UACM
Universidad Autónoma de la Ciudad de México

Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES

Aplicación de algoritmo esteganográfico LSB en imágenes, programado en Python sobre una Raspberry Pi

T E S I S

QUE PARA OPTAR POR EL TÍTULO DE

LICENCIADO EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES

P R E S E N T A:

SERGIO ARANDA JUÁREZ

D I R E C T O R

DR. EDUARDO RAMOS DÍAZ

C O D I R E C T O R

M. EN I. LUIS ENRIQUE ARANDA MELO

Ciudad de México, noviembre de 2022.

UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

CA Coordinación Académica

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO
COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS©

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

‹‹Incluso si el viento sopla fuerte
Nunca cierres los ojos, mira
siempre hacia adelante››

A.H.

Agradecimientos

En primer lugar a mis padres Alicia Juárez Martínez y Juan Manuel Aranda Herrera, que además de brindarme la oportunidad de llegar hasta este punto me han ayudado a convertirme en quien soy hoy. ¡Lo hicimos!

Mis padrinos y familiares más cercanos Claudia Margarita Juárez Martínez, y Juan Cruz Ortiz, mi familia no está completa sin ustedes, con cariño les agradezco pues siempre han logrado ser un soporte para todos nosotros.

Al Dr. Eduardo Ramos Díaz quien a pesar del poco tiempo de estar en el plantel San Lorenzo Tezonco y de todo el trabajo, me ayudó como director para que este trabajo pudiera ser realizado.

Al M. en I. Luis Enrique Aranda Melo que aceptó codirigir este trabajo.

A mis otros profesores de Casa Libertad Patricia Hong, Pedro Abarca y Oscar Valdés, de quienes aprendí muchas cosas, a pesar de lo que puedan pensar siempre estaré agradecido.

A mis primos Ricardo Cruz Juárez y José Alberto Cruz Juárez así como a mi hermano Juan Manuel Aranda Juárez que si soy o alguna vez llego a ser alguien ha sido por su guía y paciencia, de los tres siempre aprendo algo aunque a veces no lo parezca.

A Freddy quien no pudo terminar con nosotros para este momento por siempre, gracias.

A Miguel, Aldo y Carlos quienes a pesar de ir cambiando los planes a futuro siempre me dieron la mano cuando lo necesité y lo seguirían haciendo en caso de necesitarlo, lo sé y espero que sepan que de mi obtendrán lo mismo.

A Gerardo, Samuel y Gabriel que son mis más antiguos amigos que si bien las cosas han cambiado porque eso es lo normal, son personas que de alguna manera siguen conmigo o al menos en los momentos que hicieron de nuestra amistad algo de más de diez años, ¡Gracias!.

Al final no puedo agradecer a todas las personas que estuvieron ahí, podría mencionar a muchos más, compañeros, amigos o profesores, sin embargo tengo la esperanza de que sepan que los tengo presentes y tengan por seguro que esa ayuda que me brindaron ha hecho que llegara esta ocasión.

Contenido

Capítulo 1 Introducción 1

1.1. Planteamiento del problema 2

1.2. Propuesta de solución. 3

1.3. Objetivo General 3

1.4. Objetivos específicos 3

1.5. Justificación 4

1.6. Metodología 4

1.7. Organización de la Tesis 4

Capítulo 2 Esteganografía 6

2.1 Técnicas de protección de la información 10

2.2 Definición de la esteganografía 11

2.3 Principio básico de la esteganografía 14

2.4 Protocolos esteganográficos 15

2.5 Técnicas Esteganográficas 18

2.6. Sistemas de sustitución 19

2.6.1 Bit Menos Significativo ó LSB (Least Significant Bit) 19

Capítulo 3 Características y configuración del Raspberry Pi 22

3.1. Introducción a Raspberry Pi 22

3.1.1 Entorno gráfico de Raspberry Pi 25

3.1.2 Herramientas de Raspberry Pi 28

3.2 Python 31

3.2.1 Bibliotecas en Python 33

Capítulo 4 Método propuesto 34

4.1. Características de las imágenes. 35

4.1.1. Blanco y Negro 35

4.1.2. Escala de grises 36

4.1.3. RGB 36

4.1.4. Planos de Bits 37

4.2. Método Esteganográfico del Bit Menos Significativo 39

4.3. Mediciones de Calidad 44

4.3.1. PSNR 44

4.3.2. SSIM 45

Capítulo 5 Resultados 47

5.1 Resultados Blanco y Negro 47

5.1.1 UACM.bmp (512x256) 48

5.1.2 NIIL.bmp 55
5.1.3 awa.bmp 63
5.2 Resultados Escala de Grises 67
5.2.1 GS1.bmp 67
5.2.2 GS2.bmp 71
5.2.3 GS3.bmp 75
5.3 Resultados RGB 79
5.3.1 SLT.bmp 79
5.3.2 KC.jpg 83
5.3.3 RGB3.bmp 87
5.4 Comparación con una Raspberry Pi 4 91
Capítulo 6 Conclusiones 96
6.1 Trabajo a futuro 102
Referencias 103

Capítulo 1 Introducción

Según el Instituto Federal de Telecomunicaciones (IFT) en 2017 en México hubieron 71.3 millones de usuarios de internet, lo que representa el 63.9 por ciento de la población de seis años o más.

Las principales actividades de los usuarios de Internet en 2017, son: obtener información (96.9%), entretenimiento (91.4%), comunicación (90.0%), acceso a contenidos audiovisuales (78.1%) y acceso a redes sociales (76.6 por ciento).

Con estos datos se puede decir que en el país 57.51 millones de personas se comunican por este medio y por lo tanto esa misma cantidad de personas está expuesta al robo de información.

Como una medida de seguridad en este caso se propone esconder la información por medios de algoritmos esteganográficos, que básicamente esconden información en una imagen, es decir, si alguien intercepta la información, solo será una cubierta manteniendo la información de nuestro interés lejos de la vista de las personas indeseables.

En este trabajo se piensa desarrollar el algoritmo LSB (Bit Menos Significativo por sus siglas en inglés), el cual consiste en esconder la información en el bit menos significativo de cada pixel en una imagen, esto es posible debido a que, si se altera solo ese pixel, la intensidad del pixel solo cambia en una sola unidad haciendo de este un cambio mínimo y prácticamente imperceptible al ojo humano.

La finalidad de este trabajo es que además de que se logre esconder una imagen dentro de otra, se consiga con cierta calidad, por lo que se utilizarán dos medidas para comprobar su distorsión: PSNR (Peak Signal-Noise Ratio) y el SSIM (Structural Similarity Index); para demostrar el correcto funcionamiento de la técnica utilizada.

Como otra característica de este trabajo es que será implementado en Raspberry Pi, la cual es una computadora de pequeño tamaño y procesamiento, similares a los de un Smartphone de gama baja, esto para demostrar el uso de la esteganografía en estas arquitecturas.

1.1. Planteamiento del problema

Debido a la cotidianidad del uso del internet, la información de cualquier persona está expuesta y últimamente se han visto casos desde filtraciones gubernamentales, de juegos, películas e incluso fotos íntimas de la farándula, por lo que es necesario tener a la mano un algoritmo que añada un escalón más de seguridad a nuestros mensajes.

Existen distintas formas para proteger la información para un medio como la Internet, entre todas ellas está la esteganografía, la cual es capaz de ofrecer un canal oculto de información, lo cual dificulta el obtener a alguien nuestro mensaje; el método propuesto para esta tesis es conocido como LSB, entonces, la finalidad es comprobar que dicho método cumple con el propósito planteado en este trabajo.

Debido a que las implementaciones tradicionales se concentran en aplicaciones de software, se dificulta su aplicación en tiempo real, entonces, el problema que da origen a este trabajo es precisamente la falta de implementaciones de los algoritmos esteganográficos en hardware, los cuales podrían ser utilizados para proteger información en tiempo real.

1.2. Propuesta de solución.

En este trabajo se propone utilizar una tarjeta de desarrollo de hardware programable para aplicar un algoritmo esteganográfico.

Se propone el método en tres módulos: el de inserción, el de extracción y el de evaluación, y con base a las mediciones del último comprobar la efectividad del algoritmo.

1.3. Objetivo General

Implementar el algoritmo LSB en hardware programable evaluando su efectividad mediante métricas de calidad sobre imágenes esteganográficas.

1.4. Objetivos específicos

- Implementar el algoritmo de inserción y generación de la imagen esteganográfica en un lenguaje compatible con el hardware programable.

- ❖ Implementar el algoritmo de extracción de la imagen esteganográfica en un lenguaje compatible con el hardware programable.
- ❖ Evaluar los algoritmos esteganográficos usando métricas de calidad en imágenes.
- ❖ Probar el desempeño de los algoritmos de inserción y extracción usando imágenes a color y en escala de grises.

1.5. Justificación

A pesar de la existencia de la esteganografía y de distintos métodos para ocultar la información, es importante ver los alcances de herramientas para conocer la capacidad de implementar y ejecutar un algoritmo LSB funcional.

1.6. Metodología

- Estudio de la esteganografía así como métodos esteganográficos
- Estudio específico de características y variantes del algoritmo LSB
- Implementación de algoritmo LSB
- Revisión de medidas de desempeño obtenidas para así evaluar calidad del algoritmo.

1.7. Organización de la Tesis

El trabajo se organiza de la siguiente manera:

En el capítulo I se hace una introducción, se listan los objetivos, se plantea el problema y se justica este trabajo, en el capítulo II se presenta el marco teórico, donde se exploran las técnicas de ocultación de información, la esteganografía y algunas de las variaciones existentes del algoritmo LSB, en el capítulo III se da detalle del hardware y el lenguaje de programación utilizados para este proyecto, principalmente las ventajas que permitieron que fueran la elección para ser llevadas a cabo en una Raspberry Pi y programado en Python. En el capítulo IV se explicarán las características del algoritmo, así como las características de la imagen portadora y la información para ocultar, en el capítulo V se mostrarán los resultados obtenidos por el proceso propuesto, las comparaciones se harán a partir del PSNR y el SSIM, así como una comparación de resultado con un hardware más moderno y por último en el capítulo VI se plasmarán las conclusiones de este trabajo, los posibles alcances y el trabajo a futuro, tanto de los algoritmos como en la selección de hardware y como el lenguaje de programación.

Capítulo 2 Esteganografía

La palabra esteganografía significa ‘escritura cubierta’, el nombre viene de griego ‘steganos’ que significa cubierto u oculto y ‘grafos’ que es escritura, actualmente la esteganografía se define muchas veces como la práctica de esconder un archivo, mensaje, imagen o video en otro archivo, mensaje, imagen o video.

La esteganografía es recurrentemente comparada con la criptografía, sin embargo, aunque ambas esconden información lo hacen de formas distintas ya que la criptografía trata de ocultar el contenido del mensaje pero estando a la vista de todos, como ejemplo la figura 1.

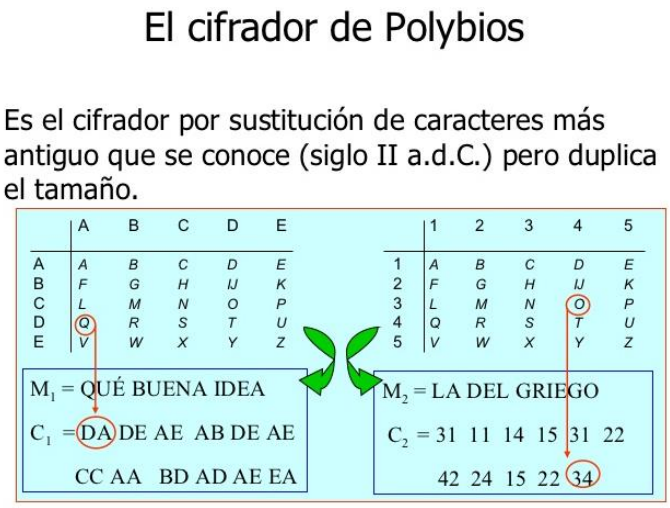


FIGURA 1 EJEMPLO DE UN SISTEMA CRIPTOGRÁFICO

Como se observa en la figura 1 el mensaje a pesar de ser inentendible para alguien que no conoce el método está a la vista de todo el mundo, en cambio la esteganografía da un paso más y hace invisible el mensaje ya que lo oculta

(actualmente) en imágenes, videos o audios que aunque conociendo el archivo original sea imposible notar un cambio entre archivo original y el modificado.

Existen antecedentes a cerca de la esteganografía que resultan fascinantes, los cuales son mencionados por Heródoto, en su obra “Los Nueve Libros De La Historia”, en el texto se narra como un famoso tirano griego llamado Histieo rapó a su más fiel esclavo y tatuó en la cabeza un mensaje en el que alentaba a un aliado a rebelarse contra los persas, el hecho de escribir el mensaje en la cabeza radica en el hecho de esperar a que creciera el pelo del esclavo para poder mandar el mensaje y el destinatario tuvo que afeitar la cabeza del emisario para poder leerlo.

A lo largo de la historia se han utilizado diferentes métodos para ocultar la información, dentro de los cuales los más conocidos son la tinta invisible, la cual fue muy usada durante la segunda guerra mundial, las marcas hechas por medio de pequeños orificios hechos con pinchazos de alfiler. Otro hecho histórico de la esteganografía fue la manera en que los griegos del siglo V a. C. se enteraron del plan de invasión de Persia pues según se narra en la obra del historiador Heródoto “Los Nueve Libros De La Historia” cuenta como Demarato, un exiliado griego de Persia, grabó los planes persas en un par de tablillas de madera y después las cubrió con cera, ocultando así el mensaje, dichas tablillas partieron desde la ciudad persa de Susa hasta Esparta sin ser interceptadas en el camino. Una vez en su destino, Gorgo, la esposa del rey Leónidas, adivinó que debajo de la cera debería encontrase algo escrito. Una vez leído el mensaje, Esparta comunicó las intenciones persas al resto de las ciudades griegas y, gracias a ello, los griegos pudieron armarse a tiempo y derrotar a los persas en las batallas de las

Termópilas, de Salamina y Platea (Heródoto, 2009). Otro antecedente es la escritura con tinta invisible, la cual seguramente es la técnica más empleada en todos los tiempos, existen antecedentes de que en el siglo I, “Plinio el viejo mostró cómo hacer tinta invisible con el jugo de ciertas plantas, dichas tintas eran llamadas simpáticas, se hacen visibles al entrar en contacto con otra sustancia o calor” (Ortega, 2006). En el siglo XV el científico italiano Giovanni Batista della Prota descubrió cómo esconder un mensaje dentro de un huevo cocido, el método consistía en preparar una tinta mezclando una onza de alumbre y una pinta de vinagre, y luego se escribía en la cáscara. “La solución penetra en la cáscara porosa y deja en la superficie de la albúmina del huevo duro, que se puede leer pelando el huevo” (López, 2012). Eneas el estratega propuso diferentes técnicas de esteganografía que para su época resultaban novedosas y efectivas, algunos de estos métodos es el ocultamiento de mensajes en los aretes de las mujeres y los mensajes enviados por medio de palomas mensajeras. La lingüística esteganográfica también llamado acróstico, fue uno de los métodos más populares de la antigua esteganografía.

Una versión más avanzada de la lingüística esteganográfica fue concebida originalmente en China y reinventada por Girolamo Cardano, la cual es conocida también como rejilla móvil, de celosía o de Cardano, en honor a su autor (siglo XVI), consiste en un cuadro de lados iguales que se puede utilizar siempre, dividido en tantas cuadriculas como se quiera, y de las cuales se vacían la cuarta parte. De ahí, que al girarlo y colocarlo en cada una de las cuatro posiciones

posibles vayan dejando al descubierto espacios distintos (Katzenbeisser & Petitcolas, 1999).

Un ejemplo esteganográfico curioso en este periodo tuvo lugar en la Inglaterra del siglo XVI donde la esteganografía tuvo una gran importancia en las conspiraciones urgidas entre los nobles católicos ingleses que querían destronar a la reina protestante Isabel I (1533-1603) y entregar el trono a la católica María I de Escocia (1542-1587). La comunicación entre los conspiradores y la reina María debía pasar lo más desapercibida posible ya que cualquier conocimiento de esta implicaría ser acusados de alta traición y condenados a muerte. Por este motivo, emplearon tanto criptografía como esteganografía para encubrir sus mensajes. Un mecanismo recurrido fue la ocultar lo mensajes en barriles de cerveza que se transportaban sin levantar la atención.

En una obra del científico alemán Gaspar Schott (1608-1666) “Schola Steganographica” se describía cómo esconder mensajes en partituras de música, haciendo equivaler una nota musical concreta con una letra. En ningún momento se buscó que las notas musicales tuvieran alguna coherencia y su resultado diera una melodía agradable, en cualquier caso, su representación, por ejemplo, en papel, permitía perfectamente ocultar un mensaje. Basado en estas ideas surgieron otros procedimientos de ocultamiento, como los basados en el número de ocurrencias de las notas.

2.1. Técnicas de protección de la información

El interés concreto de un sistema esteganográfico dependerá de tres características:

- Capacidad: Cantidad de información que puede ser ocultada.
- Seguridad/invisibilidad: Probabilidad de detección.
- Robustez: Cantidad de alteraciones dañinas que el medio puede soportar antes de que se pierda la información oculta.

Stefan Katzenbeisser en su obra "Information Hiding Techniques for Steganography and Digital Watermarking" menciona que entre las principales subdisciplinas para ocultar información se encuentran las siguientes (Katzenbeisser & Petitcolas, 1999):

Canales encubiertos: Es el lugar utilizado para transferir información, pero para que funcione, el emisor y el receptor deben de conocer este canal para que sepan donde deben encontrar el mensaje oculto.

Anonimato: Es el ocultar la información de origen o destino de la imagen, la idea es que se pueda ocultar el rastro de un mensaje, utilizando a muchos intermediarios por ejemplo, haciendo que el mensaje sea más difícil de percibir dentro del archivo modificado.

Escritura Cubierta o esteganografía: Mientras que la criptografía se encarga de proteger el contenido de los mensajes, la esteganografía es la ciencia que se encarga de ocultar la existencia del mismo. En la actualidad la esteganografía se usa para esconder mensajes dentro de otro mensaje, de tal forma que el más

externo se hace público (visible por todos) y solo el receptor de dicho mensaje, decodificando el archivo modificado, podrá recuperar el mensaje oculto. Más adelante se ampliará la definición y se presentarán diferentes ejemplos.

Watermarking o Marca de Agua: En el caso de la esteganografía, tiene un requerimiento adicional de robustez contra posibles ataques en donde se trata principalmente de hacer que la marca sea indetectable; sin embargo, las marcas de agua no siempre necesitan ocultarse, ya que algunos sistemas utilizan marcas de agua digital visibles.

2.2. Definición de la esteganografía

La palabra esteganografía procede del griego y consta de dos palabras: steganos, que significa encubierto y graphos, que significa escritura, por tanto su traducción completa es "escritura encubierta". Existen diversas definiciones que se manejan en los textos relacionados con la esteganografía, de las cuales a continuación se presentan algunas.

En el contexto de las comunicaciones seguras, dos partes pueden intercambiar información oculta sobre un canal inseguro de modo que, aunque una tercera parte no autorizada intercepte, y realice un análisis de los datos de información, no podrá detectar o probar la existencia de la información oculta y solo percibirá su portadora, contenedor, cubierta, envoltorio o texto inocuo utilizado como tapadera.

La esteganografía moderna utiliza un medio digital como archivos de texto, audio, imagen y video que son utilizados como archivos portadores, el mensaje secreto

se oculta en estos, a este tipo de archivos portadores se les denomina contenedores, cubiertas o portadoras, de igual forma el mensaje secreto o la información a ocultar, puede ser cualquier medio digital, cuando el mensaje secreto es ocultado en el contenedor a través de una técnica esteganográfica se obtiene un estego-objeto que contendrá el mensaje oculto en el archivo portador. El proceso descrito se muestra en la Figura 2, puede observarse al contenedor y el mensaje secreto que puede o no hacer uso de técnicas de cifrado para darle una mayor protección a la información. En el canal, aunque ninguna persona sospechara la existencia del mensaje, éste se encuentra expuesto a cualquier ataque. Finalmente, para recuperar el mensaje secreto se aplica el proceso inverso.

