



**UNIVERSIDADE SALVADOR - UNIFACS  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
REDES DE COMPUTADORES**

**HÉLIO OLIVEIRA QUEIROZ JUNIOR**

***THREADMAP* – UMA FERRAMENTA PARA  
MAPEAMENTO E VISUALIZAÇÃO DE *THREADS* DE  
MENSAGENS ELETRÔNICAS**

Salvador  
2006

**HÉLIO OLIVEIRA QUEIROZ JUNIOR**

***THREADMAP* – UMA FERRAMENTA PARA  
MAPEAMENTO E VISUALIZAÇÃO DE *THREADS* DE  
MENSAGENS ELETRÔNICAS**

Dissertação apresentada à Universidade Salvador,  
como parte das exigências do Curso de Mestrado  
Profissional em Redes de Computadores, área de  
concentração em Tecnologia Web e Aplicações  
Distribuídas, para obtenção do título de Mestre.

Orientador: Prof. Dr. Manoel Gomes de  
Mendonça Neto

Salvador  
2006

# **TERMO DE APROVAÇÃO**

**HÉLIO OLIVEIRA QUEIROZ JUNIOR**

## ***THREADMAP – UMA FERRAMENTA PARA MAPEAMENTO E VISUALIZAÇÃO DE THREADS DE MENSAGENS ELETRÔNICAS***

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Redes de Computadores, Universidade Salvador – UNIFACS, pela seguinte banca examinadora:

**Prof. Manoel Gomes de Mendonça Neto - Orientador**

Doutor em Ciência da Computação. Universidade de Maryland Em College Park, UMCP, Estados Unidos.

Universidade Salvador - UNIFACS

**Prof. Antônio Lopes Apolinário Júnior**

Doutor em Engenharia de Sistemas e Computação. Universidade Federal do Rio de Janeiro, UFRJ, Brasil.

Universidade Estadual de Feira de Santana - UEFS

**Laís Nascimento Salvador**

Doutora em Engenharia Elétrica. Universidade de São Paulo, USP, Brasil

Universidade Salvador (UNIFACS)

**Salvador, 04 de maio de 2006**

## **AGRADECIMENTOS**

Em primeiro lugar, agradeço a Deus, por guiar o meu caminho nesta difícil jornada.

Agradeço a minha esposa, Mônica, pela compreensão, carinho e constante incentivo, aos meus pais, Hélio e Antônia e a toda a minha família pela ajuda, apoio e companheirismo.

Aos colegas Manoel Marques Neto e Carol Passos por me permitirem aprender com a sua experiência e pela ajuda neste trabalho.

Ao professor e orientador Manoel Mendonça pela indicação do caminho, pela oportunidade de compartilhar o seu conhecimento e pela atenção dedicada.

À Secretaria da Fazenda do Estado da Bahia, por permitir a aplicação deste trabalho e conseqüente realização do estudo de caso, em especial a Geraldo Bahiense, superintendente da Superintendência de Gestão Fazendária (SGF).

Aos colegas da Gerência de Tecnologia (GETEC), pelas discussões e contribuições sobre este trabalho e pela disponibilidade de participação no estudo de caso.

Por fim, gostaria de agradecer a todos que de alguma forma contribuíram para que este trabalho fosse realizado.

## RESUMO

O uso do correio de eletrônico está cada dia mais difundido dentro das organizações, sendo utilizado não apenas para a comunicação de informativos e recados, mas também para desenvolver discussões não-presenciais e assíncronas entre membros de uma equipe. Um *Thread* é um conjunto de mensagens trocadas entre indivíduos, tipicamente em torno de um mesmo assunto, em uma ordem cronológica, através de respostas e encaminhamentos, simulando uma conversa presencial. Através do mapeamento de *Threads* é possível identificar características tais como a duração típica dessas conversas, a quantidade de mensagens trocadas e a quantidade de participantes envolvidos. Além disso, pode-se analisar o fluxo de troca de informações dentro de uma equipe de trabalho. Esta dissertação propõe uma solução para o mapeamento de *threads* baseado na interceptação direta de mensagens no servidor de correio eletrônico e nos dados contidos no cabeçalho das mensagens. A principal contribuição desta solução é evitar a perda de trechos do *threads*, permitindo um mapeamento global das mensagens trocadas. Isto contrapõe as soluções existentes atualmente que produzem uma visão particular da conversa realizada, por serem baseadas na análise do assunto da mensagem e em cópias das caixas postais dos usuários. Além da proposta de solução, esta dissertação apresenta uma implementação dessa solução utilizando um servidor de mercado, uma interface que permite a visualização dos *threads* mapeados juntamente com as respectivas mensagens trocadas, e um estudo de caso. O estudo de caso envolveu um grupo de 19 pessoas e foi realizado em um ambiente organizacional real, a Gerência de Tecnologia da Secretaria da Fazenda do Estado da Bahia. Ele demonstrou a viabilidade e utilidade da solução proposta.

**Palavras-chave:** *Thread*, Correio Eletrônico, Mensagem Eletrônica, Visualização, Mapeamento.

## ABSTRACT

The usage of e-mail is widespread in modern organizations. They are used not only for sending information, but also to develop full fledged non-presential and asynchronous discussions. These discussions are commonly referred as threads. Formally, a Thread is defined as a set of electronic message exchanged between individuals, typically around a given subject, in chronological order, through replies and forwards, simulating a presential conversation. Through thread mapping, it is possible to analyze features like typical duration of threads, number of exchanged messages, and number of participants involved in a thread. One can also use threads to analyze the flow of information inside working teams. This thesis proposes a solution for thread mapping based on intercepting messages directly at the e-mail server and on the analysis of the header of these messages. Its main contribution is to avoid the loss of thread parts that occurs when messages are analyzed at the users' postal boxes, usually based on the messages subject. For this reason, the approach proposed here allows for a broader mapping of the exchanged messages in an organizational unit. Beyond the proposal, this thesis presents an implementation of the approach for a well-know e-mail server, an interface for visualizing the mapped threads and their messages, and a case study on the usage of the implemented approach. The case study involved 19 people and was run in a real organization, the Technology Sector of the State of Bahia Revenue Services. The study showed that the proposed solution was viable and useful.

**Keywords:** Thread, Electronic Mail, Electronic Message, Visualization, Mapping.

## LISTA DE FIGURAS

Figura 1 - Fluxo geral do correio eletrônico .....	18
Figura 2 - Exemplo de <i>thread</i> .....	26
Figura 3 - Exemplo de <i>thread</i> em formato de árvore.....	32
Figura 4 - Quatro visões do mesmo <i>thread</i> .....	33
Figura 5 - Visualização tradicional de <i>Thread</i> .....	36
Figura 6 - Visualização <i>Narrow Tree</i> .....	37
Figura 7 - Visualização <i>Tree Table</i> .....	38
Figura 8 - <i>ReMail Prototype</i> .....	39
Figura 9 - Visualização <i>TimeLine</i> .....	40
Figura 10 - Modelo Misto de Visualização .....	41
Figura 11 - Representação da cronologia e relacionamento no <i>Thread Arcs</i> .....	42
Figura 12 - Representação de escala e compactação do <i>Thread Arcs</i> .....	43
Figura 13 - <i>Threads</i> cheios e estreitos.....	43
Figura 14 - Marcação de mensagens no <i>thread</i> .....	44
Figura 15 - Processo de captura da mensagem no servidor de correio eletrônico .....	50
Figura 16 - Modelo de dados do <i>ThreadMap</i> .....	51
Figura 17 - Simulação de conversação com <i>ThreadMap</i> – Mensagem original .....	56
Figura 18 - Simulação de conversação com <i>ThreadMap</i> – Mensagem 1 .....	57
Figura 19 - Simulação de conversação com <i>ThreadMap</i> – Mensagem 2 .....	58
Figura 20 - Simulação de conversação com <i>ThreadMap</i> – Mensagem 3 .....	59
Figura 21 - Simulação de conversação com <i>ThreadMap</i> – Mensagem 4 .....	60
Figura 22 - Simulação de conversação com <i>ThreadMap</i> – Mensagem 5 .....	61
Figura 23 - Simulação de conversação com <i>ThreadMap</i> – Árvore Resumo do <i>Thread</i> .....	62
Figura 24 - Diagrama de disparo dos eventos do MS Exchange Server .....	67
Figura 25 - Interface do <i>Component Service</i> .....	74
Figura 26 - Processo de Importação de componente <i>event sink</i> .....	75
Figura 27 - Interface principal do <i>ThreadViewer</i> .....	77
Figura 28 - Visualização das mensagens do <i>thread</i> no <i>ThreadViewer</i> .....	79
Figura 29 - Distribuição de <i>threads</i> x quantidade de mensagens .....	91
Figura 30 - Distribuição de <i>threads</i> x quantidade de participantes .....	92

Figura 31 - Distribuição de <i>threads</i> x duração .....	93
Figura 32 - <i>ThreadViewer</i> – Visão dos <i>threads</i> coletados na SEFAZ .....	95
Figura 33 - <i>ThreadViewer</i> – Detalhamento de um <i>thread</i> coletados na SEFAZ.....	95



## LISTA DE QUADROS

Quadro 1 - Exemplo de interação entre cliente e servidor SMTP .....	22
Quadro 2 - Cabeçalho de mensagens eletrônicas.....	24
Quadro 3 - Vantagens e desvantagens das abordagens de mapeamento.....	28
Quadro 4 - Itens do cabeçalho utilizados por Palme (1998) no mapeamento do <i>thread</i> .....	29
Quadro 5 - Análise comparativa entre técnicas de mapeamento .....	35
Quadro 6 - Qualidades de uma ferramenta de visualização de <i>threads</i> .....	45
Quadro 7 - Resumo comparativo das técnicas de visualização de <i>threads</i> .....	46
Quadro 8 - Algoritmo da solução <i>ThreadMap</i> .....	48
Quadro 9 - Fonte das informações colhidas sobre as mensagens .....	49
Quadro 10 - Detalhamento do modelo de dados do <i>Threadmap</i> .....	53
Quadro 11 - Análise comparativa entre técnicas de mapeamento e <i>ThreadMap</i> .....	55
Quadro 12 - Eventos e métodos para implementação de <i>Event Sinks</i> .....	66
Quadro 13 - Formato da marca inserida no corpo da mensagem .....	69
Quadro 14 - Parâmetros passados para a função <i>ProcessBeginSave</i> .....	69
Quadro 15 - Ligação da mensagem que disparou o evento.....	70
Quadro 16 - Acesso aos dados da mensagem.....	70
Quadro 17 - Fonte de informações das mensagens interceptadas.....	71
Quadro 18 - Localização da marca no corpo da mensagem.....	71
Quadro 19 - Inclusão da marca no corpo da mensagem.....	72
Quadro 20 - Alteração da mensagem trocada.....	73
Quadro 21 - Script para associação do <i>event sink</i> à caixa postal do usuário.....	75
Quadro 22 - Características dos <i>threads</i> e as fontes de informações.....	77
Quadro 23 - Resumo comparativo das técnicas de visualização de <i>threads</i> e o <i>ThreadViewer</i> .....	80
Quadro 24 - Dimensões e agrupamento utilizados para análise dos dados.....	90

## LISTA DE TABELAS

Tabela 1 - Distribuição de <i>threads</i> x quantidade de mensagens.....	90
Tabela 2 - Distribuição de <i>threads</i> x quantidade de participantes.....	92
Tabela 3 - Distribuição de <i>threads</i> x duração.....	93

## LISTA DE ABREVIATURAS E SIGLAS

ADO	<i>ActiveX Data Objects</i>
ASCII	<i>American Standard Code for Information Interchange</i>
COM	<i>Component Object Model</i>
DLL	<i>Dynamic Link Library</i>
IETF	<i>Internet Engineering Task Force</i>
IMAP	<i>Internet Message Access Protocol</i>
JDBC	<i>Java Database Connectivity</i>
JUNG	<i>Java Universal Network/Graph</i>
MIME	<i>Multipurpose Internet Mail Extensions</i>
MTA	<i>Mail Transfer Agent</i>
MUA	<i>Mail User Agent</i>
OLE DB	<i>Object Linking and Embedding DataBase</i>
POP3	<i>Post Office Protocol 3</i>
RFC	<i>Request for Comments</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
URL	<i>Uniform Resource Locator</i>

# SUMÁRIO

<b>CAPÍTULO 1 - INTRODUÇÃO.....</b>	<b>12</b>
1.1 OBJETIVOS .....	13
1.2 ABORDAGEM ADOTADA .....	14
1.3 ESTUDO DE CASO.....	15
1.4 PRINCIPAIS CONTRIBUIÇÕES DO TRABALHO.....	15
1.5 ORGANIZAÇÃO DA DISSERTAÇÃO .....	16
<b>CAPÍTULO 2 - CORREIO ELETRÔNICO E MAPEAMENTO DE <i>THREADS</i> ..</b>	<b>17</b>
2.1 FUNCIONAMENTO GERAL.....	17
2.2 SMTP – <i>SIMPLE MAIL TRANSFER PROTOCOL</i> .....	18
2.2.1 Terminologia SMTP .....	19
2.2.2 Principais comandos SMTP .....	19
2.3 POP3 – <i>POST OFFICE PROTOCOL</i> .....	22
2.4 IMAP – <i>INTERNET MESSAGE ACCESS PROTOCOL</i> .....	22
2.5 <i>HEADERS</i> (CABEÇALHO) DAS MENSAGENS .....	23
2.6 <i>THREADS</i> DE MENSAGENS.....	25
2.6.1 Técnicas para mapeamento de <i>Threads</i> .....	26
2.6.2 Visualização de <i>Threads</i> de mensagens.....	36
2.7 CONSIDERAÇÕES FINAIS.....	46
<b>CAPÍTULO 3 - SOLUÇÃO <i>THREADMAP</i> .....</b>	<b>47</b>
3.1 FUNCIONAMENTO E ALGORITMO .....	47
3.2 MODELAGEM DE DADOS.....	50
3.3 BENEFÍCIOS DA SOLUÇÃO .....	53
3.4 SIMULAÇÃO DE CONVERSAÇÃO COM <i>THREADMAP</i> .....	55
3.5 ALTERNATIVAS DE IMPLEMENTAÇÃO .....	62
3.6 CONSIDERAÇÕES FINAIS.....	63
<b>CAPÍTULO 4 - IMPLEMENTAÇÃO <i>THREADMAP</i>.....</b>	<b>64</b>
4.1 AMBIENTE TECNOLÓGICO UTILIZADO .....	64
4.2 <i>MS EXCHANGE SERVER</i> .....	65

4.2.1	<i>Exchange Store Events</i> .....	65
4.2.2	Exchange OLEDB Provider (ExOLEDB) .....	68
4.3	DETALHES DA IMPLEMENTAÇÃO .....	68
4.4	ROTEIRO DE INSTALAÇÃO DO <i>EVENT SINK</i> .....	73
4.5	INTERFACE DE VISUALIZAÇÃO - <i>THREADVIEWER</i> .....	76
4.5.1	Arquitetura do <i>ThreadViewer</i> .....	80
4.6	CONSIDERAÇÕES FINAIS.....	83
<b>CAPÍTULO 5</b>	<b>- ESTUDO DE CASO .....</b>	<b>84</b>
5.1	METODOLOGIA UTILIZADA .....	84
5.2	PROTOCOLO DE PREPARAÇÃO .....	86
5.2.1	Organização.....	86
5.2.2	População envolvida e escopo.....	87
5.3	APLICAÇÃO DO ESTUDO DE CASO .....	88
5.4	ANÁLISE DOS RESULTADOS .....	88
5.4.1	Protocolo de análise .....	89
5.4.2	Resultados obtidos .....	90
5.5	AVALIAÇÃO DOS RESULTADOS OBTIDOS .....	93
5.6	CONSIDERAÇÕES FINAIS.....	96
<b>CAPÍTULO 6</b>	<b>- CONCLUSÃO.....</b>	<b>97</b>
6.1	CONTRIBUIÇÕES .....	98
6.2	LIMITAÇÕES DO TRABALHO .....	99
6.3	TRABALHOS FUTUROS .....	100
6.4	CONSIDERAÇÕES FINAIS.....	101

## CAPÍTULO 1 - INTRODUÇÃO

É indiscutível a presença do correio eletrônico no dia a dia de estudantes, profissionais e pesquisadores. Diariamente, os usuários dessa ferramenta recebem dezenas e até mesmo centenas de mensagens. Whittaker e Sidner (1996) consideram que os milhares de usuários na *web* gastam parte significativa de seu tempo produtivo utilizando o correio eletrônico.

Whittaker e Sidner (1996) consideram o correio eletrônico a aplicação de maior sucesso no mundo cliente servidor e revelam que estudos apontam essa ferramenta como um dos fatores que proporcionaram o crescimento de organizações distribuídas, permitindo a comunicação entre pessoas geograficamente distantes, de uma forma ágil e rápida.

Moody (2002) apresenta um estudo onde o correio eletrônico aparece como a ferramenta de comunicação mais utilizada nos Estados Unidos e Canadá, com 97% dos trabalhadores utilizando-o todos os dias ou vários dias por semana.

A troca de mensagens, utilizando o correio eletrônico, permite aos seus usuários a participação em diversas discussões sobre temas variados, de forma simultânea. Essa troca de mensagens entre usuários, normalmente sobre o mesmo assunto é chamada de *Thread*.

Lewis e Knowles (1997) definem um *thread* como uma conversação entre duas ou mais pessoas através da troca de mensagens. Para Palme (1999), *threads* são conjuntos de mensagens que são, direta ou indiretamente, respostas a outras mensagens.

Segundo Nenkova e Bagga (2003), os *threads* de correio eletrônico são a forma mais comum de representação e armazenamento de grupos de discussão ou listas de distribuição, principalmente em função do grande volume de mensagens trocadas nestes serviços.

As definições apresentadas por Lewis e Knowles (1997) e Palme (1999) sobre a simulação de uma conversa presencial via correio eletrônico, juntamente com a consideração de Nenkova e Bagga (2003) sobre a representação para grupos de discussão e listas de distribuição apontam para possibilidade de registro, mapeamento, e posterior visualização e análise desses *threads*.

Analisando do ponto de vista da organização, o mapeamento e visualização dessa troca de mensagens podem ser utilizados para identificar processos ainda não formalizados, para verificar como se dá o fluxo de mensagens dentro de um grupo de trabalho e para auxiliar no mapeamento de redes sociais e na sua interação.

Para os usuários de correio eletrônico, a possibilidade de visualizar de forma rápida, integrada e consistente um *thread* minimiza bastante o tempo necessário para o usuário conseguir se posicionar e interagir dentro da conversa em andamento, principalmente em função do grande volume de mensagens de correio eletrônico trocadas diariamente.

## 1.1 OBJETIVOS

Esta dissertação possui quatro objetivos principais:

- Propor uma solução para o mapeamento de *threads* de mensagens eletrônicas (*ThreadMap*), baseada na interceptação de mensagens no servidor de correio eletrônico e utilizando as informações contidas no cabeçalho da mensagem para registro destes dados;
- Apresentar uma implementação da solução *ThreadMap*, utilizando um servidor de correio eletrônico de mercado, o MS Exchange Server;
- Realizar um estudo de caso em uma organização com objetivo de validar a implementação realizada e possibilitar a obtenção de seus resultados através da

caracterização dos *threads* tipicamente realizados, identificando duração, quantidade média de mensagens trocadas e quantidade de participantes envolvidos;

- Apresentar uma interface de visualização (*ThreadViewer*) para os *threads* mapeados através do *ThreadMap* e os detalhes das suas respectivas mensagens trocadas.

## 1.2 ABORDAGEM ADOTADA

Visando atender aos objetivos descritos na Seção 1.1, será apresentada a solução de mapeamento *ThreadMap*, baseada na funcionalidade de interceptação de mensagens, disponível em servidores de correio eletrônico do mercado, detalhando o seu funcionamento através da descrição do algoritmo e do modelo de dados propostos e respectivos benefícios alcançados.

Para validar a proposta apresentada, será detalhada também uma implementação do *ThreadMap* utilizando como servidor de correio eletrônico o MS Exchange Server e os eventos síncronos disponíveis no *Exchange Store Events*, que permitiram a captura e mapeamento dos *threads* no momento em que as mensagens são trocadas entre os usuários.

Com o objetivo de permitir a visualização dos *threads* mapeados e respectivas mensagens trocadas, será apresentada uma implementação de interface (*ThreadViewer*), baseada nos dados capturados pelo *ThreadMap*.

Além disso, teremos a apresentação de um estudo de caso realizado nas instalações da Secretaria da Fazenda do Estado da Bahia, com a equipe de Tecnologia da Diretoria de Tecnologia de Informação, com o objetivo de validar a ferramenta construída e os dados obtidos nesta captura.



### 1.3 ESTUDO DE CASO

O estudo de caso apresentado no Capítulo 5 foi realizado envolvendo as 19 pessoas da Gerência de Tecnologia (GETEC) da Diretoria de Tecnologia de Informação (DTI) da Secretaria da Fazenda do Estado da Bahia.

A captura das mensagens trocadas entre os membros da equipe da GETEC foi realizada no período de 20 de fevereiro de 2006 a 16 de março de 2006. Durante este período, foram interceptadas 7.902 mensagens, agrupadas em 7.288 *threads*.

A partir dos dados coletados, foi possível perceber que apenas 11,25% das mensagens capturadas pertenciam a *threads* que efetivamente simulavam uma conversação, pois possuíam mais de uma mensagem trocada. Desta forma, as análises realizadas com os dados coletados tomaram como base os *threads* com quantidade igual ou superior a duas mensagens trocadas na conversação.

O estudo de caso apresentou um elevado número de *threads* realizados com apenas duas mensagens, curtos e envolvendo um número pequeno de participantes.

Através da realização deste estudo de caso, foi possível demonstrar a viabilidade de uso da ferramenta *ThreadMap*, visto que foi aplicado em uma organização real e sem impacto na operação do serviço de correio eletrônico.

### 1.4 PRINCIPAIS CONTRIBUIÇÕES DO TRABALHO

A principal contribuição deste trabalho é oferecer uma solução que permita o mapeamento geral do *thread*, minimizando a possibilidade de captura parcial da conversação realizada entre os participantes.

A possibilidade de perdas de trechos do *thread* é comum nas soluções disponíveis, baseadas na análise das caixas postais dos usuários de correio eletrônico. Realizando o mapeamento do *thread* a partir de um repositório com as mensagens de um usuário, existe a

possibilidade de perda de trechos da conversação, visto que apenas serão capturadas as mensagens em que houve a participação do usuário em questão.

O fato de o *ThreadMap* ser uma solução que pode ser implementada em qualquer servidor de correio eletrônico com suporte a interceptação de eventos de envio e recebimento de mensagens representa uma contribuição secundária e permite a sua evolução e adaptação a diferentes ambientes.

## 1.5 ORGANIZAÇÃO DA DISSERTAÇÃO

Essa dissertação foi organizada em seis capítulos. Após esta introdução, será apresentado um capítulo sobre a revisão bibliográfica e fundamentação teórica acerca do tema dessa dissertação. Os próximos três capítulos tratam da solução *ThreadMap*, de sua implementação utilizando um servidor de correio eletrônico de mercado e da realização do estudo caso. Finalmente, temos a conclusão do trabalho e referências utilizadas. Dessa forma, o trabalho possui a seguinte estrutura:

- CAPÍTULO 1 – introdução com exposição da motivação, objetivos, contribuições e estrutura do documento;
- CAPÍTULO 2 – fundamentação teórica e revisão bibliográfica sobre correio eletrônico e técnicas para mapeamento e visualização de *threads*;
- CAPÍTULO 3 – apresentação da solução conceitual do *ThreadMap*;
- CAPÍTULO 4 – implementação do *ThreadMap* utilizando o MS Exchange Server e da ferramenta de visualização *ThreadViewer*;
- CAPÍTULO 5 – apresentação do estudo de caso realizado e resultados obtidos;
- CAPÍTULO 6 – conclusão do trabalho e indicação de trabalhos futuros;

## CAPÍTULO 2 - CORREIO ELETRÔNICO E MAPEAMENTO DE *THREADS*

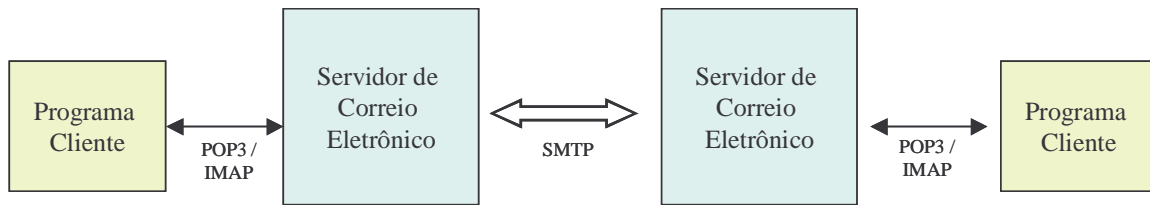
Este capítulo traz uma revisão bibliográfica sobre o correio eletrônico, apresentando as suas características gerais de funcionamento, os protocolos utilizados e enfatizando os aspectos relativos às técnicas atualmente disponíveis para o mapeamento de *threads* de mensagens eletrônicas e as suas alternativas de visualização propostas pelos diversos autores.

### 2.1 FUNCIONAMENTO GERAL

O correio eletrônico é uma ferramenta que permite a comunicação e a troca de informações entre as pessoas, utilizando uma rede de computadores como meio de transporte. Para utilizar essa ferramenta, cada usuário é identificado por um endereço virtual, seguindo o padrão: *endereço@dominio*.

Para interagir com o serviço de correio eletrônico, um usuário utiliza um programa cliente, que o permite editar o conteúdo da mensagem e enviá-la para um servidor de correio eletrônico. Esse servidor é o responsável por encaminhar essa mensagem até o servidor ou servidores destinatário(s) da mensagem, onde esta será armazenada em uma caixa postal, até que o usuário destinatário possa recuperá-la.

