Fundação Educacional do Município de Assis

Rodrigo Poletini El Chammas

COMPONENTE JAVA PARA ESTENOGRAFIA E TRANSMISSÃO VIA INTERNET

Assis – SP - Brasil
2009

COMPONENTE JAVA PARA ESTENOGRAFIA E TRANSMISSÃO VIA INTERNET

Rodrigo Poletini El Chammas

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação, analisado
pela seguinte comissão examinadora:

Orientador: _____

Analisador (1):_____

Analisador (2):_____

Assis – SP - Brasil
2009

Rodrigo Poletini El Chammas

COMPONENTE JAVA PARA ESTENOGRAFIA E TRANSMISSÃO VIA INTERNET

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação, analisado
pela seguinte comissão examinadora:

Orientador:_____

Área de Concentração:_____

Assis – SP - Brasil
2009

DEDICATÓRIA

Dedico este trabalho aos meus pais que me deram

forças para conclusão do mesmo.

AGRADECIMENTOS

Agradeço primeiramente a Deus por estar aqui neste momento tão glorioso da minha vida.

Agradeço ao meu orientador Profo. Felipe A.C. Pazinatto pela paciência, pela dedicação e motivação comigo neste projeto.

Agradeço a Profa. Dra. Marisa Atsuko Nitto por toda ajuda que me deu, e por sua generosidade incrível que passa a todos.

Agradeço aos meus familiares que tiveram a paciência de me acompanhar durante todos esses anos de graduação.

Agradeço a todos os professores e amigos que participaram diretamente e indiretamente neste projeto.

“Um homem que não se alimenta de seus sonhos, envelhece cedo.”

William Shakespeare

RESUMO

Muitas soluções tecnológicas surgem a cada dia, resolvendo soluções às vezes bastante antigas ou até mesmo algumas inéditas. Com isso vem à otimização de tempo, que é um dos principais objetivos deste projeto.

As principais características deste projeto será codificar mensagens dentro de uma imagem digital, usando a estenografia, e enviá-las via internet usando o protocolo TCP/IP, e recuperá-la posteriormente usando um algoritmo.

Palavras Chaves: estenografia, internet, tcp/ip, sockets.

ABSTRACT

Many technologics products appear all days with the aim of to solve sometimes very old or even some news problems. With that comes the optimization of time, which is one of the main objectives of this project.

The main features of this project are to hide messages inside a digital image using stenography, and send it via internet using the TCP / IP protocol, for recovery it after using a algorithm.

Keywords: stenography, internet, tcp/ip, sockets.

LISTA DE ILUSTRAÇÕES

SUMARIO

1. INTRODUÇÃO

1.1. PROCESSAMENTO DE IMAGENS

O processamento de imagens vem evoluindo com o decorrer do tempo. Imagens antes transmitidas por telégrafo usando cabos submarinos hoje trafegam comumente na internet. Novas técnicas foram desenvolvidas para o trabalho com imagens, novas formas de analisar as partes integrantes, novas formas de discretização foram criadas, permitindo que a mesma imagem fosse devidamente tratada, inserida em *web sites*, transmitida, e vista por todo o mundo de forma rápida e eficiente.

Uma imagem, antes de ser trabalhada por um programa de computador, precisa ter seu espaço característico transformado. Uma imagem, antes analógica, contínua no espaço, deve possuir sua intensidade luminosa discretizada pontualmente e os valores guardados em uma matriz de intensidades luminosas. Assim, ao trabalhar com imagens, deve-se levar em consideração que as mesmas estão guardadas em matrizes na forma bidimensional, apresentando um eixo amostral X e Y, ou no domínio funcional, Intensidade=f(x, y).

1.2. REDES DE COMPUTADORES

Durante o inicio de sua existência as redes de computadores eram usadas mais por estudantes a fim de enviar mensagem de correio eletrônico, e também por funcionários de empresas para compartilhar impressoras. Não havia necessidade de se preocupar com a segurança dos dados transmitidos. Com o avanço da internet para todas as partes do mundo, com milhares de usuários fazendo suas transações bancárias online, comprando em sites com cartões de credito e compartilhando dados, a segurança passou a ser um item extremamente importante. [2]

Essa quebra na segurança é feito por mais diversas tipos de pessoas, às vezes intencionalmente por pessoas maliciosas que tentam quebrar algum tipo de segurança para se dar bem ou até mesmo se vingar de algo que tenha acontecido [2]. Alguns invasores são demonstrados na Tabela1 abaixo.

Adversário	**Objetivo**
Estudante	Divertir-se bisbilhotando as mensagens de correio eletrônico de outras pessoas
Cracker	Testar o sistema de segurança de alguém; roubar dados
Representante de vendas	Tentar representar toda Europa e não apenas Andorra
Executivo	Descobrir estratégia de marketing do concorrente
Ex-funcionário	Vingar-se por ter sido demitido
Contador	Desviar dinheiro de uma empresa
Corretor de valores	Negar uma promessa feita a um cliente através de uma mensagem de correio eletrônico
Vigarista	Roubar números de cartão de credito e vendê-los
Espião	Descobrir segredos militares ou industriais de um inimigo
Terrorista	Roubar segredos de armas bacteriológicas

Tabela 1: Algumas pessoas que podem causar problemas de segurança e os motivos para fazê-los [2]

Segundo Tanenbaum [2], para uma rede ser totalmente segura, é necessário saber lidar com freqüência com adversários inteligentes, dedicados e, às vezes, muito bem subsidiados.

Para se manter a segurança em redes de computadores devem-se criptografar mensagens, tais como, números de cartões de crédito, números de contas bancarias, senhas de login, etc.

1.3. CRIPTOGRAFIA

Criptografia é uma cifra. Uma cifra é uma transformação de caractere por caractere ou de bit por bit. Na prática, um algoritmo de criptografia troca um caractere por outro caractere ou por símbolos especiais [2]. Em outras palavras, são métodos de transformar uma palavra legível, em outra totalmente diferente, ilegível. Apenas o destinatário desta mensagem saberá decifrá-la, reconstruindo assim a mensagem original.

1.4. ESTENOGRAFIA

Junto à criptografia, pode-se usar a estenografia, que significa esconder informações dentro de uma imagem, áudio ou vídeo.

Ao contrario da criptografia que embaralha as informações, a estenografia tem como objetivo esconder informações no domínio de uma imagem de forma a parecer que existe somente a imagem e nenhuma outra informação.

Outra característica das imagens estenografadas está no fator privacidade, onde dados nela inseridos podem ser disponibilizados de forma acidental ou por violação forçada (invasão, espionagem). A problemática de segurança da informação incide diretamente sobre o que está sendo transmitido em uma

rede, e formas de manter em sigilo devem ser usadas para garantir que a privacidade não seja quebrada sob nenhuma circunstância.

1.5. OBJETIVO

Este trabalho visa construir um componente de software para esconder dados dentro de uma imagem usando estenografia e transmiti-la via internet para um servidor remoto.

Como objetivos específicos destacam-se:

i. Construir uma aplicação para digitar informações;

ii. Usar o método LSB para esconder essas informações em uma imagem;

iii. Enviar uma imagem via Web, utilizando sockets em Java.

1.6. JUSTIFICATIVA

Como resposta a estas questões, é objetivo deste trabalho criar um componente em Java que encapsule a complexidade de estenografar, compactar, transmitir e garantir que os dados sejam recebidos integralmente em uma conexão cliente-servidor usando o protocolo TCP.

1.7. ESTRUTURA DO TRABALHO

Este projeto será subdivido em capítulos, os quais serão explicados a seguir.

O primeiro capítulo abordará uma introdução e justificativas propostas no trabalho.

