

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación



TESIS

**Envíos de mensajes estenográficos por
medio de notas musicales a través de una
guitarra eléctrica usando Transformada de
Wavelet**

Presenta: Sergio Itzman Montiel Montero

Para obtener el grado de: Licenciatura en ingeniería en Ciencias
de la Computación

Director: Dra. Barbara Emma Sánchez Rinza

Puebla, Pue, Febrero 2026

CAPÍTULO 1

ESTEGANOGRAFÍA

1.1 INTRODUCCIÓN

La esteganografía es una técnica que tiene como propósito el ocultar información a través de distintos medios y/o de distintas maneras, así como existen otras técnicas que también tienen el mismo propósito, pero lo que caracteriza a la esteganografía principalmente, es que oculta la información dentro de algún objeto o medio por el cual se necesita transmitir el mensaje, solo que con la excepción de que el mensaje no tiene que ser reconocido, aunque se sepa que existe algún mensaje dentro del mismo, por parte de algún observador que sepa o intuya del mensaje, de esta manera se esconden distintos tipos de mensajes, datos o información por distintos medios o portadores, a grandes rasgos esto es lo que la esteganografía tiene como finalidad; pero existen muchas más características o detalles de lo que se necesita indagar dentro de la esteganografía para poder explicar con mayor precisión como es que la misma funciona, una de ellas, por ejemplo, es que las dos partes deben de saber que hay un mensaje oculto, algunas cuestiones más se irán explicando a través del documento para poder generar una idea más clara de lo que es y que se trata la esteganografía.

1.2 ESTADO DEL ARTE

Esta técnica ha sido empleada desde hace mucho tiempo por lo que de alguna manera u otra la esteganografía ha estado presente en la mayor parte de la historia de la civilización humana, una muestra de ello, es que el término esteganografía proviene del griego “steganos” que significa *secreto* y “grafia” que significa *escrito*, con lo que nos da la relación de su propósito, los primeros hallazgos de ello datan de la Grecia antigua, en donde aparecía la palabra en algunos de sus libros del geógrafo e historiador Heródoto (ver figura 1.2.1), una historia relata que enviaron un mensaje a Esparta para comunicarle el riesgo de que habría una invasión por parte de Xerxes a Grecia, pero de manera de que el mensaje nunca se encontrara ni hubiera cierta sospecha de que se había difundido alguno, la forma en cómo se

logró enviar el mensaje fue por medio de unas tablillas de madera (ver figura 1.2.1), las cuales tenían cera encima, lo que se realizó según el relato es que quitaron la cera encima y grabaron el mensaje en la tablilla de madera, para después ponerle cera encima otra vez, con esto logrando transportar las tablillas sin que se tuviera alguna sospecha [1,2]. Otra historia de la misma época relata que la forma de mandar un mensaje fue cuando un personaje hizo que a su esclavo de mayor confianza se le rasurara la cabeza y se le tatuara dicho mensaje en la cabeza, después de un tiempo considerable que le había crecido el cabello fue enviado hacia el otro personaje al que iba el mensaje, con las instrucciones de que le rasuraran la cabeza [1].



Figura 1.2.1 Heródoto y las tablillas de madera con cera que se mencionan en la historia del mensaje a Xerxes.

En el siglo XV un personaje italiano, Giovanni Battista della Porta (ver figura 1.2.2), quien fuera filósofo, alquimista, comediógrafo e investigador logró esconder mensajes ocultos dentro de huevos cocidos, la técnica o método que empleo fue primero crear una tinta que está en base de una mezcla de una onza de alumbre y una pinta de vinagre, una vez haciendo la tinta ya se podía escribir en el huevo, la tinta penetra a través de la cascara porosa del huevo, dejando el mensaje en la superficie albumina del huevo, la única forma de leer el mensaje del huevo es quitando la cascara del mismo [1].



Figura 1.2.2 Retrato de Giovanni Battista della Porta.

Como se ha mencionado, llegan a existir diferentes escenas en donde se demuestra la esteganografía, pero hasta cierto punto no hubo ningún libro que lo remarcara hasta el siglo XVI en donde Johannes Trithemius ver (figura 1.2.3), quien fue un monje alemán que dedico su vida a la criptografía, matemática, escritura y astrología, escribiera un libro en donde por primera vez se vería y se daría un realce a la esteganografía, cuyo título fue “*Steganographia*”, en este libro se plasmaban temas acerca de la manera de ocultar mensajes, así como temas ocultistas donde describía métodos para conjurar espíritus, la obra fue publicada en el año 1600 pero fue removida en 1900, puesto que se hablaba de magia y algunos temas ocultistas, este libro viene con otros dos tomos, que en los últimos años se pueden consultar pero con un tema más cercano a la criptografía y no tanto al ocultismo como se redactó desde el principio [1].

Durante la segunda guerra mundial se realizaron maneras esteganográficas en donde se transportaban a través de microfilms, pero el sistema más popular que se conoció llevo por nombre “Null Cipher” el cual consistía en mandar los mensajes a través del código morse, eso lo lograban mediante los microfilms donde los puntos de las “i” significaban puntos importantes dentro del código, con esto conseguían que el mensaje pareciera ni siquiera estar ahí [1].



Figura 1.2.3 Retrato de Johannes Trithemius.

1.3 TINTA INVISIBLE

Después de explorar y ver diferentes maneras posibles para aplicar alguna técnica de esteganografía, también podemos encontrar dentro las tintas invisibles, no se tiene un registro de una época concreta en la cual surgió, ya que se encontraron rastros similares durante diferentes épocas, se encontraron registros en donde los romanos y los griegos utilizaban esta tinta, también con distintos fines, llegó o logró ser parte fundamental durante las guerras, ya que se necesita que no se filtre ninguna información. Durante la revolución de la guerra entre Estados Unidos e Inglaterra, personajes destacados utilizaron distintos métodos, entre ellos, George Washington el cual utilizaba tintas orgánicas como tintas sintéticas como técnica o medio de información, esto ayudaba porque para ver el mensaje secreto se tenían que utilizar diferentes sustancias para ver su contenido, por ejemplo, se encontraba que si el documento tenía alguna letra en alguna esquina de la carta, esto representaba que sustancia o materiales se necesitaba para revelar la información, eran diferentes tipos de sustancia o materiales, tales como si en la esquina de la carta aparecía una F esto significaba que tenían que utilizar fuego, o como si apareciera una A lo que significaba que para revelar tenían que utilizar sería ácido [1,3].