A Figura 1 ilustra o fluxo geral do serviço de correio eletrônico.



**Figura 1 - Fluxo geral do correio eletrônico**

A comunicação entre o programa cliente e os servidores de correio eletrônico é realizada através dos protocolos *Post Office Protocol* (POP3) ou *Internet Message Access Protocol* (IMAP), permitindo o acesso dos usuários às mensagens armazenadas nos servidores de correio eletrônico em suas respectivas caixas postais. A comunicação entre os servidores é realizada através do protocolo *Simple Mail Transfer Protocol* (SMTP), responsável pelo encaminhamento da mensagem ao servidor destinatário.

## 2.2 SMTP – SIMPLE MAIL TRANSFER PROTOCOL

O objetivo do protocolo SMTP é transferir mensagens eletrônicas, com confiança e eficiência, independente dos subsistemas de transmissão, requerendo para isso que exista apenas um canal com um fluxo de dados confiável (NETWORK WORKING GROUP, 2001a).

O *Internet Engineering Task Force* (IETF) é uma comunidade internacional aberta, com a missão de identificar e propor soluções para problemas relacionados à utilização da Internet e propor padronizações de tecnologias e protocolos. As recomendações do IETF são normalmente publicadas como RFCs (*Request for Comments*) (IETF, 2006).

O protocolo SMTP é normalmente utilizado sobre uma rede TCP/IP (*Transmission Control Protocol/Internet Protocol*), e o programa servidor responsável pela sua implementação usa uma porta TCP 25 para a comunicação entre clientes e servidor. A implementação do SMTP segue a especificação definida pela RFC 2821, de abril de 2001.

O TPC/IP forma um grupo de protocolos de comunicação sobre o qual a Internet funciona. Dentre estes, o *Transmission Control Protocol* (TCP) e o *Internet Protocol* (IP) são os protocolos mais importantes.

Como discutido anteriormente, o SMTP é um protocolo apenas responsável pela transmissão das mensagens entre os servidores, não podendo ser utilizado pelos usuários para a manipulação (recepção e envio) dessas mensagens.

### 2.2.1 Terminologia SMTP

Segundo Network Working Group (2001a), o SMTP transporta mensagens que contêm um envelope e conteúdo.

O envelope SMTP consiste de um endereço de origem (emissor) e de um ou mais endereços de destino. O conteúdo possui duas partes: o cabeçalho e o corpo. O cabeçalho forma uma coleção estruturada de pares de campos e valores, em conformidade com a RFC 2822 (*Internet Message Format*). O corpo possui seu conteúdo em texto, padrão *American Standard Code for Information Interchange* (ASCII) ou estruturado seguindo a especificação *Multipurpose Internet Mail Extensions* (MIME) contida na RFC 2045.

Os servidores SMTP, por proverem um serviço de transporte de mensagens eletrônicas, são também chamados de *Mail Transfer Agents* (MTAs) ou agentes de transporte de e-mail. Os softwares responsáveis pelo envio ou recepção das mensagens são também chamados de *Mail User Agents* (MUAs) ou agentes usuários de e-mail.

### 2.2.2 Principais comandos SMTP

- Extended Hello (EHLO) ou Hello (HELO)

Comandos utilizados para que um cliente SMTP possa se identificar junto a um servidor SMTP. Deve ser enviado, junto com o comando, o nome completo do cliente, seguindo o padrão internet: servidor.dominio.

Sintaxe: EHLO servidor.dominio

- MAIL

Comando utilizado para iniciar uma transação de correio, na qual os dados da mensagem de correio são entregues ao servidor SMTP que a armazena na caixa postal do usuário ou providencia a entrega ao próximo servidor SMTP. Como argumento, deve ser passado o endereço completo do emissor da mensagem.

Sintaxe: MAIL FROM: endereço@domínio

- RCPT

Comando utilizado para indicar os receptores da mensagem a ser enviada para o servidor SMTP. O endereço do receptor deve ser passado como argumento do comando.

Sintaxe: RCPT TO: endereço@domínio

- DATA

Comando utilizado para enviar o conteúdo da mensagem. Para indicar o término dos dados, deve ser usada uma linha contendo apenas o caractere “.”.

Sintaxe: DATA

*mensagem*

.

- RSET

Indica que a transação de correio iniciada deve ser abortada. Os dados já recebidos pelo servidor SMTP devem ser descartados.

Sintaxe: RSET

- VRFY

Comando que permite ao cliente solicitar que o servidor verifique a existência de um endereço de correio eletrônico, sem que uma mensagem precise ser enviada.

Sintaxe: VRFY endereço@dominio

- EXPN

Comando que permite ao cliente confirmar com o servidor que um endereço passado como argumento é uma lista de endereços de correio, e se for verdadeiro, que o servidor retorne os membros da lista.

Sintaxe: EXPN endereço@dominio

- QUIT

Comando que solicita o encerramento da transação de correio. O servidor SMTP retorna com um OK e conclui a transação.

Sintaxe: QUIT

A seguir, no Quadro 1, temos um exemplo da interação entre um cliente e um servidor SMTP. No exemplo, C representa o cliente e S o servidor. Para iniciar a conversação, o cliente deve estabelecer uma conexão, por exemplo, através do comando *telnet*.

Exemplo: *telnet www.example.com 25*

```
S: 220 www.example.com ESMTP Postfix
C: HELO domain.com.br
S: 250 Hello domain.com.br
C: MAIL FROM: emissor@domain.com.br
S: 250 Ok
C: RCPT TO: receptor@example.com
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: Subject: Mensagem de teste
C: From: sender@mydomain.com
```

```
C: To: friend@example.com
C:
C: Ola,
C: Isto é um teste.
C: Adeus.
C: .
S: 250 Ok: queued as 12345
C: quit
S: 221 Bye
```

### Quadro 1 - Exemplo de interação entre cliente e servidor SMTP

## 2.3 POP3 – *POST OFFICE PROTOCOL*

O protocolo POP3 é utilizado para realizar o acesso às mensagens de correio eletrônico de um usuário, armazenadas na sua caixa postal. O POP3 está definido na RFC1225 e permite que as mensagens sejam transferidas da caixa postal do usuário para um computador local.

Utilizando uma aplicação cliente (MUA), um usuário se autentica em um servidor de correio eletrônico (MTA) onde está a sua caixa postal e solicita que todas as mensagens sejam transferidas de forma seqüencial, para o seu computador. As mensagens são então apagadas do servidor. Opcionalmente, essas mensagens também podem ser mantidas no servidor. Após a transferência, a conexão entre MUA e MTA é encerrada e o usuário pode ler e manipular as suas mensagens, de forma off-line.

## 2.4 IMAP – *INTERNET MESSAGE ACCESS PROTOCOL*

O *Internet Message Access Protocol* (IMAP) também é utilizado para permitir o acesso dos usuários às mensagens de correio eletrônico, armazenadas no servidor.

A vantagem do uso do protocolo IMAP está no fato de que as mensagens ficam armazenadas no servidor e não são transferidas para o computador local. Dessa forma, elas podem ser acessadas de qualquer computador, inclusive via *webmail*, e possibilita a criação e manipulação de pastas no servidor.

Diferentemente do POP3, o IMAP é um protocolo que exige uma conexão permanente (*on-line*) entre o cliente (MUA) e o servidor (MTA).



As desvantagens do uso do protocolo IMAP são que o volume de mensagens armazenadas nas caixas postais está limitado ao estabelecido pelo servidor de correio eletrônico e na impossibilidade de manipulação off-line das mensagens.

## 2.5 HEADERS (CABEÇALHO) DAS MENSAGENS

As informações contidas no cabeçalho das mensagens de correio eletrônico são definidas conforme especificação da RFC 2076 – *Common Internet Message Headers*.

A seguir, no Quadro 2, são apresentados os principais campos contidos no cabeçalho das mensagens, agrupados por categoria de sua aplicação.

<b>Categoria</b>	<b>Campo do Cabeçalho</b>	<b>Descrição</b>	<b>Observação</b>
Trace (rota)	<i>Received</i>	Rota dos MTAs por onde a mensagem passou.	
	<i>Path</i>	Relação dos MTAs passados	Uso apenas para News.
Formato e Controle	<i>MIME-Version</i>	Indicação de que a mensagem está formatada no padrão MIME e respectiva versão	
	<i>Alternate-Recipient</i>	Indica onde a mensagem deve ser entregue caso não seja possível entregar ao destinatário original	
Indicação do emissor e receptor	<i>From</i>	Autor da mensagem	
	<i>Approved</i>	Nome do moderador do grupo de discussão para quem a mensagem foi enviada.	Campo não padronizado para uso em email.
	<i>Sender</i>	Agente que enviou a mensagem, caso seja diferente do autor ( <i>from</i> )	
	<i>To</i>	Destinatários primários da mensagem	
	<i>CC</i>	Destinatários secundários	
	<i>BCC</i>	Destinatários cujos endereços são serão expostos na mensagem	

<b>Categoria</b>	<b>Campo do Cabeçalho</b>	<b>Descrição</b>	<b>Observação</b>
	<i>Mail-System-Version</i>	Informações sobre o software cliente do emissor da mensagem	
Controle da resposta	<i>Reply-to</i>	Indica qual o endereço padrão do emissor da mensagem, em caso de resposta.	Possui diferentes implementações nos produtos de mercado.
	<i>Errors-To</i>	Contêm o endereço para onde as notificações de erro devem ser encaminhadas	Campo não padronizado.
Identificação da mensagem	<i>Message-id</i>	Identificador único da mensagem	
	<i>In-Reply-To</i>	Referencia o identificador da mensagem para a qual a mensagem corrente é uma resposta.	
	<i>References</i>	Referência a outras mensagens relacionadas	
Campos textuais	<i>Keywords</i>	Palavras-chave utilizadas para pesquisa e posterior recuperação da mensagem	
	<i>Subject</i>	Assunto da mensagem	
	<i>Comments</i>	Comentários na mensagem	
	<i>Summary</i>	Sumário da mensagem	Campo não padronizado para uso em email
Campos de data	<i>Delivery-Date</i>	Data e hora de quando a mensagem foi entregue na caixa postal do usuário	Campo não padronizado.
	<i>Date</i>	Data e hora de quando a mensagem foi enviada.	
	<i>Expires</i>	Sugestão de data e hora de expiração da mensagem	Campo não padronizado para uso em email.
Informação sobre qualidade	<i>Priority</i>	Indicação do nível de prioridade da mensagem	Campo não padronizado.
	<i>Importante</i>	Indicação do nível de importância da mensagem.	Campo não padronizado.

**Quadro 2 - Cabeçalho de mensagens eletrônicas**

## 2.6 THREADS DE MENSAGENS

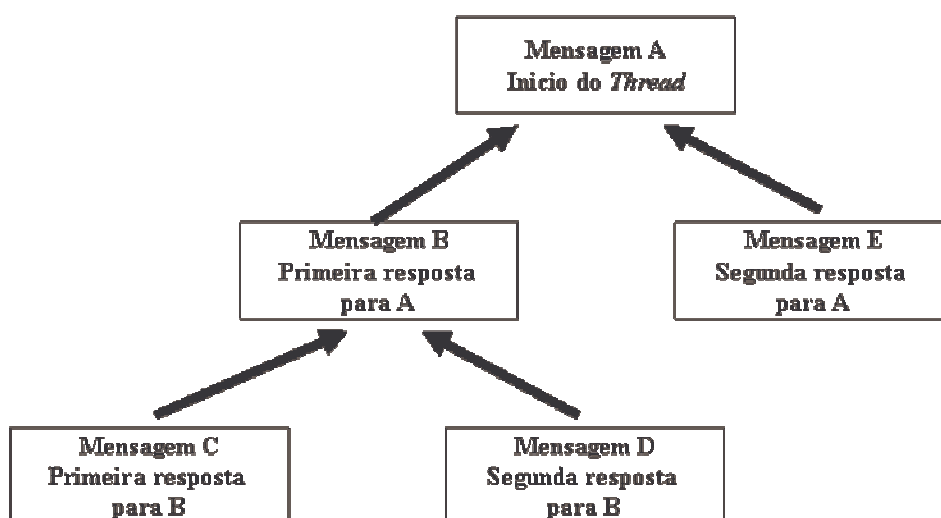
Atualmente, o correio eletrônico é uma das ferramentas de trabalho mais utilizadas dentro das organizações. Muitos dos processos organizacionais hoje são totalmente dependentes da troca de mensagens via correio eletrônico. Desde solicitações de informações e autorizações até decisões importantes estão registradas nas trocas de mensagens, de onde, certamente, muito conhecimento pode ser extraído.

Nesse contexto, diversas mensagens eletrônicas são trocadas entre indivíduos, permitindo que estes estejam participando, de uma forma simultânea, de diversas conversas ou discussões que estão ocorrendo ao mesmo tempo, entre grupos distintos de pessoas e de uma forma assíncrona.

As mensagens trocadas de forma eletrônica simulam uma conversa presencial, com a vantagem de que toda ela pode ser armazenada de forma persistente, servindo como uma base de conhecimento explícito ou meramente um registro da execução de um processo organizacional.

Um *thread* é um conjunto de mensagens trocadas entre indivíduos, tipicamente em torno de um único assunto, em uma ordem cronológica, simulando uma conversação. Segundo Palme (1998), *threads* são conjuntos de mensagens que são, direta ou indiretamente, respostas a outras mensagens.

De acordo com o conceito de Palme (1998), uma vez que as mensagens de um *thread* são sempre fruto de respostas a outras mensagens, um *thread* pode ser visto como uma árvore, onde a primeira mensagem do *thread* é a raiz da árvore, conforme a Figura 2.



**Figura 2 - Exemplo de *thread***

Nota: Adaptado a partir de Palme (1998)

### 2.6.1 Técnicas para mapeamento de *Threads*

O processo de mapeamento de *threads* de mensagens eletrônicas pode ser realizado através de diferentes algoritmos, a depender dos objetivos desse mapeamento e da fonte de informações, no caso, as mensagens eletrônicas trocadas entre usuários, disponíveis para esse fim.

Quando a técnica de mapeamento de *thread* utiliza a caixa postal de um usuário como fonte de informações, os objetivos estão focados nas necessidades do usuário, principalmente na interação com o grande volume de mensagens na sua caixa postal. Através da visualização de *threads*, torna-se mais fácil localizar a seqüência de mensagens trocadas na conversação, permitindo ao usuário se posicionar na conversa de forma mais rápida.

Nesse caso, objetivos organizacionais como identificação de processos não formais e análise do fluxo de informações dentro do grupo de trabalho são prejudicadas, pois não há como garantir que o usuário tenha participado de todo o *thread* e que ele não tenha excluído nenhuma mensagem da conversação.

Uma alternativa é a utilização de técnicas de mapeamento baseadas no repositório de mensagens trocadas em listas de discussões. Os algoritmos são basicamente os mesmos utilizados no processo de mapeamento usando a caixa postal de um usuário como fonte. Nesse caso, tem-se a garantia de que todo o *thread* é mapeado, mas normalmente esses *threads* são públicos ou muito abrangentes, dificultando o alcance de objetivos organizacionais. Outro aspecto importante é que essa fonte é formal e normalmente associada a um tema específico, o que dificulta o mapeamento de troca de informações dentro de um grupo de trabalho.

Outra alternativa é a interceptação do evento de chegada de uma nova mensagem na caixa postal dos usuários. Essa técnica não tem como restrição os limites da caixa postal de um usuário, nem tão pouco o formalismo de uma lista de discussão. Nessa técnica, o *thread* é mapeado a cada troca de mensagem dentro de um servidor, facilitando o alcance dos objetivos organizacionais já discutidos. Como desvantagem, essa técnica é dependente da existência de um mecanismo de interceptação de eventos no servidor de correio eletrônico utilizado.

O Quadro 3 apresenta um resumo sobre as vantagens e desvantagens de cada uma das alternativas para o mapeamento de *threads*.

Abordagem de Mapeamento	Vantagens	Desvantagens
Caixa postal de usuário	<ul style="list-style-type: none"> <li>* Foco na necessidade do usuário;</li> <li>* Auxílio na interação com o grande volume de mensagens;</li> <li>* Facilita a localização do usuário na conversação em andamento.</li> </ul>	<ul style="list-style-type: none"> <li>* Dificuldade na identificação de processos não formais;</li> <li>* Mapeamento do <i>thread</i> com a visão particular do usuário;</li> <li>* Grande possibilidade de perda de trechos do <i>thread</i>;</li> </ul>
Lista de discussão	<ul style="list-style-type: none"> <li>* Garantia de que todo o <i>thread</i> é mapeado;</li> <li>* Mapeamento <i>thread</i> com visão</li> </ul>	<ul style="list-style-type: none"> <li>* Threads são públicos ou muito abrangentes;</li> <li>* Fonte formal e normalmente</li> </ul>

Abordagem de Mapeamento	Vantagens	Desvantagens
	global;	associada a um tema específico.
Interceptação de mensagem	<ul style="list-style-type: none"> <li>* Garantia de que todo o <i>thread</i> é mapeado;</li> <li>* Mapeamento <i>thread</i> com visão global;</li> <li>* Não possui o formalismo das listas de discussões;</li> <li>* Auxilia na identificação de processos não formais.</li> </ul>	* Dependência de mecanismo de interceptação de mensagens no servidor de correio eletrônico.

**Quadro 3 - Vantagens e desvantagens das abordagens de mapeamento**

Seja qual for a origem da informação, as técnicas de mapeamento de *thread* utilizam informações contidas na mensagem eletrônica. Essas informações podem estar localizadas no corpo ou no cabeçalho da mensagem.

Utilizar as informações contidas no corpo da mensagem permite a identificação e agrupamento de acordo com o assunto tratado na mensagem. Como normalmente as ferramentas cliente de correio eletrônico mantêm o texto original ao enviar uma mensagem resposta, a análise do corpo da mensagem pode indicar a que *thread* ela pertence. Lewis e Knowles (1997) apresentam um estudo onde o *thread* é mapeado utilizando a similaridade de vocabulário existente entre os textos das mensagens trocadas, com uma eficácia de 71% na identificação das mensagens pai. Esse estudo avaliou um conjunto de 2435 mensagens postadas durante um período de seis meses.

Outros autores apontam técnicas de mapeamento baseadas no cabeçalho das mensagens. Por exemplo, Palme (1998) apresenta como alternativa para o mapeamento de *threads* o uso de certas informações contidas no cabeçalho das mensagens eletrônicas (*headers*). Essas informações estão apresentadas no Quadro 4.

Cabeçalho ( <i>header</i> )	Descrição
<i>Message-ID</i>	Especifica um identificador único para a mensagem eletrônica
<i>In-Reply-To</i>	Deve conter o identificador ( <i>message-id</i> ) da mensagem originária, ou seja, da mensagem a qual esta é uma resposta ( <i>reply</i> ).
<i>References</i>	Deve conter uma lista de todos os identificadores de mensagens na cadeia de mensagens trocadas anteriores à mensagem atual.

**Quadro 4 - Itens do cabeçalho utilizados por Palme (1998) no mapeamento do *thread***

Segundo Palme (1998), o uso desses campos contidos no cabeçalho das mensagens eletrônicas permite o mapeamento dos *threads*, desde que o conjunto de mensagens trocadas esteja disponível para análise.

Zawinski (1997) apresenta um algoritmo que também faz uso dos dados contidos no cabeçalho das mensagens eletrônicas para o mapeamento de *threads*. Ele utiliza os campos *In-Reply-To*, *References* e *Subject* (assunto da mensagem) para agrupar as mensagens do mesmo *thread*. No caso da manipulação do assunto da mensagem, é necessária a supressão de prefixos normalmente incluídos pelas ferramentas clientes de correio eletrônico, nas operações de resposta e encaminhamento de mensagens. Esses prefixos são: “*Re:*”, “*Res:*”, “*Fw:*”, “*Fwd:*”, dentre outros.

IMAP Extension Work Group (2004) trata de uma proposta elaborada pelo IETF (*Internet Engineering Task Force*) para extensão do protocolo IMAP para ordenação e agrupamento de *thread*, na execução de buscas de mensagens disponíveis no servidor de correio eletrônico.

Essas extensões complementam o comando SEARCH disponível no protocolo IMAP. Com relação ao agrupamento dessas mensagens em *threads*, essa extensão suporta a implementação de dois algoritmos, definidos pelas palavras reservadas ORDEREDSUBJECT e REFERENCES.

No caso do algoritmo ORDEREDSUBJECT, as mensagens pesquisadas são ordenadas pelo assunto base (campo assunto com a remoção de palavras não significativas) e então pela data de envio. As mensagens são então agrupadas em *threads*, onde cada *thread* possui o mesmo assunto base.

As mensagens do *thread* são então ordenadas por data e a primeira mensagem trocada é considerada a mensagem raiz do *thread*. As demais mensagens são todas consideradas filhas da mensagem raiz, não existindo então mais de um nível no *thread* quando o algoritmo ORDEREDSUBJECT é utilizado. Percebe-se que nessa implementação muitas informações sobre as trocas de mensagens são perdidas.

O algoritmo REFERENCES é baseado na proposta de Zawinski (1997). Esse algoritmo agrupa as mensagens, estabelecendo entre elas um relacionamento pai/filho baseado nas respostas e trocas de mensagens realizadas. Esse relacionamento pai/filho é construído usando dois métodos: reconstruindo a árvore ancestral da mensagem utilizando o campo *references* do seu cabeçalho ou tomando como base o assunto (*subject*) das mensagens trocadas.

No caso no método da reconstrução da árvore ancestral da mensagem, são usadas as seguintes regras:

- Se uma mensagem contém informações no campo *References* do cabeçalho, então esses identificadores de mensagem (*message\_id*) são utilizados como referência;
- Se uma mensagem não contém informações no campo *References* do cabeçalho ou se as informações contidas não forem válidas, então o primeiro identificador de mensagem válido encontrado no campo *In-Reply-To* deverá ser utilizado como referência, ou seja, como pai dessa mensagem.



- Se uma mensagem não contém informações válidas no campo In-Reply-To, então essa mensagem não possui uma referência ancestral, ou seja, não possui um pai.

No caso do método de análise do campo de assunto (*subject*), uma mensagem é considerada uma resposta, ou seja, uma mensagem filha, se o assunto dessa mensagem for igual ao assunto da mensagem anterior (pai) precedida dos prefixos normalmente utilizados pelas ferramentas clientes de correio eletrônico: “*Re:*”, “*Res:*”, “*Fw:*”, “*Fwd:*”.

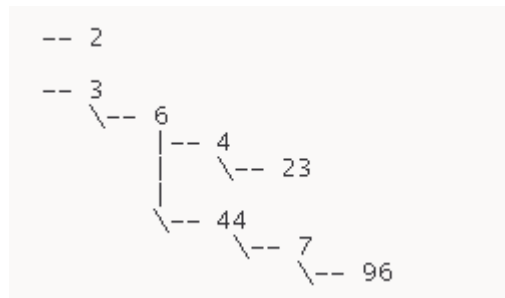
As respostas ao comando *THREAD*, proposto na referência IMAP Extension Work Group (2004) apresentam uma lista de *threads* separados por parênteses. Cada *thread* consiste de uma lista de mensagens, também agrupadas em parênteses.

As mensagens membro de um *thread* são separadas por espaço em branco, indicando sucessivos parentescos de pai e filha. Esse aninhamento continua, podendo o *thread* ser dividido em múltiplos *sub-threads* com os respectivos parentescos aninhados, sem limites de subdivisão ou aninhamento.

*Exemplo: THREAD (2) (3 6 (4 23) (44 7 96))*

Nesse exemplo, temos a ocorrência de 2 *threads*. No primeiro, ele é composto apenas da mensagem com identificador único 2. O segundo *thread* é iniciado com a mensagem 3, seguida da mensagem 6. Na seqüência, o *thread* é dividido em 2 *sub-threads*, um contendo as mensagens 4 e 23 aninhadas (pai e filha) e outro contendo as mensagens 44, 7 e 96 também aninhadas.

A Figura 3 apresenta em formato de árvore os *threads* do exemplo descrito anteriormente.

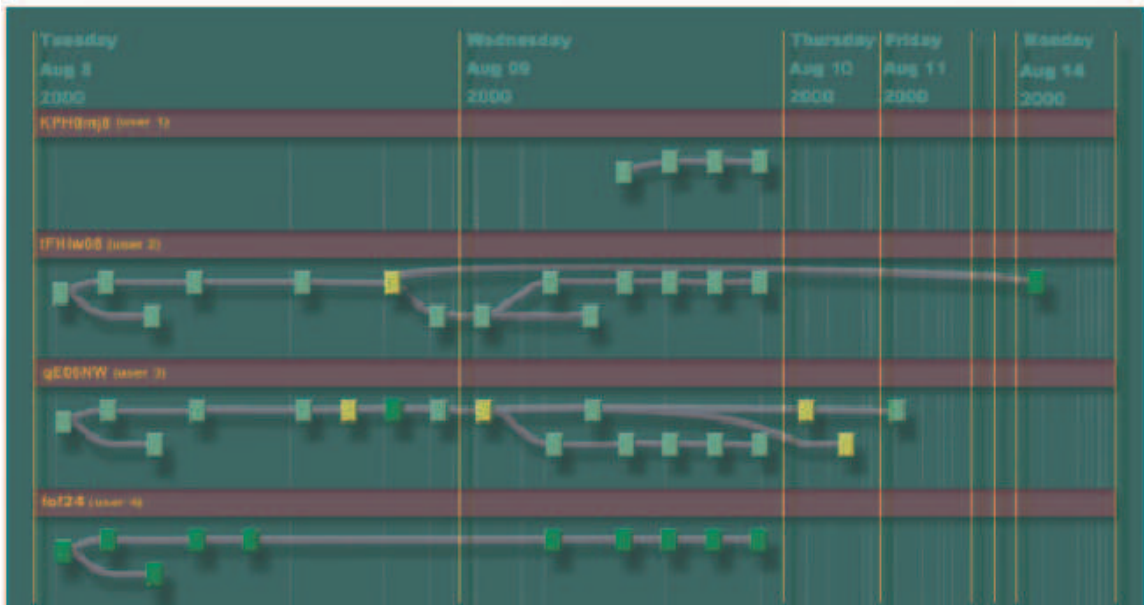


**Figura 3 - Exemplo de *thread* em formato de árvore**

Nota: Extraído de IMAP Extension Work Group (2004)

Nas diversas alternativas apresentadas, a solução possui como requisito a disponibilidade de um repositório com todas as mensagens a serem mapeadas, o que nem sempre é possível obter, uma vez que normalmente, vários usuários participam de uma conversa através do correio eletrônico. Além disso, a fonte de dados é uma foto (*snapshot*) da caixa postal de um usuário e reflete a sua participação, ou seja, a visão que um indivíduo tem sobre a conversa realmente realizada.

