HECOBI: HERRAMIENTA SOFTWARE ESTEGANOGRÁFICA PARA OCULTAR INFORMACIÓN UTILIZANDO LOS MÉTODOS DE LA TRANSFORMADA DISCRETA COSENO (DCT) Y EL BIT MENOS SIGNIFICATIVO (LSB) EN ARCHIVOS DE IMÁGENES CON FORMATO BMP DE 24 BITS

VICTOR HUMBERTO MORENO CADENA
FABIAN FAJARDO VARGAS

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2006

HECOBI: HERRAMIENTA SOFTWARE ESTEGANOGRÁFICA PARA OCULTAR INFORMACIÓN UTILIZANDO LOS MÉTODOS DE LA TRANSFORMADA DISCRETA COSENO (DCT) Y EL BIT MENOS SIGNIFICATIVO (LSB) EN ARCHIVOS DE IMÁGENES CON FORMATO BMP DE 24 BITS

VICTOR HUMBERTO MORENO CADENA
FABIAN FAJARDO VARGAS

Trabajo de grado para optar el título de
Ingeniero de Sistemas

Director
Edilberto José Reyes González

Codirector
Juan Gabriel Quintero Peña

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2006

DEDICATORIA

“Clama a mí y yo te responderé, y te anunciaré cosas grandes y misteriosas que tú desconocías”. **Jeremías 33:3**

A Dios, por permitirme lograr este sueño tan esperado.

A mis padres Miguel Angel y María Raquel, por su amor incondicional.

A mi hermana Nayibe por brindarme su apoyo y creer siempre en mí.

A mis queridos tíos, a mí sobrinito German Yesid y a toda mi familia.

A la tierra de mis amores, Barrancabermeja.

Fabian

DEDICATORIA

A Dios, por permitirme alcanzar mis metas.

A mis padres Humberto y Mairan, por su apoyo incondicional.

A mi hermano Leonardo por sus consejos, su apoyo y por creer siempre en mí.

A la mujer de mi inspiración.

A toda mi familia.

A la tierra de mis amores, Barrancabermeja.

Victor

AGRADECIMIENTOS

A nuestro Director de Proyecto, M.Sc. Edilberto Reyes Gonzaléz, por su valiosa colaboración y apoyo en el proyecto.

A nuestro Codirector, M. Sc.(c) Juan Gabriel Quintero Peña por su sentido altruista, por compartir sus conocimientos y entrega en esta labor investigativa.

A nuestros compañeros, que siempre creyeron en nosotros.

A la Universidad Industrial de Santander.

CONTENIDO

LISTA DE FIGURAS

pág.

LISTA DE TABLAS

pág.

TITULO:

"HECOBI" HERRAMIENTA SOFTWARE ESTEGANOGRÁFICA PARA OCULTAR INFORMACIÓN UTILIZANDO LOS MÉTODOS DE LA TRANSFORMADA DISCRETA COSENO (DCT) Y EL BIT MENOS SIGNIFICATIVO (LSB) EN ARCHIVOS DE IMÁGENES CON FORMATO BMP DE 24 BITS [*]

AUTORES:

MORENO CADENA VICTOR HUMBERTO, FAJARDO VARGAS FABIAN **

PALABRAS CLAVES:

BMP, ESTEGANOGRAFIA, ESTEGO IMAGEN, BIT MENOS SIGNIFICATIVO, TRANSFORMADA DISCRETA COSENO, CRIPTOGRAFIA, AES, MD5.

DESCRIPCIÓN:

HECOBI es un sistema esteganográfico, que pertenece a la rama de la seguridad informática, diseñado para ocultar información en imágenes con formato BMP de 24 bits. Con el sistema HECOBI se puede ocultar un mensaje de texto o cualquier tipo de archivo, siempre y cuando el tamaño de la imagen portadora lo permita. También es posible realizar esteganografía en imágenes JPEG, GIF ó PNG mediante el uso del Conversor de Imágenes que incluye HECOBI.

Ocultar información es posible gracias a la implementación de los métodos esteganográficos *Least Significant Bit (LSB)* y *Discrete Cosine Transform (DCT)*. Usando *LSB* los datos son ocultados en el bit menos significativo de cada byte de la imagen, mientras con el método *DCT* la imagen es transformada al dominio de los coeficientes DCT en donde la información secreta es almacenada. Además con el método *Second Least Significant Bit (2LSB)*, los datos se almacenan en los dos bits menos significativos de cada byte de la imagen

HECOBI posee un sistema de seguridad criptográfico, mediante la utilización de los algoritmos *Advanced Encripton Standard* (AES) y *Message Digest 5 (MD5),* lo que permite obtener confidencialidad (el sistema será accesible sólo por usuarios autorizados) e integridad (los componentes del sistema sólo pueden ser creados y modificados por usuarios autorizados) de los datos.

El libro contiene la descripción del proceso que se lleva a cabo para la elaboración de HECOBI. En el capítulo 1 se realiza la presentación del proyecto. En el capítulo 2 se encuentran los fundamentos teóricos y el estado del arte del proyecto. En el capítulo 3 se explica con base a la metodología utilizada, el proceso de desarrollo de HECOBI. En el capítulo 4 se encuentra el modelo esteganográfico empleado. En los capítulos posteriores se encuentran las conclusiones, recomendaciones, la bibliografía y los anexos.

---

[*] Trabajo de Grado.

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Universidad Industrial de Santander. Directores: Magíster Edilberto Reyes González, Ingeniero Juan Gabriel Quintero.

TITLE:

"HECOBI" SOFTWARE STEGANOGRAPHIC TOOL FOR HIDE INFORMATION, WITH THE METHODS DISCRETE COSINE TRANSFORM DCT AND LEAST SIGNIFICANT BIT LSB IN DIGITAL BMP IMAGES OF 24 BITS *

AUTORS:

MORENO CADENA VICTOR HUMBERTO, FAJARDO VARGAS FABIAN **

KEY WORDS:

STEGANOGRAPHY, CRIPTOGRAPHY, LEAST SIGNIFICANT BIT, BITMAP, DISCRETE COSINE TRANSFORM, ALGORITHM AES, MD5.

ABSTRACT:

HECOBI is steganographic system, this belongs to the area of the security in computer science. HECOBI hides the information in digital BMP images of 24 bits. It can hide a text message or any file, it depends of the image's size. Too, it can do steganography in JPEG, GIF or PNG images, using the images converter of HECOBI.

The steganographic methods use for to hide information are *Least Significant Bit (LSB)* and *Discrete Cosine Transform* (DCT). With the LSB method, the information can be hidden in the least significant bit of each byte of the image. Using DCT, the imagen is transformed at domain of DCT components and data are embedded here. Also, the 2LSB method, the information can be hidden in the two least significant bits of each byte of the image.

HECOBI has a cryptographic system, it use two algorithm: *Advanced Encripton Standard* (AES) y *Message Digest 5 (MD5).* The first permit get a confidential system; the second permit check up the integrity of the dates(the information only can be created and modified by autorized user).

The book contains the description of the process that was used in order to create the HECOBI software. Chapter first presents the project, chapter second exposes the sumary of the required theoretical basis of the development of the project. Chapter third explains the development process according to the metodology used. Chapter fourth shows the steganographic model used; and the next chapter contains conclusions, suggestions, reference and anexes.

---

* Word of Degree.

** Faculty of Physical-Mechanical Engineerings. Department of Systems Engineering. Universidad Industrial de Santander. Tutors: Edilberto Reyes González, Juan Gabriel Quintero.

INTRODUCCIÓN

Desde la antigüedad, el hombre ha sentido la necesidad de comunicarse con los demás por medio de gestos, señales, sonidos y signos, teniendo la posibilidad de interactuar con sus semejantes. A través de esa interacción descubrió la importancia de mantener cierta confidencialidad en la información importante que manejaba. Para este fin, a lo largo de la historia han existido multitud de métodos para ocultar información, quizás los más conocidos hayan sido la tinta invisible, muy utilizada durante la Segunda Guerra Mundial, o las marcas de cualquier tipo sobre ciertos caracteres (desde pequeños pinchazos de alfiler hasta trazos a lápiz que marcan un mensaje oculto en un texto), pero otros mecanismos más extravagantes también han sido utilizados: por ejemplo, afeitar la cabeza de un mensajero y tatuar en el cuero cabelludo el mensaje, dejando después que el crecimiento del pelo lo oculte.

Estos métodos hacen parte de la llamada esteganografía que permite ocultar un mensaje dentro de otro, de forma que terceras personas no tengan acceso a dicha información inmersa. Esta ciencia ha tomado auge debido al crecimiento acelerado que han tenido las telecomunicaciones en las últimas décadas, principalmente en la cantidad de usuarios que pueden tener acceso a Internet y

que incrementa las posibilidades de poner en riesgo tanto la información alojada en un PC, como la que se envía por este medio. Es aquí donde la esteganografía entra a jugar un papel importante: ofrecer una alternativa de seguridad plausible. El mecanismo esteganográfico más extendido está basado en las imágenes digitales y su excelente capacidad para ocultar información; dado que casi todos los estándares gráficos tienen una graduación de colores mayor de lo que el ojo humano puede apreciar, la imagen no cambiará su apariencia de forma notable.

A pesar de ser una buena alternativa de seguridad, se corre el riesgo que la información oculta pueda ser descubierta y/o alterada por terceras personas que logren rastrear el mensaje enviado.

El proyecto tiene como fin fundamental el estudio y la aplicación del concepto matemático de la Transformada Discreta Coseno (DCT) y el método LSB, para el desarrollo de una herramienta esteganográfica que permita ocultar información en imágenes, proponiendo la implementación de estos métodos y la optimización en cuanto a seguridad, utilizando el concepto criptográfico de firma digital para brindar la integridad de la información oculta. Con la combinación de esteganografía y criptografía, se logrará mejorar algunas de las características de los modelos esteganográficos actuales.

1. PRESENTACIÓN DEL PROYECTO

1.1 PRESENTACIÓN DEL INFORME

El contenido de este documento se encuentra dividido en capítulos, éstos contienen la fundamentación teórica para el desarrollo de la herramienta esteganográfica para ocultar información en imágenes BMP de 24 bits, para la escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.

Capitulo 1: PRESENTACIÓN. Presenta los Objetivos, antecedentes y la justificación del proyecto.

Capítulo 2: MARCO TEÓRICO. Presenta la fundamentación teórica de los métodos para ocultar información en imágenes, criptografía y la estructura del formato de archivo bitmap BMP utilizado en el desarrollo del proyecto.

Capítulo 3: DISEÑO DEL SOFTWARE. Presenta el diseño y la metodología empleada para el desarrollo del sistema software.

Capítulo 4: MODELO ESTEGANOGRÁFICO EMPLEADO. Presenta la propuesta del Modelo Esteganográfico utilizado en el Trabajo de Grado y los algoritmos desarrollados.

El informe finaliza con las conclusiones y recomendaciones de los autores acerca del proyecto.

1.2 DESCRIPCIÓN DEL PROYECTO

1.2.1 Objetivo General. Desarrollar una herramienta[1] software de tipo esteganográfico que utilice los métodos de la Transformada Discreta Coseno (DCT) y el Bit Menos Significativo (LSB) en archivos de imágenes en formato BMP de 24 bits.

[1] La herramienta final de ahora en adelante la identificaremos con el nombre de HECOBI.

1.2.2 Objetivos Específicos.

- ✓ Identificar las características principales de la Transformada Discreta Coseno (DCT) aplicada al ocultamiento de información en imágenes.

- ✓ Identificar las características principales del método del Bit Menos Significativo (LSB) aplicado al ocultamiento de información en imágenes.

- ✓ Implementar una herramienta software esteganográfica que permita:

  - Convertir imágenes con formato JPEG, JPG, GIF, PNG y BMP en formato BMP de 24 bits.

  - Ocultar información que provenga de un mensaje de texto o un archivo de cualquier tipo en un archivo de imagen BMP de 24 bits por medio de los métodos DCT y LSB.

  - Garantizar la integridad de la información ocultada por medio de la implementación de la función hash MD5 (Message Digest 5) y el algoritmo criptográfico AES (Advanced Encryption Standard).

- ✓ Realizar una documentación sobre la Transformada Discreta Coseno (DCT), el

método del Bit Menos Significativo (LSB) y el formato de imagen BMP.

1.3 JUSTIFICACIÓN

Desde su aparición, los computadores han traído consigo grandes aportes para las organizaciones, la comunidad científica y la comunidad en general. Entre muchos de estos avances, se encuentra Internet, que ha permitido el intercambio de información entre computadores ubicados alrededor del mundo. Con estos avances se comenzó a tratar un tema clave en nuestros días: la seguridad informática.

A raíz de la necesidad de conservar la información de una forma segura en cualquier medio de almacenamiento e incluso a través de las comunicaciones entre computadores, surgió la criptografía, que permite ocultar el contenido de la información, establecer su autenticidad, evitar su modificación no detectada, evitar su repudio y/o evitar su uso no autorizado.

A través de Internet, la información circula por puntos intermedios; estos intermediarios tienen técnicamente la posibilidad de ver lo que se está enviando, lo

que puede plantear un problema de seguridad en el caso de que terceras personas interceptasen dicha información. Cuando se utilizan métodos criptográficos, los terceros pueden ver la información pero no entenderla, ya que circula encriptada. Por esta razón, hoy en día, la mayoría de sistemas informáticos disponibles en el mercado, cuentan con algún tipo de seguridad implementado mediante el uso de la criptografía. Esto permite a los usuarios mantener su información con un grado de seguridad que es proporcionado según el sistema que emplee.

Existe una ciencia llamada esteganografía, que proporciona mecanismos para ocultar información en archivos con formato multimedia (imágenes, video y audio), por medio del uso de ciertas técnicas esteganográficas. Mención especial merece el uso de la esteganografía para exportar información sin violar las leyes restrictivas que, con respecto a la criptografía, existen en algunos países. El destinatario empleará técnicas para separar la información útil del resto. En consecuencia, se tiene un mecanismo para transmitir información, pero que sólo puede ser reconstruida por el destinatario, con lo que en realidad se ha logrado protegerla. Este sistema surgió como desafío a la política restrictiva del Gobierno de los Estados Unidos con respecto a la criptografía.

Las aplicaciones esteganográficas que existen, carecen del uso de la criptografía

para brindar mayor seguridad e integridad a la información oculta, permitiendo de una u otra manera ponerla al descubierto. Esto conlleva al decrecimiento del uso de la esteganografía como una alternativa al problema de seguridad informática. Con una herramienta como **HECOBI**, diseñada especialmente para ocultar información en imágenes BMP de 24 bits se tendrán las siguientes ventajas:

- ✓ Su entorno gráfico permite que el usuario se familiarice rápidamente con la herramienta.

- ✓ Permitirá la conversión de imágenes con formatos JPEG, JPG, GIF, PNG y BMP al formato BMP de 24 bits, haciéndola más eficiente al permitir al usuario escoger como base los formatos de archivos de imágenes más utilizados actualmente.

- ✓ La capacidad de almacenamiento de información proveída por el formato BMP de 24 bits es superior a cualquier otro formato, posibilitando el aumento de información inmerso en una imagen.

- ✓ El usuario tendrá la facilidad de escoger entre los métodos LSB y DCT para el ocultamiento de la información.

- ✓ Permite ocultar todo tipo de información proveniente de un archivo de

cualquier formato o un mensaje de texto introducido por el usuario.

- ✓ Posibilita al diseñador de imágenes incluir dentro de ellas, los datos acerca de su autoría.

- ✓ Proporciona seguridad e integridad a la información oculta, empleando una contraseña de seguridad, el encriptamiento de la información por medio del algoritmo AES y una firma digital con el uso de la función MD5.

- ✓ Ofrece un alto grado de robustez en comparación con otras herramientas similares, por la combinación de conceptos y técnicas esteganográficas y criptográficas.

- ✓ Es escalable, ya que al ser concebida dentro del paradigma orientado a objetos, ofrece una fácil integración de nuevos objetos, procedimientos y datos.

El software se complementará con una investigación sobre los temas que aborda, dando como resultado la documentación correspondiente en el libro a entregar y/o en la ayuda de la herramienta.

Es necesario abrir una ventana a ese inmenso mundo de la seguridad informática y proporcionar una base para futuras investigaciones en el área, específicamente en la esteganografía. Es por esta razón que se han estudiado, buscado y analizado los procedimientos que son llevados a cabo y los algoritmos criptográficos que son empleados, usando la Transformada Discreta Coseno y el Bit Menos Significativo, desarrollando **HECOBI**, para que el estudiante e investigador de nuevas tecnologías pueda entender con claridad su fundamento, dándole soporte para iniciar el estudio de la esteganografía, pudiendo aportar en la mejora o diseño de nuevos métodos.

1.3.1 Impacto. En la tabla 1, se diferencia el impacto según sea académico, técnico, económico o social.

Tabla 1. Impacto técnico, económico y social del proyecto.

Académico	A nivel de los autores, el desarrollo de habilidades, en el área de la esteganografía y criptografía y a nivel de la Escuela de Ingeniería de Sistemas e Informática de la UIS, fortalecimiento de la investigación en el área de desarrollo de software en la seguridad informática, lo que aportará conocimientos para el desarrollo de futuros proyectos y un gran reconocimiento de la EISI[2].

[2] Escuela de Ingeniería de Sistemas e Informática. Universidad Industrial de Santander.

Técnico	El proyecto permitirá el desarrollo de nuevas herramientas software en el ámbito de Seguridad Informática.
Económico	La puesta en marcha del proyecto traerá grandes beneficios al interior de la EISI, porque su uso se traducirá en una herramienta esteganográfica de fácil adquisición sin costo alguno para la comunidad UIS.
Social	El desarrollo de este proyecto permitirá un acercamiento de la comunidad universitaria a temas relacionados con la Seguridad Informática, especialmente a la esteganografía en Imágenes digitales, ya que no se conoce en Colombia algún software de ocultamiento de información en imágenes. El proyecto **HECOBI** se fundamenta en el mejoramiento de la imagen y la reivindicación de la Escuela de Ingeniería de Sistemas como productora de Software, consolidándose cada día como una institución capaz de afrontar compromisos de desarrollo frente a su comunidad y a si misma. Mejorará la imagen institucional y aportará al cumplimiento de la misión de la Universidad que propende por la "participación activa liderando procesos de cambio por el progreso."[3]

[3] Misión de la Universidad Industrial de Santander, primer párrafo.

1.3.2 Viabilidad. En la tabla 2, se diferencia la viabilidad según sea técnica, económica o social.

Tabla 2. Viabilidad técnica, económica y social del proyecto.

Técnica	La Universidad cuenta con las licencias software necesarias (Delphi Enterpise), los equipos requeridos para el desarrollo y el personal capacitado para asesorar el proyecto. El recurso humano requerido cuenta con el conocimiento, disposición y tiempo necesarios. La accesibilidad a lo recursos documentales es la deseada pues se cuenta con material bibliográfico y selecta información en Internet. La complejidad del problema a resolver es adecuada para un proyecto de pregrado.
Económica	La Universidad Industrial de Santander y los autores están en condiciones de asumir los costos de esta investigación, puesto que es un costo razonable. Es posible obtener los resultados esperados con los recursos solicitados en la sección de presupuesto.
Social	La comunidad a la que está dirigida este proyecto se muestra interesada en adquirir y utilizar el software.

2. MARCO TEÓRICO

2.1 CRIPTOGRAFÍA

2.1.1 Concepto. La palabra criptografía proviene del griego *kryptos*, que significa esconder y *gráphein*, escribir, es decir, escritura escondida. La criptografía es la rama de las Matemáticas y en la actualidad de la Informática y la Telemática, que hace uso de métodos y técnicas con el objeto principal de cifrar y/o proteger un mensaje o archivo por medio de un algoritmo, usando una o más claves. Esto da lugar a diferentes tipos de sistemas de cifra, llamados criptosistemas, que permiten asegurar alguno de estos tres aspectos básicos de la seguridad informática: *la confidencialidad, la integridad y el no repudio de emisor y no repudio de receptor*.

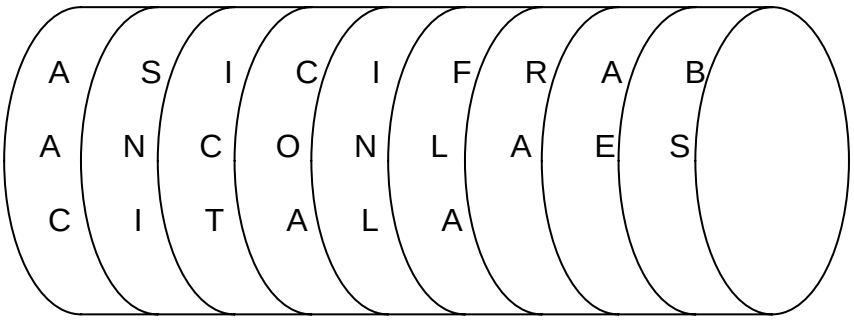
La criptografía ha sido usada a través de los años para el envío de mensajes confidenciales cuyo propósito es que sólo las personas autorizadas puedan entender el mensaje. La palabra Criptografía sólo se refiere al uso de códigos, por lo que no engloba a las técnicas que se usan para romper dichos códigos

(Criptoanálisis). El término Criptología[4], aunque no está recogido aún en el Diccionario, se emplea para agrupar estas dos disciplinas.

2.1.2 Historia.

- **La escítala.** Ya en siglo V A.C. los lacedemonios, un antiguo pueblo griego, usaban el método de la escítala para cifrar sus mensajes. El sistema consistía en una cinta que se enrollaba en un bastón y sobre el cual se escribía el mensaje en forma longitudinal como se muestra en la Figura 1.

Figura 1. Cifrado mediante sistema de escítala.



[4] Ciencia que estudia e investiga todo aquello relacionado con la criptografía: incluye cifra y criptoanálisis.

Una vez escrito el mensaje, la cinta se desenrollaba y era entregada al mensajero; si éste era interceptado por cualquier enemigo, lo único que se conseguía era un conjunto de caracteres o letras distribuidas al parecer de forma aleatoria en dicha cinta. Incluso si el enemigo intentaba enrollar la cinta en un bastón con diámetro diferente, el resultado obtenido era un conjunto de letras escritas una a continuación de otra sin sentido alguno. Por ejemplo, en el caso de la figura 1, la cinta llevará el mensaje M = ASI CIFRABAN CON LA ESCITALA si bien en ella sólo podrá leerse el criptograma C = AACSNIICTCOAINLFLARAAEBS. Para enmascarar completamente la escritura, es obvio que la cinta en cuestión debe tener caracteres en todo su contorno. La clave del sistema residía precisamente en el diámetro de aquel bastón, de forma que solamente el receptor autorizado tenía una copia exacta del mismo bastón en el que enrollaba el mensaje recibido y, por tanto, podía leer el texto en claro. En este sistema no existe modificación alguna del mensaje; es decir, éste va en claro desde el transmisor hacia el receptor.

De esta forma se lograba el objetivo de la confidencialidad, en tanto que la integridad estaba en entredicho y dependía de lo aguerrido y fiel que fuese nuestro mensajero. Si la cinta era robada y se cambiaban los caracteres, podría llegar al receptor un mensaje sin sentido y, lo que es peor, con un duplicado del bastón original podía enviarse un mensaje con sentido completamente distinto al encomendado al mensajero.

- **El cifrador del César.** En el siglo I A.C., aparece un cifrador básico conocido con el nombre genérico de cifrador del César en honor al emperador Julio César y en el que ya se aplica una transformación al texto en claro de tipo monoalfabética. El cifrador del César aplica un desplazamiento constante de tres caracteres al texto en claro, de forma que el alfabeto de cifrado es el mismo que el alfabeto del texto en claro pero desplazado 3 espacios hacia la derecha módulo n, con n el número de letras del mismo. En la Figura 2 se muestra el alfabeto y por tanto la transformación que utiliza este cifrador por sustitución de caracteres para el alfabeto castellano de 27 letras.

Figura 2. Alfabeto de cifrado del César para lenguaje castellano.

Mi	A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z
Ci	D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z A B C

2.1.3 Criptosistema. Se define como la quíntupla $(M;C;K;E;D)$ donde:

- ✓ M representa el conjunto de todos los mensajes sin cifrar (texto plano) que pueden ser enviados.

- ✓ C representa el conjunto de todos los posibles mensajes cifrados, o

criptogramas.

- ✓ K representa el conjunto de claves que se pueden emplear en el Criptosistema.

- ✓ E es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de M para obtener un elemento de C. Existe una transformación diferente Ek para cada valor posible de la clave k.

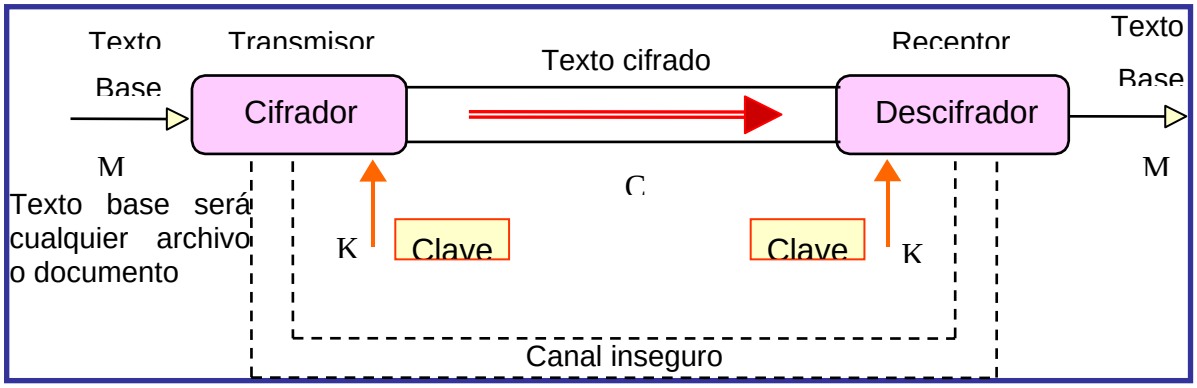
- ✓ D es el conjunto de transformaciones de descifrado, análogo a E.