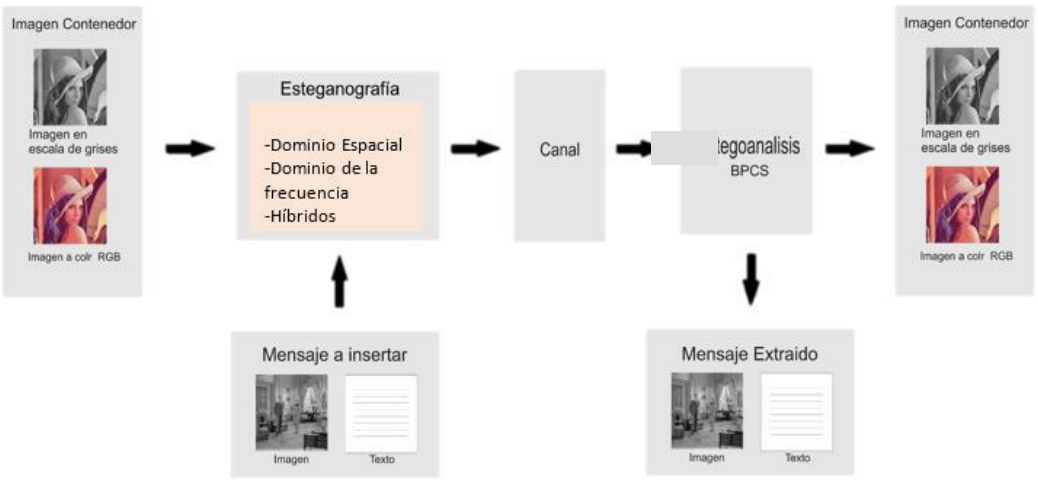


FIGURA 2 DIAGRAMA DE PROCESO ESTEGANOGRÁFICO

Según Stefan Katzenbeisser los elementos de un sistema esteganográfico son los siguientes (Katzenbeisser & Petitcolas, 1999):

- Contenedor: Se trata de la entidad que se emplea para portar el mensaje oculto, este puede ser elegido entre una gran variedad de archivos digitales, por ejemplo audio, video e imágenes.
- Mensaje: Es la información que se desea transmitir.
- Esteganograma: Se trata del objeto contenedor más el mensaje encubierto. Se refiere al objeto que ya tiene el mensaje adherido a él y se utilizará para transmitir este mismo mensaje a través del canal.
- Canal: Medio por el cual es transmitida la información.
- Estegoanálisis: Ciencia que estudia la detección de información dentro de un esteganograma. La detección de la información puede realizarse por medio de ataques pasivos o activos.
- Adversario: Son todos aquellos entes a los que se trata de ocultar la información encubierta. Este adversario puede ser pasivo o activo.
- Adversario pasivo: Es aquel que sospecha que se puede estar produciendo una comunicación encubierta y trata de descubrir el algoritmo que se extrae del estego-objeto, pero no trata de modificar dicho objeto.
- Adversario activo: Aquel que además de tratar de hallar el algoritmo de comunicación encubierta, modifica el estego-objeto con el fin de corromper cualquier intento de mensajería subliminal.

2.3. Principio básico de la esteganografía

La esteganografía con un adversario pasivo está muy bien ilustrado en el problema de los prisioneros (Simmons, 1998).

“Alice y Bob son detenidos por algún delito y son encarcelados en celdas diferentes. Ambos quieren desarrollar un plan de escape, para su desgracia cualquier tipo de comunicación entre ellos será vigilada por la guardia Wendy, misma que no les permite la comunicación por medio de mensajes cifrados y si ella nota cualquier comunicación sospechosa los confinará a una celda de aislamiento total, por lo que ambas partes desean comunicarse de manera secreta sin que Wendy se dé cuenta de los mensajes, por lo que tienen que establecer un canal subliminal de comunicación. Una forma práctica de hacerlo es ocultar el significado de la información en un mensaje que no despierte sospecha alguna. Por ejemplo, Bob podría esconder su mensaje usando la primera letra de cada palabra dentro de un párrafo aparentemente inocente el cual no llamara la atención de Wendy.

Para la desgracia de los prisioneros existen otros problemas que dificultarán el escape de Alice y Bob, por ejemplo, Wendy puede alterar el mensaje que Bob le ha mandado a Alice. Wendy podría cambiar las palabras que Bob ha usado y así destruir la información, o aún peor, si actuara de forma maliciosa podría falsificar el mensaje y enviarlo a uno de los dos criminales pretendiendo ser uno de ellos”.

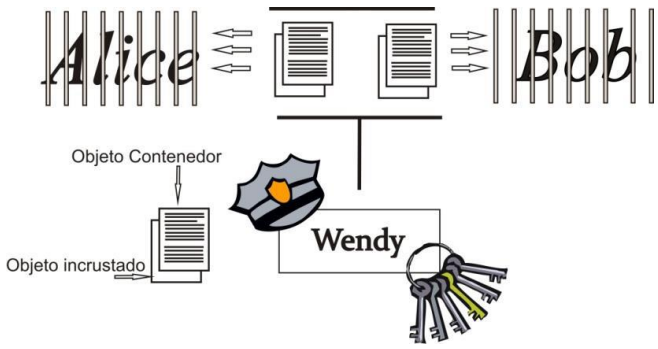


FIGURA 3 MODELO ESTEGANOGRÁFICO DEL PROBLEMA DE LOS PRISIONEROS.

El modelo presentado en la Figura 3 es generalmente aplicable en situaciones de comunicaciones esteganográficas, donde Alice y Bob representan dos partes de la comunicación (emisor y receptor) deseando intercambiar información secreta, así mismo la guardia Wendy representa un tercer sujeto con posibilidad de leer y posiblemente alterar los mensajes enviados entre emisor y receptor.

Teniendo en cuenta que pueden existir adversarios activos, una buena técnica esteganográfica debe ser robusta ante distorsiones, ya sean accidentales o fruto de la interacción de un intruso activo.

2.4. Protocolos esteganográficos

La mayoría de las aplicaciones esteganográficas surgen a partir del principio general, que se ilustra en la figura 4. Alice, tiene la intención de compartir un mensaje secreto *m* con Bob, elige al azar (usando la fuente aleatoria privada *r*) un mensaje inofensivo *c*, llamado objeto de presentación, el cual puede transmitirse a Bob sin levantar sospechas, e incrusta el mensaje secreto en *c*, probablemente

usando una clave *k*, llamado clave estego. Por lo tanto, Alice cambia la cubierta *c* para un estego-objeto *s*. Esto debe ser hecho de una manera muy cuidadosa, de manera que un tercer partido, sabiendo sólo el mensaje *s* aparentemente inofensivo, no pueda detectar la existencia del secreto. En un sistema "perfecto", una cubierta normal no debe ser distinguible de un estego-objeto, ni por un ser humano ni por un dispositivo electrónico que busque un patrón estadístico. En teoría, las cubiertas pueden ser cualquier dato entendible por una computadora, este puede ser: imagen, sonido digital o texto escrito.

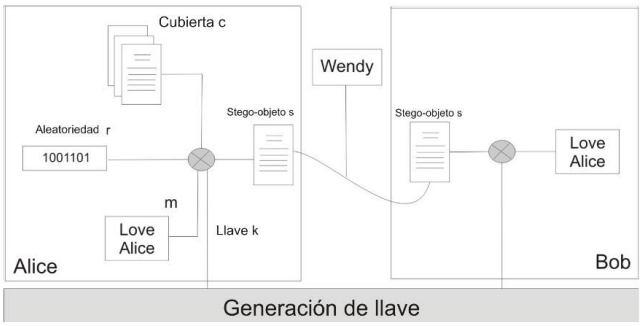


FIGURA 4 DESCRIPCIÓN ESQUEMÁTICA DEL PROBLEMA DE LOS PRISIONEROS.

Alice transmite a Bob sobre un canal inseguro, y espera que Wendy no notará el mensaje incrustado. Bob puede reconstruir *m* ya que él sabe el método de incorporación utilizado por Alice, y tiene acceso a la llave *k*, la cual es utilizada en el proceso de incrustación. Este proceso de extracción debe ser posible sin la cubierta original *c*. Una tercera persona que observa la comunicación no debe ser capaz de decidir si el emisor está activo en el sentido que envía cubiertas que contienen mensajes secretos en lugar de las tapas sin información adicional. De manera más formal, si un observador tiene acceso al conjunto {C1,C2...Cn} de

objetos de cubertura de transmisión entre ambas partes, la comunicación debe ser capaz de decidir cuáles objetos cubierta *Ci* contienen información secreta. Por lo tanto la seguridad de la comunicación invisible se presenta principalmente en la incapacidad de distinguir objetos cubierta de los estego-objetos. En la práctica sin embargo, no todos los datos pueden ser utilizados como cubiertas para la comunicación secreta, ya que las comunicaciones empleadas en el proceso de inserción no deben ser visibles para alguien no involucrado en el proceso de comunicación.

Este hecho requiere que la cubierta contenga suficientes datos, los cuales pueden ser reemplazados por información secreta. Como ejemplo, debido a errores de medición cualquier resultado de algún proceso de escaneo físico, contendrá un componente estocástico llamado ruido, tales artefactos aleatorios pueden utilizarse para él envió de información secreta. De hecho, se tiene que los datos ruidosos tienen propiedades más ventajosas en la mayoría de aplicaciones esteganográficas. Una cubierta no debe ser utilizada dos veces debido a que el atacante, tiene acceso a las dos versiones de la cubierta y puede fácilmente detectar y posiblemente recuperar el mensaje. Para evitar el rehúso accidental, tanto el emisor como el receptor deben destruir todas las cubiertas que han utilizado para la transferencia de información.

2.5. Técnicas Esteganográficas

Los métodos de esteganografía se clasifican respecto a las características que emplea, por ejemplo: uno de ellos se clasifica de acuerdo al tipo de cubierta que utiliza para la comunicación secreta, mientras que otro los clasifica de acuerdo a las modificaciones que se aplicaron en el proceso de ocultamiento.

(Katzenbeisser & Petitcolas, 1999, págs. 43-44) clasifican a los sistemas esteganográficos en las siguientes categorías:

- Sistemas de sustitución: En este sistema se sustituyen las partes redundantes de una cubierta, por un mensaje secreto.
- Técnicas en el dominio de la transformada: Inserta la información secreta en un espacio transformado de la señal, por ejemplo, Transformada Coseno Discreta (DCT) y Transformada de Fourier Discreta (DFT).
- Técnicas de espectro disperso. En estas se adoptan ideas utilizadas en las comunicaciones de espectro disperso.
- Método estadístico. Esta técnica codifica información cambiando varias propiedades estadísticas de la cubierta, y utiliza pruebas de hipótesis en el proceso de extracción.
- Técnicas de distorsión. Almacenan información por distorsión de señales y miden la desviación de la cubierta original en el paso de la codificación.
- Métodos de generación de cubierta. Codifican información de tal forma que se crea una cubierta para comunicación secreta.

2.6. Sistemas de sustitución

Estas técnicas utilizan los niveles de pixeles grises y sus valores de color directamente para que codifique los bits del mensaje. Estas técnicas son algunos de los esquemas más simples en términos de la incorporación y la complejidad de extracción. El principal inconveniente de estos métodos es la cantidad de ruido aditivo que se adhiere a la imagen, que afecta directamente a la relación pico señal a ruido y las propiedades estadísticas de la imagen. Además, estos algoritmos de incrustación son aplicables a la compresión de imágenes sin pérdida, como las imágenes TIFF. Para los esquemas de compresión con pérdida como JPEG, algunos de los bits de mensaje tienen pérdidas durante la etapa de compresión. El algoritmo más común que pertenece a esta clase de técnicas es el bit menos significativo o LSB, esta técnica se basa en el reemplazo del bit menos significativo, utiliza la representación binaria de los niveles de gris de pixel para representar el bit de mensaje.

Un sistema básico de sustitución trata de ocultar la información secreta sustituyendo las partes insignificantes de la cubierta por los bits del mensaje secreto, el mensaje puede ser extraído si se conocen los puntos donde la información fue incrustada.

2.6.1 Bit Menos Significativo ó LSB (Least Significant Bit)

El algoritmo del Bit Menos Significativo es un método escalar para marcar, comúnmente imágenes, donde cada pixel es representado por cadenas de 8 bits. En su forma básica, para cada pixel de la imagen se modifica un solo bit de información, el menos significativo (Eason, 2003, págs. 18-31).

El método de Inserción en el bit Menos Significativo, llamado también método de sustitución, consiste en reemplazar el bit menos significativo de los pixeles de una imagen por otros Bits que representan el mensaje que se quiere ocultar.

El principio de funcionamiento del método de sustitución consiste en guardar información en los bits menos significativos de manera que los cambios no sean percibidos por el ojo humano. Por ejemplo, si se quisiera esconder dentro de una imagen la letra A, la cual es representada en el código ASCII por el siguiente valor binario: 01000001, el cual está formado por una cadena de 8 bits, para una imagen a color RGB necesitaríamos utilizar 3 pixeles de la imagen, cada pixel tiene 3 bytes para cada color, en la figura 5 y figura 6 puede apreciarse el reemplazo de los bits.

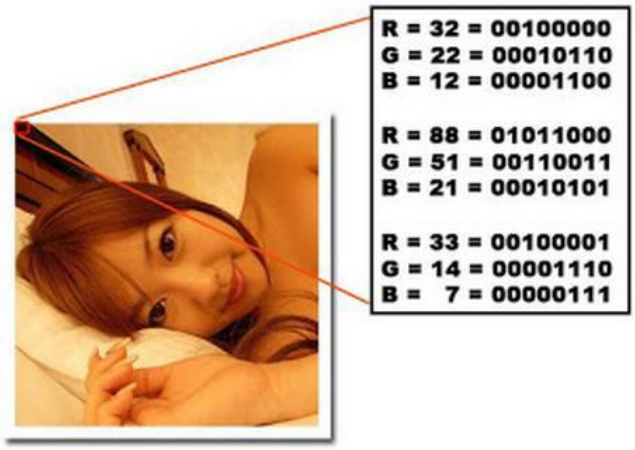


FIGURA 5 VALORES DE PIXELES EN RGB

Al aplicar la sustitución LSB de cada número binario por cada bit del carácter "A" se obtendría el mensaje encubierto (estego-objeto).

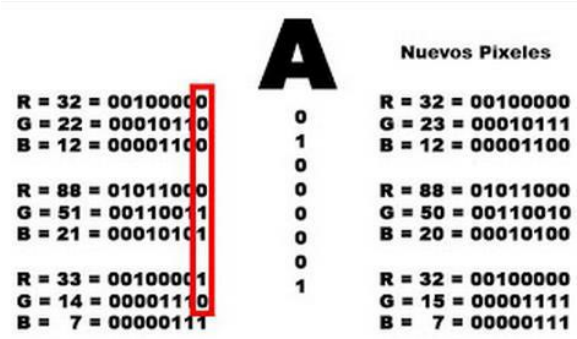


FIGURA 6 EJEMPLO DE SUSTITUCIÓN DEL BIT MENOS SIGNIFICATIVO POR EL OBJETO A.

Este método quizá sea una de las técnicas más comunes y también una de las más sencillas de la esteganografía en imágenes, este método permite el ocultamiento de una gran cantidad de información con un mínimo deterioro perceptual de la imagen cubierta.

La ventaja de este tipo de métodos es la facilidad de codificación y decodificación, sin embargo, presenta desventajas notables como graves limitaciones al momento de la compresión, por ejemplo, en el caso de utilizar una imagen digital como portadora, los valores de los bits menos significativos cambian al realizar un proceso de compresión con pérdidas por lo tanto al momento de realizar la decodificación, de los datos insertados, es imposible obtener los datos del mensaje oculto incrustado. A pesar de esto es muy fácil realizar un programa para aplicar el método de ocultamiento de datos mediante LSB.

Capítulo 3 Características y configuración del Raspberry Pi

En este trabajo es muy importante el hardware que se utilizó para la implementación del algoritmo ya que a pesar de sus desventajas en especificaciones con computadoras más potentes tiene unas cuantas ventajas que si bien no son técnicas son útiles por cuestiones prácticas.

3.1. Introducción a Raspberry Pi

La Raspberry Pi es una computadora de bajo costo y del tamaño de una tarjeta de crédito lista para usar al conectar una pantalla un teclado y un mouse estándar, están diseñadas para que personas de todas las edades exploren el cómputo y en caso de ser de interés también aprendan a programar lenguajes como Python o Scratch.

A pesar de su tamaño puede realizar tareas básicas de cualquier PC como navegar en internet, procesamiento de textos, reproducción de videos de alta definición e incluso juegos.

Además de las características antes mencionadas con la Raspberry se pueden realizar proyectos de electrónica (por ejemplo) ya que tiene pines programables de entrada y salida.

Para este trabajo se utilizó una Raspberry Pi 2 modelo B con las siguientes especificaciones:

- Procesador Cortex-A7 quad-core ARM a 900MHz
- 1GB RAM
- 4 puertos USB
- 40 pines GPIO
- Puerto HDMI
- Puerto Ethernet
- Puerto 3.5mm de audio
- Interfaz de Cámara (CSI)
- Interfaz de display (DSI)
- Lector de tarjeta Micro SD
- VideoCore IV

Esta computadora fue pensada para la educación principalmente, esa razón hace que su costo sea bajo en comparación a computadoras nuevas.

Una gran ventaja es que este equipo puede ser utilizado en algún producto comercial, ya que al ser adquirido el comprador es dueño total del equipo y de la aplicación, lo único que se debe compartir (en caso de ser solicitado) es el código fuente ya que funciona con paqueterías libres y es algo que el desarrollador del producto desea mantener, además tiene la capacidad de cubrir algunas aplicaciones industriales.

Una desventaja es que a diferencia de las computadoras de escritorio su memoria RAM o tarjeta de video no pueden incrementarse (al igual que las tarjetas de desarrollo, smartphone o tablet), sin embargo el GPU proporciona Open GL ES 2.0, hardware-accelerated Open VG, así como codificación de alto perfil a 1080p, haciendo que sus gráficas estén a un nivel altamente similar a las del original Xbox.

Lo que necesitamos para utilizar la Raspberry Pi es un monitor con entrada HDMI, una memoria Micro SD, un teclado, un mouse y una fuente de poder, como muestra la siguiente tabla:

Tipo de Raspberry	Corriente Recomendada	Corriente máxima de periféricos USB	Consumo típico de la tarjeta
Raspberry Pi Modelo A	700mA	500mA	200mA
Raspberry Pi Modelo B	1.2A	500mA	500mA
Raspberry Pi Modelo A+	700mA	500mA	180mA
Raspberry Pi Modelo B+	1.8A	600mA/1.2A (conmutado)	330mA
Raspberry Pi 2 Modelo B	1.8A	600mA/1.2A (conmutado)	350mA
Raspberry Pi 3 Modelo B	2.5A	1.2A	400mA
Raspberry Pi Zero W/WH	1.2A	Limitado por PSU	150mA
Raspberry Pi Zero	1.2A	Limitado por PSU	100mA

TABLA 1 CORRIENTES PARA LOS DISTINTOS TIPOS DE RASPBERRY PI

En el caso de la memoria Micro SD, es recomendable una de por lo menos 8 GB para que además del sistema operativo, haya espacio de sobra para paqueterías

necesarias además de otro tipo de archivos que podamos utilizar, la memoria con más capacidad que se puede utilizar en este dispositivo es de 128 GB.

En el caso específico de este trabajo se usó de Berryboot, que es un programa de licencia libre que sirve como interfaz de arranque y permite que el sistema operativo esté instalado en una memoria USB lo cual ofreció un poco más de espacio y estabilidad al funcionamiento del Sistema Operativo.

3.1.1 Entorno gráfico de Raspberry Pi

La Raspberry Pi tiene un entorno de comandos y un sistema operativo basado en Debian de Linux, llamado Raspbian, el cual está constantemente en actualización, al día en el que este trabajo fue realizado la última versión del kernel disponible es la 4.14.

