Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

SISTEMA DE SEGURIDAD ESTEGANOGRAFICO

TESIS

PARA OBTENER EL TÍTULO DE:

LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

Miguel Angel Raymundo Martínez

Dirigida por:

Dra. Bárbara Emma Sánchez Rinza

Fecha:

Enero 2024

Índice

INTRODUCCIÓN

Actualmente, la tecnología ha cambiado, la era digital nos ha proporcionado nuevos dispositivos, nuevas maneras de comunicarnos, incluso tenemos una cantidad inmensa de servicios al alcance de nuestra mano; todo esto nos ha llevado a la creciente necesidad de proteger nuestra información ya que existen agentes que se dedican a hurtar nuestros datos para su beneficio.

El hecho de cuidar nuestra información es muy importante, ya que en muchas ocasiones manejamos datos sensibles. Para salvaguardar todos estos datos se han implementado diferentes técnicas como lo es la criptografía que se encarga de cifrar la información utilizando claves públicas o privadas de manera tal que solamente quienes posean dichas llaves puedan acceder al mensaje. Otra técnica existente es la esteganografía, es la práctica de ocultar información dentro de algún objeto portador con el fin de que solo los destinatarios a quienes va dirigido el mensaje puedan acceder a él. Esta manera de resguardar información no es reciente ya que se ha aplicado desde hace miles de años, un ejemplo de ello es que en la antigüedad se afeitaba la cabeza de los esclavos y se tatuaban los mensajes que se querían transmitir en el cuero cabelludo, una vez que los esclavos tenían el cabello crecido eran enviados al receptor y de esta manera se mantenía resguarda a salvo la información. Dentro de los medios digitales podemos encontrar la aplicación de la esteganografía en archivos de texto, imágenes, videos, audios.

La esteganografía en audio consiste en incrustar el mensaje a transmitir dentro de la secuencia del archivo, existen diferentes técnicas para lograr el objetivo como lo son LSB, Cifrado y dispersión, Echo hiding, Codificación de paridad, Codificación de fase, etc. Sin embargo, en diversos estudios podemos encontrar que la esteganografía en audio cuenta con una alta complejidad para implementar dado que es muy fácil detectar algún cambio o ruido dentro del archivo portador lo cual puede levantar sospechas. A diferencia de las técnicas esteganográficas que se aplican en imágenes o videos, en las cuales la detección de cambios es más difícil a simple vista ya que el ojo humano es muy limitado en ciertas cuestiones.

CAPÍTULO 1
ESTEGANOGRAFÍA

1.1 INTRODUCCIÓN

El término esteganografía se compone de dos términos griegos *steganos* (oculto) y *graphos* (escritura), en términos informáticos, es el área de estudio que analiza el conjunto de técnicas cuyo objetivo es el encubrimiento de información sensible, mensajes u objetos, contenidos en el interior de otros denominados ficheros contenedores. En la actualidad se utilizan ficheros multimedia: archivos de audio, vídeos o imágenes digitales, teniendo como fin que la comunicación pueda transmitirse de manera inadvertida para terceros y únicamente pueda ser recuperada por un usuario legítimo.

Las técnicas esteganográficas han evolucionado acorde a la cambiante tecnología, y así por ejemplo durante la Segunda Guerra Mundial se utilizaban los periódicos para la transmisión de señales ocultas mediante la elaboración de marcas en ciertas regiones del texto, que, aunque por si solas éstas marcas pasaban desapercibidas, unidas trasmitían un mensaje.

En la actualidad, cuando hablamos de esteganografía nos referimos principalmente a ocultar mensajes en el entorno de medios digitales: audio digital, protocolos de comunicaciones, documentos de texto, archivos ejecutables, imágenes, etc. Habitualmente, el archivo contenedor es conocido pero lo que se ignora se ha utilizado un algoritmo o técnica de inserción para camuflar esta información.[1]

1.2 TÉCNICAS DE ESTEGANOGRAFÍA

A continuación, se enlistan algunas técnicas de esteganografía.

- **Enmascaramiento y filtrado (Watermarking)**

La información está oculta en el interior de una imagen digital utilizando marcas de agua. El propósito de la marca de agua digital es revelar el uso ilegal de un determinado recuso o servicio digital por parte de un usuario no autorizado. El proceso de este método es insertar un mensaje (un grupo de bits que contienen

información sobre el autor basada en los derechos de propiedad de este) en el interior de un objeto digital. Otra técnica relacionada es la huella digital, donde al mensaje se le añade no sólo información sobre el creado o dueño sino también se le agrega información del usuario que obtuvo los derechos para utilizar tal objeto. Así se puede seguir la pista a la distribución ilegal de medios o servicios digitales.

Para las imágenes digitales, debemos considerar algunos factores importantes a la hora de colocar una marca de agua.

- **Robustez**

Esta propiedad permanece en el archivo (la imagen) aun cuando la calidad de esta haya sido degradada intencionalmente o no.

Las cualidades que debe de tener un sistema para que sea robusto son:

La detección de la marca debería exigir el conocimiento de un secreto.

Varias señales no deben interferir entre sí.

Las marcas tienen que sobrevivir a los diferentes posibles ataques que no reduzcan la calidad percibida.

- **Invisibilidad**

El algoritmo debe asignar la marca de agua en el archivo de manera que ésta no altere a la calidad de la información que se encuentra contenida en ella. El proceso de inserción de la marca es completamente imperceptible si no la capacidad de distinguir la información original contra los a los datos modificados.

- **Cantidad de información**

Se puede insertar información en la imagen, para garantizar que se ha insertado una cantidad máxima de información se debe repartir por toda la imagen. En algunas aplicaciones este puede fungir como un factor determinante

- **Seguridad**

En los algoritmos de marcas de agua, la marca a insertar debe de ser secreta y únicamente debe ser accesible para las personas o partes autorizadas. Esta marca debe de persistir a las manipulaciones que se puedan producir de usuarios no autorizados. Los agentes que no sean propietarios de derechos legales para su uso no pueden ser capaces de detectar ni descifrar la información que se ha insertado en la imagen.

- **Recuperabilidad**

En los sistemas de detección ciega en el cual no tenemos acceso a la imagen original, la información que se ha introducido debe tener propiedades que permitan su recuperada posteriormente por los usuarios que posean el permiso para su empleo.

- **Rapidez de extracción**

Es de suma importancia que el proceso que se utiliza para la detección sea rápido para el titular o para el propietario original de los recursos digitales. Es un proceso principalmente importante en los sistemas en tiempo real.

- **Exclusión de ambigüedad**

Para aplicaciones de derecho de autor, la información incorporada debe estar explícitamente vinculada a la persona, empresa, agente u organización que posee estos derechos de la propiedad intelectual del fichero o a quien los adquiera.

Inserción de bits en el objeto contenedor

Se agregan los bloques de información tomando en cuenta determinadas marcas estructurales del archivo, estas pueden ser el alineamiento, el fin del archivo o los espacios. El inconveniente que presenta este método es que se aumenta considerablemente el tamaño del archivo contenedor, lo cual puede llevar a su fácil detección.

Un ejemplo de ello, dentro de un archivo con formato BMP, los primeros 54 bytes contienen los metadatos de la imagen. Cuatro de esos bytes se asignan a la distancia entre la cabecera y el primer píxel. La manera fácil de ocultar datos es realizar la inserción justo después de los metadatos, pero antes de los datos de la imagen. De esta manera, se dejará espacio para cualquier otro contenido que se desee agregar.[2]

1.3 CREACIÓN DE UN FICHERO CONTENEDOR PROPIO PARTIENDO DE LA INFORMACIÓN A OCULTAR (DCT)

El proceso general de este algoritmo se detalla a continuación. El primer paso consiste en dividir la imagen en subimágenes las cuales deben tener el mismo número de pixeles tanto en ancho como en alto. Seguido de ello, se define un determinado umbral, estas imágenes se dividen nuevamente si es que exceden el umbral antes definido. Este proceso se emplea tantas veces como sea necesario hasta que no exista actividad aplicable dentro de las subimágenes generadas. De esta manera se consigue un proceso iterativo y adaptativo, este método consigue resultados que tienen mayor efectividad en comparación con otros métodos. La técnica se puede aplicar a secuencias de imágenes.

Las técnicas adaptativas intentan aprovechar la estructura de las imágenes ya que son diferentes, para codificarlas de diferentes formas según las características con las que se cuenta, y de esta manera utilizar los bits de una forma más eficiente. El fin es realizar cambios únicamente donde sean necesarios.

Se debe crear un fichero contenedor que tenga en su interior la información que se desea ocultar. Por ejemplo, dado un algoritmo especial de reordenamiento de información que se desea ocultar se debe crear una secuencia de pixeles de un archivo con formato BMP que tenga algún significado visual.

En archivos de vídeo normalmente se utiliza el algoritmo DCT (Transformada de coseno discreta) que está basada en la DFT (Transformada de Fourier discreta), en el cual solo se utilizan números reales. DCT funciona realizando una ligera

modificación cada uno de los fotogramas del archivo, para que las modificaciones sean invisibles para el ojo humano.

La secuencia general para este método se detalla a continuación:

1. Primero se debe calcular el DCT de la imagen
2. Después, reemplazar los coeficientes que se encuentren por debajo que un cierto valor umbral con bits de la información que se desea ocultar.
3. Calcula la función inversa del DCT del archivo.
4. Salvar el archivo modificado.[3]

1.4 SPREAD SPECTRUM (ESPECTRO DISPERSO)

En la esteganografía de audio, el método de transmisión básica de Spread Spectrum (espectro ensanchado) difunde la información secreta a través del espectro de frecuencia creado por la señal de audio. Esto es similar a un sistema que utiliza una implementación de LSB donde se los bits de mensaje se distribuyen de manera aleatoria por todo el archivo de audio. Sin embargo, a diferencia de la codificación de LSB, en el método espectro ensanchado se transmite el mensaje oculto dentro el espectro de frecuencias del archivo de audio utilizando un código que no depende de la señal real. Como resultado, la señal transmitida ocupa un ancho de banda más grande de lo que realmente se utiliza para su propagación.

La tecnología de Espectro Disperso fue implementada para evitar que la transmisión de información militar y de inteligencia fueran interceptadas e interpretadas. Los mensajes enviados se distribuyen en diferentes frecuencias que emplean más espacio en el espectro para evitar que sea interceptada o interpretada.

Las señales del espectro son transmitidas en un ancho de banda grande puede convivir con señales de banda estrecha modificándola e insertando un pequeño aumento en el ruido de fondo para que los receptores de la banda estrecha puedan acceder a ella.

Una característica del método de espectro ensanchado es que puede crear una señal reversible de manera que se distribuya entre una banda donde existan frecuencias con ancho de banda mayor que a la que ocupaba originalmente.

Los datos se multiplican por un código de serie M conocido tanto como por el emisor como por quien recibe el mensaje, después de ello se insertan dentro del audio utilizado como portador. Por lo tanto, aun si el ruido afecta algún valor de la señal quedará alguna copia de cada valor para rescatar el mensaje insertado.