Todo Criptosistema debe cumplir la condición:

$$D_k\left(E_k\left(m\right)\right) = m$$

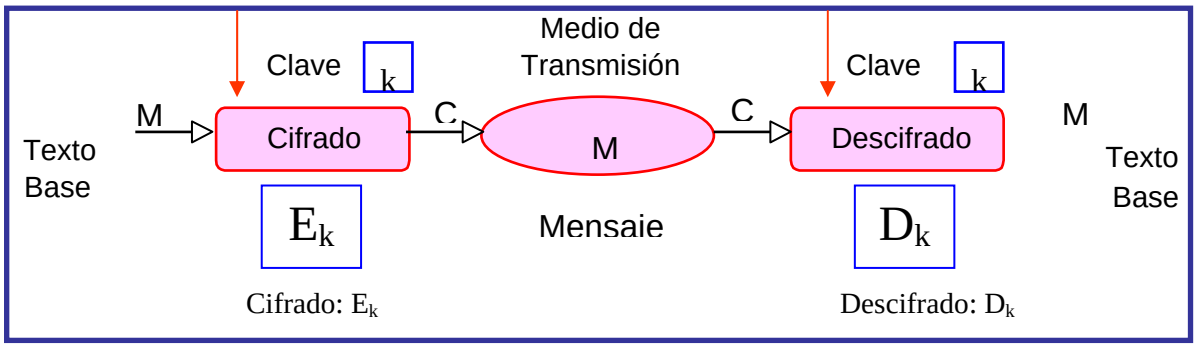
Es decir, que si se toma un mensaje m, se cifra empleando la clave K y luego se descifra empleando la misma clave, se obtiene de nuevo el mensaje original M.

Figura 3. Esquema de un Criptosistema.



2.1.4 Criptosistemas simétricos o de clave privada. Son aquellos que emplean la misma clave *K* tanto para cifrar como para descifrar. Presentan el inconveniente de que para ser empleados en comunicaciones la clave *k* debe estar tanto en el emisor como en el receptor, lo cual requiere la transmisión de la clave de forma segura.

Figura 4. Esquema de un Criptosistema de clave privada.



2.1.5 Algoritmo AES (Advanced Encryption Standard). En Octubre de 2000 el NIST (National Institute of Standards and Technology), Instituto Nacional de Estándares y Tecnología) de Estados Unidos anunció oficialmente la adopción del algoritmo Rijndael (acrónimo formado por los nombres de sus dos autores, los belgas Joan Daemen y Vincent Rijnmen) como AES (Advanced Encryption Standard, Estándar Avanzado de Encripción) para su empleo en aplicaciones criptográficas no militares, encaminado a proporcionar a la comunidad internacional un nuevo algoritmo de cifrado potente, eficiente y fácil de implementar.

AES es un sistema de cifrado por bloques, diseñado para manejar longitudes de bloque de 128 bits y de clave variable, de 128, 192 y 256 bits. Además, AES tiene 10, 12 o 14 vueltas respectivamente, cada vuelta consiste en la aplicación de una ronda estándar, que consiste de 4 transformaciones básicas, la última ronda es especial y consiste de 3 operaciones básicas, añadiendo siempre una ronda inicial. Por otro lado, existe el programa de claves o extensión de la clave, que provee resistencia a ataques conocidos.

AES es uno de los algoritmos más seguros en la actualidad, debido a la estructura de su diseño, que busca eliminar la simetría en las subclaves. Por esto, ha sido adoptado poco a poco desde los protocolos más usados, hasta las aplicaciones

más especializadas.

2.1.5.1 Estructura de AES. Se ha definido cada ronda como una composición de cuatro funciones invertibles diferentes formando tres capas. Cada una de las funciones tiene un propósito preciso:

- **La capa de mezcla lineal.** Permite obtener un alto nivel de difusión a lo largo de varias rondas.

- **La capa no lineal.** Consiste en la aplicación paralela de s-cajas con propiedades óptimas de no linealidad.

- **La capa de adición de clave.** Es un simple or-exclusivo entre el estado intermedio y la subclave correspondiente a cada ronda.

2.1.6 Funciones Hash (Resumen). De manera matemática se puede definir una función resumen como proyecciones de un conjunto, generalmente con un número elevado de elementos (incluso infinitos), sobre un conjunto de tamaño fijo y mucho más pequeño que el anterior.

El resultado de aplicar una función hash tiene las siguientes características:

- ✓ Todos los números resumen generados con un mismo método tienen el mismo tamaño sea cual sea el texto utilizado como base.

- ✓ Dado un texto base, es fácil y rápido (para un computador) calcular su número resumen.

- ✓ Es imposible reconstruir el texto base a partir del número resumen.

- ✓ Es imposible que dos textos base diferentes tengan el mismo número resumen.

Hay muchos algoritmos de este tipo; uno de los más conocidos es MD5, que se utiliza habitualmente para firmas digitales.

2.1.6.1 Algoritmos criptográficos de resumen.

- **MD5**. Creado por Ron Rivest 1991. Mejoras al MD4 y MD2 (1990), es más lento pero con mayor nivel de seguridad. Resumen de 128 bits.
- **SHA-1.** Del NIST, National Institute of Standards and Technology, 1994. Similar a MD5 pero con resumen de 160 bits. Existen otras nuevas propuestas conocidas como SHA-256 y SHA-512.

- **RIPEMD.** Creado por la Comunidad Europea, RACE, en 1992. Resumen de 160 bits.

- **Snefru.** Creado por Ralph Merkle, en 1990. Resúmenes entre 128 y 256 bits. Ha sido criptoanalizado y es lento.

2.1.7 MD5 (Message Digest 5). MD5 es uno de los algoritmos de reducción criptográficos diseñados por el profesor Ronald Rivest del MIT (Massachussets Institute of Technology, Instituto Tecnológico de Masachusets) en 1991. El algoritmo MD5 procesa los mensajes de entrada en bloques de 512 bits y genera una salida de 128 bits de longitud.

Los resúmenes MD5 se utilizan extensamente en el mundo del software para proporcionar la seguridad de que un archivo descargado de Internet no se ha alterado. Comparando una suma MD5 publicada con la suma de comprobación del archivo descargado, un usuario puede tener la confianza suficiente de que el archivo es igual que el publicado por los desarrolladores. Esto protege al usuario contra los virus que algún otro usuario malicioso pudiera incluir en el software. La comprobación de un archivo descargado contra su suma MD5 no detecta solamente los archivos alterados de una manera maliciosa, también reconoce una

descarga corrupta o incompleta.

En sistemas Linux se utiliza el algoritmo MD5 para encriptar las claves de los usuarios. En el disco se guarda el resultado del MD5 de la clave que se introduce al dar de alta un usuario, y cuando éste quiere entrar en el sistema se compara la entrada con la que hay guardada en el disco duro, si coinciden, es la misma clave y el usuario será autenticado.

2.1.7.1 Algoritmo básico de Message Digest 5.

a) Un mensaje M se convierte en un bloque múltiplo de 512 bits, añadiendo bits si es necesario al final del mismo.

b) Con los 128 bits de cuatro vectores iniciales ABCD de 32 bits cada uno y el primer bloque del mensaje de 512 bits, se realizan diversas operaciones lógicas entre ambos bloques.

c) La salida de esta operación (128 bits) se convierte en el nuevo conjunto de 4 vectores A'B'C'D' y se realiza la misma función con el segundo bloque de 512 bits del mensaje, y así hasta el último bloque del mensaje.

d) Al terminar, el algoritmo entrega un resumen que corresponde a los últimos 128 bits de estas operaciones.

2.1.8 Firma Digital. Una *firma digital* es una secuencia de bits que se añade a una pieza de información cualquiera, y que permite garantizar su autenticidad de forma independiente del proceso de transmisión, tantas veces como se desee. Presenta una analogía directa con la firma manuscrita, y para que sea equiparable a esta última debe cumplir las siguientes propiedades:

- ✓ Va ligada indisolublemente al mensaje. Una firma digital válida para un documento no puede ser válida para otro distinto.

- ✓ Sólo puede ser generada por su legítimo titular. Al igual que cada persona tiene una forma diferente de escribir, y que la escritura de dos personas diferentes puede ser distinguida mediante análisis grafológicos, una firma digital sólo puede ser construida por la persona o personas a quienes legalmente corresponde.

- ✓ Es públicamente verificable. Cualquiera puede comprobar su autenticidad en cualquier momento, de forma sencilla.

2.2 FUNDAMENTOS DE IMÁGENES DIGITALES

2.2.1 Píxel. Es el elemento básico de una imagen; un píxel corresponde a un punto de la imagen en la pantalla. El píxel es una unidad de información, no una unidad de medida, ya que no se corresponde con un tamaño concreto. Un píxel puede ser muy pequeño 0.1 mm. o muy grande 1 cm.

2.2.2 Imagen. Para un computador una imagen es un arreglo bidimensional de píxeles que representan la intensidad de luz en varios puntos.

2.2.3 Color. Un color puede definirse como la combinación de tres colores básicos: rojo, verde y azul (RGB). R, G, B representan las intensidades de cada uno de los colores básicos. La forma más sencilla de obtener un color específico es determinar la cantidad de color rojo, verde y azul que se requiere combinar para obtener el color deseado.

2.2.4 Mapa de bits. Es la representación de una imagen almacenada en la memoria de un computador como un conjunto de bits. El mapa de bits es una cuadrícula de filas y columnas de unos y ceros que el ordenador traduce en

píxeles dentro de la pantalla.

2.2.5 Formatos gráficos de mapa de bits (bitmap). Son archivos en los cuales se guarda información que conforma la imagen gráfica, formada por un patrón de píxeles. Los gráficos en mapa de bits son producidos por programas de pintura como Paintbrush, Microsoft Paint y algunos programas para escanear. Cuanto mayor sea la gama de colores, más realismo se consigue con este tipo de formato. Las imágenes bitmap poseen un tamaño natural en el cual se imprimirán perfectamente, pero si se cambia su tamaño, se pueden producir algunas distorsiones en la imagen. Si se aumenta el tamaño considerablemente, es fácil notar una disminución de la calidad. Las imágenes bitmap ocupan mucho espacio y cantidad de memoria, por eso algunos formatos bitmap utilizan diversos métodos para comprimir la información.

Las imágenes bitmaps están formadas por una rejilla de celdas. A cada una de estas celdas (píxeles), se le asigna un valor de color y luminancia propios. Por esto, cuando vemos todo el conjunto de celdas, tenemos la ilusión de una imagen de tono continuo.

Cuando creamos una imagen de mapa de bits se genera una rejilla específica de

píxeles. Por esto, al modificar su tamaño, transformamos, a su vez, la distribución y coloración de los píxeles, por lo que los objetos, dentro de la imagen, suelen deformarse. Esto es porque los objetos pierden o ganan algunos de los píxeles que los definen. Gracias a esta característica, que siempre hay que tener en cuenta, las imágenes de mapa de bits se crean con un tamaño determinado y pierden calidad si se modifican sus dimensiones.

2.2.6 Profundidad de colores en formatos bitmap. El bitmap es una imagen formada por miles o millones de puntos de colores, la profundidad ó resolución de colores está determinada por la cantidad de bits o bytes que puede contener un píxel. La tabla 3 muestra el esquema de resolución de colores.

Tabla 3. Esquemas de profundidad de colores.

Numero de Bits	*Colores*	*Píxeles por Bytes*
1	Blanco y negro	1 byte contiene 8 píxeles
4	16	1 byte contiene 2 píxeles
8	256	1 byte contiene 1 píxel
16	65536. [5]	2 bytes contienen 1 píxel
24	16 millones. [6]	3 bytes contienen 1 píxel

[5] Color de Alta Densidad.

[6] Color verdadero.

2.2.7 Imágenes de 24 bits. Las imágenes digitales son típicamente almacenadas en archivos de 8 ó 24 bits. Una imagen de 24 bits contiene más espacio para ocultar información. La variación de colores en los píxeles, son derivados de los tres colores primarios: Verde, Rojo, Azul. Cada color primario es representado por un byte; una imagen de 24 bits utiliza tres bytes por píxel para representar el valor del color. Esos tres bytes pueden ser representados como valores en el sistema hexadecimal, decimal y binario.

En las páginas web, el color de fondo es representado en el sistema hexadecimal por números de seis dígitos (tres pares de números representan el verde, rojo y azul). Para un fondo blanco el valor en el sistema hexadecimal sería FFFFFF, y los colores primarios representados en cada uno de los sistemas esta determinado en la tabla 4. Esta definición de Blanco en Páginas web, es similar a la definición de color de un píxel en una imagen de 24 bits.

Tabla 4. Color blanco representado en los diferentes sistemas.

Color primario	*Porcentaje*	*S. Hexadecimal*	*S. Decimal*	*S. Binario*
Rojo	100 %	FF	255	11111111
Verde	100 %	FF	255	11111111
Azul	100%	FF	255	11111111

2.2.8 Compresión en archivos de Imágenes. Dos clases de compresión son *lossless* (sin pérdida) y *lossy* (con pérdida). Ambos métodos ahorran espacio pero tienen diferentes resultados, interfiriendo con la información oculta al momento de realizar la descompresión.

La compresión *lossless* reconstruye el mensaje original exactamente; por lo tanto, este método es el preferido cuando se requiere que la información original permanezca intacta (en el caso de esteganografía). La compresión lossless es utilizada en imágenes con formato GIF[7] y en BMP de 8 bits.

La compresión *lossy* ahorra espacio, pero no mantiene la integridad de la imagen original. Este método es usado en imágenes con formato JPEG[8]. El formato JPEG suministra una alta aproximación en imágenes digitales pero no exactamente un duplicado.

2.3 FORMATO DE IMÁGENES BMP (Bitmapped File Format)

Con el surgimiento de Windows 3.x se desarrolla un nuevo formato gráfico bitmap

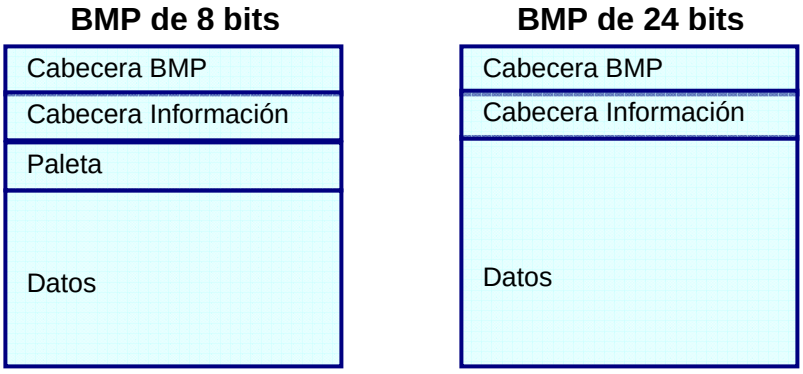
[7] Veáse numeral 2.5.1

[8] Veáse numeral 2.5.2

que constituye el estándar adoptado por este sistema operativo. Este formato guarda las imágenes descomprimidas lo que significa mayor velocidad de carga y mayor espacio requerido. Las imágenes BMP pueden tener una resolución de color de 1, 4, 8 y 24 bits.

2.3.1 Estructura Básica del formato BMP. La estructura de las imágenes BMP es sencilla, cuya lectura y escritura se realiza en el sistema binario. La figura 5 muestra los elementos que conforman la estructura básica del formato BMP de 8 y 24 bits.

Figura 5. Estructura básica del formato BMP de 8 y 24 bits.



CABECERA DEL ARCHIVO BMP, nos muestra información acerca de si se trata de un mapa de bits, el tamaño del archivo, y en qué punto del archivo comienza la

imagen en sí.

CABECERA DE INFORMACIÓN, contiene las dimensiones horizontales y verticales de la imagen, su profundidad de color, etc.

PALETA, es la paleta de colores. El formato de 24 bits no posee paleta de colores.

DATOS, es la imagen en sí.

2.3.1.1 Cabecera del archivo BMP. La tabla 5 muestra información acerca de la cabecera del archivo BMP

Tabla 5. Cabecera del archivo BMP.

Tamaño	***Contenido***	***Descripción***
2 bytes	Tipo de formato	Debe ser BM, por lo que en la practica se lee como dos bytes por separado
4 bytes	Tamaño del archivo	Tamaño en bytes del archivo
2 bytes	Reservado	Valor cero
2 bytes	Reservado	Valor cero
4 bytes	Offset	Posición en bytes en donde comienzan los datos en sí

2.3.1.2 Cabecera de información. Contiene las dimensiones horizontales y verticales de la imagen, su profundidad de color, etc. La tabla 6 muestra información acerca de la cabecera de información.

Tabla 6. Cabecera de información del formato BMP.

Tamaño	*Contenido*	*Descripción*
4 bytes	Tamaño cabecera	El tamaño en bytes (40) de la cabecera de información.
4 bytes	Ancho	El ancho en píxeles de la imagen.
4 bytes	Alto	La altura en píxeles de la imagen
2 bytes	Planos	Los planos del dispositivo de salida.
2 bytes	Bits	Los bits por píxel, para saber si es de 8 o 24 bits.
4 bytes	Compresión	Tipo de compresión, en la practica nunca se encuentran BMPs comprimidos. Por tanto vale cero.
4 bytes	Tamaño imagen	Tamaño de la estructura de datos (paleta y píxeles)
4 bytes	Píxeles por metro h	Píxeles por metro para dispositivos de salida, medida horizontal.
4 bytes	Píxeles por metro v	Píxeles por metro para dispositivos de salida, medida vertical.
4 bytes	Colores usados	Cantidad de colores usados, puede valer cero para que lo calcule el programa.
4 bytes	Colores importantes	Cantidad de colores "importantes" o cero si todos lo son.

2.3.1.3 Datos. Cargar un archivo BMP de una profundidad de color de 24 bits es más fácil que uno de 8 bits, puesto que no requiere cargar la paleta. Los píxeles se leen de izquierda a derecha y luego de abajo a arriba (en líneas horizontales y de abajo a arriba).

Cada píxel contiene una mezcla RGB que hace innecesaria una paleta. Los datos de 24 bits por tanto se guardan y se leen justo después de la cabecera de información. Para leer los datos de las imágenes de 24 bits, es decir, la información píxel a píxel, se almacena como se indica en la tabla 7**.**

Tabla 7. Contenido RGB en un píxel.

Tamaño	***Contenido***	***Descripción***
1 byte	azul	cantidad de verde
1 byte	verde	cantidad de azul
1 byte	rojo	cantidad de rojo

2.4 OTROS FORMATOS DE IMÁGENES

2.4.1 GIF[9] (Formato de Intercambio Gráfico). Fue creado por Compuserve en junio de 1987 y ampliamente utilizado para codificar e intercambiar archivos de gráficos en Internet. La primera versión se la llamó GIF87a, y a la segunda, GIF89a. Esta última versión presenta nuevas características para facilitar el manejo de imágenes en este formato. Los GIFs utilizan una paleta de entre 2 y 256 colores.

[9] Siglas en inglés de *Graphic Interchange Format*.

Los GIFs poseen una rutina de compresión muy eficaz que, aunque demora un poco la carga, reduce los archivos a un tamaño mucho menor que otros formatos como el BMP. Gracias a esa rutina de compresión que disminuye el tamaño de los archivos, el GIF es uno de los formatos preferidos para ser usado en Internet.

La resolución máxima alcanzada es la de 1024 x 768 píxeles en 256 colores, pero no hay razón por la cual no pueda crearse una imagen de mayor tamaño. Incluso hay GIFs que almacenan más de una imagen en un solo archivo. Su estructura está basada en bloques. Sobre todo desde la incorporación del GIF89a, se ha dado más importancia a los bloques. Estos pueden contener uno de estos elementos: una imagen, instrucciones acerca de cómo exhibirla, texto, información característica de alguna aplicación, un marcador que determina el final del archivo, etc. Muchos GIFs solamente contienen un bloque que determina su imagen.

Todos los GIFs poseen dos tipos de paleta: la paleta global y la paleta local. La global determina los colores de todas las imágenes almacenadas en el GIF, y la local determina específicamente la paleta de cada imagen del GIF (en el caso de haber una sola imagen, la única paleta disponible será la global). Existe un bloque llamado comment block, o "bloque de comentarios", donde puede incluirse un breve comentario personal acerca de la imagen en cuestión. Incluso existe una opción para aplicar a los GIF llamada interlacing. Consiste en lo siguiente:

generalmente, cuando un programa exhibe un GIF en pantalla, comienza desde la primera línea superior hasta llegar a la última línea inferior y de una pasada completa la imagen. Pero si el GIF es interlaced, la imagen se visualizará de otra manera: harán falta cuatro pasadas en lugar de una. En cada pasada se visualizan líneas que conforman la imagen, pero esta vez no aparecen seguidas una de la otra, sino distribuidas en la parte superior, central e inferior de la imagen. Este proceso se repite hasta finalizar las cuatro pasadas y completar esa imagen. Gracias al interlacing visualizamos distintas partes de la imagen al bajarla de Internet, y es posible darse cuenta si realmente nos sirve antes de que la imagen esté completa. Si esa imagen no es lo que esperábamos, es posible cancelar la operación.

Gracias a la popularidad de este formato, se han desarrollado infinidad de programas shareware para manipular GIFs, ya sea para exhibirlos, modificarlos, convertirlos o incluso comprimirlos.

2.4.2 JPEG[10] (*Grupo de Expertos en Fotografías Unidos*). Es un formato gráfico ideal para imágenes complejas de las escenas naturales del mundo real, como fotografías, arte realista y pinturas, también llamadas imágenes de tono

[10] Siglas en inglés de *Joint Photographic Experts Group*.

continuo o digitalizaciones de alta calidad. Fue desarrollado por el *Grupo de Expertos en Fotografías Unidos*[11].

El formato JPEG ofrece los imprescindibles 16 millones de colores, unido a una compresión realmente asombrosa. JPEG puede lograr rangos de compresión superiores a 20:1, los cuales son habituales. Sólo tiene una limitación: para obtener esos valores de compresión modifica sutilmente la imagen, descartándose su uso en aplicaciones en las que se desea mantener una calidad bit a bit. Si se intenta almacenar imágenes de tipo vectorial o dibujos sencillos no realísticos, se observará como la compresión disminuye enormemente, y las modificaciones hechas sobre la imagen original por el algoritmo de compresión se observan a simple vista.

El formato JPEG sólo puede almacenar imágenes de 24 bits, utilizando tres canales para su almacenamiento o de escala de grises, usando sólo un canal. La compresión JPEG consiste en una serie de complejas operaciones matemáticas, tales como: conversión del formato del color, transformada discreta del coseno (DCT), cuantizaciones y codificación entrópica. JPEG, junto con GIF, son los

[11] Un comité de expertos en gráficos por computadora, patrocinado en forma conjunta por la ISO (Organización Internacional de Estándares) y el CCITT (Comité Consultivo Internacional sobre Telefonía y Telegrafía).

formatos de imágenes más usados en Internet.

2.4.3 PNG[12] (Gráficos de Red Portátiles). Fue diseñado para reemplazar al formato GIF por ser este último más simple y menos completo. El objetivo de este formato es proporcionar compresión de imágenes sin pérdida para cualquier tipo de imagen.

Las características de este formato son:

- Color indexado hasta 256 colores y Color Verdadero hasta 48 bits por píxel.
- Mayor compresión que el formato GIF (+10%).
- Compresión sin pérdida.
- Visualización progresiva en dos dimensiones.
- Canal alfa (Transparencia variable)
- Detección de errores.

[12] Siglas en inglés de *Portable Network Graphics*.

- No permite animación.

2.5 ESTEGANOGRAFÍA

La palabra esteganografía viene del griego *steganos*, que significa *cubierto*, y *grafía* que significa *escritura*. Aunque en un principio se utilizó como sinónimo de criptografía, hoy en día se utiliza para definir el arte de ocultar la existencia de mensajes en una comunicación.

La esteganografía, es una ciencia que estudia todos los procedimientos utilizados para ocultar la existencia de un mensaje en una comunicación, por lo general dentro de un texto, una imagen, o una canción (en vez de ocultar su contenido). El objetivo de los métodos esteganográficos es que sea fácil ocultar un mensaje en otro y su proceso inverso; esto se consigue por medio de pequeñas modificaciones en otros datos digitales cuyo contenido no atraerá la atención de terceros. La persona que recibe el mensaje puede entonces extraer la información incrustada por medio del algoritmo de extracción.

Mientras que la criptografía pretende despistar a un intruso que consigue un

mensaje, para que éste no sea capaz de averiguar su contenido, el propósito de la Esteganografía es ocultar el mensaje dentro de otro que no contiene información importante, pero que si posee sentido por sí solo, de forma que el atacante ni siquiera se entere de la existencia de dicha información oculta. La esteganografía no trata de sustituir al cifrado convencional sino que intenta complementarlo, pues al ocultar el mensaje se reducen las posibilidades de que sea descubierto. No obstante, si se desea adicionalmente cifrar el mensaje antes de incrustarlo en unos datos que utilizaremos como escondite, se obtendrá un mayor grado de seguridad.