O segundo capítulo apresentará toda parte teórica que envolverá criptografia, estenografia, redes de computadores, processamento de imagem e a ferramenta usada para desenvolver o trabalho usando Java.

O terceiro capítulo mostrará etapa por etapa toda a estrutura de desenvolvimento junto à modelagem deste projeto.

O quarto capítulo mostrará toda a implementação do software, contendo interfaces, códigos e explicações sobre os mesmos.

O quinto capítulo apresentará as considerações finais, as conclusões a que se chegaram e as dificuldades encontradas.

O sexto capítulo conterá as referencias bibliográficas que foram utilizadas para estudos e referencias neste projeto.

2. FUNDAMENTAÇÃO TÉCNICA BÁSICA

2.1. CRIPTOGRAFIA

Criptografia vem das palavras gregas “Kryptós” e “gráphein”, que significam “ocultos” e “escrever”, respectivamente. Esta técnica tem como objetivo pegar uma mensagem com total legibilidade e codificá-la, fazendo com que ela se embaralhe toda, fugindo totalmente do seu escopo original [2]. Como demonstra a Figura1.

Figura 1: Exemplo de um documento Criptografado.

Esta arte é uma necessidade muito antiga que vem muito antes de qualquer maquina que se possa imaginar. Nas guerras medievais, onde existiam mensageiros, as mensagens transportadas por estes, eram muitas vezes confidenciais. Conseqüentemente era necessário cifrá-las, pois se caísse em mãos erradas à mensagem não seria decifrada facilmente. Apenas o emissor e o receptor, detentores do código para de criptografar saberiam decifrar esta mensagem.

Segundo Tanenbaum,

> O código mais bem-sucedido já inventado foi usado pelas forças armadas dos Estados Unidos durante a segunda guerra mundial no Pacífico. Eles simplesmente tinham índios navajo que se comunicavam uns com os outros usando palavras navajos especificas para termos militares como, por exemplo, chay-dagahi-mail-tsaidi (literalmente, assassino cágado) para indicar uma arma antitanque [2].

Com o surgimento da internet esta troca de mensagens ficou muito mais fácil e rápida, porem tornou muito mais vulnerável o acesso de terceiros sobre estas, levando a criptografia a se tornar uma ferramenta fundamental para estas transações. Varias técnicas são aplicadas para se criptografar, e em computação umas das mais comuns é o conceito de chaves criptográficas. Trata-se de um conjunto de bits e um determinado algoritmo capaz de codificar e decodificar informações usando esse conjunto. Se o receptor usar outra chave incompatível com a chave do emissor, não conseguirá extrair a informação.

Com o uso de chaves, o emissor pode usar o mesmo algoritmo para vários receptores. Basta que cada um receba uma chave diferente. Esta chave funciona de acordo com o tamanho, ou seja, pelas quantidades de “bits” que foram colocadas.

Exemplo: se for criada uma chave com 8 “bits”, apenas 256 chaves serão usadas na decodificação, pois, 2 elevado à potência de 8 é igual a 256. Isto é, apenas 8 bits não é seguro, pois pode ser quebrado usando ataque de força bruta.

Já se for criada com 128 bits terá aproximadamente 340.282.366.920.938.463.374.607.431.771.638 chaves para ser usada na decodificação. Existem dois tipos de chaves: Simétricas e Assimétricas.

2.1.1. Chaves Simétricas e chaves Assimétricas

2.1.1.1. Chaves Simétricas

A Chave Simétrica se utiliza de um conceito mais simples, onde o emissor e receptor fazem o uso da mesma chave para criptografar e decifrar (Figura2). Existem vários algoritmos para este método. Entre os mais famosos está o DES.

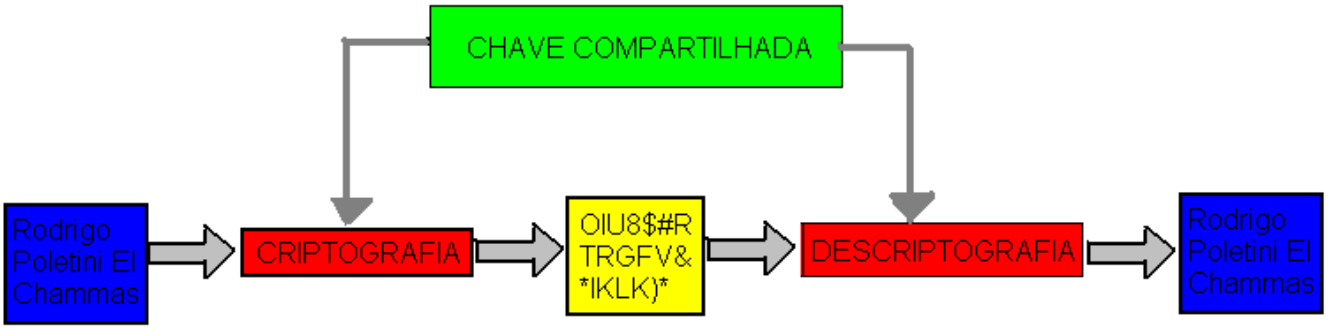

Figura 2: Modelo de criptografia convencional.

2.1.1.1.1. Algoritmo de criptografia – DES

DES (Data Encryption Standard) – criado pela IBM em 1977, usa uma chave de 56 bits. Isto gera quase 72 quatrilhões de chaves. Parece um valor bem robusto, mas não é nada para os computadores de hoje em dia.

O algoritmo de codificação é parametrizado por uma chave de k de 56 bits e possui 19 estágios diferentes (Figura3), sendo que o primeiro estágio realiza uma transposição dos dados modificando-os, e o ultimo estágio faz o inverso do primeiro voltando assim os dados em suas formas originais. Os demais estados são funcionalmente idênticos [4].

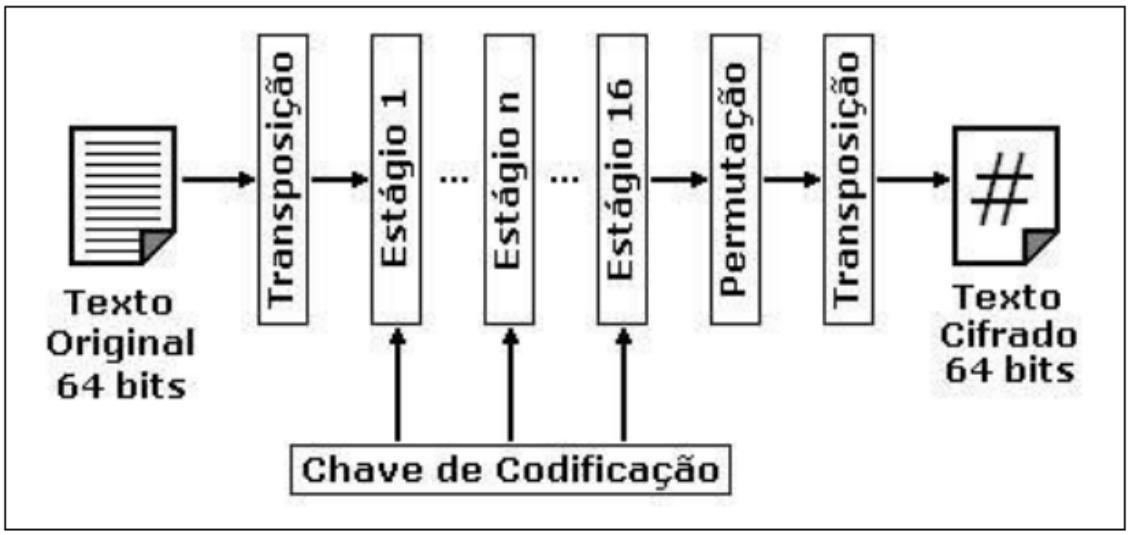

Figura 3: Processo de criptografia usando DES [4]

2.1.1.2. Chaves Assimétricas

A criptografia assimétrica, também conhecida como “chave pública”, trabalha com duas chaves: uma denominada privada e outra pública. Neste método o emissor precisa criar uma chave para que todos seus receptores tenham acesso, e assim outros usando a mesma chave pública poderão cifrar a mensagem e enviar os dados de forma segura para o emissor. Esta é a chave publica. Outra chave vai ser criada, mas ficará apenas com o emissor. Ninguém mais terá acesso a ela, sendo a mesma conhecida como chave privada (Figura4). Dentre os algoritmos mais conhecidos sobre este método tem-se o RSA.

