PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

FACULTAD DE INGENIERIA

ESCUELA DE SISTEMAS

TEMA:

CREACIÓN DE UNA HERRAMIENTA PARA OCULTAMIENTO DE INFORMACIÓN A TRAVÉS DE IMÁGENES

AUTOR:

ADRIÁN ESTEBAN OLIVA PAZ

DIRECTOR:

JORGE ALFREDO CALDERÓN SERRANO

QUITO, junio 2020

“Hay un mundo oculto detrás de lo que está a la vista de todos.

Para todos nosotros.”

Dan Brown

Agradecimiento

Doy gracias a Dios por permitirme culminar mi tesis en este tiempo de pandemia. A mis padres por haberme dado la oportunidad de seguir con mis estudios, apoyarme cada día para poder llegar a este momento y brindarme su apoyo y cariño durante toda mi vida. A mi hermana por aguantar mis desvelos y algunos gritos de felicidad cuando terminaba algún proyecto imposible. A toda mi familia por estar al pendiente de mi durante mis años en la universidad. En especial a mi abuelita Piedad.

A mis compañeros por aguantarme e iluminarme cuando ya no podía. A mis profesores por trasmitirme sus enseñanzas. A toda la familia del LTIC, gracias a ellos pase buenos e inolvidables momentos, en especial las risas que nunca faltaron.

Un agradecimiento especial a Christian Sánchez, Kevin Jarrín, Luis Vacacela, Sebastián Villacís, Francisco Mejía, Carolina Calvopiña, Leonardo Chamorro por ser un gran apoyo durante la carrera.

Finalmente agradezco a Sofy por siempre apoyarme en las buenas y en las malas, su amor incondicional me permitió seguir adelante y me ayudo a crecer como persona, sin duda alguna estar a tu lado me abrió un mundo de posibilidades infinitas y las palabras no son lo suficiente para expresar lo muy agradecido que estoy de tenerte a mi lado.

Contenido

1. Marco teórico 8
1.1 Esteganografía 8
1.1.1 Estegoanálisis 9
1.1.2 Criptología y Esteganografía 9
1.2 Planteamiento del Problema 10
1.3 Esteganografía en la historia 11
1.3.1 Esteganografía en la antigua Grecia 12
1.3.2 Esteganografía en el Renacimiento 13
1.3.3 Esteganografía en la Segunda Guerra Mundial 14
1.4 Esteganografía en imágenes digitales 16
2. Técnicas Estenográficos en Imágenes Digitales. 19
2.1 Técnica de Sustitución Bit Menos Significativo 19
2.1.1 Aplicación del LSB en la imagen 20
2.1.2 Seleccionando el pixel para su sustitución 21
2.2 Técnica Basadas en Paleta de Colores. 22
2.2.1 Ocultamiento de Información con la Paleta de Colores. 23
2.3 Técnica Basadas en Coeficientes Cuánticos. 23
2.3.1 Algoritmo de Compresión JPEG. 23
2.3.2 Ocultamiento de Información en el Algoritmo de JPEG. 25
3. Herramientas Estenográficas en Imágenes Digitales. 27
3.1 Herramienta Esteganográfica S-Tools. 27
3.2 Herramienta Esteganográfica Digital Invisible Ink Toolkit 29
3.3 Herramienta Esteganográfica F5 31
4. Desarrollo de la Herramienta Esteganográfica. 34
4.1 Metodología de desarrollo. 34

4.2 Análisis. 34
4.2.1 Casos de Uso a Detalle 35
4.3 Diseño 36
4.3.1 Diagrama de Flujo 37
4.3.2 Diseño de la Interfaz Gráfica 38
4.4 Programación. 39
4.4.1 Pre-Programación 39
4.4.2 Procesos no funcionales 39
4.4.3 Algoritmo Ocultar 40
4.4.4 Algoritmo Recuperar 42
4.5 Pruebas 45
4.5.1 Etapa 1 45
4.5.2 Etapa 2 50
4.5.3 Etapa 3 51
4.5.4 Comparación de resultados 54
5. Conclusiones y Recomendaciones 55
5.1 Conclusiones 55
5.2 Recomendaciones 56
Bibliografía 58
Anexo 60
Manual de usuario 60
Ocultar información 60
Recuperar información 65

Tabla de ilustraciones

Figura 1 Estegomedios comunes en la esteganografía. (Elgabar & Alamin , 2013) 8

Figura 2 Problema del prisionero de Gustav J. Simmons. (Oliva, 2020)..... 11

Figura 3 Mensajero griego entregando el mensaje oculto después de afeitarse el cuero cabelludo. (Zegarra, 2017)..... 12

Figura 4 Figura 1 4 Vexierbild de Erhard Schön. Los rostros pertenecen a Carlos V, Fernando I, Clemente VII y Francisco I. (Muñoz Muñoz, Crypt4you, 2014)..... 13

Figura 5 Carta enviada durante la Segunda Guerra Mundial. (Cedillo, 2019) 15

Figura 6 Composición de una imagen como una matriz bidimensional de pixeles. (Lyra, 2011) 17

Figura 7 Imagen en RGB representando el color del pixel como representación decimal y binaria. (DataGenetics, 2012)..... 20

Figura 8 Algoritmo de Compresión JPEG paso por paso. (Reyes, 2016) 24

Figura 9 Bloque de 8x8 pixeles aplicando DCT. (Weiss, 2016)..... 24

Figura 10 Herramienta S-Tools. (Oliva, 2020)..... 28

Figura 11 Imagen original a la izquierda y estegoimagen a la derecha ordenada por su luminancia. (Johnson & Jajodia, Steganalysis of Images Created Using Current Steganography Software, 1998)..... 28

Figura 12 Página principal de la herramienta Digital Invisible Ink Toolkit. (Oliva, 2020) 29

Figura 13 Ocultación de información con Digital Invisible Ink Toolkit. (Oliva, 2020) 30

Figura 14 Recuperación de información con Digital Invisible Ink Toolkit. (Oliva, 2020) 31

Figura 15 Ocultación de información con la herramienta F5. (Oliva, 2020) 32

Figura 16 Recuperación de información con la herramienta F5. (Oliva, 2020) 33

Figura 17 Diagrama General – Caso de Uso (Oliva, 2020) 35

Figura 18 Diagrama de Flujo (Oliva, 2020) 37

Figura 19 Interfaz de Ocultamiento (Oliva, 2020) 38

Figura 20 Interfaz de Recuperación (Oliva, 2020) 38

Figura 21 Prueba de ocultamiento de información (Oliva, 2020) 46

Figura 22 Imagen derecha usada para ocultar información, imagen izquierda con la información oculta. (Oliva, 2020) 47

Figura 23 Prueba de recuperación de información sin contraseña (Oliva, 2020) 48

Figura 24 Prueba de recuperación de información con contraseña no valida (Oliva, 2020) 49

Figura 25 Prueba de recuperación de información con contraseña correcta (Oliva, 2020) 50

Figura 26 Prueba de ocultamiento de información que supere la capacidad máxima de la imagen (Oliva, 2020) 51

Figura 27 Prueba de ocultamiento de información de un texto e imagen grande 52

Figura 28 Prueba de recuperación de información de texto e imagen grande 53

1. Marco teórico

1.1 Esteganografía

La ciencia de la esteganografía proviene de las palabras griegas stenagos: oculto y graphein: escribir, definiéndola así como la escritura oculta. El principal objetivo es ocultar la información en un contenedor u objeto para pasar desapercibido aun estando a la vista de todos.

El medio o contenedor en el cual la información se oculta se denomina estegomedio; usualmente la información se oculta en un objeto por lo tanto si se oculta la información por medio de imágenes el estegomedio se lo denominaría como estegoimagen, de esta manera se podría denominar varios estegomedios como estegovideo, estegoaudio, entre otras.

> La Esteganografía estudia todas las posibles técnicas utilizadas para insertar información sensible dentro de otro fichero, denominado 'fichero contenedor' (que podría ser un gráfico, un documento o un programa ejecutable), para tratar de conseguir que pueda pasar inadvertida a terceros, y solo pueda ser recuperada por parte de un usuario legítimo empleando para ello un determinado algoritmo de extracción de la información. (Gómez Vieites, 2014, p. 53)

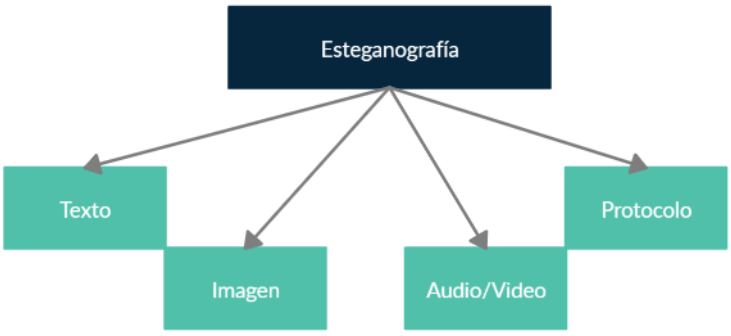

Figura 1 Estegomedios comunes en la esteganografía. (Elgabar & Alamin , 2013)

1.1.1 Estegoanálisis

El estudio de la esteganografía se divide en dos ramas:

- la esteganografía, encargada de estudiar los algoritmos, métodos, y herramientas para ocultar información en algún medio y,
- el estegoanálisis, encargado de estudia la seguridad de los algoritmos, métodos, y herramientas pasando por diferentes pruebas para detectar si el medio presenta información oculta.

Aunque parece que el estegoanálisis es la contraparte de la esteganografía, ambas son necesarias para desarrollar de algoritmos, métodos, y herramientas robustas.

Sin importar que tipo de técnica usemos para ocultar información, el estegoanálisis por medio de patrones, técnicas estadísticas e incluso de inteligencia artificial detecta anomalías en el estegomedio utilizado y esto es lo que permite conocer que tan seguro es una técnica de otra.

1.1.2 Criptología y Esteganografía

No podemos confundir la esteganografía con la criptología. Ambas son técnicas para ocultar información a través de un mensaje, no obstante la criptología envía la información de manera codifica de tal manera que sea incomprensible para una tercera persona y aunque lo sepa no pueda conocer lo codificado; la esteganografía es todo lo contrario, le importa que la información este expuesta para todos pero solo algunos saben lo que oculta

1.2 Planteamiento del Problema

En 1983 Gustav J. Simmons planteo el problema del prisionero, describiendo un escenario donde dos personajes quieren comunicarse de manera encubierta en donde el canal en el que se lo trasmite es un canal que puede leer el mensaje e incluso ser manipulado por un tercer personaje.

Las soluciones ante este problema son de dos tipos: el uso de canales subliminales, es decir son canales encubiertos dentro del mismo canal inseguro para trasmitir mensajes, o el uso de la esteganografía.

Muñoz (2017) interpreta el problema del prisionero de Simmons de la siguiente manera:

> La descripción de este problema parte de la necesidad de comunicación entre dos entidades A y B, se supone por simplificación que se habla de personas, que son arrestadas y confinadas en celdas separadas. El objetivo que se plantea consiste en desarrollar un plan de fuga, intercambiando información a través de su guardián, dado que se les impide una comunicación directa. Si el guardián es medianamente competente no permitirá una comunicación cifrada, y si tiene la más mínima sospecha interrumpirá las comunicaciones. De este modo, A y B deben comunicarse de manera invisible usando algún tipo de esteganografía. Una forma de hacer esto es ocultar información en un mensaje de apariencia inocua, por ejemplo, una imagen. B podría dibujar un dibujo de una vaca azul en un pasto verde, y pedir al guardián que se lo pase a A. Lógicamente el guardián observará el dibujo y al no alertar nada raro en él, si acaso un dibujo de arte abstracto, lo transmitirá sin saber que los colores del dibujo ocultan un mensaje. (p.49)

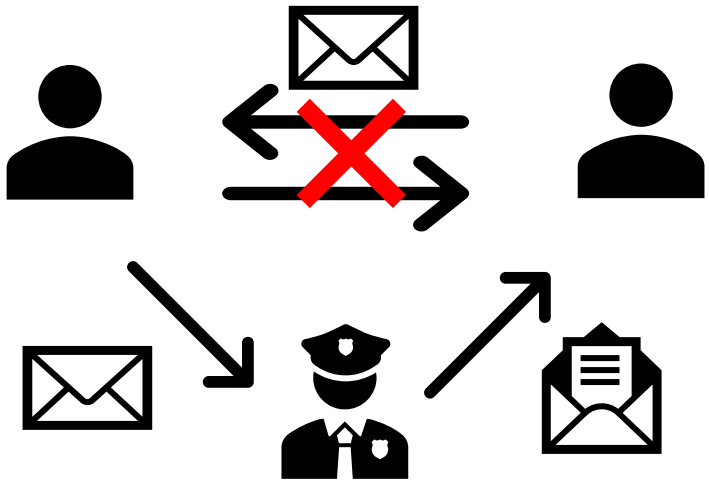

Figura 2 Problema del prisionero de Gustav J. Simmons. (Oliva, 2020)

Este planteamiento del problema es altamente aplicado a las comunicaciones actuales ya que, el internet se ha vuelto un canal común al igual que un canal inseguro y cualquiera puede acceder a dicha información. Es así como la esteganografía podría ser una posible solución para evitar que cualquiera pueda modificar la información y mantenga la privacidad y anonimato de quien lo envió.

1.3 Esteganografía en la historia

Durante años la esteganografía ha ido adaptándose a nuevos medios, desde usar esclavos o animales para tatuar la información y enviarlos cuando su cuero cabelludo o pelaje creciera hasta usar medios digitales como imágenes, audios y videos para ocultar información.

Los usos de la esteganografía en la historia han permitido desarrollar técnicas que solamente son limitadas a la imaginación de sus creadores. Parece un tema innovador colocar información en un medio para pasar desapercibido en el canal y llegar a su destinatario, pero en la historia la misma idea se ha utilizado en repetidas veces para ocultar información.

1.3.1 Esteganografía en la antigua Grecia

Los testimonios más antiguos sobre el uso de esteganografía son relatados en Las Historias de Heródoto de Halicarnaso entre 484 a.C y 425 a.C. En los conflictos de Grecia y Persia, Heródoto relata como la escritura oculta salvo a Grecia de ser invadida; cuenta entonces que Histeo quería alentar a su yerno Aristágora de Mileto rebelarse contra el rey persa, para enviar su mensaje afeito la cabeza de su mensajero y tatuó el mensaje en él, espero a que creciera su cabello y lo envió. El mensaje fue entregado a su destinatario sin levantar sospechas entre enemigos.