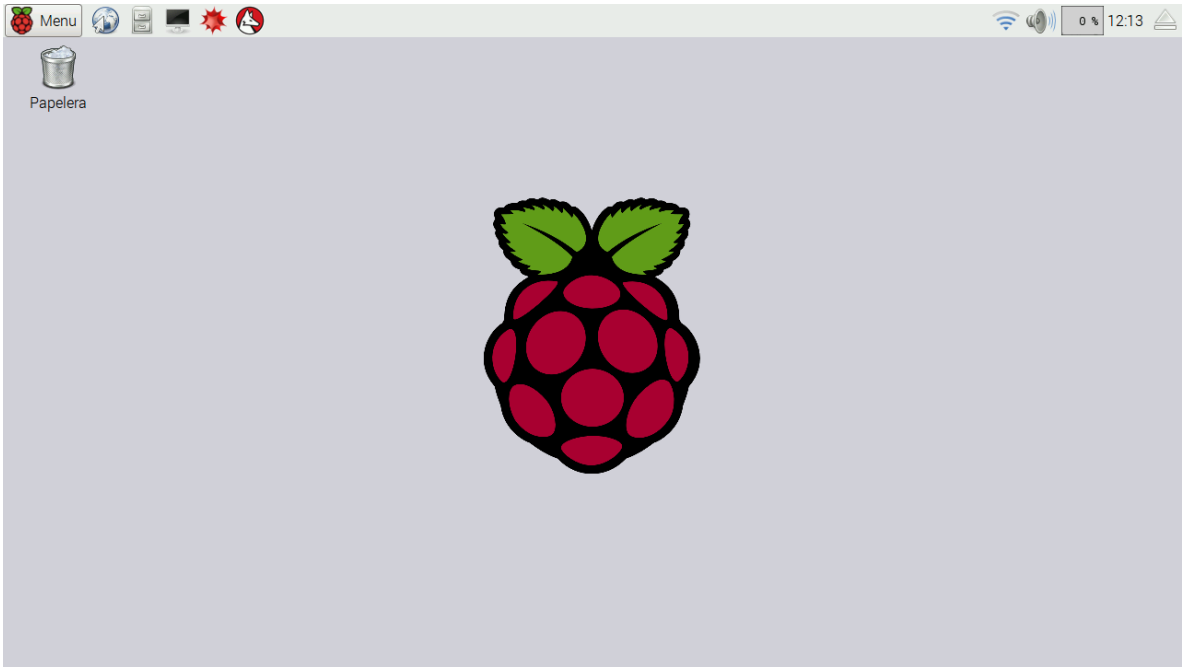


FIGURA 7 ENTORNO GRÁFICO DE RASPBIAN JESSIE

En la Figura 7 se puede ver el entorno gráfico de Raspbian Jessie (la cual es la versión utilizada para este trabajo), sin embargo es posible actualizar la versión y los software de la Raspberry.

Para actualizar el sistema operativo es necesario entrar a una ventana de terminal de Raspbian y una conexión a Internet.



FIGURA 8 UBICACIÓN DEL ÍCONO DE TERMINAL

En la terminal se escribe el comando **sudo apt-get update** y se mostrará de la siguiente forma:

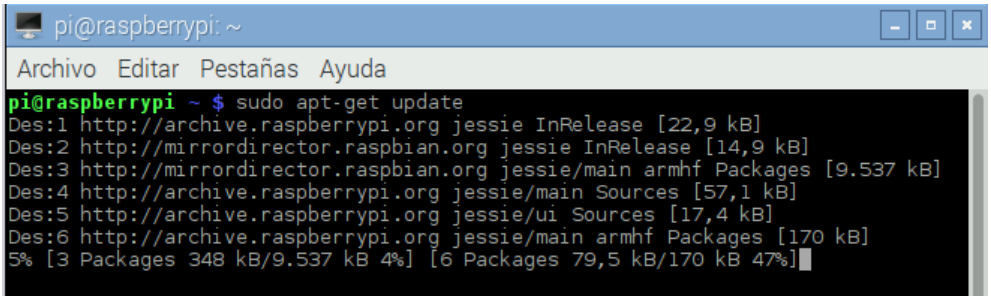


FIGURA 9 ACTUALIZANDO EL SISTEMA OPERATIVO RASPBIAN

Con este comando Raspberry compara los paquetes de sistema operativo con los que están en su repositorio oficial, primero mostrará si hay paquetes distintos entre uno y otro, en caso de haber, pregunta si quieres realizar la actualización si se responde afirmativamente comenzará a descargar los paquetes necesarios para actualizar el sistema operativo.

Una vez que el sistema operativo ha sido actualizado, es conveniente actualizar el software a la más reciente versión, de igual manera se hace desde línea de comando, solo que esta vez se utilizará la línea **sudo apt-get upgrade**, de manera muy parecida al comando **update** se mostrará la siguiente ventana:

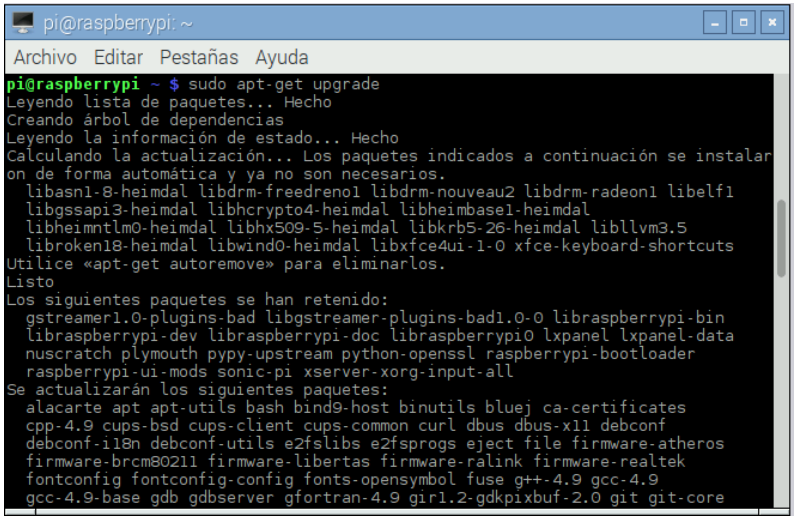


FIGURA 10 FUNCIONAMIENTO DEL COMANDO 'UPGRADE' EN RASPBIAN

En la figura 10 se muestra de forma clara cómo es que funcionan los comandos, descartando, reteniendo y actualizando las librerías (o software) que lo requieran.

De igual forma compara las librerías y sus versiones instaladas en el dispositivo con las del repositorio para que en caso de ser necesario se cambien, este proceso a diferencia del ‘update’ demora, esto debido a que es la actualización de cada parte de la Raspberry.

Como se ve en la figura 7 vemos un sistema operativo gráfico amigable, no muy lejano al que cualquier computadora tiene, esto hace que sea muy intuitivo en cuestiones de uso cotidiano.

3.1.2 Herramientas de Raspberry Pi

Raspberry Pi se muestra como una opción económica y eficiente para desarrollos de estudiantes y desarrolladores principalmente, por lo que estas son las herramientas que podemos ver en su menú:

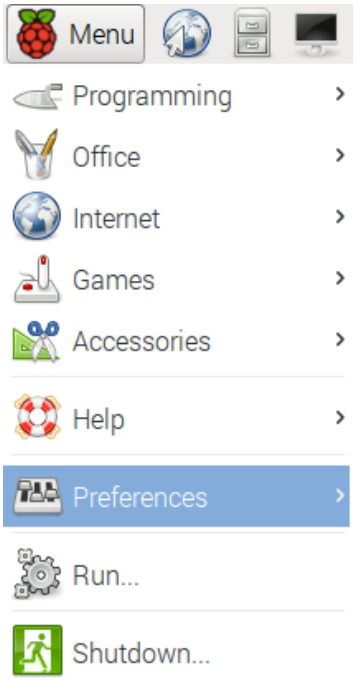


FIGURA 11 MENÚ DESPLEGADO DE RASPBERRY PI

Como se puede ver en la figura 11 Raspberry Pi ofrece lo básico necesario para cumplir como una computadora cualquiera, solo que con la posibilidad de enviar y recibir señales eléctricas a través de su módulo GPIO.

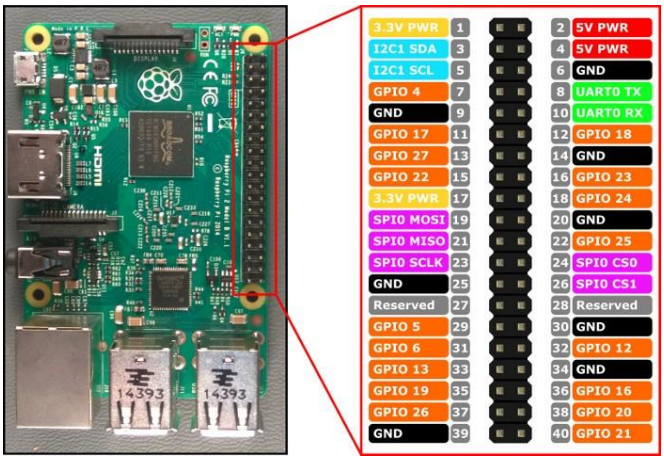


FIGURA 12 FOTO DE UN RASPBERRY CON ESQUEMA DE SU MÓDULO GPIO.

3.1.2.1 Programación

En este submenú se encuentran todas las herramientas de programación con las que cuenta Raspberry Pi por defecto y en especial el lenguaje de programación utilizado en este trabajo, Python, del cual se hablará más adelante.



FIGURA 13 LISTA DE HERRAMIENTAS DE PROGRAMACIÓN EN RASPBERRY PI.

Mathematica es un programa utilizado en áreas científicas, de ingeniería, matemática y áreas computacionales para programar proyectos a nivel escolar superior, incluso más allá.

Scratch es un lenguaje de programación visual, su principal característica consiste en que permite el desarrollo de habilidades mentales mediante el aprendizaje de la programación sin tener conocimientos profundos sobre el código. Sus características ligadas al fácil entendimiento del pensamiento computacional han hecho que sea muy difundido en la educación de niños, adolescentes y adultos.

Sonic Pi se considera un instrumento musical, dando otro tipo de uso para la programación.

Wolfram (o Wolfram language) es un lenguaje de programación multi-paradigma desarrollado por Wolfram Research, que sirve como el principal idioma de interfaz para Mathematica y Wolfram Programming Cloud. Está diseñado para ser lo más general posible, con énfasis en computación simbólica y programación funcional. Está construido para representar estructuras y datos arbitrarios.

3.1.2.2 Internet

Para este trabajo la mayoría de los menús no son de importancia vital para la implementación del algoritmo, sin embargo se mostrará la característica más importante para la realización de este trabajo correspondiente a Raspberry Pi puesto que para posible trabajo a futuro esto se pueden ocupar algunas herramientas distintas a la programación.

Actualmente cualquier dispositivo electrónico con utilidad de computadora debe tener opción de conectarse a Internet, y la Raspberry Pi no es la excepción, además nos ofrece las siguientes herramientas: Correo electrónico, explorador de internet, recursos para la computadora y aplicación para control remoto.

3.2. Python

En primer lugar se mencionó el hardware y sus características más importantes consideradas de forma general para la implementación del algoritmo, sin embargo el punto más importante para este trabajo es el lenguaje de programación seleccionado, el cual es Python.

Python es un lenguaje de programación orientada a objetos, que entre sus ventajas: Maneja una clara sintaxis, tiene clases, módulos y librerías las cuales dan un mayor rango de manipulación ya que estas incluso pueden ser programadas en C o C++, además es de licencia libre.

El hecho de que Python sea de licencia libre hace que el costo de implementación no incremente además de que si es creada una aplicación uno no pierde la propiedad de la misma, solo el código tendrá que compartirse en caso de que sea solicitado.

Según su página web, Python puede ser utilizado para personas que estén interesadas a programar por primera vez o si ya tienen experiencia con otros lenguajes, incluso muestra guías de programación.

```python
#! /usr/bin/env python
# -*- coding: utf-8 -*-

def superposicion (lista1, lista2):
    indice = 0
    for i in lista1:
        for x in lista2:
            if i == x:
                return True
    return False

a = [1,2,3]
b = [4,5,6,7,3]

print superposicion (a, b)
```

FIGURA 14 EJEMPLO DE PROGRAMA EN PYTHON

Lo importante de la figura 14 además de mostrar un programa sencillo de Python, es observar la sangría de cada línea, porque si no respeta, el programa no funciona.

Las principales aplicaciones que se pueden desarrollar en Python son:

- Desarrollo web y de internet
- Acceso de base de datos
- Interfaces Gráficas de Usuario
- Científicas y numéricas
- Educación
- Programación en línea
- Software y desarrollo de videojuegos

Para el caso de este trabajo la aplicación es numérica debido a que las imágenes pueden expresarse como matrices de pixeles.

3.2.1 Bibliotecas en Python

En general, una biblioteca es un conjunto de módulos, a su vez un módulo es un conjunto de partes estandarizadas o unidades independientes que pueden ser usadas para construir una estructura de mayor complejidad.

Capítulo 4 Método propuesto

La esteganografía es el arte de ocultar información digital en multimedia, que a diferencia de la criptografía no trata solo de mantener oculto el mensaje, más bien trata de ocultar el hecho de que hay información oculta, esto es importante porque es bien sabido que la información es vulnerable en la red, otra utilidad de este tipo de algoritmos es la marca de agua, útil para dejar una seña de que una imagen es propiedad de alguien y de ayuda para cuestiones de derechos de autor.

En este trabajo se implementa un algoritmo esteganográfico llamado LSB o Bit Menos Significativo (Least Significative Bit por sus siglas en inglés), para este método lo que se hace es tomar cada pixel de una imagen, el valor de cada pixel en bits y se cambia el bit menos significativo por otro del mensaje oculto.

La finalidad de este trabajo es ocultar imágenes de tres diferentes características en imágenes a color (en canales RGB). El primer caso es esconder una imagen en blanco y negro, el segundo es esconder una imagen en escala de grises y por ultimo esconder una imagen RGB.

Algoritmo básico de LSB

$\boldsymbol{for}\ j = 0, \dots, l(c)\ \boldsymbol{do}$

$s_j \leftarrow c_j$

$\boldsymbol{end\ for}$

$\boldsymbol{for}\ i = 0, \dots, 7\ \boldsymbol{do}$

$cuenta\ el\ subíndice\ i - ésimo\ donde\ se\ intercambiará\ el\ bit\ del\ mensaje$

$\boldsymbol{when}\ i = 7$

$s_{ji} \leftarrow c_{ji} \leftrightarrows m_i$

$end\ for$

Dónde:

$\boldsymbol{s}\ es\ la\ matriz\ correspondiente\ a\ la\ estegoimagen$

$\boldsymbol{c}\ es\ la\ matriz\ correspondiete\ a\ la\ imagen\ cubierta$

$\boldsymbol{m}\ es\ la\ matriz\ correspondiente\ al\ mensaje\ (en\ bits)$

$\boldsymbol{j}\ es\ el\ contador\ de\ pixel$

$\boldsymbol{i}\ es\ el\ contador\ de\ bits$

4.1. Características de las imágenes.

Se ha hablado de imágenes en blanco y negro, escala de grises y RGB, en este apartado, a continuación se explicaran las diferencias.

4.1.1. Blanco y Negro

Una imagen en blanco y negro existe en un solo canal y cada pixel solo puede tener dos valores disponibles 0 para negro y 1 para blanco, una imagen de n x m pixeles tiene n x m bits.



FIGURA 15 IMAGEN BLANCO Y NEGRO REFERENCIA EN ESTE TRABAJO.

4.1.2. Escala de grises

Las imágenes en escala de grises son más complejas que las imágenes a blanco y negro ya que estas en lugar de variar entre dos valores lo hacen entre 256 valore y gráficamente como su nombre lo indica muestra una variedad de grises que permiten dar mayor detalle a una imagen incluso sin tener una variedad de color, la representación numérica de las intensidades va de 0 a 255 por lo que en números binario son necesarios 8 bits, entonces una imagen *nxm* pixeles en escala de grises es de *8(nxm)* bits.



FIGURA 16 IMAGEN EN ESCALA DE GRISES UTILIZADA EN ESTE TRABAJO

4.1.3. RGB

Las imágenes RGB (Red, Green, Blue) son las que están conformados por canales de color (Rojo, Verde y Azul), cada canal es similar a uno de escala de grises, que puestos uno sobre el otro forman una imagen de color, el tamaño de una imagen RGB de n x m pixeles es de (3 x 8)(n x m).



FIGURA 17 IMAGEN EN COLOR RGB UTILIZADA EN ESTE TRABAJO.

4.1.4. Planos de Bits

Las imágenes digitales están compuestas por pixeles y cada pixel está compuesto por ocho bits (al menos en los formatos utilizados en este trabajo), entonces si se graficara cada plano formado por cada bit es lo que se le conoce como plano de bits.

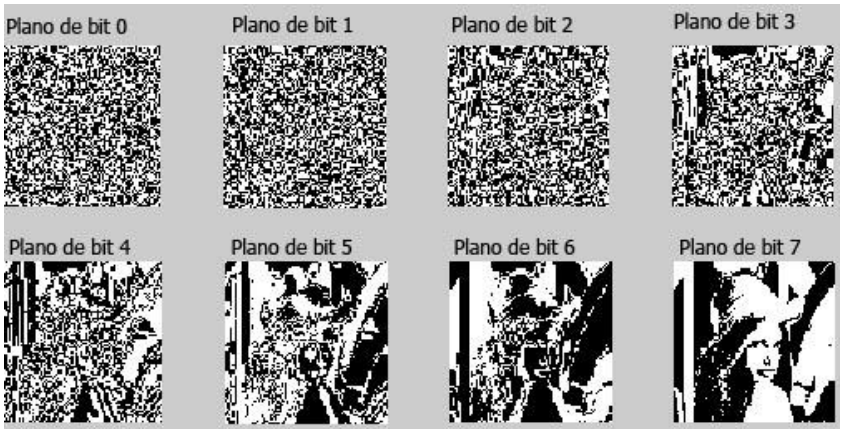


FIGURA 18 PLANOS DE BIT DE LENA

Como se puede ver en la Figura 18, cada plano de bit muestra distintos pesos en la imagen, es decir el séptimo bit es el que tiene más peso o mejor dicho es el Bit Más Significativo (MSB por sus siglas en inglés), pues al ver ese plano muestra un

esbozo de la imagen, es decir es fácil entender de qué se trata la figura dentro de la imagen.

El Bit Menos Significativo (LSB) es aquel que en el plano de bits muestra un gráfico aleatorio, sin una forma que muestre lo que es la imagen original, incluso puede ser considerado como ruido por lo que el método esteganográfico a implementar utilizará en ese plano.

En la figura 19 se muestra de forma gráfica la diferencia entre la composición en bits de los pixeles en los distintos tipos de imágenes haciendo notar que en el caso de una imagen en blanco y negro se mencionó que solo tiene dos valores posibles el 0 y el 1 ya que si se toma un bit sin importar la posición de este o incluso uno distinto de cada pixel formará la misma imagen.



FIGURA 19 FORMA DE UN PIXEL EN TRES TIPOS DE IMÁGENES.

4.2. Método Esteganográfico del Bit Menos Significativo

La función de un método esteganográfico es ocultar información, pero no solo se trata de ocultar un mensaje, se trata de ocultar el hecho de que información se está ocultando, esa es la razón de la existencia del método del Bit Menos Significativo ya que como se mostró en los planos de bits, el plano LSB es el plano que menos muestra de la figura original, es decir si este es alterado, la estego-imagen será visiblemente igual a la imagen portadora original evitando levantar sospechas que adentro hay otra información.

Entonces lo que se necesita de principio son archivos adecuados, uno portador o visible y un mensaje u oculto (para este trabajo imágenes). Es necesario mencionar que el algoritmo en general no funciona únicamente para imágenes sin embargo el desarrollado para este trabajo funciona solo para imágenes.

Una cosa que es necesario mencionar es que una imagen puede ser representada por matrices de números, esto es posible debido a que cada pixel muestra una intensidad y esa matriz representan esas intensidades que dependiendo el tipo de imagen son los valores que cada pixel puede adquirir.

Para los fines de este trabajo se convendrá que la imagen cubierta es una imagen en color en canales RGB y la imagen oculta será variable en sus características, las ya antes mencionadas blanco y negro, escala de grises y otra imagen RGB.

Pese a lo anteriormente mencionado hay que tener dos importantes consideraciones para que funcione de forma adecuada el algoritmo, el formato de la imagen y la relación de tamaños entre imagen cubierta e imagen oculta.

El método esteganográfico LSB consiste en colocar los bits de la imagen oculta únicamente en el Bit Menos Significativo de la imagen cubierta. Por ejemplo si queremos esconder un pixel cuyo valor es 151 lo primero es descomponerlo en bits y cada bit colocarlo en el bit menos significativo de cada pixel de la imagen que será visible, es decir esconder un pixel oculto necesita ocho pixeles visibles.

En la siguiente ilustración se muestra un ejemplo de que es lo que hace este algoritmo.

