 1995.