Fisher e Moody (2002) apresentam um estudo envolvendo a captura de mensagens trocadas por 74 usuários durante 60 dias, utilizando como fonte para o mapeamento dos *threads* uma foto da caixa postal de cada usuário e evidências como data e assunto para associação dessas mensagens aos *threads*. Além de dados estatísticos como número médio de mensagens trocadas nos *threads*, número de usuários envolvidos e tempo médio de resposta às mensagens trocadas, a identificação de que cada usuário possui uma percepção diferente sobre o mesmo *thread* merece destaque.



**Figura 4 - Quatro visões do mesmo *thread***

Nota: Extraído de Fisher e Moody (2002)

Como pode ser observada, a Figura 4 apresenta a visão de quatro diferentes usuários sobre o mesmo *thread*. Isso se deve ao fato da pesquisa mapear os *threads* a partir de fotos das caixas postais de cada usuário. Percebe-se facilmente que apesar de representar a mesma conversação, cada usuário possui uma visão diferente com relação à duração e pessoas envolvidas.

Pela técnica aplicada, além das diferentes visões, alguns trechos do *thread* podem ter sido perdidos, pois algumas mensagens já podem ter sido excluídas das caixas postais dos envolvidos. Tudo isso dificulta muito a possibilidade de uso dessas informações coletadas para a identificação de processos não formais e análise do fluxo de informações dentro de um grupo de trabalho.

Para fazer uma análise comparativa entre as técnicas vistas anteriormente, serão utilizados alguns critérios, conforme listado a seguir:

- Identificação do *thread* – indica quais os recursos utilizados para a identificação e mapeamento do *thread*. Esses recursos fazem referência às informações contidas nas mensagens trocadas entre os participantes da conversação.

- Fonte de dados – critério que indica qual a fonte de dados utilizada pela técnica de mapeamento, podendo ser a caixa postal dos usuários, o repositório de uma lista de discussão ou as mensagens capturadas no servidor de correio eletrônico.
- Visão do *thread* – indica se a técnica analisada proporciona uma visão global de todo o *thread* ou apenas uma visão individual de cada participante.
- Montagem do *thread* – indica qual o momento, relativo à ocorrência da conversação, em que o *thread* é mapeado. Essa montagem pode ser realizada durante ou após a conversação.

O Quadro 5 apresenta uma análise comparativa entre as técnicas vistas, de acordo com os critérios estabelecidos acima.

<b>Técnica de Mapeamento</b>	<b>Identificação do Thread</b>	<b>Fonte de Dados</b>	<b>Visão do Thread</b>	<b>Montagem do Thread</b>
Lewis (1997)	Baseada na similaridade do vocabulário das mensagens	Caixa postal dos usuários ou repositório de lista de discussão	Visão individual de cada participante	Posterior à conversação
Zawinski (1997)	Baseada em campos do cabeçalho das mensagens ( <i>In-Reply-To</i> , <i>References</i> e <i>Subject</i> )	Caixa postal dos usuários ou repositório de lista de discussão	Visão individual de cada participante	Posterior à conversação
Palme (1998)	Utiliza campos do cabeçalho das mensagens ( <i>Message-Id</i> , <i>In-Reply-To</i> e <i>References</i> )	Caixa postal dos usuários ou repositório de lista de discussão	Visão individual de cada participante	Posterior à conversação
Fisher e Moody	Utiliza o assunto e a data	Caixa postal dos	Visão individual	Posterior à

<b>Técnica de Mapeamento</b>	<b>Identificação do <i>Thread</i></b>	<b>Fonte de Dados</b>	<b>Visão do <i>Thread</i></b>	<b>Montagem do <i>Thread</i></b>
(2002)	da mensagem.	usuários ou repositório de lista de discussão	de cada participante	conversação
IMAP <i>Extension Work Group</i> (2004) ORDERED SUBJECT	Utiliza o assunto e a data da mensagem.	Caixa postal dos usuários ou repositório de lista de discussão	Visão individual de cada participante	Posterior à conversação
IMAP <i>Extension Work Group</i> (2004) REFERENCES	Baseada em campos do cabeçalho das mensagens ( <i>In-Reply-To, References e Subject</i> )  * Técnica proposta por Zawinski (1997)	Caixa postal dos usuários ou repositório de lista de discussão	Visão individual de cada participante	Posterior à conversação

**Quadro 5 - Análise comparativa entre técnicas de mapeamento**

O Quadro 5 mostra que as técnicas de mapeamento apresentam em comum a característica de possuírem como fonte de dados um repositório com as mensagens já trocadas, ou seja, a montagem do *thread* é realizada posteriormente à conversação. Como consequência da fonte utilizada, as técnicas apresentam uma visão particular da conversação, com possibilidade de perda de trechos do *thread*.

Para minimizar os problemas encontrados, uma alternativa de mapeamento de *threads*, baseado na interceptação das mensagens eletrônicas recebidas pelo servidor de correio eletrônico permitiria o mapeamento de todo o *thread*, independente de como essas mensagens estão armazenadas e organizadas em cada caixa postal de usuário. Essa alternativa, entretanto, tem como requisito a captura das mensagens no momento em que são trocadas, não sendo possível a sua recuperação a posteriori.

Nesse caminho de solução surgem dois desafios: a identificação do *thread*, ou seja, a associação correta de cada mensagem recebida ao seu respectivo *thread* e qual a posição

dessa mensagem dentro da cadeia de mensagens trocadas, uma vez que cada mensagem está frequentemente associada (sendo uma resposta ou encaminhamento) a uma mensagem pai.

### 2.6.2 Visualização de *Threads* de mensagens

Existem diversas propostas de visualização de *threads* de mensagens. Em sua grande maioria, são apresentadas como uma evolução das ferramentas cliente de correio eletrônico, com o objetivo de facilitar a participação do usuário na conversação, assim como também melhorar a visualização global do *thread* e sua distribuição no tempo.

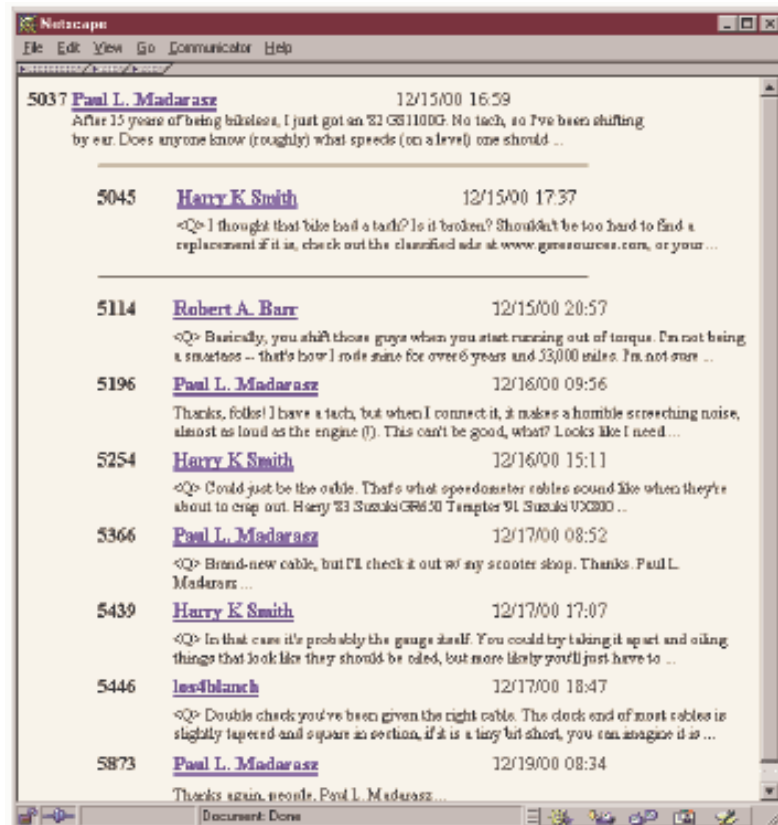
No caso das listas de discussão, é comum a apresentação de *threads* de mensagens sob o formato de árvore, vide Figura 5, onde cada mensagem de resposta é organizada de forma endentada e as mensagens são representadas pelo remetente e horário de postagem, sendo o título um apontador para o corpo da mensagem propriamente dito.

- [How to split the archives into months?](#) Paul F. Williams (Thu Feb 13 2003 - 17:38:34 CST)
  - [Re: How to split the archives into months?](#) kent landfield (Fri Feb 14 2003 - 10:01:08 CST)
    - [Re: How to split the archives into months?](#) Emre Bastuz (Fri Feb 14 2003 - 10:13:34 CST)
  - [Re: How to split the archives into months?](#) William R. Knox (Fri Feb 14 2003 - 10:14:05 CST)
  - [Re: How to split the archives into months?](#) Peter C. McCluskey (Fri Feb 14 2003 - 11:45:44 CST)
- [Hypermail Security Fixes](#) Peter C. McCluskey (Thu Feb 27 2003 - 12:00:47 CST)
  - [Re: Hypermail Security Fixes](#) kent landfield (Thu Feb 27 2003 - 13:35:21 CST)
    - [Problem with 2.1.7 make on hp-ux 10.20](#) Tom von Alten (Fri Feb 28 2003 - 10:37:13 CST)

**Figura 5 - Visualização tradicional de *Thread***

Nota: Extraído de Hypermail (2005)

Uma variação dessa estrutura tradicional de árvore foi apresentada por Newman (2001), chamada de *Narrow Tree*, cuja representação limita o processo de indentação das mensagens do *thread* a apenas um nível. Nessa técnica, as mensagens do *thread* são separadas por linhas divisórias de acordo com a sua posição dentro do *thread*.



**Figura 6 - Visualização *Narrow Tree***

Nota: Extraído de Newman (2001)

Por exemplo, na Figura 6, as mensagens 5045 e 5114 são respostas diretas à mensagem 5037. A mensagem 5196 é uma resposta à mensagem 5114 e a 5254 é uma resposta à 5196. Dessa forma, as mensagens 5114 a 5873 constituem uma cadeia de respostas sucessivas, do tipo pai e filha.

Ainda segundo Newman (2001), essa técnica de visualização é indicada para permitir uma visão geral do *thread*, incluindo o trecho inicial da mensagem na visualização, além de ser mais indicada para *threads* curtos, sem a necessidade de excessiva rolagem de tela.

Newman (2001) também apresenta uma outra técnica de visualização de *threads*, ainda baseada na árvore tradicional, mas em uma estrutura onde as mensagens eletrônicas são organizadas em linhas e colunas, como em uma tabela chamada *TreeTable*.

Na *TreeTable*, cada célula representa uma mensagem do *thread* e as mensagens que são respostas diretas a uma mensagem pai estão organizadas na linha subsequente. Todas as respostas são agrupadas juntas ocupando a mesma área horizontal da mensagem pai.

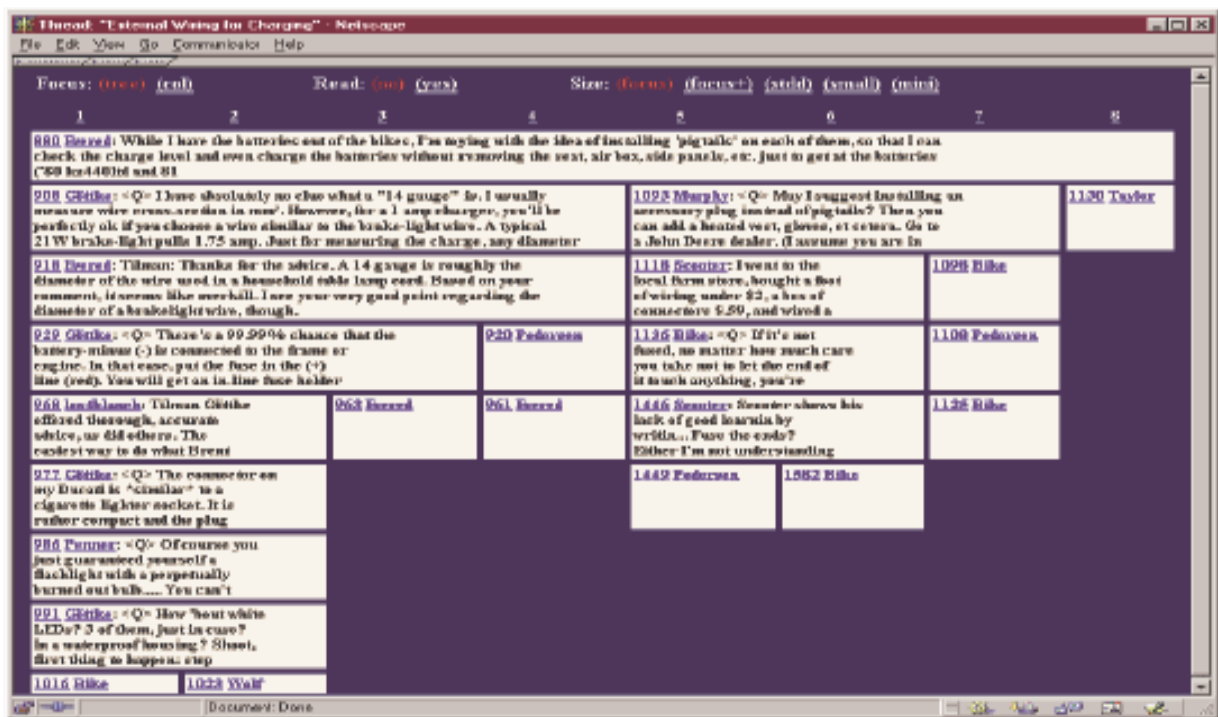


Figura 7 - Visualização *Tree Table*

Nota: Extraído de Newman (2001)

Na Figura 7, as mensagens 908, 1093 e 1130 são respostas diretas à mensagem 880. A 918 é resposta à mensagem 908, 1118 e 1098 são respostas à mensagem 1093 e assim sucessivamente.

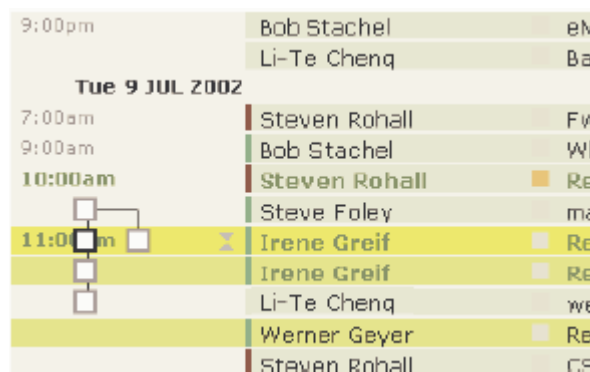


Assim como a *Narrow Tree*, a *TreeTable* também apresenta um trecho inicial da mensagem, facilitando a exploração de mensagens no thread.

Para uma leitura completa do *thread*, *TreeTables* são usadas como guias auxiliares e cada coluna representa uma linha do *thread* (ou *sub-thread*), com as respectivas respostas nas linhas subseqüentes.

Rohall e Gruen (2002) apresentam o *ReMail: A reinvented email prototype*, que incorpora novas maneiras de visualizar, gerenciar e interagir com *threads* e outros grupos de mensagens relacionadas.

A proposta de Rohall e Gruen (2002), conforme a figura 8, suporta a visualização de *threads* em árvores, em paralelo com a mensagem. Ao mesmo tempo, quando uma mensagem é selecionada, as demais mensagens do *thread* são mostradas em um tom diferente e o nó respectivo da mensagem é selecionado na árvore do *thread*, que fica ao lado esquerdo da janela de trabalho do usuário.



**Figura 8 - ReMail Prototype**

Nota: Extraído de Rohall e Gruen (2002)

O usuário também tem a possibilidade de selecionar um *thread*. Quando um *thread* é escolhido, todas as mensagens são reunidas em um único item na lista de visualização de mensagens. Esse item fica localizado na posição equivalente à mensagem individual mais recente do *thread*.

As formas de visualização apresentadas até o momento possuem como característica a ênfase no relacionamento pai-filho entre as mensagens, mas essas técnicas apresentam dificuldade em determinar as relações de tempo entre as mensagens trocadas.

Para permitir uma melhor visualização das relações de tempo entre as mensagens trocadas, Rohall (2002) apresenta uma técnica chamada *TimeLine*, onde as linhas verticais representam fronteiras temporais, no caso, um dia. O *thread* é visualizado como uma árvore disposta na horizontal, onde a dimensão tempo não é necessariamente linear, ou seja, dias com pouca ou nenhuma atividade podem ser mostrados de forma comprimida, permitindo que uma maior faixa de tempo esteja visível na tela do computador, vide Figura 9.

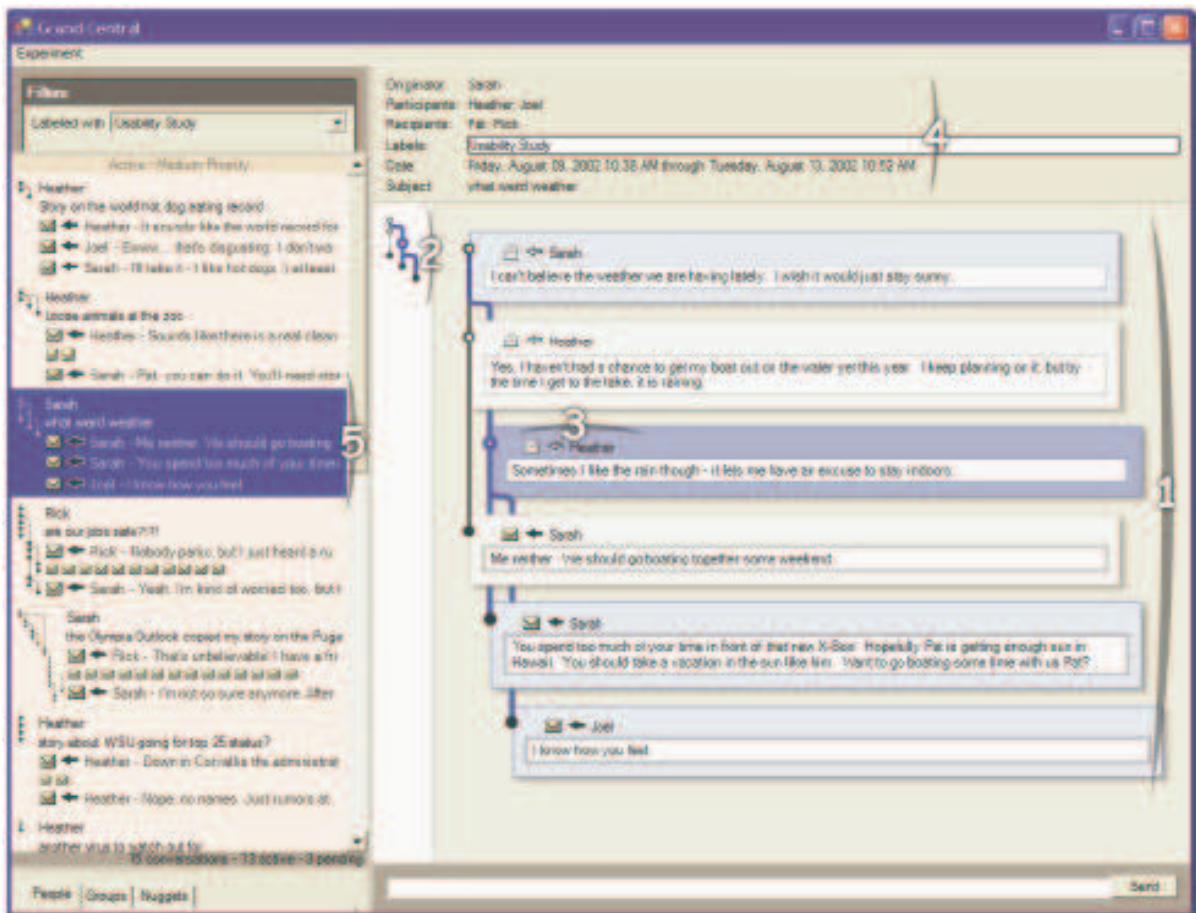


**Figura 9 - Visualização *TimeLine***

Nota: Extraído de Rohall (2002)

Esta abordagem, além de permitir a visualização e identificação clara de que mensagens são respostas a que mensagens, também permite analisar como essa troca ocorre distribuída no tempo, facilitando, por exemplo, a identificação das pessoas que conseguem responder a seus e-mails de forma mais ágil

Venólia e Neustaedter (2003) apresentam um modelo misto para visualização de conversação baseada em mensagens eletrônicas. Esse modelo, segundo os autores, foi desenhado para permitir que grandes *threads* de mensagens possam estar disponíveis e facilmente visíveis para o usuário, com pouca necessidade de rolamento vertical da tela visual. Essa técnica usa o princípio de visão geral mais detalhes, onde em uma única janela temos dois *frames*, um com os resumos e outro com os detalhes do item selecionado no *frame* resumo.



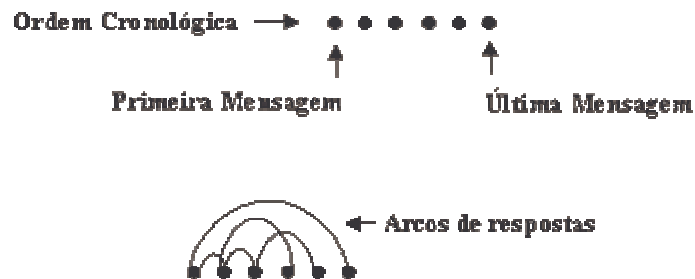
**Figura 10 - Modelo Misto de Visualização**

Nota: Extraído de Venólia e Neustaedter (2003)

Com pode ser visto na Figura 10, esse modelo apresenta do lado esquerdo uma lista de *threads* disponíveis para visualização (5). No lado direito da janela, com o *thread* já selecionado, temos uma pequena árvore esquemática do *thread* (2), um painel com uma visualização mais detalhada (1), indicando a seqüência de respostas a mensagens, um resumo do cabeçalho (*header*) da mensagem (3) e uma árvore com informações resumidas sobre o *thread* (4). Os três painéis descritos anteriormente (1, 2 e 4) provêm diferentes visões de um mesmo *thread*.

Kerr (2003) propôs uma técnica de visualização onde cada mensagem do *thread* é representada por um nó e os relacionamentos (respostas a mensagens) representados por arcos interligando os nós.

Nessa técnica, as mensagens são colocadas em seqüência, da esquerda para direita, de acordo com a sua ordem cronológica. Em seguida, os relacionamentos de respostas a mensagens são adicionados interligando os nós (mensagens) utilizando arcos. Essa técnica é chamada de *Thread Arcs*. A Figura 11 apresenta demonstra essa cronologia e a representação de arcos utilizada para as respostas a mensagens.

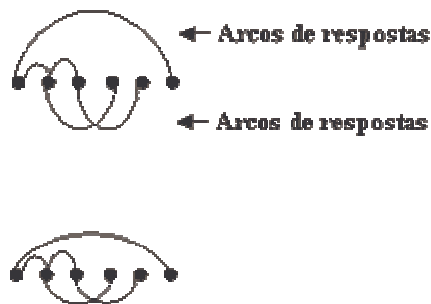


**Figura 11 - Representação da cronologia e relacionamento no *Thread Arcs***

Nota: Adaptado a partir de Kerr (2003)

Para permitir que a técnica seja eficiente não apenas para atender a pequenos, mas também a grandes *threads*, a representação dos relacionamentos através dos arcos pode ser feita não apenas pela parte superior, como também pela parte inferior do desenho, tornando-o mais claro.

Outro aspecto importante é a possibilidade de restringir a altura máxima dos arcos, para tornar a visualização mais compacta e permitir que esse recurso possa ser incluído mais facilmente nas ferramentas cliente de correio eletrônico. Nesta técnica, as mensagens são adicionadas ao *thread* à medida que chegam, sempre à direita da última mensagem. Assim, a representação de uma mensagem nunca precisa mudar de lugar, como pode ocorrer em outras técnicas vistas anteriormente. A Figura 12 demonstra essas características discutidas.



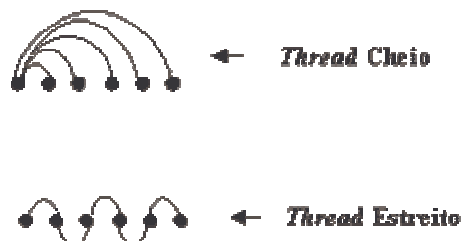
**Figura 12 - Representação de escala e compactação do *Thread Arcs***

Nota: Adaptado a partir de Kerr (2003)

O *Thread Arcs* também permite visualizar de uma forma muito clara, informações como o tamanho do *thread* (número de nós) e número de respostas a uma mensagem (número de arcos que chegam a um nó).

Segundo Kerr (2003), *threads* que possuem mensagens que recebem duas ou mais respostas são chamados de *threads* cheios, enquanto os *threads* com mensagens que possuem apenas uma resposta são chamados de *threads* estreitos.