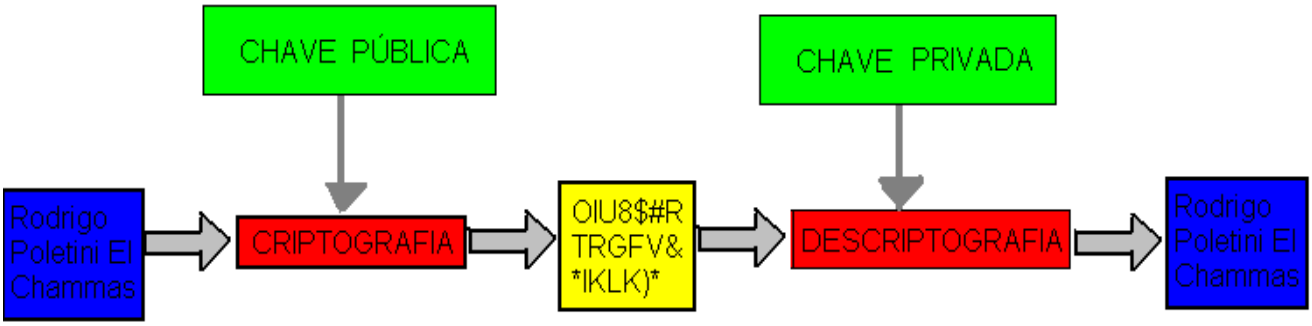

Figura 4: Processo de criptografia por Chave Pública

2.1.1.2.1. Algoritmo Assimétrico – RSA

RSA (Rivest, Shamir and Adleman) criado em 1977 por Ron Rivest, Adi Shamir e Len Adleman nos laboratórios do MIT (Massachusetts Institute of Technology), possui a seguinte característica: pegar dois números primos extremamente grandes e multiplicá-los, onde, os dois números primos serão a chave privada e o resultado desta multiplicação será a chave pública.

2.2. ESTENOGRAFIA

Estenografia vem do grego “steganós” que significa “oculto”, que consiste em ocultar informações de tal forma que não seja possível decifrá-la. Ao contrario da criptografia, a estenografia se preocupa em esconder a existência de mensagens em imagens, áudios e vídeos em vez de “embaralhar” as informações. Existem vários métodos para se estenografar, dentre elas:

- Tinta invisível;

- Técnicas de filtragem e mascaramento;

- Algoritmos e transformações;

- Inserção no bit menos significativo.

2.2.1. Tinta invisível

Este é um método manual, e constitui na inscrição de uma mensagem em um papel qualquer com uma tinta invisível, tornado visível quando se aplica outras técnicas. Exemplo:

- luz ultravioleta: algumas tintas se tornam florescente quanto expostas a luz ultravioleta;

- Ativação por calor: Tintas que em uma determinada temperatura oxidam e se revelam;

- Reações químicas: tintas que ao colocada em determinas substancias químicas se revelam;

2.2.2. **Inserção no bit menos significativo**

Mais conhecido como LSB (Least Significant Bit), ele esconde a mensagem nos bits menos significativos do pixel da imagem. Esses bits menos significativos são pouco variantes em termos de perda de qualidade.

Exemplo: a cor de um pixel pode variar de 0 a 255. Se for mudado um pixel de 255 para 254 não irá alterar quase em nada sua cor.

É exatamente isto que ocorre com os bits menos significativos. LSB pode ser aplicado nos bytes dos pixels. Cada pixel codificado tem 3 bytes. Um para o red(cores no tom vermelho na imagem), para o green(cores no tom verde na imagem) e outra para o blue(cores no tom azul na imagem), formando o RGB. A cada 3 pixels pode-se guardar um caractere. Como demonstram as Figuras 5 e 6 abaixo.

```
1 Pixel -> (10100011 11100011 00101101) [R, G, B]
1 Pixel -> (01010011 00011101 01110101) [R, G, B]
1 Pixel -> (00110110 11100011 01010101) [R, G, B]
```

Figura 5: Bits que compõe seus determinados pixels.

Agora será inserido (Figura 6) um caractere “a” que equivale a 97 em decimal na tabela ASCII e 1100001 em binário.

1 Pixel -> (10100011 11100011 00101100) [R, G, B]
1 Pixel -> (01010010 00011100 01110100) [R, G, B]
1 Pixel -> (00110110 11100011 01010101) [R, G, B]

Figura 6: Bits do caractere “a” inserido nos bits menos significativos da imagem.

Os números em vermelho são os bits que serão mixados à LSB e que esconderão o caractere “a”.

2.2.3. **Técnica de filtragem e mascaramento**

Filtragem e mascaramento fazem quase a mesma coisa que o LSB, mas ao invés de inserir nos bits menos significativos ele insere nos bits mais significativos. Mas para isso é preciso transformar as imagens em tons de cinzas para que não haja modificações dramáticas nos pixels da imagem. Este método é totalmente inviável para imagens coloridas.

2.3. REDES DE PETRI

Redes de Petri é uma modelagem representada através de um grafo direcionado com comentários. Possui nós de posições, transições, e arestas que indicam onde uma transição aponta para uma posição e vice e versa.

No Capitulo 3 será mostrado a Rede de Petri para o problema proposto no trabalho.

A figura7 abaixo ilustra um exemplo de Redes de Petri.

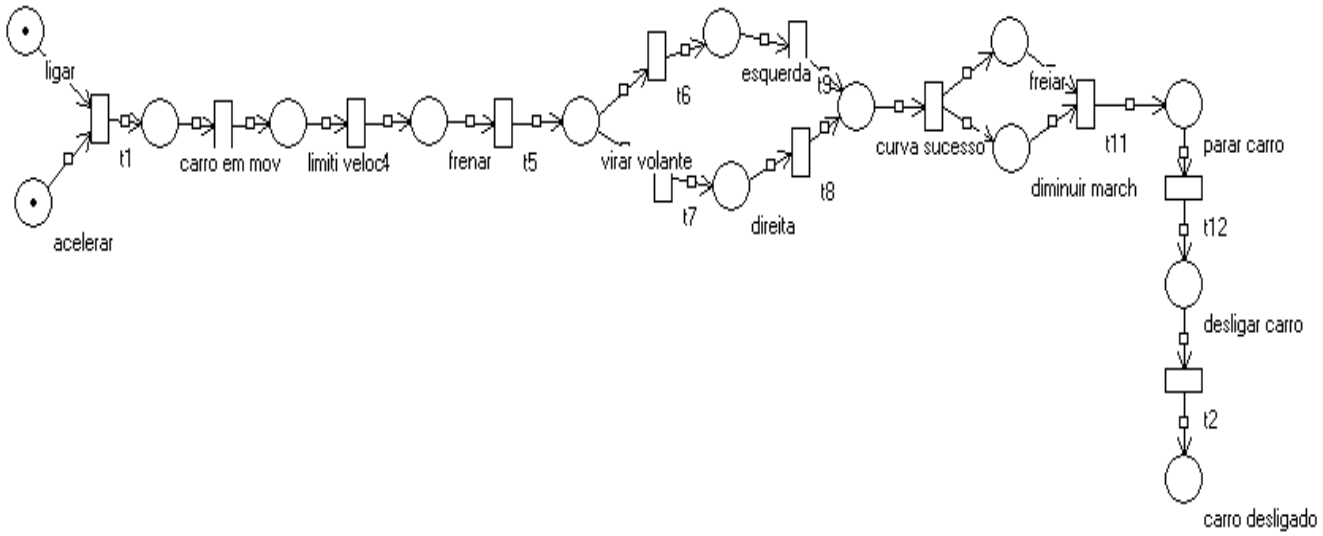

Figura 7: Modelagem de funcionalidades de um carro.