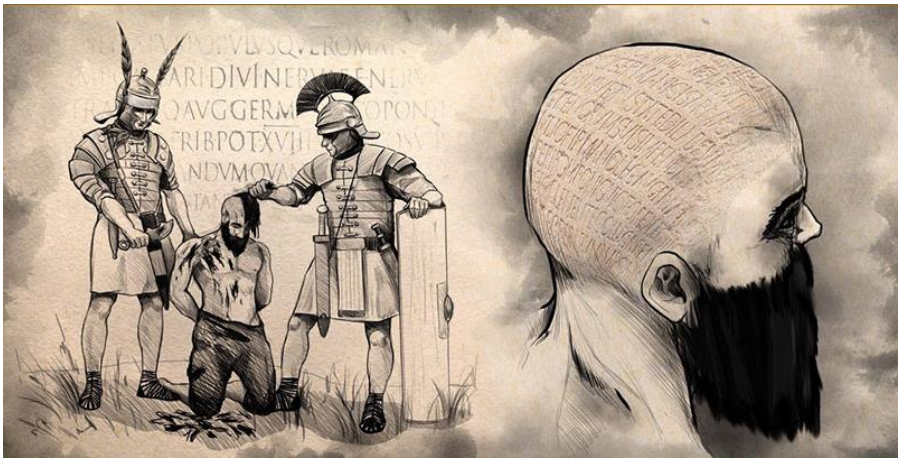

Figura 3 Mensajero griego entregando el mensaje oculto después de afeitarse el cuero cabelludo.

(Zegarra, 2017)

Otra historia relatada por Heródoto cuenta sobre Demerato en su ingenio de enviar un mensaje a Esparta sobre la invasión a Grecia por rey persa Jerjes. Demerato escribió su mensaje en tablillas de madera y luego las cubrió con cera. Estas tablas en blanco pasaron desapercibidas hasta caer en manos de los Espartanos.

1.3.2 Esteganografía en el Renacimiento

En el periodo renacentista aparecieron estudios más relevantes sobre la esteganografía. En 1550, el físico y matemático Girolamo Cardano reinvento un procedimiento chino para ocultar información, denominado después como la reja de Cardano. Consiste en un pedazo de cartón con agujeros que al sobreponer sobre el texto revelaba la información y solamente quien tenía la plantilla podía obtener la información ya que esta actuaba como un filtro en el texto; sin embargo no se podía utilizar para transmitir grandes cantidades de información.

Aprovechando el arte en el periodo renacentista se comenzó a utilizar el arte anamórfico como un medio ideal para ocultar información. La idea de este tipo de arte es deformar la imagen de manera que solo puedas apreciar su verdadera intención desde un punto de vista en particular. Erhard Schön en su obra Vexierbild se puede apreciar de frente un intento de paisaje pero desde visto desde un ángulo distinto podemos observar que se oculta los rostros de cuatro personajes reconocidos en la época.

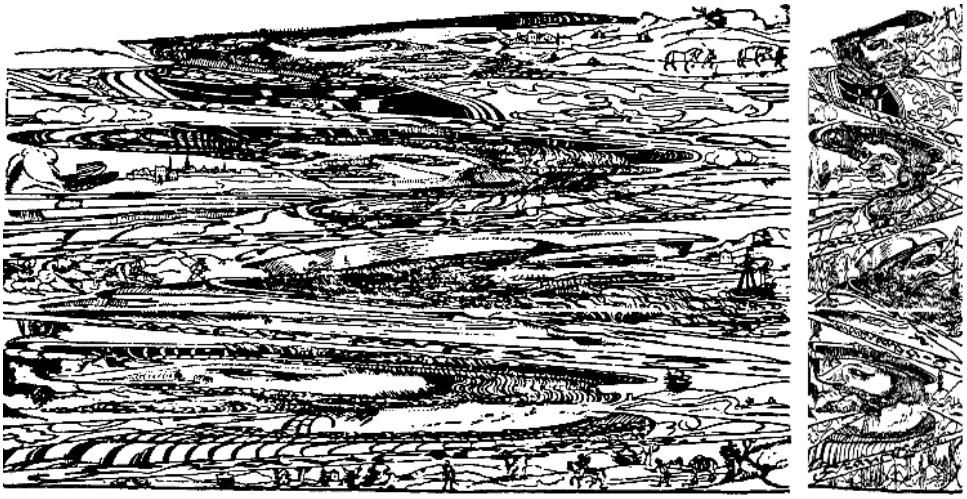

Figura 4 Figura 1 4 Vexierbild de Erhard Schön. Los rostros pertenecen a Carlos V, Fernando I, Clemente VII y Francisco I. (Muñoz Muñoz, Crypt4you, 2014)

La tinta invisible apareció también en este periodo; el científico italiano Giovanni Battista della Porta escondió un mensaje dentro de un huevo cocido mediante una tinta invisible producida al mezclar alumbre y vinagre y escribirlo en la cáscara; la tinta atraviesa la cascará y el mensaje se encontrará en la clara del huevo. Esta tinta invisible dio la apertura a que más adelante se utilice procesos químicos más sofisticados para su creación utilizando compuestos químicos inorgánicos y orgánicos.

1.3.3 Esteganografía en la Segunda Guerra Mundial

Durante la Segunda Guerra Mundial el uso de la esteganografía era más frecuente. El uso de palomas mensajeras, ocultar mensajes en periódicos o medios de comunicación por radio era común en esta época. El código en jerga fue utilizado por japoneses y alemanes para ocultar en palabras cotidianas, frases y jergas mensajes sobre ataques e información delicada. Durante el ataque a Pearl Harbor, los japoneses usaron la frase Tora! Tora! Tora! Para referirse al éxito que tuvieron en el ataque.

La técnica de micropunto fue introducida por espías alemanes donde se reducía un mensaje de un tamaño aproximado a un milímetro y se los colocaba en los puntos de las letras i y j o de cualquier signo ortográfico como puntos, tildes, diéresis y comas de tarjetas postales; el grupo de inteligencia norteamericana para combatir contra el micropunto cambiaban los sellos postales por el mismo sello con el mismo valor o sobrescribían la carta.

Figura 5 Carta enviada durante la Segunda Guerra Mundial. (Cedillo, 2019)

Finalmente en la era digital, el uso de las técnicas esteganográficas es muy variada y ampliamente aplicables gracias al uso de los lenguajes de programación, la estadística y matemática, así como el uso de internet para él envió de los estegomedios. Algunos de estos ejemplos son mencionados por Muñoz (2017):

> La irrupción de las telecomunicaciones y la informática ha decantado los procedimientos estenográficos modernos hacia canales y formatos digitales. Así, en los últimos años se han publicado propuestas de ocultación de información utilizando imágenes, audio y vídeo digitales, tecnologías web como cabeceras http o cookies , utilización de la redundancia de las instrucciones máquina en ficheros ejecutables, lenguajes de marcado web como HTML-XML (ocultación basada en caracteres invisibles, modificación de los caracteres de las etiquetas alternando mayúsculas y minúsculas al ser estas insensitive y ocultación basada en el orden de los atributos de una etiqueta), utilización esteganografía de diferentes protocolos de comunicación (SOAP, HTTP, TCP, UDP, IPv4, IPv6, DHCP, ICMP, IPSEC, IGMP, FTP, DNS, 802.2, 802.3, redes inalámbricas, “accesorios” de mails

> , etc.) para establecer canales encubiertos (network steganography) para saltarse protecciones corporativas como, por ejemplo, cortafuegos (típicamente utilizando campos reservados, campos redundantes o el reordenamiento de paquetes), ocultación de información en sistemas de información y soportes de almacenamiento, malware oculto en hardware y puertas traseras en microchips, etc. (p.71)

En el 2001, se asoció a la esteganografía como parte del ataque terrorista al World Trade Center, por lo que cualquier contendió sobre esteganografía fue censurado y prohibido durante un largo periodo de tiempo. No obstante en el mismo año Niels Provos y Peter Honeyman de la universidad de Michigan publicaron sobre un estudio sobre contenido esteganográficos en el internet, donde pasaron por distintas técnicas estegoanalíticas para buscar en 2 millones de imágenes descargadas de internet el uso de esteganografía; como conclusión llegaron a que en ninguna de los 2 millones de imágenes encontraron indicios de esteganografía.

Actualmente usar contenidos multimedia como imágenes, videos y audio son medios comunes para el uso de la estenografía debido que este contenido lo encontramos diariamente circulando por internet por medio de las redes sociales, streamings y otros contenidos.

1.4 Esteganografía en imágenes digitales

En la presente tesis se usarán las imágenes como principal estegomedio para ocultar la información, por lo que es necesario tener cierto conocimiento de cómo se compone una imagen digital y lo que se puede lograr a través de ellas.

Una imagen digital no es nada más que una representación en un plano bidimensional por medio de bits que a su vez son representados por ceros y unos. Al transfórmalos de manera digital, esta se compone de cientos de elementos con un valor y localización en particular, ha estos elementos se los conoce como pixeles. (González & Woods, 2008)

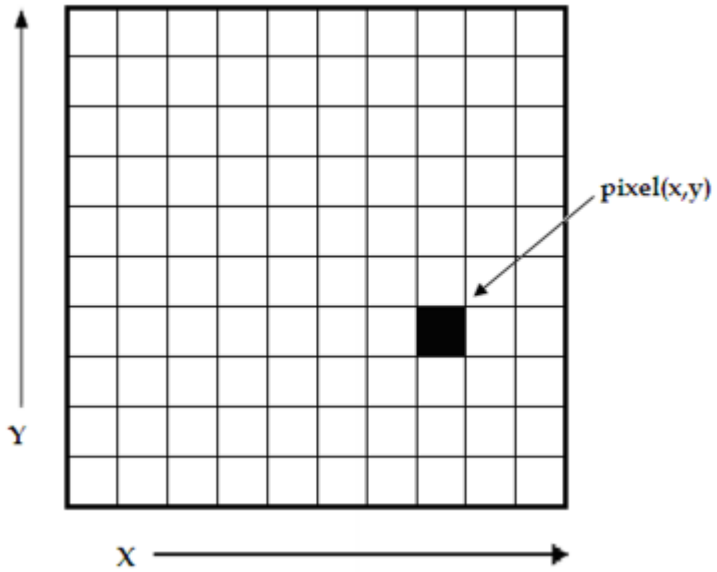

Figura 6 Composición de una imagen como una matriz bidimensional de pixeles. *(*Lyra*, 2011)*

En imágenes digitales es común escuchar el termino procesamiento de imágenes; el concepto hace referencia a almacenar imágenes analógicas de manera digital en computadoras para luego ser manipuladas por diferentes técnicas y procedimientos.

El concepto de procesamiento de imágenes digitales se remonta en el año de 1920, en donde su primera aplicación fue enviar una imagen a través de un cable submarino entre Londres y Nueva York. La idea era codificar la imagen por medio de señales eléctricas para que se reconstruya al otro lado del Atlántico, sin embargo muchos de los problemas que se dieron en ese entonces fue la calidad de la imagen, por lo que se descartó la idea de este procedimiento años más tarde. Aun así no podemos considera este ejemplo

como procesamiento de imágenes ya que según la definición no involucra el uso de computadores.

Con el surgimiento de los primeros computadores se potencio el concepto procesamiento de imágenes debido al gran interés en la exploración del espacio exterior y los avances en la medicina en los años de 1960; fue entonces donde se empezaron a probar diferentes técnicas para su manipulación ya que estas imágenes eran enviadas a través de sondas espaciales.

A través de los múltiples intentos, técnicas y procedimientos para llegar a tener lo que hoy conocemos como imágenes digitales, los expertos vieron un potencial para el uso de la esteganografía en este medio.

En la última década se han procesado múltiples algoritmos y procesos estenográficos en imágenes, algunos de estos son cálculos y fórmulas matemáticas como la trasformada de Fourier, transformada discreta del coseno, entre otras, para insertar información de manera óptima y casi indetectable al usar procesamiento de imágenes.

2. Técnicas Estenográficos en Imágenes Digitales.

Existe una diversa cantidad técnicas para ocultar información en imágenes digitales, al igual que varios ejemplos en donde la esteganografía en imágenes digitales puede ser aplicada. Podemos encontrar cualquier tipo de investigación, programas de código abierto y contenido sobre esteganografía aplicados a imágenes digitales actualmente; aunque no sea utilizada ampliamente para esconder información digital como lo hace la criptografía, su potencial solamente estaría limitado la imaginación humana.

2.1 Técnica de Sustitución Bit Menos Significativo

La técnica del bit menos significativos o LSB (por sus siglas en inglés, Low Significant Bit) es la técnica más utilizada en imágenes debido a su simplicidad para colocar información dentro de la imagen.

Las imágenes se representan en pixeles y un pixel se representa por una cadena de 8 bits. El pixel puede tener diversos niveles de rojo, verde y azul (RGB por sus siglas en inglés Red, Green, Blue) que están representados mediante 0 y 1, ya que el máximo número que se puede representar un pixel de color es de 255 entonces se puede representar un pixel en RGB de la siguiente forma: 80, 20, 55 y su representación binaria será 01010000, 00010100, 00110111 respectivamente.

Cuando se utiliza esta técnica lo ideal es cambiar el bit menos significativo es decir, el último bit a la derecha debido que al cambiar el último bit su valor en decimal no afectara mucho al resultado.

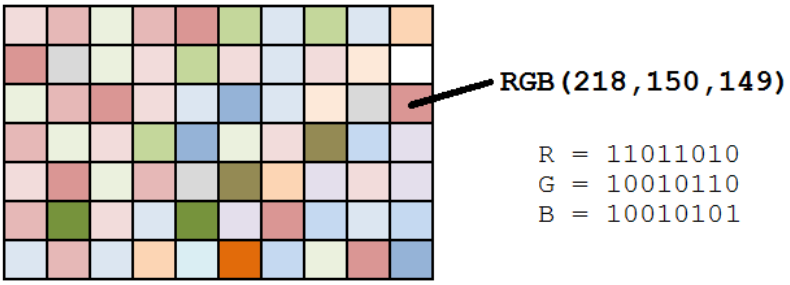

Figura 7 Imagen en RGB representando el color del pixel como representación decimal y binaria.

(DataGenetics, 2012)

Para ejemplificarlo consideremos lo siguiente: se tiene un nivel de rojo de 255 su representación binaria es 11111111, si se cambia el último valor por un 0 el resultado será 11111110 y la representación en decimal es 254 lo cual no es una gran impacto cuando se representa en color.