Tanto *threads* cheios como estreitos podem ser facilmente identificados pela técnica *Thread Arcs*, apenas com a observação dos arcos gerados, conforme visto na Figura 13.



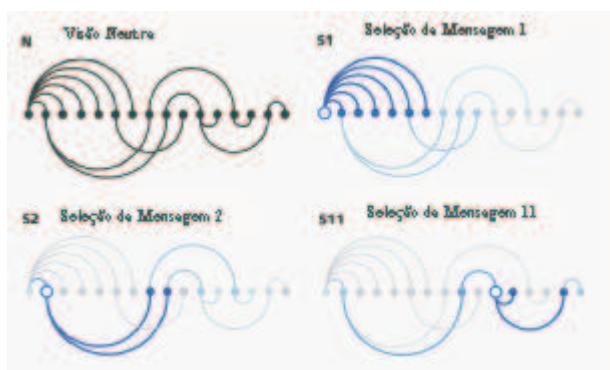
**Figura 13 - *Threads* cheios e estreitos**

Nota: Adaptado a partir de Kerr (2003)

Utilizando a técnica proposta do *Thread Arcs*, é possível a inclusão de funcionalidades que permitam o usuário interagir melhor com a ferramenta de visualização, através da marcação (*highlight*) de atributos. Dessa forma, o usuário pode selecionar uma determinada

mensagem e verificar toda a linha de mensagens anteriores e posteriores a ela ou, por exemplo, identificar todas as mensagens enviadas por um determinado usuário. Essa possibilidade traz uma evolução significativa nas técnicas de visualização de *threads*, com a inclusão da interatividade com o usuário.

A Figura 14 mostra a seleção de uma mensagem, com as mensagens anteriores e posteriores na linha de troca de mensagens dentro do *thread*.



**Figura 14 - Marcação de mensagens no *thread***

Nota: Adaptado a partir de Kerr (2003)

Segundo Kerr (2003), existem sete qualidades essenciais a uma ferramenta para permitir uma efetiva visualização de *threads* de mensagens: cronologia, relacionamento, estabilidade, compacta, marcação de atributos, escala e interpretação. Essas qualidades estão detalhadas no Quadro 6.

Qualidade da Ferramenta	Descrição
Cronologia	Deve ser possível visualizar a seqüência de mensagens e conseqüente evolução do <i>thread</i> , identificando as mensagens mais antigas e as mais recentes;
Relacionamento	As relações pai-filho de respostas a mensagens devem estar claras. Isso permite que o usuário facilmente identifique o relacionamento de

Qualidade da Ferramenta	Descrição
	uma mensagem com outras dentro do <i>thread</i> . É fundamental também que seja possível localizar rapidamente todas as mensagens anteriores a uma mensagem, assim como também as respostas subseqüentes;
Estabilidade	Como o <i>thread</i> está em constante evolução, é importante que uma determinada mensagem apareça sempre na mesma posição, facilitando a sua localização pelo usuário, mesmo que outras mensagens já tenham sido adicionadas a ele;
Compacta	Como normalmente a capacidade de visualização de <i>threads</i> está embutida em ferramentas cliente de correio eletrônico, é fundamental que ela seja compacta, em função da concorrência pelo pouco espaço disponível nas interfaces dos usuários;
Marcação de Atributos	O usuário deve ter a possibilidade de marcar atributos específicos do <i>thread</i> , como por exemplo, todas as mensagens enviadas por uma determinada pessoa, ou todas as mensagens enviadas em uma determinada data;
Escala	A técnica de visualização deve suportar tanto pequenos como grandes <i>threads</i> . Com pequenos ou grandes <i>threads</i> , a técnica deve prover mecanismos que permitam a sua visualização de forma clara;
Interpretação	Uma técnica de visualização de <i>thread</i> deve permitir ao usuário interpretar o tipo de conversação realizada, isto é, se trata de uma seqüência de respostas entre duas pessoas ou a busca por informações, com diversas respostas a uma única mensagem.

**Quadro 6 - Qualidades de uma ferramenta de visualização de *threads***

Com base nas qualidades propostas por Kerr (2003), o Quadro 7 apresenta uma análise resumo das técnicas de visualização de *threads* vistas neste capítulo. É importante ressaltar que a análise das características foi realizada a partir das informações apresentadas pelos autores das ferramentas, não tendo sido realizada uma avaliação prática destas.

Técnica de Visualização	Cronologia	Relacionamento Pai x Filho	Estabilidade	Compacto	Marcação de Atributos	Escala	Interpretação
<i>Narrow Tree</i> (NEWMAN, 2001)	Não	Sim	Não	Não	Não	Não	Sim
<i>Tree Table</i> (NEWMAN, 2001)	Não	Sim	Não	Não	Não	Não	Sim
<i>Re-mail Prototype</i> (ROHALL e GRUEN, 2002)	Não	Sim	Sim	Sim	Não	Sim	Não
<i>TimeLine</i> (ROHALL, 2002)	Sim	Sim	Sim	Sim	Não	Sim	Sim
Modelo Misto (VENÓLIA e NEUSTAEDTER, 2003)	Não	Sim	Sim	Sim	Não	Sim	Sim
<i>ThreadArcs</i> (KERR, 2003)	Sim	Sim	Sim	Sim	Sim	Sim	Sim

**Quadro 7 - Resumo comparativo das técnicas de visualização de *threads***

## 2.7 CONSIDERAÇÕES FINAIS

Este capítulo trouxe uma visão geral da estrutura do serviço de correio eletrônico, e foram apresentadas as principais características das técnicas disponíveis para a realização do mapeamento de *threads* e sua posterior visualização. No capítulo seguinte será apresentada uma proposta de solução para mapeamento de *threads* que visa minimizar os problemas e restrições encontradas nas alternativas vistas.



## CAPÍTULO 3 - SOLUÇÃO *THREADMAP*

Esse capítulo apresenta o *ThreadMap*, uma proposta de solução para o problema do mapeamento de *threads* de mensagens eletrônicas, focada na interceptação de mensagens no servidor de correio eletrônico como forma de minimizar os problemas anteriormente identificados no mapeamento de *threads* a partir da caixa postal de um usuário, evitando uma possível perda de trechos da conversação realizada.

### 3.1 FUNCIONAMENTO E ALGORITMO

Como já foi discutido anteriormente, uma solução de mapeamento de *threads* de mensagens eletrônicas baseado na interceptação de mensagens recebidas pelo servidor de correio eletrônico traz como principal benefício a possibilidade de mapear toda a conversação, independente da visão de cada usuário, e minimizando o risco de perda de trechos do *thread*.

Uma solução de interceptação de mensagens possui dois desafios principais: a identificação do *thread*, ou seja, a associação correta de cada mensagem recebida ao seu respectivo *thread* e qual a mensagem pai da atual, ou seja, a sua posição dentro da cadeia de mensagens trocadas nesse *thread*, uma vez que cada mensagem está normalmente associada a uma mensagem pai, seja na forma de resposta ou encaminhamento.

Para resolver os problemas mapeados acima, essa solução propõe a inclusão de uma marca na mensagem de forma a identificar o *thread* da mensagem atual e a sua posição dentro da cadeia de mensagens da conversação. Essa marca pode ser implementada a partir da

inclusão de cabeçalhos na mensagem de correio ou, caso essa funcionalidade não seja possível, através da sua inclusão no final do corpo da mensagem.

Com essa marca, é possível determinar a qual *thread* a mensagem pertence e a sua ausência indica o início de um novo *thread*. Além da identificação do *thread*, a marca deve registrar qual a mensagem anterior na cadeia de troca de mensagens, para a correta associação do relacionamento pai e filho.

O Quadro 8 apresenta o funcionamento da solução proposta.

<p><i>Para cada mensagem capturada</i></p> <p><i>Verifica-se se a mensagem capturada já possui a marca do thread.</i></p> <p><i>Se não possui a marca (significa o início de um novo thread)</i></p> <p><i>Registra o início do thread;</i></p> <p><i>Armazena as informações sobre o thread e a mensagem capturada;</i></p> <p><i>Insere a marca na mensagem capturada, com o número do novo thread e da mensagem atual, que será a mensagem pai das respostas e encaminhamentos;</i></p> <p><i>Caso contrário (nova mensagem de thread já iniciado)</i></p> <p><i>Obtém a informação do thread atual e da mensagem pai a partir da marca da mensagem;</i></p> <p><i>Armazena as informações sobre a mensagem capturada;</i></p> <p><i>Atualiza a marca da mensagem com o novo identificador da mensagem.</i></p> <p><i>Fim</i></p>
--

**Quadro 8 - Algoritmo da solução *ThreadMap***

De acordo com esse algoritmo, para cada mensagem capturada é verificada a que *thread* ela pertence e qual a mensagem pai, ou seja, a mensagem origem a partir da qual um determinado usuário enviou uma resposta ou um encaminhamento desta. As informações sobre o *thread* vão permitir o registro do usuário que o iniciou, a sua duração (data e hora de início e fim), os usuários envolvidos, a quantidade de mensagens trocadas e o intervalo entre elas, o assunto da conversação e quais foram as mensagens trocadas.

As informações sobre cada mensagem capturada devem incluir o usuário emissor, os usuários receptores (diretos e indiretos), o assunto, a data e hora do envio, a que *thread* ela

pertence, qual a sua mensagem pai (origem) e se houve ou não arquivos anexados. Para preservar a confidencialidade das informações contidas nas mensagens, essa solução não armazena o conteúdo da mensagem, sendo todas as informações acima relacionadas obtidas a partir do cabeçalho (*header*) da mensagem.

O Quadro 9 mostra as informações colhidas de cada mensagem e as respectivas fontes nos campos do cabeçalho (*header*).

<b>Informação</b>	<b>Fonte</b>
Identificador da mensagem	<i>Message-ID (header)</i>
Identificador do <i>thread</i>	Identificador do thread contido na marca criada pelo algoritmo.
Identificador da mensagem pai (origem)	Identificador da mensagem pai (origem) na marca criada pelo algoritmo
Assunto	<i>Subject (header)</i>
Data e hora da mensagem	<i>Date (header)</i>
Usuário emissor	<i>From (header)</i>
Usuários receptores	<i>To e Cc (header)</i>
Existência de arquivos anexados	<i>Attachment (header)</i>

**Quadro 9 - Fonte das informações colhidas sobre as mensagens**

Esta solução tem como requisito para a sua implementação que o servidor de correio eletrônico possua um mecanismo de suporte a eventos que permita a interceptação e captura das mensagens recebidas, e a possibilidade de manipulação das mensagens.

A captura de mensagens e mapeamento do *thread* em tempo real, na medida em que as mensagens forem trocadas, possuem como vantagem principal evitar que sejam perdidos trechos do *thread*, mas apresenta como desvantagem a impossibilidade de mapeamento do thread após a realização da conversa.

A Figura 15 ilustra o processo de captura da mensagem no servidor de correio eletrônico e o seu processamento pelo *ThreadMap* e posterior registro na base de dados.

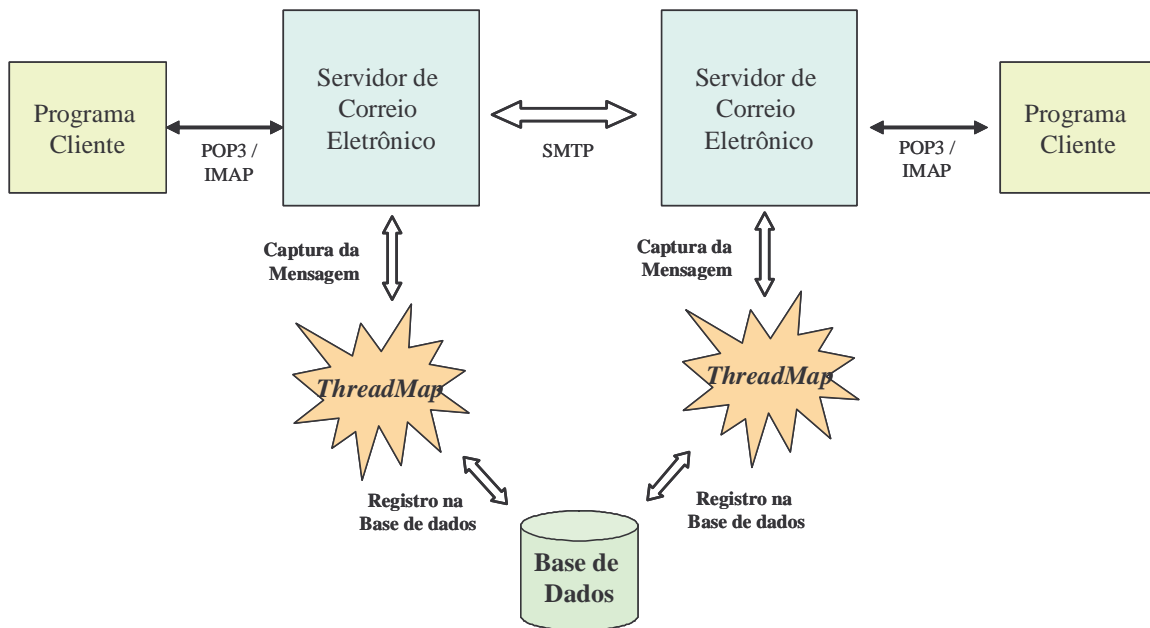
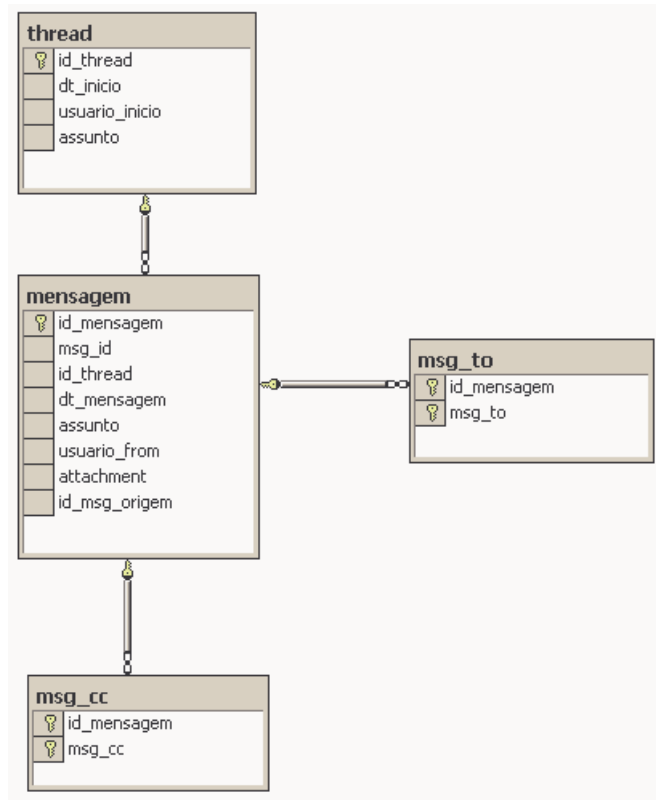


Figura 15 - Processo de captura da mensagem no servidor de correio eletrônico

### 3.2 MODELAGEM DE DADOS

As informações colhidas pelo *ThreadMap* sobre os *threads* e suas respectivas mensagens são registradas em uma base de dados, seguindo o modelo de dados apresentado na Figura 16.



**Figura 16 - Modelo de dados do *ThreadMap***

As informações do modelo de dados estão descritos no Quadro 10:

Tabela	Atributo	Descrição
<i>Thread</i>	Id_thread	Identificador do <i>thread</i> . Gerado automaticamente através de um campo auto-incremento.
	Dt_inicio	Data e hora de início do <i>thread</i> , Obtida através da data e hora da primeira mensagem do <i>thread</i> .

Tabela	Atributo	Descrição
	Usuario_inicio	Endereço de correio eletrônico do usuário que iniciou o <i>thread</i> . Obtido através do campo <i>from</i> da primeira mensagem do <i>thread</i> .
	Assunto	Assunto do <i>thread</i> . Obtido através do campo <i>subject</i> da primeira mensagem do <i>thread</i> .
Mensagem	Id_mensagem	Identificador da mensagem. Gerado automaticamente através de um campo auto-incremento.
	Msg_id	Campo <i>message-id</i> disponível no cabeçalho da mensagem.
	Id_thread	Identificador do <i>thread</i> a que a mensagem pertence.
	Dt_mensagem	Date e hora da mensagem. Obtido através do campo <i>date</i> do cabeçalho da mensagem.
	Assunto	Assunto da mensagem. Obtido através do campo <i>subject</i> do cabeçalho da mensagem.
	Usuario_from	Endereço de correio eletrônico do emissor da mensagem. Obtido através do campo <i>from</i> o cabeçalho da mensagem.
	Attachment	Indicador da existência de arquivos anexados na mensagem. Obtido através do campo <i>attachment</i> do cabeçalho da mensagem.

Tabela	Atributo	Descrição
	Id_msg_origem	Identificador da mensagem pai (origem). Essa informação é obtida na marca registrada pela aplicação de mapeamento.
Msg_to	Id_mensagem	Identificador da mensagem.
	Msg_to	Endereço de correio eletrônico do receptor da mensagem. Obtido através do campo <i>to</i> do cabeçalho da mensagem
Msg_cc	Id_mensagem	Identificador da mensagem.
	Msg_cc	Endereço de correio eletrônico do receptor da mensagem. Obtido através do campo <i>cc</i> do cabeçalho da mensagem

**Quadro 10 - Detalhamento do modelo de dados do *Threadmap***

### 3.3 BENEFÍCIOS DA SOLUÇÃO

Como descrito anteriormente, essa solução apresenta uma proposta de mapeamento de *threads* baseado na interceptação de mensagens no servidor de correio eletrônico, minimizando a possibilidade de perdas de trechos do *thread* em função de uma possível distribuição geográfica desses dados nas diversas caixas postais dos usuários envolvidos.

Com base na solução proposta, é possível a realização do mapeamento completo de todos os *threads*, cujas mensagens sejam trocadas no âmbito de um servidor de correio eletrônico. Esta solução pode ser ampliada para permitir o mapeamento além das fronteiras de um servidor, desde que instalada em todos os servidores de correio eletrônico dos usuários envolvidos e utilizando uma base de dados centralizada. Para tal, a solução proposta pelo *ThreadMap* pode ser implementada para qualquer servidor de correio eletrônico para permita a interceptação da mensagens e o acesso aos seus dados.

A partir das informações de cada *thread*, estão disponíveis dados que permitem mapear a estrutura em árvore de mensagens trocadas entre os envolvidos.

Além dos dados sobre cada mensagem, podem ser obtidas informações como a duração de cada *thread*, a quantidade de participantes (usuários) envolvidos, a quantidade de mensagens trocadas de cada *thread*, e realizar análises sobre a distribuição estatística desses dados dentro do universo amostrado.

Outro aspecto que pode ser explorado diz respeito à participação dos indivíduos. Sob essa ótica, é possível mapear a participação de cada usuário na conversação e o fluxo da comunicação dentro dos grupos de trabalho, sendo possível identificar os usuários que mais contribuem nessas comunicações, a intensidade da comunicação entre dois indivíduos e o tempo médio de resposta desses usuários no momento de sua contribuição.

O Quadro 11 traz uma análise comparativa com as técnicas de mapeamento vistas na seção 2.6.1, incluindo uma avaliação da solução *ThreadMap*.

<b>Técnica de Mapeamento</b>	<b>Identificação do <i>Thread</i></b>	<b>Fonte de Dados</b>	<b>Visão do <i>Thread</i></b>	<b>Montagem do <i>Thread</i></b>
Lewis (1997)	Baseada na similaridade do vocabulário das mensagens	Caixa postal dos usuários ou repositório de lista de discussão	Visão individual de cada participante	Posterior à conversação
Zawinski (1997)	Baseada em campos do cabeçalho das mensagens ( <i>In-Reply-To</i> , <i>References</i> e <i>Subject</i> )	Caixa postal dos usuários ou repositório de lista de discussão	Visão individual de cada participante	Posterior à conversação
Palme (1998)	Utiliza campos do cabeçalho das mensagens ( <i>Message-Id</i> , <i>In-Reply-To</i> e <i>References</i> )	Caixa postal dos usuários ou repositório de lista de discussão	Visão individual de cada participante	Posterior à conversação
Fisher e Moody (2002)	Utiliza o assunto e a data da mensagem.	Caixa postal dos usuários ou repositório de lista de discussão	Visão individual de cada participante	Posterior à conversação



<b>Técnica de Mapeamento</b>	<b>Identificação do <i>Thread</i></b>	<b>Fonte de Dados</b>	<b>Visão do <i>Thread</i></b>	<b>Montagem do <i>Thread</i></b>
IMAP Extension Work Group (2004) ORDERED SUBJECT	Utiliza o assunto e a data da mensagem.	Caixa postal dos usuários ou repositório de lista de discussão	Visão individual de cada participante	Posterior à conversação
IMAP Extension Work Group (2004) REFERENCES	Baseada em campos do cabeçalho das mensagens ( <i>In-Reply-To, References e Subject</i> )  * Técnica proposta por ZAWINSKI(1997)	Caixa postal dos usuários ou repositório de lista de discussão	Visão individual de cada participante	Posterior à conversação
THREADMAP	Baseada em uma marca incluída da mensagem.	Captura da mensagem, no servidor de correio eletrônico.	Visão geral do <i>thread</i> ,	Durante a conversação.

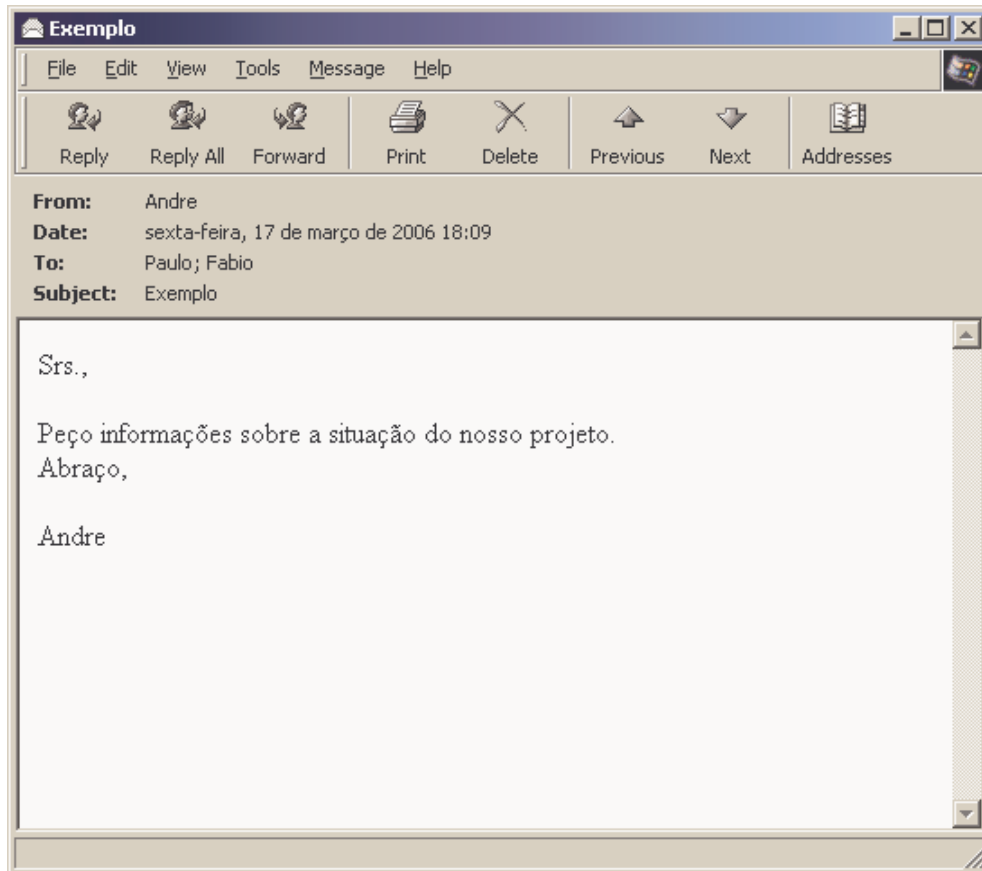
**Quadro 11 - Análise comparativa entre técnicas de mapeamento e *ThreadMap***

Como apresentado no Quadro 11, a solução *ThreadMap* utiliza a marca incluída das mensagens trocadas para o controle e identificação dos *threads*, utilizando como fonte de dados as mensagens interceptadas no servidor de correio eletrônico. Como consequência, o *ThreadMap* permite uma visão geral do *thread* e a sua montagem é realizada durante a realização da conversação.