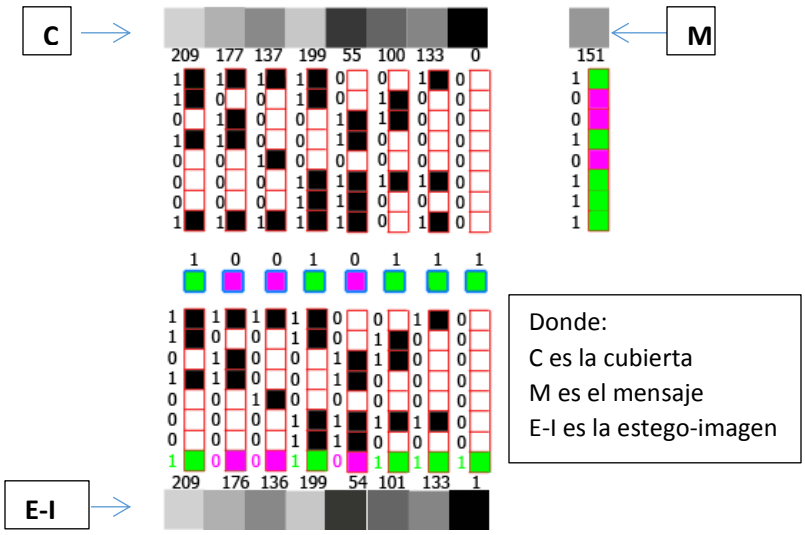


FIGURA 20 EJEMPLO DE CÓMO SE ESCONDE UN PIXEL POR MÉTODO LSB

La figura 20 muestra cómo funciona el método esteganográfico por LSB para un pixel, primero el valor del pixel mensaje *M* se descompone en bits y cada bit del mensaje sustituye el bit menos significativo de ocho pixeles de la cubierta *C* y al resultado de ese cambio se le llama estego-imagen *E-I*. El cambio entre cada pixel es máximo de 1 (en un máximo de 255) por lo que es imperceptible al ojo humano.

4.2.1. Consideraciones para el método

En base a lo anterior mencionado, es necesario revisar dos puntos importantes, los formatos en imágenes digitales y la relación de tamaños entre la imagen portadora con la imagen mensaje.

4.2.1.1. Formato de la imagen

Los formatos digitales son distintos entre ellos, por eso en la estego-imagen (la imagen que contiene el mensaje oculto) es necesario utilizar formatos que no tengan algún tipo de pérdida de información, si esta llega a existir no habría forma de recuperar de forma íntegra el mensaje oculto.

Debido a que con las conversiones o cambios de formato puede haber cambios en las intensidades de cada pixel, es recomendable utilizar un solo formato para las tres imágenes, así se evita que cambien los valores de los pixeles en algún punto.

Entonces se mencionaran los tres formatos más comunes y con posibilidades de ser utilizados en este trabajo.

4.2.1.1.1. BMP

BMP (o Windows Bitmap) es un formato estándar usado por Windows para almacenar imágenes independientes del dispositivo e independiente de la aplicación. El número de bits por píxel (1, 4, 8, 15, 24, 32 o 64) de un archivo BMP determinado se especifica en un encabezado de archivo. Archivos BMP con 24 bits por píxel son comunes y seleccionable para este trabajo ya que se ha mencionado que en imágenes de color se trabaja con esta cantidad de bits (8 por canal por tres canales de color). Archivos BMP no suelen comprimirse permitiendo

que sea posible la recuperación de información al no alterar los valores donde esta puede estar oculta.

4.2.1.1.2. PNG

Los archivos PNG (Portable Network Graphics) se comprimen sin pérdida de información. Archivos PNG pueden almacenar los colores con 8, 24, 48 bits por píxel y escalas de grises con 1, 2, 4, 8 o 16 bits por píxel. Un archivo PNG de igual manera puede ser utilizado para cualquier parte de este trabajo ya que al no tener pérdidas de información podría existir la recuperación de la imagen mensaje además de que este tipo de imágenes si puede trabajar a uno o tres canales en caso de que la imagen lo requiera (es decir si está a escala de grises o a color).

4.2.1.1.3. JPEG

JPEG (Joint Photographic Experts Group) es un esquema de compresión que funciona bien para escenas naturales como las fotografías digitalizadas. Se pierde alguna información en el proceso de compresión, pero a menudo la pérdida es imperceptible para el ojo humano. Archivos JPEG almacenan 24 bits por píxel, por lo que son capaces de mostrar más de 16 millones de colores. Incluso con todas las bondades del formato JPEG, no es elegible para algunas aplicaciones, esto es porque todos los métodos de compresión correspondientes a JPEG en su uso general, provocan pérdidas, lo que hace al formato inadecuado como paso intermedio si se van a editar las imágenes y más si se hace en repetidas ocasiones. (John Miano, 1999)

4.2.1.2. Relación de tamaños entre imagen cubierta e imagen oculta

Debido a la forma de ocultar los bits existe una relación dentro de la imagen oculta y la imagen visible y la forma más fácil de calcularla es por medio de los bits es decir si utilizamos como referencia una imagen a color de *n x m* pixeles como cubierta tiene *24 x n x m* bits, entonces si una imagen en blanco y negro de *n x m* pixeles contiene *n x m* bits y se necesita un pixel de la imagen cubierta para esconder uno de la imagen en blanco y negro se puede decir que podemos esconder una imagen en blanco y negro 24 veces más grande que una a color y bajo esta misma lógica podemos decir que podemos ocultar una imagen a escala de grises tres veces mayor y otra imagen a color un octavo menor, pero la realidad es algo distinta. Hay que recordar que es con matrices con lo que se está trabajando, por lo tanto es necesario mencionar que las matrices para la comparación deben ser con matrices de su mismo tamaño, por lo que muchos bits no estarán del todo utilizados (en el caso de utilizar imágenes en blanco y negro así como escala de grises).

Para imágenes en RGB como cubiertas se propone en cuestión de tamaños de los mensajes la siguiente forma:

- Blanco y Negro: El tamaño sea igual entre imagen cubierta e imagen oculta ya que es pixel por pixel solo que el proceso se repetirá en los tres canales de forma redundante
- Escala de grises: El tamaño de la imagen oculta sea ocho veces mayor que la imagen cubierta debido a que se necesitan ocho pixeles de la imagen

cubierta para ocultar uno de la imagen oculta, y se repetirá en los tres canales de forma redundante al igual que en la imagen blanco y negro.

- A Color: Al igual que la imagen oculta a escala de grises se propone que la imagen a color oculta sea ocho veces menor que la cubierta pero con la diferencia que los tres canales serán ocupados en su totalidad.

4.3. Mediciones de Calidad

El manejo de señales y el efecto de los mecanismos usados en ellas debe determinarse mediante medidas cuantitativas (métricas), esas medidas serán la PSNR (Relación pico de señal a ruido) y el SSIM (Índice de Similitud Estructural).

4.3.1. PSNR

La PSNR (por la siglas en inglés de Peak Signal to Noise Ratio) es la relación que hay entre la máxima potencia (el pico) de una señal y la potencia del ruido de la misma señal. Usualmente la medición de PSNR es utilizada como medida de calidad de imágenes reconstruidas generalmente por cuestiones de compresión y descompresión de imágenes, sin embargo como en este algoritmo trabaja pixel por pixel modificando su intensidad este tipo de medidas al comparar la imagen original con la modificada es de mucha ayuda para cuantificar la calidad del algoritmo (en escala dB) se calcula con la siguiente ecuación:

$$PSNR = 10\log_{10}\left[\frac{(255)^2}{\sqrt{MSE}}\right] \quad \ldots(1)$$

Donde MSE es el error cuadrático medio (por sus siglas en inglés) la cual es la medición que compara cada pixel de ambas imágenes y representa una medida

de distorsión de la imagen original al agregar el mensaje oculto por lo que se busca que sea pequeño para que el PSNR sea alto y se calcula con la siguiente ecuación:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[C(i,j) - S(i,j)] \quad \dots(2)$$

Donde *mn* es el tamaño de la imagen (*m x n*), *C* es la matriz de la imagen cubierta, *S* es la matriz de la estego-imagen e *i,j* representa el recorrido pixel por pixel de las imágenes que al estar siendo comparadas son evidentemente del mismo tamaño. Sus tamaños típicos y preferibles están entre 30 y 59 dB sin embargo puede ser mayor (A. Horé & D. Ziou, 2010).

4.3.2. SSIM

El Índice de Similitud Estructural (SSIM) cuantifica la degradación de imágenes causadas por procesamiento como compresión o pérdidas en transmisión de datos. Esta medida compara dos imágenes, una de referencia y otra procesada, evidentemente tienen que ser (visiblemente al menos) la misma imagen.

Esta medición comparativa es propuesta debido a es indicador más parecido al sistema de visión humana pues esta expresado como una función que depende de tres factores: Luminancia, Contraste y Similitud Estructural.

El SSIM puede obtenerse a través de la Siguiente ecuación:

$$SSIM(v,s) = \frac{(2\mu_v\mu_s + c_1)(2\sigma_{vs} + c_2)}{(\mu_v^2 + \mu_s^2 + c_1)(\sigma_v^2 + \sigma_s^2 + c_2)} \quad \dots(3)$$

Donde μ_v es el promedio de la matriz correspondiente a la imagen visible (v), μ_s es el promedio de la matriz correspondiente a la estegoimagen (s), σ_v es la varianza de la matriz de la imagen visible, σ_s es la varianza de la matriz de la estegoimagen y σ_{vs} es la covarianza que hay entre las matrices de las imágenes visible y estegoimagen (A. Horé & D. Ziou, 2010).

Capítulo 5 Resultados

En este capítulo se mostrarán los resultados obtenidos, es decir se mostrará la estegoimagen así como los valores numéricos correspondientes a PSNR y SSIM.

Los resultados obtenidos son en base a la comparación de dos pares de imágenes, el primer par corresponde a la imagen cubierta con la estegoimagen para que sea visible la similitud entre la imagen original y la imagen modificada, el segundo par corresponde al mensaje a esconder y el mensaje extraído ya que si se pierde información en el mensaje el resultado no es del todo satisfactorio (en el caso del primer par sirve para cuantificar su diferencia a nivel numérico y en el caso del segundo para comprobar que el mensaje enviado y el extraído son iguales).

Por ultimo al final del capítulo habrá una comparación de resultados entre una Raspberry Pi 2 y una Raspberry Pi 4ya que esta última supera en procesador y tiene una memoria RAM cuatro veces mayor, por lo que se comprobará si el programa funciona, que resultado obtiene y el tiempo en el que trabaja.

5.1. Resultados Blanco y Negro

Primero se mostrarán los resultados obtenidos de los mensajes en blanco y negro, para estos el mensaje es del mismo tamaño que la imagen cubierta (que en este trabajo están en formato bmp).

5.1.1 UACM.bmp (512x256)

El primer mensaje en blanco y negro que se repetirá en las portadoras cuyos tamaños son iguales: KC.bmp (512x256), SLT.bmp (512x256), RGB3.bmp (512x256), msg1.bmp (512x256), GB.bmp (512x256), FO.bmp (512x256) y LC.bmp (512x256).



FIGURA 21 MENSAJE 1 UACM.JPG (512X256)

5.1.1.1 KC.bmp (512x256)



FIGURA 22 CUBIERTA 1 KC.BMP (512X256)

5.1.1.1.1 Muestra de Resultados KC.bmp

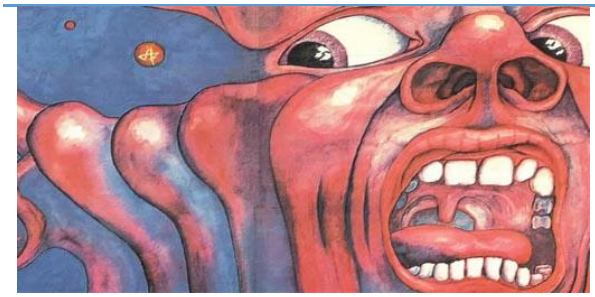


FIGURA 23 ESTEGOIMAGEN 1 SIM18 (512X256)



FIGURA 14 IMAGEN EXTRAÍDA 1 EIM18.BMP (512X256)

Estegoimagen: **SIM18.bmp**	Imagen extraída: **EIM18.bmp**
PSNR: **51.6498**	PSNR: **1**
SSIM: **0.9979382**	SSIM: **1**

5.1.1.2 SLT.bmp (512x256)



FIGURA 25 CUBIERTA 2 SLT.BMP (512X256)

5.1.1.2.1 Muestra de Resultados SLT.bmp



FIGURA 26 ESTEGOIMAGEN 1 SIM20.BMP (512X256)



FIGURA 27 IMAGEN EXTRAÍDA 1 EIM20.BMP (512X256)

Estegoimagen: **SIM20.bmp**	Imagen extraída: **EIM20.bmp**
PSNR: **51.6812**	PSNR: **1**
SSIM: **0.998364**	SSIM: **1**

5.1.1.3 RGB3.bmp (512x256)



FIGURA 28 CUBIERTA 3 RGB3.BMP (512X256)

5.1.1.3.1 Muestra de Resultados RGB3.bmp



FIGURA 29 ESTEGOIMAGEN 2 SIM21.BMP (512X256)



FIGURA 30 IMAGEN EXTRAÍDA 2 EIM21.BMP (512X256)

Estegoimagen: **SIM21.bmp**	Imagen extraída: **EIM21.bmp**
PSNR: **51.5060**	PSNR: **1**
SSIM: **0.9994518**	SSIM: **1**

5.1.1.4. msg1.bmp (512x256)



FIGURA 31 CUBIERTA 4 MSG1.BMP (512X256)

5.1.1.4.1 Muestra de Resultados msg1.bmp



FIGURA 32 ESTEGOIMAGEN 3 SIM63.BMP (512X256)



FIGURA 33 IMAGEN EXTRAÍDA 3 EIM63.BMP (512X256)

Estegoimagen: **SIM63.bmp**	Imagen extraída: **EIM63.bmp**
PSNR: **51.6286**	PSNR: **1**
SSIM: **0.998687**	SSIM: **1**

5.1.1.5. GB.bmp (512x256)



FIGURA 34 CUBIERTA 5 GB.BMP (512X256)

5.1.1.5.1 Muestra de Resultados GB.bmp



FIGURA 35 ESTEGOIMAGEN 4 SIM60.BMP (512X256)



FIGURA 36 IMAGEN EXTRAÍDA 4 EIM60.BMP (512X256)

Estegoimagen: **SIM60.bmp**	Imagen extraída: **EIM60.bmp**
PSNR: **51.7943**	PSNR: **1**
SSIM: **0.9975763**	SSIM: **1**

5.1.1.6. FO.bmp (512x256)



FIGURA 37 CUBIERTA 6 FO.BMP (256X512)

5.1.1.6.1 Muestra de Resultados FO.bmp

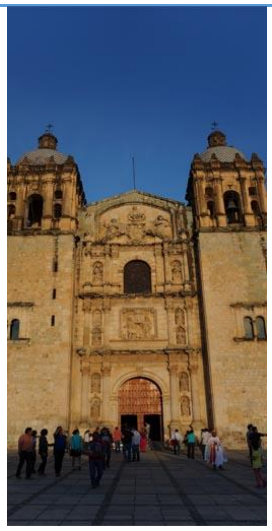


FIGURA 38 ESTEGOIMAGEN 5 SIM23.BMP (256X512)



FIGURA 39 IMAGEN EXTRAÍDA 5 EIM23.BMP (512X256)

Estegoimagen: **SIM23.bmp**	Imagen extraída: **EIM23.bmp**
PSNR: **51.7635**	PSNR: **1**
SSIM: **0.997951**	SSIM: **1**

5.1.1.7. LC.bmp (512x256)



FIGURA 40 CUBIERTA 7 LC.BMP (512X256)

5.1.1.7.1 Muestra de Resultados LC.bmp



FIGURA 41 ESTEGOIMAGEN 6 SIM33.BMP (512X256)



FIGURA 43 IMAGEN EXTRAÍDA 6 EIM33.BMP (512X256)

Estegoimagen: **SIM33.bmp**	Imagen extraída: **EIM33.bmp**
PSNR: **51.6680**	PSNR: **1**
SSIM: **0.9978685**	SSIM: **1**

5.1.2 NIIL.bmp

El segundo mensaje en blanco y negro, que se esconderá en las cubiertas de mismo tamaño: Lena.bmp (1024x1024), airplane1.bmp (1024x1024), baboon1.bmp (1024x1024), jellybeans.bmp (1024x1024), peppers1.bmp (1024x1024), sailboat1.bmp (1024x1024) y splash1.bmp (1024x1024).

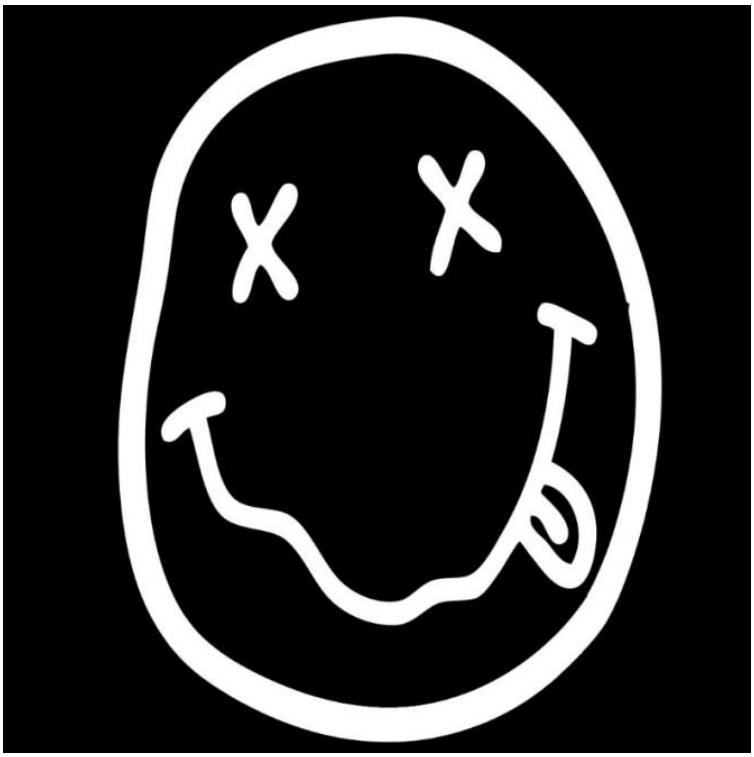


FIGURA 43 MENSAJE 2 NIIL.BMP (1024X1024)

5.1.2.1 Lena.bmp (1024x1024)

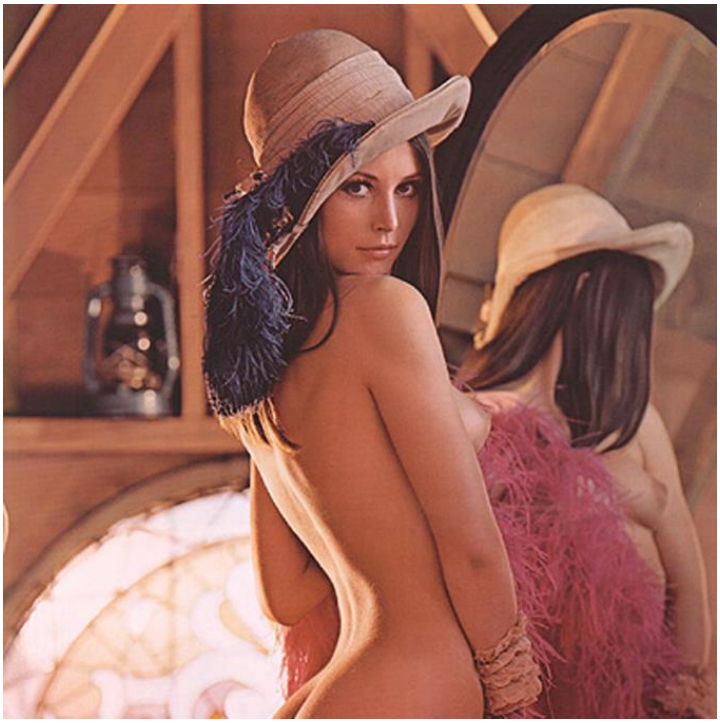


FIGURA 44 CUBIERTA 8 LENA.BMP (1024X1024)

5.1.2.1.1 Muestra de Resultados Lena.bmp

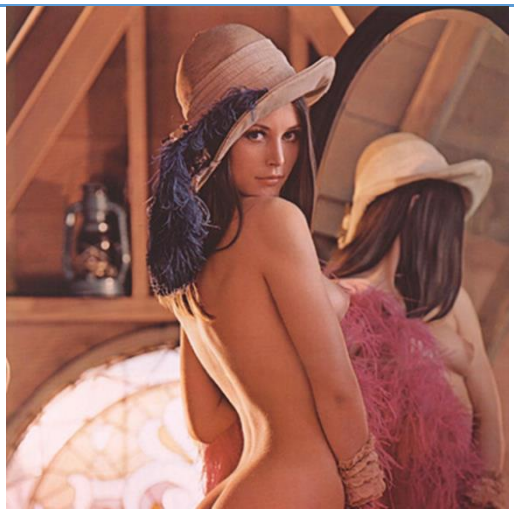


FIGURA 45 ESTEGOIMAGEN 7 SIM12.BMP (1024X1024)