Aunque la técnica de LBS es cuestionada debido a que se puede detectar anomalías fácilmente con técnicas estegoanalíticas su uso es el más común en herramientas comerciales y de distribución libre. Su popularidad se debe a dos razones:

1. Permite ocultar grandes cantidades de información, casi 1 bit de información por cada pixel de la imagen.
2. Se puede programar e implementar de manera más sencilla esta técnica debido a que muchas herramientas esteganográficas están disponible con esta técnica y permite su estudio por ser de libre acceso.

2.1.1 Aplicación del LSB en la imagen

Para ocultar información en la imagen se debe elegir primero que alfabeto o código de caracteres se utiliza, para el ejemplo siguiente se usará el código ASCII y la letra M. Primero se debe traducir la letra M con su representación en ASCII dando como resultado

el número 77, al transformarlo a binario se tendrá 1001101, por lo tanto se ocultará el número binario en la imagen.

Se debe obtener los pixeles de la imagen y representarlo en RGB para luego convertirlo en binario. En el siguiente ejemplo se tomará los siguientes pixeles ya trasformados.

	Rojo	Verde	Azul
pixel 0	00110110	11110010	01010101
pixel 1	01110110	11000000	00000000
pixel 2	11110001	01100010	00100000

Para insertar la letra M en la imagen los pixeles cambiarán de la siguiente manera y se lo representará con rojo para diferenciar el cambio.

	Rojo	Verde	Azul
pixel 0	00110111	11110010	01010100
pixel 1	01110111	11000001	00000000
pixel 2	11110001	01100010	00100000

2.1.2 Seleccionando el pixel para su sustitución

La técnica LBS depende principalmente de la sustitución de los bits menos significativos, por lo tanto la elección de los pixeles a utilizar dependerá del criterio propio al emplear esta técnica.

Se puede seleccionar los bits a utilizar de forma secuencial donde se sustituirá de manera secuencial los bits a ser codificados, ya sea que se parta de una posición establecida o a partir de un cálculo externo.

También se puede seleccionarlos utilizando un número pseudoaleatoria el cual, se generará a partir de una semilla y así elegir que pixeles de la imagen ocultará la información.

Otra manera de seleccionarlos es basándonos en una función de selección definiendo varios criterios, un ejemplo es dividir la imagen en bloques y a cada bloque utilizar la sustitución por LBS.

2.2 Técnica Basadas en Paleta de Colores.

No todas las imágenes tienen la misma forma de almacenamiento, tradicionalmente se representan las imágenes en un archivo en donde su tamaño final no es excesivamente grande facilitando él envió de la imagen por internet.

Una de las soluciones para este problema fue crear imágenes con un número limitado de colores, de tal manera que cada pixel en lugar de almacenar el color apunte a una tabla de colores, esto hace que el tamaño con el que la codificación sea menor ya de tamaño mucho más pequeña.

Las imágenes de formato GIF (por sus siglas en inglés, Graphics Interchange Format) es un buen ejemplo que utiliza la paleta de colores permitiendo representar hasta 256 colores diferentes. Cada valor de los pixeles en RGB no es almacenado de forma directa en el archivo de la imagen sino se almacena índices que indican el color en la paleta de color permitiendo que cada pixel se puede representar con 8 bits o 2^8 colores posibles en la paleta. (Muñoz Muñoz, Privacidad y ocultación de información digital Esteganografía, 2017)

2.2.1 Ocultamiento de Información con la Paleta de Colores.

Se puede usar una imagen de formato GIF para ocultar información con la técnica de LSB, sin embargo se debe tener cuidado debido a que al hacer un cambio en el bit menos significativo, este pueda presentar un color distinto en la paleta de colores.

Una manera de solucionar este problema es ordenar la paleta de colores ya que estos estarán ordenados del color más usado al menos usado. Una vez ordenado se puede minimizar la diferencia de colores insertando colores que sean parecidos al de alado como si se tratara de ruido en la imagen. (Muñoz Muñoz, Privacidad y ocultación de información digital Esteganografía, 2017)

2.3 Técnica Basadas en Coeficientes Cuánticos.

En la evolución del procesamiento de imágenes digitales ha dado la apertura para que la esteganografía aproveche de algoritmos y trasformaciones para ocultar información, algunos de ellos como el algoritmos para compresión del tamaño de la imagen o algoritmos que permiten usar filtros en la imagen son usualmente usados.

El mejor ejemplo para el uso de algoritmos son imágenes con extensión JPEG (por sus siglas en inglés, Joint Photographic Experts Group) en donde se aprovecha la compresión de la imagen para ocultar la información utilizando los coeficientes cuánticos que se pueden obtener al aplicar la transformada discreta del coseno en la imagen.

2.3.1 Algoritmo de Compresión JPEG.

Para entender de mejor manera cómo funciona la esteganografía en las imágenes JPEG, se debe conocer cómo trabaja su algoritmo de compresión:

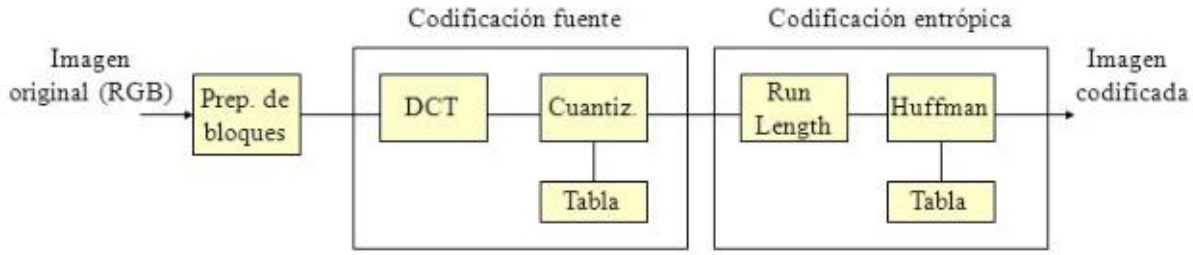

Figura 8 Algoritmo de Compresión JPEG paso por paso. (Reyes, 2016)

1. Lo primero que se hace es una conversión de la imagen digital del formato RGB a un formato YCbCr donde, Y es la iluminación de la imagen y los coeficientes que definen los colores Cb y Cr. Luego de realizar esta conversión dividir en bloques de 8x8 pixeles.
2. A cada bloque que se dividió se aplica la Trasformada Discreta del Coseno (DCT por sus siglas en inglés Discrete Cosine Transform). Se trasforma los valores de los 64 pixeles que comprende el bloque en frecuencias donde el valor superior izquierdo será la frecuencia más baja dando como resultado como la Figura 2-3

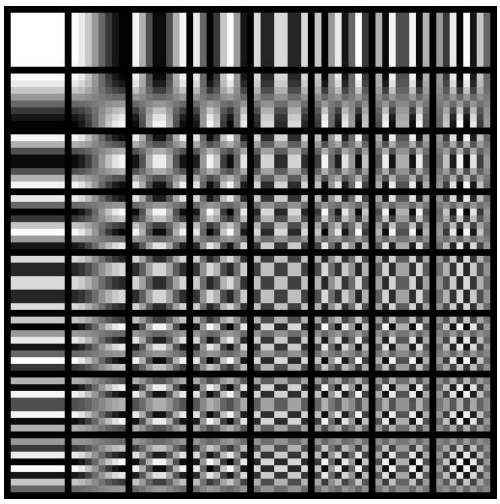

Figura 9 Bloque de 8x8 pixeles aplicando DCT. (Weiss, 2016)

3. Se realiza la cuantización en donde los coeficientes de los bloques se dividirán con una tabla de valores fijos y se redondean resultando en la eliminación todas las frecuencias altas; esta tabla de valores fijos se guarda en la cabecera del archivo de la imagen; entre más alto es el valor de esta tabla más detalles se eliminan. El resultado de esta operación dejara una matriz con las frecuencias bajas significante y con muchos ceros.
4. Luego se debe ordenar cada bloque de 8x8 tratando de tener la mayor cantidad de ceros juntos para una mejor compresión; para esto se utiliza el ordenamiento por zigzag iniciando desde el primer valor superior izquierdo.
5. Finalmente se comprime la imagen mediante algún algoritmo que permita comprimir los coeficientes con altas frecuencias o con bajas frecuencia, luego se utilizar la codificación de Huffman para comprimirlo en un árbol y guardarlo finalmente en la cabecera.

2.3.2 Ocultamiento de Información en el Algoritmo de JPEG.

La primera forma de para ocultar información en imágenes JPEG es mediante la aplicación de la técnica de LSB de forma secuencial e individualizar a los coeficientes cuántico DCT; la ventaja es que se puede almacenar grandes cantidades de información y su desventaja es que se puede detectar su modificación mediante ataques estadísticos. (Muñoz Muñoz, Privacidad y ocultación de información digital Esteganografía, 2017)

Otra alternativa para ocultar la información tiene que ver con el algoritmo F5 ideado por Andreas Westfield. Este algoritmo decrementa los valores de los coeficientes DCT, eligiéndolos con sumo cuidado para ocultar la información. Dado que F5 es uno del algoritmo más robusto es también el más difícil de detectar con ataques estadísticos y

visuales sin embargo, puede ser susceptible a otro tipo de ataques; uno de ellos fue realizado en un estudio realizado por Jessica Fridrich y un grupo de investigadores para romper sobre el algoritmo F5 dando como resultado que los coeficientes modificados puedan ser detectados mediante un proceso de recompresión de la imagen.

3. Herramientas Estenográficas en Imágenes Digitales.

Existe una gran cantidad de herramientas esteganográficas publicadas en internet sin embargo, sin una guía puede ser difícil elegir una sin saber las ventajas que posee. Con este fin el investigador Neil F. Johnson ha recolectado en su página web *https://www.jjtc.com/index.html* donde expone más de 100 herramientas esteganográficas que podemos encontrar en internet.

A continuación se expondrá alguna de ellas dando sus características principales y el uso práctico al guardar información.

3.1 Herramienta Esteganográfica S-Tools.

Esta herramienta esteganográficas fue desarrollada por Andy Brown en 1995 y una de las más populares en la época. Permite ocultar información en diferentes estegomedios digitales como en archivos de sonido .WAV o archivos de imagen .GIF y .BMP aplicando la técnica de LBS mediante la paleta de colores y seleccionando los bits de manera pseudoaleatoria. Jonhson (1995) comentó que Brown fue capaz de desarrollar una interfaz muy intuitiva para su uso a pesar de la limitación de recursos en la época.

La herramienta permite arrastrar un archivo de imagen .GIF, .BMP o .WAV a la pantalla principal. Luego se debe seleccionar el archivo de texto plano que se va a ocultar sobre la imagen, seguidamente se coloca una contraseña para guardar el archivo.

Acabado lo anterior se despliega la misma imagen pero con el mensaje guardado. Para revelar la información se da clic derecho sobre la imagen que oculta la información, se escoge la opción revelar y se coloca la contraseña; segundos después la información es revelada en otra ventana

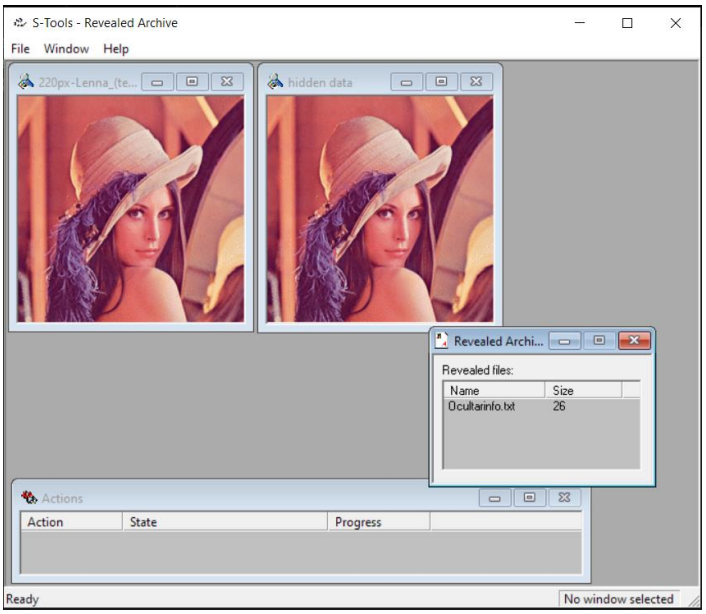

Figura 10 Herramienta S-Tools. (Oliva, 2020)

El ocultamiento en imágenes con esta herramienta tenía un patrón poco común, cuando se aplica la técnica LBS en imágenes GIF de 8 bits la paleta de colores se puede reordenar por los valores de luminiscencia y podemos observar que cada color tiende a ser el mismo pero variando 1 bit cada uno.

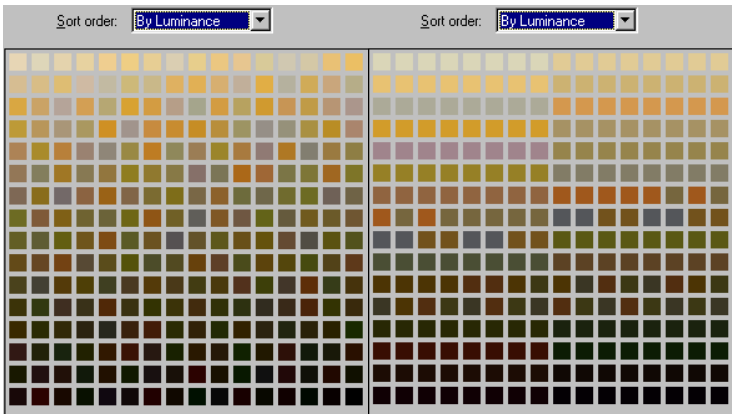

Figura 11 Imagen original a la izquierda y estegoimagen a la derecha ordenada por su luminancia. (Johnson & Jajodia, Steganalysis of Images Created Using Current Steganography Software, 1998)

El único limitante de esta herramienta es el tipo de formato que se puede utilizar ya que, solamente se limita a los formatos GIF y BMP en imágenes.

3.2 Herramienta Esteganográfica Digital Invisible Ink Toolkit

Esta herramienta está desarrollada en Java y es una de las más completas se puede encontrar en internet; según Muñoz (2017) esta herramienta permite usar varias técnicas esteganográficas basadas en LBS y a su vez ofrece ataques estegoanalíticos para comprobar que tan oculto se encuentra nuestra información en la imagen.

Figura 12 Página principal de la herramienta Digital Invisible Ink Toolkit. (Oliva, 2020)

En la página web *http://diit.sourceforge.net/* se puede encontrar la documentación de esta herramienta. Lo que llama la atención es que detalla de manera comprensiva los algoritmos que utiliza y la cantidad de información que podemos ocultar con esta herramienta. También tiene un apartado sobre el origen de esta herramienta la cual se inspira en el uso de la tinta invisible de forma digital y separa la idea principal de la esteganografía y el uso de la marca de agua (Watermarking).