### 3.4 SIMULAÇÃO DE CONVERSAÇÃO COM *THREADMAP*

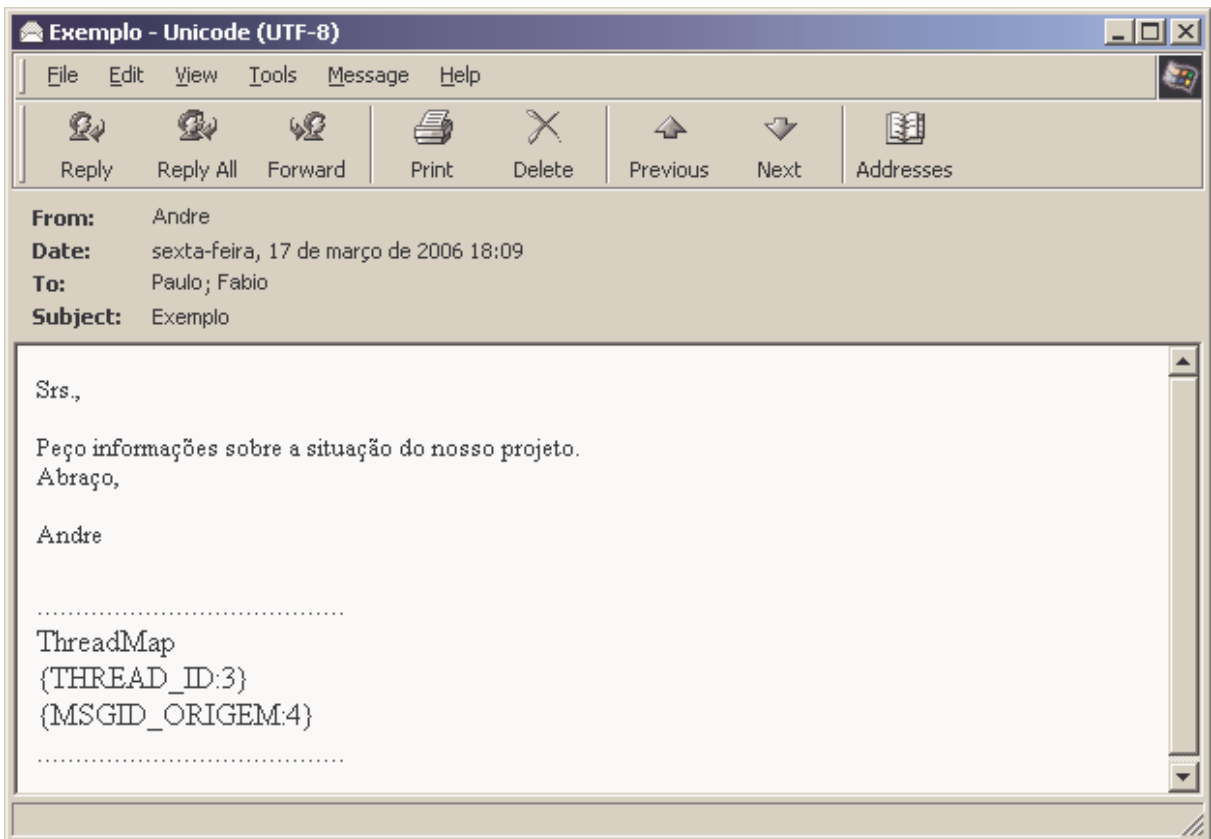
A seguir, é apresentada a realização de uma simulação de conversação via correio eletrônico, como objetivo de facilitar o entendimento do leitor de como o *ThreadMap* funciona e qual o impacto de sua utilização para os usuários.

Suponha que o usuário André tenha enviado uma mensagem para os usuários Paulo e Fabio, conforme a Figura 17.



**Figura 17 - Simulação de conversação com *ThreadMap* – Mensagem original**

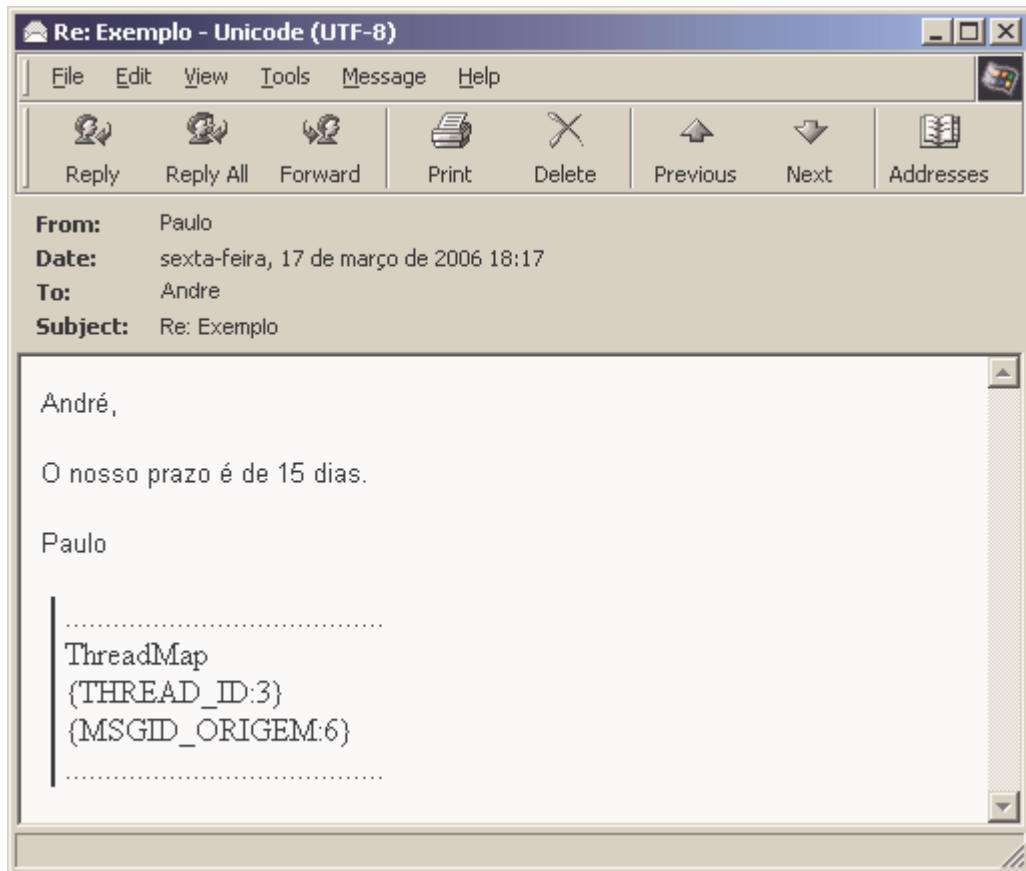
A aplicação *ThreadMap* irá interceptar a mensagem enviada pelo usuário André. Como nesta mensagem ainda não existe a marca que identifica a que *thread* ela pertence, a aplicação entende que se trata do início de um novo *thread*. Neste caso, será registrado na base de dados o início de um *thread* e que essa mensagem pertence a ele. A mensagem é então alterada e os usuários receptores, Paulo e Fabio, receberão a mensagem conforme a Figura 18.



**Figura 18 - Simulação de conversação com *ThreadMap* – Mensagem 1**

Assim, os usuários Paulo e Fabio receberão a mensagem de André com a inclusão da marca criada pelo *ThreadMap*, com as informações sobre o identificador do *thread* (`THREAD_ID:3`) e da mensagem origem, ou mensagem pai (`MSGID_ORIGEM:4`). Neste caso, o *ThreadMap* incluiu na sua base um novo *thread* com identificador 3 e uma nova mensagem com identificador 4, sem uma mensagem origem, por se tratar da primeira mensagem do *thread*.

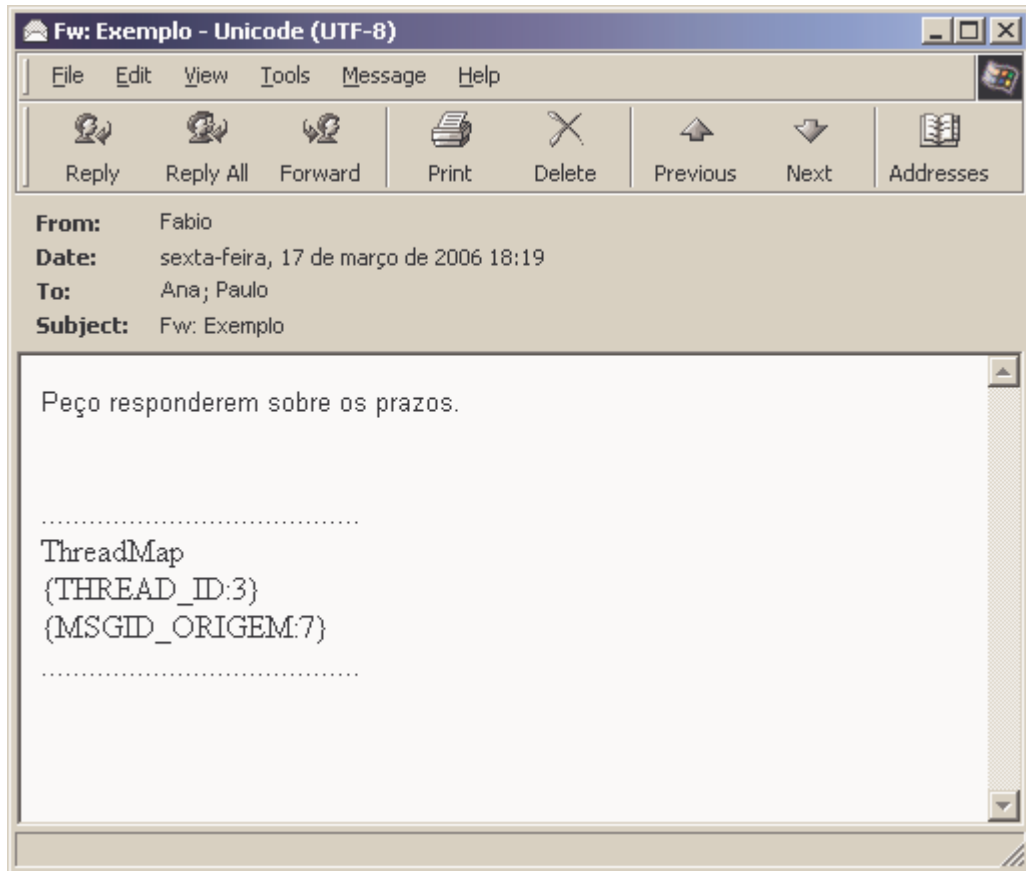
O usuário André recebe então a resposta enviada por Paulo, conforme a Figura 19. Agora, como a marca já existe na mensagem enviada, o *ThreadMap* captura a mensagem, registra na base de dados a mensagem, a mensagem de origem (pai) e a que *thread* ela pertence. Para dar continuidade ao processo, o identificador da mensagem origem é alterado para o identificador da mensagem atual (`MSGID_ORIGEM:6`).



**Figura 19 - Simulação de conversação com *ThreadMap* – Mensagem 2**

Neste caso, o *ThreadMap* incluiu uma nova mensagem com identificador 6, pertencente ao *thread* 3 e com mensagem origem (pai) 4.

Fabio por sua vez, encaminha a mensagem de André para os usuários Ana e Paulo. A Figura 20 apresenta a mensagem recebida por Ana e Paulo, com o mesmo identificador de *thread* (THREAD\_ID:3) e identificador de mensagem origem alterado (MSGID\_ORIGEM:7).



**Figura 20 - Simulação de conversação com *ThreadMap* – Mensagem 3**

Agora, o *ThreadMap* incluiu na sua base uma mensagem com identificador 7, pertencente ao *thread* 3 e com mensagem origem 4.

A Figura 21 apresenta a mensagem de resposta do usuário Paulo recebida por Fabio, com a marca da mensagem origem do *ThreadMap* alterada para 8 (MSGID\_ORIGEM:8). Neste caso, foi inserido na base de dados o registro de uma nova mensagem com identificador 8, pertencente ao *thread* 3 e com mensagem origem 7.

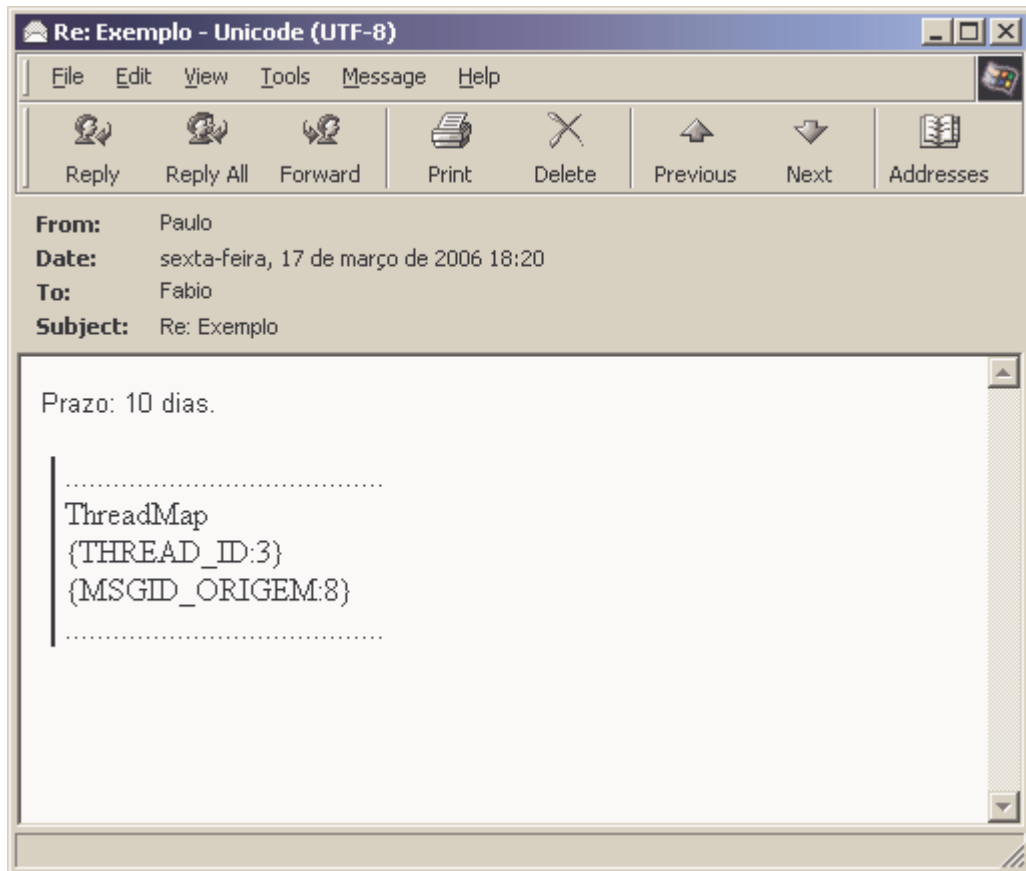


Figura 21 - Simulação de conversação com *ThreadMap* – Mensagem 4

A Figura 22 apresenta a mensagem de resposta da usuária Ana recebida por Fabio, com a marca da mensagem origem do *ThreadMap* alterada para 9 (MSGID\_ORIGEM:9). Neste caso, foi inserido na base de dados o registro de uma nova mensagem com identificador 9, pertencente ao *thread* 3 e com mensagem origem 7.

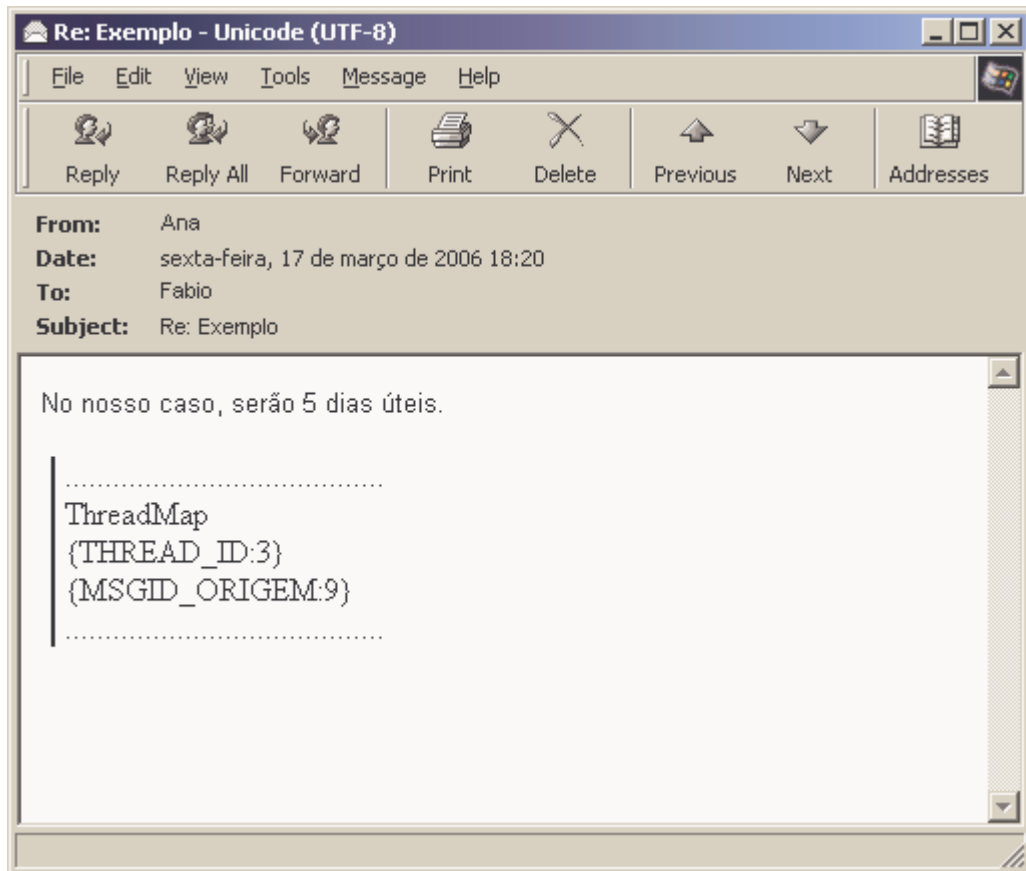
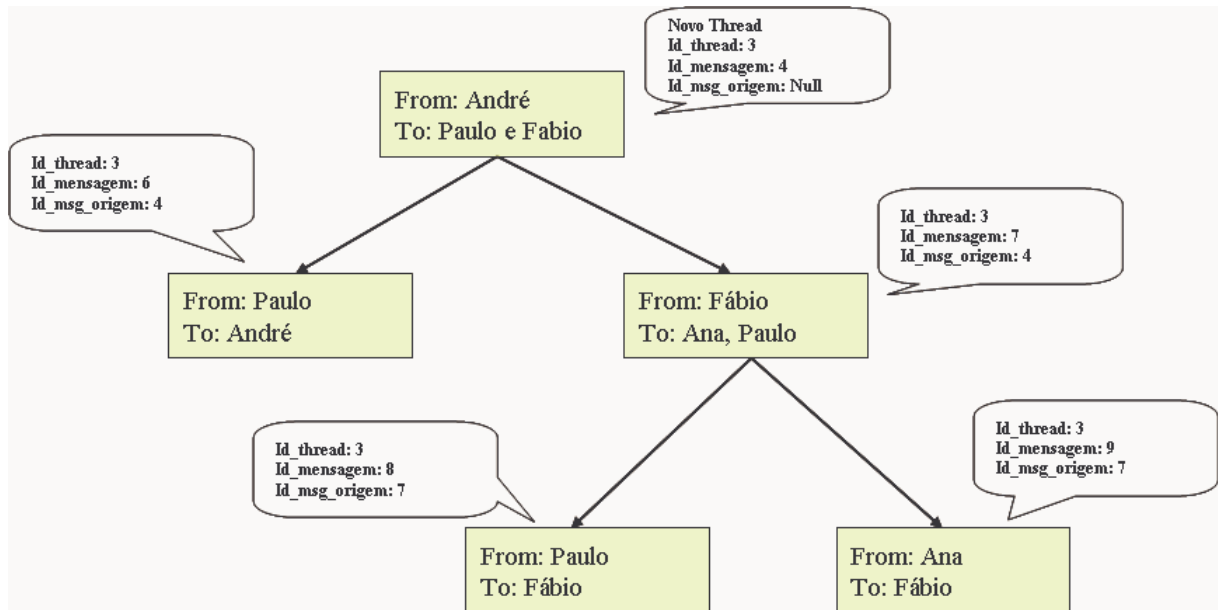


Figura 22 - Simulação de conversação com *ThreadMap* – Mensagem 5

Resumindo esta simulação, a figura 23 apresenta o *thread* no formato de árvore, com as informações sobre as mensagens trocadas a fim de facilitar o entendimento.



**Figura 23 - Simulação de conversação com *ThreadMap* – Árvore Resumo do *Thread***

Este *thread* simulado envolveu a participação de quatro usuários (André, Paulo, Fábio e Ana), com a troca de 5 mensagens.

### 3.5 ALTERNATIVAS DE IMPLEMENTAÇÃO

Como já citado anteriormente, a solução *ThreadMap* pode ser implementada em qualquer servidor de correio eletrônico que permita a captura e acesso às mensagens eletrônicas recebidas pelos usuários.

A Revista WSS apresenta um estudo realizado pelo Gartner que aponta o Microsoft Exchange e o IBM Lotus Notes como os servidores de correio eletrônico mais utilizados pelas organizações. Estes dados são relativos ao ano de 2004, e segundo ele, o MS Exchange possui 51% do mercado contra 40% do IBM Lotus Notes, e os dois juntos dominam 91% do mercado de servidores de correio eletrônico (MICROSOFT, 2005b).

Segundo Microsoft (2005a) e Nyamgondalu (2005), tanto a solução Microsoft quanto a IBM permitem a captura e manipulação das mensagens dos usuários.



Nyamgondalu (2005) apresenta uma biblioteca chamada C API Toolkit, que traz uma série de funcionalidades para integração com a solução Lotus Notes, em especial o *Extension Manager*, que permite a execução de rotinas customizadas e implementadas em C, antes ou após eventos no servidor, através do registro de rotinas de retorno (*callback routines*).

Microsoft (2005a) apresenta o recurso *Exchange Store Event* que permite a interceptação de eventos no servidor Exchange e a implementação de uma rotina que possa manipular as mensagens recebidas pelos usuários.

Desta forma, o *ThreadMap* pode ser implementado utilizando os recursos e funcionalidades disponíveis nos principais servidores de correio eletrônico do mercado.

### 3.6 CONSIDERAÇÕES FINAIS

Este capítulo apresentou a proposta *ThreadMap* de solução para mapeamento de *threads*, baseada na interceptação de mensagens no servidor de correio eletrônico, que tem como característica principal, a possibilidade de mapeamento durante a realização da conversação, minimizando a possibilidade de perdas de trechos do *thread*, e permitindo uma visualização que vai além da visão individual de cada participante da conversação.

No próximo capítulo será apresentada uma implementação da solução proposta neste capítulo, utilizando um servidor de correio eletrônico de mercado, detalhando a tecnologia adotada e os aspectos mais importantes desta implementação.

## CAPÍTULO 4 - IMPLEMENTAÇÃO *THREADMAP*

Este capítulo traz uma implementação para o *ThreadMap*, apresentando a tecnologia disponível para suporte a eventos do servidor de correio eletrônico adotado, assim como os detalhes relevantes de implementação, orientações necessárias para o seu funcionamento e uma ferramenta criada para permitir a visualização no formato de árvore desses *threads* mapeados – *ThreadViewer*.

### 4.1 AMBIENTE TECNOLÓGICO UTILIZADO

Como dito anteriormente, a solução *ThreadMap* possui como requisitos para implementação a utilização de um servidor de correio eletrônico com suporte a interceptação de mensagens e a possibilidade de manipulação das mensagens.

A implementação da solução proposta no Capítulo 3 foi realizada utilizando o seguinte ambiente tecnológico:

- Linguagem de programação: MS Visual Basic;
- Servidor de correio eletrônico: MS Exchange Server;

A escolha desse ambiente está embasada no fato do MS Exchange Server ser um servidor de correio eletrônico com suporte a eventos de interceptação, com ampla documentação disponível para pesquisa e pela possibilidade de realização de estudo de caso. O estudo de caso foi realizado nas instalações da Secretaria da Fazenda do Estado da Bahia, que possui esse produto como servidor de correio eletrônico corporativo.

A linguagem adotada está diretamente vinculada ao servidor de correio eletrônico escolhido para a implementação da solução.

## 4.2 *MS EXCHANGE SERVER*

O MS Exchange Server é o servidor de correio eletrônico e colaboração da Microsoft, projetado para permitir o acesso dos usuários às suas mensagens através de diversos programas clientes, com suporte aos protocolos SMTP, IMAP e POP3 descritos no Capítulo 2.

O MS Exchange Server armazena as mensagens trocadas pelos usuários em uma estrutura chamada *Information Store*. Dentro dessa base dados, estão contidas as caixas postais dos usuários, que podem estar organizadas em sub-pastas.

### 4.2.1 *Exchange Store Events*

Segundo Microsoft (2005a), o MS Exchange Server possui um mecanismo chamado *Exchange Store Event*, que permite a programação de regras de negócio associadas a eventos, como por exemplo: salvar, excluir, modificar, copiar ou mover itens na base de dados do MS Exchange Server (*information store*).

O *Exchange Store Event* suporta 3 (três) tipos de eventos: síncronos, assíncronos e de sistema. Os eventos síncronos e assíncronos são relativos a operações como o armazenamento ou remoção de itens na base de dados. Os eventos síncronos são disparados antes da conclusão (*commit*) da operação, enquanto que os eventos assíncronos são disparados após a conclusão das operações de armazenamento ou remoção dos dados. Os eventos de sistema são disparados em horários programados, ou nos eventos de partida e encerramento da base de dados do MS Exchange Server.

A partir dos eventos disponíveis, é possível a criação de código aplicativo, sob o formato de *Dynamic Link Library* (DLL) que são disparados na ocorrência desses eventos e são chamados de *event sinks*. Estes *event sinks* são então associados aos eventos disponíveis no MS Exchange Server e às caixas postais dos usuários. Dessa forma, é possível interceptar eventos e associa-los às caixas postais de alguns usuários.

O Quadro 12 apresenta um resumo dos eventos e respectivos métodos disponíveis para a implementação de *event sinks*.