Durante los años posteriores se siguió utilizando la tinta invisible, pero logrando una mejoría en todos los aspectos, durante la primera y segunda guerra mundial, así como la guerra fría, en estas ocasiones la tinta invisible se utilizó como recurso técnico más avanzado y elaborado, ya que empezaron a utilizar laxantes y remedios antipiréticos lo cual generaban mejores tintas invisibles, aparte de las inclusiones de esos materiales también se incluyeron otros como hierro, sulfato de cobre y sales de cobalto, las cuales permitían mostrar los mensajes a través del amoníaco y carbonato de sodio. En muchas otras manera se utilizó la tinta invisible para realizar actividades seguridad así como se vio en los anteriores casos de las guerras que se han llevado a cabo, existen dos tipos de tintas invisibles, con esto se explicara o se dará un panorama mejor de cómo fue la evolución y como ayudo esto a que se realizaran distintos tipos de tintas, la primera tinta es la **tinta orgánica** la cual utiliza elementos naturales en su mayor parte, tales como vinagre, jugo de limón, jugo de cebolla, leche y en algunos otros casos como sudor, sangre y saliva; para poder descifrar el mensaje de este tipo de tintas es importante el calor, por lo que se utiliza fuego o algo incandescente como focos y/o bombillas que permitan generar el calor y lograr ver el mensaje con la tinta, el otro tipo de tinta es la **tinta simpática o sintética**, esta tinta son más complicadas para su preparación, ya que se necesita de un químico el cual sepa las combinaciones de ciertas sustancias para su creación, ya que todas estas tintas necesitan de químicos para que funcionen [3].

1.4 PROBLEMA DE LOS PRISIONEROS

Ya que se han explicado como surgieron distintas alternativas o técnicas para realizar la esteganografía, existe un problema que plantea la esteganografía como situación y cuales tendrían que ser sus características, el problema es comúnmente llamado como **el problema de los prisioneros** el cual plantea la siguiente situación; se encuentran dos prisioneros cómplices encerrados en celdas separadas, el *prisionero 1* y el *prisionero 2*, los cuales son vigilados por un *guardia*, el cual no les permitió tener un medio de comunicación con las condiciones de que el único medio de envío sea el mismo y que el mensaje antes de ser recibido por el otro prisionero tendría que ser verificado por él, también contemplando que el guardia podría

modificar o alterar los mensajes ya que llegaría existir algún mensaje oculto y con esto distorsione todo el mensaje, después de todo las restricciones y contemplaciones extras que pueden llegar a ocurrir, los prisioneros analizan la situación y deciden arriesgarse para lograr concretar un plan de escape aun sabiendo que se puede sospechar de algún plan de escape, también tienen contemplado que el guardia se detecta algún fallo o alguna intención extra como el plan de escape, él automáticamente cortara la comunicación y todos sus planes se verán afectados [1].

Como muestra o describe el problema tenían que encontrar alguna manera de mandar mensajes encubiertos, es aquí donde surge o se plantea la idea de la esteganografía con este problema de los prisioneros teniendo así el panorama de lo que se ha venido explicando en el documento con este ejemplo. Existen ciertas características que se deben tomar en cuenta para lograr alguna solución como es que no se deberá usar ningún tipo de mensaje cifrado que sea leíble o visible porque si no el guardia sospechara y por consiguiente como se explicó en el problema, quitara la comunicación o podría alterar el mensaje, otra característica demasiado importante y a la cual se le hizo mención en la introducción al inicio de este documento, es que las dos partes estén en un acuerdo acerca de cómo será el mensaje oculto y como se revelara, así como se vio con las tintas invisibles y su forma de cómo lograr ver el mensaje a través de alguna letra en la esquina de la carta [1].

1.5 TIPOS DE GUARDIAS

En el anterior caso del problema, podrían existir diferentes tipos de guardianes debido a las acciones que puede realizar o los diferentes permisos a los que él se puede otorgar referente al canal de comunicación, los tres diferentes tipos de guardianes son el guardián pasivo, guardián activo y guardián malicioso, estos tipos de guardianes son utilizados para demostrar más qué tanta manipulación puede llegar a tener una entidad externa sobre el canal de comunicación [4].

El **guardián pasivo**, como su nombre podría llegar hacer alguna referencia, es el guardián que no interviene dentro del canal, solo tiene la posibilidad o solo le limita

a leer, el no realiza ningún cambio al mensaje ni tampoco los altera, pero si sospecha de algún tipo de mensaje cifrado tomara alguna acción como la cancelación o el cierre del canal de comunicación como en este problema [4].

El **guardián activo** tiene una acción diferente a comparación del guardián pasivo, ya que este no solo tiene o logra realizar la acción de lectura sino que él también puede modificar lo que está escrito en el mensaje, como se había mencionado también como cierto problema que se tiene que evitar con la esteganografía, pero si encuentra la manera podría alterar la información del mensaje y tiene manipulación del canal de comunicación, pero solo con la condición de que se detecte alguna sospecha o irregularidad [4].

El **guardián malicioso** tiene, al igual que el guardián activo, realizar la acción de modificar o alterar los mensajes así como la lectura del guardián pasivo, su mayor diferencia de él con el activo es que él no tiene condición como el activo, ya que él puede alterar los mensajes y el canal de comunicación en cualquier momento, por lo cual este es peligroso y por eso lleva ese nombre, ya que puede alterar cualquier parte de la comunicación, tomando esta parte fuera del problema esto podría dañar cuando se habla de un sistema de seguridad y lo que menos se intenta es tener intrusos en el canal de comunicación [4].

Como se ha descrito en los párrafos anteriores, la esteganografía se ha presentado desde hace mucho tiempo durante toda la civilización humana, desde la época de los griegos hasta la época de la guerra fría, teniendo en consideración lo anterior, en la actualidad la esteganografía moderna que se utiliza es totalmente diferente, ahora lo podemos encontrar en su mayoría de manera digital, debido a la gran influencia de la tecnología que se ve representada en todos lados actualmente, esto se puede notar porque empieza desde celulares hasta ordenadores y/o laptops, donde la cantidad de datos que se almacenan a través de servicios por internet o en redes sociales genera que se necesita una mayor configuración para tener mejor seguridad para cubrir los datos o encubrir paquetes digitales donde no se necesita información expuesta o que se sepa existe alguna información dentro, con esto sabiendo que se pueden generar diferentes maneras de mandar diferentes tipos de

archivos con la certeza de que no será fácil descifrarlos. De esta forma se encuentran, así como la tinta invisible o las tablillas de madera que fueron en su momento un medio de esteganografía, ahora se encuentran a través de los medios digitales multimedia, los cuales se pueden encontrar de manera como por medio de imagen, audio, video y otro de diferente categoría, que son los documentos, cada uno de estos medios mencionados con anterioridad necesitan ciertos algoritmos y trabajo o manipulación de ciertas técnicas digitales y computacionales. Pero antes de especificar y dar más profundidad a los medios y como se utilizan, se destacarán las técnicas de esteganografía según el modelo general.

1.6 TÉCNICAS DE ESTEGANOGRAFÍA PURA

Las técnicas de esteganografía se dividen en tres, la primera técnica es el **sistema de esteganografía de manera pura**, en este sistema no se requiere o necesita de ninguna información previa para el intercambio de mensajes, puedan ser claves o algún otro método para cierta seguridad, lo único que requiere este medio es suficiente prudencia o moderación. La segunda técnica es el **sistema de esteganografía de clave secreta** en este sistema se necesita información previa para lograr encontrar el mensaje, ya que el emisor pone el mensaje en el objeto portador, y la única forma que sepa de este mensaje el emisor y su forma de conocerlo es con la clave que ambos comparten, que es la respuesta o solución al descifrado del mensaje, por ende, tiene que existir comunicación previa como se definió en la primera oración describiendo este sistema. Por último, la tercera técnica es el **sistema de esteganografía de clave pública**, este sistema funciona a partir de dos claves, una privada y una pública, la clave privada es por donde se enviará el mensaje del emisor, mientras que la clave pública será almacenada en una base de datos pública, el punto clave de este sistema es que la clave privada sirve para reconstruir y generar el mensaje que se mandó [5].