La interfaz divide en cuatro pestañas las diversas funciones que podemos realizar. La pestaña Encode permite ocultar la imagen, primero se debe seleccionar el archivo de texto plano y la imagen que se va a utilizar. Seguidamente se coloca la contraseña, se elige el tipo de algoritmo, la cantidad de bits a modificar y donde la imagen va a ser guardada, Finalmente se presiona al botón Go para que el proceso inicie, una vez finalizado la herramienta permite ver la imagen final.

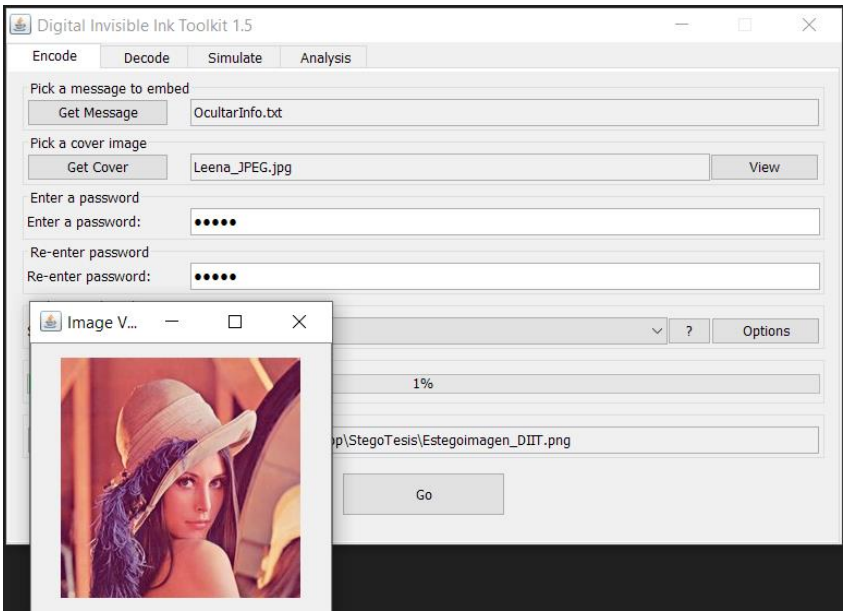

Figura 13 Ocultación de información con Digital Invisible Ink Toolkit. (Oliva, 2020)

Para recuperar la información se selecciona la pestaña Decode, primero se selecciona la estegoimagen, se coloca la contraseña y se selecciona el algoritmo que se utilizó para hacer el ocultamiento de la imagen. Finalmente se crea un nuevo un archivo de texto en blanco y se lo direcciona para recuperar la información.

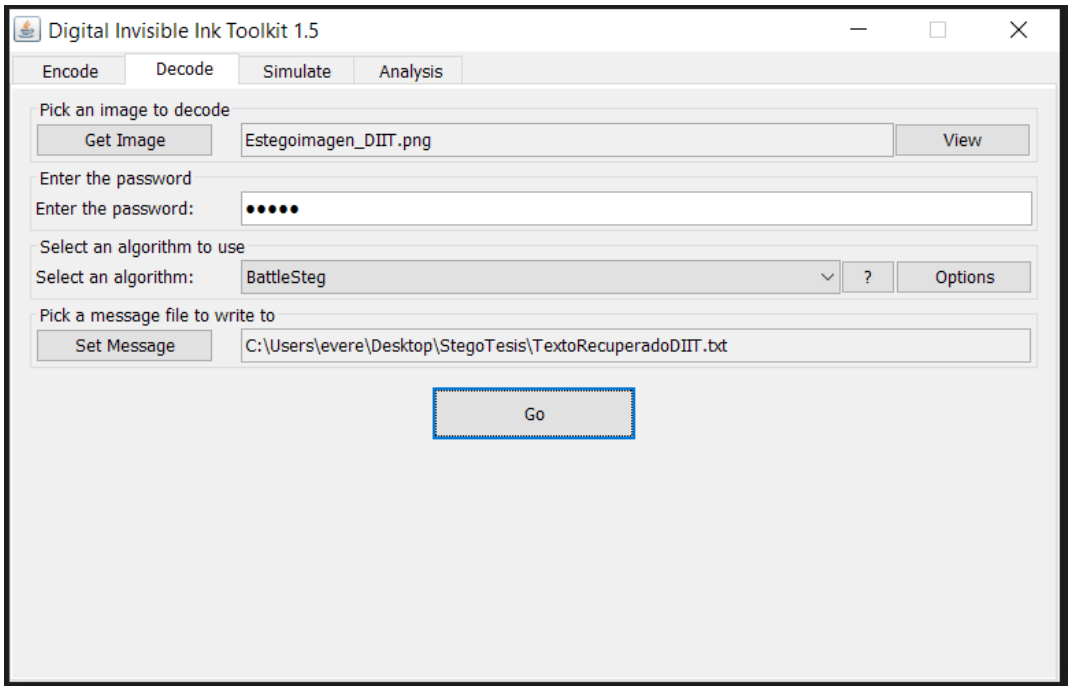

Figura 14 Recuperación de información con Digital Invisible Ink Toolkit. (Oliva, 2020)

La herramienta Digital Invisible Ink Toolkit es muy practica para entender cómo funciona distintos algoritmos LBS para ocultar información, es intuitiva y permite usar imágenes de formato PNG y JPEG que son los formatos más utilizados en internet. Adicionalmente permite usar el estegoanálisis en la misma herramienta, algo que no muchas herramientas ofrecen para conocer si el resultado de la imagen que pasa por un proceso esteganográfico fue efectivo o no.

3.3 Herramienta Esteganográfica F5

La herramienta F5 se destaca por ser la que oculta imágenes de formato JPEG de manera robusta y casi indetectable aun siendo atacadas por técnicas estegoanalíticas que involucre estadística o ataques visuales. Muñoz (2017) describe el algoritmo F5 desarrollado por Andreas Westfield en 2001 como la evolución de técnicas anteriores como lo fue F3 y F4. Dado que la imagen JPEG tiene una limitación para guardar grandes

cantidades de información F5 optimizo el uso de los coeficientes DCT de tal manera que los coeficientes elegidos mediante un algoritmo pseudoaleatorio son decrementados a un valor absoluto del coeficiente DCT.

F5 no cuenta con una interfaz gráfica para ser utilizado, por lo que se utiliza una ventana de comandos o terminal para su ejecución. Como la herramienta está desarrollada en Java se debe ejecutar el comando java -jar f5.jar, seguidamente se coloca el nombre del archivo plano a ocultar y la imagen de formato .JPG, tal cual podemos observar en la figura 3-6. Una vez ejecutado se podrá observar parte del proceso que realiza F5 para ocultar la información.

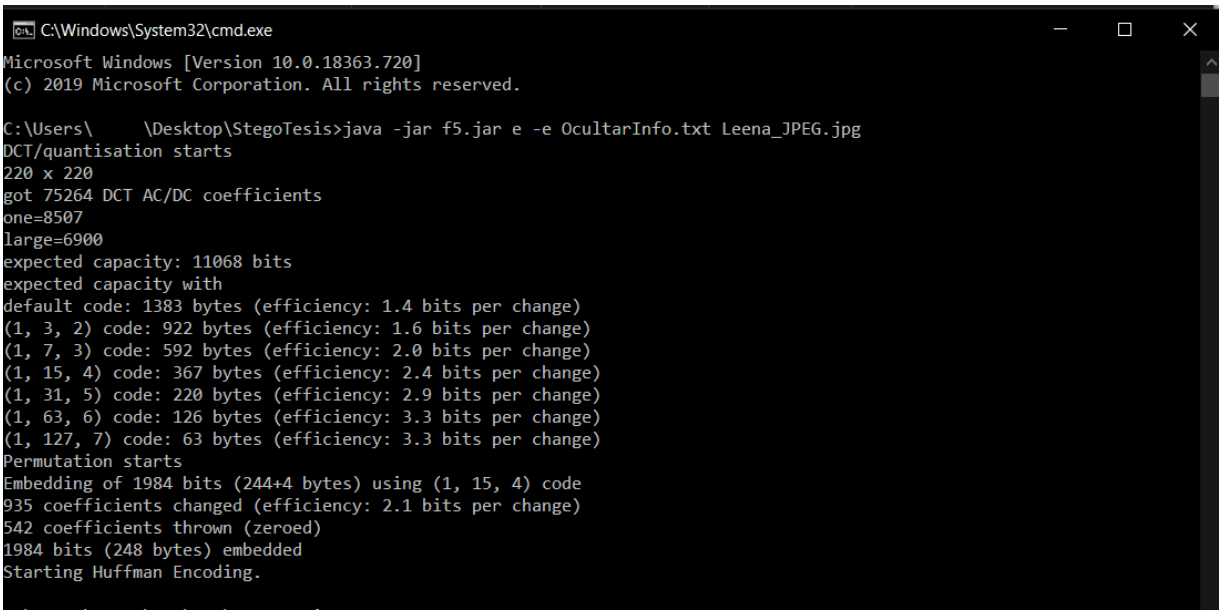

Figura 15 Ocultación de información con la herramienta F5. (Oliva, 2020)

Para recuperar la información ejecutamos el comando java -jar f5.jar, seguidamente se coloca el nuevo nombre del archivo plano para recuperar la información y de que imagen se va a recuperar, se puede observar en la figura 3-7 el resultado de su recuperación.

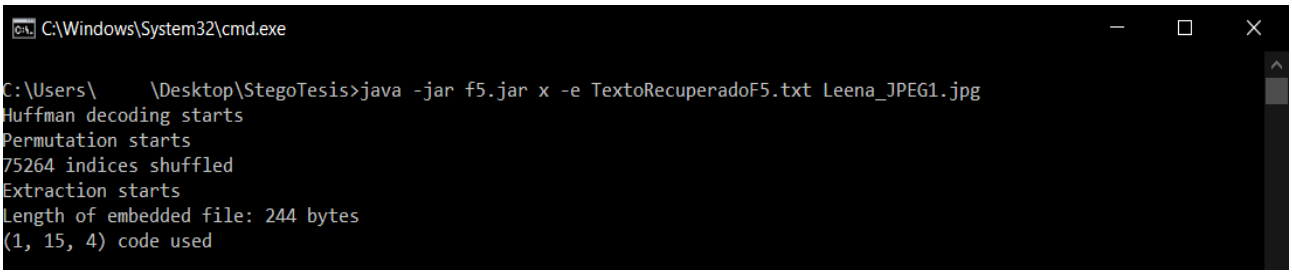

Figura 16 Recuperación de información con la herramienta F5. (Oliva, 2020)

La mayor desventaja de usar esta herramienta es su falta de interfaz gráfica para el usuario sin embargo, cumple con su objetivo de ocultar la información. Adicionalmente se pude colocar una contraseña a la estegoimagen mediante el flag -p para tener una estegoimagen más segura y especificar la resolución de la imagen con el flag -q si se quiere ampliar o disminuir la capacidad de almacenamiento.

4. Desarrollo de la Herramienta Esteganográfica.

La presente tesis surge a partir del problema del prisionero planteado con anterioridad en el capítulo 1.

Con la finalidad de resolver dicho problema se plantea crear una herramienta con una interfaz gráfica amigable e intuitiva que permita al usuario inserte en una imagen PNG un texto de archivo plano o de formato TXT mediante la técnica de sustitución por LBS. Adicionalmente el usuario generará una contraseña la cual solamente se podrá recuperará la información si la contraseña es igual a la que ingreso al ocultarlo.

4.1 Metodología de desarrollo.

Para el desarrollo de la herramienta esteganográfica se ha elegido el uso de la metodología en cascada debido a que:

- La herramienta tiene una estructura simple para su desarrollo.
- No requiere realizar un mantenimiento en la herramienta a lo largo del tiempo.
- Se va a seguir una planificación y documentar todo lo realizado en el proceso.

4.2 Análisis.

Para realizar la herramienta se ha identificado las siguientes funcionalidades:

- F1 Ocultar información
- F2 Recuperar información

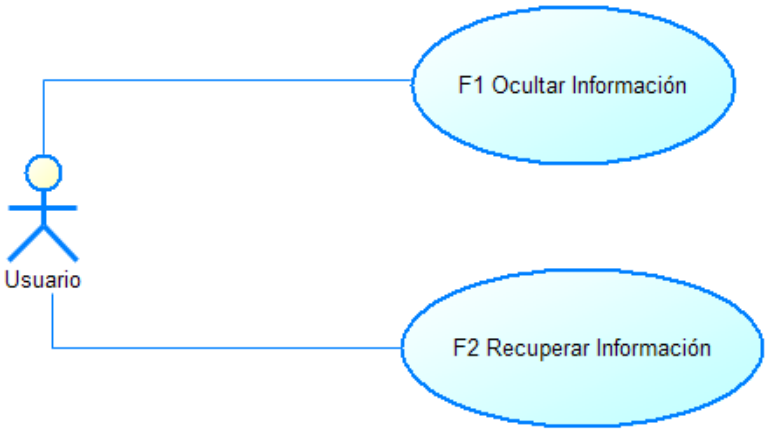

Figura 17 Diagrama General – Caso de Uso (Oliva, 2020)

4.2.1 Casos de Uso a Detalle

A continuación se detalla los casos de uso de la herramienta de cada una de sus funcionalidades.

4.2.2 Caso de uso – Ocultar Información

Nombre		Ocultar Información
Autor		Adrián Oliva
Actores		Usuario
Descripción: Permite ocultar la información en la imagen		
Flujo Principal		
1	El actor selecciona la pestaña ‘Ocultar’	
2	El actor selecciona la imagen	
3	El actor selecciona el archivo de texto plano. (3.1)	
4	El actor presiona el botón ‘Ocultar	
5	El sistema oculta el texto en la imagen.	
6	El sistema despliega un mensaje de completado (E1)	
Flujo Alterno		
3.1	El actor introduce una contraseña	
Excepción	Causa	Mensaje
E1	Sobre exceso de información a guardar	El mensaje es demasiado grande para ser guardado

4.2.3 Caso de uso – Recuperar Información

Nombre	Recuperar Información	
Autor	Adrián Oliva	
Actores	Usuario	
Descripción: Permite recuperar la información en la imagen		
Flujo Principal		
1	El actor selecciona la pestaña ‘Recuperar	
2	El actor selecciona la imagen (2.1)	
3	El actor introduce la contraseña	
4	El actor presiona el botón ‘Recuperar’	
5	El sistema despliega el texto oculto	
6	El sistema despliega un mensaje de completado (E1, E2)	
Flujo Alterno		
2.1	El actor introduce una contraseña	
Excepción	Causa	Mensaje
E1	Contraseña vacía	Ingrese la contraseña
E2	Contraseña incorrecta	Error al Recuperar

4.3 Diseño

Después de ser definidos los requerimientos pasamos a el diseño de la herramienta donde se podrá observar con mayor detalle cómo está constituido cada funcionalidad.