2.3.1. **Características da Rede de Petri**

A seguir serão mostradas todas as características usadas na Rede de Petri da Figura7.

2.3.1.1. Junção

Junção é a rede que sincroniza as atividades concorrentes. Na figura8 a transição t1 só será disparada caso houver fichas em “ligar” e “acelerar”, promovendo assim o sincronismo na rede.

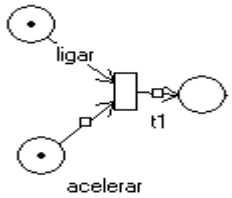

Figura 8: Junção

2.3.1.2. Seqüência

Seqüência é quando ocorrem várias sessões respectivamente, ou seja, uma após a outra, como ilustra a Figura9 e a Figura10 abaixo.

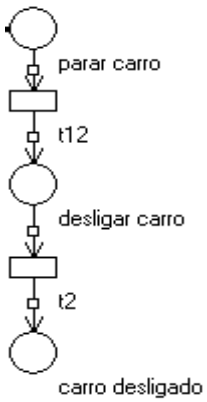

Figura 9: Seqüência

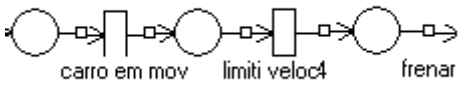

Figura 10: Seqüência

2.3.1.3. Caminhos alternativos divisão

Casos em que a execução continua por um caminho escolhido entre diversas alternativas, como mostra a Figura11 abaixo.

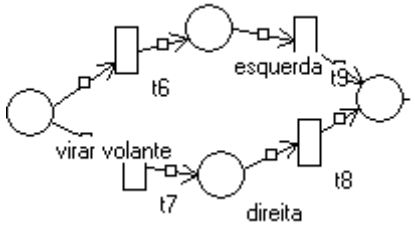

Figura 11: Conflito estrutural

2.3.1.4. Paralelismo

Paralelismo na rede é toda ação que ocorre junta com outra ação, ou seja, as duas vão ser executas ao mesmo tempo como demonstra a figura12 abaixo.

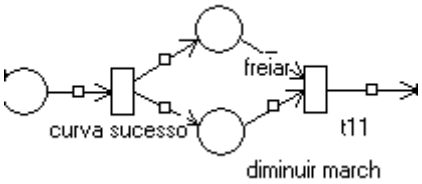

Figura 12: Paralelismo

2.4. PROCESSAMENTO DE IMAGEM

O Processamento de Imagens parte de uma imagem qualquer (geralmente vinda de uma câmera ou scanner) para se obter certas informações de acordo com a técnica aplicada. Seja ela, a compreensão de imagens, a analise estatística, a codificação, a transmissão de imagens, etc. Essas técnicas influem diretamente nos pixels da imagem e conseqüentemente nos bits da mesma [3]. Cada imagem tem a sua extensão. As mais usadas são:

- JPEG;
- PNG;
- BMP.

2.4.1. Pixel

Um pixel é o elemento principal em uma imagem. Imagens são nada mais que pequenos pontos luminosos denominados pixels. Uma imagem pode ter uma infinidade de pixels, de acordo com sua resolução e/ou extensão. A Figura 13 demonstra um zoom de sua imagem original, para demonstrar os pixels que à no espaço demarcado por um quadrado vermelho [7].

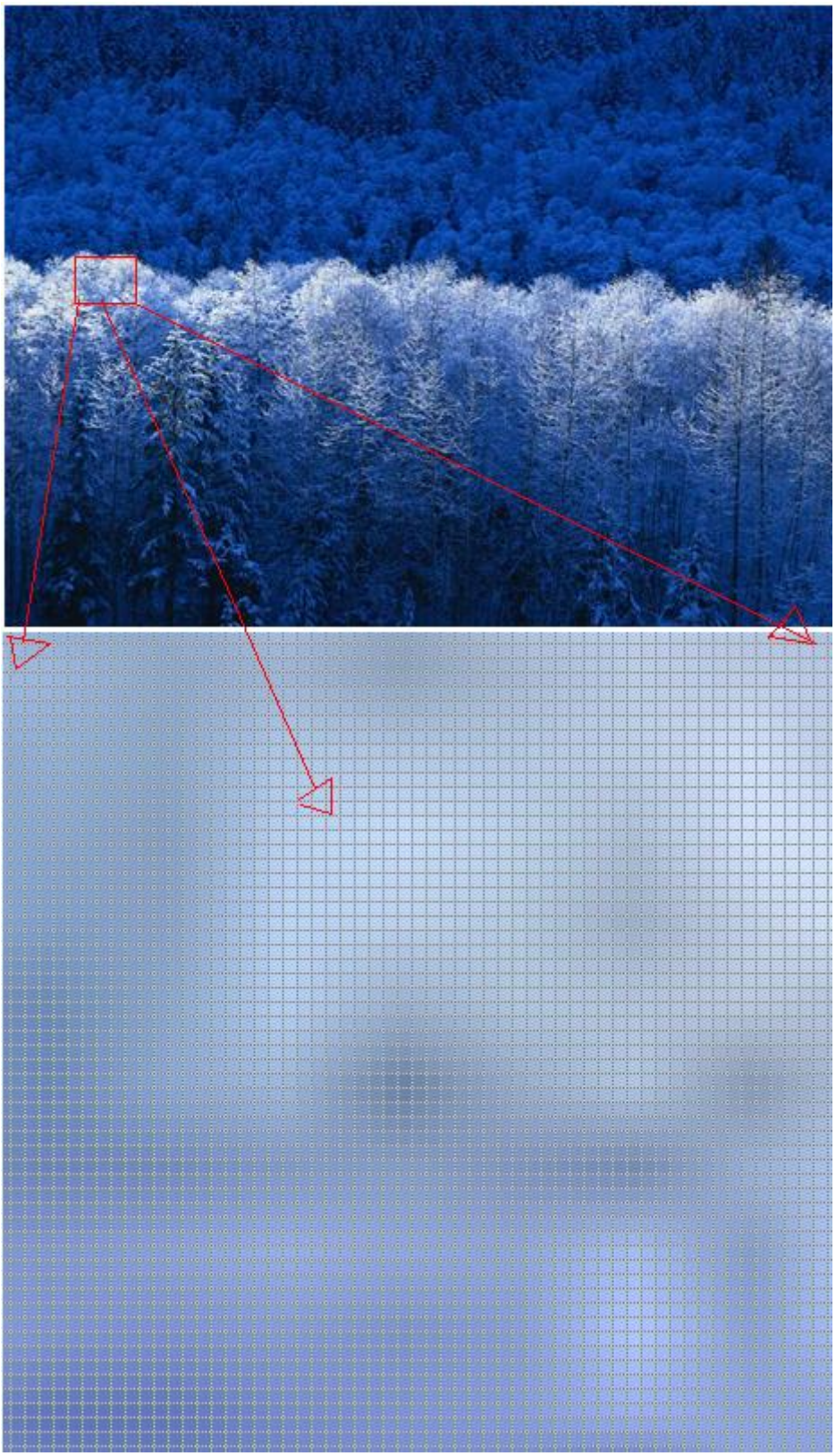

Figura 13: Exemplo de Pixel.