La esteganografía requiere dos archivos. El primero es el denominado ***archivo portador*** (en nuestro caso, una imagen BMP de 24 bits). El segundo archivo es el ***mensaje*** a ocultar; este mensaje puede ser texto plano, un texto encriptado, una imagen, un sonido o cualquier archivo que pueda leerse en forma de bits. Como resultado, este sistema arroja una información parecida a la original pero lleva incrustada el mensaje que se quiere transmitir[13]. Los métodos esteganográficos aprovechan la información redundante (bits) que puede haber en los archivos participantes del proceso.

[13] Para el proyecto HECOBI, el archivo de salida es denominado “Estego Imagen”.

Los expertos en esteganografía[14] no recomiendan usar imágenes JPEG, en lugar de esto recomiendan utilizar el método *lossless* en imágenes de 24 bits. La siguiente mejor alternativa a imágenes de 24 bits es 256 colores o en escala de grises. En imágenes de ocho bits como el formato GIF, cada píxel es representado por un byte, y cada píxel simplemente pinta un color de una tabla (paleta de colores) con 256 colores. Los valores de los píxeles están entre 0 y 255.

La esteganografía ha surgido actualmente por la utilización de archivos informáticos de imágenes digitalizadas para ocultar datos. Es una forma de proteger la propiedad intelectual de una imagen publicada. Si alguna persona realiza una copia, pasará desapercibido que en el código digital de la imagen esta introducido el nombre del autor.

Algunas organizaciones han optado por el uso de la esteganografía para proteger sus datos y enviarlos a través del Internet, esto debido a las fuertes medidas que han tomado sus países acerca del uso de la criptografía.

2.5.1 Historia de la esteganografía. A través de la historia, las personas han

[14] http://www.jjtc.com/neil

utilizado diferentes métodos para ocultar información. En la antigua Grecia de Herodoto, los textos eran escritos en tablas cubiertas de cera. Para ocultar un mensaje, una persona debía quitar la cera de la tableta, luego escribían el mensaje sobre la madera y volvían a cubrirla con cera. De esta forma pasaban inspecciones sin levantar sospechas.

Otro método era afeitar la cabeza de los mensajeros, tatuarle el mensaje sobre el cuero cabelludo, esperar que le creciera el pelo y mandarlos con el mensaje oculto por el cabello. El destinatario lo afeitaba nuevamente para poder leerlo. Otra forma inocente de enviar mensajes era escribiendo un texto y luego tomando la primera letra de cada palabra, se podía leer el mensaje que en realidad se quería transmitir.

Quizás la forma más conocida de hacer esteganografía es utilizando tintas invisibles. Estas tintas fueron utilizadas con éxito durante la Segunda Guerra Mundial y por la resistencia en los Campos Nazis. Así, una carta que parecía inofensiva, sin ningún mensaje importante, podía contener un mensaje escrito entre las líneas de ésta. Entre las tintas más utilizadas estaba la leche, el vinagre, los zumos de frutas y la orina, debido a que cuando se calentaban éstas se oscurecían. La facilidad para descubrir estos mensajes ocultos y el avance de la tecnología hicieron que se inventasen tintas que sólo eran visibles si se les hacía

reaccionar con ciertos reactivos específicos.

Otro sistema esteganográfico de gran utilidad en la Segunda Guerra Mundial, fue el uso de plantillas con agujeros, los cuales seleccionaban letras o palabras de un mensaje y de esta forma se leía el mensaje escondido. También en aquella misma época comenzó a utilizarse micropunto. Un mensaje secreto era fotográficamente reducido a la medida de un punto y pegado como el de la letra i en un papel conteniendo un mensaje escrito.

2.5.2 Usos de la Esteganografía. Los principales usos de la esteganografía son los siguientes:

- ✓ Envío y protección de información confidencial a través de un medio como Internet, esto debido a las fuertes medidas que han tomado algunos países acerca del uso de la criptografía.
- ✓ Protección de la propiedad intelectual de imágenes publicadas en Internet.

- ✓ En el sector de la Salud, la ocultación de mensajes en secuencias de ADN; esto es utilizado para proteger la propiedad intelectual en medicina y biotecnología.

- ✓ En Medicina como medida de seguridad, ocultamiento de los datos en la imagen de los pacientes (nombre, fecha de nacimiento, historia clínica).

La Esteganografía ha tenido algunos usos indebidos, en la comunicación de terroristas a través de Internet, en redes de pornografía infantil que comparten sus imágenes ocultas dentro de otros archivos inofensivos. Esto ha generado que algunos países desarrollen técnicas para detectar mensajes ocultos.

2.5.3 Utilidades derivadas de la Esteganografía.

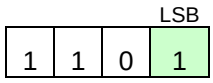
- ✓ **Marcas de agua digitales.** Los sistemas de marca de agua modifican sutilmente los bits que constituyen el archivo, marcándolo como perteneciente a él, de forma que el resultado es sensiblemente igual que el original. Lo que se oculta al usar marcas de agua es información sobre el propietario, con el fin de proteger los derechos de copia; por eso la información incrustada es perfectamente visible, lo que requiere una mayor robustez contra manipulaciones que pudiesen intentar eliminar la marca de agua.

- ✓ **Inclusión de huellas dactilares.** Esta técnica sirve para identificar objetos

concretos entre otros similares. Por ejemplo, añadir números de serie en películas digitales, discos compactos, libros y otros productos multimedia. Es una forma de proteger la propiedad intelectual; si alguien trata de realizar una copia fraudulenta, pasará desapercibido que en el código digital está intercalado el nombre del autor.

2.6 Método LSB[15] (Bit Menos Significativo). La información puede ser escondida de distinta forma en las imágenes, pero el LSB es uno de los métodos más usados en esteganografía.

Figura 6. Bit menos significativo. Número decimal 13.



El LSB es el bit que menos información brinda. Por ejemplo, si al número binario 1101 (13) se le modifica el primer Bit se obtiene 0101 (5), pero si se le modifica el último Bit se obtiene 1100 (12); esto implica un cambio en el valor total del número mucho menor, y ayuda a que el cambio pase desapercibido. De esta forma el color de la imagen casi no varía, por eso para el ojo humano con el resultado de esta

[15] Siglas en Inglés de *Least Significant Bit*.

esteganografía, sería imposible distinguir esa diferencia en una imagen total.

Si se convierte una imagen de formato BMP ó GIF, los cuales reconstruyen exactamente el mensaje (compresión sin pérdida), a JPEG (compresión con pérdida) la información oculta en los Bits Menos Significativos podría perderse o ser destruida.

2.6.1 LSB en imágenes de 8 bits. Las imágenes de 8 bits no son ideales para ocultar mensajes usando el método LSB, debido a la limitación de colores (256 opciones de colores por píxel frente a 16777216 colores que maneja una imagen de 24 bits). Sin embargo, se idean algunas estrategias para ocultar información en imágenes de 8 bits. La imagen debe ser cuidadosamente seleccionada para no presentar sospechas sobre la existencia de un mensaje oculto.

Cuando la información oculta es insertada en los Bits Menos Significativos, la entrada de colores en la paleta es modificada. Por ejemplo, una paleta de cuatro colores (blanco, rojo, azul y verde), tiene como posiciones de entrada de colores:

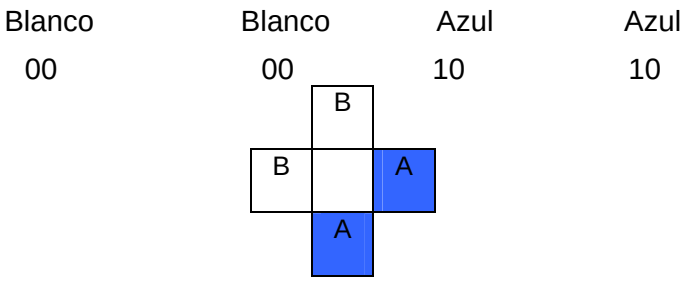
0 (00) Blanco

1 (01) Rojo

2 (10) Azul

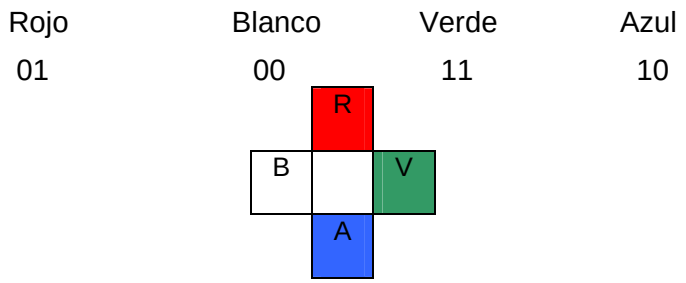
3 (11) Verde

Figura 7. Píxeles adyacentes B, B, A, A.



Supongamos cuatro píxeles adyacentes (Figura 7), necesitamos ocultar el valor de 10 que en sistema binario es 1010, al implementar el LSB quedaría como se muestra en la figura 8.

Figura 8. Píxeles adyacentes B, B, A, A con LSB.

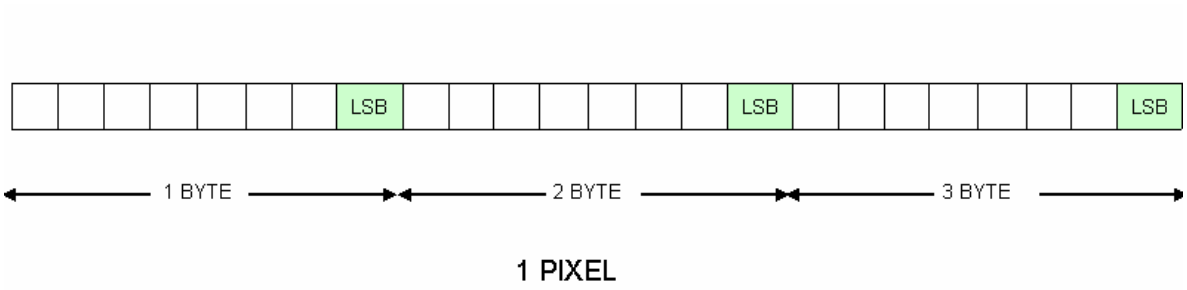


Este brusco cambio en la imagen es visible y claramente detectable la vulnerabilidad en imágenes de 8 bits.

2.6.2 LSB en imágenes de 24 bits. Una imagen de 24 bits es lo ideal para esconder información, aunque estas imágenes pueden llegar a ser grandes en tamaño, pero para solucionar este problema, existe algo que nos puede ayudar: la compresión de archivos. Programas como Winzip® o Winrar® nos permiten realizar una compresión bastante considerable a los archivos de imágenes de 24 bits. El método LSB es magnífico para ocultar información en imágenes de 24 bits que utilicen compresión sin pérdida (BMP o GIF).

Para esconder un mensaje en una imagen de 24 bits en los Bits Menos Significativos de cada byte, se puede almacenar 3 bits en cada píxel (1 píxel = 3 bytes) como se indica en la figura 9.

Figura 9. Implementación del LSB en imagen de 24 bits.



Una imagen de 1024x768 (786432 píxeles) equivalente a 2359296 bytes puede ocultar igual cantidad de bits como máximo (un bit por cada byte), es decir, puede ocultar 294912 bytes (288 Kb) de información. En algunas imágenes de 24 bits se puede ocultar datos en el segundo bit menos significativo y aún así el ojo humano no podría observar las modificaciones realizadas en la imagen.

La siguiente fórmula permite calcular la cantidad de datos (Kilobytes) que se pueden ocultar en una imagen digital usando el método LSB:

Cantidad de datos a ocultar (Kilobytes) = (**TA – TC) / (8 * 1024)**

Donde,

TA (bytes) = Tamaño del archivo de la imagen

TC (bytes) = Tamaño de la cabecera de la imagen

2.7 Método DCT (Discrete Cosine Transform, Transformada Discreta Coseno). Las llamadas *transformadas* son herramientas matemáticas que permiten representar cualquier función - continua o discreta - a través de una serie de coeficientes.

En general, se necesitarán un número infinito de coeficientes en el caso continuo, y tantos coeficientes como valores en el caso discreto, para poder representar de manera absolutamente precisa la imagen. La ventaja que tiene trabajar con transformadas es que la mayor parte de la información suele estar concentrada en un número relativamente pequeño de coeficientes, por lo que se pueden obtener buenas aproximaciones de la imagen original a partir de un subconjunto de la totalidad de sus coeficientes, que serán más o menos precisas en función del número de coeficientes que se conserven.

Los formatos de compresión de archivos multimedia más comunes emplean distintos tipos de transformada, como la Transformada Discreta del Coseno. La imagen se divide en regiones de 8 x 8 píxeles de tamaño. A cada trozo se le aplica una transformada, y los coeficientes resultantes se truncan, teniendo en cuenta que los sentidos del ser humano son más sensibles a determinadas características, se les da prioridad a los que mejor se percibe, para que las distorsiones sean poco perceptibles. Finalmente se aplica un algoritmo de compresión sin pérdida al resultado, que permite recuperar exactamente los mismos datos que fueron obtenidos para cada una de las muestras durante el proceso de digitalización, eliminando la redundancia de la cadena de bits correspondiente, de forma que al descomprimirla se obtiene una copia idéntica a la original, y se obtiene el archivo final.

Si se quiere ocultar un mensaje dentro de una imagen, habrá que manipular directamente los coeficientes, e introducir en ellos los bits del mensaje. Como es lógico, ya que los bits que se preservan en los coeficientes son los que mayor cantidad de información transportan, también serán más sensibles a alteraciones, por lo que se podrá ocultar muchos menos bits del mensaje huésped si se quiere mantener un nivel de distorsión en el resultado final que sea realmente imperceptible.

2.7.1 Características del DCT. En general, cualquier trasformación a la que se someta una imagen, pretende encontrar una función matemática que se "adapte a las imágenes" y por tanto debe presentar las siguientes características:

a) Permite pasar de valores de brillos o componentes a unos elementos matemáticos denominados coeficientes.

b) La transformación es reversible.

c) Debe concentrarse la energía en pocos coeficientes, para así poder desechar otros muchos con valores bajos sin cometer apenas error.

d) Los coeficientes deben ser lo mas independientes posibles. Con esto se consigue que la información se encuentre concentrada sobre cada coeficiente y no

sobre el conjunto de ellos.

e) La transformación debe ser sencilla de implementar.

2.7.2 Fórmula Matemática del DCT. La DCT se define como:

$$C[u,v] = \frac{2}{\sqrt{MN}} K(u) K(v) \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m,n] \cos\frac{(2m+1)\pi u}{2M} \cos\frac{(2n+1)\pi v}{2N}$$

Donde,

x[m, n] es una matriz de muestras (imagen)

$$x[m,n] = \begin{pmatrix} x_{0,0} & x_{1,0} & .... & x_{M-1,0} \\ x_{0,1} & . & . & ... \\ . & . & ..... & .... \\ x_{0,N-1} & . & ..... & x_{M-1,N-1} \end{pmatrix} \begin{matrix} \xrightarrow{DCT} \\ \xleftarrow{IDCT} \end{matrix} \begin{pmatrix} C_{0,0} & C_{1,0} & .... & C_{M-1,0} \\ C_{0,1} & . & . & ... \\ . & . & ..... & .... \\ C_{0,N-1} & . & ..... & C_{M-1,N-1} \end{pmatrix} = C[u,v]$$

m, n u, v son posiciones en las matrices

M, N son las dimensiones de la matrices

$K(u) = \frac{1}{\sqrt{2}}$ si u = 0 $K(u) = 1$ si u ≠ 0

$K(v) = \frac{1}{\sqrt{2}}$ si v = 0 $K(v) = 1$ si v ≠ 0

La IDCT se define como:

$$x[m,n] = \frac{2}{\sqrt{MN}} \sum_{u=0}^{M-1} K(u) \sum_{v=0}^{N-1} K(v)\, C[u,v] \cos\frac{(2m+1)\pi\, u}{2M} \cos\frac{(2n+1)\pi\, v}{2N}$$

Los valores de los coeficientes C[u,v] formarán una matriz de M filas y N columnas donde:

$0 \leq u \leq M$ $0 \leq v \leq N$

2.7.3 Funciones base para la DCT. Si implementáramos estas fórmulas con cualquier lenguaje, no cabe duda que obtendríamos estos mismos resultados de forma más rápida y menos tediosa que de forma manual. No obstante, esto resultaría bastante poco eficiente dado el alto grado de anidamiento del código (varios “for” anidados). Con objeto de mejorar esto, podemos introducir una serie de constantes que van a mejorar la eficiencia del código.

Para la DCT particularizando para M = N =8 aparecerán 64 funciones base:

f [m , n] u =0 v = 0
f [m , n] u =0 v = 1
....
f [m , n] u =7 v = 0
....
f [m , n] u = 7 v = 7

Una función base genérica f [m, n] se obtendría, particularizando “u” y “v”, sobre la expresión:

$$\cos\frac{(2m+1)\pi\, u}{2M}\cos\frac{(2n+1)\pi\, v}{2N}$$

Particularizando además para N = M = 8, cada función base será una matriz de 8x8 valores hasta un total de M x N funciones base.

Ejemplo:

$$f[m, n]_{u=0v=0} = \cos\frac{(2m+1)\pi\, 0}{16}\cos\frac{(2n+1)\pi\, 0}{16} = 1$$

Independientemente de “m” y “n”. Se obtiene una matriz de 8x8 unos.

Análogamente, si se calcula la función base para el caso de u = 3 v = 5 se obtiene la siguiente matriz:

$$f[m, n]_{u=3v=5} = \cos\frac{(2m+1)\pi\, 3}{16}\cos\frac{(2n+1)\pi\, 5}{16}$$ dependiente de “m” y ”n”

$$f[m,n]_{u=3v=5} = \begin{pmatrix} 0.46 & -0.81 & 0.16 & 0.69 & 0.69 & -0.16 & 0.81 & -0.46 \\ -0.10 & 0.19 & 0.03 & -0.16 & 0.16 & 0.03 & -0.19 & 0.10 \\ -0.54 & 0.96 & -0.19 & -0.81 & 0.81 & 0.19 & -0.96 & 0.54 \\ -0.30 & 0.54 & -0.10 & -0.46 & 0.46 & 0.10 & -0.54 & 0.30 \\ 0.30 & -0.54 & 0.10 & 0.46 & -0.46 & -0.10 & 0.54 & -0.30 \\ 0.54 & -0.96 & 0.19 & 0.81 & -0.81 & -0.19 & 0.96 & -0.54 \\ 0.10 & -0.19 & 0.03 & 0.16 & -0.16 & -0.03 & 0.19 & -0.10 \\ -0.46 & 0.81 & -0.16 & -0.69 & 0.69 & 0.16 & -0.81 & 0.46 \end{pmatrix}$$

Una representación grafica de gran utilidad vendría dada al asignar distintas tonalidades de brillo a cada una de estos valores comprendidos entre 1 y -1 (por tratarse de un coseno) asignando como extremos el color negro al -1 y el blanco al +1. Mientras tanto los valores intermedios tomarían brillos entre 0 y 255 (8 bits para el brillo). La representación más sencilla es la correspondiente a la función f [m ,n] u=0 v=0 para la que se obtiene una matriz de 8 x 8 unos y en consecuencia una representación de 8 x 8 blancos.

Existe otra forma para obtener la DCT e IDCT que resulta más eficiente. En su aplicación es necesario conocer de antemano el tamaño de la muestra a codificar. Para el caso de nuestro ejemplo M = N = 2 y por tanto:

$$C[u,v] = K(u)K(v)\sum_{m=0}^{1}\sum_{n=0}^{1} x[m,n]\cos\frac{(2m+1)\pi u}{2M}\cos\frac{(2n+1)\pi v}{2N}$$

Ecuación que podemos expresar como:

$$C[u,v] = \sum_{m=0}^{1} K(u)\cos\frac{(2m+1)\pi\, u}{4} \sum_{n=0}^{1} x[m,n]\cos\frac{(2n+1)\pi\, v}{4}$$

Matricialmente esta ecuación puede escribirse como:

$[C] = [A]^T.[X].[A]$ donde $[A]$ tiene la expresión

aij= $k\cos\frac{(2j+1)\pi\, i}{4}$ con k = 1 cuando i $\neq$0

k = $\frac{1}{\sqrt{2}}$ cuando i =0

La expresión para la IDCT se obtiene despejando [x]:

$[X] = [A].[C].[A]^T$

Se trata en este punto de asociar brillos (escala de grises) a los valores de los coeficientes, para poder interpretar mejor las variaciones relativas entre los diferentes coeficientes obtenidos tras la transformación. Realmente lo que se hace es tomar el valor absoluto de los coeficientes puesto lo que nos interesa ahora es observar las posiciones en las que se concentra mayor energía y no así el valor de los coeficientes.

Si el conjunto obtenido se muestra como una imagen, de un vistazo se aprecia en qué posición de la matriz los coeficientes son altos (blanco), en cuáles son bajos (negro) y las zonas que presentan características similares.

2.8 ESTADO DEL ARTE

2.8.1 Perspectiva mundial. A continuación algunos aportes y hechos relacionados con la esteganografía, criptografía, firma digital y temas afines, se presentan haciendo énfasis en el año en que fueron realizados:

En 1970, se logra que la criptografía sea digital gracias al aumento en la capacidad de procesamiento de los computadores.

En 1976, se desarrolló un sistema criptográfico llamado DES (Data Encryption Standard). Este sistema se soporta en complicados sistemas matemáticos de sustitución y transposición.

En 1976, Diffie y Hellman crean el protocolo de clave pública y esquema de intercambio de claves.

En 1977, crean el RSA (Rivest Shamir Adelman) que lleva las siglas de sus autores.

En 1980, es la década de la revolución criptográfica, en donde se proponen nuevos algoritmos, funciones hash y sistemas de autentificación basados en algoritmos criptográficos como PKI (Infraestructura de clave pública) y Kerberos.

En 1990, la tecnología digital brinda una nueva vía para aplicar técnicas esteganográficas, con la aparición de los formatos de imagen digital.

En 1991, el profesor Ronald Rivest del MIT creó el algoritmo criptográfico MD5.

En 2000, fue designado el algoritmo Rijndael como AES (Advanced Encryption Standard).

2.8.2 Perspectiva nacional y local. Se destacan dentro de los trabajos desarrollados en el ámbito nacional y local:

En 1992, se dictó el seminario de criptografía en la UIS por parte del profesor Hernán Porras Díaz, Doctor en telecomunicaciones, quien dirigió el proyecto de grado “Implementación del algoritmo criptográfico DES”. Universidad Industrial de Santander. 1990.

Firmas digitales utilizando curvas elípticas. Universidad Autónoma de Bucaramanga. Martínez, Juan Carlos. Asesor: Zúñiga G., Wilson. 2000.

Stegamail: criptografía y esteganografía sobre una conexión de tipo SMTP. Universidad Nacional de Colombia. Duque Méndez, Néstor Darío. 2003.

Esteganografía en HTTP[16]. Bonilla, Luis Jorge. Llano, Mónica. Ramírez, Alfonso, Especialistas en Telecomunicaciones y Negocios por Internet. Escuela Colombiana de Ingeniería. 2004.

En la Universidad Distrital Francisco José de Caldas actualmente se dicta un Diplomado en Seguridad Informática, tratando temas como la criptografía, esteganografía y la Técnica del Bit Menos Significativo.

En la actualidad en las Universidades de Colombia, aunque existen grupos de Investigación en Seguridad en Informática, no se han desarrollado proyectos de Esteganografía en imágenes, utilizando técnicas como la Transformada Discreta del Coseno y el LSB.

[16] HyperText Transfer Protocol. Protocolo que permite realizar saltos hipertextuales.

2.9 MARCO LEGAL

El congreso de Colombia, aprobó el uso de las firmas digitales como elemento probatorio y reglamentó su uso mediante la ley 527 del 18 de agosto de 1999, “por medio de la cual se define y reglamenta el acceso y uso de los mensajes, del comercio electrónico y de las firmas digitales, y se establecen las entidades de certificación”.

En el capitulo I se definen las firmas digitales como:

“Se entenderá como un valor numérico que se adhiere a un mensaje de datos y que, utilizando un procedimiento matemático conocido, vinculado a la clave del iniciador y al texto del mensaje permite determinar que este valor se ha obtenido exclusivamente con la clave del iniciador y que el mensaje inicial no ha sido modificado después de efectuada la transformación”.

En la parte III “Firmas digitales, Certificados digitales y Entidades de certificación”, dice:

Artículo 28. Atributos jurídicos de una firma digital. Cuando una firma digital haya sido fijada en un mensaje de datos se presume que el suscriptor de aquella tenía la intención de acreditar ese mensaje de datos y de ser vinculado con el contenido del mismo.

Parágrafo. El uso de una firma digital tendrá la misma fuerza y efectos que el uso de una firma manuscrita, si aquélla incorpora los siguientes atributos:

1. Es única a la persona que la usa.
2. Es susceptible de ser verificada.
3. Está bajo el control exclusivo de la persona que la usa.
4. Está ligada a la información o mensaje, de tal manera que si éstos son cambiados, la firma digital es invalidada.
5. Está conforme a las reglamentaciones adoptadas por el Gobierno Nacional.