1.5 CODIFICACIÓN DE PARIDAD

La codificación de paridad es una poderosa técnica de esteganografía que se aplica en audio. Esta técnica no divide la señal en pequeñas porciones individuales si no que divide la señal en muestras separadas y cambia cada bit del mensaje a ocultar en un bit de paridad de la señal. Si el bit de paridad de una determinada región no coincide con el bit secreto para su codificación, se debe de reemplazar el bit menos significativo de una de las muestras en esa región. Finalmente, quien envía el mensaje tiene más opciones para insertar el mensaje secreto y la señal se puede cambiar de una manera más discreta. La figura 1.5.1 muestra el procedimiento de codificación de paridad.[4]

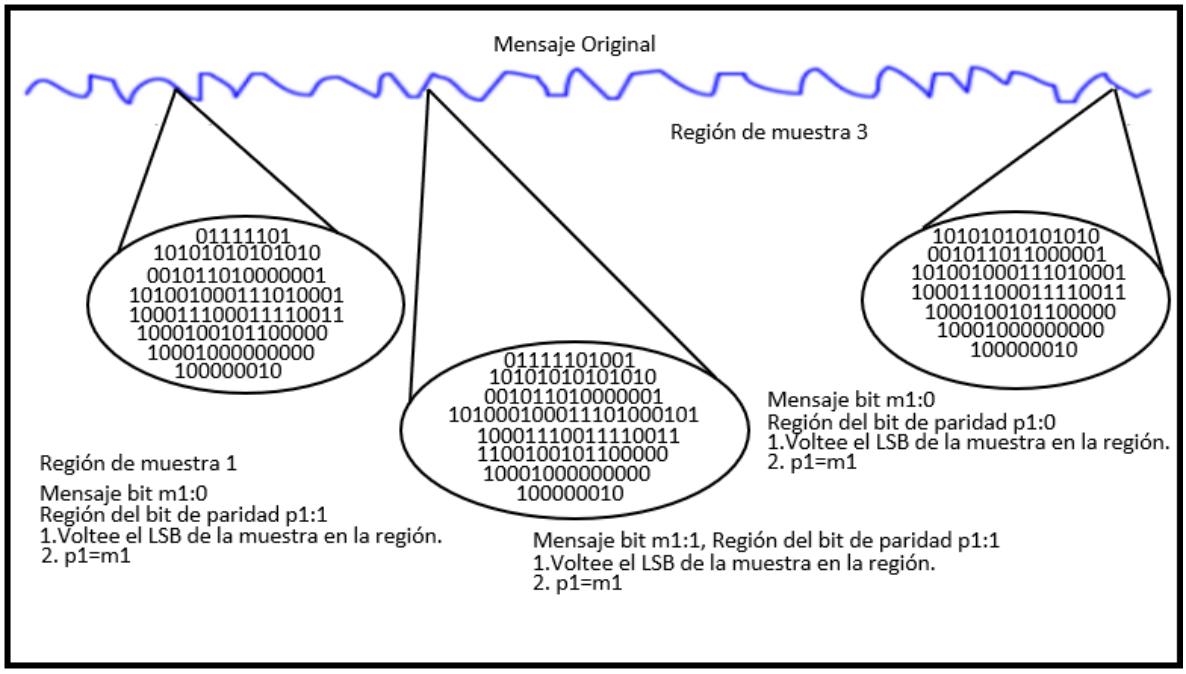

Figura 1.5.1 Procedimiento de codificación de paridad

1.6 CODIFICACIÓN DE FASE

La técnica de codificación de fase funciona reemplazando la fase en un segmento de audio inicial con un segmento de referencia que representa los datos. La secuencia de las siguientes secciones se ajusta para conservar la fase relativa entre las secciones.

Este algoritmo es considerado como uno de los métodos de codificación con mayor eficiencia en términos de la relación señal y ruido percibido. Si se produce una modificación drástica entre la fase y cada componente de la frecuencia se encontrará un cambio notable en la dispersión. Por el contrario, si se llega a una modificación de la fase sea lo suficientemente diminuta, se puede lograr una codificación imperceptible para el oído humano.

La codificación de fase se basa en la idea de que los componentes de fase del sonido son invisibles para el oído humano, por el contrario, el ruido tiene tendencia a ser detectado. En lugar de añadir interferencias, esta técnica codifica los bits del mensaje como cambios de fase en el espectro de fase de una señal digital, logrando una codificación imperceptible.

La codificación de fase se define en el procedimiento presentado:

a) La señal de audio original se divide en pequeños segmentos los cuales tienen una longitud de tamaño igual a la del mensaje que se va a codificar.
b) Aplique una transformada discreta de Fourier (DFT) a cada segmento para generar una matriz.
c) Se debe calcular las diferencias de fase entre los segmentos.
d) Al realizar cambios en segmentos de fase adyacentes se notarán fácilmente. En otras palabras, las fases absolutas de los segmentos pueden cambiarse, pero deben conservarse las diferencias de fase relativas entre segmentos adyacentes.
e) En este paso crea una nueva matriz de fase utilizando la nueva fase del primer segmento y las diferencias de fase originales.

f) Usando la nueva matriz de fase y la magnitud original matriz, la señal de sonido se reconstruye aplicando la DFT inversa y luego concatenar los segmentos de sonido de nuevo juntos.

El receptor puede usar el DFT para obtener las fases y extraer la información, ver figura 1.6.1.

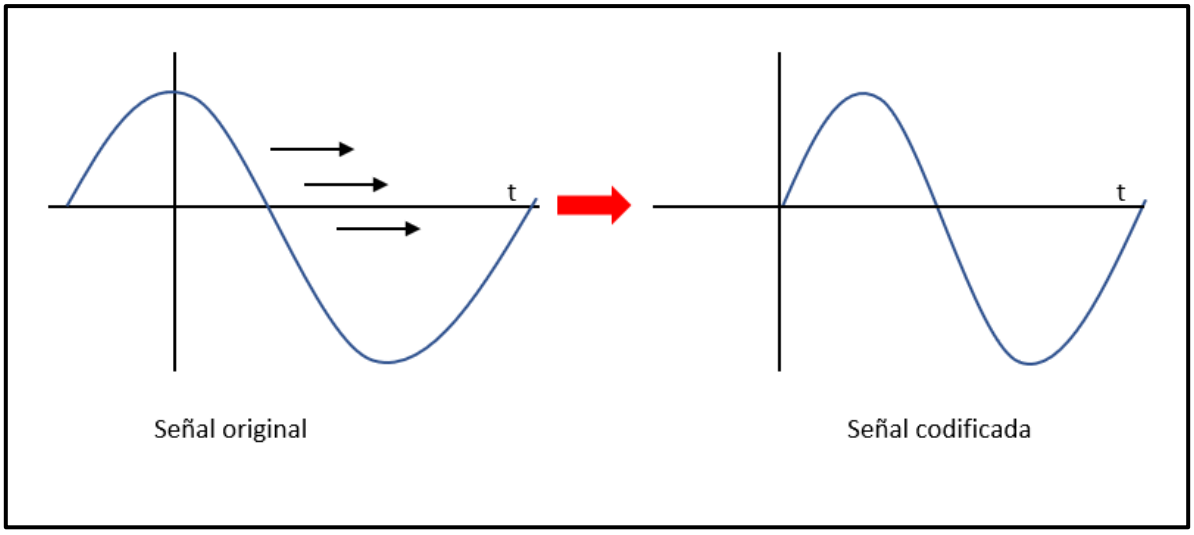

Figura 1.6.1 Ejemplo de codificación de fase utilizando DFT

1.7 Echo hiding

En el método de echo hiding, la información se inserta en un archivo de sonido al añadir un eco en la señal discreta. Ofrece ventajas como una alta velocidad de transmisión de datos y una alta solidez en comparación a otros métodos que agregan ruido a la señal; similar a el método del espectro ensanchado. Si únicamente se produce un eco a partir de la señal original, solo se podría codificar un bit del mensaje a ocultar. Por ello, la señal original se divide en bloques, antes de comenzar con el proceso de inserción. Una vez que se ha finalizado el proceso de codificación, los bloques se vuelven a concatenar para producir la señal final, ver 1.7.1.[5]

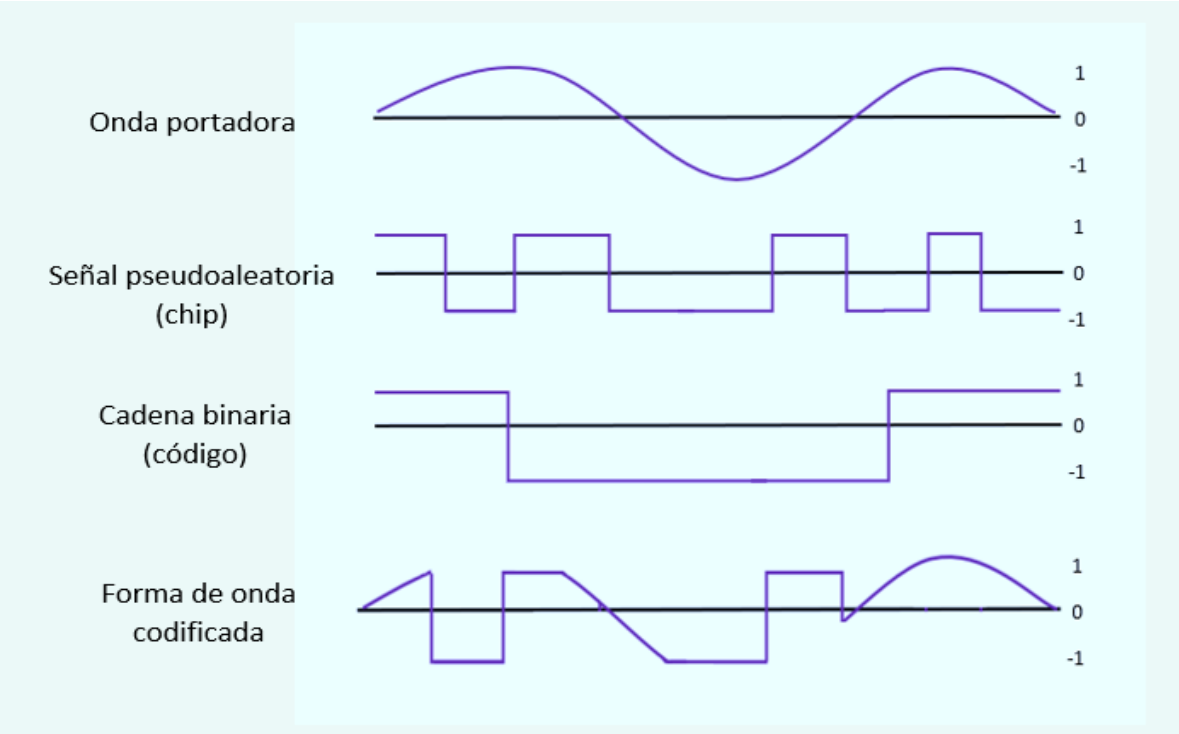

Figura 1.7.1 Proceso Echo Hiding

1.8 CIFRAR Y DISPERSIÓN

Las técnicas de encriptación y dispersión disfrazan el mensaje como White Noise Strom (ruido blanco), esta técnica utiliza el espectro ensanchado y el salto de frecuencia. El ancho de la ventana y el canal de datos anteriores se utilizan para generar un número aleatorio. Y con este número aleatorio, en los ocho canales, el mensaje se dispersa a lo largo de la señal. Cada canal se transforma, transmite, interactúa y se entrelaza con todos los demás canales. Un solo canal representa un bit y, como resultado, cada canal contiene muchos bits sin ser afectados. Esta técnica tiene como característica que el proceso de extracción del mensaje oculto del objeto contenedor tiene una alta complejidad. Este método es más eficiente y seguro comparado con LSB, porque requiere un algoritmo y una clave para decodificar el mensaje de bits del objeto contenedor. Algunos usuarios prefieren este método por razones de seguridad, ya que necesita tanto el algoritmo como la clave a pesar de la imagen.

1.9 CODIFICACIÓN DE PATRÓN REDUNDANTE

La codificación de patrón redundante es hasta cierto punto similar a la técnica de espectro ensanchado. En esta técnica, el mensaje se distribuye por todo el archivo dependiendo del algoritmo utilizado. Este método no funciona para recortar y rotar imágenes. El hecho de contar con múltiples imágenes más pequeñas con redundancia aumenta la posibilidad de recuperación incluso cuando se distorsiona la imagen original.

Histograma

El histograma de una imagen contiene la cantidad de píxeles con el mismo nivel de gris, el cual brinda información sobre el brillo y el contraste de la imagen, y puede utilizarse para ajustar estos parámetros y eliminar ciertas tonalidades molestas, etc.

1.10 MÉTODO ET (ENTROPY THRESHOLDING)

El método de selección del umbral de entropía (ET) utilizar la energía de cada bloque de 8x8 para decidir si se incorpora el mensaje secreto o no. Solo bloques que cuentan con energía mayor que el valor de umbral que se ha considerado anteriormente son utilizados para esconder el mensaje secreto.