2.4.2. Extensões

2.4.2.1. JPEG

JPEG (Joint Photographic Experts Group) é uns dos formatos de imagem mais utilizados hoje em dia, devido ao seu algoritmo de compactação gerar imagens muitas vezes mais leves que as originais. Porem, quando essas compactações são feitas muitos dados são perdidos causando assim perda na qualidade da imagem. A maioria das câmeras digitais grava as fotos neste formato [8].

2.4.2.2. PNG

PNG (Portable Network Graphics) é um formato de imagem muito robusto, pois não perde dados e nem qualidade na imagem, além de suportar grande quantidade de informações, como canal alfa, correção de gama, verificação de intensidade, etc. [8]

2.4.2.3. BMP

BMP (Bitmap) é um formato de imagem já antigo da Microsoft. O MS Paint usa este formato para salvar suas imagens como padrão. Este formato suporta indexação e TrueColor, mas não é compactado. Também não suporta canal alfa, nem transparências [8].

2.5. REDES DE COMPUTADORES

Redes de computadores são computadores autônomos interconectados por uma única tecnologia. Só estará interconectado se trocarem informações entre si.

Existem vários tipos de redes e cada uma com seu tamanho, forma e modelo. Exemplo:

2.5.1. Extensão geográfica

- PAN (personal area network): rede de área pessoal de pouco alcance, apenas 10 metro. Serve apenas para executar atividades pessoais tais como, impressoras, mouse e teclado sem fio, aparelhos Bluetooth e etc.
- LAN (local area network): rede de área local a fim de trocar informações entre computadores conectados a esta rede. Seu alcance é de até 100 metros;
- CAN (campus area network): é uma rede que usa ligações inter prediais. São mais usadas em campus universitário e empresas que tenham várias centrais por perto, pois, seu alcance é de até 2.000 metros;
- MAN (metropolitan area network): é uma rede metropolitana que abrange uma cidade. Exemplo: uma rede de farmácias que usam o mesmo sistema pode-se usar uma MAN para deixar todas as farmácias da cidade interconectadas. Seu alcance é de 50.000 metros;
- WAN (wide area network): é uma rede geograficamente distribuída por seu alcance ser muito grande. É usada em nível de continente ou pais. Seu alcance é de 100.000 km. [6]

2.5.2. TCP/IP

A arquitetura do TCP/IP pode ser vista na Figura 14.

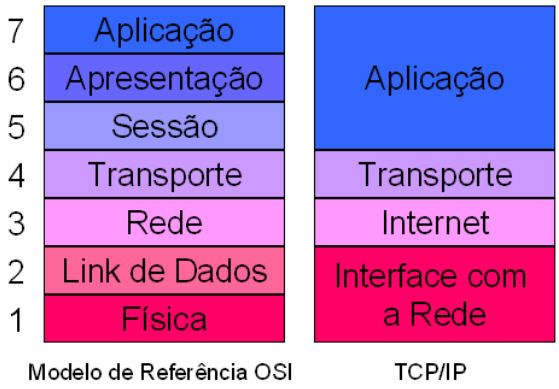

Figura 14: Modelagem de referência do TCP/IP [10]

TCP (Transmission Control Protocol): é um protocolo orientado a conexões totalmente confiável que permite a entrega de um fluxo de pacotes originário de uma determinada máquina de qualquer computador que esteja na internet. Este protocolo fragmenta os pacotes de entrada e manda para a camada interface com a rede (Figura 14). No destino, o processo receptor remonta os pacotes recebidos do fluxo de saída. Este protocolo também controla os fluxos na rede, evitando o congestionamento de pacotes e impedindo que um transmissor rápido sobrecarregue um receptor lento.

2.5.3. **Camada de Redes**

A camada de redes do modelo de referência OSI (Figura 14), se encarrega de fazer toda transição de pacotes desde a origem até seu destino. Durante a transmissão dos pacotes, vários saltos podem ser dados entre roteadores intermediários [5] (Figura 15).

A camada de rede também se encarrega de escolher rotas que evitem a sobrecarga de algumas linhas enquanto outras ficam ociosas [5].

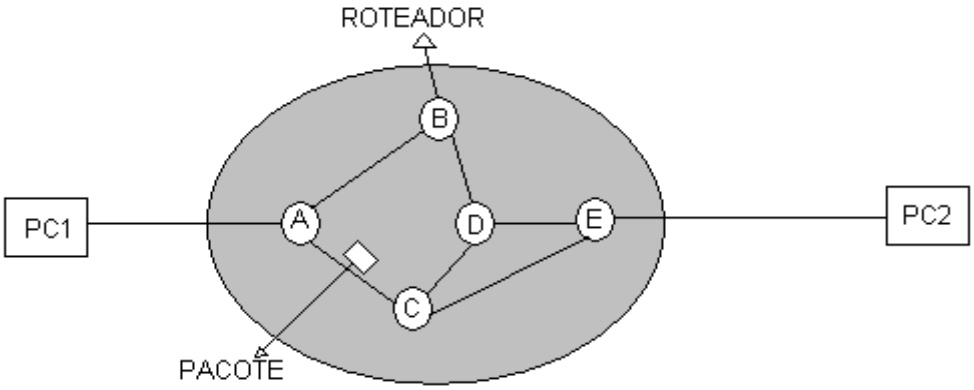

Figura 15: Exemplo de roteadores e transmissão de pacotes na camada de redes [2].

O pacote (Figura 15) pode seguir qualquer rota de acordo com seu algoritmo programado. Neste caso o pacote terá que passar por 4 roteadores até chegar ao seu destino, seja ele qual for a rota. Segundo Tanenbaum, um dos elementos mais importante na camada de rede são os algoritmos que escolhem as rotas e a estrutura de dados que eles utilizaram.

2.5.4. Algoritmos de Roteamento

Os algoritmos de roteamento fazem parte do software da camada de rede, e é responsável por tomar decisões sobre qual caminho será tomado na transmissão do pacote de entrada [2].

Para construção destes algoritmos é preciso levar em consideração as seguintes características:

a) Correção;

b) Simplicidade;

c) Robustez;

d) Estabilidade;

e) Otimização.

Segundo Tanenbaum,

> Em roteadores, pode-se dizer que existe dois processos internos sobre eles. Um deles trata cada pacote que chega, procurando a linha de saída. Esse processo é o encaminhamento. O outro processo é responsável pelo preenchimento e pela atualização das tabelas de roteamento [2].

Alguns algoritmos de roteamento podem ser conferidos abaixo:

i. Roteamento pelo caminho mais curto;
ii. Roteamento com vetor à distância;
iii. Roteamento por estado de enlace;
iv. Roteamento hieráquico;
v. Roteamento por difusão;
vi. Roteamento por multidifusão;
vii. Roteamento para hosts móveis;
viii. Roteamento em redes ad hoc.

2.6. JAVA

Java foi criado por James Goling em 1963. Java teve este nome pois, os programadores da linguagem bebiam muito café no decorrer do projeto, e Java era o nome da cafeteira onde eles faziam seus coffee breaks [1].

2.6.1. Principais características

- Totalmente Orientada a Objetos
- Encapsulada
- Portátil
- Concisa e simples
- Robusta
- Segura
- Concorrente
- Independente de Plataformas
- Compilada

2.6.2. Máquina virtual Java

A JVM (Java Virtual Machine – Maquina Virtual Java) é um programa que carrega e executas os aplicativos Java, transformando-os em bytecode[1]. Os programas feitos em Java rodam em qualquer plataforma desde que esteja instalado a JVM, ficando assim livre de Sistemas operacionais [1].