3. DISEÑO DEL SOFTWARE

3.1 PROTOTIPO INICIAL

Este prototipo toma como fundamento la investigación previamente hecha, dando un primer paso hacia la implementación de los métodos más sencillos a analizar para realizar esteganografía en imágenes. Este proceso constituye una herramienta de análisis de la estructura del formato BMP y la implementación del método LSB en imágenes. En la tabla 8 se presentan los casos de uso generales identificados para el primer prototipo, para el cual se implementó una interfaz de prueba.

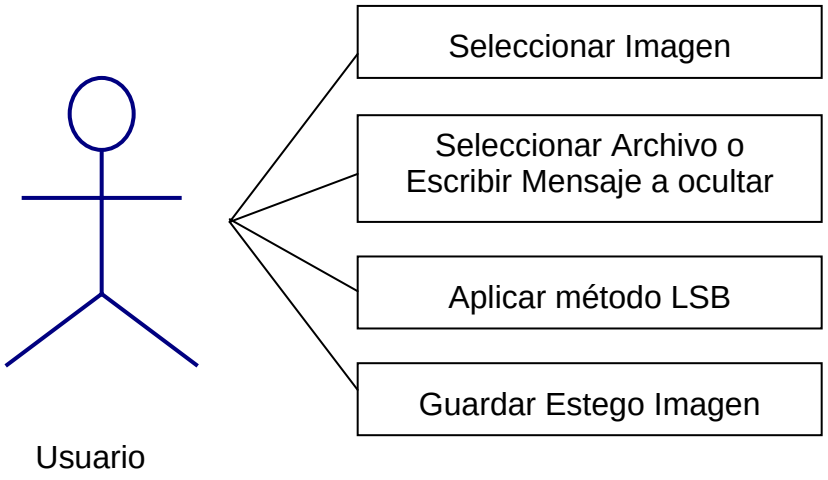
Tabla 8. Caso de uso general. Prototipo Inicial.

	Caso de uso	***Descripción***
1	Ocultar Información	El usuario puede ocultar un mensaje de texto o archivo de cualquier tipo en una imagen BMP de 24 bits
2	Revelar Información	El usuario extrae el archivo o mensaje de texto oculto en la imagen BMP de 24 bits

3.1.1 Descripción de los casos de uso.

- ✓ **Caso de uso Ocultar Información.** El diagrama de caso de uso ocultar información se observa en la figura 10. El usuario debe inicialmente abrir un archivo de imagen con formato BMP de 24 bits y luego seleccionar la información a ocultar; es necesario que el tamaño de la información que se desea esconder, sea mucho menor que el tamaño de la imagen. Posteriormente inicia el proceso de esteganografía con el método LSB y por último guarda la estego imagen.

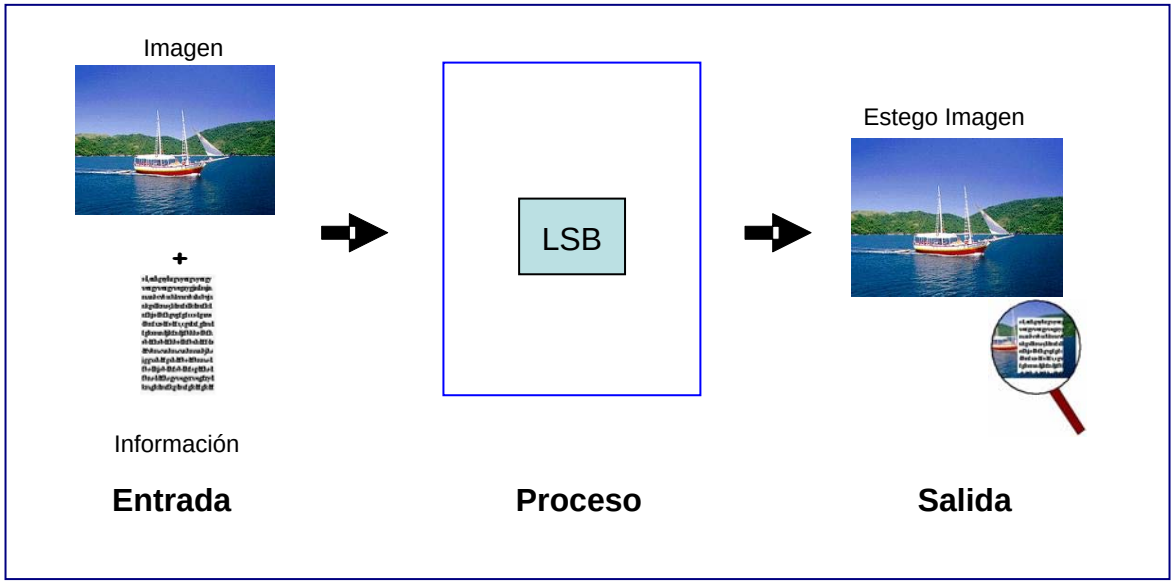
Figura 10. Caso de uso Ocultar Información. Prototipo Inicial.



La figura 11 muestra el esquema de funcionamiento para ocultar información en el desarrollo del primer prototipo. La imagen BMP y la información a ocultar son los datos de entrada al proceso, que en este caso es la aplicación del método LSB;

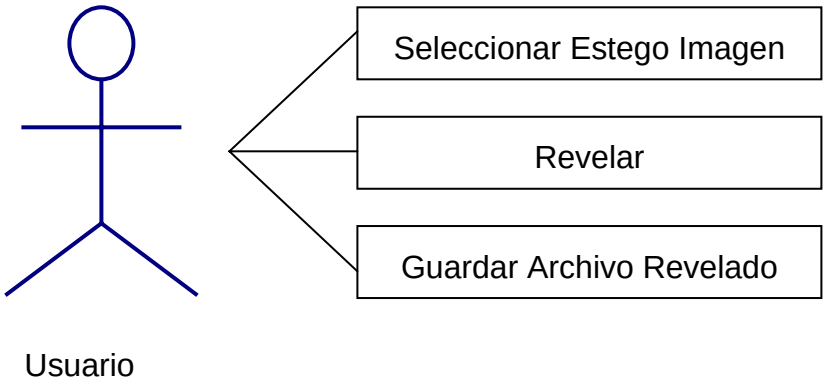
como resultado se tiene una estego imagen de salida, quien representa el producto principal.

Figura 11. Esquema de funcionamiento para ocultar información. Prototipo Inicial.



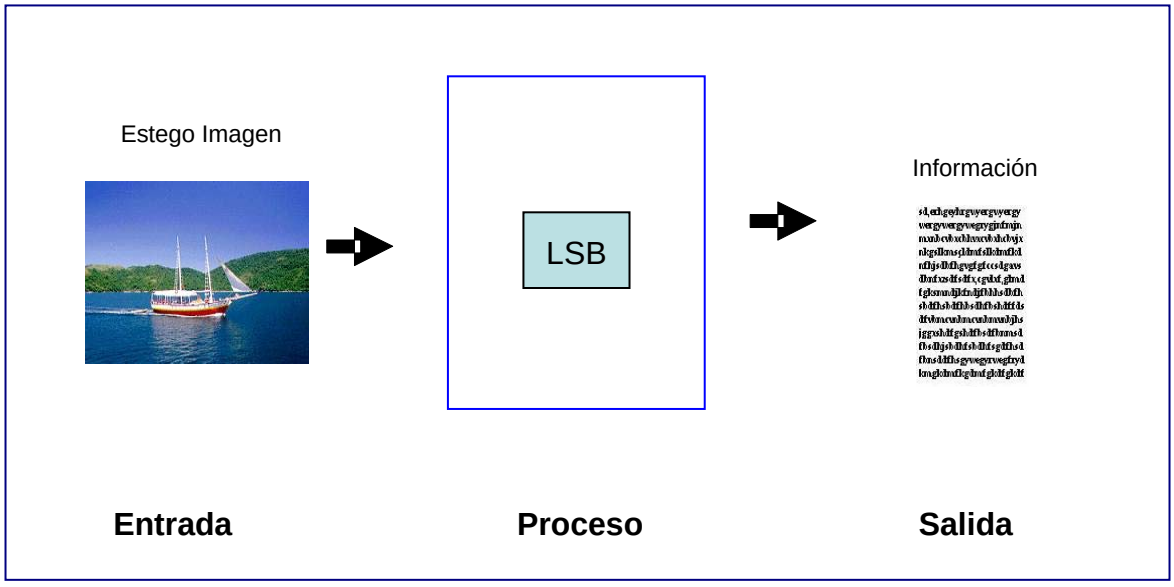
- ✓ **Caso de uso Revelar Información.** El diagrama de caso de uso revelar información se observa en la figura 12. El usuario inicialmente selecciona un archivo de imagen con formato BMP de 24 bits, la cual debe contener información oculta (estego imagen). Posteriormente aplica el método esteganográfico LSB para revelar la información y la guarda en un archivo.

Figura 12. Caso de uso Revelar Información. Prototipo Inicial.



La figura 13 muestra el esquema de funcionamiento para revelar la información del primer prototipo. La estego imagen BMP es el dato de entrada al proceso, que en este caso es la aplicación del método LSB; como resultado se tiene la información que se encontraba oculta.

Figura 13. Esquema de funcionamiento para revelar información. Prototipo Inicial.



3.2 REFINAMIENTO DEL PROTOTIPO

Las pruebas efectuadas al primer prototipo dieron como exitosos los procedimientos y algoritmos utilizados para efectuar la esteganografía. En el refinamiento del prototipo se agrega un nuevo método esteganográfico, denominado Segundo Bit Menos Significativo (2LSB), el cual utiliza el primer y segundo bit menos significativo para ocultar datos.

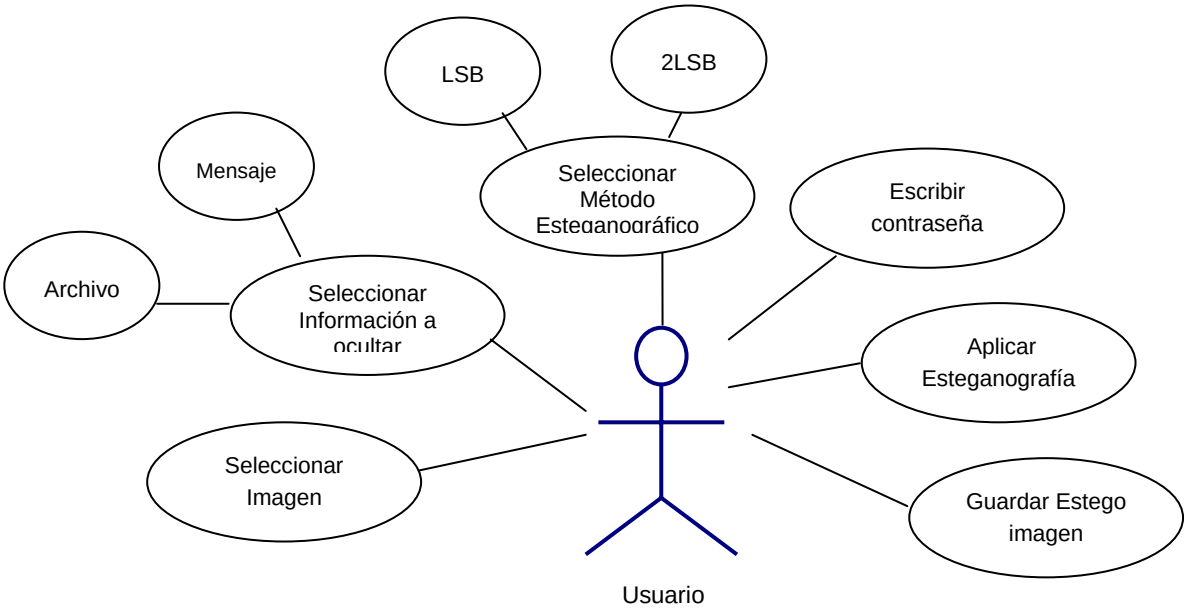
Para darle mayor seguridad a la herramienta se utilizaron conceptos criptográficos analizados en la etapa de análisis, se implementan el algoritmo AES con la finalidad de encriptar la información que se quiere ocultar y el algoritmo MD5, para verificar la integridad de la misma.

3.2.1 Descripción de los casos de uso. El diagrama de casos de uso general se conserva[17], pero se realizan algunas mejoras a los casos de uso Ocultar Información y Revelar Información.

- ✓ **Caso de uso Ocultar Información.** Las mejoras al diagrama de caso de uso ocultar información se observa en la figura 14.

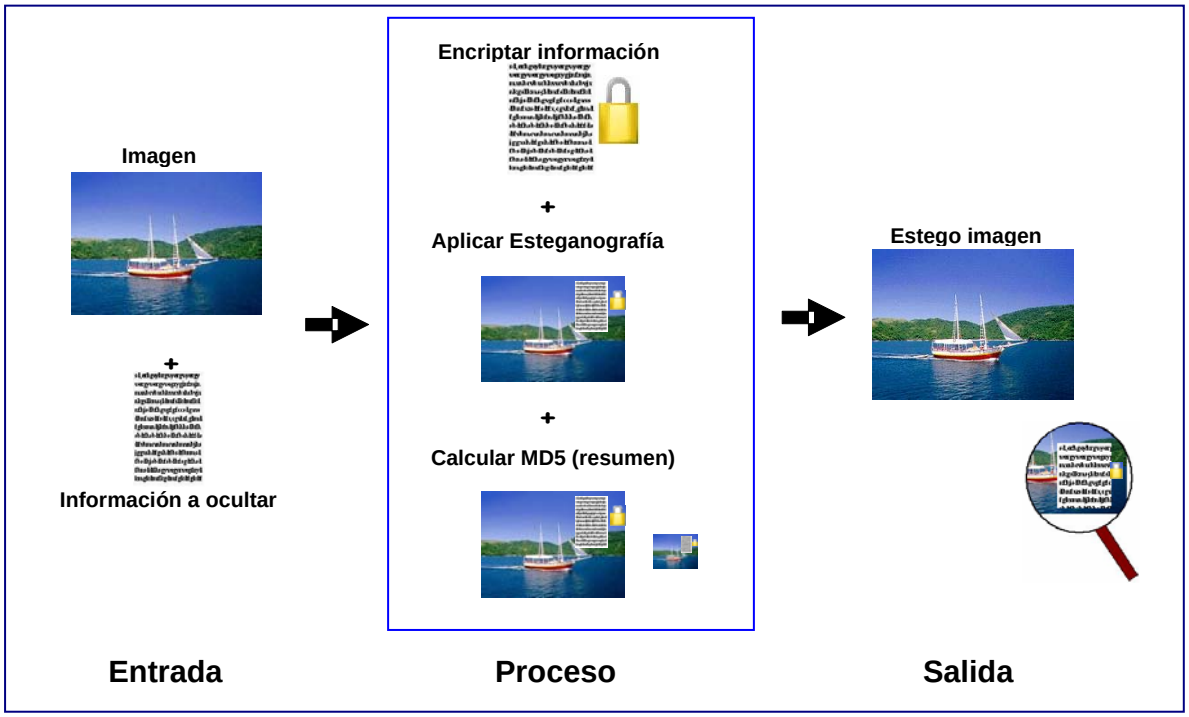
[17] Véase Tabla 8.

Figura 14. Caso de uso Ocultar Información. Prototipo Intermedio.



La figura 15 muestra el esquema de funcionamiento para ocultar información en el desarrollo del prototipo intermedio.

Figura 15. Esquema de funcionamiento para ocultar información. Prototipo Intermedio.



✓ **Caso de uso Revelar Información.** Las mejoras al diagrama de caso de uso revelar información se observa en la figura 16.

Figura 16. Caso de uso Revelar Información. Prototipo Intermedio.

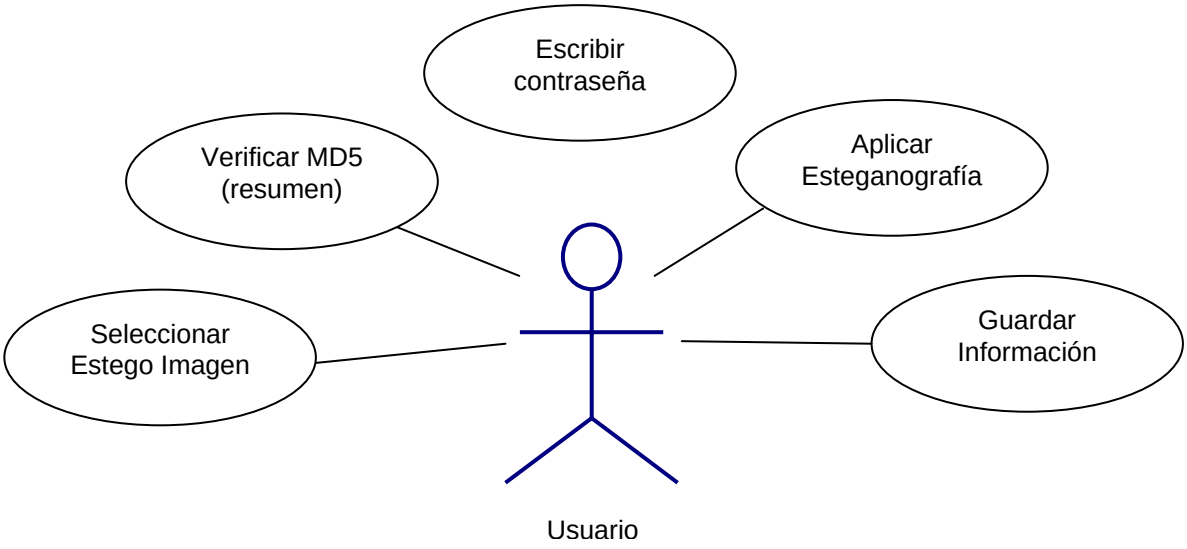
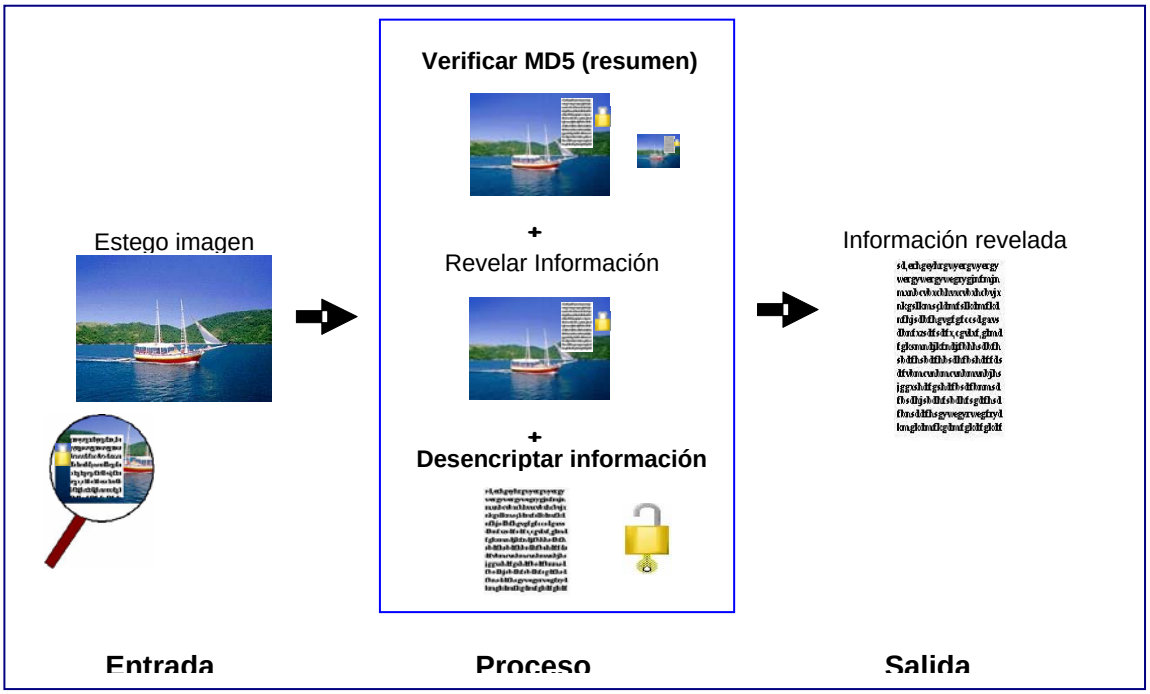


Figura 17. Esquema de funcionamiento para revelar información. Prototipo Intermedio.



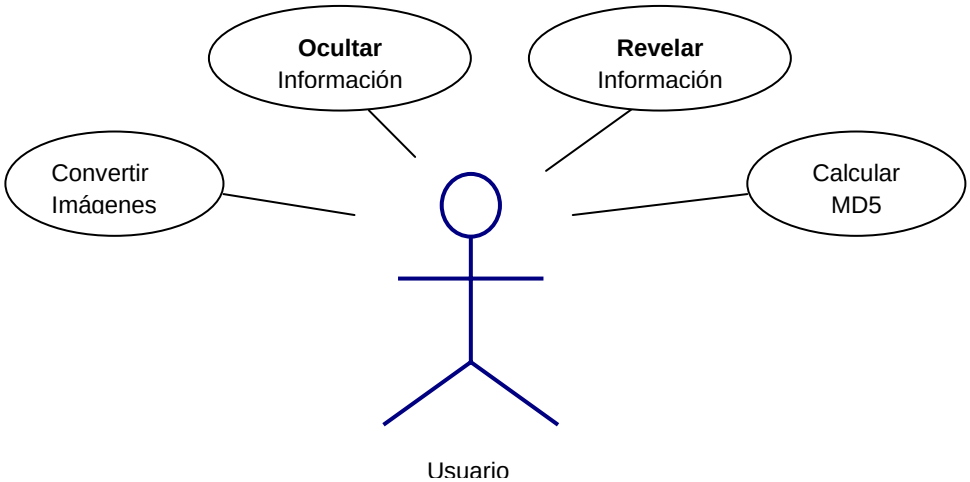
La figura 17 muestra el esquema de funcionamiento para revelar la información en el desarrollo del prototipo intermedio.

3.3 PROTOTIPO FINAL

En el desarrollo del prototipo final se tuvo en cuenta la detección de errores al realizar las pruebas y la recopilación de cambios sugeridos a la herramienta. Además se definió el diseño gráfico que tendrá finalmente la herramienta, así como el logotipo que la identifica.

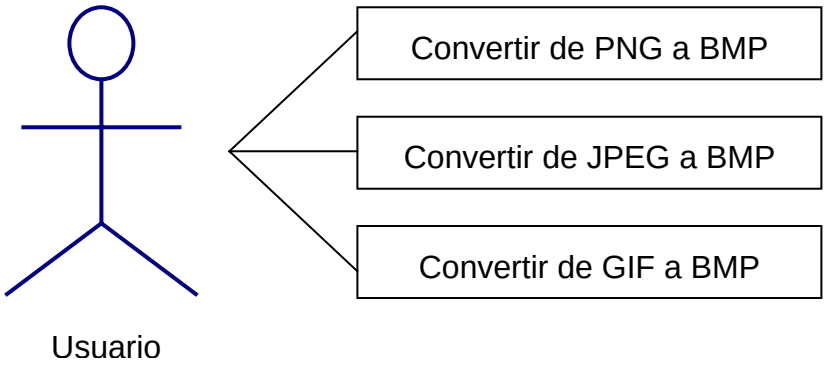
3.3.1 Descripción de los casos de uso. En el diagrama de casos de uso general se incluyen dos nuevos casos de uso: Caso de uso *convertir imágenes* y caso de uso *calcular MD5*. En el caso de uso *ocultar información* se incluye un nuevo método esteganográfico: el método DCT. La figura 18 muestra el diagrama de uso general del prototipo final.

Figura 18. Diagrama de casos de uso general. Prototipo General.



✓ **Caso de uso Convertir Imágenes.** El diagrama de caso de uso *convertir imágenes* se observa en la figura 19. El usuario tiene la posibilidad de utilizar otros tipos de formatos de imágenes para realizar esteganografía: PNG, JPEG y GIF. Para que esto sea posible, el usuario selecciona la imagen, si el formato de la imagen es PNG, JPEG o GIF debe convertirla al formato BMP de 24 bits.

Figura 19. Caso de uso Convertir imágenes. Prototipo Final.



3.4 ENTORNO DEL SISTEMA

3.4.1 Población. En la etapa de diseño, se advirtió la necesidad de crear un software amigable, sencillo y fácil de manejar para personas con poca experiencia en el ámbito de la computación. El software va dirigido a empresas o personas particulares que deseen brindar una mayor seguridad a su información en el momento de enviarla en un medio como Internet.

3.4.2 Equipo necesario.

- ✓ Sistema operativo Microsoft Windows 98 o posterior.
- ✓ Equipo con procesador Pentium III o superior.
- ✓ Lector de CD-ROM.
- ✓ 30 MB libres en el disco.
- ✓ Tarjeta de video SVGA con resolución 1024 X 768.

3.4.3 Hardware utilizado para el desarrollo de HECOBI. HECOBI se desarrolló en 2 equipos con la siguiente configuración:

- ✓ Procesador Pentium III

- ✓ Memoria RAM de 256 MB
- ✓ Disco Duro de 40 GB
- ✓ Unidad CD-RW de 52X
- ✓ Tarjeta de sonido de 32 bits
- ✓ Monitor SVGA 15”
- ✓ Drive de 3.5 “

3.4.4 Software utilizado para el desarrollo de HECOBI.

- **Lenguaje de programación**: Se utilizó Borland Delphi 7.0 para el desarrollo de las estructuras lógicas y el diseño grafico de pantallas. El entorno de programación de Delphi 7.0 contiene varias herramientas de desarrollo de gran potencia que permiten de una manera clara construir, ejecutar y administrar programas. Delphi es un lenguaje de programación orientado a objetos, donde se escribe un programa que responde a las acciones del usuario: elegir un comando, hacer clic en una ventana, mover el ratón, etc., que se ejecutan a raíz de eventos iniciados por él mismo.