1.7 REQUERIMIENTOS IMPORTANTES DE LA ESTEGANOGRAFÍA

Una vez explicadas estas técnicas tocaremos puntos importantes para entender porque los medios antes mencionados deben de cumplir con cierta rubrica por lo que se explicarán a continuación los requerimientos fundamentales de la esteganografía. Como sabemos la esteganografía tiene que pasar de ser percibida, por lo que un requerimiento fundamental será la **invisibilidad**, esto hace mención de que el objeto portador que va a llevar mensaje debe contener una buena calidad para que el proceso de la esteganografía sea eficiente y eficaz, sino por lo contrario no serviría la esteganografía, existen dos tipos de subcategoría *la invisibilidad perceptiva y la invisibilidad estadística y algorítmica*. Como también sabemos puede llegar a sufrir algún grado de manipulación nuestro objeto portador del mensaje por parte de entidades externas se debe de procurar, lo que nos lleva al siguiente requerimiento; la **robustez**, la cual se encargara de crear cierta protección o inmunidad ante la manipulación del portador, este es el que brindara que tan eficiente será la seguridad ante la manipulación de alguna entidad externa y aun así no se encuentre ninguna información. Sabemos que también depende el medio de donde se envié la información será más viable para poder enviarlo y tener mayor seguridad, aunque en buena medida también depende que tan eficiente sea para nuestro mensaje, por lo que la **capacidad de embebido** nos dará la cantidad eficiente y si el medio es viable para su utilización [6].

1.8 Técnicas de esteganografía a través de medios multimedia

Así como se ha mencionado a través de esta introducción, que se puede realizar la esteganografía a través de distintos medios, tales como imágenes, audio, video y textos, se explicarán a continuación algunos formas y métodos que son utilizados para la esteganografía actual en algunos de esos medios, logrando con esto concretar cómo es que hoy en día esos tipos de archivos son utilizado ya que se tiene una idea más general acerca de lo que es la esteganografía, así como algunos métodos antiguos y propiedades de la misma.

1.8.1 Esteganografía por medio de imágenes

Para empezar a visualizar en un plano más realista la esteganografía moderna, se empezará por describir los diferentes algoritmos que se utilizan para el tipo de medio de imágenes. Por lo general las imágenes utilizan una paleta de colores, a excepción que la imagen este en escala de grises, encontraremos que tiene ciertas combinaciones de colores, a cada color se le asignara en un vector y un índice, con lo que con esto se podrá formar la paleta de colores, por lo que el índice que apunte a ese vector será el color guardado que tenga dentro ese píxel, por lo que con esto se generara la imagen. Sabiendo esto podemos llegar a inferir que lo que tendremos que hacer es modificar los valores de los pixeles, es decir, tendremos que modificar los vectores en donde se aloja el color, teniendo esto en cuenta es donde se dará la primera muestra de algoritmo que se puede emplear, el algoritmo LSB, el cual será detallado a continuación en este documento; sabiendo que podemos modificar la paleta de colores esto nos da una gran variedad de campo el cual podemos realizar, ya que si tenemos N colores, tendremos $N!$ maneras disponibles de como colocarlos, por lo que aunque sean pocos colores tendremos suficientes formas de cambiar los vectores y en cierta manera la paleta de colores [7].

El algoritmo LSB, que por sus siglas en inglés significa *Low Significant Bit*, se basa en modificar los bits que tienen menor valor o aportan menor valor en el anfitrión, teniendo esto en cuenta sabemos que estaremos modificando los bits menos significantes, por eso el nombre de esta técnica, existen diferentes métodos de estudio de esta técnica y se nombrara algunos de ellos a continuación, estos métodos pertenecen a los métodos de sustitución, por cómo se explicó previamente que realizan un cambio a los pixeles, todos ellos pertenecen a dicho grupo y en esta sección solo se hablaran de estos.

Los métodos son: *Simple LSB*, *Optimal LSB* y *Método PVD*, cada uno de ellos será explicado de forma general para lograr dar un entendimiento mejor de la técnica del algoritmo LSB [8].

- **Método Simple LSB**

Es el método más simple, no existe gran dificultad ya que también es el más directo, este consiste en ocultar cierta cantidad de bits en cada píxel de la imagen, modificando los bits menos significativos de cada píxel y con esto logrando introducir el mensaje que queremos ocultar en cada píxel.

- **Método Optimal LSB**

Este método trata de lo mismo que el anterior, la diferencia más notable y como su nombre lo menciona, es que logra realizar el método de una manera óptima, la manera en cómo lo logra es que cambia el bit menos significativo después de la cierta cantidad de bits que se utilizaron para cambiarse, explicado de otra manera, la cantidad x de bits que se quieran modificar, hasta este punto es como el método *Simple LSB*, pero este caso se modificara el bit $x+1$. Se mostrará un ejemplo para lograr una mejor explicación de lo que este método realiza. Encontramos el siguiente vector de bits que pertenecen a un píxel (*Figura 1.8.1*) que como resultado o valor de este píxel es de 201 en base diez, y valor que se ocultara dentro del píxel será el 111 en base dos que en base diez tiene el valor 7, como el valor se transfiere a la cantidad(x) de los bits menos significativos, por lo que el nuevo vector(*Figura 1.8.2*) obtiene el valor 207 en base diez, pero como se mencionó antes en el texto el cambio que se realizara será en el bit $x+1$, generando con esto un nuevo vector (*Figura 1.8.3*) el cual tendrá un valor de 199 en base diez, consiguiendo con esto una diferencia de menor en comparación al del píxel original con la forma en la que solo se añadió la cantidad en concreto, ya que nuestro bit original tiene un valor de 201 y nuestro segundo vector de 207, teniendo como resultado una diferencia en valor absoluto de 6, a diferencia de que nuestro último vector cuenta con una diferencia en valor absoluto de 2.

1	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---

Figura 1.8.1 Vector original del píxel.

1	1	0	0	1	1	1	1
---	---	---	---	---	---	---	---

Figura 1.8.2 Vector del píxel modificado con los bits que se ocultaran.

1	1	0	0	0	1	1	1
---	---	---	---	---	---	---	---

Figura 1.8.3 Vector del píxel modificado con los bits que se ocultaran.

- **Método PVD**

El método PVD o *Pixel-Value Differencing* permite variar la capacidad de los píxeles dependiendo de la zona en la se encuentren, ya que para poder modificarlos es más fácil modificar los que se encuentran en los bordes a diferencia de los que se encuentran internamente, para que este método funcione se tiene que partir en bloques la imagen para poder realizar este método, en una imagen de escala de grises que su rango de valor va del 0 al 255 y que de cierta manera cambia el valor pero no puede ser tan drástico, se tiene que emplear que dependiendo de los píxeles que tenga alado se pueda realizar la variación de la capacidad del píxel.