A continuación, se describe el proceso de inserción utilizando la técnica de ET

1. Se debe realizar una modificación de histograma.

2. El archivo se segmenta en bloques de 8x8 píxeles.

3. Se aplica el método DCT a cada uno de los bloques generados.

4. La entropía de cada bloque se calcula de la siguiente manera:

$$E = \sum_{i,j} || \ C_{ij}||^2, \forall \ i,j \ \epsilon \{0,1, \dots ,7\}, (i,j) \neq 0$$

donde C_{ij} es (i, j)-ésimo coeficiente de un bloque. En el cálculo de entropía no se usa el componente DC de los coeficientes.

5. Se seleccionan de bloques cuya entropía es mayor que el valor umbral previamente determinado.

6. Cada bloque que se haya seleccionado se divide utilizando una matriz de cuantificación M basada en el factor de calidad de compresión.

$$\tilde{c}_{i,j} = \frac{c_{i,j}}{M_{i,j}^{fc}} \, , \forall \, i, j \, \{0,1, \dots ,7\}$$

7. Se realiza un escaneo a cada bloque seleccionado, para obtener un vector de longitud 64.

8. El mensaje secreto que se desea transmitir es insertado en los primeros ocho coeficientes utilizando el método del bit menos significativo (LSB). Se debe conservar el signo de todos los coeficientes.

9. El vector con el mensaje oculto se reordena en una matriz de tamaño 8x8.

10. Se multiplican los bloques seleccionados por la misma matriz de cuantificación, seguido de ello se le aplica el proceso inverso del método DCT para reensamblar la imagen, obteniendo así nuestro estenograma. Véase en la figurar 1.10.1.

El proceso de extracción en el método de ET El proceso de extracción escribe en siguientes pasos:

1. El estenograma es dividido en bloques de 8x8 píxeles.

2. Se aplica la DCT a cada bloque generado en el paso anterior.

3. Se calcula la entropía de todos los bloques para seleccionar los bloques con el mensaje oculto.

4. Cada bloque seleccionado es divido por la matriz de cuantificación M.

5. Se realiza un escaneo a cada uno de los bloques obteniendo un vector de longitud 64.

6. Se resguarda el bit menos significativo de los primeros 8 de cada bloque que se ha seleccionado.

7. La información que se ha obtenido en el paso anterior se une generar como resultado el mensaje que se ha insertado en el archivo.

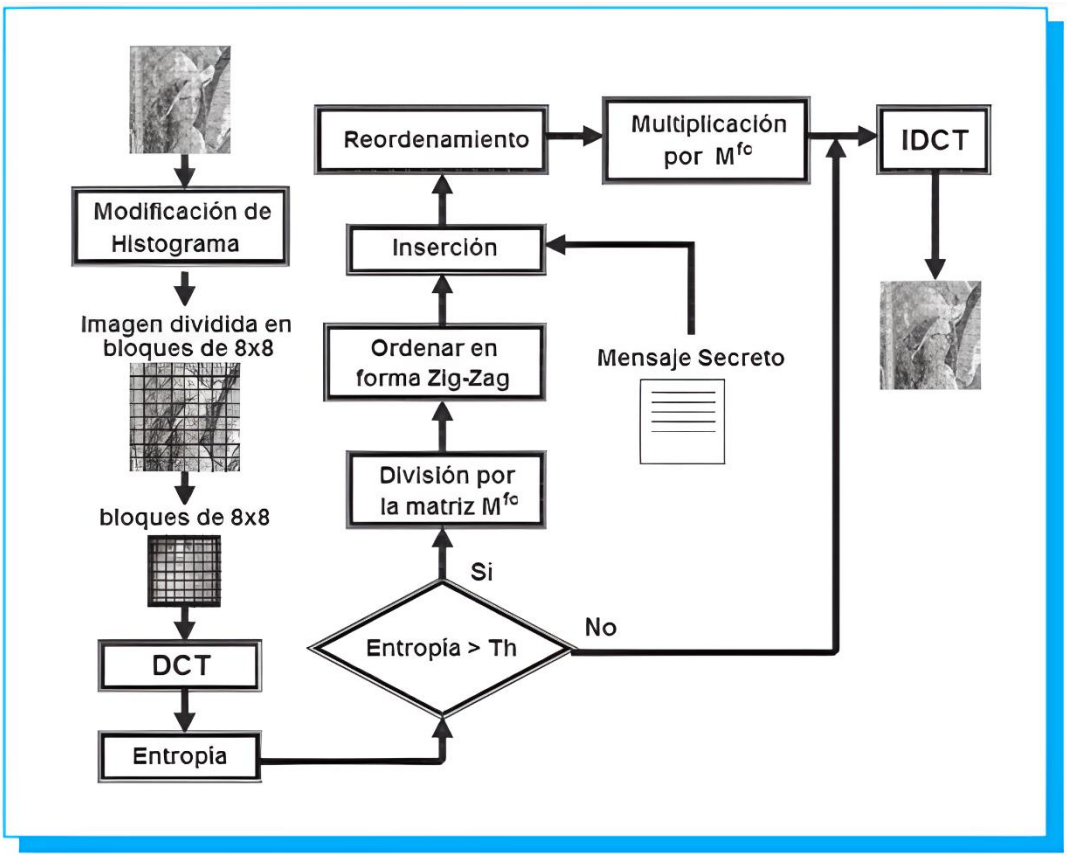

Figura 1.10.1 Proceso estenográfico utilizando el método ET

1.11 MÉTODO DE SEC (SELECTIVELY EMBEDDING IN COEFFICIENTS)

En el esquema de SEC, se realiza una selección de área de inserción para el mensaje de coeficiente por coeficiente para lograr una distorsión visual mínima. Las figuras muestran el proceso de inserción y extracción, respectivamente.

Proceso de inserción usando el método de SEC

1. Modificación de histograma.

2. La imagen se divide en bloques de 8x8 píxeles.

3. Se aplica la DCT a todos los bloques de la imagen.

4. Los coeficientes de DCT de cada bloque es dividido por la matriz de cuantificación M.

5. Se hace un escaneo en zigzag a cada bloque seleccionado, obteniendo un vector de longitud 64.

6. Se considera la banda de baja frecuencia para la inserción del mensaje secreto.

7. El valor absoluto de cada coeficiente c_{ij} es redondeado.

8. Si $r_k > t$, entonces al bit menos significativo del coeficiente c_k se le inserta un bit del mensaje secreto.

9. Se reordena el vector a una matriz de 8x8 para cada bloque.

10. A cada bloque seleccionado se multiplica por la misma matriz M y se aplica IDCT para obtener la imagen con el mensaje oculto.

Proceso de extracción en el método de SEC

1. El estenograma se divide en bloques de 8x8 píxeles.

2. Se aplica la DCT a todos los bloques de la imagen.

3. Los coeficientes de DCT de cada bloque son divididos por la matriz de cuantificación M.

4. Se hace un escaneo en zigzag a cada bloque seleccionado obteniendo un vector de longitud 64. 5. Se calcula r_k

5. Si $r_k > t$ entonces se extrae el bit menos significativo del r_k como bit del mensaje secreto.[6]

CAPÍTULO 2
Descripción del algoritmo propuesto

2.1 Objetivo general

Determinar qué tan eficiente es ocultar un texto dentro de un audio, utilizando una técnica esteganográfica basada en el algoritmo de spread Spectrum.

2.2 Objetivos específicos

- Realizar un análisis sobre el funcionamiento del algoritmo elegido.
- Proponer una técnica basada en el algoritmo anterior.
- Implementar la técnica esteganográfica propuesta.
- Realizar un análisis dentro de una determinada población para determinar si pueden detectar la presencia del texto oculto.

2.3 Spread Spectrum (espectro extendido)

2.3.1 Historia

Los primeros documentos acerca del espectro extendido se remontan a la época de la segunda guerra mundial (1939-1945); En aquella época la actriz Hedy Lamarr y el pianista George Antheil idearon un sistema de radiofrecuencias que permitía destruir los submarinos alemanes. En un principio la idea no era del todo funcional ya que se podía interferir el sistema y comprometer todas las operaciones; a Lamarr se le ocurrió que la frecuencia podría cambiar constantemente y de manera aleatoria para poder llevar a cabo las operaciones sin ser intervenidas.

2.3.2 Descripción general

Spread Spectrum utiliza las señales a transmitir, realiza la expansión de espectros utilizando secuencias ortogonales. Quien recibe la señal puede acceder a la información oculta sólo si conoce la secuencia que se ha utilizado para su modulación. El resto de los usuarios que utilicen el mismo canal de transmisión pueden ver la señal, pero no pueden acceder a su

contenido. Para poder recuperar la información que se encuentra en la señal debe existir una estricta sincronización entre las señales recibidas y el código generado por el emisor.[7]

La detección y decodificación de la señal puede presentar una alta complejidad ya que se crea una dispersión de la señal al realizarse el ensanchado, una de sus características de este proceso es que se conserva la energía.

Para realizar las operaciones de modulación y demodulación debemos tomar en cuenta una señal de pseudo ruido que se obtiene a partir de una secuencia pseudo aleatoria de bits.

La operación de ensanchado y desensanchado del espectro de la señal se realizan operando sobre una señal de pseudo ruido que se obtiene a partir de una secuencia pseudo aleatoria de bits. Estas secuencias tienen unas propiedades muy parecidas a las de una secuencia puramente aleatoria de bits, con la diferencia de que las primeras son periódicas y pueden ser reproducidas. Una secuencia pseudo aleatoria es, por lo tanto, una secuencia periódica de bits, con un período largo, dentro del cual sus propiedades son iguales a las de una secuencia aleatoria. Se deben de cumplir las siguientes características en un periodo de secuencia:

- Si se llegan a encontrar secuencias de ceros o unos repetidas, deben estar distribuidas.
- Se debe de considerar una relación entre el número de ceros y unos menor o igual que la unidad, que se encuentren en el periodo.

La señal generada por Spread Spectrum puede convivir con diferentes señales que se encuentran sobre el mismo canal, éstas señales cuentan con espectro estrecho, la señal del espectro ensanchado tiene un espectro de banda más estrecho, se añade un pequeño ruido de fondo lo cual garantiza que los demás usuarios que tengan acceso al canal no puedan detectar la señal producida. El receptor destinado a extraer la información de señal no

accede a las señales estrechas que fluyen por el canal ya que él monitorea un ancho de banda más amplio.

La idea de este método se basa en que la señal de audio debe contar con un proceso reversible para la extracción de la información. La implementación de este algoritmo cuenta con las siguientes características:

- Debemos tomar en cuenta un valor para la velocidad de la transmisión de, éste se asigna a la variable R, al ancho de banda que se utiliza por la señal de transmisión lo asignamos a la variable W. Se tiene que cumplir que el ancho de banda dividido por la velocidad de transmisión debe ser mayor que 1 (W/R>1).
- La señal que se envía solamente se puede demodular si es que se utiliza la misma señal con la que fue creada, esta señal tiene apariencia de ruido, pero no es así ya que ha sido generada por una secuencia pseudoaleatoria.

Otros aspectos interesantes de la señal que se genera después del proceso de ensanchamiento se presentan a continuación:

- Estas señales cuentan con una mayor resistencia a interferencias que otros tipos de transmisión.
- Es ampliamente difícil detectar la señal y la recuperación de la información oculta se puede hacer únicamente después de la demodulación.
- Si se llegase a capturar la señal en el canal de transmisión, la secuencia pseudoaleatoria utilizada impedirá que se extraiga la información.

2.3.3 Técnica de secuencia directa

El algoritmo de la técnica de secuencia directa es relativamente sencillo de implementar y es uno de los métodos más frecuentes, es también conocido como Acceso Múltiple por División de Código en Secuencia Directa (DS-CDMA).

El DSSS depende de la tasa de bits de la secuencia pseudoaleatoria por bit de información. En un sistema DSSS, una señal de bajo ancho de banda se distribuye en un amplio rango de frecuencias. Por lo tanto, la potencia de la señal disminuye y, por lo tanto, la señal se desvanece en el ruido de los medios de cobertura. Para extraer una señal incrustada de la cubierta, el receptor necesita conocer el proceso de propagación. [8]

Las características principales de DS-CDMA son:

- La ampliación de la señal para su transmisión.
- La disminución de ruido de banda estrecha, como consecuencia del efecto de ensanchado.
- Las frecuencias de espectro ampliado son de 900 MHz y 2.412 a 2.484 GHz.
- La división del espectro de frecuencia en varios canales de 1.25 MHz aproximadamente, aunque esto puede variar según las regulaciones de cada país.