[1] Bytecodes: é uma codificação do código feito em Java que a maquina entenda. É gerado após a compilação e interpretado na JVM. Como demonstra a Figura16.

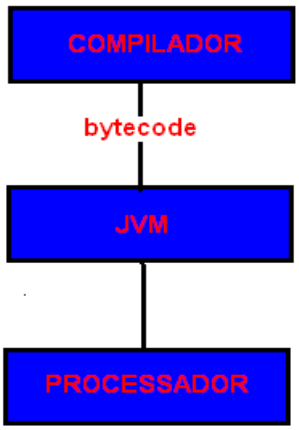

Figura 16: Exemplo de Maquina Virtual Java.

2.6.3. Extensões

- J2SE (Standard Edition)
- J2EE (Enterprise Edition)
- J2ME (Micro-Edition for PDAs and cellular phones)
- JDMK (Java Dynamic Management Kit)
- Jini (a network architecture for the construction of distributed systems)
- JXTA (open source-based peer-to-peer infrastructure)
- JSP (JavaServer Pages)
- JSF (JavaServer Faces)
- JNI (Java Native Interface)
- J3D (A high level API for 3D graphics programming)
- JOGL (A low level API for 3D graphics programming, using OpenGL)
- OSGi (Dynamic Service Management and Remote Maintenance)

2.6.4. Ambientes de desenvolvimento

- Eclipse — um projeto aberto iniciado pela IBM;
- NetBeans — um ambiente criado pela empresa Sun Microsystems;
- JBuilder — um ambiente desenvolvido pela empresa Borland;

2.6.5. Redes com Java

Em diversas linguagens de programação, o desenvolvimento de algoritmos capazes de manipular redes é extremamente complexo, exigindo várias linhas de código.

Com o Java, já é diferente. Todo seu desenvolvimento voltado para Web são códigos nativos da plataforma, ou seja, basta apenas usar algumas instruções e funções que já estão configuradas no Java para manipular grandes ou pequenas redes.

O mecanismo mais comum hoje em dia para possibilitar comunicações entre aplicações é chamado sockets. Java utiliza sockets através do protocolo TCP/IP [9].

2.6.5.1. Sockets TCP/IP em Java

O processo de comunicação em TCP/IP funciona da seguinte forma: o servidor escolhe uma determinada porta e aguarda conexões de entrada nesta porta. O cliente por sua vez solicita conexão a esta porta (Figura 17).

Figura 17: exemplo de requisição de conexão

Se nenhum problema ocorrer, o servidor aceita a conexão e cria um socket em uma porta qualquer do seu lado. Estabelecendo assim uma comunicação entre o cliente e o servidor [9] (Figura 18).

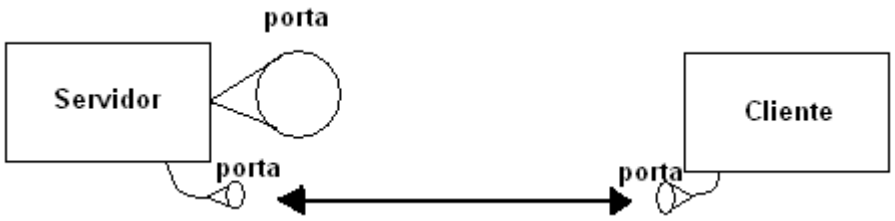

Figura 18: exemplo de conexão estabelecida cliente e servidor.

3. FORMULAÇÃO DO PROBLEMA

Desenvolver um componente feito em Java para esconder mensagens dentro de uma imagem (este método é chamado de “estenografia”) e enviá-las via Web.

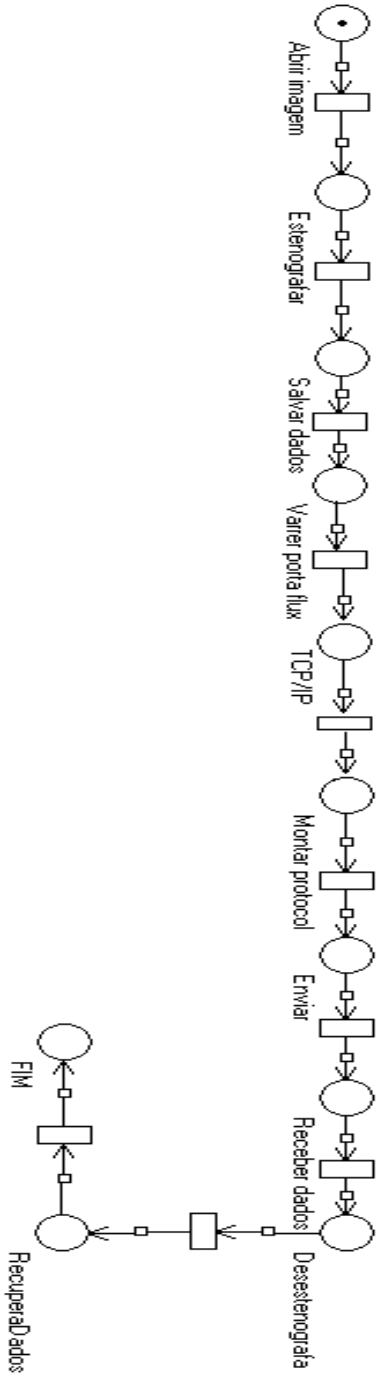

Figura 19: Modelagem do problema

3.1. DESENVOLVIMENTO DO PROJETO

O projeto foi dividido em vários módulos, como mostra a Figura 19 do modelo do problema que são:

3.1.1. Módulo 1: abrir imagem

Uma imagem já capturada por um dispositivo qualquer vai ser aberta pelo software. Esta imagem será do tipo PNG para que se obtenha menos perda de qualidade nas transmissões.

3.1.2. Módulo 2: estenografia

Com a imagem já aberta e a mensagem já digitada será feita a estenografia.

3.1.3. Módulo 3: salvar dados

Após a estenografia ter concluído com sucesso, a imagem será gravada em disco para que se possa enviar ela mais tarde a outro dispositivo via Web.

3.1.4. Módulo 4: varrer portas de fluxos

A conexão é aberta para que se possa enviar a imagem com os dados do paciente.

3.1.5. Módulo 5: organizar cabeçalhos TCP/IP

O protocolo TCP/IP é organizado para que toda a transação seja feita de maneira segura sem perdas de dados.

3.1.6. Módulo 6: enviar

A imagem é enviada para o receptor.

3.1.7. Módulo 7: receber dados

O servidor que estava em espera, agora recebe a imagem.

3.1.8. Módulo 8: decifrar a mensagem estenografada

O método para decifrar a mensagem estenografada é aplicado nesta fase.

3.1.9. Módulo 9: recuperar dados

Com a imagem já decifrada o programa irá recuperar os dados que foram passados via Web, fazendo com que os caracteres que estão dentro da imagem sejam impressos na tela juntamente com a imagem.

4. IMPLEMENTAÇÃO

Para a implementação foi usado a linguagem Java devido à facilidade do mesmo em sala de aula, livros e internet. Fora a liberdade de plataformas que a linguagem oferece.

4.1. INTERFACE

4.1.1. Interface entrada

Figura 20: Interface entrada.

Esta é a primeira tela do software. O usuário terá que escolher cliente ou servidor. Clicando em cliente uma interface se abrirá como demonstra na Figura21. Caso contrário foi clicado em servido outra interface se abrirá que é a tela do Servidor como demonstra a Figura22.