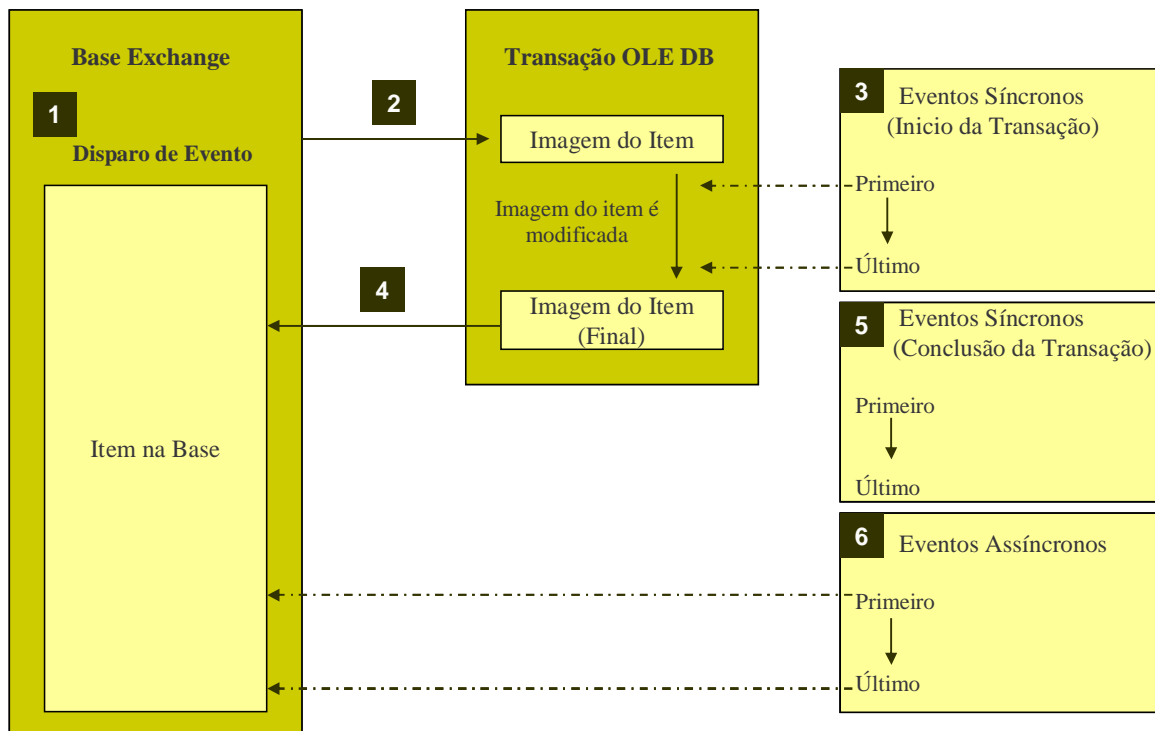
Tipo de evento	Método	Descrição
Assíncrono	<i>OnSave</i>	Executado assincronamente, quando um item é salvo na base de dados do Exchange.
	<i>OnDelete</i>	Executado assincronamente, quando um item é excluído da base de dados do Exchange.
Síncrono	<i>OnSyncSave</i>	Executado sincronamente, quando um item é salvo na base de dados do Exchange.
	<i>OnSyncDelete</i>	Executado sincronamente, quando um item é excluído da base de dados do Exchange.
Sistema	<i>OnTimer</i>	Executado após um específico intervalo de tempo.
	<i>OnMDBStartUp</i>	Executado quando a base do Exchange inicia o seu funcionamento.
	<i>OnMDBShutDown</i>	Executado quando a base do Exchange encerra o seu funcionamento ( <i>shutdown</i> )

**Quadro 12 - Eventos e métodos para implementação de *Event Sinks***

Para permitir a manipulação dos dados contidos na mensagem, assim como a inclusão de uma marca para registro do identificador do *thread* e da mensagem pai (origem), os

eventos síncronos são mais apropriados, visto que são os únicos que permitem a alteração do conteúdo das mensagens.

Os *event sinks* associados aos eventos síncronos são chamados duas vezes para cada ocorrência, no início e no final do processamento da mensagem salva (*OnSyncSave*) ou excluída (*OnSyncDelete*). Apenas durante a primeira fase do *event sink* as mensagens podem ter o seu conteúdo alterado. Assim, a inclusão da marca que permite a identificação do thread corrente e da mensagem origem precisa ser realizada nessa fase.



**Figura 24 - Diagrama de disparo dos eventos do MS Exchange Server**

Nota: Adaptado a partir de Microsoft (2005a)

A Figura 24 apresenta a seqüência de disparo de eventos disponíveis no MS Exchange Server:

- (1) O evento ocorre na base de dados do MS Exchange Server;

- (2) Uma imagem do item é criada para permitir a sua manipulação;
- (3) Métodos do evento síncrono recebem a notificação de início de operação. Nessa fase, o item pode ser modificado;
- (4) A imagem do item é salva na base de dados do MS Exchange Server;
- (5) Métodos do evento síncrono recebem a notificação da fase de conclusão da operação;
- (6) Métodos do evento assíncrono recebem a notificação de operação.

#### 4.2.2 Exchange OLEDB Provider (ExOLEDB)

Segundo Microsoft (2005a), os itens armazenados no MS Exchange Server podem ser acessados e manipulados através de uma fonte OLE DB, o ExOLEDB. O ExOLEDB é um componente de software que roda no servidor de correio eletrônico, que juntamente com o uso do modelo de objetos *Microsoft ActiveX Data Objects* (ADO), permite o acesso às mensagens e caixas de correio eletrônico disponíveis no servidor através da ligação (*bind*) desses itens a objetos ADO.

Com o uso de objetos ADO ligados às mensagens e caixas postais no servidor de correio eletrônico, a sua manipulação pode ser realizada através das referências URL: "*urn:schemas:mailheader*" e "*urn:schemas:httpmail*" disponíveis na propriedade *EventRecord* da interface *IExStoreDispEventInfo*. Através do acesso a essa interface, que é passada como parâmetro para a implementação nos métodos síncronos, os *event sinks* têm acesso às mensagens responsáveis pelo evento disparado.

Assim, é possível acessar os dados da mensagem para registro das informações na base de dados do *ThreadMap* e inclusão da marca de identificação do *thread* e mensagem origem.

#### 4.3 DETALHES DA IMPLEMENTAÇÃO

A solução *ThreadMap* aponta para a necessidade de criação de uma marca na mensagem como solução para os problemas de identificação do *thread* e da mensagem origem. O ideal seria que essa marca fosse criada no cabeçalho da mensagem, entretanto, a fonte de dados *ExOLEDB* permite a manipulação da mensagem, mas não permite a inclusão

de dados no cabeçalho da mensagem. Em função dessa restrição, foi adotada a alternativa de criação de um texto padrão que foi inserido no final de cada mensagem interceptada no servidor, com a identificação do *thread* corrente e da mensagem pai (origem).

O Quadro 13 apresenta o formato da marca inserida no corpo da mensagem:

```

.....
ThreadMap
{THREAD_ID:nnn}
{MSGID_ORIGEM:nnn}
.....

```

**Quadro 13 - Formato da marca inserida no corpo da mensagem**

Como discutido anteriormente, os eventos síncronos disparam a execução dos *event sinks* duas vezes, uma no início e outra no final do processamento da mensagem. No início do processamento, o disparo é realizado através da chamada à função *ProcessBeginSave* e no final a função *ProcessCommitSave* é chamada.

Como a solução *ThreadMap* precisa alterar o conteúdo da mensagem, a codificação necessária para captura das informações e alterações necessárias foi realizada na função *ProcessBeginSave*, executada a cada interceptação de nova mensagem armazenada na base dados do servidor de correio eletrônico. Essa função recebe como parâmetro um ponteiro para a interface do item (mensagem) que disparou o evento. Com esse ponteiro e com o uso dos componentes ADO, é possível acessar e proceder às alterações necessárias na mensagem que acabou de ser recebida pelo servidor.

O Quadro 14 apresenta os parâmetros passados para a função *ProcessBeginSave*.

```

Function ProcessBeginSave(ByVal pEventInfo As Exoledb.IExStoreEventInfo, ByVal bstrURLItem
                          As String, ByVal IFlags As Long) As Boolean

```

**Quadro 14 - Parâmetros passados para a função *ProcessBeginSave***

A partir do ponteiro para a interface *IExStoreDispEventInfo*, recebido como parâmetro na função *ProcessBeginSave* é feita a ligação com a mensagem que disparou o evento e que deverá ser tratada, conforme Quadro 15.

```
...  
Dim iDispEvtInfo As IExStoreDispEventInfo  
Dim Rec As ADODB.Record  
...  
Set iDispEvtInfo = pEventInfo  
Set Rec = iDispEvtInfo.EventRecord  
...
```

**Quadro 15 - Ligação da mensagem que disparou o evento**

Com a mensagem ligada ao objeto ADO, os dados da mensagem podem ser acessados através de fontes de dados (URL) disponíveis no *ExOLEDB provider*, conforme trecho de código apresentado no Quadro 16.

```
...  
Dim Flds As ADODB.Fields  
Dim msg_id, msg_from As String  
...  
Set Flds = Rec.Fields  
msg_id = Rec.Fields("urn:schemas:mailheader:message-id")  
msg_from = Rec.Fields("urn:schemas:mailheader:from")  
...
```

**Quadro 16 - Acesso aos dados da mensagem**

O Quadro 17 apresenta a relação de informações colhidas sobre as mensagens interceptadas e as suas respectivas fontes.



<b>Informação</b>	<b>Fonte (URL)</b>
Identificador da mensagem ( <i>message_id</i> )	<i>urn:schemas:mailheader:message-id</i>
Emissor da mensagem ( <i>from</i> )	<i>urn:schemas:mailheader:from</i>
Assunto da mensagem ( <i>subject</i> )	<i>urn:schemas:httpmail:subject</i>
Data da mensagem ( <i>date</i> )	<i>urn:schemas:mailheader:date</i>
Destinatários da mensagem ( <i>to e cc</i> )	<i>urn:schemas:mailheader:to</i> <i>urn:schemas:mailheader:cc</i>
Existência de anexos na mensagem ( <i>attachment</i> )	<i>urn:schemas:httpmail:hasattachment</i>
Corpo da mensagem	<i>urn:schemas:httpmail:textdescription</i> <i>urn:schemas:httpmail:htmldescription</i>

**Quadro 17 - Fonte de informações das mensagens interceptadas**

A determinação do início do *thread* é estabelecida a partir da localização da marca do *ThreadMap* no corpo da mensagem. Caso a marca seja encontrada, é possível identificar a qual *thread* essa mensagem pertence, caso contrário, um novo *thread* é iniciado e a marca incluída, como pode ser visto no trecho de código do Quadro 18.

```

...
msg_texto_ant = Rec.Fields("urn:schemas:httpmail:textdescription")
If InStr(1, msg_texto_ant, "{THREAD_ID:", vbTextCompare) <= 0 Then
    ' Marca não existe
    ' Início de um novo Thread
Else
    ' Marca ThreadMap encontrada.
    ' Localizar o identificador do thread e da mensagem origem.
End if
...

```

**Quadro 18 - Localização da marca no corpo da mensagem**

Como esse *event sink* estará associado a diversas caixas postais de usuários, é importante atentar para a possibilidade de vários usuários receberem uma mesma mensagem. Como no MS Exchange Server são armazenadas cópias dessa mensagem nas diversas caixas postais dos envolvidos e existirão diversas instâncias do *event sink* executando concorrentemente, a implementação *ThreadMap* precisa verificar se o registro do início desse *thread* já não foi realizado. Nesse caso, o *event sink* deve obter o identificador do *thread* na base de dados do *ThreadMap*, a partir do campo *message\_id* contido no cabeçalho da mensagem, evitando assim o registro do início de um novo *thread*.

As mensagens interceptadas possuem o seu conteúdo (corpo) codificado em texto e no formato HTML, disponíveis nas fontes *urn:schemas:httpmail:textdescription* e *urn:schemas:httpmail:htmldescription*, respectivamente. Em função disso, a inclusão da marca *ThreadMap* precisa ser realizada também nas duas codificações. No caso do conteúdo em texto, a marca é incluída no final do campo. No caso do conteúdo em HTML, é localizado a tag HTML `</BODY>` e a marca é incluída dentro dessa tag, conforme Quadro 19 abaixo.

```

...
msg_texto_ant = Rec.Fields("urn:schemas:httpmail:textdescription")
msg_html_ant = Rec.Fields("urn:schemas:httpmail:htmldescription")
pos = InStr(1, msg_html, "</BODY>", vbTextCompare)
If pos > 0 Then
    nova_msg_html = Left(msg_html_ant, pos - 1) & "<P></P><P>....." & _
        "<BR>ThreadMap" & "<BR>{THREAD_ID:" & id_thread & "}" & _
        "<BR>{MSGID_ORIGEM:" & id_msg & "}" & "<BR>....." & vbCrLf & _
        Right(msg_html_ant, Len(msg_html_ant) - (pos - 1))
End if
nova_msg = msg_texto_ant & vbCrLf & "....." & _
    vbCrLf & "ThreadMap" & vbCrLf & "{THREAD_ID:" & id_thread & "}" & _
    vbCrLf & "{MSGID_ORIGEM:" & id_msg & "}" & vbCrLf & "....." & vbCrLf
...

```

**Quadro 19 - Inclusão da marca no corpo da mensagem**

Para o registro das marcas indicativas do identificador do *thread* e da mensagem origem (pai) no corpo da mensagem interceptada é necessária a confirmação de alteração do seu conteúdo, vide Quadro 20.

```
...  
Rec.Fields("urn:schemas:httpmail:textdescription") = nova_msg  
Rec.Fields("urn:schemas:httpmail:htmldescription") = nova_msg_html  
Rec.Fields.Update  
...
```

**Quadro 20 - Alteração da mensagem trocada.**

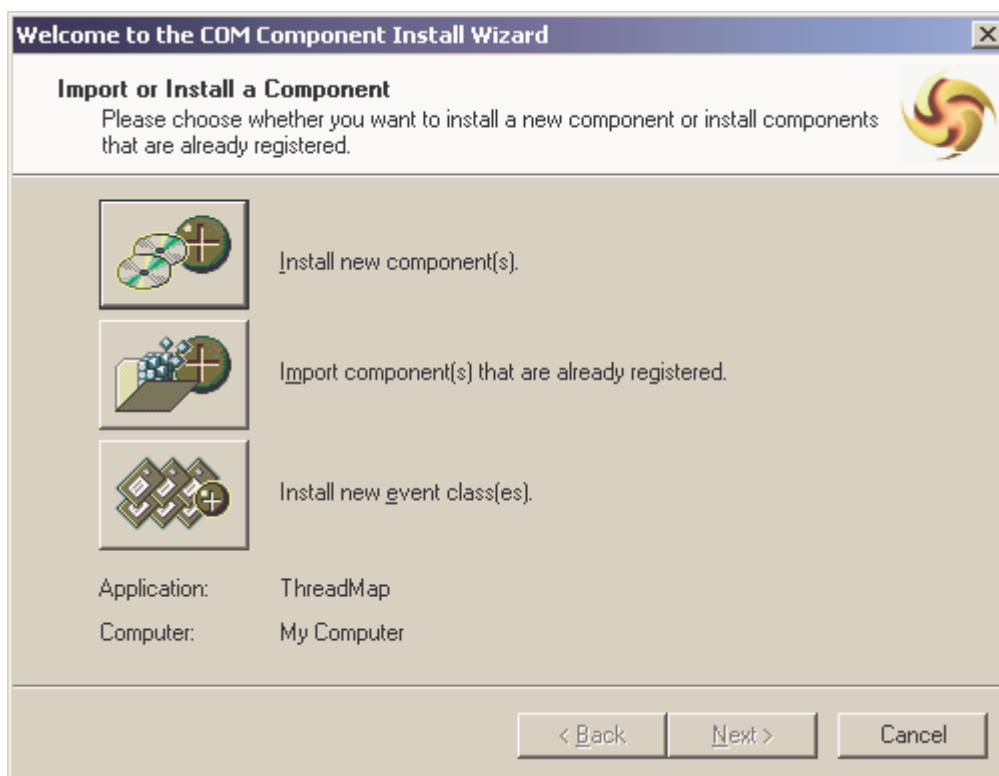
#### 4.4 ROTEIRO DE INSTALAÇÃO DO *EVENT SINK*

Como foi discutido anteriormente, o *event sink* é construído no formato DLL. Esta DLL precisa ser registrada no servidor de correio eletrônico MS Exchange Server, através do utilitário `regsvr32.exe`.

```
regsvr32 sinc.dll
```

Após o registro da DLL, é necessária a criação de uma aplicação COM+ (*Component Object Model*) e registro do *event sink* criado. A Figura 25 apresenta a ferramenta *Componente Service*, disponível no servidor de correio eletrônico, com uma aplicação *ThreadMap* criada na árvore COM+ *Applications*.





**Figura 26 - Processo de Importação de componente *event sink***

Após o registro no *Component Service*, essa DLL precisa ser associada ao evento síncrono *OnSyncSave*, nas caixas postais de todos os usuários envolvidos no processo de captura das mensagens trocadas. Esse registro pode ser realizado através do script *regevent.vbs*, disponível no Exchange SDK, vide Quadro 21.

```
cscript regevent.vbs add "onsyncsave;onsyncdelete" project2.class2
file://./backofficestorage/SERVIDOR/mbx/USUARIO/inbox/sink1
```

**Quadro 21 - Script para associação do *event sink* à caixa postal do usuário**

A referência à caixa postal do usuário deve seguir o formato *URL namespace*:  
 file://./backofficestorage/<domain.name>/MBX/<exchange\_user>/pasta.

#### 4.5 INTERFACE DE VISUALIZAÇÃO - *THREADVIEWER*

O objetivo da ferramenta *ThreadViewer* é apresentar uma interface que permita aos seus usuários visualizar os *thread* mapeados com o *ThreadMap*, possibilitando a localização de um *thread* específico e posterior visualização de seus detalhes.

Diferentemente das interfaces apresentadas no Capítulo 2 desse documento, o *ThreadViewer* não se propõe a ajudar o usuário de correio eletrônico a interagir com a conversação em curso. A intenção é facilitar a busca por *threads* mapeados, a partir de suas características, sendo criada apenas para permitir a visualização dos *threads* mapeados pelo *ThreadMap*, não se tratando de uma ferramenta com todas as funcionalidades necessárias para a realização de uma análise visual dos *threads*..

Para facilitar esse processo de busca, foi estabelecido o seguinte conjunto de características básicas do *thread*, tomando como referência os dados mapeados pelo *ThreadMap*: quantidade de usuários participantes, quantidade mensagens trocadas, duração, usuário que iniciou o *thread* e a sua data de início.

O Quadro 22 apresenta um resumo dessas características e descreve a fonte dessas informações. Vale lembrar que todos esses dados estão disponíveis na base do *ThreadMap*.

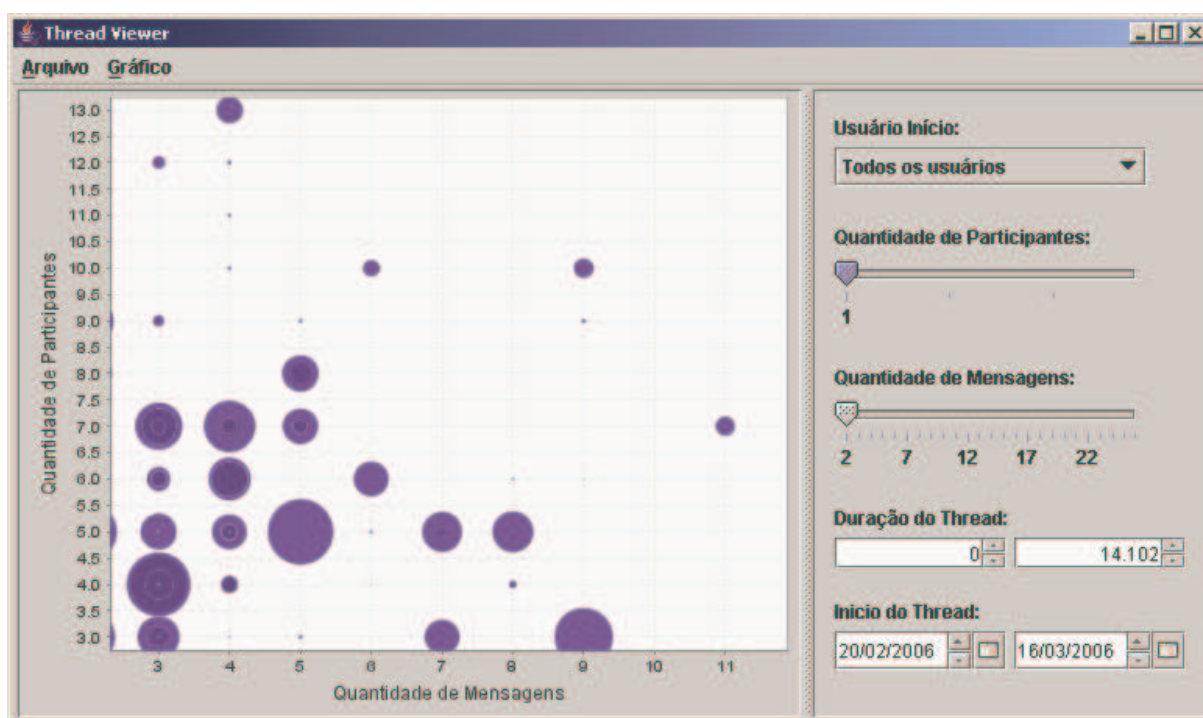
Característica do thread	Fonte da informação
Quantidade de participantes	Quantidade de usuários envolvidos no <i>thread</i> , obtida a partir das informações de emissor e receptores da mensagem (to e cc).
Quantidade de mensagens	Quantidade de mensagens trocadas e mapeadas pertencentes ao <i>thread</i> .
Duração do <i>Thread</i>	Duração do <i>thread</i> , obtido através da diferença de dados entre a primeira e última mensagem mapeada do <i>thread</i> .
Usuário início	Usuário que iniciou o <i>thread</i> , ou seja, o emissor da primeira mensagem do <i>thread</i> mapeado.

Característica do thread	Fonte da informação
Início do <i>Thread</i>	Data em que se iniciou o <i>thread</i> , ou seja, a data da primeira mensagem do <i>thread</i> mapeado.

**Quadro 22 - Características dos *threads* e as fontes de informações**

Para apresentar uma interface de localização mais amigável, os *threads* são apresentados em um gráfico e foram escolhidas 3 dimensões principais para a busca de um *thread*, são elas: a duração, a quantidade de participantes e quantidade de mensagens.

A Figura 27 apresenta a interface principal do *ThreadViewer*, com os *threads* desenhados em formato de círculos, em um gráfico tendo a quantidade de mensagens e de participantes como eixos X e Y, respectivamente. A duração representa a terceira dimensão do gráfico e está ilustrada pelo tamanho dos círculos no gráfico.



**Figura 27 - Interface principal do *ThreadViewer***

Ao lado do gráfico com os *threads* desenhados, é disponibilizado uma paleta com 5 controles que atuam dinamicamente na seleção dos *threads* à medida em que são

manipulados. Esses controles aplicam filtros ao conjunto de dados consultados, conforme descrito a seguir:

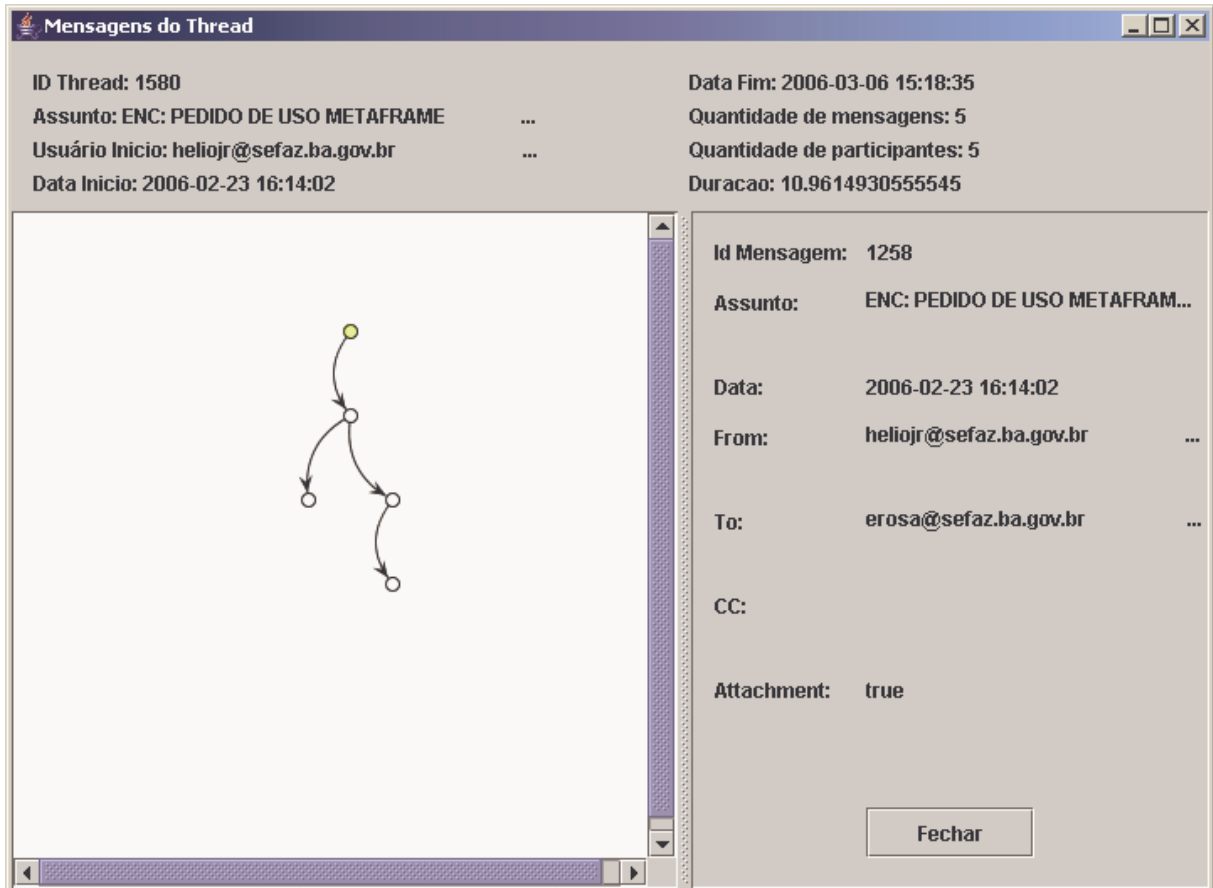
- Usuário início – Seleção de um usuário que tenha iniciado o *thread*.
- Quantidade de participantes – Controle que permite selecionar a quantidade de participantes do *thread*. Os *threads* selecionados devem ter quantidade de participantes superior ou igual ao valor selecionado no controle.
- Quantidade de mensagens – Controle que permite selecionar a quantidade de mensagens do *thread*. Os *threads* selecionados devem ter quantidade de mensagens superior ou igual ao valor selecionado no controle.
- Duração do *thread* – Controles estabelecem um período de tempo para selecionar os *threads* que possuem duração dentro desse intervalo.
- Início do *thread* – Controles estabelecem um período de tempo para selecionar os *threads* que iniciaram dentro desse intervalo.