El espectro ensanchado aprovecha el desvanecimiento multitrayecto para fortalecer la señal original (El desvanecimiento multitrayecto se produce cuando una terminal móvil en conexión radio con una estación base, recibe, no sólo la señal proveniente del trayecto más corto entre ambos, sino también las de otras provenientes de rebotes con los diferentes objetos circundantes).

La técnica de acceso múltiple por división de código en secuencia directa combina la modulación DSSS con la técnica de multiplexado CDMA, para que las diferentes señales que acceden al medio compartan tiempo y frecuencia, diferenciándose

entre sí mediante asignación de códigos binarios ortogonales.

Las características principales de esta técnica son:

- La disminución de ruido.
- La ampliación de la señal para transmisión.

- Las frecuencias de espectro ampliado están entre 2.412 y 2.484 GHz.
- La división del espectro de frecuencia en varios canales de 1.25 MHz.
- Protección de la señal ante el desvanecimiento multitrayecto. [9]

2.4 Técnica propuesta detallada

El principal objetivo del algoritmo Spread Spectrum es transmitir un mensaje oculto en una señal de ruido, este mensaje se envía disperso por partes utilizando una secuencia aleatoria, sin embargo, la transmisión no es del todo aleatoria ya que tanto el emisor como el receptor deben estar sincronizados para poder recuperar el mensaje. La propuesta de implementación está basada directamente en esta idea, el proceso de ocultación y extracción se divide en una serie de pequeños pasos.

2.4.1 Ocultación

Se tienen dos requerimientos necesarios para el proceso de ocultación, el primero es tener preparado un audio en el cual se incrustará el texto y el segundo es el texto por ocultar que deberá ingresar el usuario.

1. Agregar un carácter especial al texto dado por el usuario.
2. Convertir el texto en su representación binaria.
3. Convertir el audio a su representación binaria.
4. Hacer un ciclo sobre la representación binaria del audio para cambiar los bits por los bits del texto.
5. Finalmente se guarda obtiene el nuevo audio modificado y se guarda en un nuevo archivo, ver figura 2.4.1.1.

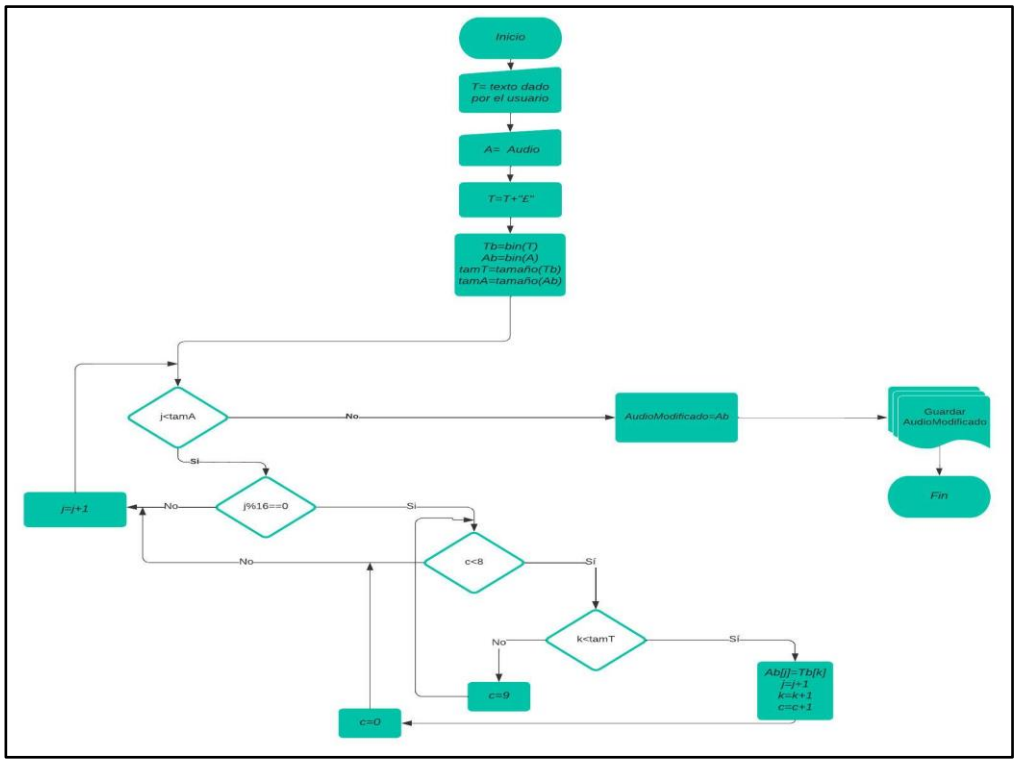

Figura 2.4.1.1 Diagrama para ocultar

El primer paso del proceso es sumamente sencillo, simplemente es agregar un carácter especial el final del texto que se desea ocultar, la razón de ello es que al extraer el mensaje tendremos una condición de paro y de esta manera solo se mostrará el mensaje oculto. Supongamos lo siguiente.

> *El texto que se desea ocultar es "Hola"; después del primer paso, el texto queda de la siguiente manera:* ***Hola£***

En el segundo paso tenemos como objetivo convertir el mensaje a su representación binaria para poder modificar el audio (objeto portador). Seguimos con el ejemplo anterior.

> *Al llevar el texto anterior a su forma binaria, obtenemos lo siguiente:* ***01001000 01101111 01101100 01100001 10100011***

Para poder modificar el audio con los bits del texto, debemos también obtener los bits de éste.

El cuarto paso es uno de los más importantes, ya que aquí ocurre la modificación; como la técnica está basada en el algoritmo de espectro ensanchado, se insertarán 8 bits del texto a ocultar cada 16 bits del audio. Es una forma de modificación aleatoria, pero debemos recordar que no es del todo aleatoria para poder recuperar el mensaje.

El paso final es guardar el audio modificado, para saber cuál es el nuevo audio tendrá un nombre diferente cada vez que se utilice el sistema. Este tendrá por nombre la fecha y hora en que se guarda el archivo.

2.4.2 Extracción

El proceso de extracción es más sencillo que el proceso de ocultación; aquí solamente necesitamos un recurso para obtener el mensaje oculto que es el audio modificado previamente. A continuación, se enlistan los pasos a seguir.

1. Llevar el audio a su representación binaria.
2. Realizar un proceso iterativo y agregar los bits del mensaje a una lista.
3. Convertir los datos de la lista a su representación ASCII.
4. Eliminar los caracteres sobrantes, véase en la figura 2.4.2.1.

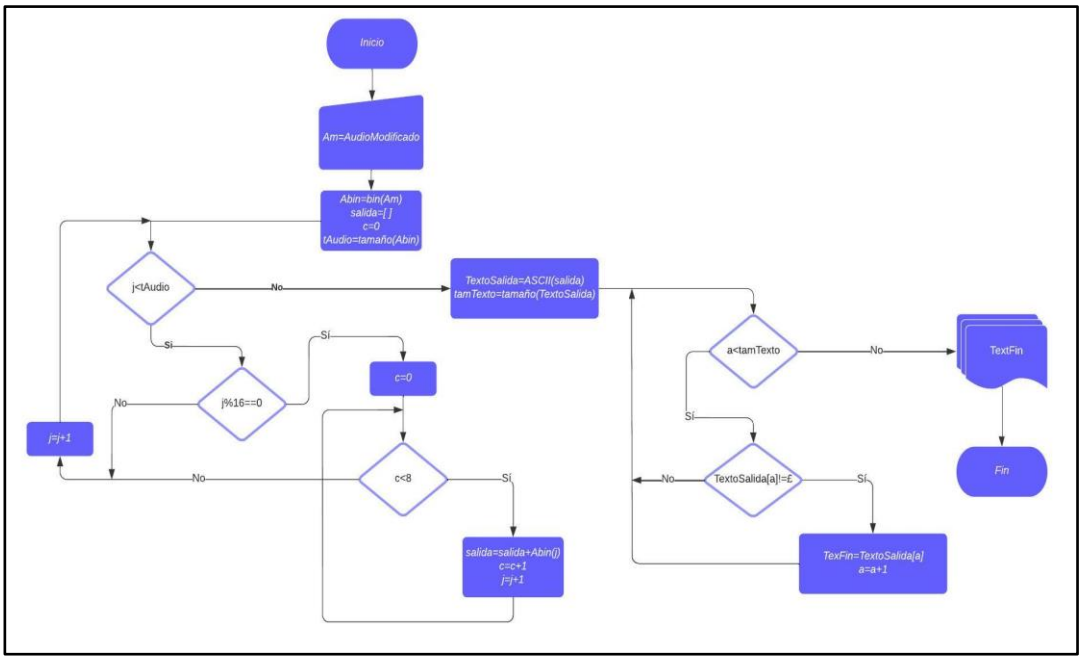

Figura 2.4.2.1 Diagrama para extraer

Para obtener el mensaje oculto, el primer paso es llevar el audio previamente modificado a su correspondiente binario.

Ahora se tiene que recorrer esta representación para tomar los bits que se insertaron en el proceso anterior. Como se explicó anteriormente los bits del texto están distribuidos en el audio.

Una vez que se han obtenido todos los bits podemos llevar esta información nuevamente a su correspondiente ASCII, pero aún no está totalmente depurado el mensaje, para ello tenemos el paso siguiente.

Uno de los pasos del proceso de inserción fue agregar un carácter de terminación, el cual se ocupa en este paso; se deben eliminar los caracteres que están a partir de punto y de esta manera nos aseguramos de que conservamos el mensaje original.

2.5 Interacción con el sistema

El sistema propuesto es bastante sencillo de utilizar y entender, ya que solamente se necesitan muy pocos elementos e interactúan dos usuarios que pueden estar

cambiando de rol; estos usuarios los definimos como “emisor” y “receptor”. El emisor es quien envía el audio modificado y el receptor es quien extrae el texto del audio.

Como se explica anteriormente, existen dos principales procesos o recursos que necesita cada uno de los usuarios para poder interactuar con el sistema; sin embargo, existen algunos otros pequeños procesos que no son visibles para el usuario, pero dependen unos de otros para el correcto funcionamiento del sistema véase la figura 2.5.1.

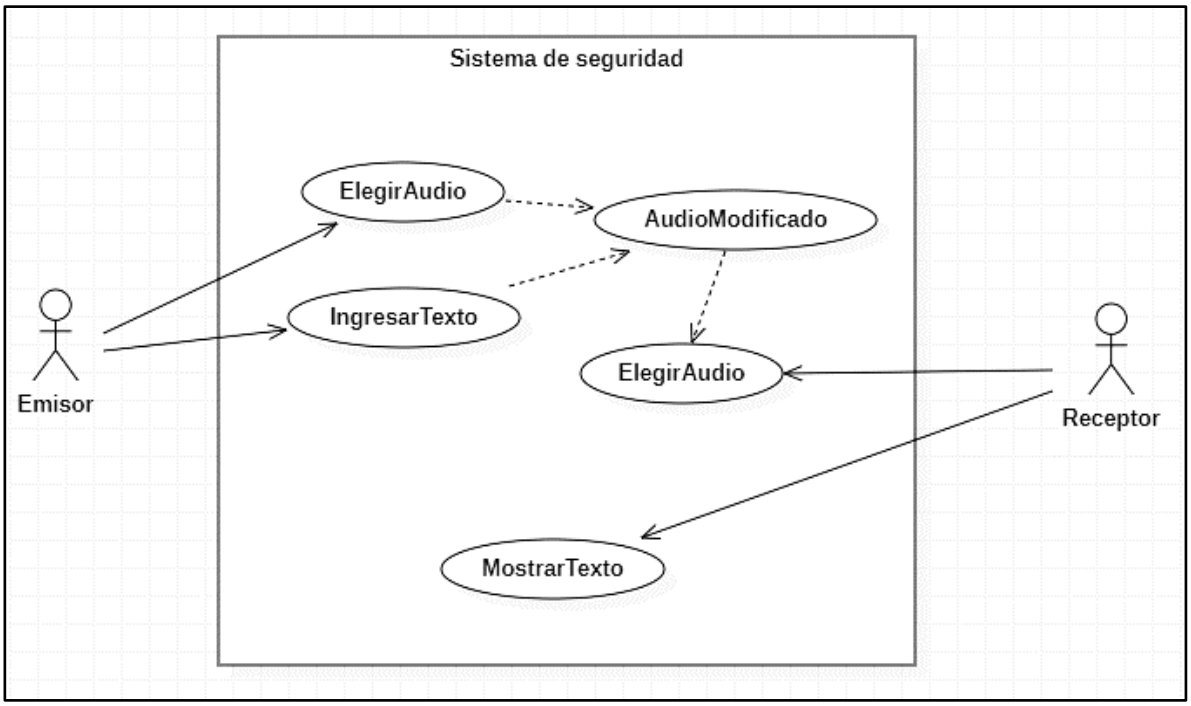

Figura 2.5.1 Funcionamiento del sistema

Como podemos observar en el diagrama anterior, los dos usuarios (emisor y receptor) interactúan con dos recursos; por parte del *Emisor* debe elegir un audio en el cual se guardará el que texto que él mismo ingresa, seguido de ello se genera el *Audio Modificado*. Por otro lado, el *Receptor* también elige un audio para mostrar el mensaje oculto, pero se necesita el audio previamente modificado.