- **Microsoft Office**: utilizado para la edición de los archivos de teoría.

- **IconWorkshop**: para el diseño, creación y adecuación de los diferentes

elementos gráficos que componen a HECOBI

- **Microsoft Internet Explorer**: utilizado en la fase de investigación.

3.5 DISEÑO COMUNICACIONAL

Incluye el estudio de los elementos de la interacción y comunicación visual que contribuyen a mejorar el desempeño de los usuarios en el uso de la herramienta.

3.5.1 Dispositivos de entrada y salida. Los dispositivos de entrada y salida necesarios para la interacción con el programa son los utilizados comúnmente, el ratón y el teclado como dispositivos de entrada y el monitor como dispositivo de salida.

3.5.2 Zonas de comunicación entre el usuario y el programa. La figura 20 muestra las diferentes zonas de comunicación.

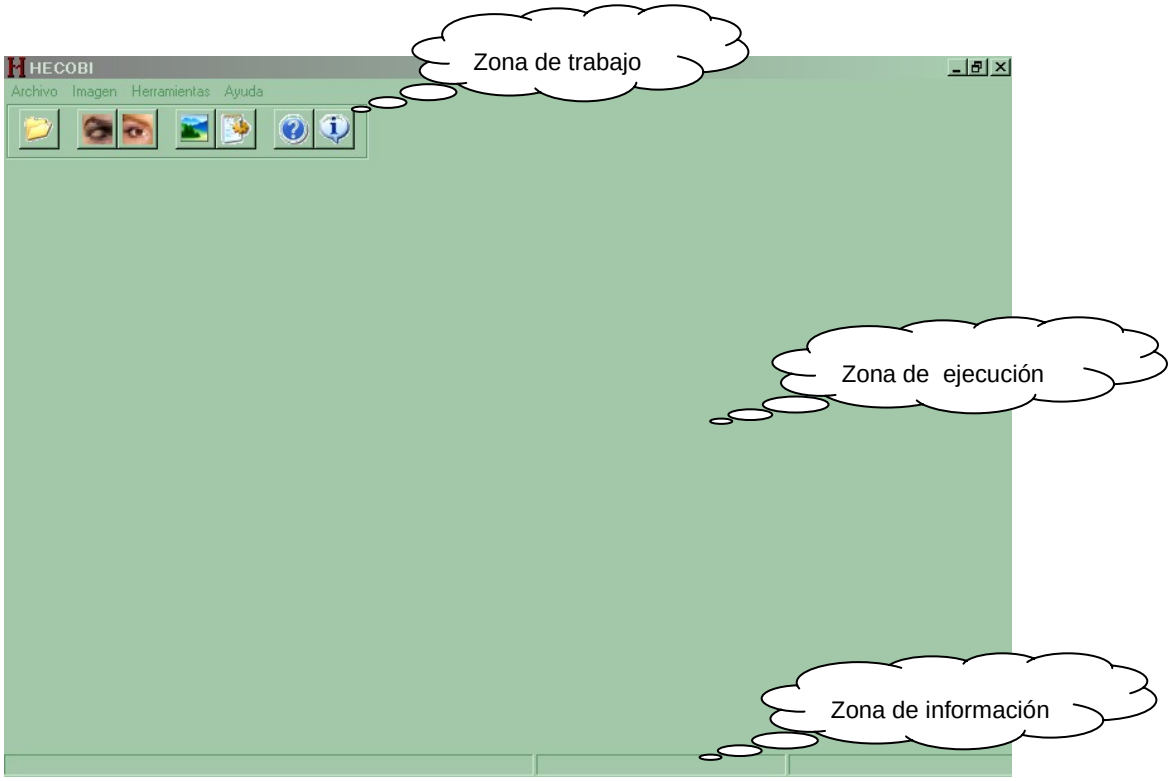
✓ Zona de trabajo: está ubicada en la parte superior de la pantalla, permite al

usuario navegar por el software.

- ✓ Zona de información: se encuentra ubicada en la parte inferior, con ella el usuario podrá obtener información acerca de los procesos de operación del software.

- ✓ Zona de ejecución: ocupa la mayor parte de la pantalla y en ella se desarrollan las diferentes actividades planteadas por el programa.

Figura 20. Zonas de comunicación.



3.5.3 Elementos de las diferentes zonas de comunicación.

- **Menús.** En su gran mayoría son gráficos y serán activados por el ratón dando un clic, dichos menús permitirán que el alumno avance a su propio ritmo.

- **Textos.** Serán desplegados en párrafos cortos, que suministren una idea clara pero concisa, su duración en la pantalla será controlada por el usuario.

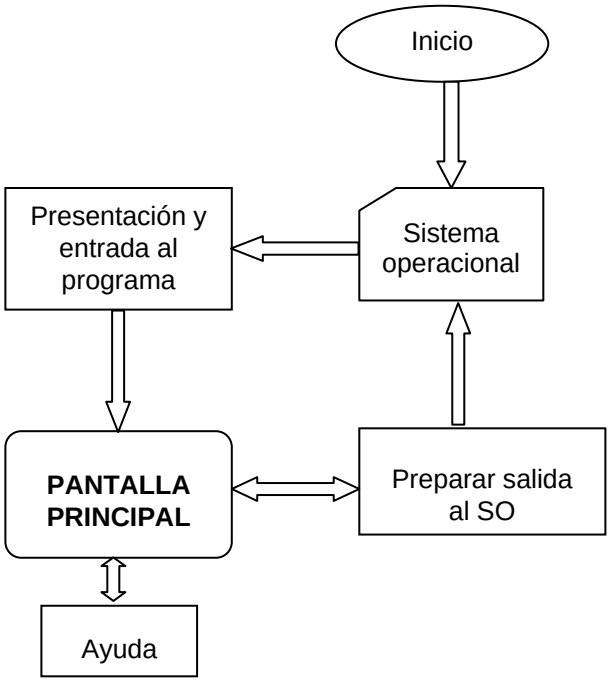
- **Mensajes.** El manejo de mensajes contribuye a mejorar la calidad de la herramienta y reduce significativamente la frustración de los usuarios cuando aparecen errores.

3.6 DISEÑO COMPUTACIONAL

3.6.1 Funciones de apoyo para el usuario. Ofrece explicación sobre el sistema, ayudas de contenido, teoría como base para aprender, opción de abandono y selección de opciones por medio del teclado y del ratón.

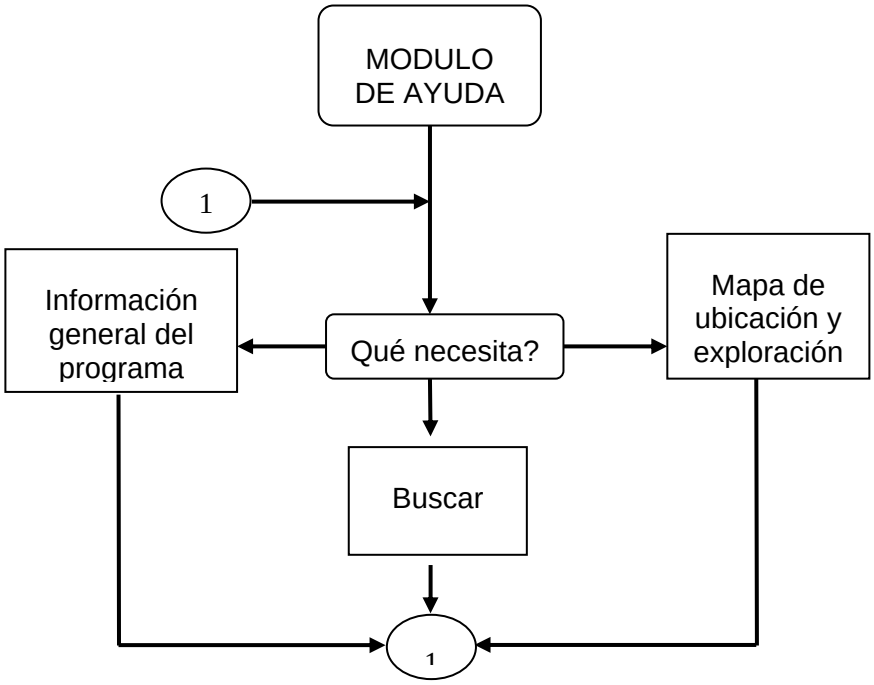
3.6.2 Estructura lógica para la interacción con el software. La estructura lógica se representa en forma modular y se expresa mediante diagramas de flujo. La figura 21 muestra el diagrama de transición de los diferentes estados que pueden estar activos en el programa.

Figura 21. Estructura lógica.



3.6.3 Módulo de Ayuda. Los usuarios encontrarán en éste modulo las respuestas correspondientes al uso y manejo del software, así como la explicación de las herramientas utilizadas en el transcurso de la interacción con éste.

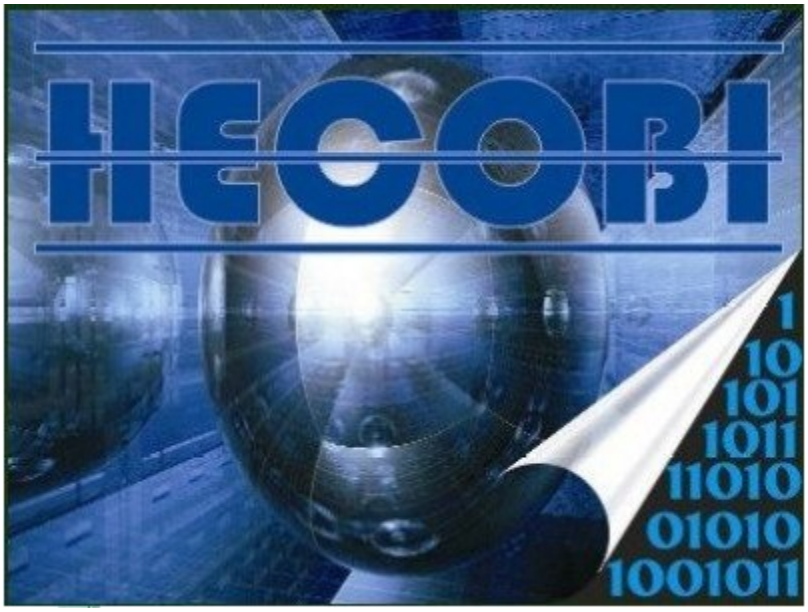
Figura 22. Diagrama de flujo módulo de ayuda.



3.6.4 Descripción Funcional de la Interfaz de HECOBI.

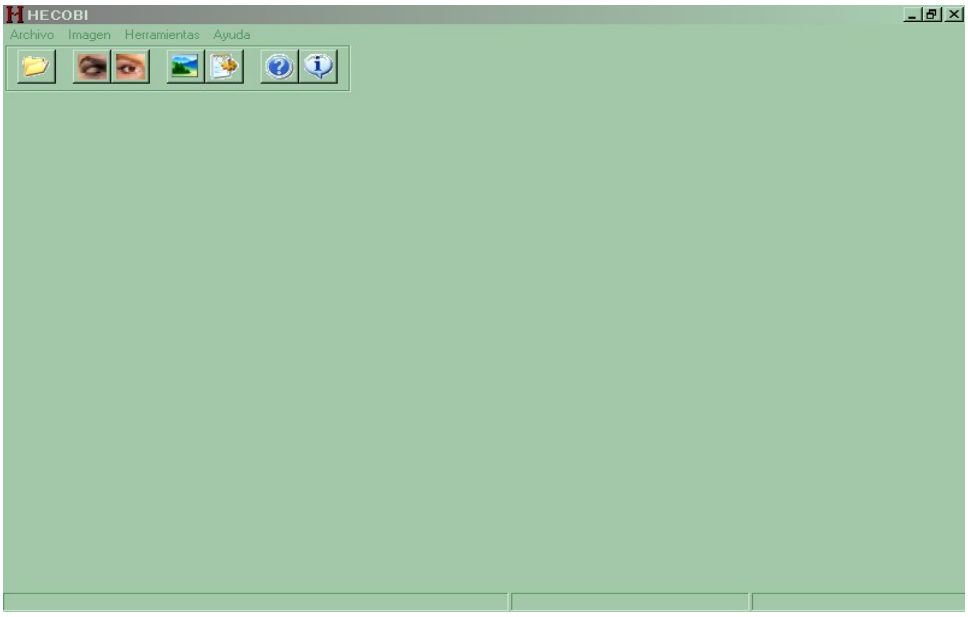
3.6.4.1 Inicio de HECOBI. Al tener acceso a HECOBI aparece una pantalla de presentación, la cual contempla el nombre del software. Esta pantalla sólo permanece unos segundos activa, como se observa en la figura 23.

Figura 23. Incio de HECOBI.



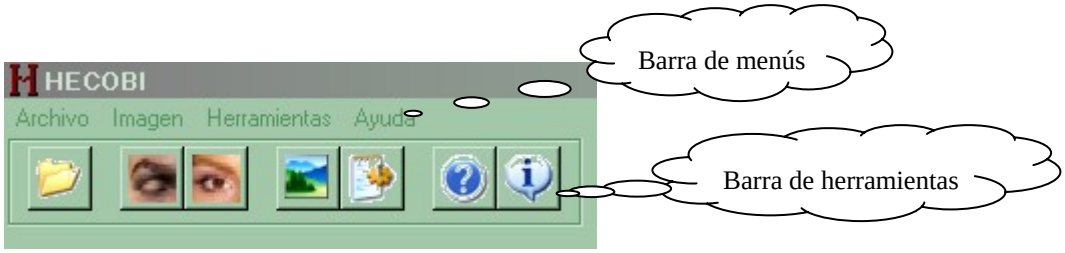
3.6.4.2 Pantalla Principal. La finalidad de esta pantalla es que el usuario podrá acceder al sistema sin ningún problema con solo dar clic en el respectivo vínculo. La pantalla principal de HECOBI se observa en la figura 24.

Figura 24. Pantalla principal de HECOBI.



3.6.4.3 Barra de Menús y barra de herramientas. En la barra de menú se encuentran una serie de opciones que le permiten al usuario interactuar con el software esteganográfico. La barra de herramientas contiene las principales funciones de HECOBI: abrir imágenes, utilidades del software, métodos esteganográficos, acceder a la ayuda del uso del software, ver la ventana de créditos y salir de la aplicación. En la figura 25 se observa la barra de menús y la barra de herramientas de HECOBI.

Figura 25. Barra de menús y barra de herramientas.



Los botones de la barra de herramientas se describen a continuación:

Abre un archivo de imagen BMP de 24 bits

Oculta información en un archivo de imagen BMP de 24 bits

Revela información oculta en un archivo BMP de 24 bits

Convierte un archivo de imagen JPEG, GIF o PNG al formato BMP de 24 bits

Calcula un resumen MD5 a un archivo de cualquier formato

Uso de la Ayuda

Información acerca del Grupo de desarrollo de HECOBI

3.6.4.4 Menú Archivo. Las opciones del menú archivo se observan en la figura 26.

Figura 26. Menú archivo.

- **Submenú Abrir.** Este menú nos permite seleccionar una imagen BMP de 24 bits, la cual utilizaremos para esconder nuestra información (archivo portador).

- **Submenú Cerrar.** El usuario puede cerrar la imagen activa al hacer clic en este submenú.

- **Submenú Salir.** El usuario puede finalizar la herramienta HECOBI con este submenú. Al hacer clic en el submenú salir, aparece en mensaje preguntando

al usuario si está seguro de salir, con las opciones “No” para regresar y “Sí” para cerrar la aplicación.

3.6.4.5 Menú Imagen. Las opciones del menú ayuda se observan en la figura 27.

Figura 27. Menú Imagen.

- **Submenú Ocultar.** Este submenú es la parte principal de la aplicación, permite ocultar un mensaje de texto o un archivo de cualquier tipo en una imagen BMP de 24 bits. Para conocer acerca de cómo Ocultar Información ver el Anexo C.

- **Submenú Revelar.** Permite Revelar un mensaje de texto o un archivo de cualquier tipo en una imagen BMP de 24 bits. Para conocer acerca de cómo Revelar Información ver el Anexo C.

3.6.4.6 Menú Herramientas. Las opciones del menú herramientas se observan en la figura 28.

Figura 28. Menú herramientas.

- **Submenú Convertir Imagen.** Permite al usuario convertir algunos formatos de imágenes al BMP de 24 bits:

  PNG a BMP

  GIF a BMP

  BMP a BMP

  JPEG a BMP

- **Submenú MD5.** Permite al usuario calcular el MD5 de cualquier tipo de archivo existente.

3.6.4.7 Menú Ayuda. Las opciones del menú ayuda se observan en la figura 29.

Figura 29. Menú Ayuda.

- **Submenú Ayuda de HECOBI.** Los usuarios encontrarán en éste submenú las respuestas correspondientes al uso y manejo de HECOBI, así como la explicación de las herramientas utilizadas en el transcurso de la interacción con el software.

- **Submenú Acerca de.** Aquí se encontrará la información referente acerca del Grupo de Desarrollo del proyecto HECOBI.

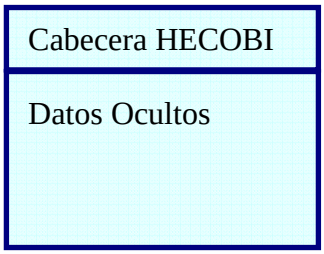
4. MODELO ESTEGANOGRÁFICO EMPLEADO

Para el desarrollo del modelo esteganográfico a emplear en el Trabajo de Grado, se ha creado un formato que lleva el nombre de la herramienta (HECOBI), el cual contiene la información necesaria del proceso esteganográfico.

4.1 FORMATO HECOBI

El formato HECOBI se define como la estructura básica que presenta una imagen a la cual se le ha aplicado esteganografía con el software HECOBI; este formato se encuentra en los datos de la imagen[18]. La figura 30 muestra la estructura básica del formato HECOBI.

Figura 30. Formato HECOBI.



[18] Ver Figura 5. Estructura básica formato BMP.

4.1.1 Cabecera HECOBI. Contiene los variables necesarias para realizar el proceso esteganográfico. Consta de los siguientes elementos**:**

- **HECOBIXYZ**
- **Tamaño de la contraseña**
- **Contraseña**
- **Tamaño del archivo o mensaje**
- **Extensión del archivo**

A continuación se describen los elementos que conforman la Cabecera HECOBI:

4.1.1.1 HECOBIXYZ. Es una marca que nos permite determinar si la imagen ha sido utilizada con la herramienta HECOBI. X, Y, Z son valores numéricos que determinan alguna información como se muestra en la tabla 9.

Tabla 9. Descripción de *HECOBIXYZ*.

Variable	***Valor***	***Descripción***
X	0	La contraseña está guardada en la imagen
	1	La contraseña no está guardada en la imagen

Y	0	El método estaganográfico utilizado es LSB
	1	El método estaganográfico utilizado es 2LSB
	2	El método estaganográfico utilizado es DCT
Z	0	La información oculta es un archivo de cualquier tipo
	1	La información oculta es un mensaje de texto

HECOBIXYZ necesita 72 bits para ser almacenado. La tabla 10 muestra la forma de almacenar la variable HECOBIXYZ, según el método esteganográfico empleado.

Tabla 10. Almacenamiento de *HECOBIXYZ*.

Método	***# bytes requeridos de la imagen***	***# bits almacenados en cada byte de la imagen***
LSB	72	1
2LSB	36	2
DCT	72	1

4.1.1.2 Tamaño de la contraseña. Almacena el tamaño de la contraseña si va incluida en la estego imagen, necesita 8 bits para ser almacenada. La tabla 11 muestra la forma de almacenar *el tamaño de la contraseña*, según el método esteganográfico empleado.

Tabla 11. Almacenamiento del *Tamaño de la Contraseña*.

Método	*# bytes requeridos de la imagen*	*# bits almacenados en cada byte de la imagen*
LSB	8	1
2LSB	4	2
DCT	8	1

4.1.1.3 Contraseña. La contraseña ofrece la opción de incluirla o no en la imagen; presenta valores alfanuméricos, mínimo 6 y máximo 16 caracteres.

Tabla 12. Almacenamiento de la *Contraseña*.

Método	*# bytes requeridos de la imagen*	*# bits almacenados en cada byte de la imagen*
LSB	Mínimo 48 - Máximo 128	1
2LSB	Mínimo 24 - Máximo 64	2
DCT	Mínimo 48 - Máximo 128	1

4.1.1.4 Tamaño del archivo o mensaje de texto. Almacena el valor del tamaño (bytes) del mensaje de texto o archivo a ocultar en la imagen. Necesita 32 bits para ser almacenado.

Tabla 13. Almacenamiento del *Tamaño del archivo o mensaje*.

Método	*# bytes requeridos de la imagen*	*# bits almacenados en cada byte de la imagen*
LSB	32	1
2LSB	16	2
DCT	32	1

4.1.1.5 Extensión del archivo. Si la información a ocultar es de tipo archivo, se guardará en esta variable su extensión (por ejemplo, doc, exe, pdf, htm). Su tamaño es de 3 bytes.

Tabla 14. Almacenamiento de la *Extensión del Archivo*.

Método	*# bytes requeridos de la imagen*	*# bits almacenados en cada byte de la imagen*
LSB	24	1
2LSB	12	2
DCT	24	1

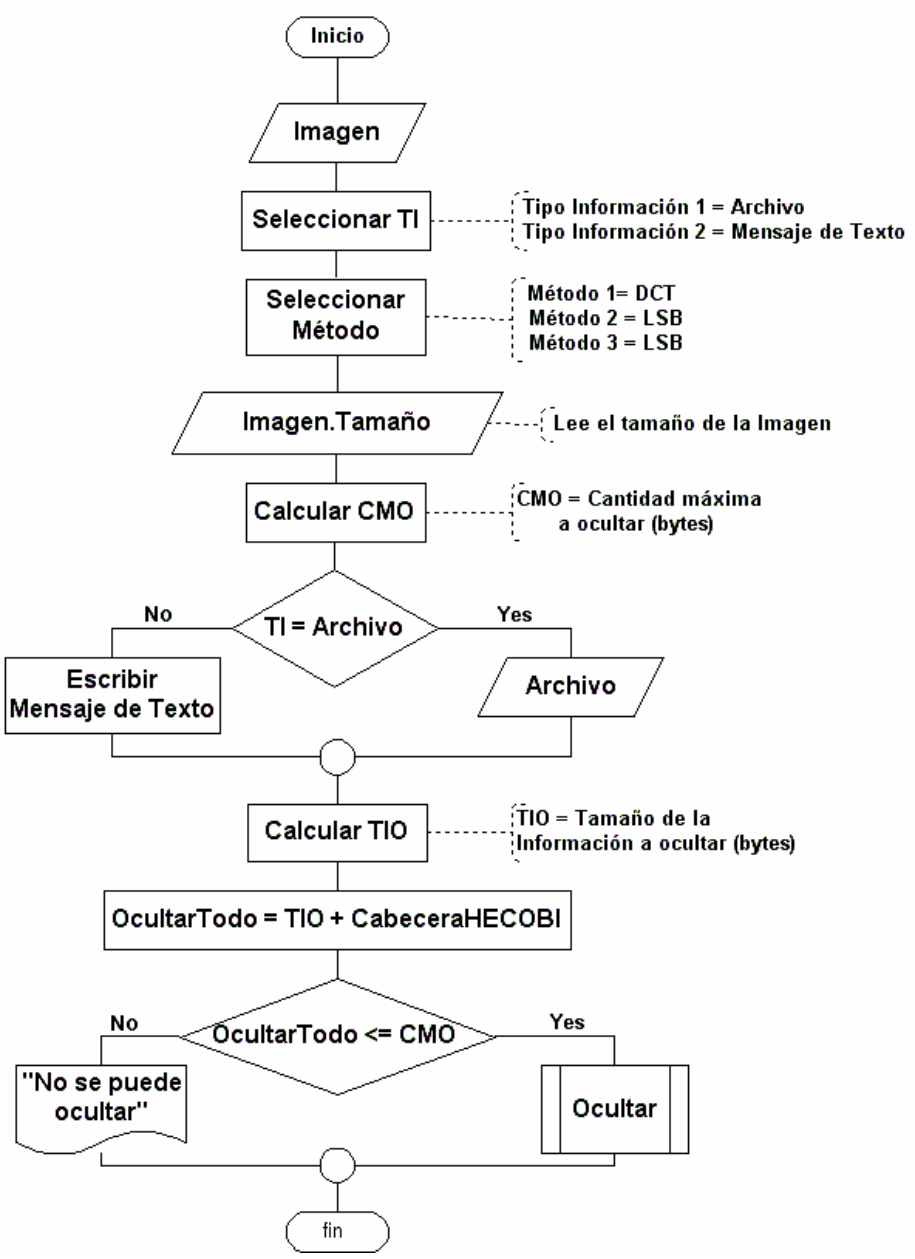
4.1.2 Datos ocultos. Es la información en sí a ocultar que ha sido previamente encriptada con el algoritmo AES en la refinación del prototipo.

4.2 ALGORITMOS ESTEGANOGRÁFICOS

A continuación se presentan los algoritmos principales implementados en la elaboración de HECOBI.