Para entender cómo funciona la herramienta se realizará un diagrama de flujo para representar la secuencia de actividades de manera ordenada que debe seguir el programa.

También se realizará un diseño de interfaz gráfica el cual se usará para que el usuario interactúe con la herramienta de una forma amigable

4.3.1 Diagrama de Flujo

Siendo un programa no muy complejo se puede elaborar un diagrama de flujo el cual permite entender cómo trabaja cada proceso de la herramienta.

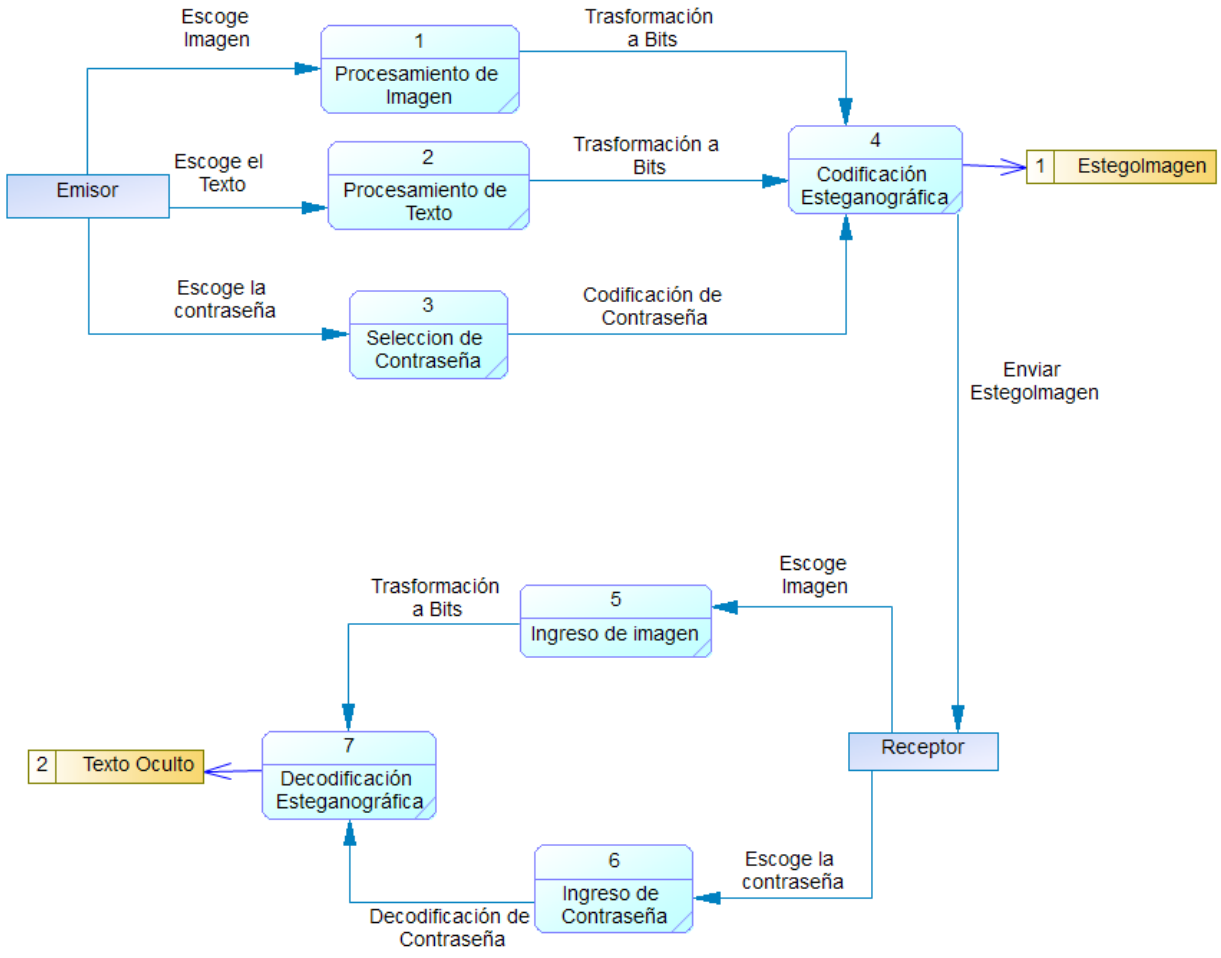

Figura 18 Diagrama de Flujo (Oliva, 2020)

4.3.2 Diseño de la Interfaz Gráfica

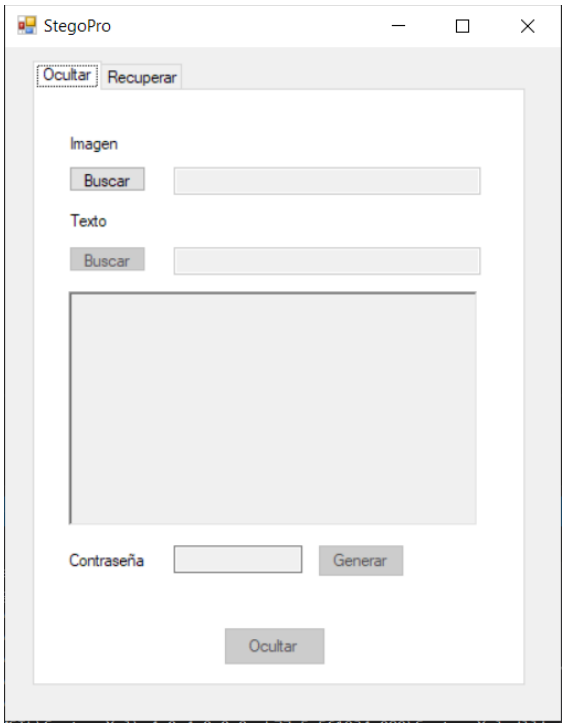

Figura 19 Interfaz de Ocultamiento (Oliva, 2020)

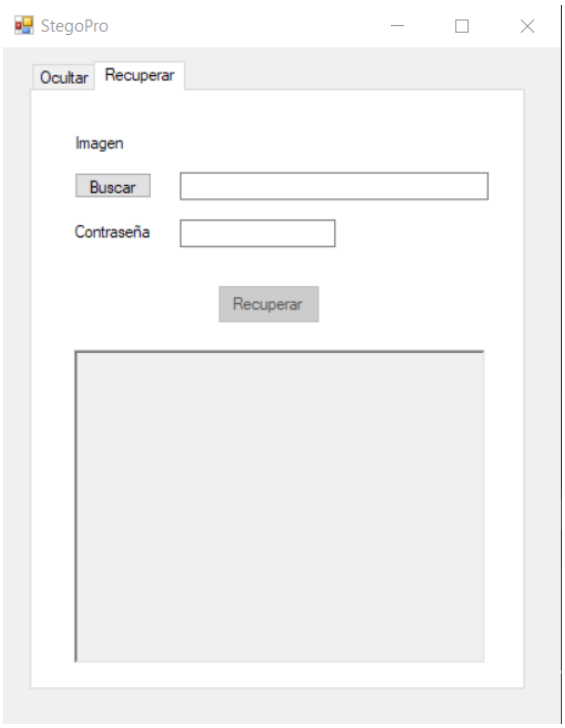

Figura 20 Interfaz de Recuperación (Oliva, 2020)

4.4 Programación.

4.4.1 Pre-Programación

Para el desarrollo de la herramienta se ha elegido el lenguaje de programación C# y el IDE de Visual Studio. La interfaz se creará con Windows Forms que se incluye como parte de los paquetes que ofrece .NET

Dado que la técnica esteganográfica que se utilizará es la sustitución por LBS descrito en el capítulo 2; se tendrá que adicionar la introducción de la contraseña para ocultar y recuperar la información. Para eso se realizará una función que genere una contraseña de cinco caracteres que constará de letras minúsculas y números del uno al nueve y se adaptará el algoritmo de LBS para ocultar y recuperara la contraseña.

4.4.2 Procesos no funcionales

Al desarrollar el programa se presentaron dos procesos nos funcionales los cuales se tuvieron que adaptar para la aplicación del algoritmo.

La primera situación fue el ocultar la contraseña dentro de imagen y recuperarla para verificar si esta es correcta o no. Dado que ocultamos una palabra cada 3 pixeles se tomó los primeros 15 pixeles para colocar la contraseña. Esta situación no está establecida dentro del algoritmo LBS por lo que su implementación fue una decisión personal al incluirlo en los primeros pixeles.

La segunda situación tuvo que ver con el código de caracteres usando ASCII. Ya que cada letra se trasformaba a su representación en ASCII para luego ser nuevamente trasformada en binario existió un inconveniente, los caracteres especiales como las

vocales con tilde y la ñ se las representaba como dos números distintos pero al estar seguidos representaba el carácter correcto. La solución más extensa fue usar los caracteres como parte de la representación que la memoria ofrece pero esto hacía que el programa demorara incluso más si se trataba de textos grandes. La segunda opción fue el codificar estos dos números dentro de la imagen y al recuperar la información se debía leerlos seguidos para que al trasformar de binario a un carácter de como resultado el carácter especial.

4.4.3 Algoritmo Ocultar

Variables

```
Entero      bit ← 0
            indice ← 0
            rojo ← 0
            verde ← 0
            azul ← 0
Booleano    terminado ← falso
Byte        arr_bits_char[8]
            Info[] ← Concatenar(obtenerContra(), obtenerMensaje(),255)
Bitmap      img ← obtenerImagen(ruta)
Color       pixel
Inicio
```

```
Para i ← 0 hasta i < Altura(img) incrementar 1
    Para j ← 0 hasta j < Ancho(img) incrementar 1
        Si (bit = 0 y indice < Longitud(info))
            arr_bits_char ← obtener_bits(info[indice])
            indice ← indice + 1
        Sino_Si (bit = 0 y indice = Longitud(info))
            Terminado ← verdadero
            Interrumpir
        Fin_Si
        pixel ← obtenerPixel(i,j)
        rojo ← modificar_color(pixel, 'R', arr_bits_char[bit])
        bit ← bit + 1
        verde ← modificar_color(pixel, 'G', arr_bits_char[bit])
        bit ← bit + 1
        Si (bit < Longitud(arr_bits_char))
            azul ← modificar_color(pixel, 'B', arr_bits_char[bit])
            bit ← bit + 1
        Sino
```

```
                    bit ← 0

                    azul ←obtenerAzul(pixel)

                Fin_Si

                colocarPixel(img,i,j, colocarColor(rojo,verde,azul))

            Fin_Para

            Si (terminado = verdadero)

                interrumpir

            Fin_si

        Fin_Para

Fin
```

4.4.4 Algoritmo Recuperar

```
Variables

Lista       lista_bytes ← listaVacia()

Cadena      cadena_bytes← “”

            cadena_anterior ← “”

            cadena_final ← “11111111”

            contra ← obtenerContra()

            mensaje ← “”
```

```
Booleano    finalizado ← falso

Byte        rojo ← 0

            verde ← 0

            azul ← 0

Bitmap      img ← obtenerImagen(ruta)

Color       pixel

Inicio

    Para i ← 0 hasta i < Altura(img) incrementar 1

        Para j ← 0 hasta j < Ancho(img) incrementar 1

            pixel ← obtenerPixel(i,j)

            rojo ← obtener_ultimo_bit(pixel,'R')

            cadena_byte ←Concatenar(rojo)

            verde ← obtener_ultimo_bit(pixel,'G')

            cadena_byte ←Concatenar(verde)

            Si (Longitud(cadena_bytes) < 8)

                azul ← obtener_ultimo_bit(pixel,'B')

                cadena_byte ←Concatenar(azul)

            Sino
```

```
                cadena_anterior ←cadena_bytes
                Si (cadena_anterior = cadena_final)
                        finalizo ← verdadero
                        interrumpir
                Sino
                        Añadir(lista_bytes, ConvertirByte(cadena_byte))
                Fin_Si
                cadena_bytes ← “”
        Fin_Si
        Si (i = 0 y j = 14)
                Si (obtenerCadena(lista_bytes) = contra)
                        lista_bytes ← listaVacia()
        Sino
                finalizo ← verdadero
                interrumpir
        Fin_Si
Fin_Para
```

```
        Si (finalizo = verdadero)
            mensaje ← obtenerCadena(lista_bytes)
            Interrumpir
        Fin_Si
    Fin_Para
Fin
```

4.5 Pruebas

Dada las anteriores fases el programa final está listo para ser probado. Las pruebas del programa pasarán por 3 etapas las cuales se irán describiendo a continuación.

4.5.1 Etapa 1

Para la etapa 1 se insertará un texto corto de 156 palabras en una imagen de 220 por 220 pixeles. El texto para ocultar estará conformado por las siguientes palabras:

> The quick brown fox jumps over the lazy dog
>
> El veloz murciélago hindú comía feliz cardillo y kiwi. La cigüeña tocaba el saxofón detrás del palenque de paja

Estas dos frases han sido elegidas debido a que cumplen con el uso de todas las letras del abecedario que va de la A a la Z incluyendo caracteres especiales como tildes, diéresis y la ñ.

Al elegir el texto este se debe desplegar en el programa y al generar una contraseña este debe generar una contraseña de cinco caracteres entre letras minúsculas y números del uno al diez.

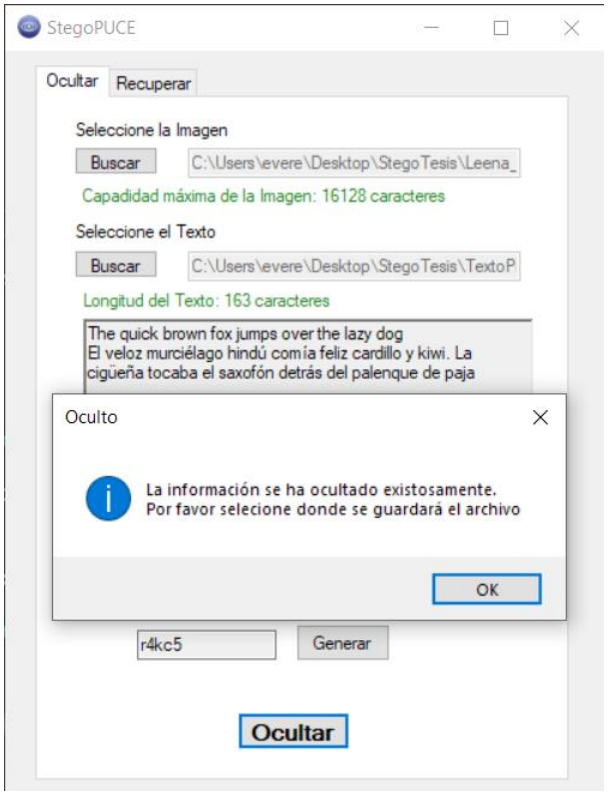

Figura 21 Prueba de ocultamiento de información (Oliva, 2020)

El resultado observado en la figura 4-5 se da cuando el proceso es satisfactorio debido a que la operación de ocultamiento se ha completado sin ningún problema. En la figura 4-6 se puede observar el resultado de la imagen guardada.