Los usuarios también pueden cambiar de rol, es decir, que el receptor también puede ocultar un texto en un nuevo audio y de esta manera pasará de ser el receptor a ser el emisor.

CAPÍTULO 3
Interacción con el sistema

3.1 Funciones implementadas

El sistema propuesto cuenta con siete funciones que se dividen en dos categorías, funciones necesarias y funciones complementarias

3.1.1 Funciones necesarias

Estas funciones podemos definirlas como vitales para el correcto funcionamiento del sistema; entre ellas se encuentran tres funciones (oculta, leer y la función principal).

3.1.1.1 Oculta

Como su nombre lo indica, esta función sirve para ocultar el texto dentro del audio, primero se recibe el nombre del archivo además del texto; sabemos que se tiene que hacer un preprocesamiento que se ha explicado anteriormente. Después del preprocesamiento obtenemos el correspondiente binario del audio y del texto para poder aplicar la técnica propuesta y finalmente guardar el resultado.

3.1.1.2 Leer

La función tiene como objetivo leer el audio modificado para extraer la información, debe tomar el nombre del audio modificado, es muy importante seleccionar el archivo correcto ya que si se introduce un archivo distinto no se mostrará nada como salida.

Después de recopilar todos los datos y obtener nuevamente el texto oculto, se mostrará esta información en la interfaz gráfica.

3.1.1.3 función principal

Dentro de la función principal definimos todos los elementos necesarios para la ventana, también se llaman a las demás funciones ya sea de manera directa o indirectamente; esto es porque alguna de las funciones llama a otra función al momento de ejecutarse.

3.1.2 Funciones complementarias

Las funciones complementarias solamente proporcionan estética, estas fueron pensadas para mayor comodidad del usuario; por ejemplo, se contempla un menú para elegir el audio en el cual se guarda o se extrae la información y de esta manera no tener que escribir la ruta completa del archivo.

3.1.2.1 Agregar menú

La función menú es parte de la ventana, esta crea un apartado que permite al usuario navegar por el sistema para seleccionar el archivo deseado, esta sub-ventana se abre inicialmente desde el disco C que habitualmente se encuentra instalado ya que si se especifica una carpeta y no existe sería un error del sistema, por ello se trata de estandarizar y usar el disco C, aunque existe la posibilidad de que el disco no este nombrado así.

El usuario puede elegir que acción le interesa “seleccionar un audio para ocultar información o seleccionar un audio para extraer el texto oculto”.

3.1.2.2 Abrir

La función antes detallada solamente crea el menú con las opciones, pero para poder recuperar la ruta total del archivo fue creada esta función, al seleccionar un archivo se guarda la ruta en una nueva variable para mostrarla en la ventana y de esta manera el usuario puede saber que audio eligió, además esta variable se devuelve para poder trabajar con la ruta.

Existe otra función bastante parecida a esta, la diferencia es que selecciona archivos para poder mostrar la información oculta en lugar de guardar el texto.

3.1.2.3 Default

El objetivo de estas líneas de código es dar mayor estética a la interfaz, por ello no se ocupan etiquetas para indicar al usuario que es lo que debe de hacer, si no, que dentro de la misma caja de entrada de datos se escribe un letrero indicando la acción a realizar o lo que se espera obtener.

Este letrero aparece y desaparece cuando el usuario comienza a escribir sobre la caja de texto, tratando de replicar la función de un placeholder en HTML, la cual da un consejo o indicación al usuario en los elementos input.

3.2 Elementos de la interfaz

En la interfaz del sistema se toma en cuenta el ancho y alto de la ventana, además el tipo de fuente que se utilizará (Roboto Condensed) ya que es fácil de leer, también se utiliza una imagen de fondo, un icono en la parte superior izquierda de la ventana y el menú ya mencionado; existen elementos con los cuales el usuario puede interactuar, como son: dos botones que servirán para ocultar o mostrar el texto; tres cajas de texto tipo entry, en ellas el usuario podrá ingresar la información, ver el archivo elegido o visualizar el texto recuperado.

3.3 Interfaz propuesta

Ahora que se tiene una propuesta de interfaz abstracta, y se puede proponer un diseño contemplando todo lo que se menciona en la sección anterior. El ancho de la ventana es de 900*460 pixeles, esta toma el tamaño del ordenador donde se esté ejecutando para posicionarla a la mitad de la pantalla, en la parte superior de la pantalla se coloca el menú desplegable, este solamente cuenta con tres opciones, seguido de ello y es el elemento con mayor espacio dentro de la ventana se encuentra un label para indicar el archivo que elige el usuario.

En la parte inferior de la ventana se encuentran dos cajas de texto y dos botones, primero se encuentra el área donde el usuario ingresa la información y el botón que acciona el proceso para ocultar dicha información; al final está la caja que mostrará

el texto después de utilizar el botón para mostrar el contenido del audio ver figura 3.3.1.

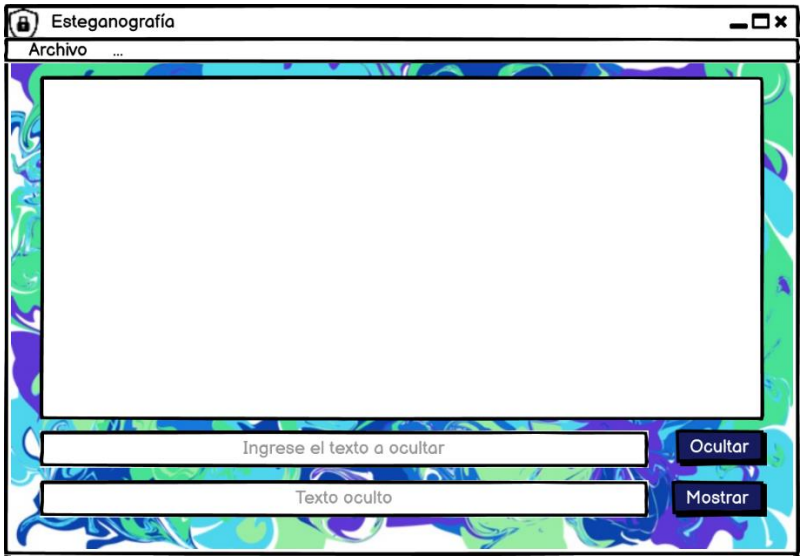

Figura 3.3.1 Interfaz propuesta

3.4 Ejemplo de uso

Al iniciar el sistema la primera pantalla que se muestra está totalmente en “blanco”, podemos ver los botones, las cajas de textos y demás elementos, ver figura 3.4.1.

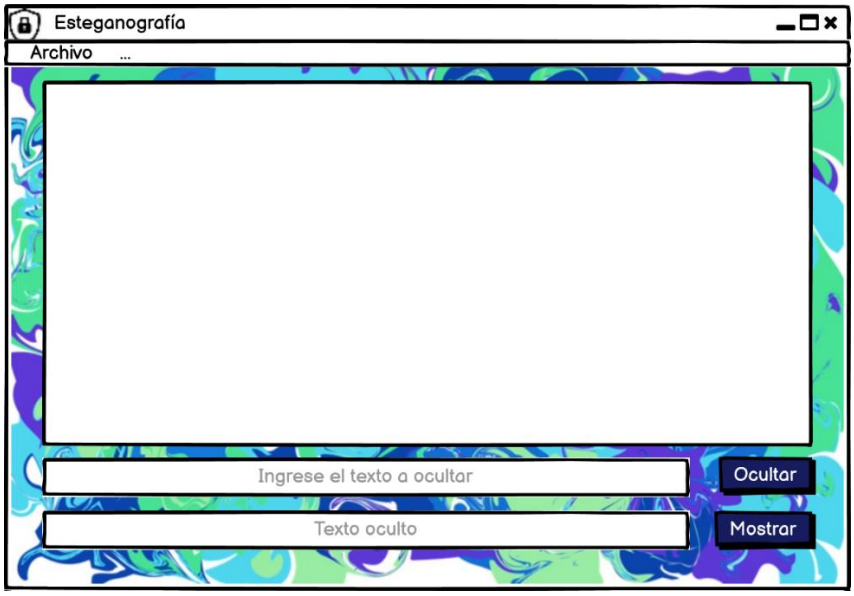

Figura 3.4.1 pantalla inicial

Lo primero que se debe hacer es elegir un audio para ocultar la información, para ello el usuario debe dirigirse al menú desplegable que aparece en la parte superior izquierda, figura 3.4.2.

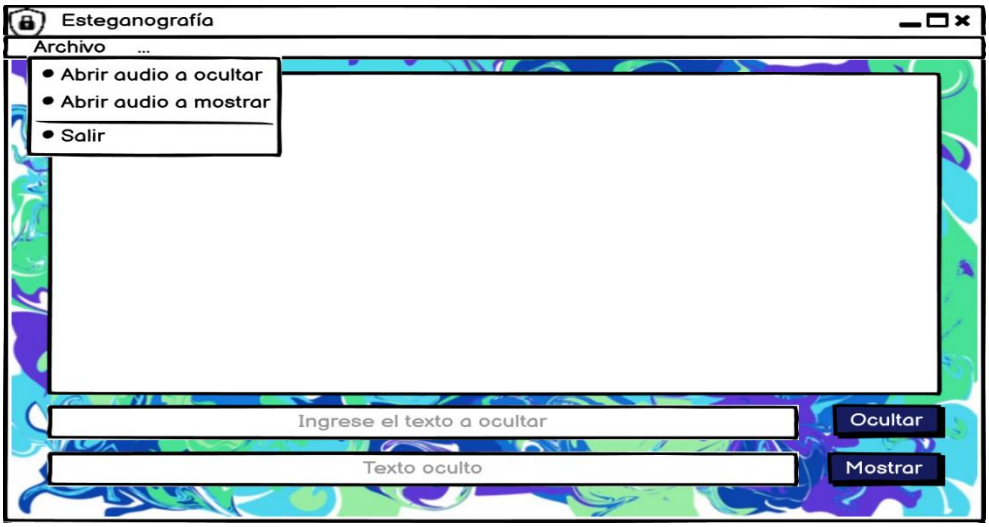

Figura 3.4.2 Menú desplegable

Ahora se muestra una ventana de navegación por los archivos del sistema operativo, permite el usuario elegir el archivo necesario para el proceso de ocultamiento, ver figura 3.4.3.