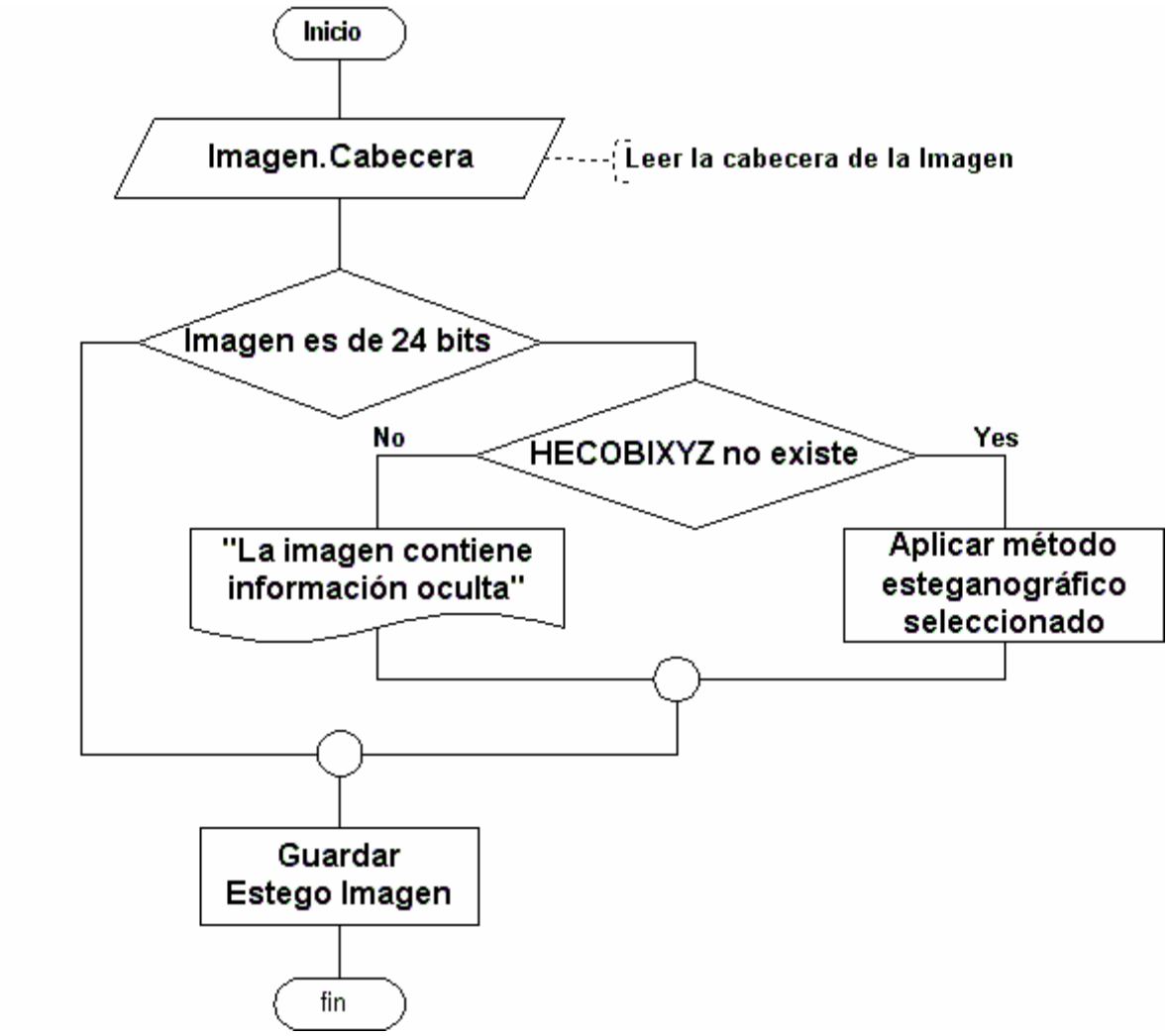
4.2.1 Algoritmo Ocultar Información. En la Figura 31 se puede observar el algoritmo general utilizado por los tres métodos (LSB, 2LSB y DCT) para iniciar el ocultamiento de información en un imagen BMP de 24 bits.

Figura 31. Algoritmo Ocultar Información.


Inicio
Imagen
Seleccionar TI
Tipo Información 1 = Archivo
Tipo Información 2 = Mensaje de Texto
Seleccionar Método
Método 1= DCT
Método 2 = LSB
Método 3 = LSB
Imagen.Tamaño
Lee el tamaño de la Imagen
Calcular CMO
CMO = Cantidad máxima a ocultar (bytes)
No
TI = Archivo
Yes
Escribir Mensaje de Texto
Archivo
Calcular TIO
TIO = Tamaño de la Información a ocultar (bytes)
OcultarTodo = TIO + CabeceraHECOBI
No
OcultarTodo <= CMO
Yes
"No se puede ocultar"
Ocultar
fin

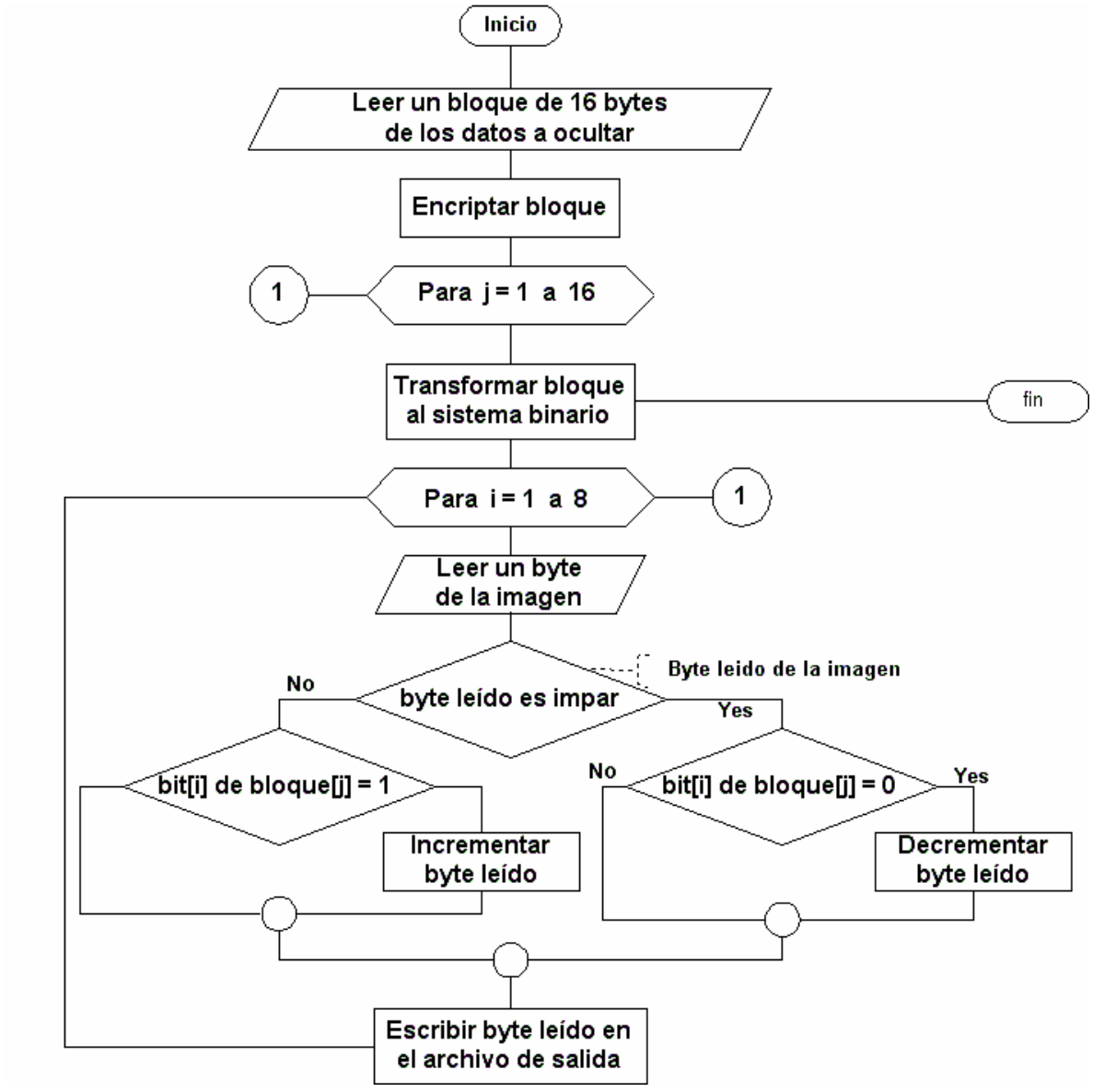
4.2.1.1 Algoritmo procedimiento ocultar.

Figura 32. Algoritmo Procedimiento Ocultar.



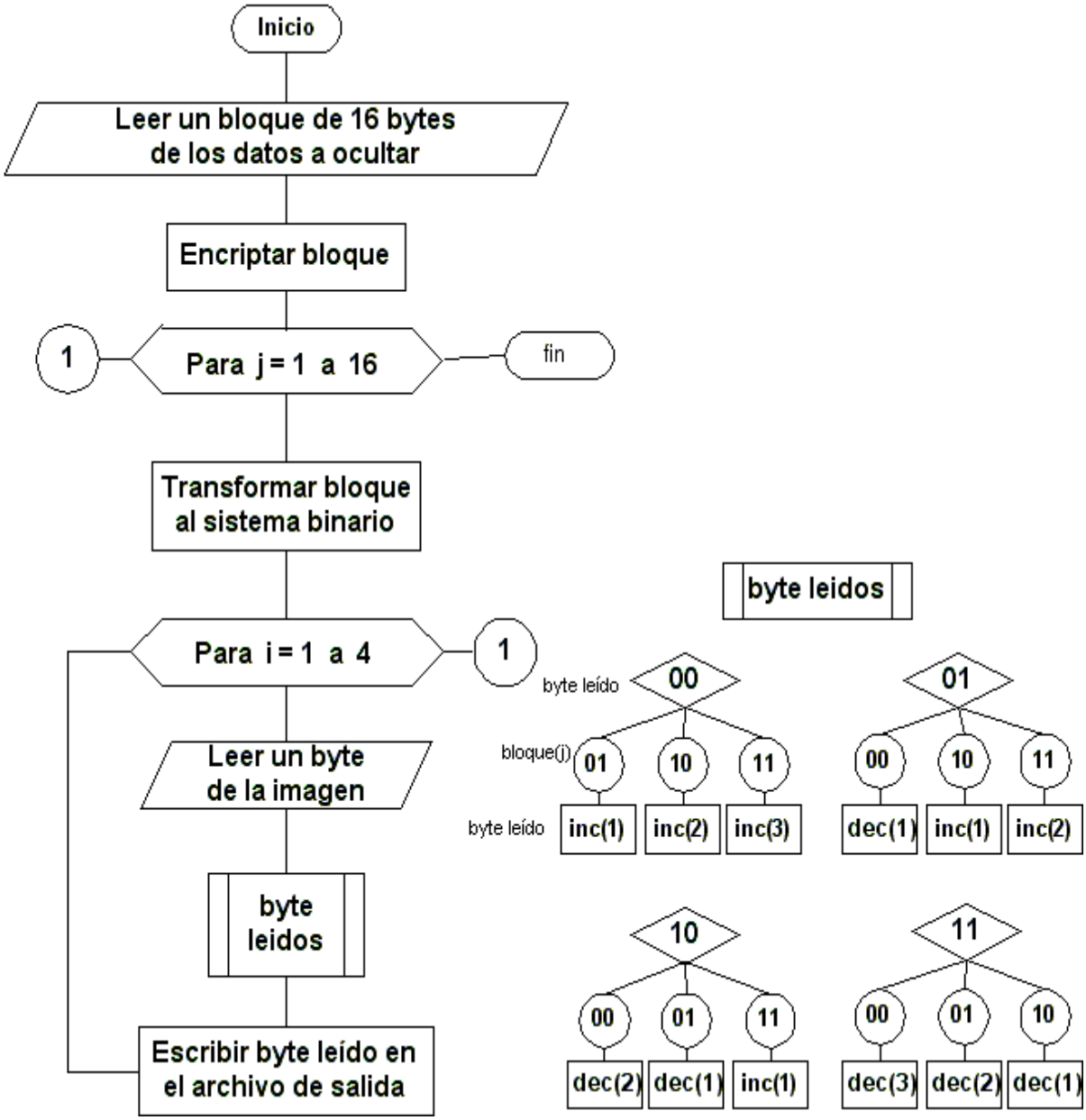
4.2.1.2 Algoritmo Ocultar con Método LSB.

Figura 33. Algoritmo Ocultar con Método LSB.



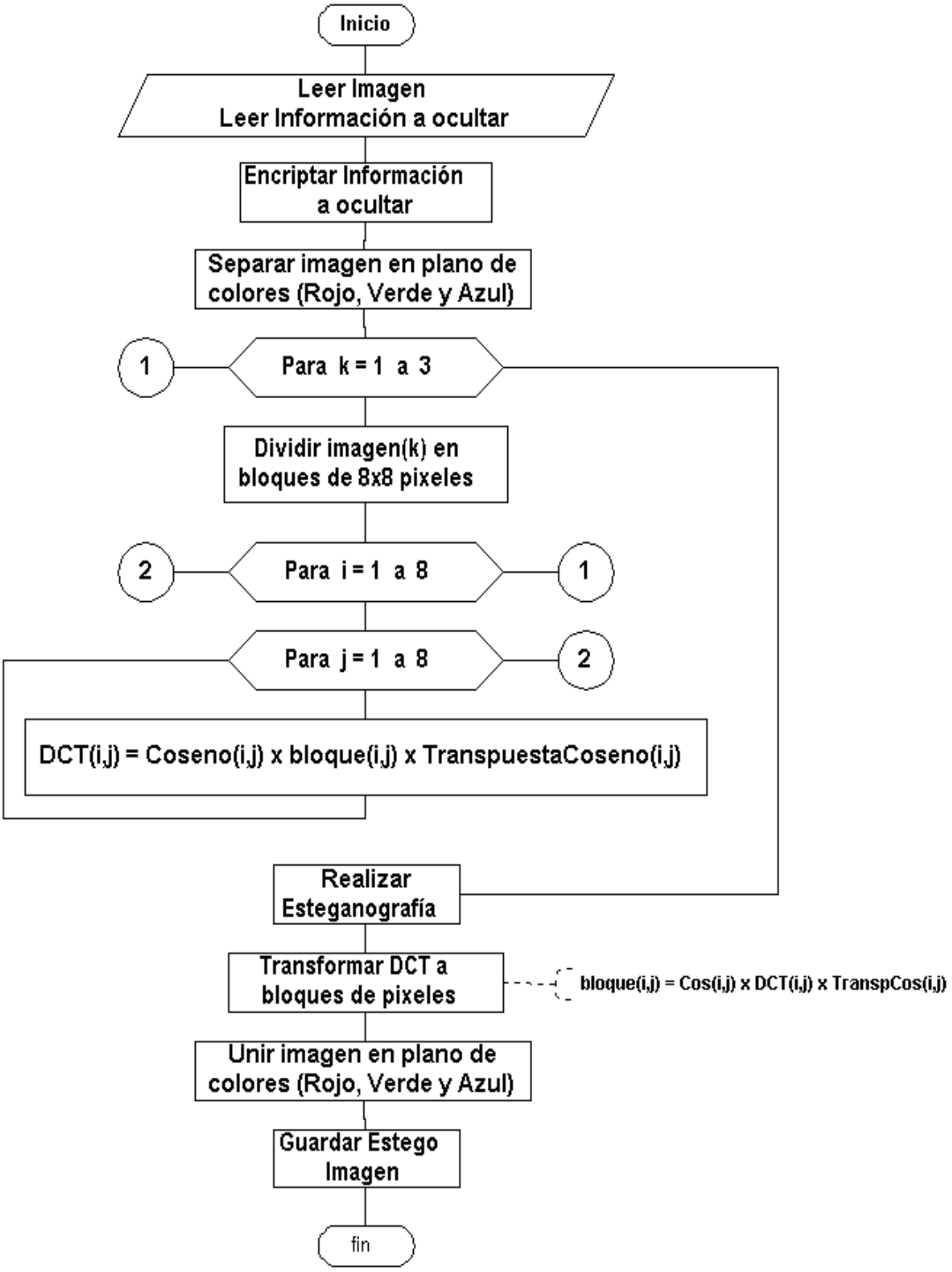
4.2.1.3 Algoritmo Ocultar con Método 2LSB.

Figura 34. Algoritmo Ocultar con Método 2LSB.



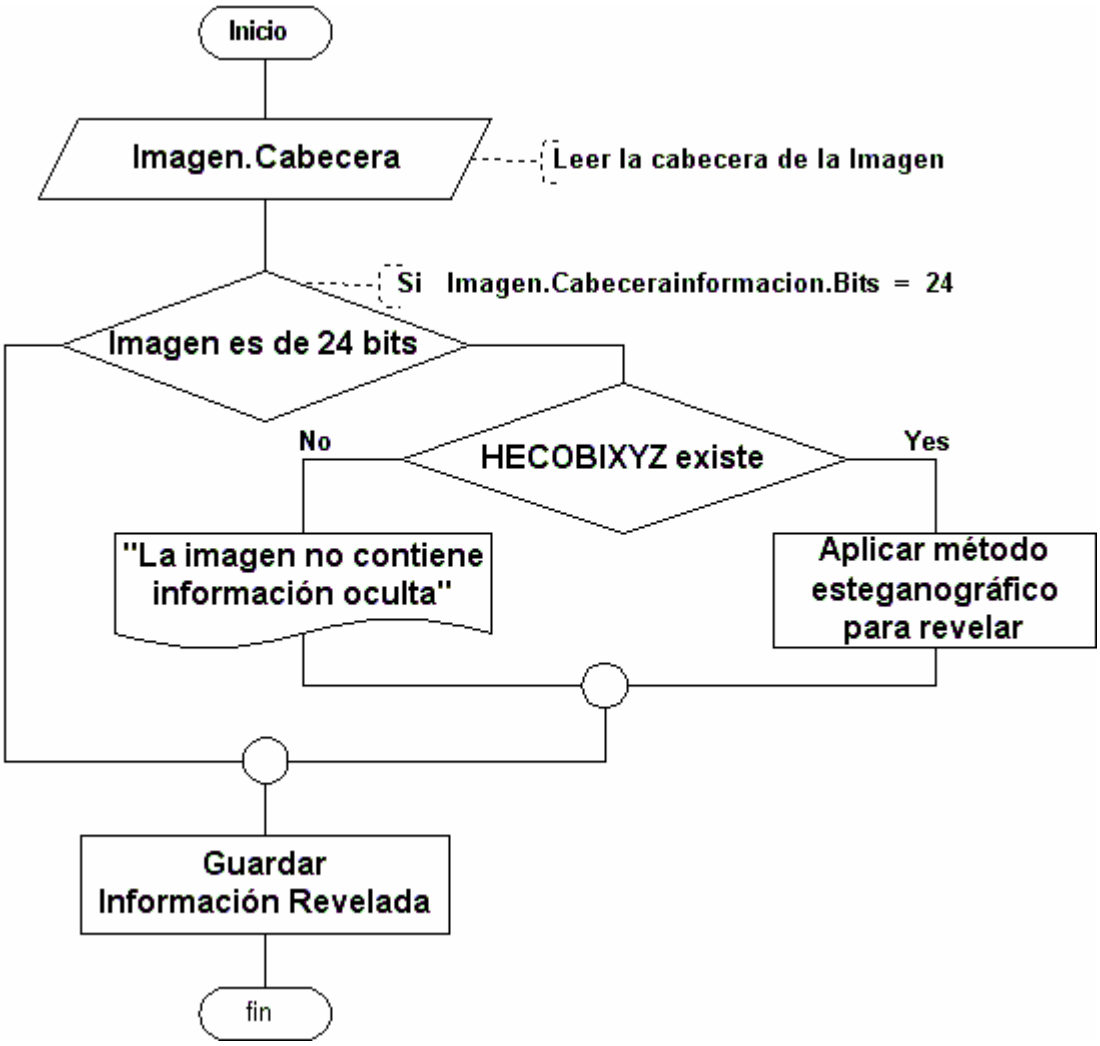
4.2.1.4 Algoritmo Ocultar con Método DCT.

Figura 35. Algoritmo Ocultar con Método DCT.



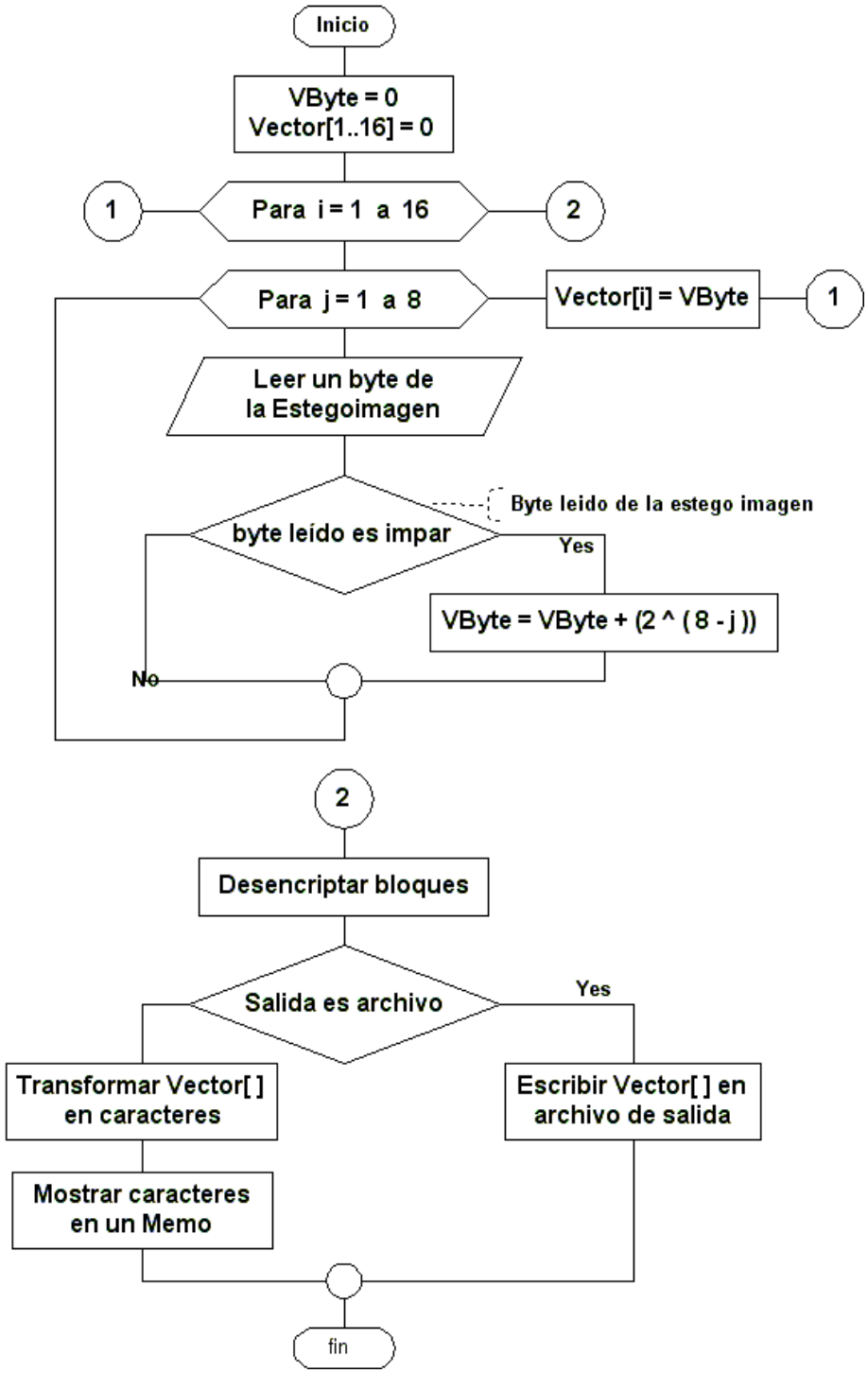
4.2.2 Algoritmo Revelar información.

Figura 36. Algoritmo Revelar Información.