Figura 22 Imagen derecha usada para ocultar información, imagen izquierda con la información oculta. (Oliva, 2020)

Una vez que se ha guardado la imagen se procede a recuperar el texto con la contraseña de la figura anterior. En esta parte de la prueba se dan tres situaciones.

El primer caso se lo representa en la figura 4-7 donde no se ha ingresado la contraseña, en respuesta el sistema dará una advertencia para que el usuario conozca cual fue el error en el proceso de recuperación.

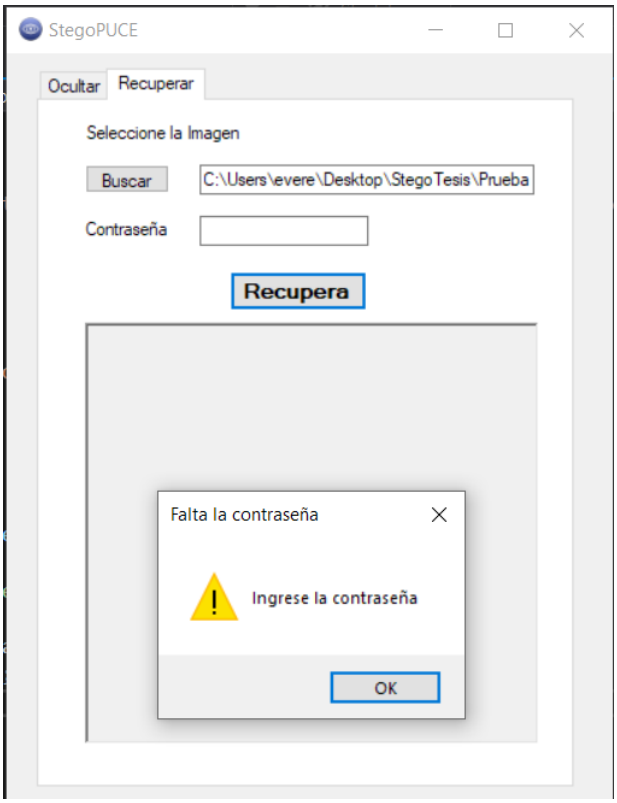

Figura 23 Prueba de recuperación de información sin contraseña (Oliva, 2020)

El segundo caso se da cuando se ingresa una contraseña que no corresponde a la imagen, dado que la contraseña está oculto dentro de la imagen la contraseña oculta debe coincidir con la que el usuario escribe. Para este caso entonces el sistema despliega un mensaje de error como se observa en la figura 4-8 la cual indica al usuario que la contraseña no es válida.

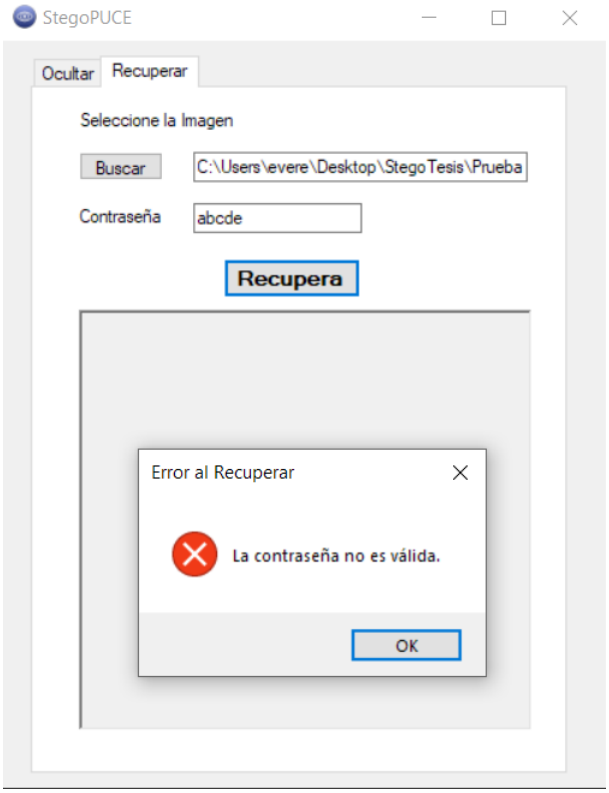

Figura 24 Prueba de recuperación de información con contraseña no valida (Oliva, 2020)

Finalmente para el tercer caso donde la contraseña sea correcta se desplegará el mensaje de recuperado y la información guardada en la imagen se recuperará desplegándolo en el programa como se puede observar en la figura 4-9.

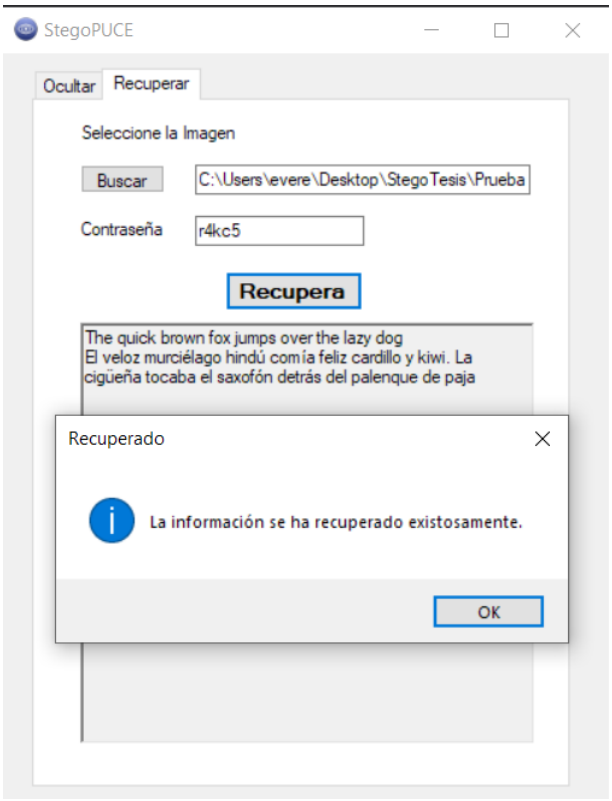

Figura 25 Prueba de recuperación de información con contraseña correcta (Oliva, 2020)

Se da por concluido que la etapa 1 se ha completado sin ningún error en el proceso de ocultar y recuperar el texto.

4.5.2 Etapa 2

Para la etapa 2 se ingresará un texto que supere el límite de la imagen de 220 por 220 pixeles. Este texto contara con los primeros 35 capítulos más la introducción del libro El Código Da Vinci escrito por Dan Brown. Como resultado de esta prueba se tendrá lo siguiente.

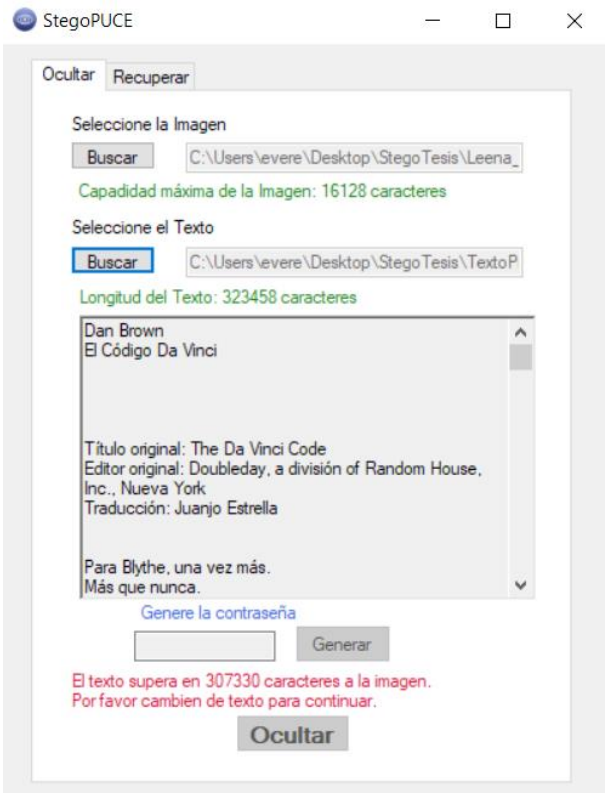

Figura 26 Prueba de ocultamiento de información que supere la capacidad máxima de la imagen (Oliva, 2020)

Debido a que dentro de la imagen podemos ingresar hasta 16128 caracteres y el texto sobrepasa al límite de la imagen en 308207 caracteres los botones de generar contraseña y ocultar estarán bloqueados hasta que se seleccione otra imagen que permita guardar toda la información o que se cambie el texto.

Se puede concluir que esta prueba es un éxito ya que no permite ingresar más información de lo que la imagen es capaz de guardar.

4.5.3 Etapa 3

En la última prueba se ocultará el texto de los 35 capítulos más la introducción del libro El Código Da Vinci escrito por Dan Brown en una imagen de 1600 por 900 pixeles.

Al ocultar la información en la imagen se despliega el mensaje de éxito al ocultar por lo que la operación de ocultar un texto grande en una imagen grande no tiene ningún inconveniente sin embargo, el tiempo que tomo en ocultar la información ha tardado unos segundos más que al hacerlo con un texto pequeño.

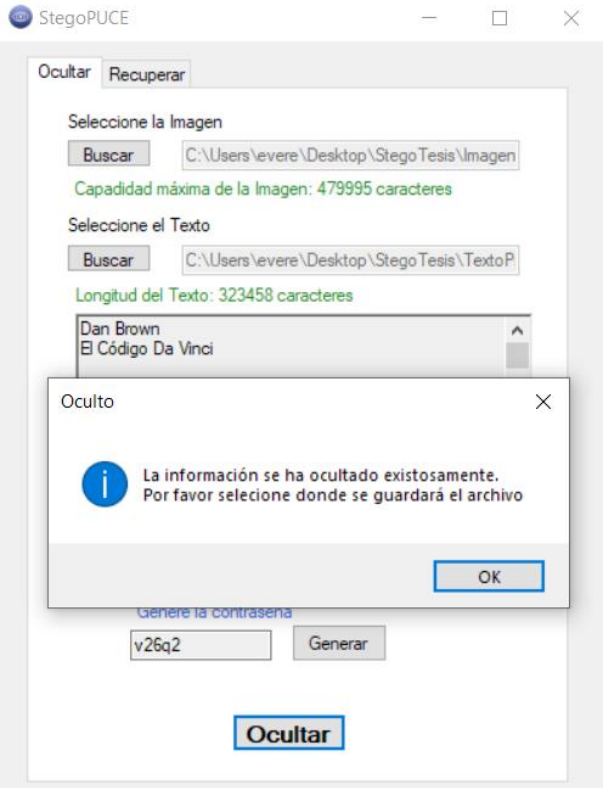

Figura 27 Prueba de ocultamiento de información de un texto e imagen grande

(Oliva, 2020)

Al recuperar la información de la imagen se despliega el mensaje de éxito al recuperar por lo que la operación de recuperar un texto grande en una imagen grande no tiene ningún inconveniente sin embargo, el tiempo que tomo en recuperar la información ha tardado unos segundos más que al hacerlo con un texto pequeño.

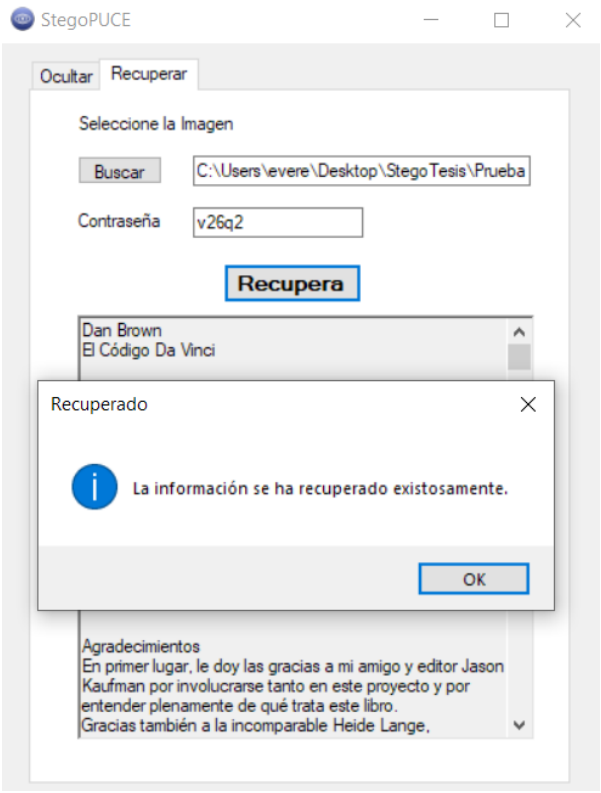

Figura 28 Prueba de recuperación de información de texto e imagen grande

(Oliva, 2020)

En conclusión la información ocultar y recuperar exitosamente sin embargo, el tiempo que demoró en hacer ambos procesos se debe a que el texto tiene una mayor cantidad de caracteres y dado que cada carácter es guardado cada 3 pixeles el tiempo en relación con la cantidad de caracteres será proporcional para concluir los procesos de ocultar y recuperar; al no concluir en algún error el objetivo de esta etapa es un éxito.