4.1.2. Interface cliente

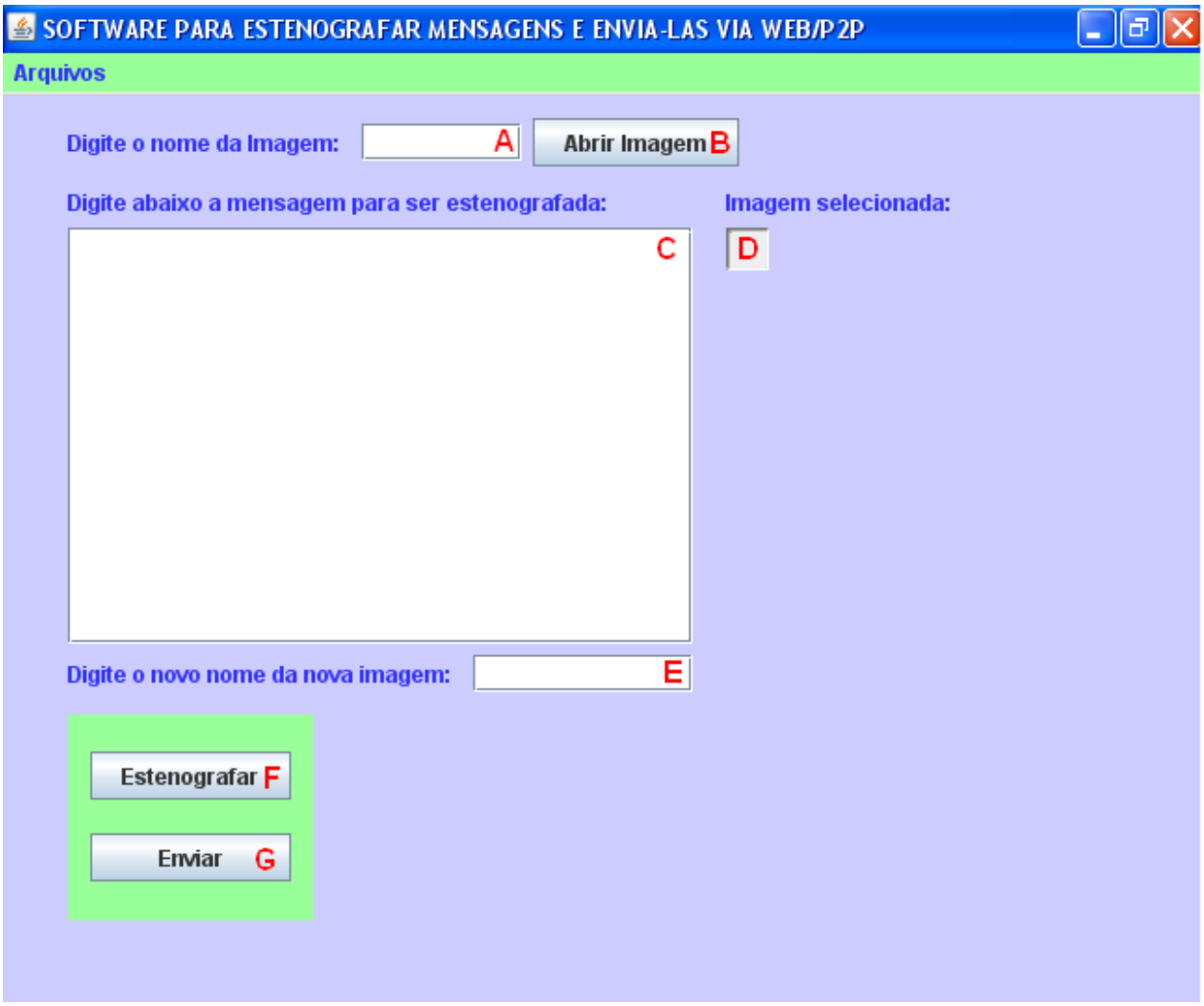

Figura 21: Interface Cliente.

Nesta interface será feita toda parte principal do software. Como demarca as letras na cor vermelha.

A: o nome da imagem será escrito neste local. A imagem terá que ser colocado dentro da pasta onde o programa foi instalado.

B: este botão é para abrir a imagem. O código fonte de como abrir a imagem pode ser conferido a seguir:

```
try{
  //o nome que foi digitado no JTextField1 é setado no metodo setImagemw
  setImagemw(jTextField1.getText());
  // um get é invocado para carregar a imagem para dentro da instancia "img"
  RenderedImage img = JAI.create("fileload",getImagemw());
  int width = img.getWidth();// é definido a largura da imagem
  int height = img.getHeight();// é definido o comprimento da imagem
  //um painel é criado passando a imagem, altura e largura dela
  ScrollingImagePanel panel = new ScrollingImagePanel(img, width, height);
  jScrollPane1.add(panel);// aqui é adicionado o painel já com a imagem no jScrollPane1
  jScrollPane1.show();// exibe a imagem dentro do jScrollPane1

 }catch(Exception e){
 System.out.println(e.getMessage());

 }
```

C: a mensagem será digitada neste JTextArea.

D: a imagem selecionada será aberta neste JScrollPane;

E: aqui será digitado o nome da nova imagem que desejará ser salva, já com a mensagem estenografada nela.

F: botão que ao ser clicado irá gerar a estenografia da mensagem para dentro da imagem selecionada. Segue um exemplo do código fonte de como os caracteres da mensagem foram concatenados com os bits menos significantes da imagem e outro método que retorna estes valores já concatenados:

```
 //metodo que faz a mixagem do valor do pixel do ultimo bit com os valores da mensagem já codificados
 public void mixLSB(int pixel0, int pixel1, int pixel2, int pixel3, int pixel4, int pixel5,
                                                        int pixel6, int pixel7, int pixel8){

        vetPixelvs[0] = ((pixel0 & 254) | decomp_cod[0]);//atribui em vetPixelvs o resultado do valor
        vetPixelvs[1] = ((pixel1 & 254) | decomp_cod[1]);// do pixel&254 | valor decomposto da
        vetPixelvs[2] = ((pixel2 & 254) | decomp_cod[2]);//mensagem. EX:10101010 & 11111110 | 11011010
        vetPixelvs[3] = ((pixel3 & 254) | decomp_cod[3]);
        vetPixelvs[4] = ((pixel4 & 254) | decomp_cod[4]);
        vetPixelvs[5] = ((pixel5 & 254) | decomp_cod[5]);
        vetPixelvs[6] = ((pixel6 & 254) | decomp_cod[6]);
        vetPixelvs[7] = ((pixel7 & 254) | decomp_cod[7]);
     //vetPixelvs[8]= pixel8;// ((pixel8 & 254) | decomp_cod[8]);

}

  public int retValor(int pos){
    return vetPixelvs[pos];

}
```

G: após todas as etapas se concluírem, o botão “Enviar” enviará a imagem para outro dispositivo via Web.