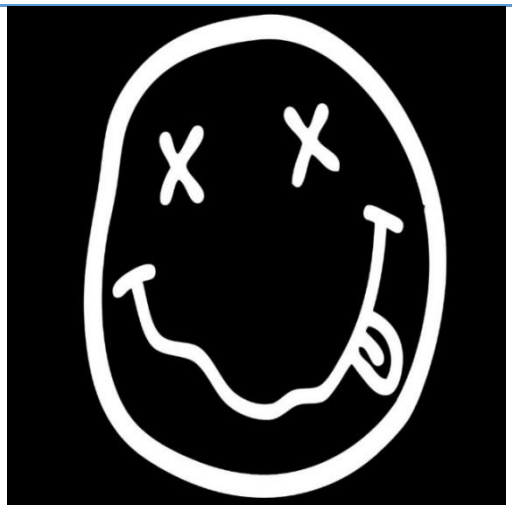


FIGURA 46 IMAGEN EXTRAÍDA 7 EIM12.BMP (1024X1024)

Estegoimagen: **SIM12.bmp**	Imagen extraída: **EIM12.bmp**
PSNR: **55.559**	PSNR: **1**
SSIM: **0.9977588**	SSIM: **1**

5.1.2.2 airplane1.bmp (1024x1024)



FIGURA 47 CUBIERTA 9 AIRPLANE1.BMP (1024X1024)

5.1.2.2.1 Muestra de Resultados airplane1.bmp



FIGURA 48 ESTEGOIMAGEN 8 SIM13.BMP (1024X1024)

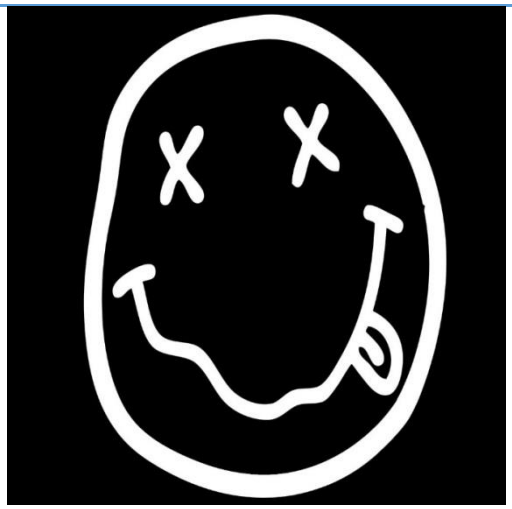


FIGURA 49 IMAGEN EXTRAÍDA 8 EIM13.BMP (1024X1024)

Estegoimagen: **SIM13.bmp**	Imagen extraída: **EIM13.bmp**
PSNR: **40.4899**	PSNR: **1**
SSIM: **0.9820831**	SSIM: **1**

5.1.2.3 baboon1.bmp (1024x1024)



FIGURA 50 CUBIERTA 10 BABOON1.BMP (1024X1024)

5.1.2.3.1 Muestra de Resultados baboon1.bmp

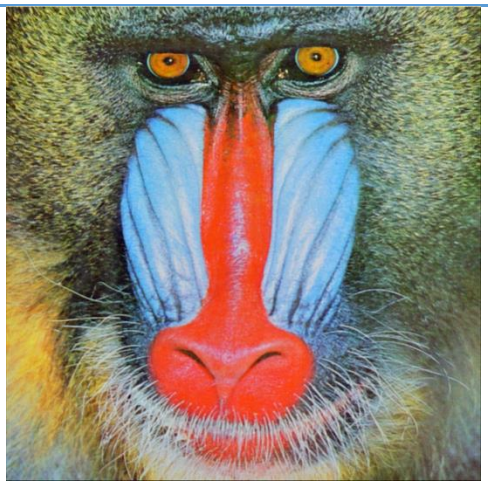


FIGURA 51 ESTEGOIMAGEN 9 SIM14.BMP (1024X1024)

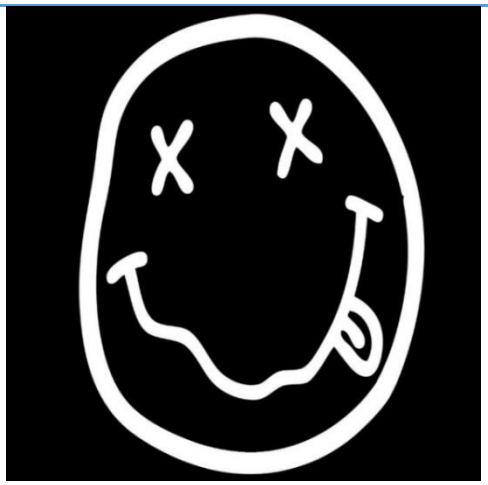


FIGURA 52 IMAGEN EXTRAÍDA 9 EIM14.BMP

Estegoimagen: **SIM14.bmp**	Imagen extraída: **EIM14.bmp**
PSNR: **34.2299**	PSNR: **1**
SSIM: **0.9832578**	SSIM: **1**

5.1.2.4 jellybeans.bmp (1024x1024)



FIGURA 53 CUBIERTA 11 JELLYBEANS.BMP (1024X1024)

5.1.2.3.1 Muestra de Resultados jellybeans.bmp



FIGURA 54 ESTEGOIMAGEN 10 SIM15.BMP (1024X1024)

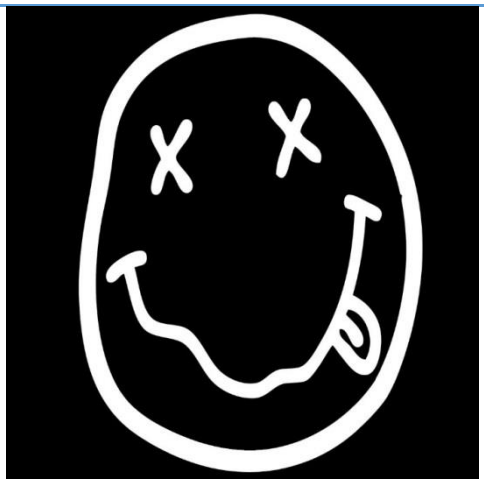


FIGURA 55 IMAGEN EXTRAÍDA 10 EIM15.BMP (1024X1024)

Estegoimagen: **SIM15.bmp**	Imagen extraída: **EIM15.bmp**
PSNR: **45.0494**	PSNR: **1**
SSIM: **0.9932919**	SSIM: **1**

5.1.2.5 peppers1.bmp (1024x1024)



FIGURA 56 CUBIERTA 12 PEPPERS1.BMP (1024X1024)

5.1.2.5.1 Muestra de Resultados peppers1.bmp



FIGURA 57 ESTEGOIMAGEN 11 SIM89.BMP (1024X1024)

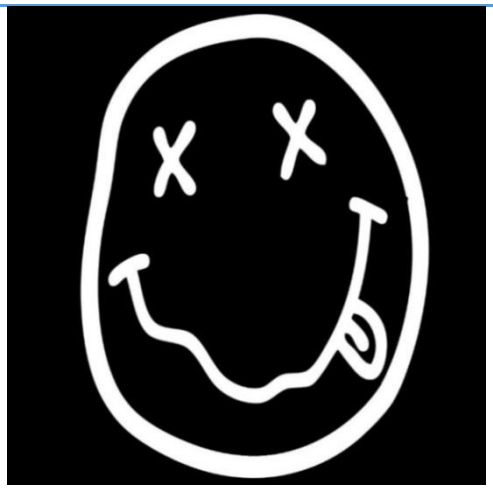


FIGURA 58 IMAGEN EXTRAÍDA 11 EIM89.BMP (1024X1024)

Estegoimagen: **SIM89.bmp**	Imagen extraída: **EIM89.bmp**
PSNR: **39.9219**	PSNR: **1**
SSIM: **0.9769036**	SSIM: **1**

5.1.2.6. sailboat1.bmp (1024x1024)



FIGURA 59 CUBIERTA 13 SAILBOAT.BMP (1024X1024)

5.1.2.6.1 Muestra de Resultados sailboat1.bmp



FIGURA 60 ESTEGOIMAGEN 12 SIM75.BMP (1024X1024)

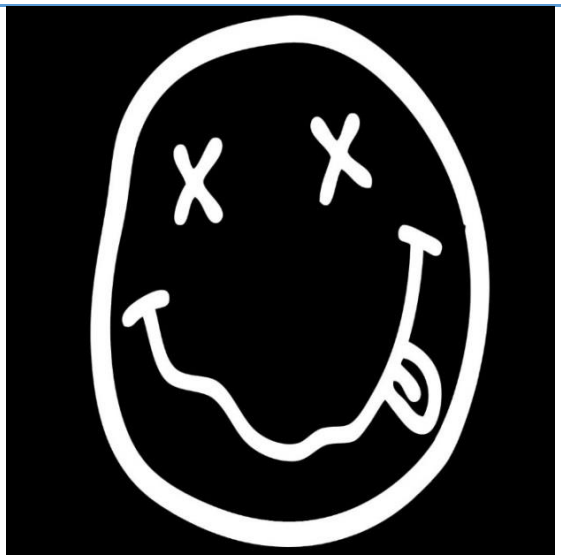


FIGURA 61 IMAGEN EXTRAÍDA 12 EIM75.BMP (1024X1024)

Estegoimagen: **SIM75.bmp**	Imagen extraída: **EIM75.bmp**
PSNR: **36.6749**	PSNR: **1**
SSIM: **0.9747266**	SSIM: **1**

5.1.2.7. splash1.bmp (1024x1024)

Esta será la última cubierta diferente, para las de más pruebas se repitieron las cubiertas de 1024x1024 (Lena.bmp, airplane1.bmp, baboon1.bmp, jellybeans.bmp, peppers1.bmp, sailboat1.bmp y splash1.bmp).

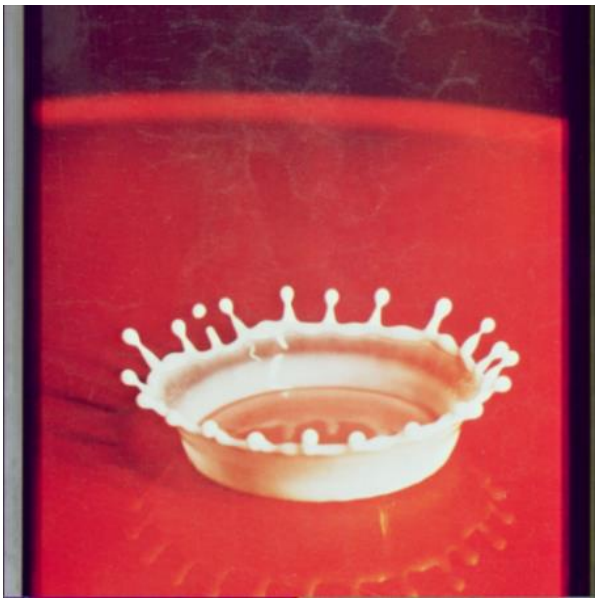


FIGURA 62 CUBIERTA 14 SPLASH1.BMP (1024X1024)

5.1.2.7.1 Muestra de Resultados splash1.bmp



FIGURA 63 ESTEGOIMAGEN 13 SIM0.BMP (1024X1024)

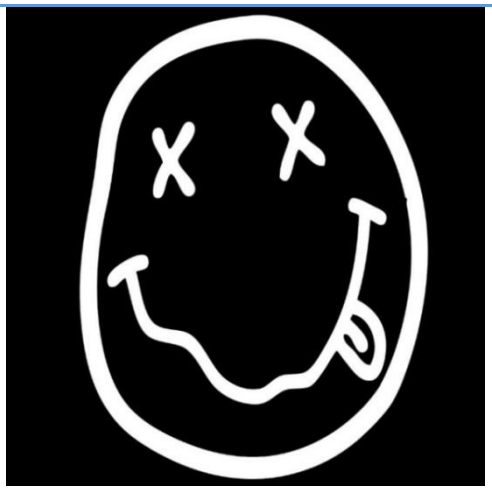


FIGURA 64 IMAGEN EXTRAÍDA 13 EIM0.BMP (1024X1024)

Estegoimagen: **SIM0.bmp**	Imagen extraída: **EIM0.bmp**
PSNR: **41.6246**	PSNR: **1**
SSIM: **0.9793685**	SSIM: **1**

5.1.3 awa.bmp

El tercer y último mensaje a Blanco y Negro en las pruebas hechas:



FIGURA 65 MENSAJE 3 AWA.BMP (1024X1024)

5.1.3.1 Lena.bmp

Por favor revise *figura 44* o en su defecto el punto *5.1.2.1 Lena.bmp.*

5.1.3.1.1 Muestra de Resultados Lena.bmp

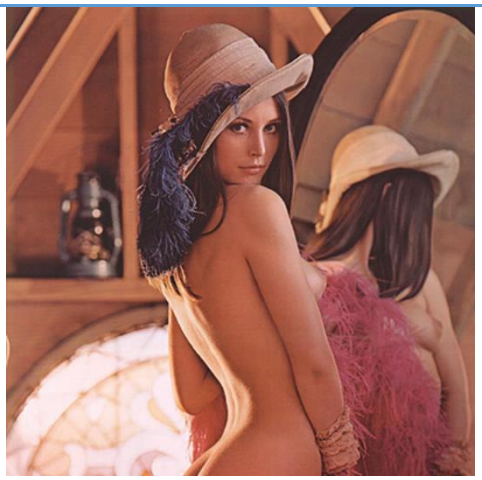


FIGURA 66 ESTEGOIMAGEN 14 SIM98.BMP (1024X1024)



FIGURA 67 IMAGEN EXTRAÍDA 2 EIM98.BMP (1024X1024)

Estegoimagen: **SIM98.bmp**	Imagen extraída: **EIM98.bmp**
PSNR: **53.4140**	PSNR: **1**
SSIM**: 0.99712**	SSIM: **1**

5.1.3.2 airplane1.bmp

Por favor revise *figura 47* o en su defecto el punto *5.1.2.2 airplane1.bmp.*

5.1.3.2.1 Muestra de Resultados airplane1.bmp



FIGURA 68 ESTEGOIMAGEN 15 SIM16.BMP (1024X1024)



FIGURA 69 IMAGEN EXTRAÍDA 3 EIM16.BMP (1024X1024)

Estegoimagen: **SIM16.bmp**	Imagen extraída: **EIM16.bmp**
PSNR: **40.4152**	PSNR: **1**
SSIM: **0.9817841**	SSIM: **1**

5.1.3.3 baboon1.bmp

Por favor revise *figura 50* o en su defecto el punto *5.1.2.3 baboon1.bmp.*

5.1.3.3.1 Muestra de Resultados baboon1.bmp

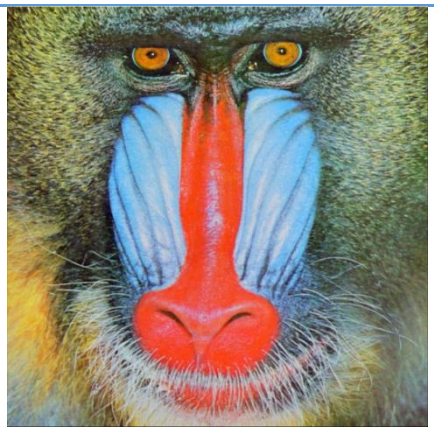


FIGURA 70 ESTEGOIMAGEN 16 SIM50.BMP (1024X1024)



FIGURA 71 IMAGEN EXTRAÍDA 4 EIM50.BMP (1024X1024)

Estegoimagen: **SIM50.bmp**	Imagen extraída: **EIM50.bmp**
PSNR: **40.4152**	PSNR: **1**
SSIM: **0.9817841**	SSIM: **1**

5.1.3.4 jellybeans.bmp

Por favor revise *figura 53* o en su defecto el punto *5.1.2.4 jellybeans.bmp.*

5.1.3.4.1 Muestra de Resultados jellybeans.bmp



FIGURA 72 ESTEGOIMAGEN 17 SIM17.BMP (1024X1024)



FIGURA 73 IMAGEN EXTRAÍDA 5 EIM17.BMP (1024X1024)

Estegoimagen: **SIM17.bmp**	Imagen extraída: **EIM17.bmp**
PSNR: **44.7773**	PSNR: **1**
SSIM: **0.9928271**	SSIM: **1**

5.1.3.5 peppers1.bmp

Por favor revise *figura 56* o en su defecto el punto *5.1.2.5 peppers1.bmp.*

5.1.3.5.1 Muestra de Resultados peppers1.bmp



FIGURA 74 ESTEGOIMAGEN 18 SIM69.BMP (1024X1024)



FIGURA 75 IMAGEN EXTRAÍDA 6 EIM69.BMP (1024X1024)

Estegoimagen: **SIM69.bmp**	Imagen extraída: **EIM69.bmp**
PSNR: **39.8658**	PSNR: **1**
SSIM: **0.9767524**	SSIM: **1**

5.1.3.6 sailboat1.bmp

Por favor revise *figura 59* o en su defecto el punto *5.1.2.6 saoilboat1.bmp.*

5.1.3.6.1 Muestra de Resultados sailboat1.bmp



FIGURA 76 ESTEGOIMAGEN 19 SIM39.BMP (1024X1024)



FIGURA 77 IMAGEN EXTRAÍDA 7 EIM39.BMP (1024X1024)

Estegoimagen: **SIM39.bmp**	Imagen extraída: **EIM39.bmp**
PSNR: **36.6463**	PSNR: **1**
SSIM: **0.9745986**	SSIM: **1**

5.1.3.7 splash1.bmp

Por favor revise *figura 62* o en su defecto el punto *5.1.2.7 splash1.bmp.*

5.1.3.7.1 Muestra de Resultados splash1.bmp

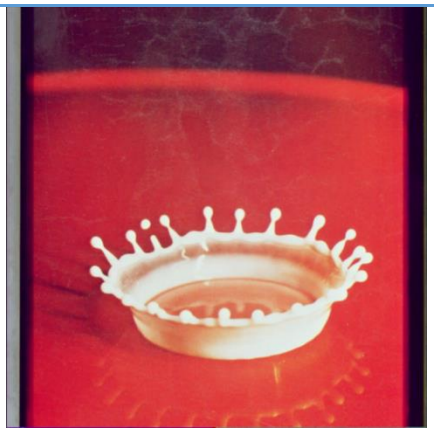


FIGURA 78 ESTEGOIMAGEN 20 SIM91.BMP (1024X1024)



FIGURA 79 IMAGEN EXTRAÍDA 8 EIM91.BMP (1024X1024)

Estegoimagen: **SIM39.bmp**	Imagen extraída: **EIM39.bmp**
PSNR: **41.5411**	PSNR: **1**
SSIM: **0.9791749**	SSIM: **1**

5.2. Resultados Escala de Grises

En segunda instancia se mostrarán los resultados al tener como mensajes imágenes en escala de grises, en esta ocasión los mensajes serán *1/2mx1/4n* o viceversa (es decir un octavo menor que *mxn*).