4.5.4 Comparación de resultados

Resultado 1	Imagen Original	Estegoimagen
Imagen		
Tamaño	88.235 bytes	134.614 bytes
Hash	7B898D02011194942D0B F600BF92332B5A7C4558	8C0F7C2E8FF237329DEE C3C8CFF1DEF76AB1338A

Resultado 2	Imagen Original	Estegoimagen
Imagen		
Tamaño	1.772.761 bytes	2.543.857 bytes
Hash	FD5AB8D57C5A135FF08D 850C08628EB09ABC1ABA	8F51F2D52426E1080EF9 B8D0C5EB43814F761CAC

5. Conclusiones y Recomendaciones

5.1 Conclusiones

La investigación sobre esteganografía lleva a las siguientes conclusiones:

- El desarrollo de la herramienta esteganográfica que permita ocultar información a través de imágenes se dio como consecuencia de una investigación previa, esto permitió reunir varios conceptos los cuales, fueron aplicados es su elaboración.
- El uso del algoritmo por sustitución del bit menos significativo ayudo a comprender como podemos ocultar la información en imágenes a través de los bits que lo conforman y a su vez, este algoritmo puede ser usado en otros contenidos multimedia como audio y video.
- La representación visual del problema del prisionero a través de un flujo de datos permitió identificar los procesos básicos que tiene que llevar acabo cualquier herramienta esteganográfica para cumplir con la finalidad de ocultar la información.
- Las pruebas realizadas para corregir el funcionamiento de la herramienta sirvieron para optimizar las funciones implementadas de ocultamiento y recuperación de la información, así como corregir errores que se puedan presentar durante la ejecución y en la interfaz gráfica.
- El campo de la esteganografía aún es poco popular en la ciencia de la computación, no ha tenido oportunidad de presentarse en situaciones reales para la seguridad de datos debido a que se puede vulnerar los estegomedios aplicando varias técnicas y esto ha permitido llevado a cabo varias investigaciones para lograr algoritmos que permitan optimizar el ocultamiento de información de tal

manera que, sea seguro y confiable utilizar la esteganografía para la seguridad de datos.

- Los conceptos de programación aprendidos y el uso de diferentes herramientas y técnicas para el desarrollo de software en el transcurso de la carrera ha hecho posible la entrega de la herramienta esteganográfica.
- El poco o nulo conocimiento que poseen las personas sobre esteganografía llega a ser perjudicial. Su uso para fines inapropiados como: la fuga de información, implantación de malware a través de estego-programas o para uso de terrorismo están presentes con esta tecnología que no es nueva y aun así, su falta de popularidad ha hecho que se utilice muchas veces con estos fines. Aunque sus ventajas para ocultar información siempre estarán presentes para proteger o perjudicar, se debe enfocar a que el uso de la esteganografía sea un salvavidas para el actual panorama que se está enfrentando la protección de datos y ser discutido en ámbitos técnicos, legales y éticos para su apropiado uso.

5.2 Recomendaciones

- Dado que se obtiene los pixeles de la imagen como un arreglo de bits, se puede elegir la forma y la manera en la que la información sea oculta y recuperada, esta consideración puede ser aplicada para hacer que la herramienta sea más robusta.
- Al usar el algoritmo de sustitución por bits menos significativos tiene la ventaja de poder guardar una gran cantidad de caracteres en la imagen sin embargo, entre más se quiera guardar más tiempo requerirá por lo que, optimizar el uso de los bits se ha vuelto el objetivo último entre la comunidad esteganográfica.

- Se puede hacer uso de la criptografía para mantener mejor guardado nuestros mensajes, el uso de ambos, criptografía como esteganografía llevaría a que las herramientas sean más robustas sin embargo, también requerirá más tiempo para realizar ambas operaciones.
- Mediante la presente investigación y el desarrollo de la herramienta se puede aplicar otros algoritmos y técnicas con el fin de indagar y crear nuevos algoritmos para optimizar el ocultamiento y la recuperación de información con esteganografía.

Bibliografía

Cedillo, J. (2019). *Enlace Judío*. Obtenido de https://www.enlacejudio.com/2019/05/16/prisma-microdots-obra-maestra-del-espionaje-aleman-durante-la-segunda-guerra-mundial/

Computerphile. (22 de Mayo de 2015). JPEG DCT, Discrete Cosine Transform (JPEG Pt2)- Computerphile. Obtenido de https://www.youtube.com/watch?v=Q2aEzeMDHMA

DataGenetics. (2012). Obtenido de http://datagenetics.com/blog/march12012/index.html

Elgabar, E., & Alamin , H. (2013). Comparison of LSB Steganography in GIF and. *vol. 3(no. 4)*, 79–83. International Journal of Soft Computing and Engineering (IJSCE).

Gómez Vieites, Á. (2014). *Sistemas seguro de acceso y transmisión de datos.* España: RA-MA.

Gómez, M. (04 de 2017). *Interartive*. Obtenido de https://interartive.org/2017/04/historias-de-la-imagen-digital-marisa-gomez

González, R., & Woods, R. (2008). *Digital Image Processing Third Edition.* Londres: Pearson Education.

Hempstal, K. (2005). *Digital Invisible Ink Toolkit*. Obtenido de Digital Invisible Ink Toolkit: http://diit.sourceforge.net/background.html

Johnson, N. (1995). *STEGANOGRAPHY SOFTWARE*. Obtenido de STEGANOGRAPHY SOFTWARE: https://www.jjtc.com/Steganography/tools.html

Johnson, N., & Jajodia, S. (1998). *Steganalysis of Images Created Using Current Steganography Software*. Obtenido de Steganalysis of Images Created Using Current Steganography Software: https://www.jjtc.com/ihws98/jjgmu.html

Lyra, M. (2011). *ResearchGate*. Obtenido de https://www.researchgate.net/publication/221918148_MATLAB_as_a_Tool_in_Nuclear_Medicine_Image_Processing

Muñoz Muñoz, A. (2014). *Crypt4you*. Obtenido de http://www.criptored.upm.es/crypt4you/temas/privacidad-proteccion/leccion7/leccion7.html

Muñoz Muñoz, A. (2017). *Privacidad y ocultación de información digital Esteganografía.* Colombia: RA-MA.

Oliva, A. (2020). Creación de una herramienta para ocultamiento de información a través de imágenes. Quito.

Provos, N., & Honeyman, P. (2001). *Detecting Steganographic Content on the Internet.*

Reyes, M. (2016). *SliderPlayer*. Obtenido de https://slideplayer.es/slide/5478236

Rodríguez Morales, R., & Sossa Azuela, J. (2012). *Procesamiento y análisis digital de imágenes.* México: Alfaomega.

Weiss, Y. (2016). *O'Reilly*. Obtenido de https://www.oreilly.com/library/view/high-performance-images/9781491925799/ch04.html

Westfeld, A. (2011). *f5-steganography*. Obtenido de f5-steganography: https://code.google.com/archive/p/f5-steganography/

Zegarra, J. (2017). *IT/Users*. Obtenido de https://itusers.today/hackers-utilizan-esteganografia-ocultar-informacion-robada/

Anexo

Manual de usuario

Ocultar información

Primero se debe situar en la pestaña ‘Ocultar’ como se indica la ilustración 1.

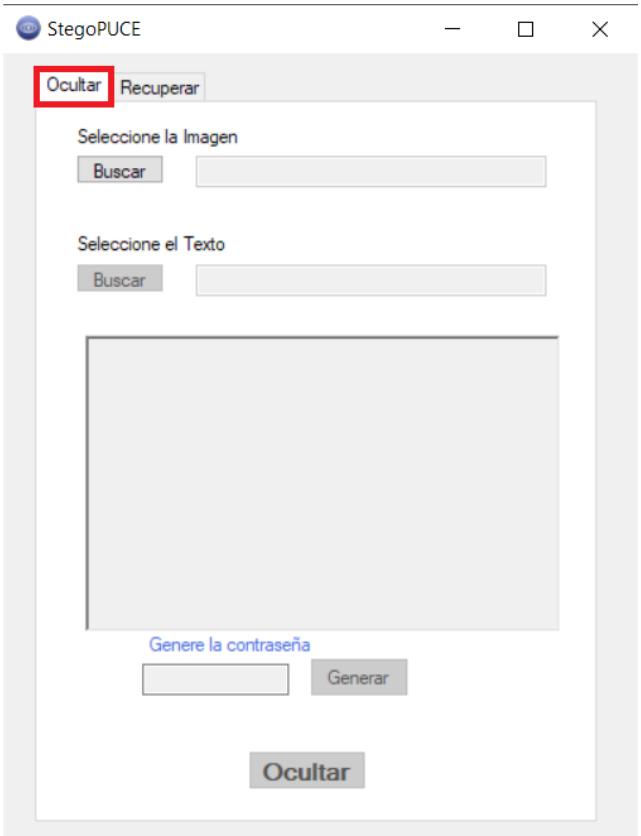

Ilustración 1 Pestaña ocultar

Se debe buscar la imagen en formato PNG y el texto a guardar con los botones de ‘Buscar’ marcador en la ilustración 2. El texto se desplegará en la parte de abajo indicando lo que se ha seleccionado.

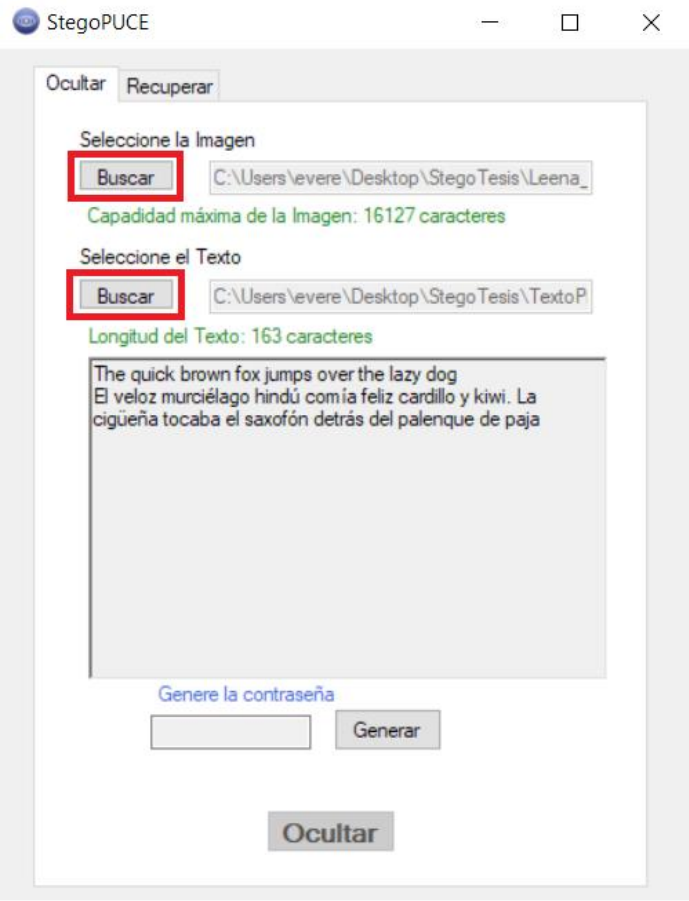

Ilustración 2 Selección de imagen y texto

Seguidamente se debe generar una contraseña con el botón de 'Generar', este se encuentra marcado en la ilustración 3. En el caso de que la contraseña sea muy fácil o compleja se puede volver a generar otra contraseña antes de ocultar la información.

Se recomienda guardar la contraseña generada para recuperar la información caso contrario no se podrá recuperar al momento de hacer la verificación de contraseña.

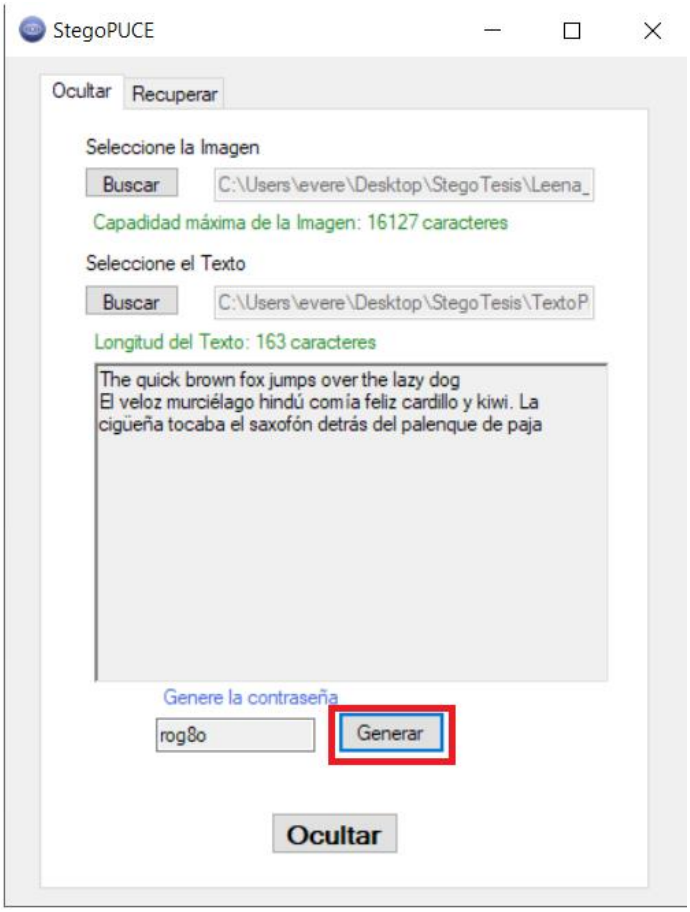

Ilustración 3 Generación de contraseña

Por ultimo se debe presionar en el botón de 'Ocultar' marcada en la ilustración 4, para que el proceso de guardar la información en la imagen comience. Una vez concluido aparecerá un mensaje con la tarea completada como la ilustración 5 y seguidamente podremos elegir la ubicación donde se la guardará

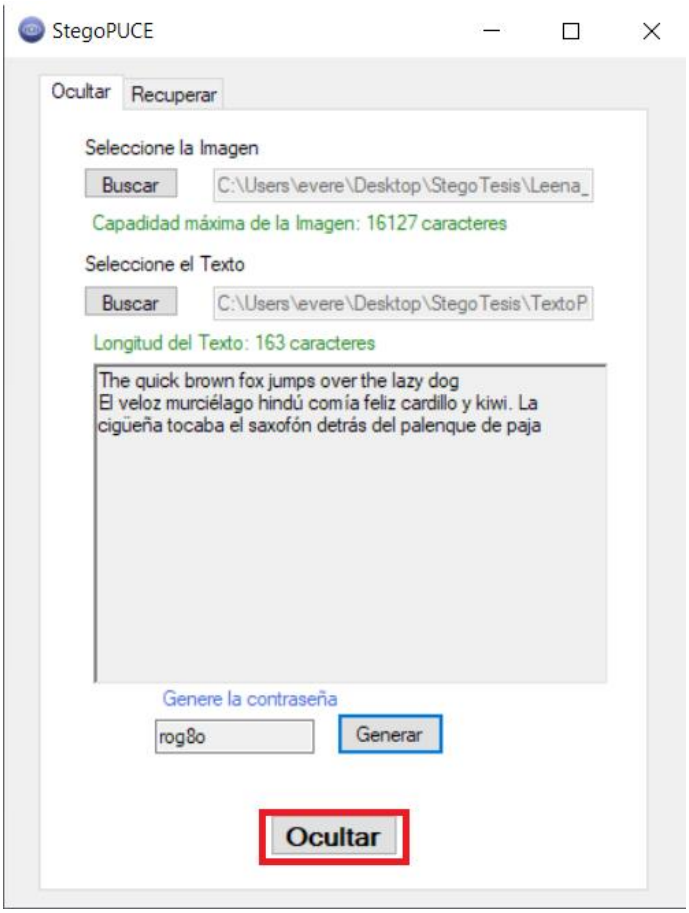

Ilustración 4 Pulsar el botón 'Ocultar para guardar

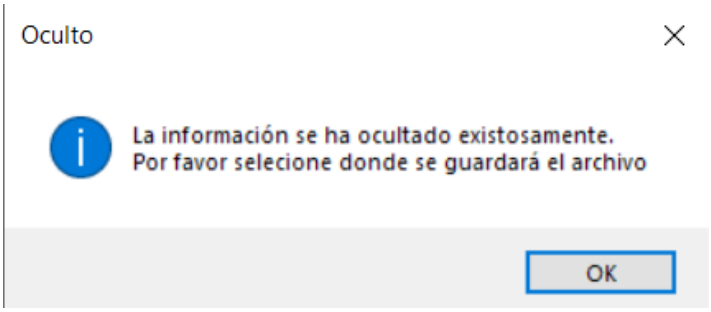

Ilustración 5 Mensaje de finalización

En el caso de que se el número de caracteres supere el tamaño máximo de almacenamiento, como en la ilustración 6, la herramienta no permitirá que se genere la

contraseña ni se guarde hasta que se cambie a una imagen con la capacidad para el almacenamiento o se cambie el texto.