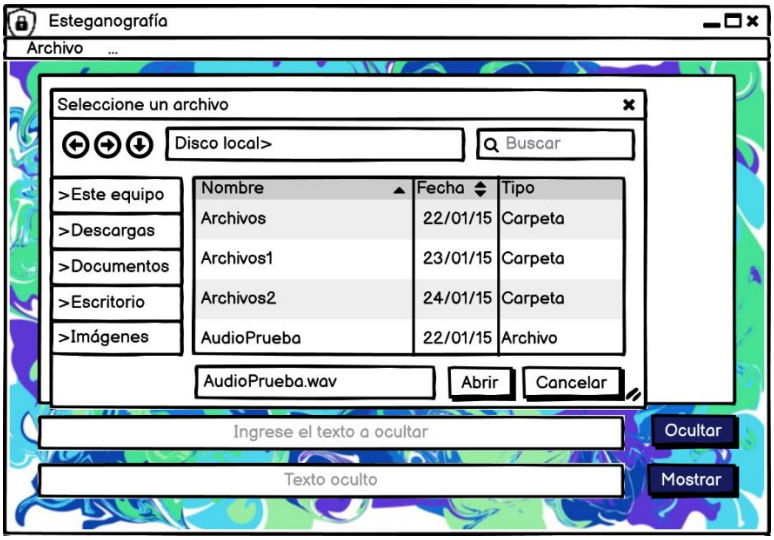

Figura 3.4.3 Ventana de navegación

Para que el usuario pueda verificar el archivo que eligió, se muestra la ruta y nombre de éste; además de un pequeño icono. La carga del audio habrá sido exitosa, ver figura 3.4.4.

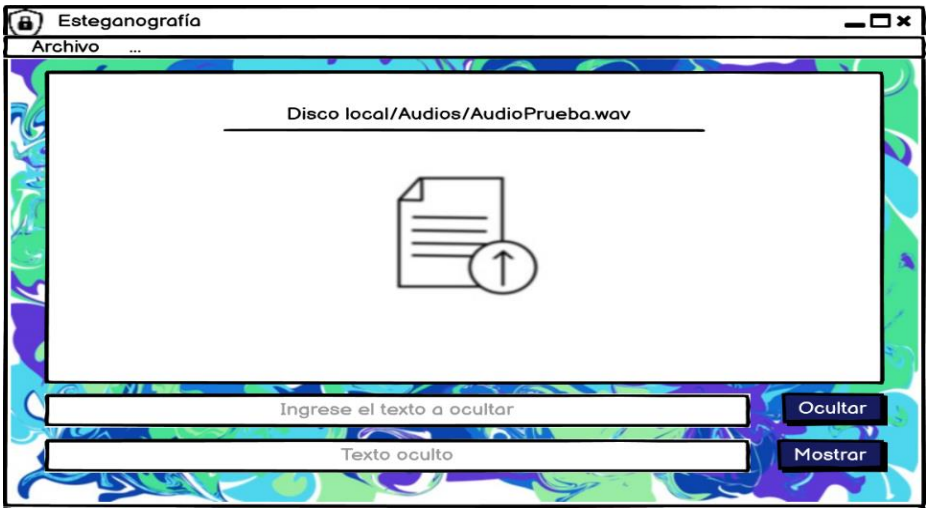

Figura 3.4.4 Carga de archivo

En el siguiente paso el usuario debe ingresar el texto que desea a ocultar, esto se ingresa mediante la caja de texto que indica tal acción; finalmente se utiliza el botón “ocultar” para iniciar el proceso, este proceso es algo tardado dependiendo del tamaño del texto dado por el usuario (también se debe de tomar en cuenta que la longitud de este debe ser menor al tamaño del audio)., ver figura 3.4.5.

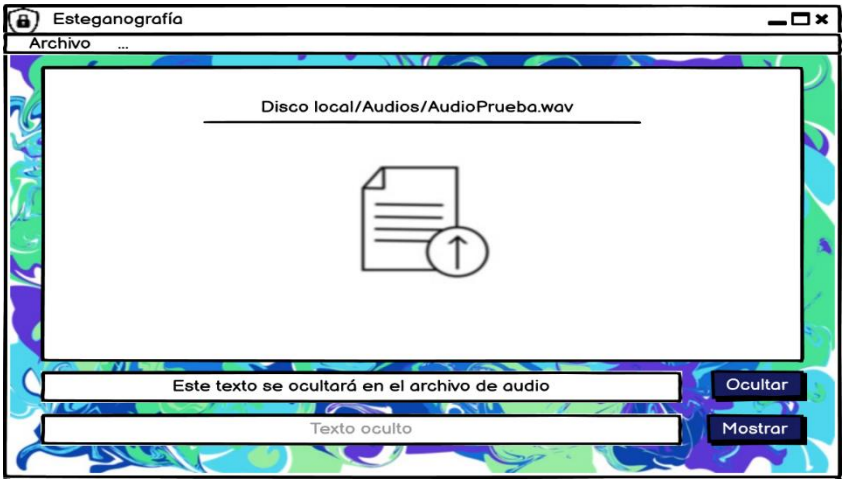

Figura 3.4.5 Texto a ocultar

Una vez terminado el proceso, se muestra un letrero que indica el nombre del archivo generado, el nombre del archivo de audio de salida se compone de dos partes, primero lleva el escrito “AudioSalida” aunado con la fecha y hora que retorna la función datetime.datetime.now() con extensión .wav, un ejemplo de nombre de salida sería “AudioSalida20220831212628611646.wav”; este se guarda en la carpeta desde donde se ha ejecutado el archivo .py (correspondiente a python) o el archivo .exe. Como podemos ver en la figura 3.4.6.

Figura 3.4.6 Archivo de salida

Ahora para recuperar el texto oculto tenemos dos opciones, seguir ocupando la misma ventana o reabrirla, en este caso, pensando en que otra persona es quien desea recuperar el texto, se mostrará la ventana de inicio, como lo vemos en la figura 3.4.7.

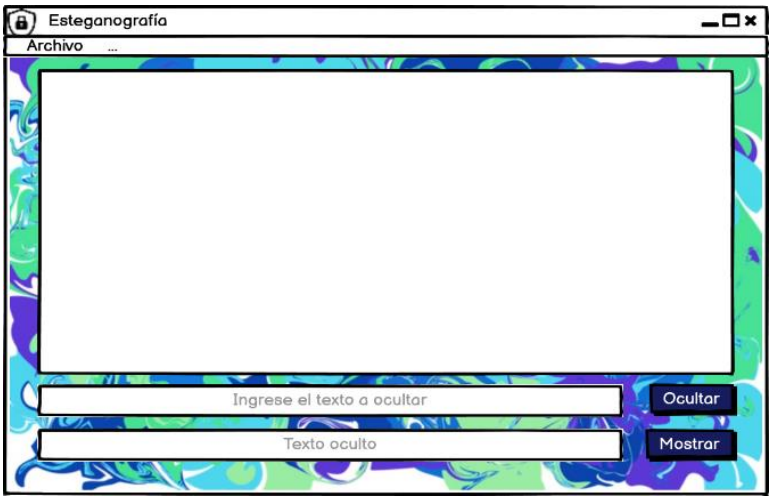

Figura 3.4.7 Pantalla de inicio para decodificar

El proceso para recuperar el texto es muy parecido al que se sigue cuando se desea ocular, se debe abrir el menú que se encuentra en la esquina de la ventana, pero en este caso el usuario tiene que elegir la opción “Abrir audio a mostrar”, ver figura 3.4.8.

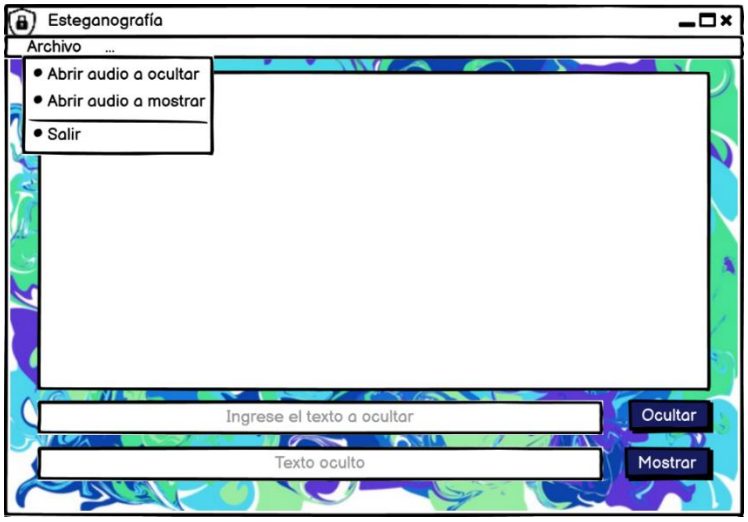

Figura 3.4.8 Menú para elegir archivo

Una vez abierta la nueva ventana, podremos navegar por los archivos del sistema operativo y seleccionar el audio correcto, esto podría ser algo difícil ya que el nombre del archivo generado anteriormente es bastante largo, ver figura 3.4.9.

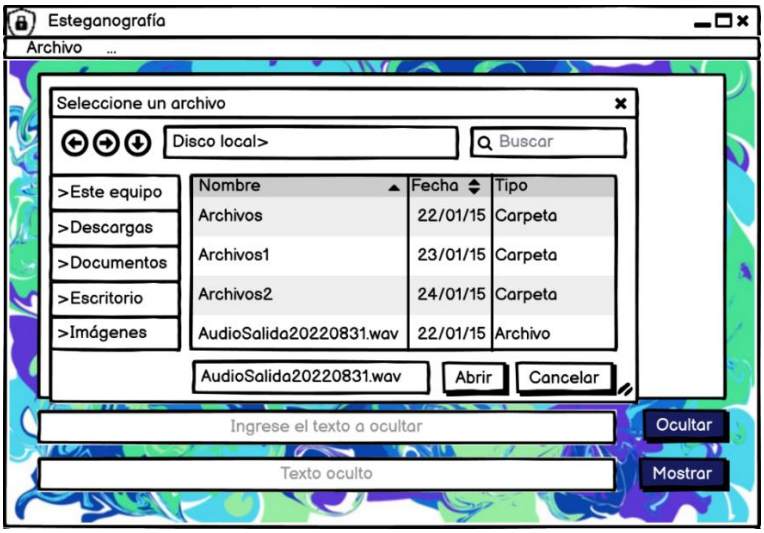

Figura 3.4.9. Selección de audio de salida

Ahora que el usuario seleccionó un archivo y se cargó en el sistema, volvemos a ver la ruta exacta en la que se encuentra. El usuario debe elegir el audio correcto que contenga el mensaje oculto, esto se debe a que al procesar la información se busca el carácter especial que se usa como condición de paro, si no se encuentra este carácter, el sistema marcaría un error, ver figura 3.4.10.

Figura 3.4.10 Carga de archivo de salida

El último paso para ver el mensaje oculto es simplemente accionar el botón “Mostrar” que se encuentra en la esquina inferior derecha de la ventana, la única condición para ello es tener cargado el audio en el sistema, cuando se haya completado todo, el escrito que contiene la caja de texto será remplazado por la información extraída, como se ve en la figura 3.4.11.

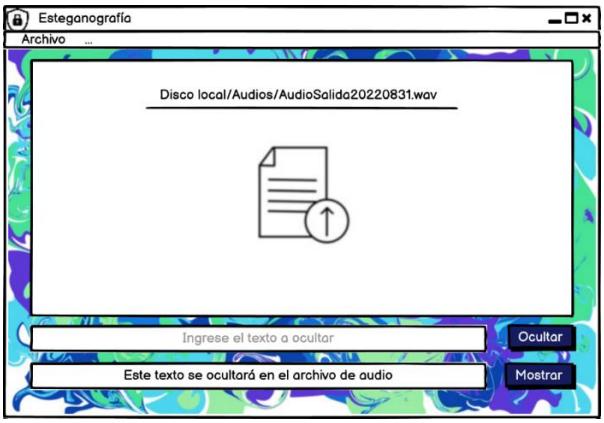

Figura 3.4.11 Texto oculto

CAPÍTULO 4
Resultados y conclusiones

4.1 Logros de la esteganografía

El oído tiene la capacidad de diferenciar cambios en casi cualquier ambiente, aunque sean muy pequeños y que, a diferencia del ojo, siempre están activos. “A diferencia de los ojos, que podemos enfocar, dirigir y cerrar, los oídos permanecen abiertos todo el tiempo, recibiendo todos los sonidos que nos rodean de una manera aparentemente pasiva, Pero en realidad el cerebro procesa la información que le proporciona el oído de una manera muy activa. Al percibir un sonido, nuestro sentido del oído compara, calcula y relaciona alturas, volúmenes y timbres para dar significado a lo que escuchamos y poder interpretarlo correctamente, ya sea lenguaje hablado, sonidos ambientales o música. El oído se las arregla para distinguir con claridad sonidos que son muy cortos. Las diferencias entre la L, M y la N son sutilísimas variaciones en intensidad y timbre, y sin embargo podemos distinguirlas con precisión”. A Partir de todo se sabe que, a diferencia de la esteganografía en imágenes digitales, la esteganografía en audios tiene mayor tendencia a notarse una diferencia entre los archivos originales y modificados.