Após a localização de um *thread* específico, o usuário do *ThreadViewer* pode visualizar os seus detalhes e a árvore de mensagens que compõem o *thread*, bastando para isso selecionar o *thread* desejado.

A Figura 28 apresenta a interface de detalhes do *thread*, dividida em três partes: na parte superior, os dados do *thread* selecionado como assunto, usuário início, duração, quantidade de participantes e data de início e fim; na parte inferior do lado esquerdo temos uma árvore montada a partir das mensagens trocadas e os respectivos relacionamentos de pai e filho e finalmente na parte inferior à direita, os detalhes obtidos de cada mensagem selecionada, como assunto, usuário emissor, usuários receptores (*TO* e *CC*), data da mensagem e se houve ou não arquivos em anexo. À medida que as mensagens montadas na



árvore são selecionadas, os seus dados são exibidos do lado direito da interface de visualização.



**Figura 28 - Visualização das mensagens do *thread* no *ThreadViewer***

O Quadro 23 apresenta um resumo comparativo entre as técnicas de visualização de *threads* vistas na seção 2.6.2, incluindo nesta análise as características apresentadas pelo *ThreadViewer*. Tomando como base as características apresentadas por Kerr (2003), o *ThreadViewer* permite ao seu usuário a visualização dos relacionamentos pai x filho entre as mensagens trocadas e é uma solução que permite a visualização de pequenos e grandes *threads*, de forma compacta e clara.

Técnica de Visualização	Cronologia	Relacionamento Pai x Filho	Estabilidade	Compacto	Marcação de Atributos	Escala	Interpretação
<i>Narrow Tree</i> (NEWMAN, 2001)	Não	Sim	Não	Não	Não	Não	Sim
<i>Tree Table</i> (NEWMAN, 2001)	Não	Sim	Não	Não	Não	Não	Sim
<i>Re-mail Prototype</i> (ROHALL e GRUEN, 2002)	Não	Sim	Sim	Sim	Não	Sim	Não
<i>TimeLine</i> (ROHALL, 2002)	Sim	Sim	Sim	Sim	Não	Sim	Sim
Modelo Misto (VENÓLIA e NEUSTAEDTER, 2003)	Não	Sim	Sim	Sim	Não	Sim	Sim
<i>ThreadArcs</i> (KERR, 2003)	Sim	Sim	Sim	Sim	Sim	Sim	Sim
<i>ThreadViewer</i>	Não	Sim	Não	Sim	Não	Sim	Sim

**Quadro 23 - Resumo comparativo das técnicas de visualização de *threads* e o *ThreadViewer***

#### 4.5.1 Arquitetura do *ThreadViewer*

A ferramenta *ThreadViewer* foi construída utilizando como ambiente tecnológico a linguagem de programação Java, acessando a base de dados do *ThreadMap* via *Java Database Connectivity (JDBC)*, exportada para MS Access.

A aplicação foi projetada em cinco classes, com os seguintes objetivos:

- *ThreadViewerApp* – Classe responsável pela construção da interface principal do *ThreadViewer*, gráfico e controles já sensibilizados com os dados disponíveis na base de dados do *ThreadMap*. Utiliza o *framework JFreeChart* para desenho do gráfico dos *threads*, que será apresentado posteriormente.
- *TV\_XYZDataSet* – Classe que implementa a interface *XYZDataSet*, disponível no *framework JFreeChart*, responsável por controlar e filtrar os *threads* que serão exibidos no gráfico, de acordo com os dados selecionados nos controles disponíveis para os usuários.
- *HashThread* – Classe que estende a classe *hashtable* disponível no Java, com o objetivo de manter em memória os *threads* mapeados e servir como repositório de dados para que a classe *TV\_XYZDataSet* possa filtrar os dados que serão exibidos no gráfico dos *threads*. Essa classe acessa a base de dados do *ThreadMap*, adicionando todos os *threads* mapeados para um *hashtable* de *threads*, em memória.
- *Thread* – Classe que possui os dados de cada *thread* disponível na base de dados e que é inserida no *hashtable* implementado pela classe *HashThread*.
- *TreeThread* – Classe responsável pela interface de visualização de mensagens do *thread*. Utiliza o *framework JUNG* para montagem da árvore de *threads*, a partir do mapeamento realizado pelo *ThreadMap* e busca os detalhes da mensagem diretamente da base de dados, via JDBC.

Conforme citado acima, para a implementação do *ThreadViewer* foram utilizados os *frameworks JFreeChart* e *JUNG (Java Universal Network/Graph)* para auxiliar no processo de criação dos gráficos necessários à visualização dos *threads* e respectivas mensagens.

O *JFreeChart* é um *framework* desenvolvido em Java, de código aberto e disponível no site <http://www.jfree.org/jfreechart/index.php>. Trata-se de um conjunto de classes criadas

com o objetivo de facilitar a implementação de aplicações que manipulem gráficos, dos mais variados formatos: pizza, barra, linha, área, séries de tempo, entre outras (GILBERT, 2005).

Na implementação do *ThreadViewer*, foi utilizado o gráfico de bolhas (*BubbleChart*) para permitir uma melhor visualização das três dimensões principais do *thread* (duração, quantidade de participantes e quantidade de mensagens) utilizando apenas os eixos X e Y.

O gráfico de bolhas é criado utilizando a classe *ChartFactory* e o método *createBubbleChart*, passando como parâmetro, dentre outros, o conjunto de dados gerado pela classe *TV\_XYZDataSet* apresentada anteriormente.

A cada interação do usuário com a paleta de controles para seleção dos *threads*, a classe *TV\_XYZDataSet* atualiza o conjunto de dados, obtidos do *hashtable* de *threads* (*HashThreads*), e dispara um evento (*fireDatasetChanged()*) para que o gráfico seja atualizado com os novos dados.

O *Java Universal Network/Graph Framework* (JUNG) é uma biblioteca também desenvolvida em Java que auxilia na análise e visualização de dados que possam ser representados como grafos ou redes. Trata-se de uma biblioteca de código aberto e está disponível no site: <http://jung.sourceforge.net> (MADADHAIN, 2003).

Na implementação do *ThreadViewer*, o JUNG foi utilizado para desenhar a árvore de *threads* na interface de visualização das mensagens, após a seleção de um *thread*. Nesse caso, a manipulação com as classes JUNG está encapsulada na classe *TreeThread*.

A classe *TreeThread* utiliza a classe *SparseTree*, disponível no JUNG, que representa um conjunto de vértices (ou nós) e arcos direcionados. No caso do *ThreadViewer*, os vértices representam as mensagens trocadas e o arco indica a relação de pai e filho entre as mensagens de um *thread*.

Os vértices e arcos são adicionados ao gráfico (*SparseTree*) de acordo com os dados obtidos na base do *ThreadMap*, especificamente, os dados de identificador da mensagem e identificador da mensagem origem (pai).

#### 4.6 CONSIDERAÇÕES FINAIS

Neste capítulo foi discutida uma implementação para a solução *ThreadMap* de mapeamento de mensagens, utilizando o servidor MS Exchange Server, juntamente com o *ThreadViewer*, uma interface para visualização dos *threads* mapeados.

No próximo capítulo será apresentado um estudo de caso, realizado na Secretaria da Fazenda do Estado da Bahia, utilizando o *ThreadMap*.

## CAPÍTULO 5 - ESTUDO DE CASO

Este capítulo apresenta o estudo de caso realizado na Secretaria da Fazenda do Estado da Bahia, com a utilização do *ThreadMap* para interceptação e mapeamento dos *threads* de mensagens eletrônicas envolvendo a equipe da Gerência de Tecnologia (GETEC) da Diretoria de Tecnologia da Informação (DTI).

### 5.1 METODOLOGIA UTILIZADA

Para a realização deste estudo de caso, foi seguida a metodologia de trabalho apresentada abaixo.

- Solicitação de autorização para estudo de caso – Para a realização deste estudo de caso e conseqüente acesso às mensagens de correio eletrônico trocadas entre os profissionais desta organização, uma solicitação de autorização foi feita à Superintendência de Gestão Fazendária (SGF), órgão responsável pela Diretoria de Tecnologia da Informação. A sua realização foi autorizada pelo superintendente da SGF;
- Definição de escopo – Para a realização deste estudo foi necessário estabelecer um escopo de pessoas envolvidas. Certamente, quanto mais amplo o estudo, maior a possibilidade de captura de todo o *thread*. Entretanto, maior seria a dificuldade de autorização de acesso a estas informações. Visando mapear a

troca de informações dentro de uma equipe de trabalho, foi definido o escopo de uma gerência intermediária, dentro da Diretoria de Tecnologia da Informação.

- Solicitação de autorização aos participantes – Uma vez estabelecido o escopo da realização do estudo de caso e identificados os potenciais participantes, foi realizada uma consulta a todos eles, apresentando o projeto de pesquisa e solicitando autorização para a realização do mapeamento das mensagens. Neste caso, todos os membros da equipe escolhida aceitaram participar do estudo de caso.
- Acesso e configuração do ambiente – Como se trata de uma aplicação que é executada no servidor de correio eletrônico e acessa uma base de dados em um servidor de banco dados, foi necessário apoio da equipe técnica para a instalação do *ThreadMap* nos servidores da Sefaz.
- Comunicação do início da coleta – Após a configuração do ambiente da SEFAZ para a realização do estudo de caso, todos os participantes foram informados do início da coleta, através de uma mensagem de correio eletrônico. Este comunicado esclarecia os detalhes de sua utilização, quais as informações seriam coletadas e solicitando retorno em caso de qualquer impacto ao adequado funcionamento da sua ferramenta de correio eletrônico.
- Acompanhamento da execução – Durante o período de coleta dos dados, foi realizado um acompanhamento, por amostras, do recebimento e captura das mensagens e eventual impacto do funcionamento do correio eletrônico da Sefaz.

- Comunicação do término da coleta – Todos os participantes foram informados, através de uma mensagem de correio eletrônico, do término do processo de coleta e da posterior divulgação dos resultados através do relatório de análise.
- Coleta dos dados – Após o período estabelecido para coleta, os dados armazenados no servidor de banco de dados foram importados em uma base de dados MS Access para permitir o seu acesso através da ferramenta *ThreadViewer*.
- Análise dos resultados – Com base dos dados coletados, foi realizada análise da quantidade de mensagens e *threads* mapeados, tamanho médio dos *threads*, distribuição da quantidade de participantes e duração.
- Divulgação dos resultados – A divulgação dos resultados obtidos foi realizada através de elaboração de um relatório de análise dos dados coletados.

## 5.2 PROTOCOLO DE PREPARAÇÃO

Neste item do capítulo, serão descritas a organização alvo do estudo de caso, a população envolvida e período necessário de coleta dos dados para posterior análise.

### 5.2.1 Organização

A Secretaria da Fazenda do Estado da Bahia (Sefaz) foi criada em 16 de agosto de 1895, transformada pelo Decreto-Lei nº 11.889, de 30 de abril de 1941, e reorganizada pelas Leis nº 6.074, de 22 de maio de 1991, 7.249, de 07 de janeiro de 1998, e 7.435, de 30 de dezembro de 1998, tem por finalidade formular, coordenar e executar as funções de administração tributária, financeira e contábil do Estado, assim como planejar, coordenar, executar e controlar as atividades do Fundo de Custeio da Previdência Social dos Servidores Públicos do Estado da Bahia (FUNPREV) (BAHIA, 2001).



A Sefaz possui a sua estrutura operacional subdividida em seis áreas: Auditoria Geral do Estado (AGE), Conselho Fazendário, Diretoria Geral, Superintendência de Desenvolvimento da Gestão Fazendária (SGF), Superintendência de Administração Financeira (SAF) e Superintendência de Administração Tributária (SAT).

A SGF, onde o estudo de caso do *ThreadMap* foi realizado, tem por finalidade o desenvolvimento organizacional da Sefaz e a sua modernização tecnológica. A gestão da tecnologia da informação é realizada pela Diretoria de Tecnologia da Informação (DTI), responsável pelo desenvolvimento, manutenção de sistemas, administração do parque tecnológico e ambientes computacionais, redes de comunicação corporativa e segurança da informação.

### **5.2.2 População envolvida e escopo**

Dentro da DTI, foi escolhida a Gerência de Tecnologia (GETEC) para a realização do estudo de caso, pela facilidade de acesso a esta gerência, pela responsabilidade desta gerência pelo ambiente computacional da Sefaz e por representar uma equipe média dentro da organização, sendo composta por 19 pessoas.

A GETEC é responsável pela administração do ambiente computacional corporativo da Sefaz, incluindo suporte aos equipamentos servidores e principais serviços corporativos disponíveis como intranet, internet, correio eletrônico, banco de dados e *data warehouse*. Esta gerência também é responsável pela prospecção de novas tecnologias e pela sua incorporação no ambiente da Sefaz.

A GETEC é composta por 19 pessoas, sendo organizada em duas coordenações: infraestrutura e suporte a desenvolvimento e banco de dados. A equipe de infra-estrutura é responsável pelo suporte aos ambientes de correio eletrônico, sistemas operacionais e softwares básicos. A equipe de suporte a desenvolvimento e banco de dados é responsável por apoiar a gerência de desenvolvimento de sistemas, definir padrões de desenvolvimento e administrar os ambientes de banco de dados corporativos.

A duração da coleta dos dados foi estabelecida em aproximadamente 20 dias, após conversa com a equipe da GETEC, para determinação de um período necessário para mapeamento de *threads* envolvendo as atividades operacionais desta gerência.

### 5.3 APLICAÇÃO DO ESTUDO DE CASO

O processo de coleta e mapeamento dos *threads* foi realizado durante o período de 20 de fevereiro de 2006 a 16 de março de 2006, envolvendo os 19 profissionais da GETEC.

Durante esse período foram interceptadas e registradas em banco de dados 7.902 mensagens recebidas e trocadas entre os envolvidos no estudo de caso. Essas mensagens foram agrupadas pelo *ThreadMap* em 7.288 *threads*.

Ainda no período de coleta das informações foi identificado que algumas mensagens foram interceptadas e registradas na base de dados, mas a marca não estava presente após a etapa de processamento da mensagem. Após uma investigação foi verificado tratar-se de uma falha (*bug*) no serviço de eventos do MS Exchange Server, conforme artigo da Microsoft publicado, sob o número Q273233, disponível em <http://support.microsoft.com/?kbid=273233>. As alternativas de soluções apresentadas neste artigo implicam em mudanças no ambiente operacional, e dessa forma não foi possível a sua realização.

Como consequência desse problema, algumas mensagens que deveriam estar associadas a *threads* já registrados figuraram como um novo *thread*, uma vez que a solução dada pelo *ThreadMap* para identificar o início do *thread* é justamente a interceptação de uma mensagem sem a sua marca. Como estas mensagens não tiveram o seu conteúdo alterado e a marca incluída, as respostas e encaminhamentos foram identificados como o início de um novo *thread*.

### 5.4 ANÁLISE DOS RESULTADOS

Nesta seção, serão descritas as regras utilizadas para a análise dos dados coletados e os resultados obtidos com a realização do estudo de caso na Gerência de Tecnologia da Secretaria da Fazenda do Estado da Bahia.

### 5.4.1 Protocolo de análise

A análise dos dados coletados foi realizada utilizando como premissa a não identificação individual dos participantes, assim como também o não tratamento dos assuntos das mensagens trocadas durante o período da coleta dos dados.

Dessa forma, os dados apresentados buscam caracterizar o uso do correio eletrônico neste grupo de trabalho e identificação do volume de mensagens trocadas, *threads* mapeados e suas características, como quantidade de indivíduos participantes, quantidade de mensagens trocadas e duração.

Foram desconsiderados como *threads* a serem analisados, aqueles que tiveram apenas uma mensagem trocada, o que representa apenas o ato de informar ou divulgar algo, não caracterizando o uso do correio eletrônico como mecanismo de discussão.

Os *threads* analisados foram agrupados de acordo com a distribuição apresentada no Quadro 24, em cada uma das dimensões verificadas.

Dimensão	Agrupamento
Quantidade de mensagens	2 mensagens
	3 a 5 mensagens
	6 a 8 mensagens
	9 a 12 mensagens
	Acima de 12 mensagens
Quantidade de participantes	1 a 4 participantes
	5 a 8 participantes
	9 a 12 participantes
	13 a 16 participantes
	Acima de 16 participantes
Duração	Menor ou igual a 1 dia
	Entre 1 e 3 dias
	Entre 3 e 5 dias

Dimensão	Agrupamento
	Entre 5 e 7 dias
	Superior a 7 dias

**Quadro 24 - Dimensões e agrupamento utilizados para análise dos dados**

### 5.4.2 Resultados obtidos

Durante o período de realização do estudo de caso da GETEC, foram interceptadas 7.902 mensagens, agrupadas em 7.288 *threads*. Como para efeito de análise dos resultados obtidos foram considerados apenas os *threads* com mais de uma mensagem, esse universo se resumiu a 261 *threads*, com um total de 889 mensagens.

A seguir, iremos detalhar os números obtidos neste estudo de caso, consolidando a análise a partir das características quantidade de mensagens, quantidade de participantes e duração dos *threads*.

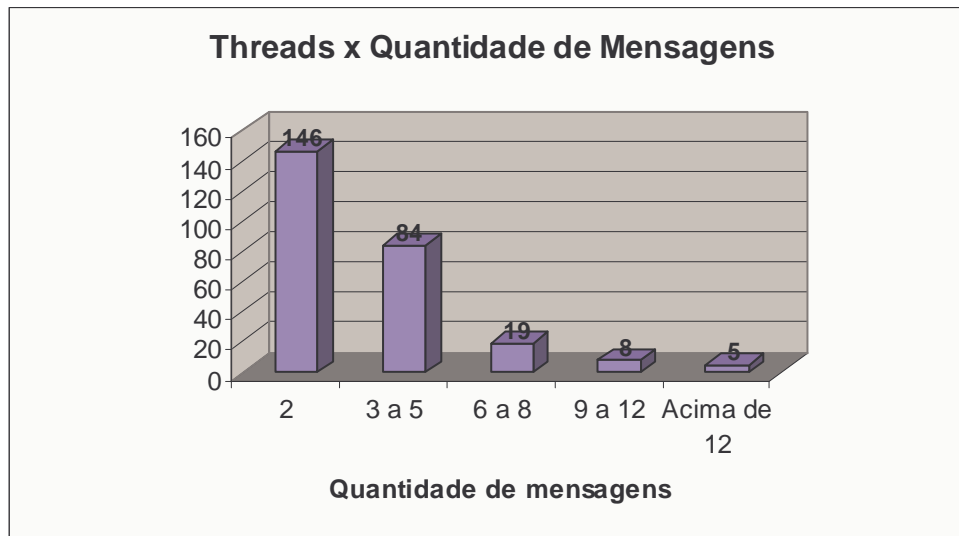
- **Quantidade de Mensagens**

Dentre os 261 *threads* analisados, houve o registro de 889 mensagens trocadas, o perfaz uma média de 3,4 mensagens por *thread*.

A Tabela 1 e a Figura 29 apresentam a distribuição de *threads* de acordo com o agrupamento de quantidade de mensagens, conforme proposto no Quadro 24.

**Tabela 1 - Distribuição de *threads* x quantidade de mensagens**

Quantidade de mensagens	Quantidade de <i>Threads</i>	Percentual (%)
02	146	56%
03 a 05	84	32%
06 a 08	19	7%
9 a 12	8	3%
Acima de 12	5	2%



**Figura 29 - Distribuição de *threads* x quantidade de mensagens**

- **Quantidade de Participantes**

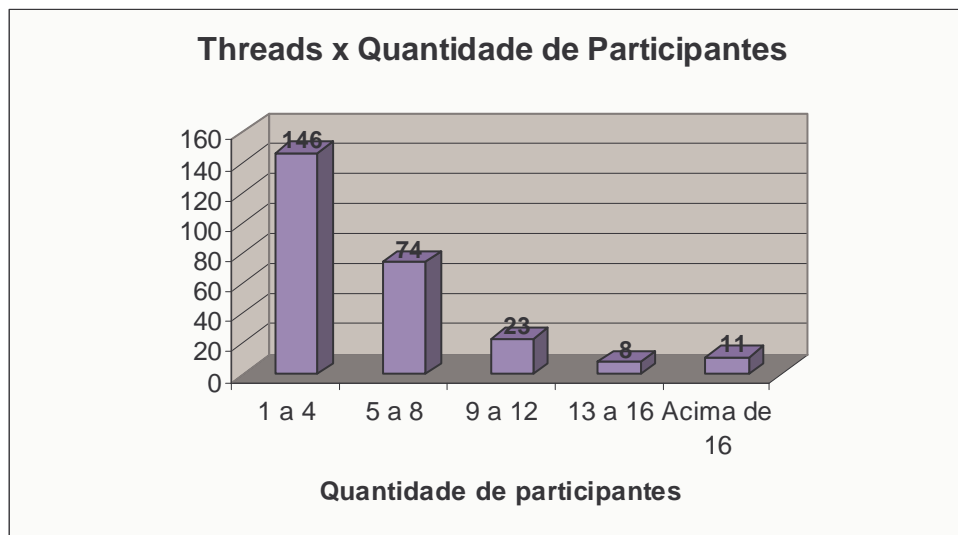
Para a análise da quantidade de participantes de cada *thread*, foram considerados os envolvidos como emissor e receptor de mensagens, através dos campos *From*, *To* e *CC*, contando a sua ocorrência uma única vez, em todo o *thread*.

O número total de participantes nos *threads* coletados foi de 305 pessoas, com uma média de 5,5 por *thread*. Vale lembrar que apesar do estudo de caso englobar a interceptação de mensagens de um grupo de 19 pessoas, esses participantes aqui referenciados incluem outras pessoas que enviaram e/ou receberam estas mensagens.

A Tabela 2 e a Figura 30 apresentam a distribuição de *threads* de acordo com o agrupamento de quantidade de participantes, conforme proposto no Quadro 24.

Tabela 2 - Distribuição de *threads* x quantidade de participantes

Quantidade de participantes	Quantidade de <i>Threads</i>	Percentual (%)
01 a 04	146	56%
05 a 08	74	28%
09 a 12	23	9%
13 a 16	8	3%
Acima de 16	11	4%

Figura 30 - Distribuição de *threads* x quantidade de participantes

- **Duração**

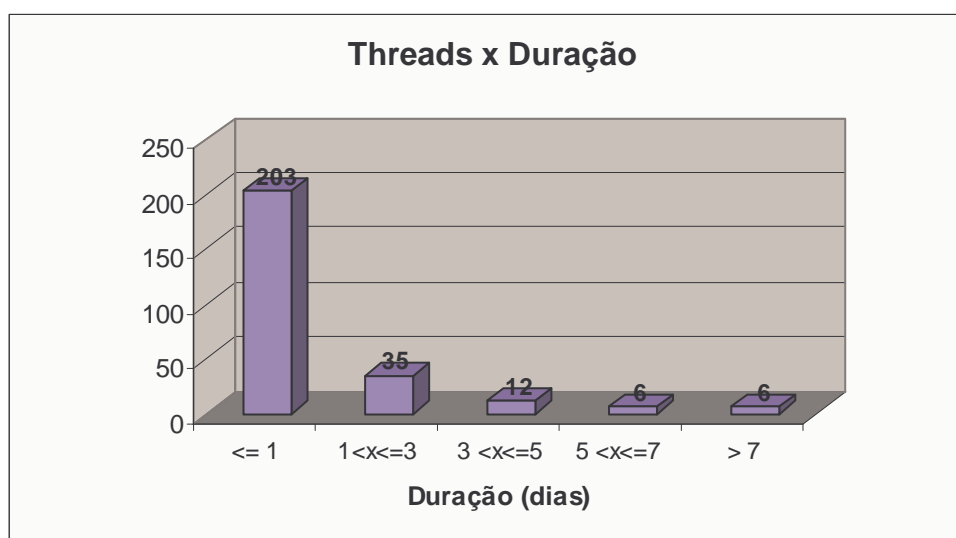
Para a análise da duração, foram consideradas as datas das mensagens mais antigas e mais recentes de cada *thread*. Desta forma, a duração foi calculada a partir da diferença entre a maior e a menor data das mensagens de um *thread*, contada em dias.

De acordo com os dados coletados, a duração média entre os 261 *threads* mapeados foi de 0,95 dias.

A Tabela 3 e a Figura 31 apresentam a distribuição de *threads* de acordo com o agrupamento de duração, conforme proposto no Quadro 24.

**Tabela 3 - Distribuição de *threads* x duração**

Duração (dias)	Quantidade de <i>Threads</i>	Percentual (%)
Menor ou igual a 1	203	77%
Entre 1 e 3	35	13%
Entre 3 e 5	12	5%
Entre 5 e 7	6	2%
Acima de 7	6	2%

**Figura 31 - Distribuição de *threads* x duração**

## 5.5 AVALIAÇÃO DOS RESULTADOS OBTIDOS

Como já citado anteriormente, durante o período de coleta, foram capturadas 7.902 mensagens, agrupadas em 7.288 *threads*, sendo consideradas para efeito de análise dos resultados apenas os 261 *threads* com mais de uma mensagem, reduzindo este universo a 889 mensagens, ou seja, 11,25% das mensagens interceptadas e registradas na base de dados do *ThreadMap*.