1.8.2 Esteganografía por medio de audio

Lo expuesto anteriormente nos describe el poder de la manipulación de la información por píxeles de una imagen, lo mismo lo podemos encontrar en el aspecto de audio y el cual es el más importante para definir este trabajo, en base a esto se darán conceptos y algunos algoritmo o métodos que nos harán entender, sin llegar a profundizar, como es que funciona la esteganografía en el aspecto de audio. Hay que recalcar que algunos de los siguientes algoritmos también se pueden emplear para el uso de medios de imágenes, basándose principalmente en el mismo principio, pero en este caso solo se desarrollara de la manera por medio de audio, también se mencionarán algunas fórmulas matemáticas las cuales se verán con mayor detenimiento mucho más adelante en este documento.

En esta sección se explicarán métodos de sustitución, así como los de la sección de *esteganografía por imágenes* y también se incluirá el método de ocultación en el dominio transformado, cada uno con métodos diferentes los cuales determinan alguna manera para ocultar la información.

Los métodos de sustitución estarán compuestos por los métodos: *paridad de bloques* y *modificación de codebooks*, y los métodos de ocultación en el dominio transformado serán: *modificación de los coeficientes frecuenciales* y *phase coding*.

- **Paridad de bloques**

Este método, como se mencionó que pertenece al método de sustitución, lo que realiza es ingresar o modificar datos dentro de su objeto portador, así como los demás vistos de este método, pero lo que este método realiza es dividir al objeto portador en diferentes segmentos o secciones en tamaños diferentes y especificados, con esto haciendo referencia a su nombre, después de esto se calculara la semejanza de entre cada bloque, y si un bloque en posición n es igual al n -ésimo bit del mensaje a ocultar no se hará nada, pero de lo contrario, si no es igual se modificara el bit menos significativo de algún elemento del bloque. Por lo que por parte del receptor lo único que tendrá que realizar será la revisión de que los bloques sean iguales, es decir, su paridad, pero también necesitara usar la misma ordenación que ocupo el emisor [8].

- **Modificación de codebooks**

Este método realiza algunas operaciones que requieren más conocimientos por parte de algunas operaciones que se necesitan, en este método se tendrán que utilizar dos transformadas, la transformada de Fourier y la transformada de Wavelet, por lo que su grado de complejidad como se comentó anteriormente, se necesita un conocimiento previo para su mejor entendimiento, pero lo que se explicara en los siguientes párrafos será una forma general de lo que este método de *modificación de codebooks* realiza, sin entrar en demasiada terminología. Este método se concentra en utilizar la transformada de Fourier a corto plazo y la transformada de Wavelet las cuales son alimentadas en paralelo con la señal original, la transformada de Wavelet es un tipo diferente de transformada ya que en esta su principal acción o propósito, ya que es una herramienta matemática, trabajar en base a señales y con esto poder trabajar de alguna manera matemática con ellas, en este caso la utilizaremos para que se puede manipular las señales y poder insertar mensajes ocultos, ya que hemos podido ver diferentes formas de cómo

lograrlo, con esto se explicara una manera diferente y con fórmulas extras a las de solo la variación de los bits [9].

La transformada de Fourier corta se utiliza ya que nos podrá brindar cierto rango de ruido dentro de nuestra onda o señal, con lo que en si se encargara de definirnos la cantidad de componentes tonales, sabiendo que nos brinda los componentes tonales, nos brinda como se comentó, el rango de ruido en el que podre manipular para poder obtener algún enmascaramiento dentro de la señal, ahora después de tener un rango, viene la inserción del mensaje en el portador, en este caso nuestra señal, en la cual se empleará la transformada de Wavelet, ya que en esta los datos con los que contamos para ocultar funcionaran como índices que servirán para la selección del cuantificador con el cual se usara para cuantificar lo que se obtenga de la señal de la transformada de Wavelet. Lo siguiente es lo que le da el nombre a este método, ya que usaremos un codebook o libro de códigos, en el cual se nos guardarán nuestros códigos disponibles para poder cuantificar nuestra señal de la transformada, pero los códigos serán guardados en forma de árbol binario, cada cadena de datos a ocultar seleccionara entre la rama 0 y 1 en cada nodo, y así de esta manera llegara a una hoja del árbol binario con lo que con esto obtendremos la hoja inicial para la manera de la descryptación, lo que se explicara más adelante, una vez con hecho los pasos anteriores, árbol binario lleno y nuestra hoja representante, se empleará el uso de la máscara obtenida gracias a la transformada de Fourier a corto plazo y se comprobaran si los datos en manera de ruido es aceptable para la introducción dentro de la señal. Mientras que, por parte del receptor, así como se mencionó, empezara un recorrido hacia atrás iniciando un recorrido desde la hoja del árbol donde se terminó guardando la última parte del mensaje oculto [9].

Teniendo nuestro libro de códigos o codebooks todas nuestras palabras, determinaremos que la distancia entre dos palabras nos demostrara la cantidad de ruido que puede soportar el algoritmo para su introducción al canal. Cabe resaltar que la propiedad que se usa para la aplicación de este tipo de técnica se utiliza la adición de ruido gaussiano blanco hacia la señal, la cual como se dijo en un

principio, también puede ser añadida por medio de imágenes a través de la misma forma del ruido [9].

- **Modificación de los coeficientes frecuenciales**

Esta técnica se concentra en modificar los coeficientes frecuenciales dentro de la señal, por lo general los métodos que alteran el dominio de la señal tienden a resultar muy poco robustos, debido a que si la señal no se altera de alguna manera no se tendría problema, pero si llega a ser alterada de alguna manera podría resultar bastante perjudicial debido a que toda la información que se ocultó en la señal o portador se perdería; por lo que esta manera de alterar los coeficientes resulta más robusta es la siguiente y es la manera en la que funciona, la forma más normal de su uso es mover los valores de la zona media de la señal, con esto para garantizar de alguna forma ningún ruido o alteración extraña que se pueda llegar notar, después de esta información, lo primero que se realizara será partir la señal en diferentes bloques y se asignaran dos valores cercanos dentro de cada bloque, de los cuales dependerá si la inserción dentro de la señal será de 1 o 0, esto significa, que si dentro de cada bloque tenemos un valor a y un valor b , tendremos la comparación y condición, si el valor a es mayor que el valor b se asignara un numero 0, pero si sucede el caso contrario se asignara el valor 1. Para que el receptor pueda obtener acceso al mensaje oculto lo que tendrá que obtener es la cantidad y el tamaño de bloques dentro de la señal, para que pueda encontrar los valores modificados, en esto podría llevar cierto rasgo de fortaleza ya que el mensaje solo puede ser visto por alguien que en verdad sepa de la señal y cómo manejarla.

- **Phase coding**

Este método no tiene un rasgo de complejidad alto, ya que así como su nombre lo dice, se realiza la codificación a la fase de la señal original, esto quiere decir que la señal se dividirá en segmentos y dentro de cada segmento se modificara la fase que contenga el mensaje a ocultar, pero con esto se obtiene una consecuencia, la cual es que así como se cambien el bit de la fase de ese segmento se tendrán que modificar las fases posteriores para que no se note cambio significativo con las fases siguientes. Con esto también se realiza que entre más cantidad de

información se introduzca se tendrá más robustez, pero perderemos perceptibilidad, al igual si se hacen segmentos muy pequeños, tendremos mayor capacidad, pero perderemos perceptibilidad [10,11].