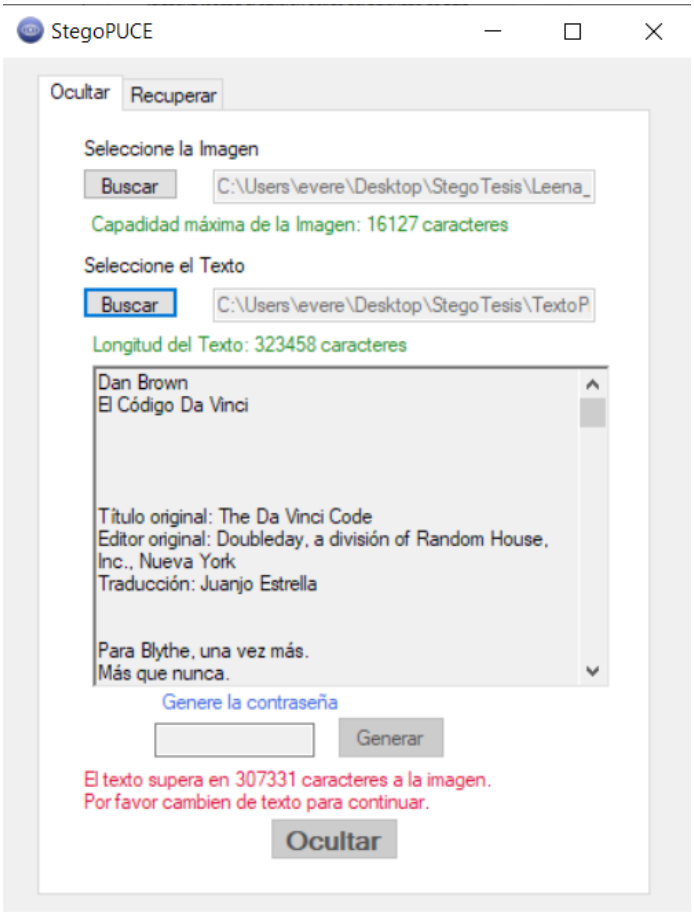

Ilustración 6 Error al superar la capacidad máxima de caracteres

Recuperar información

Primero se debe situar en la pestaña ‘Recuperar’ como se indica la ilustración 7.

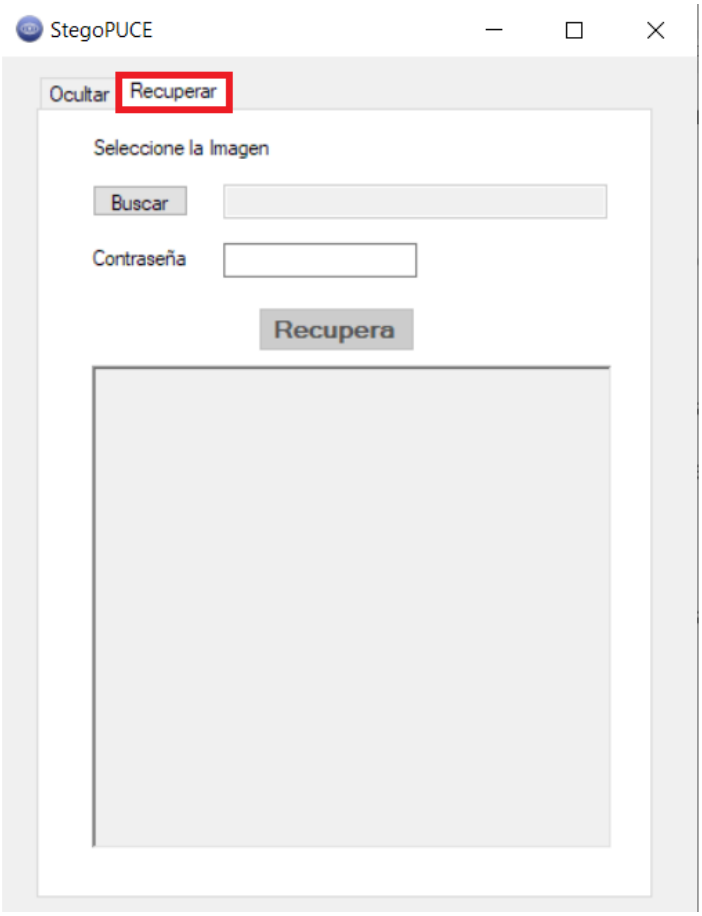

Ilustración 7 Pestaña recuperar

Se debe seleccionar la imagen que contenga la información guardada con el botón de ‘Buscar’ marcado en la ilustración 8.

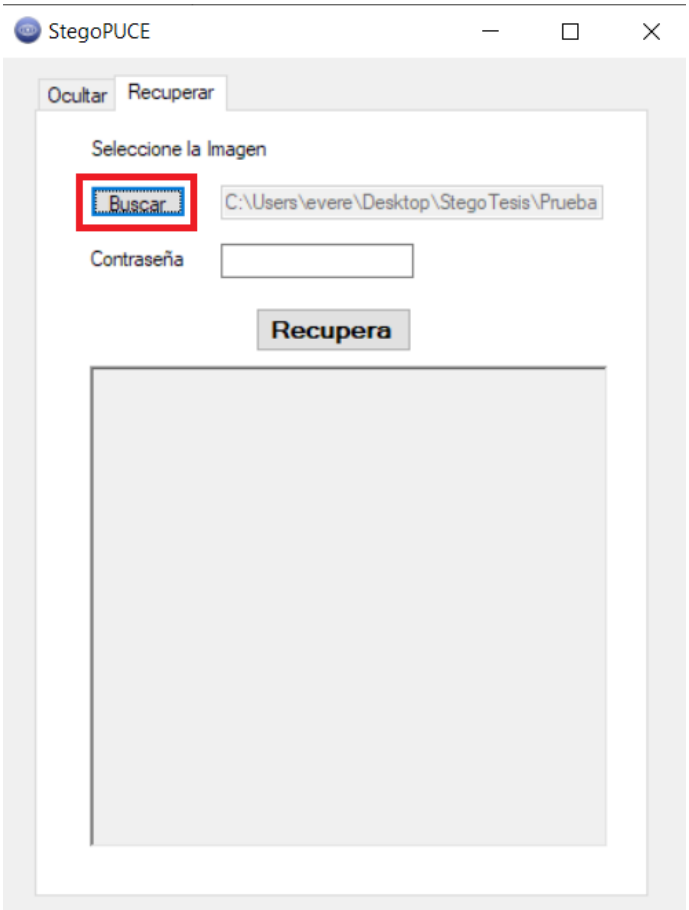

Ilustración 8 Buscar imagen

A continuación se coloca la contraseña en el cuadro correspondiente y se recupera la información con el botón ‘Recuperar’, como está marcado en la ilustración 9.

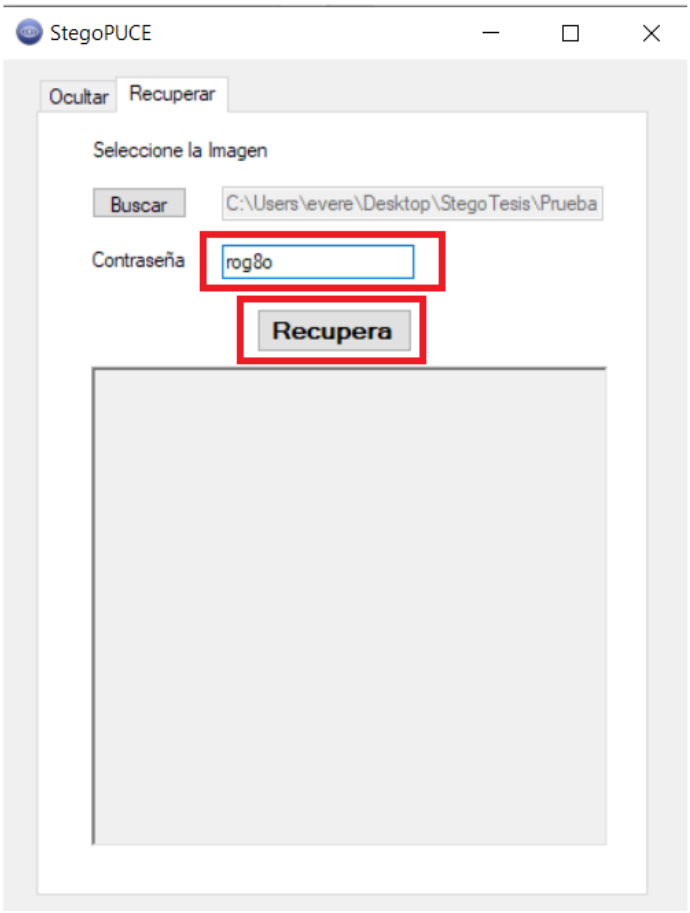

Ilustración 9 Colocar contraseña y pulsar el botón 'Recuperar'

Si la contraseña corresponde a la imagen, la herramienta indicará un mensaje como en la ilustración 10 y el mensaje se desplegará en la herramienta como en la ilustración 11.

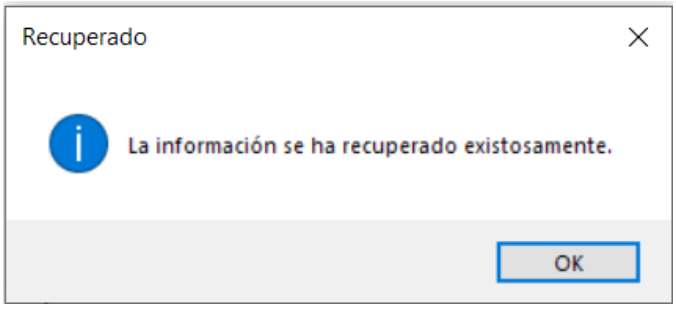

Ilustración 10 Mensaje de finalización exitoso

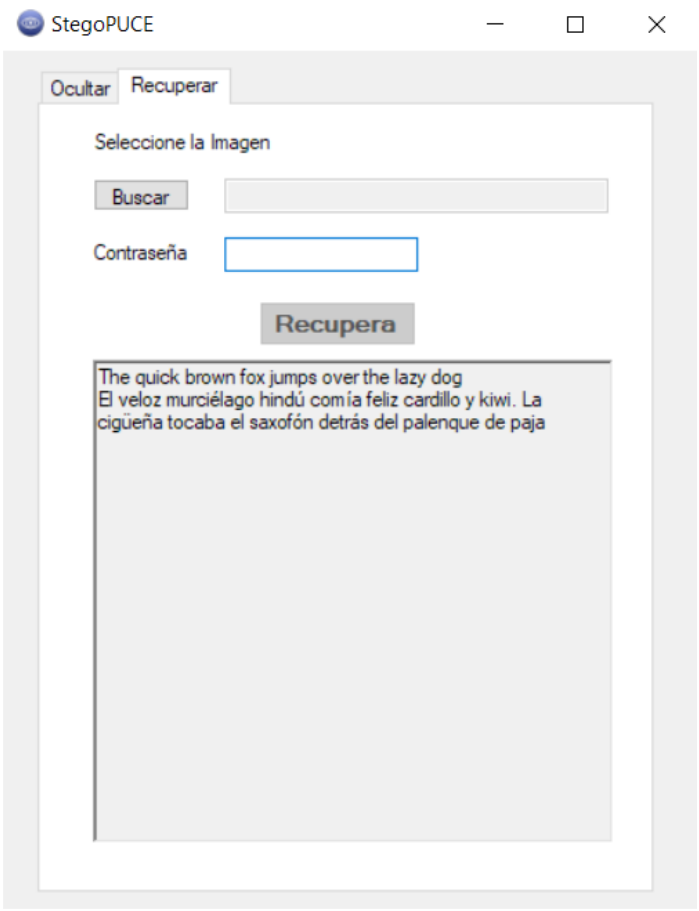

Ilustración 11 Despliegue del mensaje recuperado

En el caso de que no se introduzca una contraseña la herramienta indicara el mensaje como en la ilustración 12.

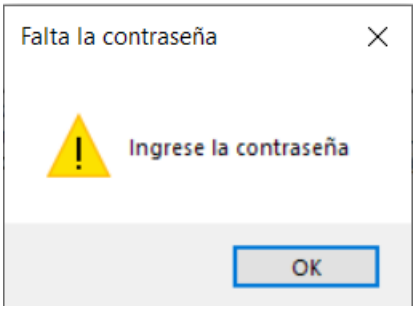

Ilustración 12 Mensaje de advertencia por falta de contraseña

En el caso de que se introduzca otra contraseña la herramienta indicara el mensaje como en la ilustración 13.

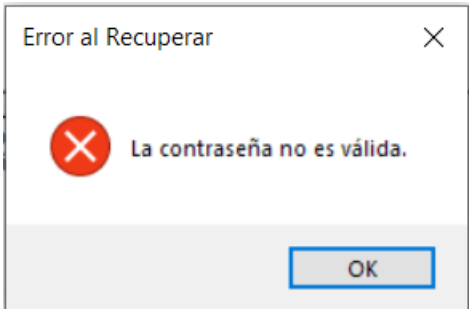

Ilustración 13 Mensaje de error por contraseña incorrecta