Este grande volume de *threads* com apenas uma mensagem pode ser atribuído aos seguintes fatores:

- Grande volume de mensagens *spam* – Esta informação foi passada pela equipe da GETEC, que está inclusive trabalhando na implantação de uma ferramenta para *anti-spam*.
- Ferramenta de Gerência de Ambiente – A GETEC utiliza a ferramenta UNICENTER TNG da *Computer Associates* (CA) para o monitoramento e gerenciamento dos servidores e serviços disponíveis e todos os alertas gerados por ela são enviados via correio eletrônico para os analistas de suporte da equipe.
- Falha identificada no serviço de eventos do MS Exchange, que teve como consequência a criação de um novo *thread* quando a mensagem foi capturada, mas o seu conteúdo não foi alterado, conforme descrito na seção 5.3.

Segundo os dados coletados, dentre os 261 *threads* analisados, 84, ou seja, 32% destes foram realizados com a troca de apenas duas mensagens, a participação de até quatro pessoas e com duração máxima de um dia, caracterizando assim este como o *thread* típico da equipe GETEC.

Como pode ser visto pelo gráfico apresentado na Figura 29, percebe-se uma grande concentração de *threads* com até cinco mensagens trocadas, representando 88% dos *threads*, indicando principalmente a realização de curtas conversações utilizando o correio eletrônico.

A Figura 30 indica um decréscimo acentuado no número de *threads* à medida que aumenta o número de participantes, com uma consequente concentração de *threads* de até 8 participantes, representado 84% do total.

Algo semelhante pode ser percebido na figura 31, que analisa a duração dos *threads*, onde 91% destes ocorreram em até 3 dias.



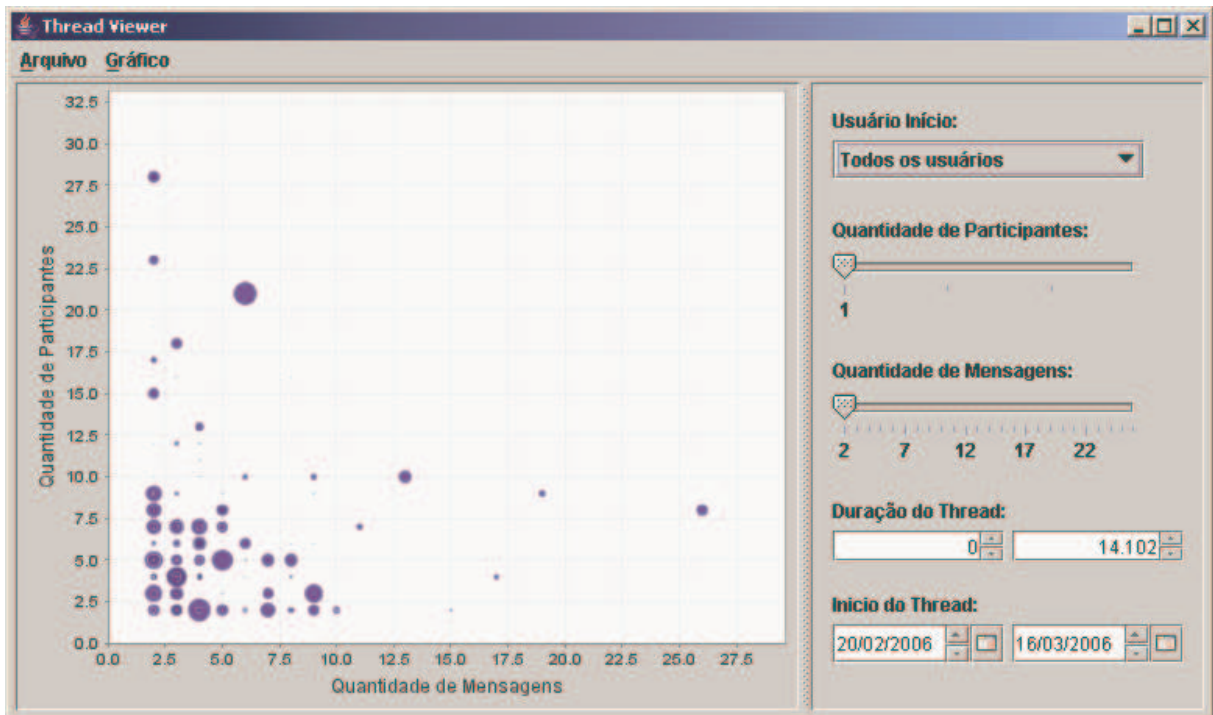


Figura 32 - ThreadViewer – Visão dos threads coletados na SEFAZ

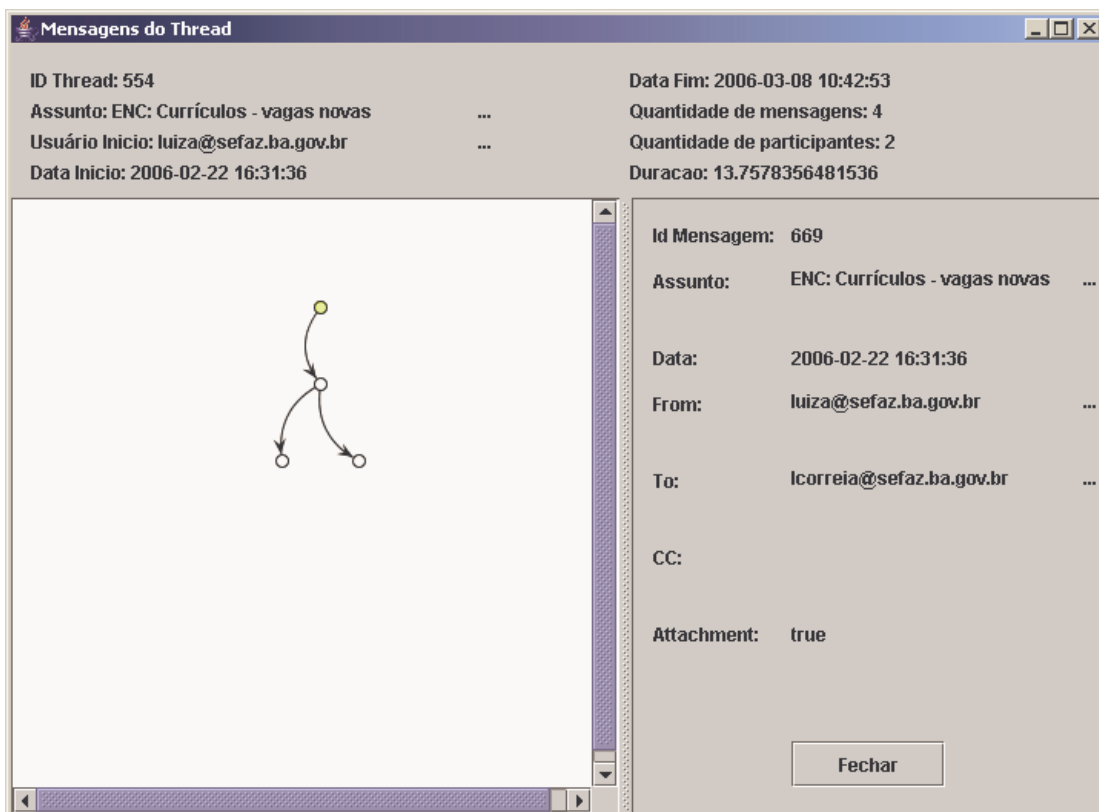


Figura 33 - ThreadViewer – Detalhamento de um thread coletados na SEFAZ

As Figuras 32 e 33 apresentadas, trazem uma visão de todos os *threads* mapeados, utilizando o *ThreadViewer* e o exemplo de uma árvore de mensagens trocadas por um *thread*.

Como pôde ser observado, com exceção da falha encontrada no serviço de eventos do MS Exchange Server, o estudo de caso realizado permitiu verificar o adequado funcionamento da ferramenta *ThreadMap*, podendo ser utilizada em organizações reais, sem impacto para o uso do correio eletrônico pelos seus usuários.

## 5.6 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentada a aplicação do estudo de caso do *ThreadMap* realizado na Secretaria da Fazenda do Estado da Bahia, no período de 20 de fevereiro de 2006 a 16 de março de 2006, com a captura de 7.902 mensagens e uma análise dos resultados obtidos.

No próximo capítulo será discutida a conclusão deste trabalho de dissertação.

## CAPÍTULO 6 - CONCLUSÃO

O correio eletrônico é uma ferramenta bastante difundida entre os usuários consumidores de tecnologia da informação e o seu uso representa uma importante fonte de informações.

Uma das utilizações comuns do correio eletrônico é possibilitar a integração de pessoas geograficamente distribuídas, encurtando as distâncias e facilitando a disseminação e troca de informações entre grupos de trabalho ou pessoas interessadas em assuntos em comum. Esta interação propicia a realização de conversações de uma forma virtual, entre indivíduos geograficamente espalhados, sem a necessidade de que todos estejam disponíveis ao mesmo tempo, pelo caráter assíncrono desta ferramenta.

Estas conversações através do correio eletrônico, uma vez mapeadas, permitem a montagem de uma base de dados que auxilia as organizações a entenderem o uso desta ferramenta por suas equipes de trabalho e permite a identificação de fluxos de informação ainda não conhecidos.

Este trabalho teve como objetivo principal identificar as alternativas atualmente utilizadas para o mapeamento destes *threads* de mensagens eletrônicas e apresentar uma solução que minimizasse as possíveis perdas de trechos destes *threads* e permitisse o seu mapeamento de uma forma mais abrangente.

Para tal, foi apresentada a solução *ThreadMap*, que permite o mapeamento dos *threads* a partir da interceptação e captura dos eventos de troca de mensagens entre um grupo de usuários. Esta solução foi implementada utilizando o servidor de correio eletrônico MS Exchange Server, que assim como outros, permite esta manipulação.

Para permitir a visualização dos dados coletados pelo *ThreadMap*, foi apresentada o *ThreadViewer*, uma ferramenta que permite a identificação e localização de um *thread* a partir de informações como duração, quantidade de mensagens, quantidade de participante e período e posterior visualização dos detalhes das mensagens que o compõem, organizadas em um formato de árvore.

Com o objetivo de avaliar a implementação realizada em uma situação real, foi realizado um estudo de caso na Secretaria da Fazenda do Estado da Bahia, envolvendo 19 pessoas da equipe de Tecnologia, entre 20 de fevereiro de 2006 e 16 de março de 2006. Durante este período, foram capturadas 7.902 mensagens e analisados 261 *threads* que tiveram mais de uma mensagem trocada. Neste estudo, ficou caracterizado o uso de correio eletrônico para a realização de conversações curtas, com 32% dos *threads* analisados, sendo realizados com apenas 2 mensagens trocadas envolvendo 1 a 4 participantes, durante um dia.

A principal dificuldade encontrada durante a elaboração deste trabalho foi a falha encontrada no servidor de eventos do MS Exchange Server, onde algumas mensagens apesar de capturas e registradas não tiveram o seu conteúdo alterado pelo *ThreadMap*.

## 6.1 CONTRIBUIÇÕES

Baseado no objetivo principal de permitir o mapeamento das trocas de mensagens de correio eletrônico realizadas entre um grupo de pessoas, este trabalho traz as seguintes contribuições:

- A apresentação de uma proposta de mapeamento baseada na interceptação de mensagens no servidor de correio eletrônico, minimizando a possibilidade de perda de trechos destes *threads*;

- A realização de um mapeamento que possibilite a visão global do *thread*, ao contrário das soluções de mapeamento que utilizam as caixas postais dos usuários, que apresentam a visão de cada um dos envolvidos na conversação;
- A solução *ThreadMap* pode ser implementada em qualquer servidor de correio eletrônico que permita a interceptação e manipulação das mensagens trocadas entre os usuários;
- A apresentação de um estudo de caso que permitiu caracterizar os *threads* realizados por uma equipe de trabalho.

## 6.2 LIMITAÇÕES DO TRABALHO

Como o conceito do *ThreadMap* está baseado da possibilidade de interceptação de mensagens no servidor de correio eletrônico, a disponibilidade desta captura e o acesso às mensagens são limitações desta solução.

Outra limitação está no fato de que o mapeamento precisa ser realizado no momento em que as mensagens são trocadas entre os usuários, não sendo possível a realização deste mapeamento posteriormente, como são realizados pelas técnicas que utilizam como fonte de informações, as caixas postais dos usuários.

A solução *ThreadMap* pode atender a plataformas de correio eletrônico distribuídas, desde que esteja implantada em todos os servidores de correio eletrônico envolvidos, com a limitação de utilização de uma base de dados centralizada, de forma que seja possível manter o controle sobre o identificador do *thread*. Durante a troca de mensagens, caso exista a participação de um servidor de correio eletrônico externo, este trecho do *thread* não terá como ser mapeado.

### 6.3 TRABALHOS FUTUROS

É constante a necessidade de evolução e adaptação às mudanças de tecnologias disponíveis. A seguir, são listadas algumas indicações de trabalhos futuros para evolução da solução *ThreadMap*:

- Implementação da solução *ThreadMap* utilizando outros servidores de correio eletrônico que permitam a interceptação e captura das mensagens trocadas;
- Realização de outros estudos de caso, de forma a permitir a comparação entre os resultados obtidos;
- Evolução da ferramenta *ThreadViewer* para visualização dos *threads*, utilizando técnicas de mineração visual de dados para facilitar a localização de um *thread*;
- Definição e implementação de uma ferramenta que permita a visualização dos *threads* sob a ótica dos envolvidos, ou seja, que auxilie na identificação de fluxos de informações entre os usuários.
- Integração com ferramentas de análise do conteúdo das mensagens de forma que estas possam ser agrupadas por assunto e auxiliem nas análises de fluxo de informações.
- Integração com ferramentas de análise de comunicação organizacional para auxílio na identificação de fluxos de informações e análises de redes sociais.

Estes dois últimos itens, análise de conteúdo de mensagens e análise da comunicação organizacional, já foram temas de outras dissertações de mestrado na Unifacs.

A ferramenta proposta por Oliveira e Mendonça Neto (2004) permite a classificação automática de textos, podendo ser aplicada na categorização de mensagens de correio eletrônico.

Andion e Mendonça Neto (2005) apresentam uma ferramenta que permite a análise da comunicação organizacional através da visualização de grafos e análise dos fluxos de

comunicação em redes sociais. A integração do trabalho aqui realizado com os dois trabalhos acima mencionados, cria a possibilidade de criação de um mecanismo poderoso de análise da comunicação organizacional a partir dos fluxos de mensagens correio eletrônico de uma empresa.

#### 6.4 CONSIDERAÇÕES FINAIS

Este trabalho apresentou a proposta de uma solução para o problema de mapeamento de mensagens de correio eletrônico. O *ThreadMap* é uma ferramenta baseada na interceptação das mensagens no servidor de correio eletrônico, no momento do seu recebimento. Durante este trabalho foram apresentadas a revisão bibliográfica sobre o assunto, o detalhamento da solução *ThreadMap*, uma implementação utilizando o MS Exchange Server, da interface de visualização *ThreadViewer*, a aplicação de um estudo de caso na Secretaria da Fazenda do Estado da Bahia e seus resultados, como também as contribuições, dificuldades e possíveis trabalhos futuros.

## REFERÊNCIAS

ANDION, I.; MENDONÇA NETO, M. G.. Uma Ferramenta para Análise da Comunicação Organizacional Através de Redes Sociais. In: Simpósio Brasileiro de Sistemas de Informação, 2005, Florianópolis. **Anais ...** Florianópolis: 2005. v. 1. p. 1-8.

BAHIA. Decreto nº 7.921 de 02 de abril de 2001. Aprova o regimento da Secretaria da Fazenda. **Diário Oficial do Estado da Bahia**, Poder Executivo, Salvador, BA, 03 abr. 2001.

BALTER, O. **Electronic mail in a working context**. 1998. 234 f. Ph.D. thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm.

CULOTTA, A.; BEKKERMAN, R.; McCALLUM A. Extracting social networks and contact information from email and the web. In: CONFERENCE ON EMAIL AND ANTI-SPAM, 1., 2004, Mountain View. **Anais eletrônicos...** Mountain View, 2004. Disponível em: <<http://www.ceas.cc/papers-2004/176.pdf>>. Acesso em: 17 ago. 2005.

FERRIS RESEARCH. **Electronic Message Archiving**. 2001. Disponível em: <<http://www.ferris.com>>. Acesso em: 04 nov. 2005.

FISHER, D.; MOODY, P. **Studies of automated collection of email records**: technical report UCI-ISR-02-4. Institute of Software Research – University of California, Irvine, 1 ago. 2002. Disponível em: <<http://www.isr.uci.edu/tech-reports.html>>. Acesso em: 08 abr. 2005.

FISHER, D.; MULLER, M. **Conducting quantitative studies of email records**. Science Department University of California, Berkeley, 2000. Disponível em: <[www.ics.uci.edu/~danyelf/publications/lotus\\_chi2001.pdf](http://www.ics.uci.edu/~danyelf/publications/lotus_chi2001.pdf)>. Acesso em: 12 set. 2005.

GILBERT, D. **The JFreeChart Class Library**. 2005. Disponível em: <<http://www.jfree.org/jfreechart/index.php>>. Acesso em: 15 set. 2005.

HECKEL, B.; HAMANN, B. EmVis – A Visual E-mail Analysis Tool. In: WORKSHOP ON NEW PARADIGMS IN INFORMATION VISUALIZATION AND MANIPULATION, 1997, Las Vegas. **Anais eletrônicos...** Las Vegas, 1997. Disponível em: <<http://portal.acm.org/>>. Acesso em: 22 ago. 2005.

HYPERMAIL. **Hypermail Documentation Overview**. [Site Institucional]. Disponível em: <<http://www.hypermail.org>>. Acesso em: 20 mai. 2005.

IETF. **Internet Engineering Task Force**. [Site Institucional]. Disponível em: <<http://www.ietf.org/overview.html>>. Acesso em: 05 fev. 2006.



IMAP EXTENSION WORK GROUP. *Internet message access protocol: sort and thread extensions*. 2004. Disponível em: <<http://www.ietf.org/internet-drafts/draft-ietf-imapext-sort-17.txt>>. Acesso em: 5 jan. 2006.

KERR, B. Thread Arcs: An Email Thread Visualization. In: IEEE Symposium on Information Visualization, 2003, Seattle. **Anais eletrônicos...** Seattle: IEEE, 2003. Disponível em:<<http://www.infovis.org/infovis2003>>. Acesso em: 2 abr. 2005.

KLAAS M. **Toward indicative discussion fora summarization**. Department of Computer Science, University of British Columbia, Vancouver, 2004. Disponível em: <<http://www.cs.ubc.ca/~klaas/research/usenetsummary.pdf>>. Acesso em: 10 nov. 2005.

LEWIS, D.; KNOWLES K. **Threading electronic mail**: a preliminary study. Information Processing and Management, [s.l.], mar. 1997. Disponível em: <<http://portal.acm.org/>>. Acesso em: 20 mar. 2005.

MADADHAIN, J. et al. **The java universal network/graph (JUNG) framework**. Technical Report UCI-ICS 03-17. School of Information and Computer Science – University of California, Irvine, 2003. Disponível em: <[http://www.datalab.uci.edu/papers/JUNG\\_tech\\_report.html](http://www.datalab.uci.edu/papers/JUNG_tech_report.html)>. Acesso em: 09 out. 2005.

MICROSOFT. **Microsoft Exchange SDK Development Tools**. 2005a. Disponível em: <<http://www.microsoft.com/>>. Acesso em: 10 mai. 2005.

\_\_\_\_\_. Exchange é líder em e-mail. **Revista Windows Server System Update**, [s.l.], dez. 2005b. Disponível em: <<http://www.microsoft.com/brasil/revistawss/>>. Acesso em: 10 fev. 2006.

MOODY, P. Reinventing Email. In: CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, 2002, New Orleans. **Anais eletrônicos...** New Orleans, 2002. Disponível em: <<http://www.acm.org/cscw2002/>>. Acesso em: 22 abr. 2005

NYAMGONDALU, N. **C API Programming for Lotus Notes/Domino**. 2005. Disponível em: <<http://www-128.ibm.com/developerworks/lotus/library/capi-nd/>>. Acesso em: 12 dez. 2005.

NENKOVA A.; BAGGA, A. Facilitating Email Thread Access by Extractive Summary Generation. In: INTERNATIONAL CONFERENCE RECENT ADVANCES IN NATURAL LANGUAGE PROCESSING, 2003, Borovets. **Anais eletrônicos...** Borovets, 2003. Disponível em: <<http://lml.bas.bg/ranlp2003/>>. Acesso em: 05 mai 2005.

NEUSTAEDTER, C.; SMITH, M.; VELONIA, G. **Grand central usenet**: the design and evaluation of a thread-based usenet browser. Microsoft Tech Report MSR-TR-2003-89, 2003. Disponível em: <<http://pages.cpsc.ucalgary.ca/~carman/research.html>>. Acesso em: 07 jun. 2005.

NEWMAN, P. **Treetables and other visualizations for email threads**. Palo Alto Research Center, 2001. Disponível em: <[http://www2.parc.com/istl/groups/hdi/papers/psn\\_emailvis01.pdf](http://www2.parc.com/istl/groups/hdi/papers/psn_emailvis01.pdf)>. Acesso em: 12 ago. 2004.

OLIVEIRA, G; MENDONÇA NETO, M. G.. ExperText: uma ferramenta de combinação de múltiplos classificadores Naive Bayes. In: 4a Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería de Conocimiento, 2004, Madri. **Anais...** Madri: 2004. v. 1. p. 317-332.

PALME, J. **Message Threading in E-mail Software**, 1998. Disponível em: <<http://dsv.su.se/jpalme/ietf/message-threading.pdf>>. Acesso em: 12 ago. 2004.

RAMBOW O. et al. Summarizing Email Threads. In: HUMAN LANGUAGE TECHNOLOGY CONFERENCE, 2004, Boston. **Anais eletrônicos...** Boston, 2004. Disponível em: <<http://www.cs.columbia.edu/~pablo/hlt-naacl04/>>. Acesso em: 25 mai. 2005.

NETWORK WORKING GROUP. **Post Office Protocol Version 3 - RFC1225**, 1991. Disponível em: <<http://www.rfc-editor.org/rfc/>>. Acesso em: 05 fev. 2005.

\_\_\_\_\_. **SMTP Service Extensions – RFC1869**, 1995. Disponível em: <<http://www.rfc-editor.org/rfc/>>. Acesso em: 05 fev. 2005.

\_\_\_\_\_. **Common Internet Message Headers - RFC2076**, 1997. Disponível em: <<http://www.rfc-editor.org/rfc/>>. Acesso em: 20 fev. 2005.

\_\_\_\_\_. **Simple Mail Transfer Protocol – RFC2821**, 2001a. Disponível em: <<http://www.rfc-editor.org/rfc/>>. Acesso em: 20 fev. 2005.

\_\_\_\_\_. **Internet Message Format - RFC2822**, 2001b. Disponível em: <<http://www.rfc-editor.org/rfc/>>. Acesso em: 5 fev. 2005.

ROHALL, S. Redesigning Email for the 21th Century Wokshop Position Paper. In: CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, 2002, New Orleans. **Anais eletrônicos...** New Orleans, 2002. Disponível em: <<http://www.acm.org/cscw2002/>>. Acesso em: 22 abr. 2005.

ROHALL, S.; GRUEN, D. Remail: A Reinvented Email Prototype. In: CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, 2002, New Orleans. **Anais eletrônicos...** New Orleans, 2002. Disponível em: <<http://www.acm.org/cscw2002/>>. Acesso em: 18 mai. 2005.

ROHALL, S. et al. Email Visualization to Aid Communications. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 2001, San Diego. **Anais eletrônicos...** San Diego, 2001. Disponível em: <<http://www.infovis.org/infovis2001/>>. Acesso em 17 ago. 2005.

SACK W. Conversation Map: an interface for very-large-scale conversations. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEMS SCIENCES, 2000, Hawaii. **Anais eletrônicos...** Hawaii, 2000. Disponível em: < <http://www.hicss.hawaii.edu/> >. Acesso em: 12 jul. 2005.

VENÓLIA, G.; NEUSTAEDTER, C. Understanding Sequence and Reply Relationships within Email Conversations: a mixed-model visualization. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEM, 2003, Fort Lauderdale. **Anais eletrônicos...** Fort Lauderdale, 2003. Disponível em: < <http://www.chi2003.org/index.cgi> >. Acesso em: 25 mai. 2005.

VENÓLIA, G. et al. **Supporting email workflow**. Microsoft Technical Report MSR-TR-2001-88, 2001. Disponível em: <[http://research.microsoft.com/research/pubs/view.aspx?msr\\_tr\\_id=MSR-TR-2001-88](http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-2001-88)>. Acesso em: 15 fev. 2005.

WHITTAKER, S.; SIDNER, C. Email Overload: exploring personal information management of email. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEM, 1996, Vancouver. **Anais eletrônicos...** Vancouver, 1996. Disponível em: < <http://www.acm.org/sigchi/chi96/> >. Acesso em: 17 abr. 2005.

YATES, J. et al. Virtual Organizing: using threads to coordinate distributed work. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEM, 36., 2003, Hawaii. **Anais eletrônicos...** Hawaii, 2003. Disponível em: < <http://www.hicss.hawaii.edu/hicss36/apahome36.htm> >. Acesso em: 19 mai. 2005.

ZAWINSKI, J. **Message Threading**, 1997. Disponível em: <<http://www.jwz.org/doc/threading.html>>. Acesso em: 20 ago. 2004.