CAPÍTULO 2

INGENIERÍA DE SOFTWARE

2.1 INTRODUCCIÓN

Como todo software, se requiere realizar una planeación para un correcto funcionamiento y un óptimo desempeño, por lo que se seleccionaron ciertos diagramas que ayudaran a la creación del software, tales como el diagrama de flujo, diagrama de actividades, diagrama de casos de uso y el modelado de clases. Estos diagramas servirán de distintas maneras, para la comprensión del lector y la implementación de esta tesis, ya que con esto se logrará una correcta programación y un buen flujo de trabajo en la forma de organizar las tareas que se realizarán. Por lo que, en este capítulo se mostrarán los diferentes diagramas que enseñarán a nivel general todo el funcionamiento de esta tesis, desde las actividades iniciales hasta las finales que se obtengan en el momento de la ejecución.

Cabe destacar que para lograr esta tesis se necesitarán de herramientas específicas, como lo son en hardware, software y terminología propia del tema, los cuales son indispensables para que todo el proceso sea completado. Todas estas especificaciones serán descritas en la sección de Requerimientos 2.2 y las subsecuentes para con esto generar un panorama claro sobre el proceso esteganográfico y la visualización de las actividades y usuarios explicados en los diagramas.

2.2 REQUERIMIENTOS

Para explicar todos los requerimientos necesarios de esta tesis, se explicará en tres subsecciones diferentes dentro de esta sección, las cuales son hardware, software y glosario, como se describió en la sección al final de la tesis, cada una de ellas cuenta con diferentes características son necesarias para un conocimiento concreto

de todo en lo que se trabajó durante esta tesis, así como se destacaran las herramientas que se emplearan para su implementación.

2.2.1 Hardware

Para realizar la grabación del instrumento y poder manipular dicha grabación se hará uso de herramientas de hardware, como lo son:

Guitarra eléctrica: Este será el instrumento con el que se realizara todo el proceso de la tesis, la cual cuenta con seis cuerdas y veintidós trastes, los cuales nos ayudaran después para poder definir letras en cada uno de los trastes.

Interfaz de audio: Esta herramienta de hardware es de vital importancia, debido a que la interfaz de audio es la encargada de convertir el sonido analógico producido por la guitarra a digital para que la computadora portátil pueda procesarlo.

Computadora portátil (laptop): El hardware de vital importancia en esta tesis, debido a que todo el proceso y toda la implementación se realizará gracias al mismo, todo será desarrollado con computadora portátil con las siguientes características:

Procesador: Intel Core i7-8750H

Memoria: 16 GB

Almacenamiento: 1TB

Tarjeta Gráfica: Nvidia Geforce GTX 1050

2.2.2 Software

Toda la implementación y el desarrollo principal de esta tesis será realizado por medio de software, donde se realizará la implementación de la herramienta matemática, la Transformada de Wavelet, con la cual se realizará la codificación y decodificación del mensaje oculto dentro del audio. Por lo que en esta sección se definirán tres herramientas:

Pycharm: IDE el cual se utilizará como entorno de trabajo para poder desarrollar la codificación y decodificación de la Transformada de Wavelet.

Anaconda: Su principal función será proporcionar herramientas y librerías del lenguaje de Python para poder realizar con éxito todo lo que conlleva esta tesis.

Python: Lenguaje de programación empleado para el desarrollo de toda la implementación de esta tesis, seleccionado por su versatilidad en el manejo de datos y amplias librerías con las cuales se puede manipular de audio, video e imágenes.

DAW: Digital Audio Workstation, por sus siglas en inglés, es un software especializado para producción de audio, este se dedica principalmente a la grabación y producción musical. Por lo que se hará uso de este software para poder realizar la grabación del instrumento y así obtener el sonido digital por medio de la computadora portátil. Existen diferentes DAW's en el mercado, algunos solicitan pago de su licencia, en este caso se hará uso del DAW Reaper, ya que este cuenta con su versión completa sin necesidad de comprar alguna de sus licencias, siempre y cuando su uso no sea lucrativo.

2.3 DIAGRAMA DE CONEXIONES

Como se describió anteriormente, el proceso incluye múltiples herramientas las cuales cuentan con funciones específicas, cada una de ellas teniendo un rol importante dentro de todo el esquema esteganográfico, por lo que el primer paso dentro del proceso de implementación es realizar todas las conexiones correspondientes del instrumento hacia la computadora portátil, por lo que en esta sección se explicarán todas las conexiones con sus especificaciones para obtener el sonido emitido por la guitarra eléctrica hacia la computadora portátil.

Para lograr lo mencionado en el párrafo anterior se necesitara realizar primero la conexión de la guitarra eléctrica a la interfaz de audio (Figura 2.3.1), para este propósito no será necesario realizar alguna conexión entre la guitarra eléctrica y un amplificador, debido a que el amplificador proporciona diferentes matices y mejor claridad en sonido con fines de producción de musical, en este caso la función principal es generar el sonido por medio de la guitarra y obtener la frecuencia emitida por dichos sonidos, para así poder realizar todo el esquema esteganográfico

de manera correcta. Por lo que, la conexión será realizada por medio de un cable de 1/4 pulgada de aproximadamente 2 metros, denominado de esta manera por su entrada, ya que dicha entrada es mucho más grande que la convencional de 3.5 mm con la que cuentan los audífonos analógicos, esta entrada de 1/4 pulgadas es común en conexiones de instrumentos; dicho cable cuenta con la misma entrada de ambos extremos por lo que es indiferente su conexión entre ambos dispositivos.

En el diagrama (Figura 2.3.1) se muestra la interfaz de audio por la parte frontal y trasera, debido a que esta interfaz tiene conexiones por dichos lados de ella. Como se muestra el diagrama la parte frontal es la parte donde se realizó la conexión entre la misma interfaz y la guitarra eléctrica, mientras que en la parte trasera se muestra del lado izquierdo la entrada que conectara la interfaz de audio hacia la computadora portátil, esta conexión será realizada por un cable USB denominado A/B de aproximadamente 40 centímetros, este tipo de cable es particularmente encontrada en producción musical, como en este caso con la interfaz de audio, y en conexiones de impresoras. Este cable cuenta con la entrada USB A la cual se conecta en la computadora portátil mientras que el otro extremo del cable se conecta en la parte trasera de la interfaz del lado izquierdo. En las secciones posteriores se realizará la descripción las configuraciones que se requieren en cuanto a software para realizar su correcta implementación.



Figura 2.3.1 Diagrama de conexión entre dispositivos de hardware.

2.4 Diagrama de flujo

Para poder representar todo el proceso de una manera sencilla y fácil de entender se realizó el siguiente diagrama de flujo (figura 2.4.1) debido a que este tipo de diagramas tiene una lectura simple, ya que cuenta con demasiadas características que facilitan su misma lectura, como lo es, que contenga un inicio y un final bastante marcado o ya sea que se encuentre de manera vertical u horizontal todo el proceso se realiza con un mismo flujo el cual siguiendo detalladamente todas las líneas, así como los procesos que se encuentren dentro del mismo. Por lo que, en este diagrama se pueden identificar que se denotaron factores externos al software como él fue realizar las conexiones de hardware, lo cual se explica en la sección anterior, esto se debe a que no todo el proceso se realiza en software, y para lograr el objetivo final intervienen diferentes factores como los que son totalmente hardware y a estos mismos será necesario realizar la configuración correspondiente para su correcto funcionamiento.