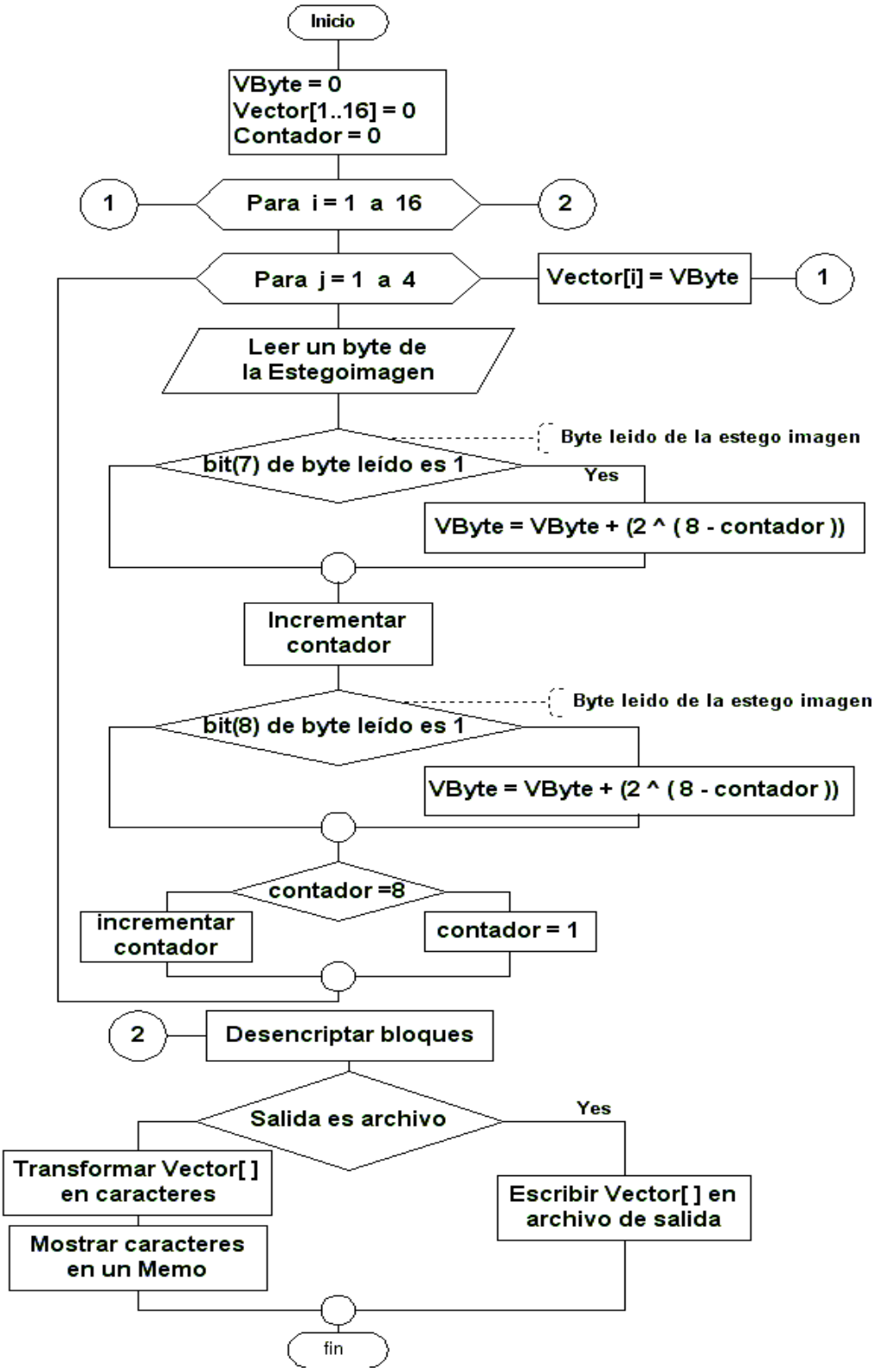
4.2.2.1 Algoritmo Revelar con Método LSB.

Figura 37. Algoritmo Revelar con Método LSB.



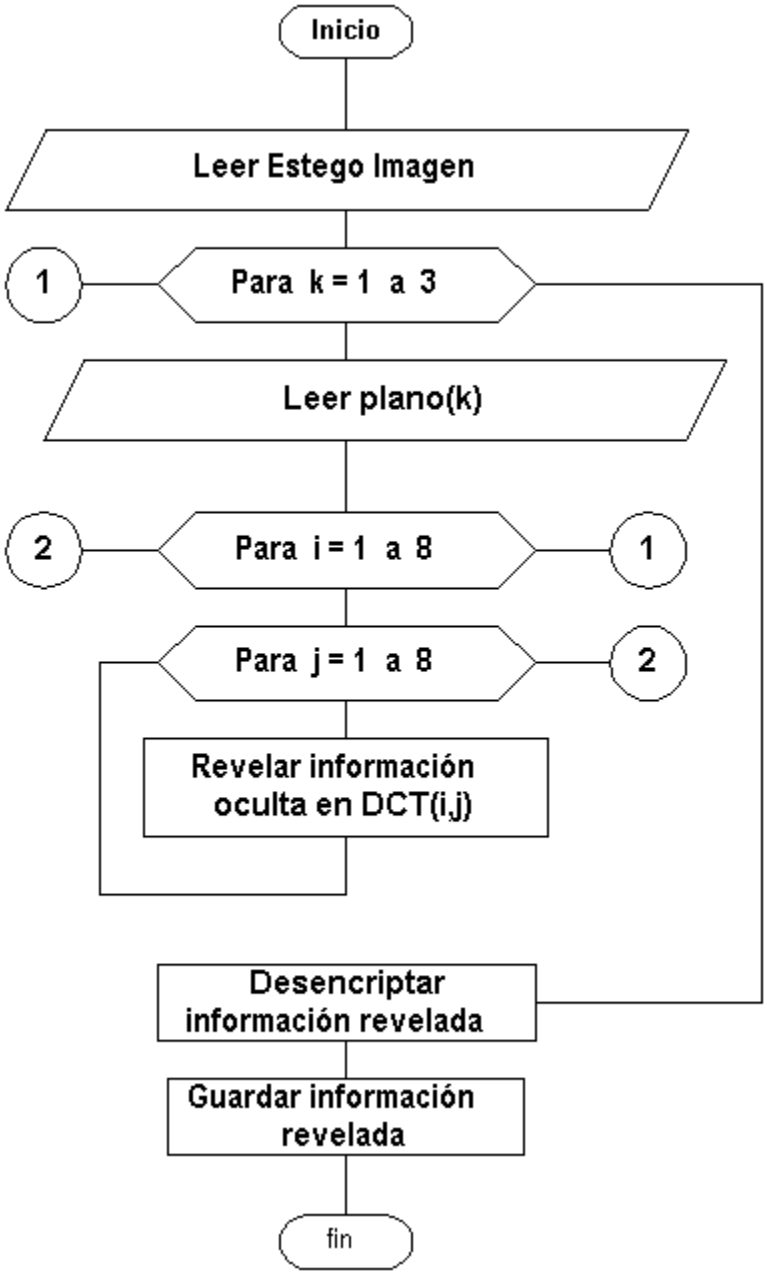
4.2.2.2 Algoritmo Revelar con Método 2LSB.

Figura 38. Algoritmo Revelar con Método 2LSB.



4.2.2.3 Algoritmo Revelar con Método DCT.

Figura 39. Algoritmo Revelar con Método DCT.



5. PRUEBAS

Se realizaron pruebas de almacenamiento de información proveniente tanto de un archivo como de un mensaje de texto con 2 imágenes diferentes. Los resultados de los tiempos empleados por la herramienta para los procesos de ocultar y revelar fueron los siguientes:

Prueba 1. Se tomó una imagen del "Cristo Petrolero de Barrancabermeja", tiene un tamaño de 8.01 Megabytes (8.400.054 bytes); se ocultó un archivo de 995 Kilobytes de tamaño, los tiempos en segundos de los diferentes métodos tanto guardando la contraseña dentro del archivo se muestran en la tabla 15.

Tabla 15. Tiempo ocultando archivo en imagen "Cristo Petrolero".

Método	***Con Contraseña***	***Sin Contraseña***
LSB	29.2	28.9
2LSB	29.4	29.3
DCT	29.6	29.5

Los tiempos del proceso de revelado (en segundos) para los diferentes métodos se muestran en la tabla 16.

Tabla 16. Tiempo revelando archivo en imagen "Cristo Petrolero".

Método	*Con Contraseña*	*Sin Contraseña*
LSB	15.5	15.6
2LSB	10.6	11
DCT	15.7	15.8

Las imágenes resultantes de guardar el archivo de 995 Kilobytes de tamaño con los diferentes métodos se observan en la figura 40.

Figura 40. Imagen "Cristo Petrolero de Barrancabermeja" con Archivo.

Método LSB

Método 2LSB

Método DCT

En la misma imagen se ocultó un mensaje de aproximadamente 10000 caracteres, los tiempos (segundos) de los diferentes métodos se muestran en la tabla 17.

Tabla 17. Tiempo ocultando mensaje en imagen "Cristo Petrolero".

Método	*Con Contraseña*	*Sin Contraseña*
LSB	31.1	32.5
2LSB	31.7	31.4
DCT	31.8	32.7

Los tiempos (segundos) del proceso de revelado del mensaje, usando los diferentes métodos se muestran en la tabla 18.

Tabla 18. Tiempo revelando mensaje en imagen "Cristo Petrolero".

Método	*Con Contraseña*	*Sin Contraseña*
LSB	38.1	38.4
2LSB	38.5	39.9
DCT	39.1	39.4

Las imágenes resultantes de guardar el mensaje de aproximadamente 10000 caracteres con los diferentes métodos se observan en la figura 41.

Figura 41. Imagen "Cristo Petrolero de Barrancabermeja" con Mensaje.

Método LSB

Método 2LSB

Método DCT

Prueba 2. Se tomó una imagen de las desaparecidas Torres Gemelas que tiene un tamaño de 2.03 Megabytes (2.138.166 bytes), se ocultó un archivo de 220 Kilobytes de tamaño, los tiempos (segundos) de los diferentes métodos se muestran en la tabla 19.

Tabla 19. Tiempo ocultando archivo en imagen "Torres Gemelas".

Método	***Con Contraseña***	***Sin Contraseña***
LSB	7.6	7.4
2LSB	7.7	7.7
DCT	8.3	8.3

Los tiempos del proceso de revelado (en segundos) para los diferentes métodos se muestran en la tabla 20.

Tabla 20. Tiempo revelando archivo en imagen "Torres Gemelas".

Método	***Con Contraseña***	***Sin Contraseña***
LSB	3	3
2LSB	1.9	1.9
DCT	2.9	2.9

Las imágenes resultantes de guardar el archivo de 220 Kilobytes de tamaño con los diferentes métodos se observan en la figura 42.

Figura 42. Imagen "Torres Gemelas" con Archivo.

Método LSB Método 2LSB Método DCT

En la misma imagen se ocultó un mensaje de aproximadamente 10000 caracteres,

los resultados (en segundos) se observan en la tabla 21.

Tabla 21. Tiempo ocultando mensaje en imagen "Torres Gemelas".

Método	*Con Contraseña*	*Sin Contraseña*
LSB	10.6	10.5
2LSB	10.7	10.7
DCT	11.3	11.5

Los tiempos del proceso de revelado (en segundos) para los diferentes métodos fueron se muestran en la tabla 22.

Tabla 22. Tiempo revelando mensaje en imagen "Torres Gemelas".

Método	*Con Contraseña*	*Sin Contraseña*
LSB	40.3	40.3
2LSB	40.3	39.9
DCT	40.9	40.9

Las imágenes resultantes de guardar el mensaje de aproximadamente 10000 caracteres con los diferentes métodos se observan en la figura 43.

Figura 43. Imagen “Torres Gemelas” con Mensaje.

Método LSB Método 2LSB Método DCT

6. CONCLUSIONES

El uso del formato BMP de 24 bits permitió una mayor capacidad de almacenamiento para ocultar información, en comparación con otros formatos de imágenes conocidos.

Sé permitió la conversión de imágenes con formatos JPEG, GIF, PNG y BMP al formato BMP de 24 bits, dando al usuario la posibilidad de escoger como base los formatos de archivos de imágenes más utilizados actualmente.

El método 2LSB permitió almacenar mayor información que el LSB y DCT, pero con un mayor grado de distorsión. El método DCT aunque computacionalmente más costoso, permitió mayor seguridad que los otros dos métodos, al transformar la imagen del dominio RGB al dominio de los coeficientes DCT. Para el ojo humano fue imposible detectar la distorsión en las imágenes al aplicar estos métodos.

Utilizando los algoritmos AES y MD5 se logró obtener confidencialidad e integridad de la información. Esto permitió el desarrollo de un sistema muy confiable y seguro, por la combinación de técnicas criptográficas y esteganográficas.

La esteganografía es una opción diferente de brindar seguridad a información que viaja a través de un medio no seguro como el Internet, sobre todo para aquellos países donde es restringido el uso de sistemas criptográficos.

7. RECOMENDACIONES

1. Para un mejor desempeño de la herramienta se recomienda utilizar imágenes con gran variedad de colores, preferiblemente paisajes y fotografías. Además no se debe utilizar varias veces la misma imagen, debido a que terceras personas pueden realizar ataques por comparación.

2. A nivel del software HECOBI, se propone su extensión y mantenimiento, añadiendo mejoras como se describen a continuación:

- ✓ Estudiar otros tipos de archivos de imágenes como el formato PNG, GIF o JPEG, con el propósito de implementar nuevos módulos que permitan realizar esteganografía en estos formatos.

- ✓ Investigar y proponer algoritmos de compresión de datos con el fin de ocultar mayor cantidad de información en las imágenes.

- ✓ Realizar mejoras al sistema criptográfico de HECOBI, implementando un módulo de Intercambio de claves con el fin de garantizar la seguridad al momento de divulgar información relevante.

- ✓ Investigar nuevos métodos esteganográficos preferiblemente más efectivos que los utilizados en este proyecto, para su posterior implementación.

- ✓ Estudiar técnicas que permitan detectar la presencia de información oculta en Estego Imágenes generadas por el software HECOBI.

3. A nivel de la Universidad, promover la creación de un Grupo de Investigación que permita a los estudiantes de la EISI de la UIS realizar proyectos como:

- ✓ Esteganografía en archivos de música.
- ✓ Esteganografía en archivos de video.
- ✓ Marcas de agua.
- ✓ Técnicas de Detección de la información oculta en archivos de imágenes.
- ✓ Aplicación de Esteganografía en Bases de Datos (Hospitales, Fuerzas Militares, Registraduría Nacional, etc.).

ANEXOS

ANEXO A. ALGORITMO AES

Elementos de AES. AES es un algoritmo que se basa en aplicar un número determinado de rondas a un valor intermedio que se denomina *estado*. Dicho estado puede representarse mediante una matriz rectangular de bytes, que posee cuatro filas, y Nb columnas. Así, si nuestro bloque tiene 160 bits, Nb será igual a 5.

Tabla 23. Ejemplo de matriz de estado con Nb = 5 (160 bits).

a0,0	a 0,1	a 0,2	a 0,3	a 0,4
a 1,0	a 1,1	a 1,2	a 1,3	a 1,4
a 2,0	a 2,1	a 2,2	a 2,3	a 2,4
a 3,0	a 3,1	a 3,2	a 3,3	a 3,4

La llave tiene una estructura análoga a la del estado, y se representará mediante una tabla con cuatro filas y Nk columnas. Si nuestra clave tiene, por ejemplo, 128 bits, Nk será igual a 4.

Tabla 24. Ejemplo de clave con Nk = 4 (128 bits).

K0,0	K 0,1	K 0,2	K 0,3
K 1,0	K 1,1	K 1,2	K 1,3
K 2,0	K 2,1	K 2,2	K 2,3

En algunos casos, tanto el estado como la clave se consideran como vectores de registros de 32 bits, estando cada registro constituido por los bytes de la columna correspondiente, ordenados de arriba a abajo. El bloque que se pretende cifrar o descifrar se traslada directamente byte a byte sobre la matriz de estado, siguiendo la secuencia a0,0, a1,0, a2,0, a3,0, a0,1 . . ., y análogamente, los bytes de la clave se copian sobre la matriz de clave en el mismo orden, a saber, k0,0, k1,0, k2,0, k3,0, k0,1 . . ..

Siendo B el bloque que queremos cifrar, y S la matriz de estado, el algoritmo AES con n rondas queda como sigue:

1. Calcular K0, K1, . . .Kn subclaves a partir de la clave K.
2. $B \oplus K0 = S$
3. Para i = 1 hasta n hacer

4. Aplicar ronda i-ésima del algoritmo con la subclave Ki.

Puesto que cada ronda es una sucesión de funciones invertibles, el algoritmo de descifrado consistirá en aplicar las inversas de cada una de las funciones en el orden contrario, y utilizar los mismos Ki que en el cifrado, sólo que comenzando por el último.

ANEXO B. ALGORITMO MD5

Descripción del algoritmo MD5. Siendo m un mensaje de b bits de longitud, en primer lugar se alarga m hasta que su longitud sea exactamente 64 bits inferior a un múltiplo de 512. El alargamiento se lleva a cabo añadiendo un uno seguido de tantos ceros como sea necesario. En segundo lugar, se añaden 64 bits con el valor de b, empezando por el byte menos significativo. De esta forma tenemos el mensaje como un número entero de bloques de 512 bits, y además le hemos añadido información sobre su longitud.

Antes de procesar el primer bloque del mensaje, se inicializan cuatro registros de 32 bits con los siguientes valores hexadecimales, según el criterio little endian (el byte menos significativo queda en la dirección de memoria más baja).

A = 67452301

B = EFCDAB89

C = 98BADCFE

D = 10325476

Posteriormente comienza el lazo principal del algoritmo, que se repetirá para cada

bloque de 512 bits del mensaje. En primer lugar copiaremos los valores de A, B ,C y D en otras cuatro variables, a, b, c y d. Luego definiremos las siguientes cuatro funciones:

F(X, Y, Z) = (X ^ Y) v ((¬X) ^ Z)

G(X, Y, Z) = (X ^ Z) v ((Y ^ (¬Z))

H(X, Y, Z) = X ⊕ Y ⊕ Z

I(X, Y, Z) = Y ⊕ (X v (¬Z))

Ahora representaremos por mj el j-´esimo bloque de 32 bits del mensaje m (de 0 a 15), y definiremos otras cuatro funciones:

FF(a, b, c, d, mj , s, ti) representa a = b + ((a + F(b, c, d) + mj + ti) <<< s)

GG(a, b, c, d, mj , s, ti) representa a = b + ((a + G(b, c, d) + mj + ti) <<< s)

HH(a, b, c, d, mj , s, ti) representa a = b + ((a + H(b, c, d) + mj + ti) <<< s)

II(a, b, c, d, mj , s, ti) representa a = b + ((a + I(b, c, d) + mj + ti) <<< s)

donde la función a <<< s representa desplazar circularmente la representación binaria del valor a s bits a la izquierda, con reentrada. Las 64 operaciones que se

realizan en total quedan agrupadas en cuatro rondas.

Primera Ronda:

FF(a, b, c, d, m0, 7,D76AA478)

FF(d, a, b, c, m1, 12,E8C7B756)

FF(c, d, a, b, m2, 17, 242070DB)

FF(b, c, d, a, m3, 22,C1BDCEEE)

FF(a, b, c, d, m4, 7, F57C0FAF)

FF(d, a, b, c, m5, 12, 4787C62A)

FF(c, d, a, b, m6, 17,A8304613)

FF(b, c, d, a, m7, 22, FD469501)

FF(a, b, c, d, m8, 7, 698098D8)

FF(d, a, b, c, m9, 12, 8B44F7AF)

FF(c, d, a, b, m10, 17, FFFF5BB1)

FF(b, c, d, a, m11, 22, 895CD7BE)

FF(a, b, c, d, m12, 7, 6B901122)

FF(d, a, b, c, m13, 12, FD987193)

FF(c, d, a, b, m14, 17, A679438E)

FF(b, c, d, a, m15, 22, 49B40821)

Segunda Ronda:

GG(a, b, c, d, m1, 5, F61E2562)

GG(d, a, b, c, m6, 9,C040B340)

GG(c, d, a, b, m11, 14, 265E5A51)

GG(b, c, d, a, m0, 20,E9B6C7AA)

GG(a, b, c, d, m5, 5,D62F105D)

GG(d, a, b, c, m10, 9, 02441453)

GG(c, d, a, b, m15, 14, D8A1E681)

GG(b, c, d, a, m4, 20,E7D3FBC8)

GG(a, b, c, d, m9, 5, 21E1CDE6)

GG(d, a, b, c, m14, 9,C33707D6)

GG(c, d, a, b, m3, 14, F4D50D87)

GG(b, c, d, a, m8, 20, 455A14ED)

GG(a, b, c, d, m13, 5, A9E3E905)

GG(d, a, b, c, m2, 9, FCEFA3F8)

GG(c, d, a, b, m7, 14, 676F02D9)

GG(b, c, d, a, m12, 20, 8D2A4C8A)

Tercera Ronda:

HH(a, b, c, d,m5, 4, FFFA3942)

HH(d, a, b, c,m8, 11, 8771F681)

HH(c, d, a, b,m11, 16, 6D9D6122)

HH(b, c, d, a,m14, 23, FDE5380C)

HH(a, b, c, d,m1, 4,A4BEEA44)

HH(d, a, b, c,m4, 11, 4BDECFA9)

HH(c, d, a, b,m7, 16, F6BB4B60)

HH(b, c, d, a,m10, 23,BEBFBC70)

HH(a, b, c, d,m13, 4, 289B7EC6)

HH(d, a, b, c,m0, 11,EAA127FA)

HH(c, d, a, b,m3, 16,D4EF3085)

HH(b, c, d, a,m6, 23, 04881D05)

HH(a, b, c, d,m9, 4,D9D4D039)

HH(d, a, b, c,m12, 11,E6DB99E5)

HH(c, d, a, b,m15, 16, 1FA27CF8)

HH(b, c, d, a,m2, 23,C4AC5665)

Cuarta Ronda:

II(a, b, c, d, m0, 6, F4292244)

II(d, a, b, c, m7, 10, 432AFF97)

II(c, d, a, b, m14, 15,AB9423A7)

II(b, c, d, a, m5, 21, FC93A039)

II(a, b, c, d, m12, 6, 655B59C3)

II(d, a, b, c, m3, 10, 8F0CCC92)

II(c, d, a, b, m10, 15, FFEFF47D)

II(b, c, d, a, m1, 21, 85845DD1)

II(a, b, c, d, m8, 6, 6FA87E4F)

II(d, a, b, c, m15, 10, FE2CE6E0)

II(c, d, a, b, m6, 15, A3014314)

II(b, c, d, a, m13, 21, 4E0811A1)

II(a, b, c, d, m4, 6, F7537E82)

II(d, a, b, c, m11, 10,BD3AF235)

II(c, d, a, b, m2, 15, 2AD7D2BB)

II(b, c, d, a, m9, 21,EB86D391)

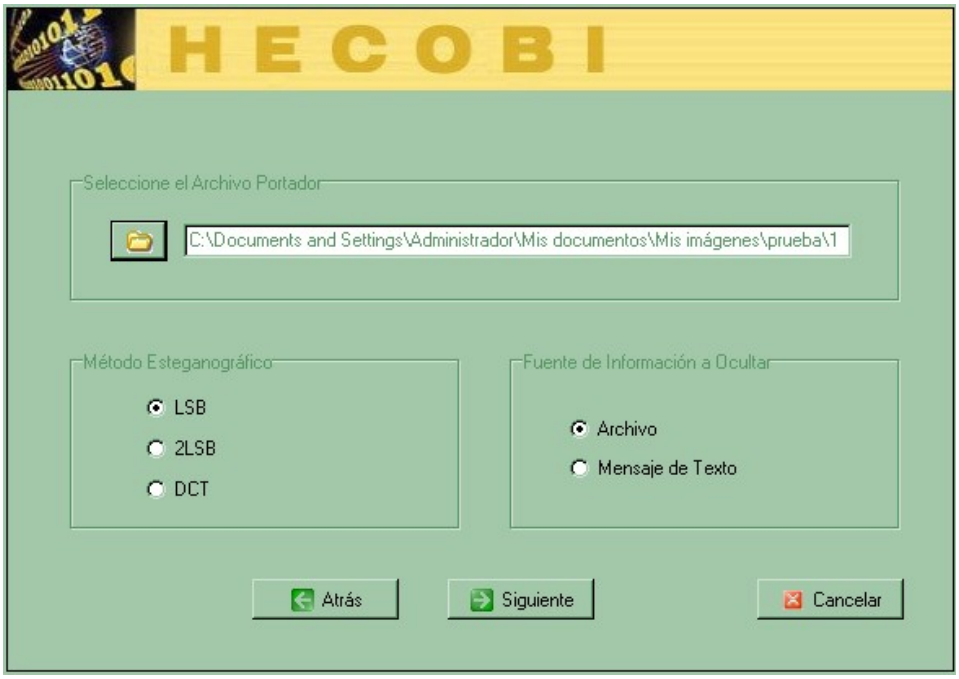
Finalmente, los valores resultantes de a, b, c y d son sumados con A, B, C y D, quedando listos para procesar el siguiente bloque de datos. El resultado final del algoritmo es la concatenación de A, B, C y D. Las constantes ti empleadas en cada paso son la parte entera del resultado de la operación $2^{32} \cdot \text{abs}(\sin(i))$, estando i representado en radianes.