5.2.1 GS1.bmp

El primer mensaje en escala de grises:

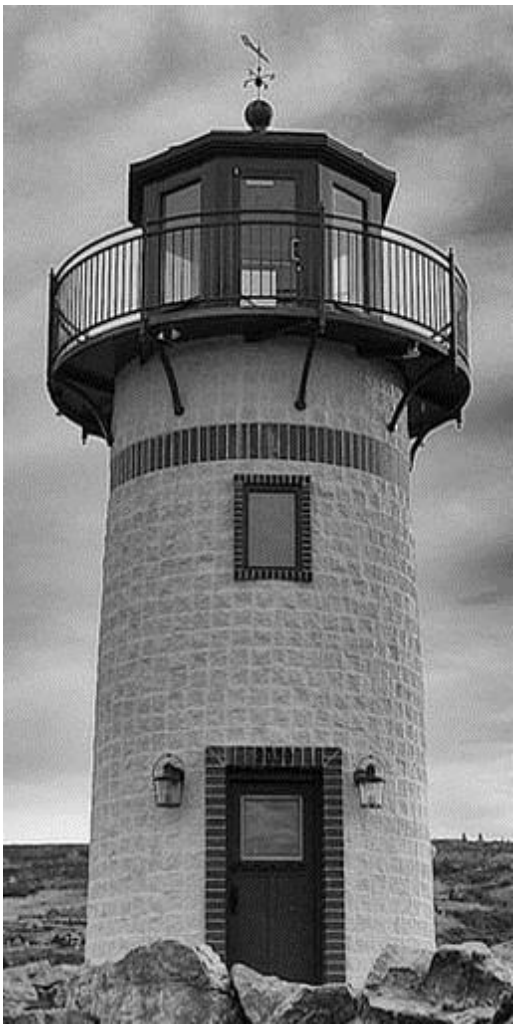


FIGURA 80 MENSAJE 4 GS1.BMP (256X512)

5.2.1.1 Lena.bmp

Por favor revise *figura 44* o en su defecto el punto *5.1.2.1 Lena.bmp.*

5.2.1.1.1 Muestra de Resultados Lena.bmp

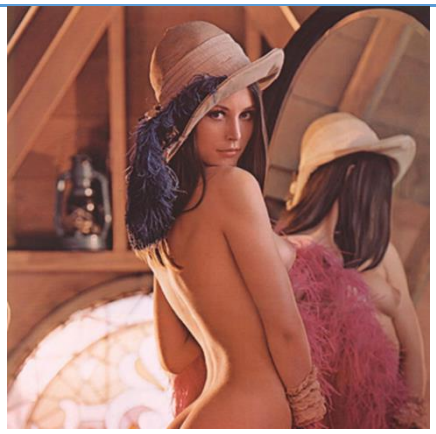


FIGURA 81 ESTEGOIMAGEN 21 SIM74.BMP (1024X1024)

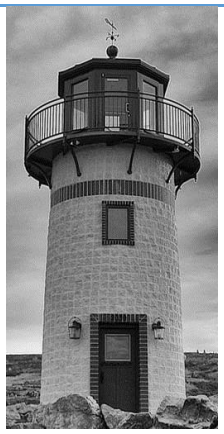


FIGURA 82 IMAGEN EXTRAÍDA 9 EIM74.BMP (256X512)

Estegoimagen: **SIM74.bmp**	Imagen extraída: **EIM74.bmp**
PSNR: **55.7942**	PSNR: **1**
SSIM: **0.9978189**	SSIM: **1**

5.2.1.2 airplane1.bmp

Por favor revise *figura 47* o en su defecto el punto *5.1.2.2 airplane1.bmp.*

5.2.1.2.1 Muestra de Resultados airplane1.bmp



FIGURA 83 ESTEGOIMAGEN 22 SIM76.BMP (1024X1024)

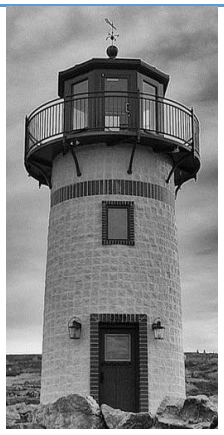


FIGURA 84 IMAGEN EXTRAÍDA 10 EIM76.BMP (256X512)

Estegoimagen: **SIM76.bmp**	Imagen extraída: **EIM76.bmp**
PSNR: **40.5214**	PSNR: **1**
SSIM: **0.9822458**	SSIM: **1**

5.2.1.3 baboon1.bmp

Por favor revise *figura 50* o en su defecto el punto *5.1.2.3 baboon1.bmp.*

5.2.1.3.1 Muestra de Resultados baboon1.bmp

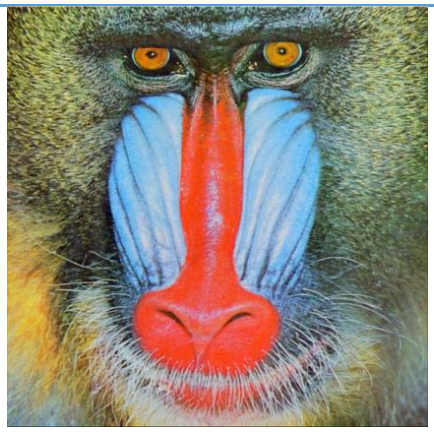


FIGURA 85 ESTEGOIMAGEN 23 SIM2.BMP (1024X1024)

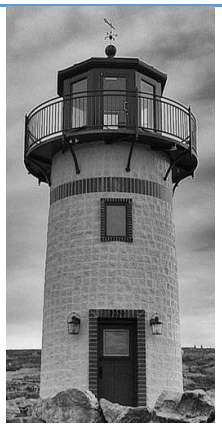


FIGURA 86 IMAGEN EXTRAÍDA 23 EIM2.BMP (256X512)

Estegoimagen: **SIM2.bmp**	Imagen extraída: **EIM2.bmp**
PSNR: **34.2349**	PSNR: **1**
SSIM: **0.9832782**	SSIM: **1**

5.2.1.4 jellybeans.bmp

Por favor revise *figura 53* o en su defecto el punto *5.1.2.4 jellybeans.bmp.*

5.2.1.4.1 Muestra de Resultados jellybeans.bmp



FIGURA 87 ESTEGOIMAGEN 24 SIM9.BMP (1024X1024)

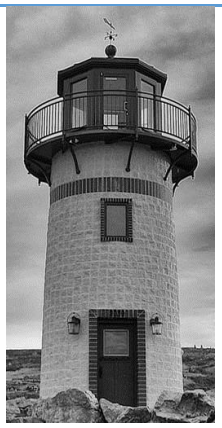


FIGURA 88 IMAGEN EXTRAÍDA 11 EIM9.BMP (256X512)

Estegoimagen: **SIM9.bmp**	Imagen extraída: **EIM9.bmp**
PSNR: **45.0708**	PSNR: **1**
SSIM: **0.9933607**	SSIM: **1**

5.2.1.5 peppers1.bmp

Por favor revise *figura 56* o en su defecto el punto *5.1.2.5 peppers1.bmp.*

5.2.1.5.1 Muestra de Resultados peppers1.bmp



FIGURA 89 ESTEGOIMAGEN 25 SIM83.BMP (1024X1024)

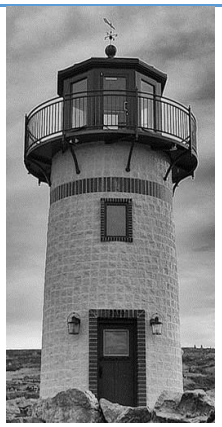


FIGURA 90 IMAGEN EXTRAÍDA 125 EIM83.BMP (256X512)

Estegoimagen: **SIM83.bmp**	Imagen extraída: **EIM83.bmp**
PSNR: **39.9480**	PSNR: **1**
SSIM: **0.9769805**	SSIM: **1**

5.2.1.6 sailboat1.bmp

Por favor revise *figura 59* o en su defecto el punto *5.1.2.6 sailboat1.bmp.*

5.2.1.6.1 Muestra de Resultados sailboat1.bmp

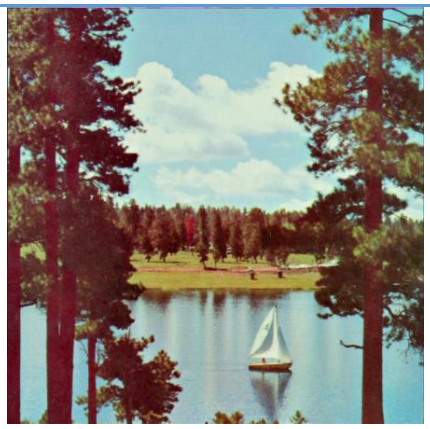


FIGURA 91 ESTEGOIMAGEN 26 SIM80.BMP (1024X1024)

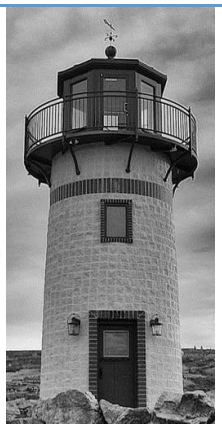


FIGURA 92 IMAGEN EXTRAÍDA 136 EIM80.BMP (256X512)

Estegoimagen: **SIM83.bmp**	Imagen extraída: **EIM83.bmp**
PSNR: **36.6881**	PSNR: **1**
SSIM: **0.9747775**	SSIM: **1**

5.2.1.7 splash1.bmp

Por favor revise *figura 62* o en su defecto el punto *5.1.2.7 splash1.bmp.*

5.2.1.7.1 Muestra de Resultados splash1.bmp

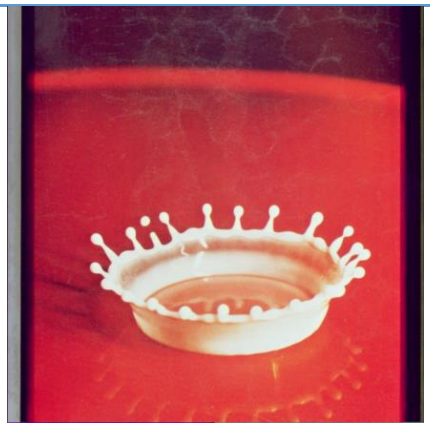


FIGURA 93 ESTEGOIMAGEN 27 SIM85.BMP (1024X1024)

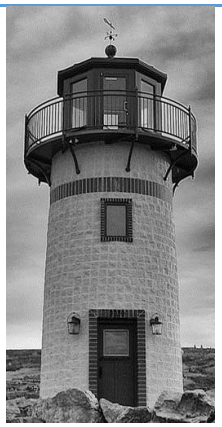


FIGURA 94 IMAGEN EXTRAÍDA 147 EIM85.JPG (256X512)

Estegoimagen: **SIM85.bmp**	Imagen extraída: **EIM85.bmp**
PSNR: **41.6653**	PSNR: **1**
SSIM: **0.9795032**	SSIM: **1**

5.2.2 GS2.bmp

El segundo mensaje en escala de grises:



FIGURA 95 MENSAJE 5 GS2.BMP (512X256)

5.2.2.1 Lena.bmp

Por favor revise *figura 44* o en su defecto el punto *5.1.2.1 Lena.bmp.*

5.2.2.1.1 Muestra de Resultados Lena.bmp

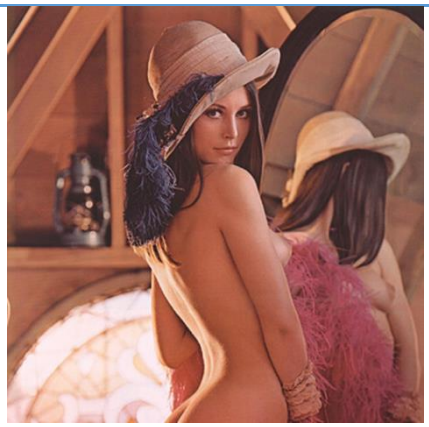


FIGURA 96 ESTEGOIMAGEN 28 SIM87.BMP (1024X1024)



FIGURA 97 IMAGEN EXTRAÍDA 158 EIM87.BMP (512X256)

Estegoimagen: **SIM87.bmp**	Imagen extraída: **EIM87.bmp**
PSNR: **55.0639**	PSNR: **1**
SSIM: **0.9976084**	SSIM: **1**

5.2.2.2 airplane1.bmp

Por favor revise *figura 47* o en su defecto el punto *5.1.2.2 airplane1.bmp.*

5.2.2.2.1 Muestra de Resultados airplane1.bmp



FIGURA 98 ESTEGOIMAGEN 29 SIM3.BMP (1024X1024)

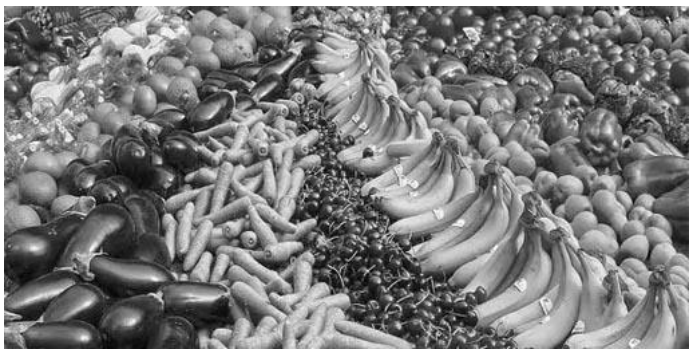


FIGURA 99 IMAGEN EXTRAÍDA 29 EIM3.BMP (512X256)

Estegoimagen: **SIM3.bmp**	Imagen extraída: **EIM3.bmp**
PSNR: **40.5143**	PSNR: **1**
SSIM: **0.9822092**	SSIM: **1**

5.2.2.3 baboon1.bmp

Por favor revise *figura 50* o en su defecto el punto *5.1.2.3 baboon1.bmp.*

5.2.2.3.1 Muestra de Resultados baboon1.bmp

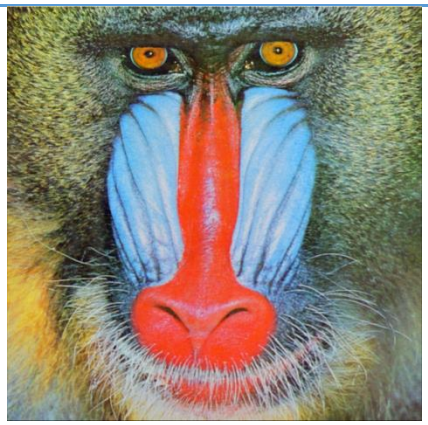


FIGURA 100 ESTEGOIMAGEN 1 SIM1.BMP (1024X1024)



FIGURA 101 IMAGEN EXTRAÍDA 3016 EIM1.BMP (512X256)

Estegoimagen: **SIM1.bmp**	Imagen extraída: **EIM1.bmp**
PSNR: **34.2347**	PSNR: **1**
SSIM: **0.9832706**	SSIM: **1**

5.2.2.4 jellybeans.bmp

Por favor revise *figura 53* o en su defecto el punto *5.1.2.4 jellybeans.bmp.*

5.2.2.4.1 Muestra de Resultados jellybeans.bmp



FIGURA 102 ESTEGOIMAGEN 30 SIM10.BMP (1024X1024)



FIGURA 103 IMAGEN EXTRAÍDA 3117 EIM10.BMP (512X256)

Estegoimagen: **SIM10.bmp**	Imagen extraída: **EIM10.bmp**
PSNR: **45.0000**	PSNR: **1**
SSIM: **0.993163**	SSIM: **1**

5.2.2.5 peppers1.bmp

Por favor revise *figura 56* o en su defecto el punto *5.1.2.5 peppers1.bmp.*

5.2.2.5.1 Muestra de Resultados peppers1.bmp



FIGURA 104 ESTEGOIMAGEN 31 SIM5.BMP (1024X1024)



FIGURA 105 IMAGEN EXTRAÍDA 32 EIM5.BMP (512X256)

Estegoimagen: **SIM5.bmp**	Imagen extraída: **EIM5.bmp**
PSNR: **39.9426**	PSNR: **1**
SSIM: **0.9769618**	SSIM: **1**

5.2.2.6 sailboat1.bmp

Por favor revise *figura 59* o en su defecto el punto *5.1.2.6 sailboat1.bmp.*

5.2.2.6.1 Muestra de Resultados sailboat1.bmp



FIGURA 106 ESTEGOIMAGEN 32 SIM11.BMP (1024X1024)



FIGURA 107 IMAGEN EXTRAÍDA 3318 EIM11.BMP (512X256)

Estegoimagen: **SIM11.bmp**	Imagen extraída: **EIM11.bmp**
PSNR: **36.6773**	PSNR: **1**
SSIM: **0.9747253**	SSIM: **1**

5.2.2.7 splash1.bmp

Por favor revise *figura 62* o en su defecto el punto *5.1.2.7 splash1.bmp.*

5.2.2.7.1 Muestra de Resultados splash1.bmp

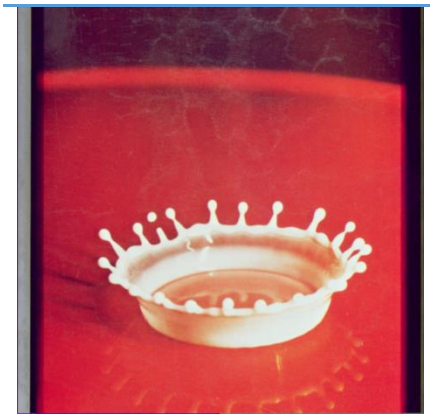


FIGURA 108 ESTEGOIMAGEN 33 SIM36.BMP (1024X1024)

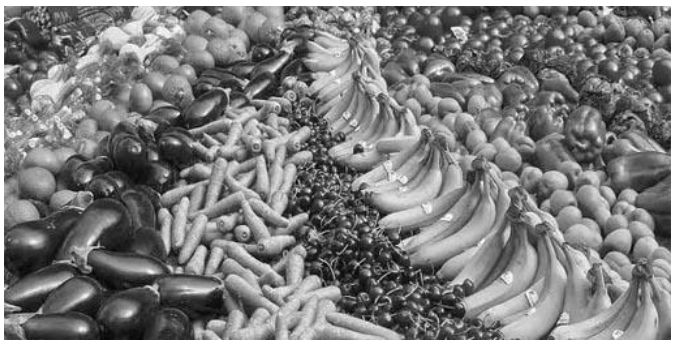


FIGURA 109 IMAGEN EXTRAÍDA 194 EIM36.JPG (512X256)

Estegoimagen: **SIM36.bmp**	Imagen extraída: **EIM36.bmp**
PSNR: **41.6573**	PSNR: **1**
SSIM: **0.9794883**	SSIM: **1**

5.2.3 GS3.bmp

El último mensaje en escala de grises:



FIGURA 110 MENSAJE 6 GS3.BMP (512X256)

5.2.3.1 Lena.bmp

Por favor revise *figura 44* o en su defecto el punto *5.1.2.1 Lena.bmp.*

5.2.3.1.1 Muestra de Resultados Lena.bmp

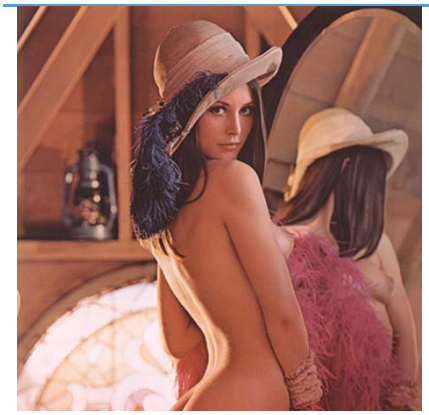


FIGURA 111 ESTEGOIMAGEN 34 SIM44.BMP (1024X1024)



FIGURA 112 IMAGEN EXTRAÍDA 35 EIM44.BMP (512X256)

Estegoimagen: **SIM44.bmp**	Imagen extraída: **EIM44.bmp**
PSNR: **55.6071**	PSNR: **1**
SSIM: **0.9977732**	SSIM: **1**

5.2.3.2 airplane1.bmp

Por favor revise *figura 47* o en su defecto el punto *5.1.2.2 airplane1.bmp.*

5.2.3.2.1 Muestra de Resultados airplane1.bmp



FIGURA 113 ESTEGOIMAGEN 35 SIM48.BMP (1024X1024)



FIGURA 114 IMAGEN EXTRAÍDA 36 EIM48.BMP (512X256)

Estegoimagen: **SIM48.bmp**	Imagen extraída: **EIM48.bmp**
PSNR: **40.5143**	PSNR: **1**
SSIM: **0.9822092**	SSIM: **1**

5.2.3.3 baboon1.bmp

Por favor revise *figura 50* o en su defecto el punto *5.1.2.3 baboon1.bmp.*

5.2.3.3.1 Muestra de Resultados baboon1.bmp

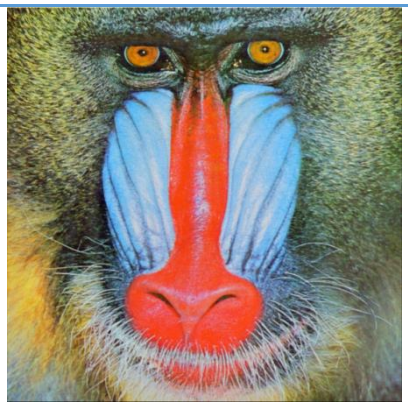


FIGURA 115 ESTEGOIMAGEN 36 SIM95.BMP (1024X1024)



FIGURA 116 IMAGEN EXTRAÍDA 37 EIM95.BMP (512X256)

Estegoimagen: **SIM95.bmp**	Imagen extraída: **EIM95.bmp**
PSNR: **34.2347**	PSNR: **1**
SSIM: **0.9832706**	SSIM: **1**

5.2.3.4 jellybeans.bmp

Por favor revise *figura 53* o en su defecto el punto *5.1.2.4 jellybeans.bmp.*

5.2.3.4.1 Muestra de Resultados jellybeans.bmp



FIGURA 117 ESTEGOIMAGEN 37 SIM30.BMP (1024X1024)



FIGURA 118 IMAGEN EXTRAÍDA 208 EIM30.BMP (512X256)

Estegoimagen: **SIM30.bmp**	Imagen extraída: **EIM30.bmp**
PSNR: **45.0538**	PSNR: **1**
SSIM: **0.9933065**	SSIM: **1**

5.2.3.5 peppers1.bmp

Por favor revise *figura 56* o en su defecto el punto *5.1.2.5 peppers1.bmp.*

5.2.3.5.1 Muestra de Resultados peppers1.bmp



FIGURA 119 ESTEGOIMAGEN 38 SIM31.BMP (1024X1024)



FIGURA 120 IMAGEN EXTRAÍDA 39 EIM31.BMP (512X256)

Estegoimagen: **SIM31.bmp**	Imagen extraída: **EIM31.bmp**
PSNR: **39.9426**	PSNR: **1**
SSIM: **0.9769618**	SSIM: **1**

5.2.3.6 sailboat1.bmp

Por favor revise *figura 59* o en su defecto el punto *5.1.2.6 sailboat1.bmp.*

5.2.3.6.1 Muestra de Resultados sailboat1.bmp



FIGURA 121 ESTEGOIMAGEN 39 SIM76.BMP (1024X1024)



FIGURA 122 IMAGEN EXTRAÍDA 21 EIM76.BMP (512X256)

Estegoimagen: **SIM76.bmp**	Imagen extraída: **EIM76.bmp**
PSNR: **36.6855**	PSNR: **1**
SSIM: **0.9747624**	SSIM: **1**

5.2.3.7 splash1.bmp

Por favor revise *figura 62* o en su defecto el punto *5.1.2.7 splash1.bmp.*

5.2.3.7.1 Muestra de Resultados splash1.bmp

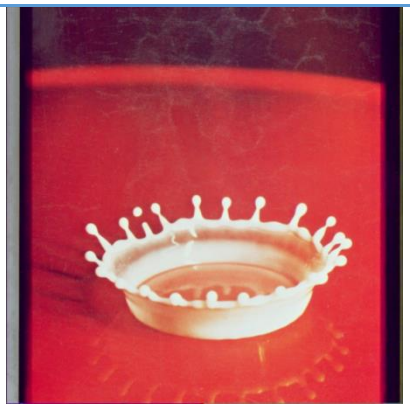


FIGURA 123 ESTEGOIMAGEN 40 SIM77.BMP (1024X1024)



FIGURA 124 IMAGEN EXTRAÍDA 41 EIM77.BMP (512X256)

Estegoimagen: **SIM77.bmp**	Imagen extraída: **EIM77.bmp**
PSNR: **41.6573**	PSNR: **1**
SSIM: **0.9794883**	SSIM: **1**

5.3. Resultados RGB

En la última instancia se mostrarán los resultados al tener como mensajes imágenes de color (por canales RGB) y de igual manera que con los mensajes de escala de grises tendrá un tamaño *1/2mx1/4n* o viceversa (es decir un octavo menor que *mxn*).