Cada una de las actividades mostrados en el diagrama representan la manera en cómo se realizará el funcionamiento correcto, esto cuando todo este correctamente implementado, por lo que se destaca que en este diagrama de flujo general existen diferentes procesos en los que en cada uno se tendrá que realizar un diagrama de flujo independiente para conocer la correcta estructuración de toda la implementación a nivel detallada, posteriormente se describirán cada y uno de los pasos.

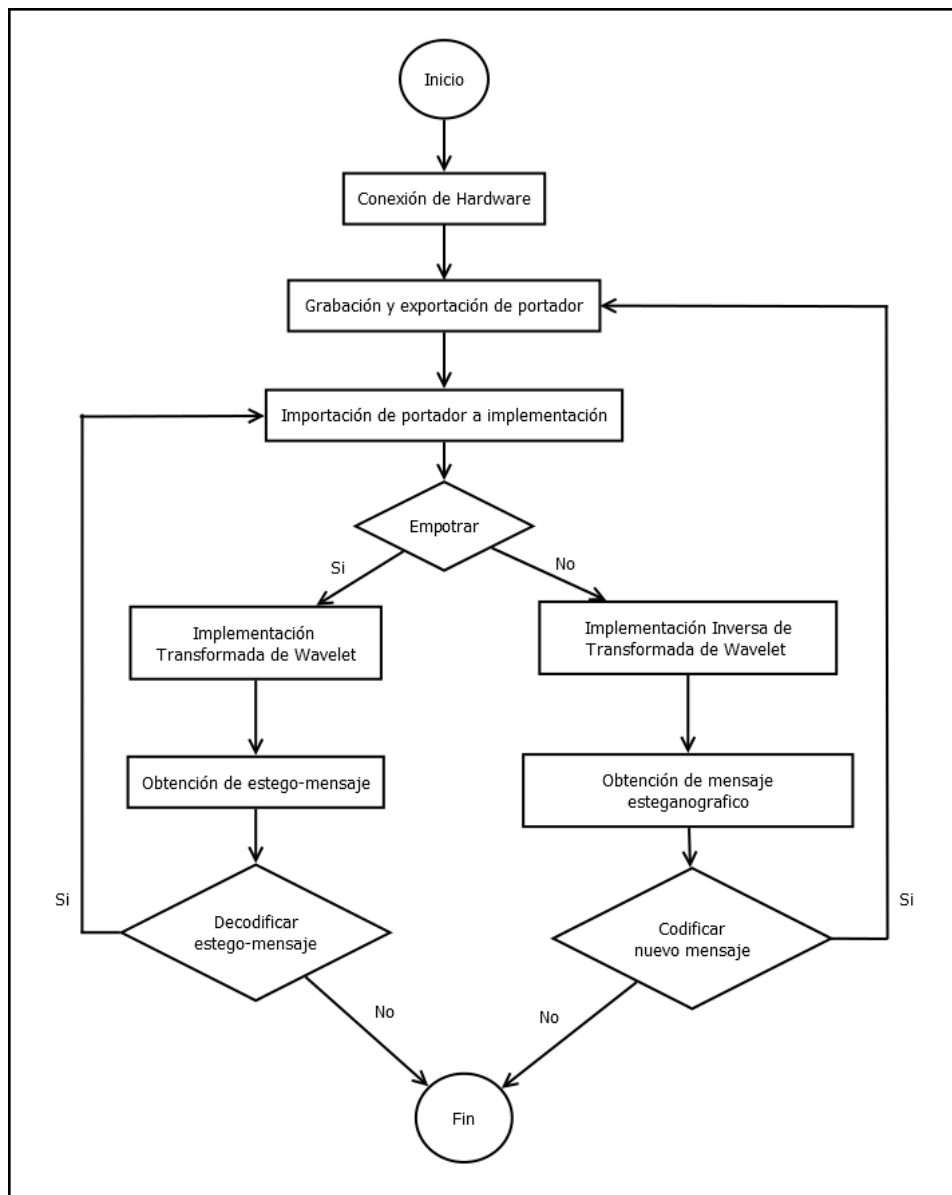


Figura 2.4.1 Diagrama de flujo que representa el esquema esteganográfico.

2.5 DIAGRAMA DE ACTIVIDADES

En la sección anterior se demostró el proceso de generar un estego-mensaje y como revelar el contenido de dicho estego-mensaje, por lo que, para poder resolver el problema principal de la esteganografía, el cual recordando, es enviar información de manera oculta por medio de un canal de comunicación entre dos personas, se decidió realizar el diagrama de actividades para demostrar el funcionamiento de la implementación entre dos usuarios haciendo uso del programa, lo cual combina lo descrito en la sección anterior en el diagrama de flujo, pero a su vez, demuestra la generalidad de toda la implementación incluyendo los distintos factores que se puedan generar con cada uno de los usuarios y sus acciones, como por ejemplo el cifrado y descifrado.

Este diagrama muestra de manera simplificada la parte inicial con respecto a las conexiones y exportaciones del portador o del archivo de audio, como se describió en la sección anterior (2.4), por lo que, en este diagrama muestra que está conformado tres carriles, los cuales representan al usuario embebedor, al usuario extractor y el esquema esteganográfico. En cada uno de ellos se detallan las acciones con las que deben cumplir para completar todo el proceso, lo que demuestra que el usuario embebedor será el emisor del mensaje y el usuario extractor el receptor, hablando del canal de comunicación; el usuario embebedor cuenta con las acciones de la conexiones y la grabación del portador, el cual a su vez pasa por todo el esquema esteganográfico (posteriormente se explicara esta sección de manera más detallada), mientras que el usuario extractor espera la recepción del estego-mensaje, para poder revelar y dar lectura al mensaje esteganográfico que se encuentra cifrado dentro del mismo.

En cuanto al esquema esteganográfico ver figura 2.5.1, se muestra de manera específica las acciones principales que realizara el programa cuando ya sea implementado, esta muestra el flujo de trabajo correspondiente del programa para cada uno de los usuarios, así como los requisitos del programa para su funcionamiento y el resultado obtenido después de haberse realizado todo el proceso con éxito. Como muestra el diagrama, se inicializa el programa importando

nuestro portador o que es lo mismo, un archivo de audio, el cual se desconoce si contenga o no un mensaje, posteriormente se realizara la decisión por parte del usuario si desea realizar el proceso de empotrar o extraer dependiendo la elección y función del usuario, el programa funcionara para ejecutar cualquier situación. Si la decisión fuera empotrar un mensaje se realizaría la implementación de transformada de Wavelet y posteriormente se obtendrá el estego-mensaje que deseamos, para que el usuario embebedor envíe ese archivo al usuario extractor, mientras que en el proceso de extraer el estego-mensaje se realizara en el proceso de la Transformada de Wavelet a la inversa para obtener el mensaje esteganográfico y con esto el extractor conozca el contenido de dicho mensaje.