4.1.3. Interface servidor

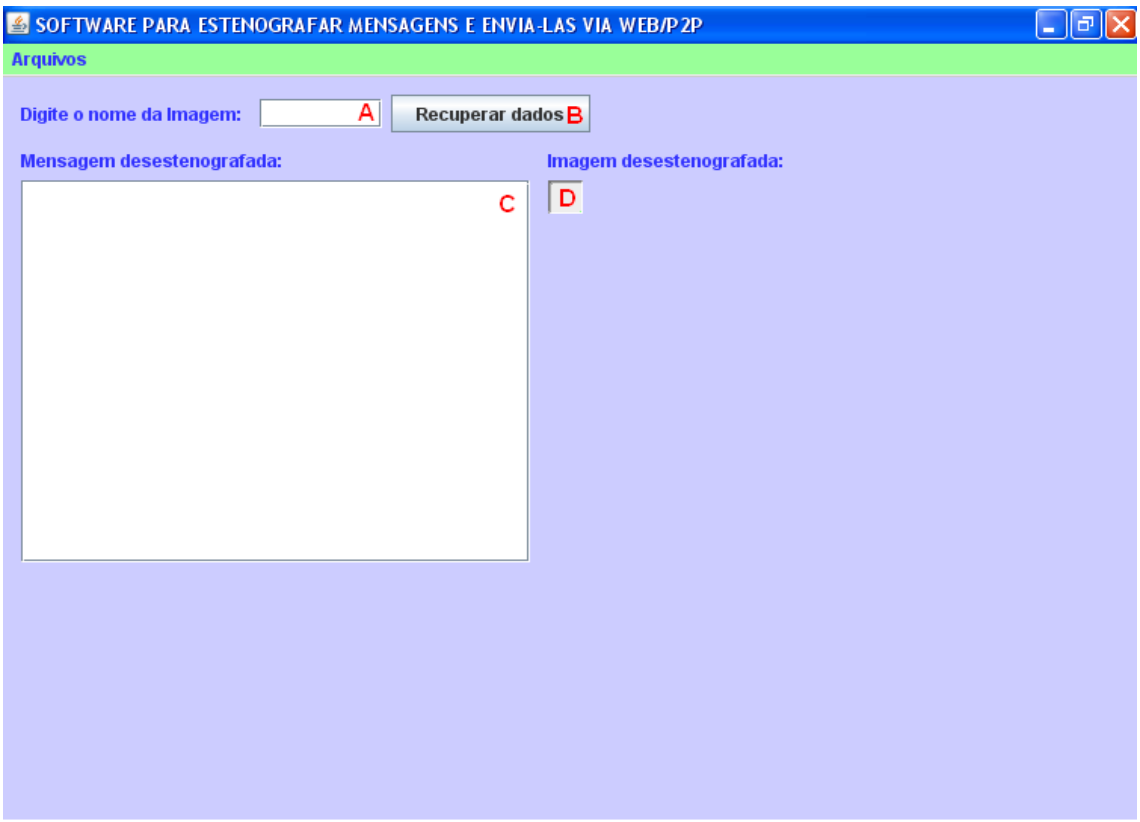

Figura 22: Interface Servidor

Esta interface será a “volta”, ou seja, recuperação dos dados e da imagem que foi enviada pelo Cliente. Que será exibido dentro dos componentes demarcador pelas letras vermelhas.

A: o nome da imagem estenografada será digitado neste local. A imagem estenografada terá que ser colocado dentro da pasta onde o programa foi instalado.

B: este botão é para abrir a imagem estenografada e recuperar a mensagem. O código fonte para fazer esta volta é apresentado abaixo:

```
//metodo que faz a desmixagem do valor do pixel do ultimo bit já com os valores da mensagem
//concatenado a ele
public void deMixSLB(int pixel0, int pixel1, int pixel2, int pixel3, int pixel4, int pixel5,
                                                         int pixel6, int pixel7, int pixel8){

    comp_cod[0]= (byte)pixel0;//o vetor comp_cod na posiçao "x" receberá o pixel "x"
    comp_cod[1]= (byte)pixel1;
    comp_cod[2]= (byte)pixel2;
    comp_cod[3]= (byte)pixel3;
    comp_cod[4]= (byte)pixel4;
    comp_cod[5]= (byte)pixel5;
    comp_cod[6]= (byte)pixel6;
    comp_cod[7]= (byte)pixel7;
    // o vertor sentence na posição "x" rebebe o metodo voltaEightVector().
    sentence[ci]= (int)voltaEightVector();
    ci++;

 }
```

```
//Retorna o valor demixado dos caracteres
public int retValorDemix(int pos){
        return sentence[pos];

    }

//Método que faz a volta dos bits dos caracteres
public byte voltaEightVector(){

    byte value;
    int R=0,G=1,B=2;
    byte[] pixel = comp_cod;
    value = (byte) (pixel[G+6] & 1);
    value = (byte) (value << 1);
    value += (pixel[R+6] & 1);
    value = (byte) (value << 1);
    value += (pixel[B+3] & 1);
    value = (byte) (value << 1);
    value += (pixel[G+3] & 1);
    value = (byte) (value << 1);
    value += (pixel[R+3] & 1);
    value = (byte) (value << 1);
    value += (pixel[B] & 1);
    value = (byte) (value << 1);
    value += (pixel[G] & 1);
    value = (byte) (value << 1);
    value += (pixel[R] & 1);

    return value;

}

}
```

C: jTextArea que irá receber a mensagem;

D: jScrollPane que irá receber a imagem que foi enviada via Web;

4.2. RESULTADOS ALCANÇADOS

Texto que foi usado para estenografar:

“Respostas chegam ao vento mais forte
como um suspiro que não deixa respira
vidros fechados não é proibido
subir ao alto e me deixa voar
olhos abertos para enxergar as mudanças
que vem e deixam me sentir enquanto é bom.”

Original	Antes(zoom)	Depois(zoom)	Depois
			

Observação: Imagens “antes” e “depois” ampliadas.

5. CONCLUSÃO

De acordo com o estudo feito chega-se a conclusão que este projeto será de grande importância para projetos que necessitem estenografar mensagens em uma imagem usando uma ferramenta multi-plataforma. A ferramenta pode ser usada online por causa da escolha da linguagem utilizada. Ainda permite enviar imagens com uma certa segurança pela internet, com mensagens inseridas nas partes da imagem. O foco de estudo abrange várias áreas computacionais, destacando-se a área de conhecimento de redes de computadores, sistemas operacionais, programação em Java, etc.

As maiores dificuldades no projeto foram:

i. Falta de material não encontrado para se estudar a estenografia;

ii. Como mixar os dados da mensagem com a imagem;

iii. Fazer a volta dos dados sem perdas, tanto na mensagem quanto na imagem;

iv. Programação.

5.1. PROJETOS FUTUROS

A manutenção do aspecto visual pode ser mapeada em uma função no cabeçalho da imagem posteriormente visando minimizar a alteração estrutural do que está sendo visto

Fazer a implantação do projeto em algum setor que será de grande utilidade. Algumas implementações:

i. Terroristas;

ii. Governos;

iii. Policias secreta;

iv. Entidades secretas.

6. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Deitel H.M., Deitel P.J., **Java Como Programar**, Porto Alegre, Bookman, 2001. ISBN: 0-13-012507-5.

[2] Tanenbaum A.S, **Redes de Computadores**, Rio de Janeiro, Elsevier, 2003. ISBN: 85-352-1185-3.

[3] Gomes J, Velho L, **Computação Gráfica: Imagem**, Rio de Janeiro, 1994. ISBN: 85-244-0088-9.

[4] Jascone F.L, **Protótipo de software para ocultar texto criptografado em imagens digitais**, Blumenau, 2003.

[5] Tanenbaum A.S, **Redes de Computadores, Rio de Janeiro**, 1994. ISBN: 85-7001-880-0

[6] Rede: http://www.tutorzone.com.br/index.php?ind=reviews&op=entry_view&iden=2071 acessado em outubro de 2009

[7] Processamento de Imagem: http://www.cbpf.br/cat/pdsi/pdf/ProcessamentoImagens.PDF acessado em novembro de 2009.

[8] Extensões de Imagens: http://www.colivre.coop.br/CursoGIMP/ConceitosB%E1sicosI acessado em novembro de 2009

[9] Socket em Java: http://www.sumersoft.com/publicacoes/SocketsEmJAVA.pdf acessado em novembro de 2009.

[10] Figura 14: TCP/IP: http://www.clubedohardware.com.br/artigos/1351 acessado em junho de 2009.