5.3.1 SLT.bmp

El primer mensaje en RGB:



FIGURA 125 MENSAJE 7 SLT.BMP (512X256)

5.3.1.1 Lena.bmp

Por favor revise *figura 44* o en su defecto el punto *5.1.2.1 Lena.bmp.*

5.3.1.1.1 Muestra de Resultados Lena.bmp

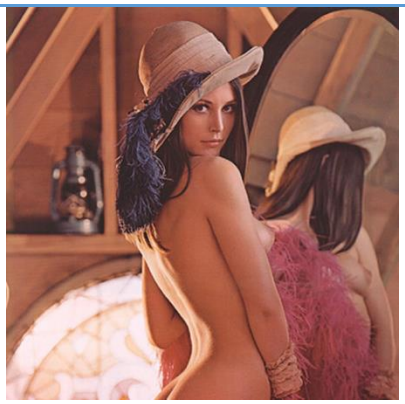


FIGURA 126 ESTEGOIMAGEN 41 SIM59.BMP (1024X1024)



FIGURA 127 IMAGEN EXTRAÍDA 42 EIM59.BMP (512X256)

Estegoimagen: **SIM59.bmp**	Imagen extraída: **EIM52.bmp**
PSNR: **55.559**	PSNR: **1**
SSIM: **0.9977588**	SSIM: **1**

5.3.1.2 airplane1.bmp

Por favor revise *figura 47* o en su defecto el punto *5.1.2.2 airplane1.bmp.*

5.3.1.2.1 Muestra de Resultados airplane1.bmp



FIGURA 128 ESTEGOIMAGEN 42 SIM4.BMP (1024X1024)



FIGURA 129 IMAGEN EXTRAÍDA 43 EIM4.BMP (512X256)

Estegoimagen: **SIM4.bmp**	Imagen extraída: **EIM4.bmp**
PSNR: **40.5124**	PSNR: **1**
SSIM: **0.9822005**	SSIM: **1**

5.3.1.3 baboon1.bmp

Por favor revise *figura 50* o en su defecto el punto *5.1.2.3 baboon1.bmp.*

5.3.1.3.1 Muestra de Resultados baboon1.bmp

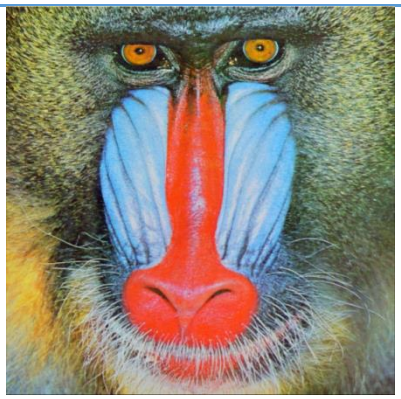


FIGURA 130 ESTEGOIMAGEN 43 SIM97.BMP (1024X1024)



FIGURA 131 IMAGEN EXTRAÍDA 44 EIM97.BMP (512X256)

Estegoimagen: **SIM97.bmp**	Imagen extraída: **EIM97.bmp**
PSNR: **34.234**	PSNR: **1**
SSIM: **0.9832686**	SSIM: **1**

5.3.1.4 jellybeans.bmp

Por favor revise *figura 53* o en su defecto el punto *5.1.2.4 jellybeans.bmp.*

5.3.1.4.1 Muestra de Resultados jellybeans.bmp



FIGURA 132 ESTEGOIMAGEN 44 SIM62.BMP (1024X1024)



FIGURA 133 IMAGEN EXTRAÍDA 45 EIM62.BMP (512X256)

Estegoimagen: **SIM62.bmp**	Imagen extraída: **EIM62.bmp**
PSNR: **45.0494**	PSNR: **1**
SSIM: **0.9932919**	SSIM: **1**

5.3.1.5 peppers1.bmp

Por favor revise *figura 56* o en su defecto el punto *5.1.2.5 peppers1.bmp*

5.3.1.5.1 Muestra de Resultados peppers1.bmp



FIGURA 134 ESTEGOIMAGEN 45 SIM8.BMP (1024X1024)



FIGURA 135 IMAGEN EXTRAÍDA 46 EIM8.BMP (512X256)

Estegoimagen: **SIM8.bmp**	Imagen extraída: **EIM8.bmp**
PSNR: **39.9421**	PSNR: **1**
SSIM: **0.9769537**	SSIM: **1**

5.3.1.6 sailboat1.bmp

Por favor revise *figura 59* o en su defecto el punto *5.1.2.6 sailboat1.bmp.*

5.3.1.6.1 Muestra de Resultados sailboat1.bmp



FIGURA 136 ESTEGOIMAGEN 46 SIM7.BMP (1024X1024)



FIGURA 137 IMAGEN EXTRAÍDA 47 EIM7.BMP (512X256)

Estegoimagen: **SIM7.bmp**	Imagen extraída: **EIM7.bmp**
PSNR: **36.6847**	PSNR: **1**
SSIM: **0.9747631**	SSIM: **1**

5.3.1.7 splash1.bmp

Por favor revise *figura 62* o en su defecto el punto *5.1.2.7 splash1.bmp.*

5.3.1.7.1 Muestra de Resultados splash1.bmp

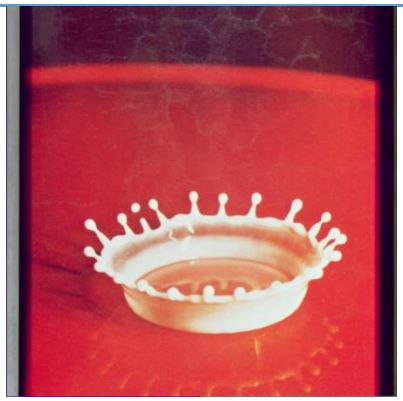


FIGURA 138 ESTEGOIMAGEN 47 SIM81.BMP (1024X1024)



FIGURA 139 IMAGEN EXTRAÍDA 48 EIM81.BMP (512X256)

Estegoimagen: **SIM81.bmp**	Imagen extraída: **EIM81.bmp**
PSNR: **41.6562**	PSNR: **1**
SSIM: **0.9794844**	SSIM: **1**

5.3.2 KC.jpg

El segundo mensaje en RGB:



FIGURA 140 MENSAJE 8 KC.BMP (512X256)

5.3.2.1 Lena.bmp

Por favor revise *figura 44* o en su defecto el punto *5.1.2.1 Lena.bmp.*

5.3.2.1.1 Muestra de Resultados Lena.bmp

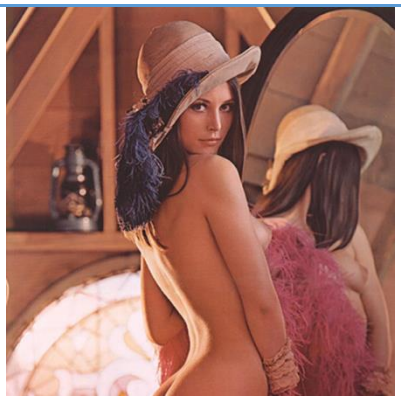


FIGURA 141 ESTEGOIMAGEN 48 SIM64.BMP (1024X1024)



FIGURA 142 IMAGEN EXTRAÍDA 49 EIM64.BMP (512X256)

Estegoimagen: **SIM64.bmp**	Imagen extraída: **EIM64.bmp**
PSNR: **55.6052**	PSNR: **1**
SSIM: **0.9977698**	SSIM: **1**

5.3.2.2 airplane1.bmp

Por favor revise *figura 47* o en su defecto el punto *5.1.2.2 airplane1.bmp.*

5.3.2.2.1 Muestra de Resultados airplane1.bmp



FIGURA 143 ESTEGOIMAGEN 49 SIM42.BMP (1024X1024)



FIGURA 144 IMAGEN EXTRAÍDA 50 EIM42.BMP (512X256)

Estegoimagen: **SIM42.bmp**	Imagen extraída: **EIM42.bmp**
PSNR: **40.5149**	PSNR: **1**
SSIM: **0.9822199**	SSIM: **1**

5.3.2.3 baboon1.bmp

Por favor revise *figura 50* o en su defecto el punto *5.1.2.3 baboon1.bmp.*

5.3.2.3.1 Muestra de Resultados baboon1.bmp

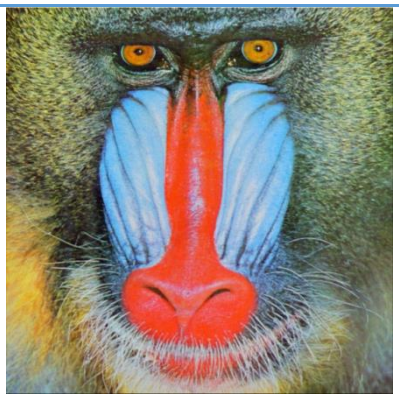


FIGURA 145 ESTEGOIMAGEN 50 SIM51.BMP (1024X1024)



FIGURA 146 IMAGEN EXTRAÍDA 51 EIM51.BMP (512X256)

Estegoimagen: **SIM51.bmp**	Imagen extraída: **EIM51.bmp**
PSNR: **34.2350**	PSNR: **1**
SSIM: **0.9832678**	SSIM: **1**

5.3.2.4 jellybeans.bmp

Por favor revise *figura 53* o en su defecto el punto *5.1.2.4 jellybeans.bmp.*

5.3.2.4.1 Muestra de Resultados jellybeans.bmp



FIGURA 147 ESTEGOIMAGEN 51 SIM40.BMP (1024X1024)



FIGURA 148 IMAGEN EXTRAÍDA 52 EIM40.BMP (512X256)

Estegoimagen: **SIM40.bmp**	Imagen extraída: **EIM40.bmp**
PSNR: **45.0534**	PSNR: **1**
SSIM: **0.9933022**	SSIM: **1**

5.3.2.5 peppers1.bmp

Por favor revise *figura 56* o en su defecto el punto *5.1.2.5 peppers1.bmp.*

5.3.2.5.1 Muestra de Resultados peppers1.bmp



FIGURA 149 ESTEGOIMAGEN 52 SIM55.BMP (1024X1024)



FIGURA 150 IMAGEN EXTRAÍDA 53 EIM55.BMP (512X256)

Estegoimagen: **SIM55.bmp**	Imagen extraída: **EIM55.bmp**
PSNR: **39.9420**	PSNR: **1**
SSIM: **0.9769575**	SSIM: **1**

5.3.2.6 sailboat1.bmp

Por favor revise *figura 59* o en su defecto el punto *5.1.2.6 sailboat1.bmp.*

5.3.2.6.1 Muestra de Resultados sailboat1.bmp



FIGURA 151 ESTEGOIMAGEN 53 SIM27.BMP (1024X1024)



FIGURA 152 IMAGEN EXTRAÍDA 54 EIM27.BMP (512X256)

Estegoimagen: **SIM27.bmp**	Imagen extraída: **EIM27.bmp**
PSNR: **36.6860**	PSNR: **1**
SSIM: **0.9747721**	SSIM: **1**

5.3.2.7 splash1.bmp

Por favor revise *figura 62* o en su defecto el punto *5.1.2.7 splash1.bmp.*

5.3.2.7.1 Muestra de Resultados splash1.bmp

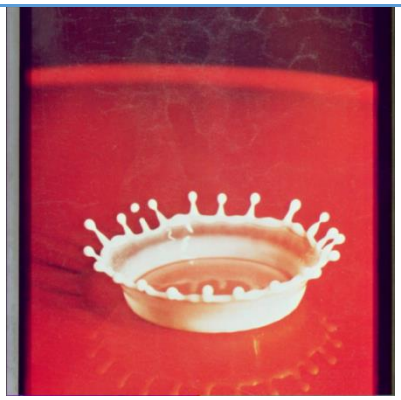


FIGURA 153 ESTEGOIMAGEN 54 SIM37.BMP (1024X1024)



FIGURA 154 IMAGEN EXTRAÍDA 55 EIM37.BMP (512X256)

Estegoimagen: **SIM37.bmp**	Imagen extraída: **EIM37.bmp**
PSNR: **41.6579**	PSNR: **1**
SSIM: **0.9794797**	SSIM: **1**

5.3.3 RGB3.bmp

El último mensaje en RGB:



FIGURA 155 MENSAJE 9 RGB3.BMP (512X256)

5.3.3.1 Lena.bmp

Por favor revise *figura 44* o en su defecto el punto *5.1.2.1 Lena.bmp.*

5.3.3.1.1 Muestra de Resultados Lena.bmp

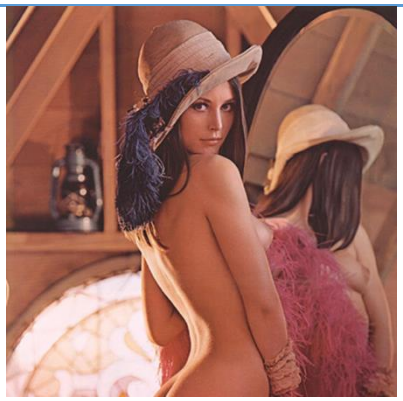


FIGURA 156 ESTEGOIMAGEN 55 SIM41.BMP (1024X1024)



Figura 157 Imagen Extraída 56 EIM41.bmp (512x256)

Estegoimagen: **SIM41.bmp**	Imagen extraída: **EIM41.bmp**
PSNR: **54.8055**	PSNR: **1**
SSIM: **0.9975276**	SSIM: **1**

5.3.3.2 airplane1.bmp

Por favor revise *figura 47* o en su defecto el punto *5.1.2.2 airplane1.bmp.*

5.3.3.2.1 Muestra de Resultados airplane1.bmp



FIGURA 158 ESTEGOIMAGEN 56 SIM96.BMP (1024X1024)



Figura 159 Imagen Extraída 57 EIM96.bmp (512x256)

Estegoimagen: **SIM96.bmp**	Imagen extraída: **EIM96.bmp**
PSNR: **40.4879**	PSNR: **1**
SSIM: **0.9820434**	SSIM: **1**

5.3.3.3 baboon1.bmp

Por favor revise *figura 50* o en su defecto el punto *5.1.2.3 baboon1.bmp.*

5.3.3.3.1 Muestra de Resultados baboon1.bmp

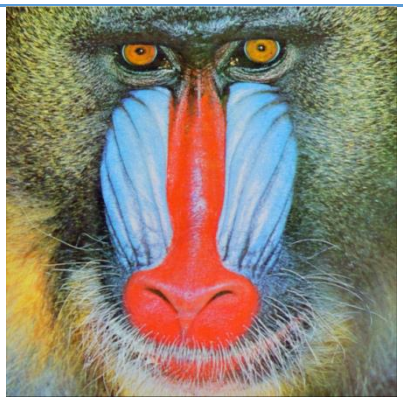


FIGURA 160 ESTEGOIMAGEN 57 SIM94.BMP (1024X1024)



Figura 161 Imagen Extraída 58 EIM94.bmp (512x256)

Estegoimagen: **SIM94.bmp**	Imagen extraída: **EIM94.bmp**
PSNR: **34.2300**	PSNR: **1**
SSIM: **0.9832541**	SSIM: **1**

5.3.3.4 jellybeans.bmp

Por favor revise *figura 53* o en su defecto el punto *5.1.2.4 jellybeans.bmp*

5.3.3.4.1 Muestra de Resultados jellybeans.bmp



FIGURA 162 ESTEGOIMAGEN 58 SIM22.BMP (1024X1024)



Figura 163 Imagen Extraída 59 EIM22.bmp (512x256)

Estegoimagen: **SIM22.bmp**	Imagen extraída: **EIM22.bmp**
PSNR: **44.9750**	PSNR: **1**
SSIM: **0.9930992**	SSIM: **1**

5.3.3.5 peppers1.bmp

Por favor revise *figura 56* o en su defecto el punto *5.1.2.5 peppers1.bmp.*

5.3.3.5.1 Muestra de Resultados peppers1.bmp



FIGURA 164 ESTEGOIMAGEN 59 SIM26.BMP (1024X1024)



Figura 165 Imagen Extraída 60 EIM26.bmp (512x256)

Estegoimagen: **SIM26.bmp**	Imagen extraída: **EIM26.bmp**
PSNR: **39.9209**	PSNR: **1**
SSIM: **0.9768864**	SSIM: **1**

5.3.3.6 sailboat1.bmp

Por favor revise *figura 59* o en su defecto el punto *5.1.2.6 sailboat1.bmp.*

5.3.3.6.1 Muestra de Resultados sailboat1.bmp



FIGURA 166 ESTEGOIMAGEN 60 SIM67.BMP (1024X1024)



Figura 167 Imagen Extraída 61 EIM67.bmp (512x256)

Estegoimagen: **SIM67.bmp**	Imagen extraída: **EIM67.bmp**
PSNR: **36.6746**	PSNR: **1**
SSIM: **0.9747152**	SSIM: **1**

5.3.3.7 splash1.bmp

Por favor revise *figura 62* o en su defecto el punto *5.1.2.7 splash1.bmp.*

5.3.3.7.1 Muestra de Resultados splash1.bmp

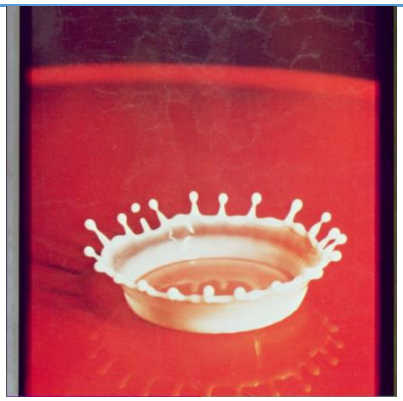


FIGURA 168 ESTEGOIMAGEN 61 SIM88.BMP (1024X1024)



Figura 169 Imagen Extraída 62 EIM88.bmp (512x256)

Estegoimagen: **SIM88.bmp**	Imagen extraída: **EIM88.bmp**
PSNR: **41.6227**	PSNR: **1**
SSIM: **0.9793622**	SSIM: **1**

5.4. Comparación con una Raspberry Pi 4

Este trabajo se hizo sobre una Raspberry Pi 2 pensado en mostrar que hardware con recursos limitados pueden satisfacer algunas funciones, sin embargo, hay preguntas que el trabajo por sí solo no responde, entonces utilizando una Raspberry Pi 4 (es decir más moderna y con cuatro veces más memoria RAM y un procesador a 1.5GHzy con un sistema operativo a 64 bits), se probarán los siguientes puntos:

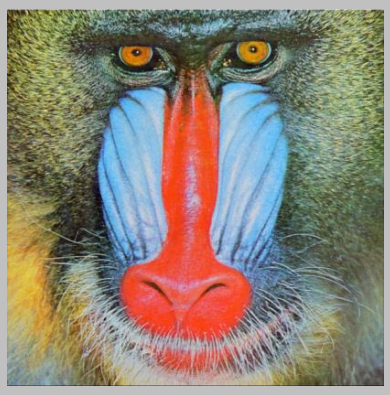
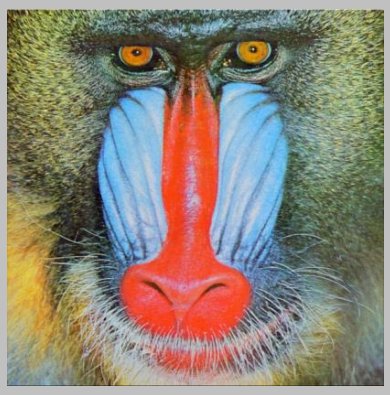
a) Comprobar si el programa funciona
b) Comparar resultados entre la estego-imagen obtenida en ambas Raspberry Pi
c) El tiempo que se necesita para ocultar una imagen en ambas Raspberry
d) El tiempo para recuperar un mensaje en ambas Raspberry.