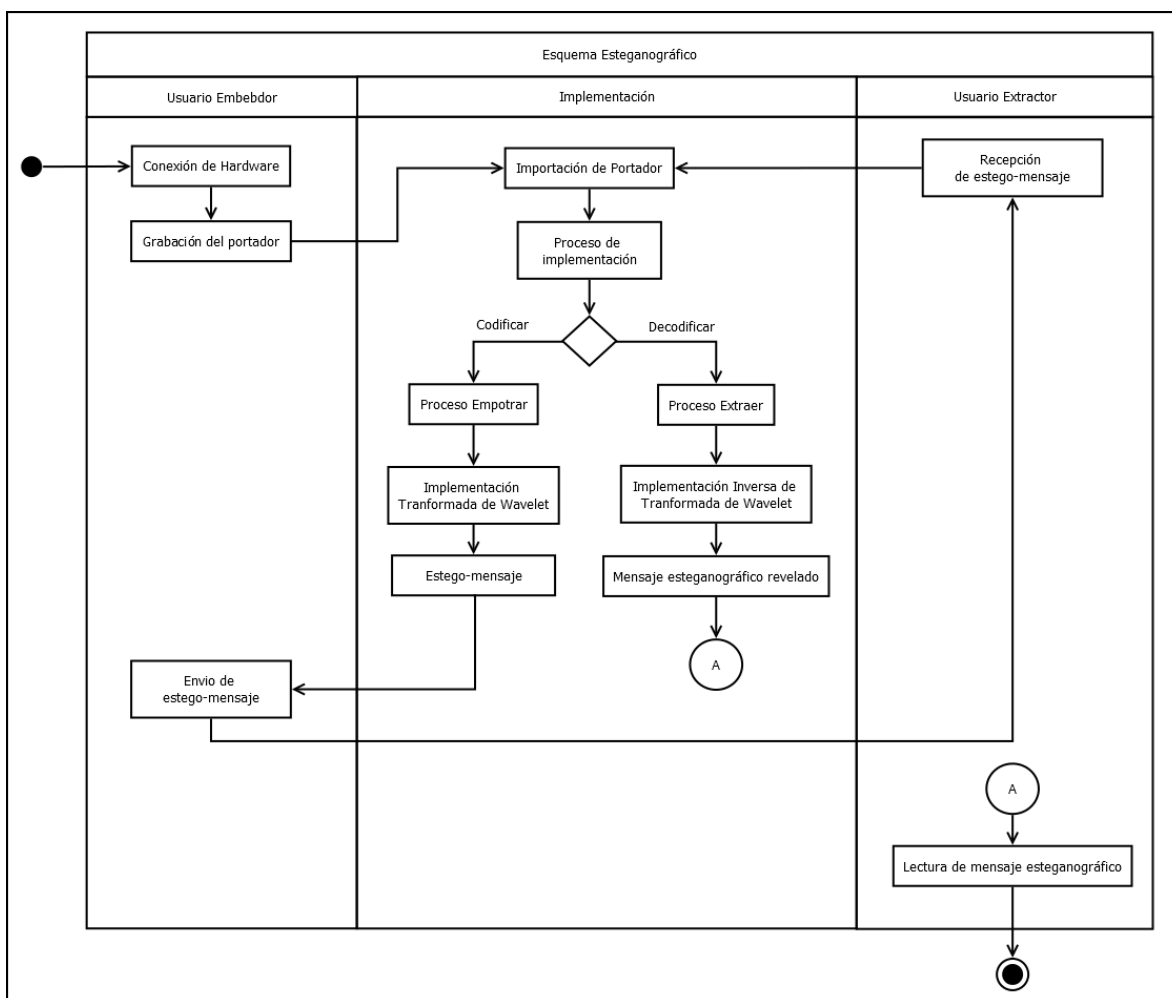


Figura 2.5.1 Diagrama de actividades que representa proceso de comunicación.

2.6 DIAGRAMA DE CASOS DE USO

Para definir las actividades de cada usuario, embebedor o extractor, se realizó el diagrama de casos de uso (figura 2.6.1) en el cual se muestran las acciones que deberán realizar cada uno, así como que actividades pertenecen a cada uno en específico y como son las acciones de empotrar y extraer específicamente. Se hace uso de las actividades en donde el proceso no es explícito en su manera, como en la actividad para empotrar el mensaje dentro del portador, el cual requiere de la grabación y exportación del portador pero que dentro del mismo proceso es esencial para su realización, ya que la finalidad de este diagrama es definir de manera explícita y concreta las acciones y limitaciones de cada usuario. Como se muestra en el diagrama todo el esquema estenográfico contiene todas las actividades por lo que se puede observar dos contenedores, los cuales son el proceso de empotrar y extraer, cada uno contiene las actividades que cada usuario podrá realizar, sea el usuario embebedor o el usuario extractor. Es importante denotar que este diagrama muestra de manera más explícita lo mostrado en los anteriores diagramas, ya que los anteriores diagramas mostraron el flujo completo del proceso, así como con la intervención de ambos usuarios, mientras que este diagrama muestra los procesos de cada uno, con esto se diagrama se puede realizar una visualización aún más concreta de las actividades que se deberán realizar para lograr la implementación del sistema estenográfico a software.

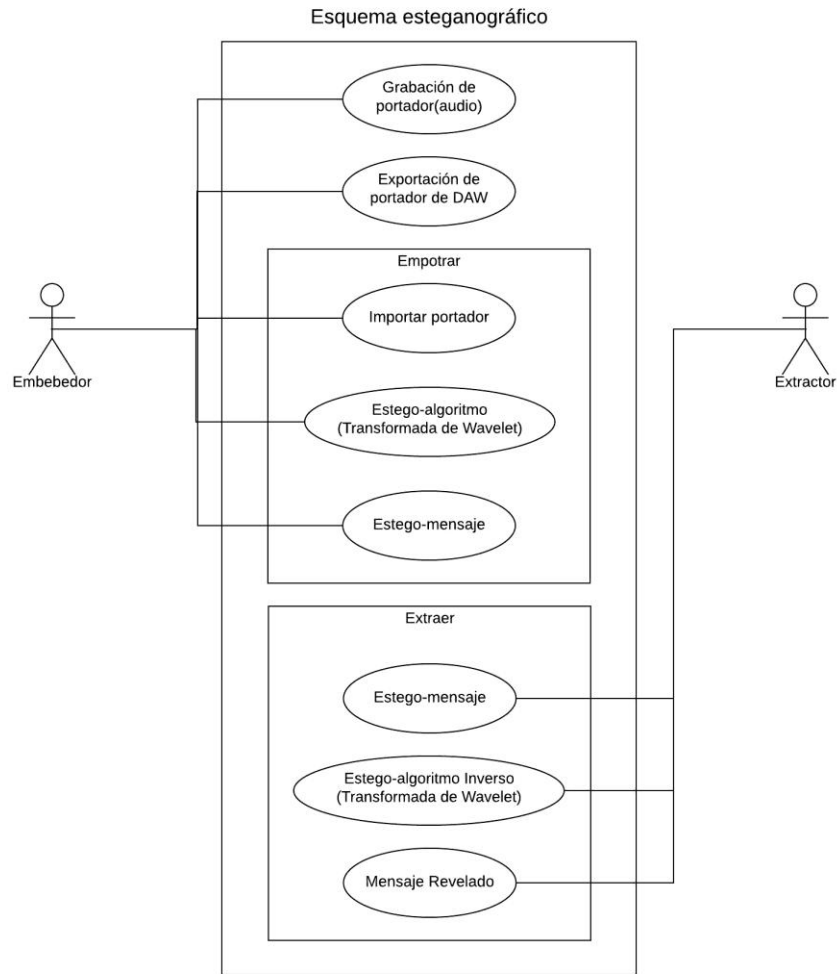


Figura 2.6.1 Diagrama de casos de uso que representa procesos de ambos usuarios.