Para determinar el nivel de seguridad, se propone el siguiente estudio. Se desea encuestar a 100 personas mayores de quince años, ellos deben elegir una canción con la cual estén familiarizados no importa el género, idioma o duración del audio. Una vez que se ha elegido una melodía se procesará utilizando el sistema esteganográfico y se ocultará un mensaje personalizado para cada uno de los encuestados. Seguido de ello se pide que escuchen el nuevo audio (modificado, que contiene el mensaje oculto) y que contesten un formulario; ellos pueden volver a escuchar tanto la canción original como la modificada. Al final de toda la prueba se explica que se hizo durante todo el proceso.

El estudio tiene como objetivo saber si los encuestados encontraron alguna modificación dentro de las melodías, modificaciones muy marcadas como lo son el ruido o una distorsión. De esta manera se obtiene una cifra que nos ayuda a determinar el nivel de seguridad.

Se propone también hacer un segundo estudio, para ello se comparan los espectros del audio original y el audio de salida, verificando si existe un cambio muy notorio entre los espectros a simple vista. Se utilizan las aplicaciones de Spek – Acoustic Spectrum Analyser desarrollada por Alexander Kojevnikov y Audacity creada por Dominic Mazzoni y Roger Dannenberg

4.2 Resultados de estudio

4.2.1 Resultados de la encuesta

Como se menciona anteriormente, se hizo un estudio dentro de una población de 100 personas, a las cuales se les pidió elegir una melodía con la cual estuvieran familiarizados. El audio se inserta en el sistema, se genera la salida con el mensaje oculto y se pide a los participantes escuchar los dos audios para finalmente contestar una encuesta.

Existen dos preguntas dentro de la encuesta que cuentan con un mayor peso, en estas se pregunta si notan alguna diferencia entre los audios y si existe se pide que den una breve descripción de ella. Dentro de la encuesta, el 7% de los participantes afirma haber encontrado alguna diferencia entre los dos audios.

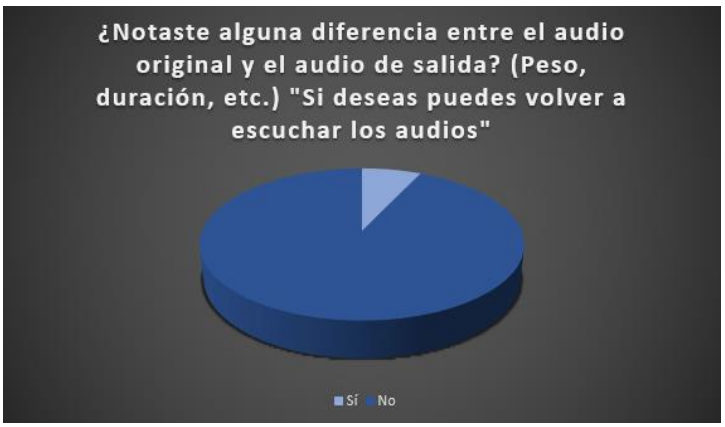

Figura 4.2.1 grafica de la pregunta eje.

También se pidió a los encuestados que dieran una breve descripción de los cambios notados para saber las diferencias que encuentran, se busca que no haya cambios como el ruido o distorsión. Dentro de las 7 personas que

respondieron que si había cambios podemos encontrar las siguientes descripciones.

> *Noto un mínimo de ruido, dentro de la canción. No es notable la primera vez que se escucha, sin embargo, al detenerse a escucharla con mayor atención se puede notar algo.*
>
> *Menor duración y se siente más inmersiva*
>
> *Una pequeña disminución en la calidad de sonido*
>
> *Creo que el segundo audio se escuchaba más lento como un poco pesado y aparte el sonido de la música en el principio se escuchaba un poco de eco se nota muy poco, pero si, aunque escuche un poco más fuerte la música que en el primero me refiero a un poco más cerca.*
>
> *Como que sentí que se tomaba más tiempo para pronunciar las palabras en ciertas partes de la canción, que en el centro de la canción escuché diferente los tonos de los instrumentos.*
>
> *Algo más lenta*

El restante 93% de las personas ha respondido que no hubo cambios entre los dos audios.

Al final de todo, se les explico a la mayoría de los participantes sobre la esteganografía, una breve plática de cuál es el objetivo y como es que se relaciona con este trabajo; explicándoles porque era necesario que eligieran una melodía a su gusto, mostrándoles cual es el mensaje que se ocultó y a grandes rasgos como es que trabaja el sistema, ya que el 86% de los encuestados desconocían el termino y aunque se pudiera encontrar algún cambio significativo no podrían saber cuál sería el origen del mismo.

4.2.2 Análisis de espectro

En este estudio se analiza el espectro de dos audios, el audio original y el modificado por el sistema, estos audios se examinan con la salida de las herramientas mencionadas anteriormente “Spek – Acoustic Spectrum Analyser y Audacity”.

Para estudio se ha elegido una canción In memorian de Mago de oz, el pequeño texto que se oculta es “Este audio sirve como prueba para el análisis de espectros.”; se realiza el proceso con la técnica propuesta, ver figura 4.2.2.

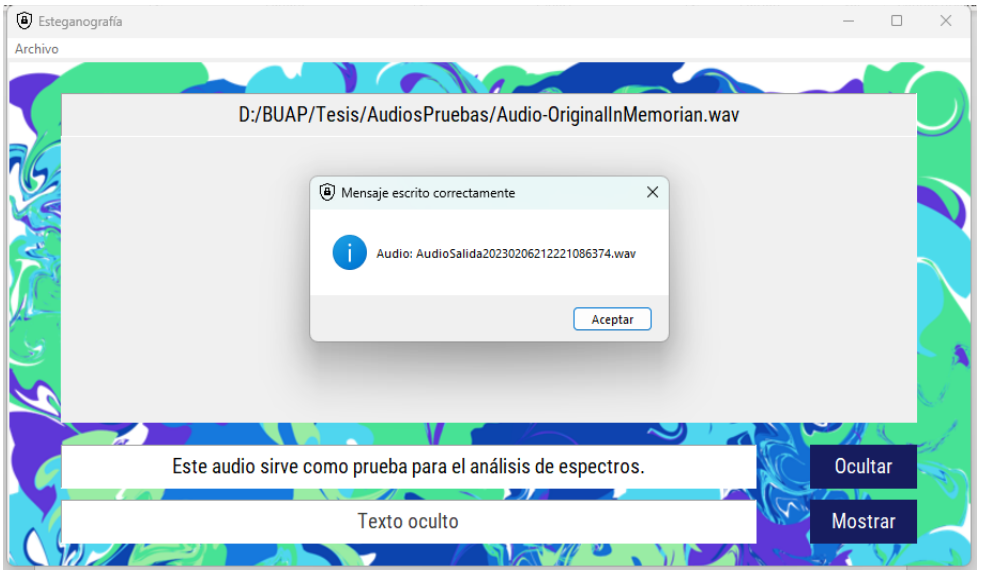

Figura 4.2.2. Texto oculto para análisis de espectros

También se comprueba que se puede leer nuevamente el texto en el audio de salida, ver figura 4.2.3.

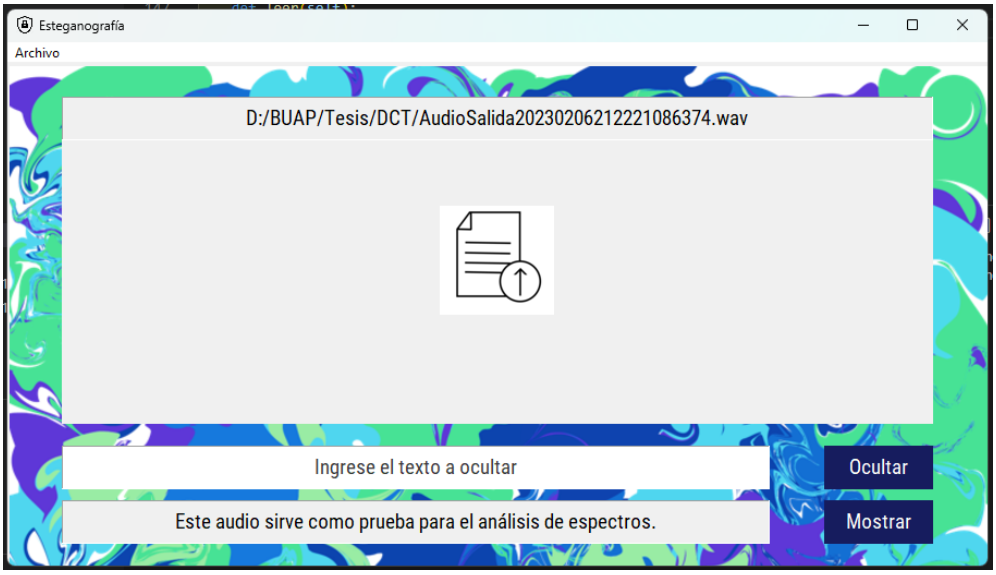

Figura 4.2.3 Extracción de mensaje

Ahora que hemos ocultado el mensaje y comprobamos que se muestra la información sin inconvenientes se procede al análisis de los espectros. Se

han elegido las dos herramientas ya que se muestran dos tipos de espectros diferentes.

4.2.3 Analisis con Spek – Acoustic Spectrum Analyser

El estudio se realiza en la versión 0.8.2 de la herramienta, es una aplicación bastante rápida que utiliza la transformada discreta de Fourier para generar su salida.

En las figuras 4.2.4 y 4.2.5.se muestra la salida de los espectros de los audios utilizados para este estudio, para identificarlos se muestra el nombre de este en la parte de arriba de la ventana, estos nombres coinciden con los utilizados en la sección anterior.

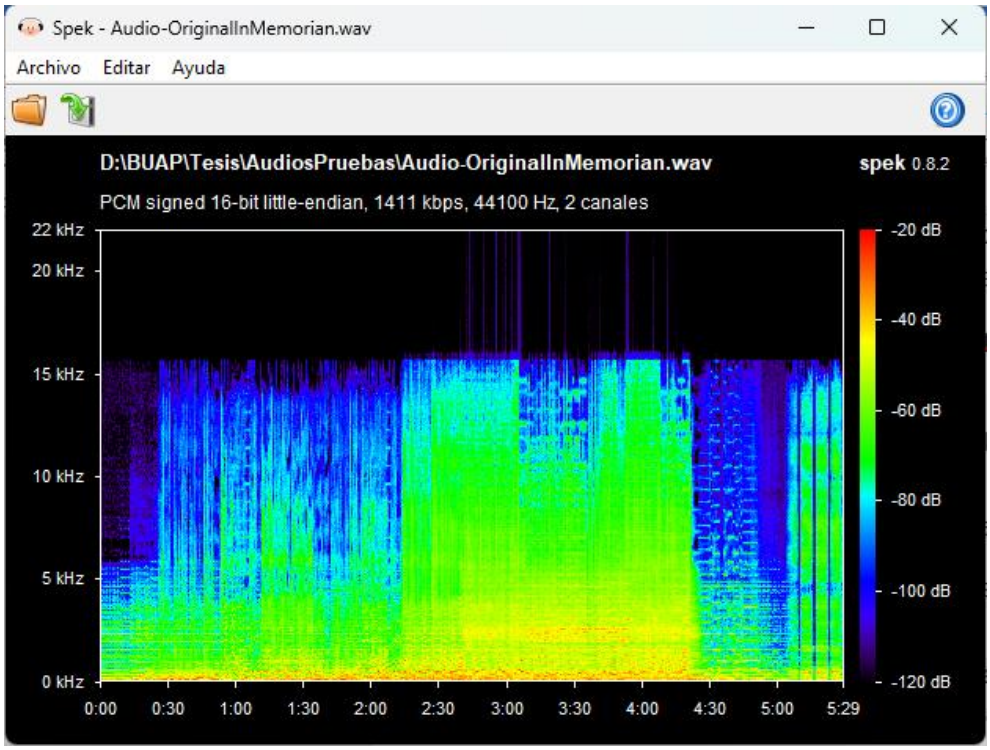

Figura 4.2.4 Audio solo con melodía.