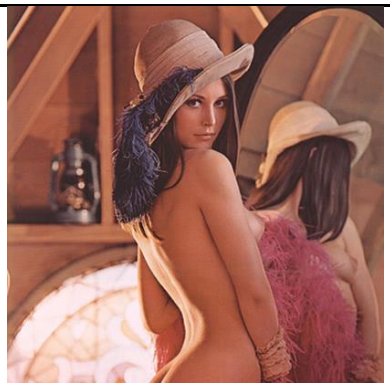
5.4.1. Comparación de Resultados

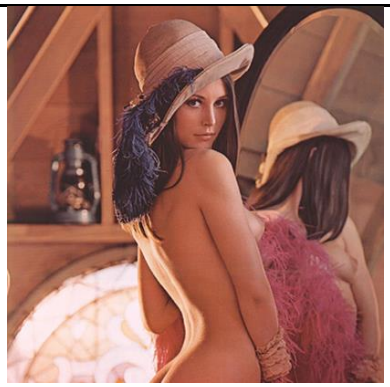
Para que el programa funcionara en la Raspberry Pi 4 lo único necesario fue copiar el programa y funciona de igual manera que en la Raspberry Pi 2, sin embargo, es necesario mencionar que la única forma que encontré para manipular el programa es a través de la terminal, esto es debido a que es el único compilador de Python 2 que tiene la Raspberry Pi 4 por defecto, actualmente el compilador con interfaz gráfica solo funciona para Python 3.

Para este apartado solo se compararon una imagen cuarto portadoras distintas con un mensaje RGB, uno a escala de grises y dos a blanco y negro uno de distinto tamaño.

La siguiente tabla muestra la comparación de estego-imágenes:

Raspberry Pi 2	Raspberry Pi 4
Cubierta: baboon1.bmp **Mensaje:** GS1.bmp (EdG) **Estego-imagen:** SIM36.bmp **Tiempo:** 1m 22.19s **PSNR:** 34.2349 **SSIM:** 0.9832782 	**Cubierta:** baboon1.bmp **Mensaje:** GS1.bmp (EdG) **Estego-imagen:** SIM0.bmp **Tiempo:** 12.21s **PSNR:** 34.2349 **SSIM:** 0.9832782 
Cubierta: Lena.bmp **Mensaje:** SLT.bmp (RGB) **Estego-imagen:** SIM49.bmp **Tiempo:** 1m 55.57s **PSNR:** 55.559 **SSIM:** 0.9977588	**Cubierta:** Lena.bmp **Mensaje:** SLT.bmp (RGB) **Estego-imagen:** SIM76.bmp **Tiempo:** 16.29s **PSNR:** 55.559 **SSIM:** 0.9977588





Cubierta: FO.bmp

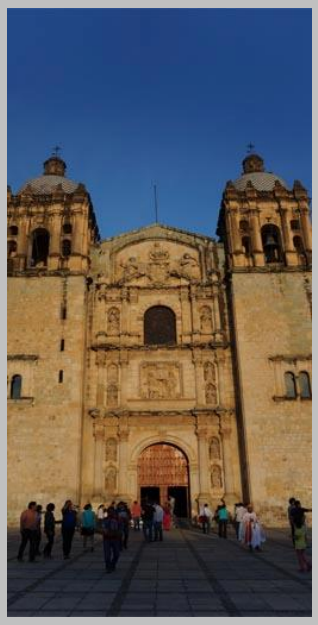
Mensaje: UACM.bmp (ByN)

Estego-imagen: SIM35.bmp

Tiempo: 1m 06.13s

PSNR: 51.7635

SSIM: 0.997951



Cubierta: FO.bmp

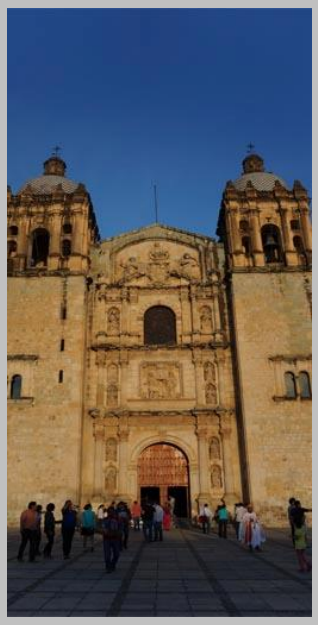
Mensaje: UACM.bmp (ByN)

Estego-imagen: SIM33.bmp

Tiempo: 9.77s

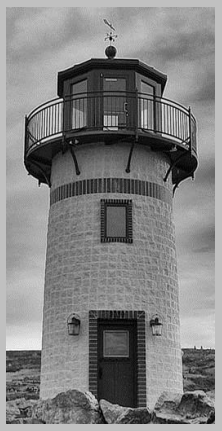
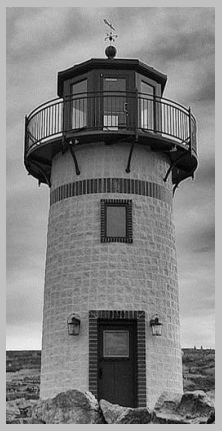
PSNR: 51.7635

SSIM: 0.997951



Cubierta: jellybeans.bmp	**Cubierta:** jellybeans.bmp
Mensaje: NIIL.bmp (ByN)	**Mensaje:** NIIL.bmp (ByN)
Estego-imagen: SIM25.bmp	**Estego-imagen:** SIM6.bmp
Tiempo: 5m 29.72s	**Tiempo:** 40.75s
PSNR: 45.0494	**PSNR:** 45.0494
SSIM: 0.9932919	**SSIM:** 0.9932919
	

En la siguiente tabla se compararan los mensajes extraídos:

Raspberry Pi 2	**Raspberry Pi 4**
Estego-imagen: SIM36.bmp	**Estego-imagen:** SIM0.bmp
Imagen extraída: EIM19.bmp	**Imagen extraída:** EIM11.bmp
Tiempo: 7m 47.74s	**Tiempo:** 1m 04.32s
PSNR: 1	**PSNR:** 1
SSIM: 1	**SSIM:** 1
	

Estego-imagen: SIM49.bmp

Imagen extraída: EIM40.bmp

Tiempo: 8m 50.43s

PSNR: 1

SSIM: 1



Estego-imagen: SIM76.bmp

Imagen extraída: EIM41.bmp

Tiempo: 1m 14.74s

PSNR: 1

SSIM: 1



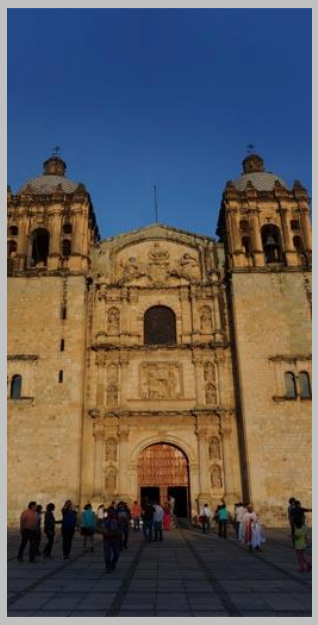
Estego-imagen: SIM35.bmp

Imagen extraída: EIM11.bmp

Tiempo: 19.98s

PSNR: 1

SSIM: 1



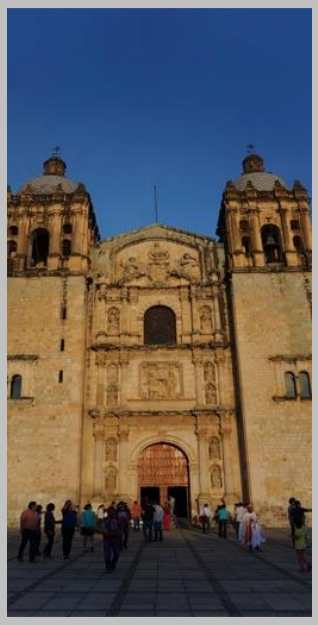
Estego-imagen: SIM33.bmp

Imagen extraída: EIM75.bmp

Tiempo: 2.89s

PSNR: 1

SSIM: 1



Estego-imagen: SIM25.bmp	**Estego-imagen:** SIM6.bmp
Imagen extraída: EIM62.bmp	**Imagen extraída:** EIM60.bmp
Tiempo: 5m 29.72s	**Tiempo:** 1m:43.36s
PSNR: 1	**PSNR:** 1
SSIM: 1	**SSIM:** 1
	

:

Capítulo 6 Conclusiones

Para poder mejor interpretación a los resultados, se resumirán en las siguientes tablas:

UACM		
Imagen	**PSNR**	**SSIM**
KC	51.6498	0.9979382
SLT	51.6812	0.998364
RGB3	51.506	0.9994518
msg1	51.6286	0.998687
GB	51.7943	0.9975763
FO	51.7635	0.997951
LC	51.6680	0.9978685

NIIL		
Imagen	**PSNR**	**SSIM**
Lena	54.8984	0.9975646
Airplane	40.4899	0.9820831
Babbon	34.2299	0.9832578
Jellybeans	44.9816	0.9931334
Peppers	39.9219	0.9769036
Sailboat	36.6749	0.9747266
Splash	41.6246	0.9793685

awa		
Imagen	**PSNR**	**SSIM**
Lena	53.4140	0.99712
Airplane	40.4152	0.9817841
Babbon	34.2175	0.9832166
Jellybeans	44.7773	0.9928271
Peppers	39.8658	0.9767524
Sailboat	36.6463	0.9745986
Splash	41.5411	0.9791749

Gs1		
Imagen	**PSNR**	**SSIM**
Lena	55.7942	0.9978189
Airplane	40.5214	0.9822458
Babbon	34.2349	0.9832782
Jellybeans	45.0708	0.9933607
Peppers	39.9480	0.9769805
Sailboat	36.6881	0.9747775
Splash	41.6653	0.9795032

Gs2		
Imagen	**PSNR**	**SSIM**
Lena	55.0639	0.9976084
Airplane	40.4967	0.9821056
Babbon	34.2347	0.9832706
Jellybeans	45.0000	0.993163
Peppers	39.9426	0.9769618
Sailboat	36.6773	0.9747253
Splash	41.6573	0.9794883

Gs3		
Imagen	**PSNR**	**SSIM**
Lena	55.6071	0.9977732
Airplane	40.5143	0.9822092
Babbon	34.2347	0.9832706
Jellybeans	45.0538	0.9933065
Peppers	39.9426	0.9769618
Sailboat	36.6855	0.9747624
Splash	41.6573	0.9794883

SLT		
Imagen	PSNR	SSIM
Lena	55.559	0.9977588
Airplane	40.5124	0.9822005
Babbon	34.234	0.9832686
Jellybeans	45.0494	0.9932919
Peppers	39.9421	0.9769537
Sailboat	36.6847	0.9747631
Splash	41.6562	0.9794844

KC		
Imagen	PSNR	SSIM
Lena	55.6052	0.9977698
Airplane	40.5149	0.9822199
Babbon	34.2350	0.9832678
Jellybeans	45.0534	0.9933022
Peppers	39.9420	0.9769575
Sailboat	36.6860	0.9747721
Splash	41.6579	0.9794797

RGB3		
Imagen	PSNR	SSIM
Lena	54.8055	0.9975276
Airplane	40.4879	0.9820434
Babbon	34.2300	0.9832541
Jellybeans	44.9750	0.9930992
Peppers	39.9209	0.9768864
Sailboat	36.6746	0.9747152
Splash	41.6227	0.9793622

Con ayuda de la tablas es posible notar que el algoritmo trabaja de excelente manera ya que los rangos de PSNR de las imágenes muestra seleccionadas varía entre los 34.2175dB correspondiente al mensaje *awa.jpg* y con la cubierta *baboon1.bmp* y el PSNR más alto es de 55.7942dB correspondiente al mensaje *GS1.jpg* dentro de la cubierta *Lena.bmp*.

De las tablas se encuentra en color rojo las imágenes con menor desempeño, es visible que *baboon1.bmp* es la cubierta cuyo PSNR es menor y por lo tanto con menor calidad al comparar la estegoimagen con la cubierta original, sin embargo, el valor mínimo de PSNR para considerar un trabajo de esteganografía de calidad es si este alcanza los 30dB y *baboon1.bmp* a pesar de tener menor rendimiento

en estas pruebas, nunca obtuvo un resultado menor a esa cifra, por lo que sí es un trabajo esteganográfico de calidad.

En caso contrario las imágenes en color verde indican la que mejor rendimiento obtuvieron, y es notorio que *Lena.bmp* tiene lo mayores PSNR en la pruebas referente a este trabajo y no solo eso, sus valores nunca estuvieron por debajo de los 50db, lo que significa que sus medidas son correspondientes a un trabajo de alta calidad, pues estar por encima de los 50dB está considerado como tal.

La otra medida de calidad de las estegoimágenes que se utilizó en este trabajo fue el Índice de Similaridad Estructural (SSIM) el cual a diferencia de los PSNR de las imágenes, no había mucha diferencia ya que los SSIM estaban entre el 97% y el 99%, es decir todas las pruebas realizadas tendrían un Índice de Similaridad Estructural muy alto, lo cual da más valor a los resultados obtenidos como al propio trabajo.

Por último, el punto más importante, es el mensaje, si bien para que una imagen pueda esconder otra es importante que pase desapercibida, eso es inútil si el mensaje (el cual debe de ser la información más importante) se altera, por lo que al momento de realizar un estegoanálisis y que los PSNR y SSIM obtenidas de los mensajes extraídos sean de 1 y 100% respectivamente demuestra que el mensaje está protegido de cierta manera.

Lo tiempos en los que la Raspberry 2 oculta una imagen varían entre un minuto y pocos segundos, hasta algunos segundos después de los cinco minutos y para la extracción varía entre los 20 segundos hasta más 12 minutos, estos valores

pueden cambiar por las características de las imágenes, más notoriamente por el tamaño.

Con los resultados obtenidos es posible validar a la Raspberry Pi como una tarjeta capaz de replicar algoritmos dedicados, con lo cual la valida como una herramienta muy conveniente debido a su precio y tamaño.

En comparación con un equipo más reciente y con mayor capacidad, se puede concluir los siguientes puntos:

a) El programa funciona en equipos mas modernos siempre y cuando tengan la posibilidad de ejecutar programas hechos en Python 2 y tengan las bibliotecas necesarias (en el caso de la Raspberry 4 el sistema Debian las tenía por defecto)
b) Los resultados numéricos de la estego-imagen no se ven afectados por el cambio de sistema por lo que no mejoraron los resultados de las estego-imágenes
c) Los resultados numéricos de la imagen extraída no se ven afectados por el cambio de sistema, por lo que la recuperación de imágenes no se vieron afectadas, ya que el mensaje en cualquiera de los dos sistemas se mantiene exactamente igual al original.
d) El tiempo de ejecución del algoritmo mejora mucho con el cambio de sistema ya que los tiempos de espera se reducen, sin embargo, hay que tener en cuenta que estos dependen en ambos casos del tamaño de las imágenes a manipular ya que el par de imágenes con menor tamaño (FO.bmp y UACM.bmp, ambas de 512x256 pixeles) fueron las que en

menos tiempo ocultaron la imagen con : 1m 06.13s para Raspberry Pi 2 y 9.77s para Raspberry Pi 4 y se pudo extraer la imagen en: 5m 19.98s en Raspberry Pi 2 y 2.89s en Raspberry Pi4. En el caso contrario el par de imágenes más grandes (jellybeans.bmp y NIIL.bmp ambas de 1024x1024 pixeles, es decir 8 veces más grandes que el par con menor tamaño) fueron la que en más tiempo ocultaron la imagen, necesitando: 5m 29.72s para Raspberry Pi 2 y 40.75s para Raspberry Pi 4 y se pudo extraer la imagen en 12m 18.73s en Raspberry Pi 2 y 1m 43.36s para Raspberry Pi 4.

A lo largo de esta tesis se han hablado de los siguientes puntos:

1. La utilización de un hardware económico (en recursos, espacio y costo) es suficiente para la implementación de un algoritmo esteganográfico.
2. Es posible hacerlo en un software de licencia libre como lo es Python.
3. Las imágenes digitales pueden ser descompuestas en planos de bits y cada uno es importante, sin embargo, si se cambian los menos significativos, existe un cambio, pero es muy pequeño para ser percibido para el ojo humano.
4. La importancia de las medidas de calidad para comprobar lo robusto que puede o no ser un algoritmo.
5. La relación de dimensiones de las cubiertas respecto a los mensajes.
6. Comparación de su funcionamiento en dos Raspberry PI de distintas capacidades.

6.1. Trabajo a futuro

- Probar algoritmos más complejos en una Raspberry Pi.
- Mejorar la robustez del algoritmo.
- Implementación para hardware limitado al igual que Raspberry Pi (como puede ser un Smartphone de gama media).
- Trabajar en algoritmos basados en la identificación de estegoimágenes y si es posible al estegoanálisis.

Referencias

Bibliográficas

Areitio, J. (2008). *Seguridad de la información. Redes, informática y sistemas de información*. Madrid: Praninfo.

Ortega Jesús, López Miguel Ángel y García Eugenio C. (2006). *Introducción a la criptografía historia y actualidad*. Cuenca : Ediciones de la Universidad de Castilla - La Mancha

Ashok Ambardar. (2003). *Procesamiento de Señales analógicas y digitales.* ISBN 970686038X: Thomson Learning.

Castleman, K. R. (2002). *Digital image processing*. Pearson Education.

Dixon, A. N. (1994). *Copyright Protection for the Information Superhighway*. Maine: European Intellectual Property Review.

Stefan Katzenbeisser, Fabien A. P. Petitcolas (1999). *Information hiding techniques for steganography and digital watermarking*.

Cox J. Ingemar, Miller Matthew L. (2008). *Digital Watermarking & Steganography.*

Miano, John. (1999). *Compressed Image File Formats.*

A. Horé, D. Ziou. (2010). *"Image Quality Metrics: PSNR vs. SSIM".* 20th *International Conference on Pattern Recognition*, Istanbul, 2010, pp. 2366-2369, doi: 10.1109/ICPR.2010.579

Mesográficas

Herodoto (2009). *Los nueve libros de la historia*. Biblioteca Virtual Universal. Obtenido de https://www.biblioteca.org.ar/libros/157772.pdf

Instituto Federal de Telecomunicaciones *En México hay 74.3 millones de usuarios de internet y 18.3 millones de hogares con conexión a este servicio: ENDUTIH 2018.* Obtenido de http://www.ift.org.mx/comunicacion-y-medios/comunicados-ift/es/en-mexico-hay-743-millones-de-usuarios-de-internet-y-183-millones-de-hogares-con-conexion-este.

Raspberry Pi. *Raspberry Pi 2 Modelo B* obtenido de https://www.raspberrypi.org/products/raspberry-pi-2-model-b/