ANEXO C. USANDO HECOBI

CÓMO OCULTAR UN ARCHIVO

Dar clic en el botón  de la barra de herramientas o seleccione *Ocultar* del menú *Imagen.*

Aparece la siguiente ventana:



Método Esteganográfico: Seleccione el método esteganográfico que desee utilizar:

DCT: Transformada Discreta Coseno

LSB2: Segundo Bit Menos Significativo

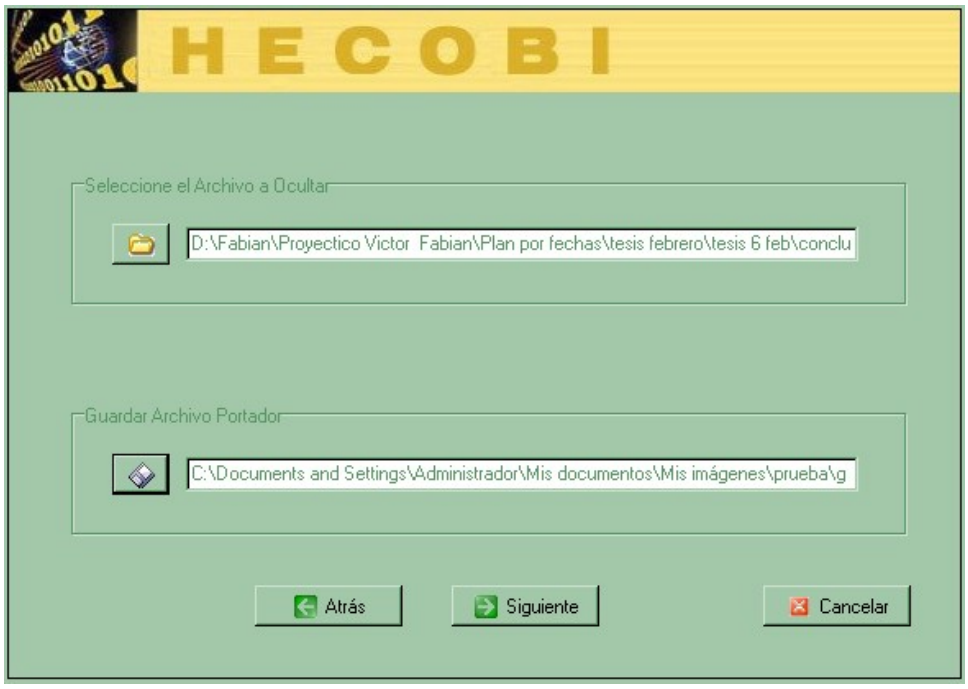
LSB: Primer Bit Menos Significativo

Fuente de *Información a Ocultar*: Dar clic en *Archivo* para ocultar un archivo de cualquier tipo de formato.

***Seleccione el Archivo Portador*:** Dar clic en el botón  para buscar una imagen BMP de 24 bits (archivo portador), donde se ocultará la información secreta.

Al seleccionar la imagen, aparece en la pantalla la imagen escogida:

Dar clic en el botón *Siguiente,* aparece la siguiente ventana:



Seleccione el Archivo a Ocultar: Dar clic en el botón  para buscar el archivo que usted quiere ocultar.

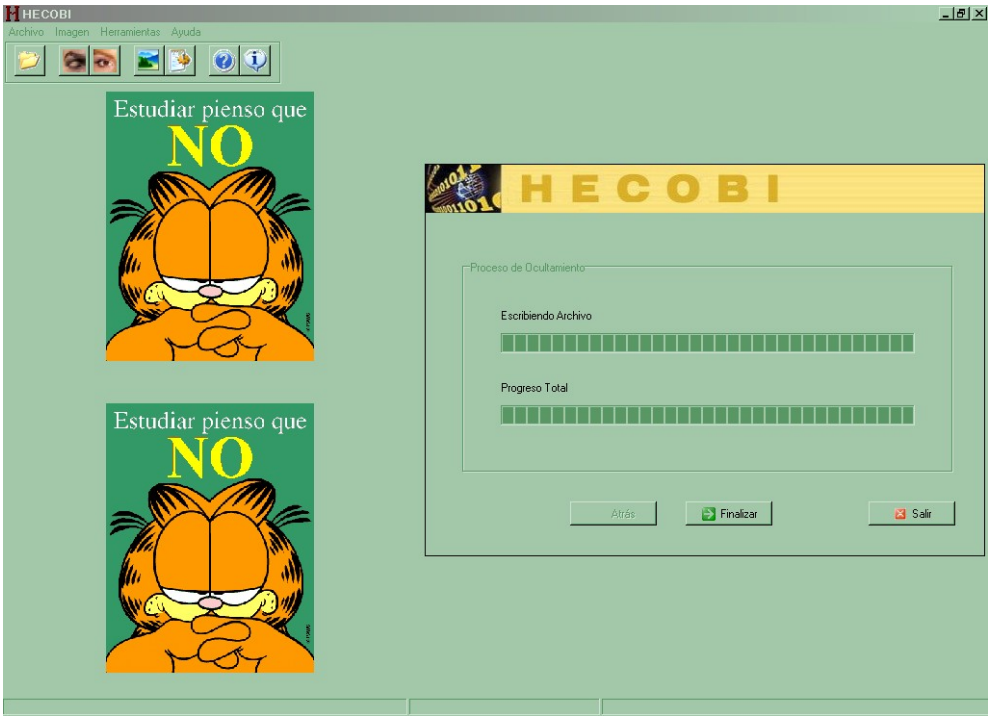
Guardar Archivo Portador: Dar clic en *la caja de texto o en* el botón  para asignarle un nombre a la imagen que llevará el archivo oculto (Estego Imagen).

Dar clic en el botón *Siguiente,* aparece la siguiente ventana:

Permitir Guardar Contraseña: Confirme la selección en la caja de chequeo para guardar la contraseña.

Contraseña (Mínimo 6 caracteres): Escriba su contraseña en las cajas de texto *Digite la Contraseña* y *Confirme la Contraseña.*

Dar clic en el botón *Siguiente* y luego clic en el botón *Ocultar,* aparece la siguiente ventana:



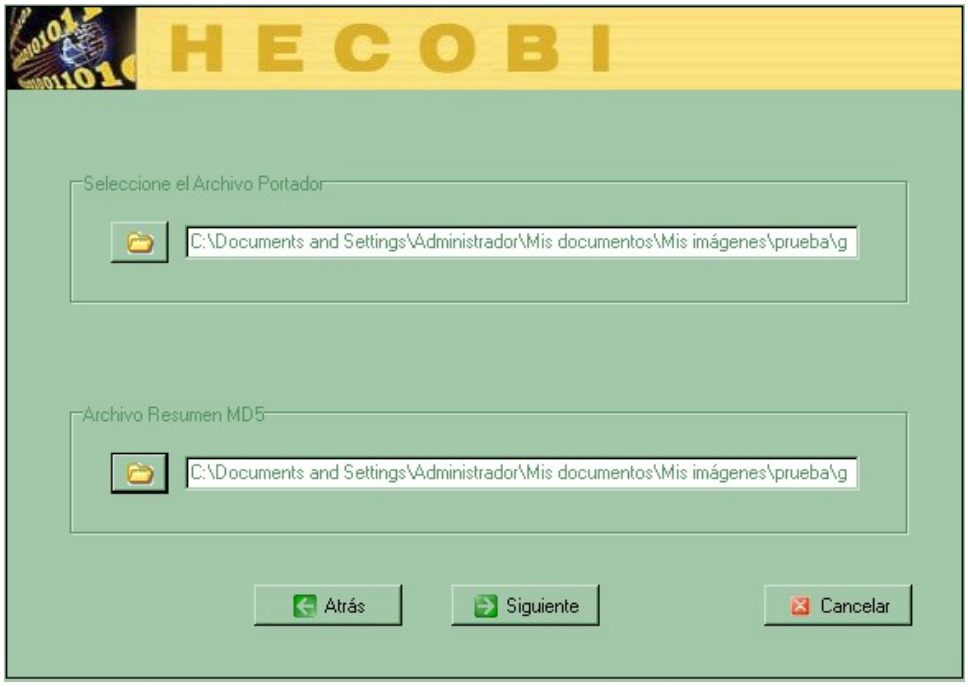
Las barras de progreso indican la evolución del proceso de ocultamiento. HECOBI mostrará la imagen antes y después de ocultar el archivo.

Dar clic en botón *Finalizar* para terminar exitosamente el proceso.

CÓMO REVELAR UN ARCHIVO

Dar clic en el botón  de la barra de herramientas o seleccione *Revelar* del menú *Imagen.*

Aparece la siguiente ventana:



Seleccione el Archivo Portador: dar clic en el botón *Seleccione Archivo Portador* para buscar una imagen BMP de 24 bits (Estego Imagen), la cual contiene el archivo oculto.

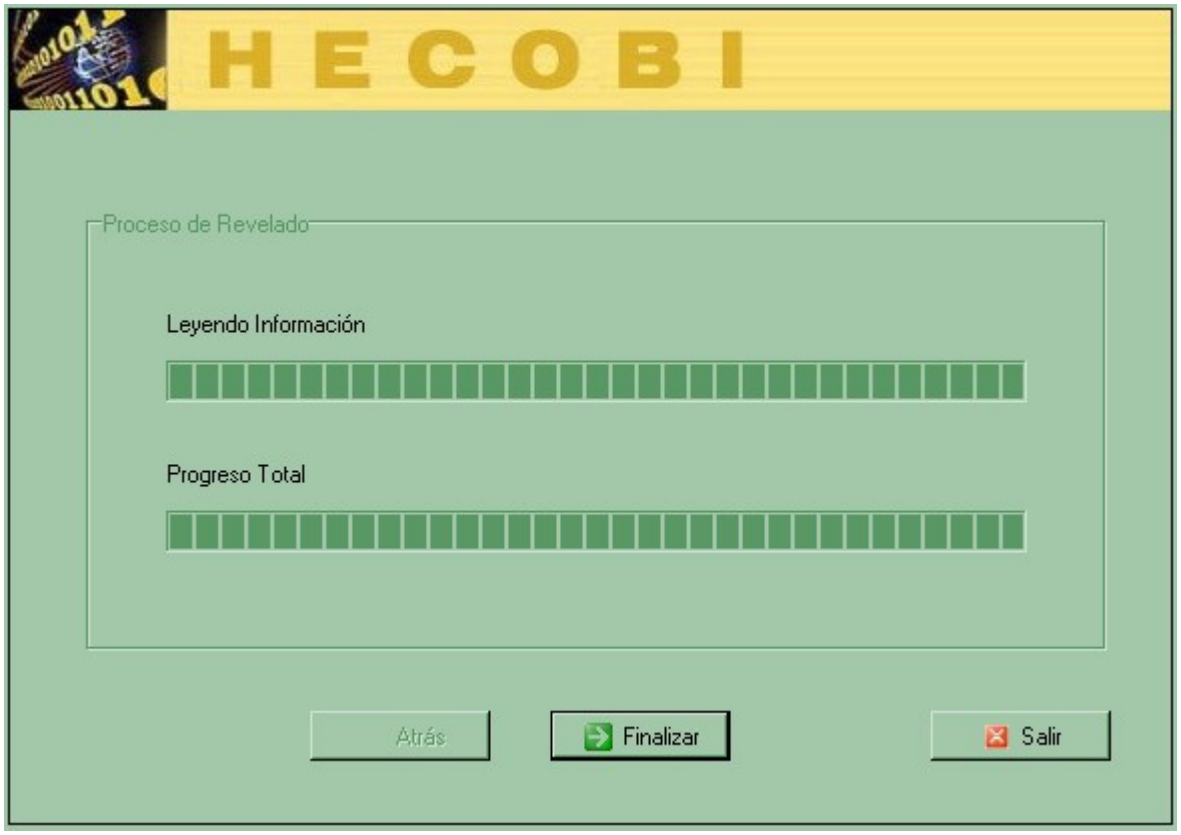
Archivo Verifique MD5: dar clic en el botón *Archivo MD5* para seleccionar un archivo con extensión *.MD5, el cual contiene el MD5 de la Estego Imagen

Dar clic en el botón *Siguiente,* aparece la siguiente ventana:

Guardar Archivo Como: Dar clic en *la caja de texto o en* el botón  para asignarle un nombre al archivo que será revelado.

Contraseña (Mínimo 6 caracteres): Escriba su contraseña en las cajas de texto *Digite la Contraseña* y *Confirme la Contraseña.* Si la contraseña no fue guardada, no se requiere digitarla.

Dar clic en el botón *Siguiente* y luego clic en el botón *Revelar,* aparece la siguiente ventana:



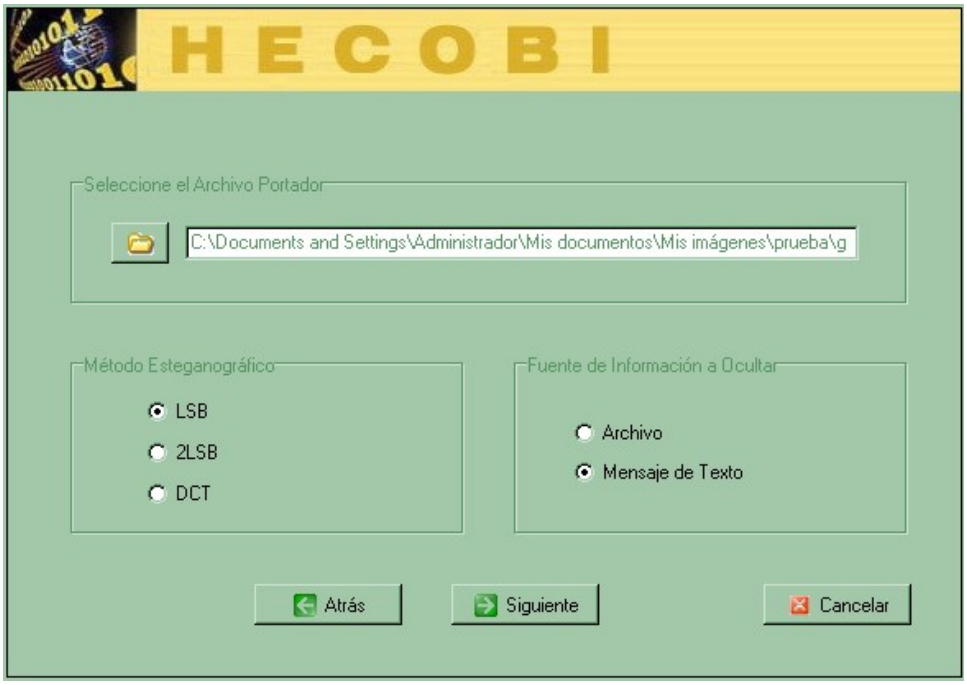
Las barras de progreso indican la evolución del proceso de Revelado.

Dar clic en botón *Finalizar* para terminar exitosamente el proceso.

COMO OCULTAR UN MENSAJE DE TEXTO

Dar clic en el botón  de la barra de herramientas o seleccione *Ocultar* del menú *Imagen.*

Aparece la siguiente ventana:



Método Esteganográfico: Seleccione el método esteganográfico que desee utilizar:

DCT: Transformada Discreta Coseno

LSB2: Segundo Bit Menos Significativo

LSB: Primer Bit Menos Significativo

Fuente de *Información a Ocultar*: Dar clic en Mensaje de Texto.

***Seleccione el Archivo Portador* :** Dar clic en el botón  para buscar una imagen BMP de 24 bits (archivo portador), donde se ocultará la información secreta.

Dar clic en el botón *Siguiente,* aparece la siguiente ventana:

Escriba el Mensaje de Texto en el Memo que aparece en la ventana.

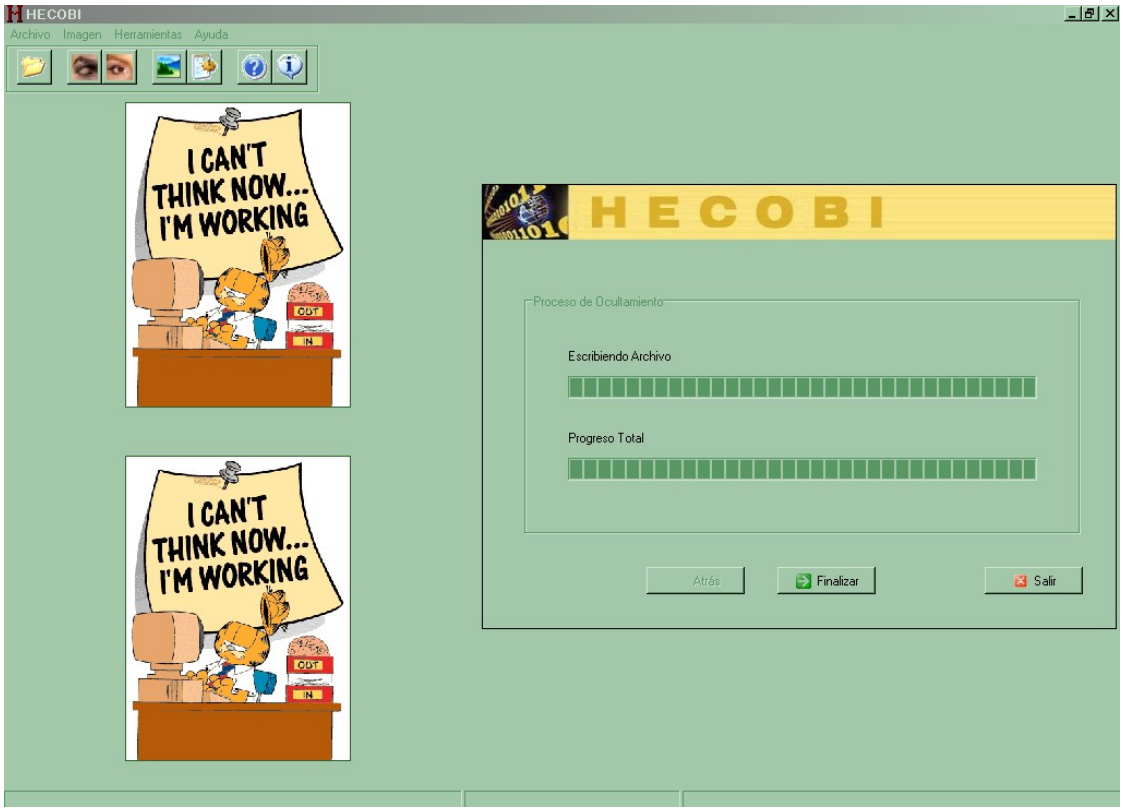
Guardar Archivo Portador: Dar clic en *la caja de texto o en* el botón  para asignarle un nombre a la imagen que llevará el Mensaje de Texto (Estego Imagen).

Dar clic en el botón *Siguiente,* aparece la siguiente ventana:

Permitir Guardar Contraseña: Confirme la selección en la caja de chequeo para guardar la contraseña.

Contraseña (Mínimo 6 caracteres): Escriba su contraseña en las cajas de texto *Digite la Contraseña* y *Confirme la Contraseña.*

Dar clic en el botón *Siguiente* y luego clic en el botón *Ocultar,* aparece la siguiente ventana:



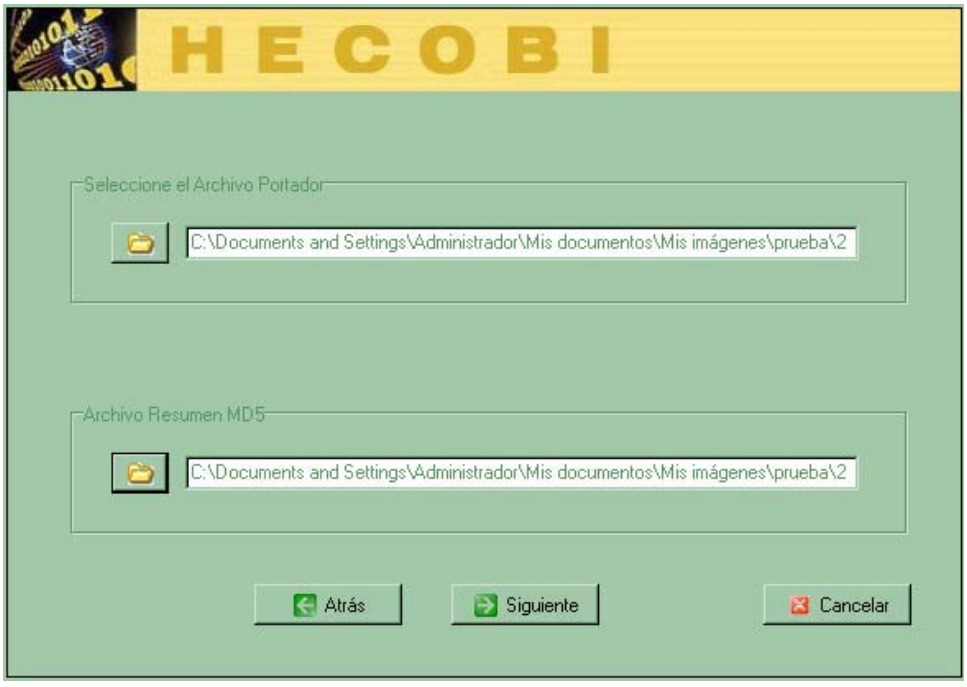
Las barras de progreso indican la evolución del proceso de ocultamiento. HECOBI mostrará la imagen antes y después de ocultar el Mensaje de Texto.

Dar clic en botón *Finalizar* para terminar exitosamente el proceso.

COMO REVELAR UN MENSAJE DE TEXTO

Dar clic en el botón  de la barra de herramientas o seleccione *Revelar* del menú *Imagen*

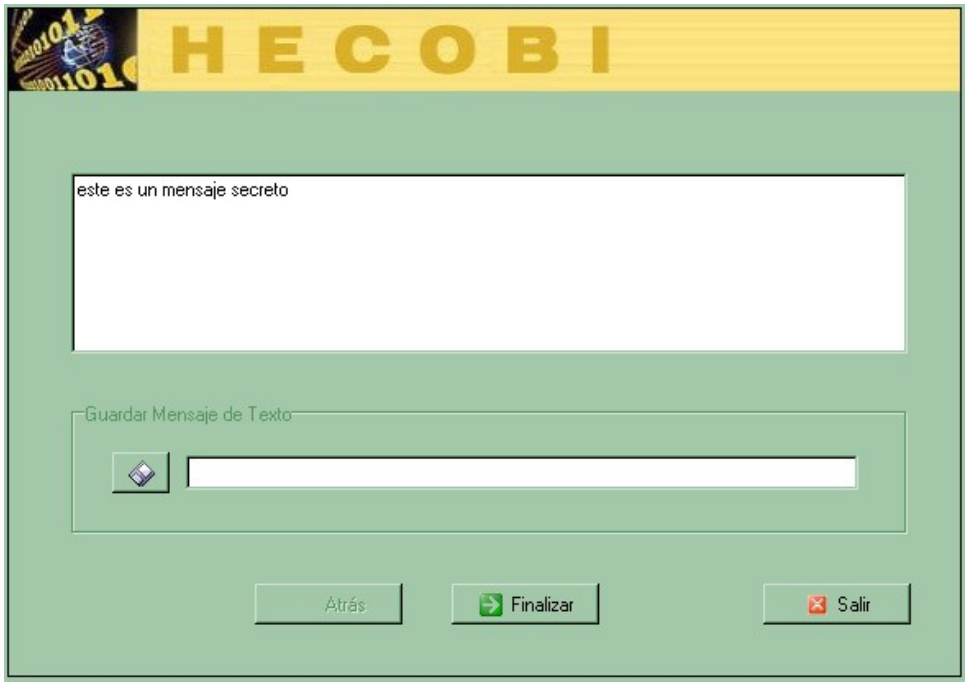
Aparece la siguiente ventana:



Seleccione el Archivo Portador: dar clic en el botón *Seleccione Archivo Portador* para buscar una imagen BMP de 24 bits (Estego Imagen), la cual contiene el Mensaje de Texto oculto.

Verifique MD5: dar clic en el botón *Archivo* para seleccionar un archivo con extensión *.MD5, el cual contiene el MD5 de la Estego Imagen

Dar clic en el botón *Siguiente,* aparece la siguiente ventana:



Aparece el Mensaje de Texto Revelado.

Guardar Mensaje de Texto: dar clic en el botón  si desea guardar el mensaje de texto en un archivo.

Dar clic en botón *Finalizar* para terminar exitosamente el proceso.

BIBLIOGRAFÍA

ANGEL, José de Jesús. AES - Advanced Encryption Standard. 2005, Principiantes. [Citado en octubre de 2005] <URL: http://www. kriptopolis.org >

CHU, Rufeng; YOU, Xinggang; KONG, Xiangwei y BA, Xiaohui. A DCT-based Image Steganographic Method Resisting Statistical Attacks. Proc. IEEE, v 5, 2004. p. 953-956.

ECO, Humberto. Cómo se hace una Tesis - Técnicas y Procedimientos de Estudio, Investigación y Escritura. 23 ed. Barcelona, España: Gedisa, s.f.

GÓMEZ FLÓREZ, Luis Carlos. Guía para el Desarrollo de Proyectos de Grado**.** Bucaramanga, Colombia: UIS, 2003.

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACIÓN. Compendio de Tesis y otros Trabajos de Grado. Bogotá, Colombia: ICONTEC, 2005. 112 p.

JACOBSON, Ivar; BOOCH, Grady y RUMBAUGH, James. El Proceso Unificado de Desarrollo de Software. 1 ed. España: Addison Wesley, 2000.

JOHNSON, Neil F. y JAJODIA, Sushil. Exploring Steganography: Seeing the Unseen. Proc. IEEE, v 31, n° 2, 1998. p. 26-34.

LUCENA LÓPEZ, Manuel José. Criptografía y Seguridad en Computadores. 2 ed. España: Universidad Jadén, 1999.

NAKAMURA, Daisuke; OGIHARA, Takeshi y YOKOYA, Naokazu. Data Embedding into Pictorial Images with Less Distortion Using Discrete Cosine Transform. Universidad Kobe. Proc. ICPR, 1996. p. 675-679.

PIATTINI, Mario G., CALVO MANZANO, José. Análisis y Diseño de Aplicaciones Informáticas de Gestión: Una Perspectiva de Ingeniería del Software. Madrid: Alfaomega Ra-Ma, 2004.

PRESUMAN, Roger. Ingeniería Del Software. Un enfoque práctico. 5 ed. España: McGraw Hill, 2002.

PROVOS, Niels y HONEYVAN, Meter. Hide and Seek: An Introduction to Steganography. Universidad de Michigan. [Citado en octubre de 2005] <URL: http://www.jjtc.com/neil>