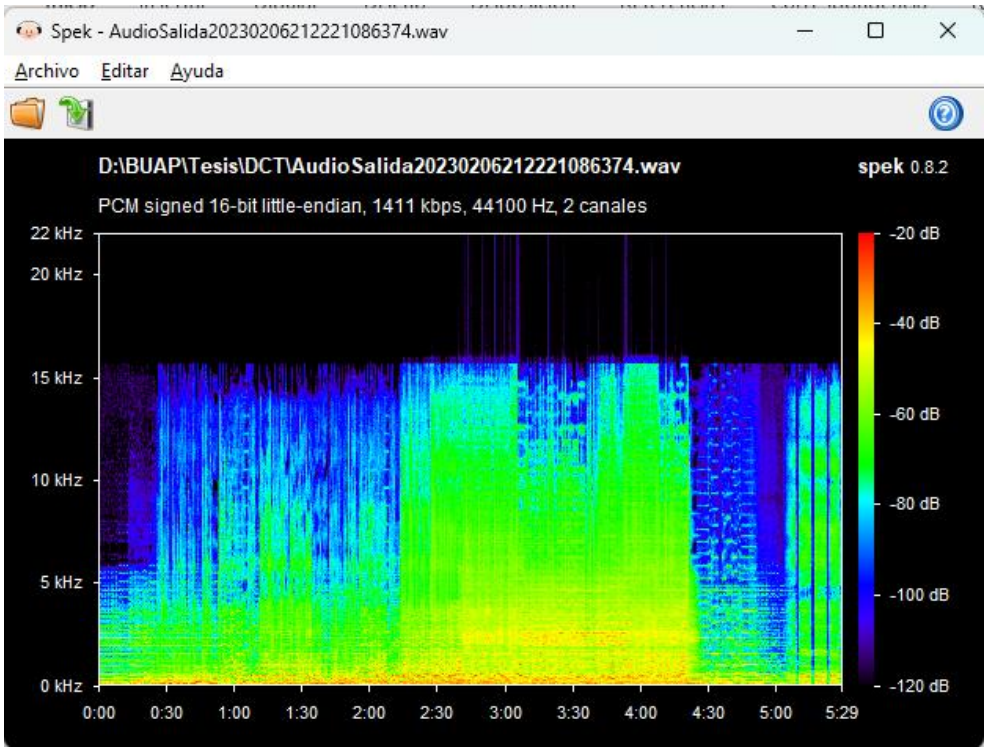

Figura 4.2.5 Audio con melodía más mensaje

Como se pude observar, los espectros son prácticamente iguales. En la parte de arriba se muestra la salida del audio original y en la parte de abajo el audio modificado.

4.2.4 Análisis con Audacity

Audacity fue creado por Dominic Mazzoni y Roger Dannenberg; está orientado principalmente en la edición de audios y cuenta con herramientas para ello; en este caso se utiliza ya que muestra un espectro diferente al observado anteriormente; Se analizan también los audios anteriormente procesados. En este espectro se muestran dos canales de salida, el izquierdo y derecho. Al importar los audios se muestra de la siguiente manera, ver figura 4.2.6 y 4.2.7.

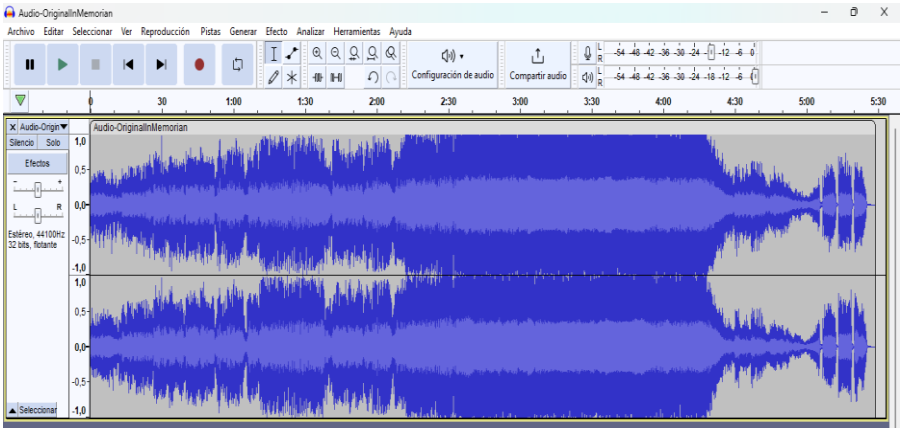

Figura 4.2.6 Espectro de audio original

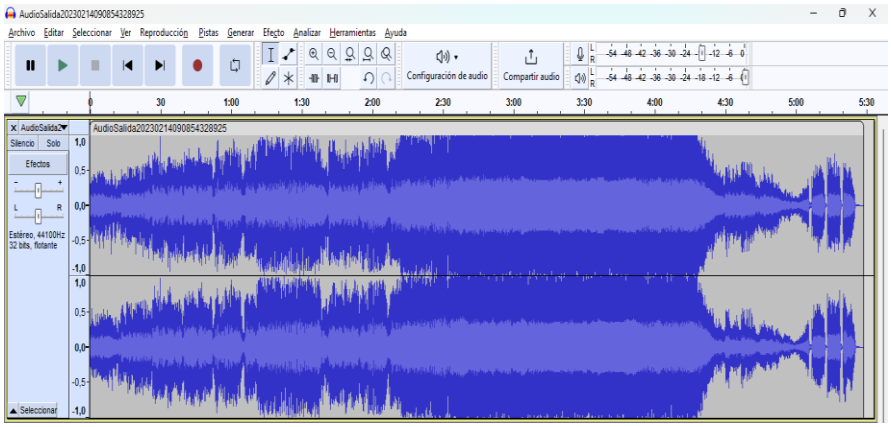

Figura 4.2.7 Espectro de audio modificado

El software, a diferencia del anterior; tiene la capacidad de acercar o alejar la imagen, lo cual es muy útil para analizar con mayor detenimiento. Después de acercar la imagen en los dos audios, se puede notar que en los primeros 1.25 minutos no se muestra ningún cambio muy marcado, es decir que no se puede observar ruidos o algún otro factor que afecte el rendimiento, ver figura 4.2.8 y 4.2.9.

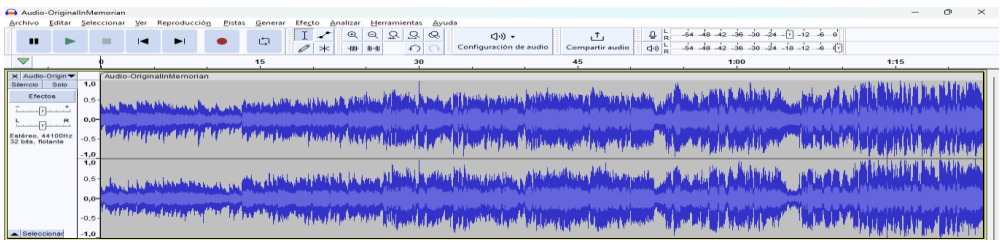

Figura 4.2.8 Espectro original ampliado

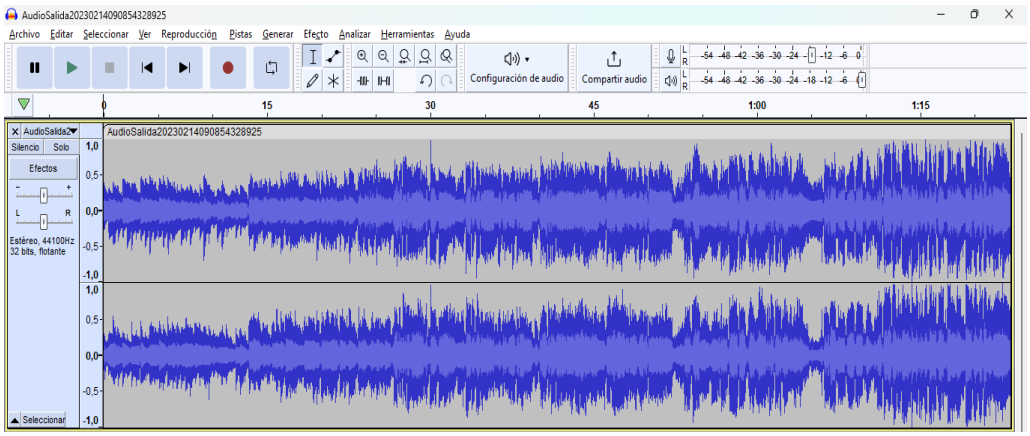

Figura 4.2.9 Espectro modificado ampliado

4.3 Conclusiones

Después de realizar pruebas podemos llegar al punto de decir que la técnica propuesta cuenta con un buen nivel de seguridad, ya que no es percibido por el oído humano, solo se observa con software y esto es realmente bueno ya que pasa desapercibido por el ser humano.

También se obtuvo el espectro de dos archivos, el audio original y el modificado, en el cual se pidió ayuda a dos personas para buscar algún cambio a simple vista y no se encontró ninguno.

Además, se cree que se puede hacer alguna mejora dentro de la técnica; entre ellos hacer alguna fusión con criptografía para que aparte de estar oculto el texto se pueda transmitir de manera segura y en el cual se debe saber el algoritmo criptográfico usado y la clave si fuera necesaria, además de los aspectos con los que ya se cuenta.

Referencias

[1]*"Masoud Nosrat; Ronak Karimi; Mehid Hariri, An introducction to steganography methods, Wordl Applied Programming, vol 1, núm 3, 2011, p. 191-195, ISSN: 222-2510"*

[2]*"Martínez Villacampa Elena; Sayrol Clos Elisa, Introducción al Watermarking, Buran, vol. 19, 2003, p. 73-77, ISSN: 2013-9713".*

[3]*"Jasiya Fayaz; Sanjay Sharma, Video Based Steganography using DCT and LSB - Survey, Journal of Applied Science and Computations, vol. 6, núm 3, 2019, p. 262-268, ISSN: 1076-5131."*

[4] *"Swati Malviya; Manish Saxena; Dr. Anibhuti Khare, Audio Steganography by Different Methods, Inernational Journal of Emerging Technology and Advanced Engineering, vol. 2, 2012, p. 371-375, ISSN: 2250-2459".*

[5] *"Kadir Tekeli; Rifat Asliyan; A Comparison of Echo Hiding Methos, The Eurasia Proceedings of Science, Technology, Engineering & Mathematics, vol 1, 2017, p. 397-403, ISSN 2602-3199."*

[6] *"Carlos Velasco Bautista; Julio López Hernández;Mariko Miyatake Nakano; Héctor Pérez Meana. Esteganografía en una imagen digital en el dominio DCT, Científica, vol 11, núm 4, 2007, p. 169-176, ISSN: 1665-0654."*

[7] *"José Noé Poveda Zafra, Spread Spectrum, Academia y Desarrollo, vol. 5, 1999, p. 71-78, ISSN: 2000-0000."*

[8] *"Rupansh; Preti; Vandana,Esteganografía de audio por secuencia directa Especrto Ensanchado, Internaional Journal of Computer Trends and Technology, vol. 13, 2014, p. 83-86, ISSN: 2231-2803."*

[9] *"Martha Isabel Ladino A.; Paula Andrea Villa S.,Espectro Ensanchado Por Secuencia Directa, Scientia Et Technica, vol. 44, 2010, p. 167-172, ISSN: 0122-1701."*