2.7 MODELADO DE CLASES

La manera de poder emplear todo el esquema esteganográfico en una forma mucho más realista y concreta en cuanto a nivel de programación es realizando el modelado de clases, debido a que este modelado tiene la principal función de representar las clases que serán ocupadas dentro de la implementación. En este caso, por el mismo esquema esteganográfico, se representó como una clase el proceso de grabación del portador como la clase Grabación, para con esto mantener la idea principal del esquema mismo, esta clase lo único que representa es el proceso dentro del esquema esteganográfico, ya que todo el proceso se ha

explicado de manera más concreta en secciones anteriores. Mientras las otras clases representadas en este diagrama son clases que representan a un nivel más detallado las variables con las que contarán cada uno de ellos, así como, las funciones o métodos con las que contarán en la implementación.

La clase Implementación, será la clase que manejará todo el proceso de la implementación, como su nombre indica, por eso su único atributo será el de portador, ya que esta clase tendrá como función primordial el importar el audio a la implementación del programa, por medio de la función *import_audio()*, y después realizar la decisión del siguiente paso a ejecutar, por lo que podrá realizar la acción empotrar o extraer, dependiendo en función del usuario.

Las clases codificadores/decodificadoras serán independientes en cada una de ellas, la única clase en común que tendrán será la clase Implementación, debido a que ella será la clase que le permitirá tener acceso y uso del portador. Por lo que la clase Empotrar tendrá como variable extra *estego_msj* dentro de sus atributos, ya que esta será el resultado de la función *transformada()*, debido a que en esta función se realizará toda la implementación de la Transformada de Wavelet. Mientras que la clase Extraer cuenta con la variable *msj_esteg* dentro de sus atributos y con la función *transformadaINV()* que se encargara de realizar el proceso de extraer el mensaje esteganográfico realizado por el proceso de empotrar, por eso, la variable dentro de esta clase (*msj_esteg*), será la variable donde se almacenara el mensaje guardado dentro del estego-mensaje.

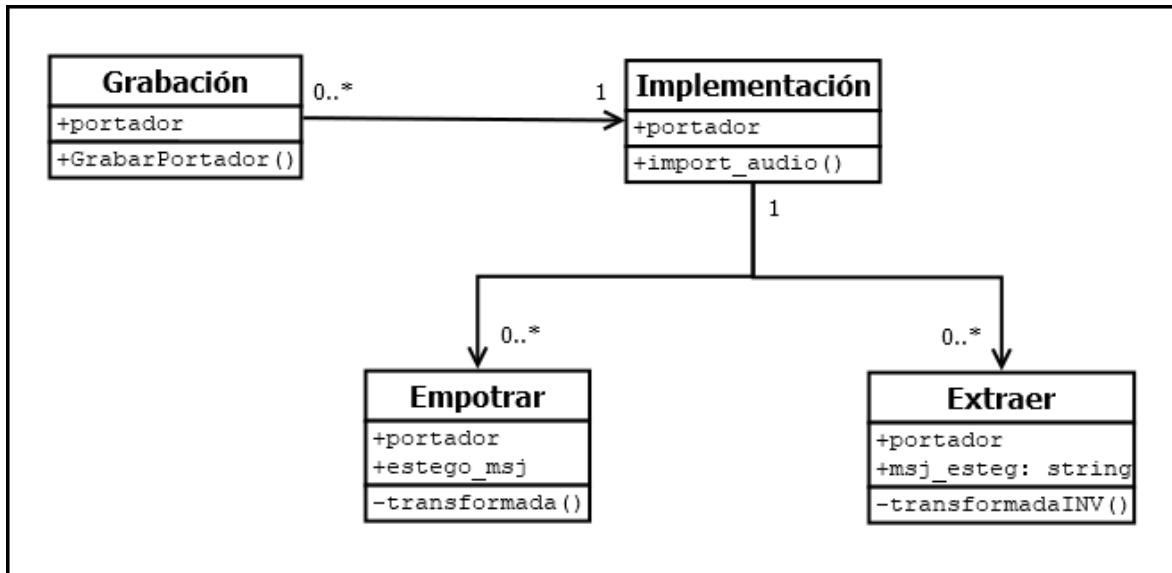


Figura 2.7.1 Modelado de clases para la implementación en programación.

2.8 DIAGRAMA DE FLUJO IMPLEMENTACIÓN (EMPOTRAR/EXTRAER)

Para lograr realizar la implementación por completo, será necesario especificar el procedimiento dos factores primordiales para generar todo el esquema esteganográfico, los procesos embeber y extraer, los cuales contendrán la herramienta matemática la transformada de Wavelet y su inversa, por lo que la figura (2.8.1) muestra cada diagrama con ese título cada uno, para con esto, hacer referencia a ese proceso en concreto explicado en los anteriores diagramas, por lo que este diagrama es un complemento hacia el resto de los diagramas para explicar de manera más explícita los procesos de embeber y extraer.

El diagrama del proceso transformada de Wavelet muestra las etapas posteriores después haber importado el portador en la implementación, por lo que, se denota la primera etapa dentro de este proceso, la cual, se encargara de detectar las frecuencias que contiene el portador y con esto proceder al siguiente proceso, el cual realizara la asignación entre dichas frecuencias con sus caracteres correspondientes y finalmente obtener el portador modificado, el cual se vuelve automáticamente en el estego-mensaje. Por el otro lado, el diagrama del proceso inverso de transformada de Wavelet, realizara un procedimiento similar pero con

diferencias notables, por lo que este se encargará de realizar el reconocimiento de las frecuencias en lugar de su asignación, debido a que para este proceso será necesario reconocer dichas frecuencias para lograr realizar la relación con sus caracteres correspondientes, para posteriormente concatenar dichos caracteres y finalmente obtener el mensaje esteganográfico enviado por el usuario embebedor.

Cabe destacar, que en el proceso embeber no se presenta ninguna situación en donde se reciba el mensaje esteganográfico dado por el usuario embebedor, esto es, debido a que dependiendo de las notas emitidas por el instrumento se generara el mensaje esteganográfico que el usuario embebedor enviara al usuario extractor, por lo tanto, la manera en que se generarán los mensajes esteganográficos serán de una manera totalmente directa con el portador, esto significa que no se solicitara ningún mensaje esteganográfico dentro de la implementación en el proceso transformada de Wavelet; generando con esto que el único mensaje emitido por la implementación este en el proceso inverso de transformada de wavelet y consiguiendo que el mensaje esteganográfico sea de solo lectura para el usuario extractor.

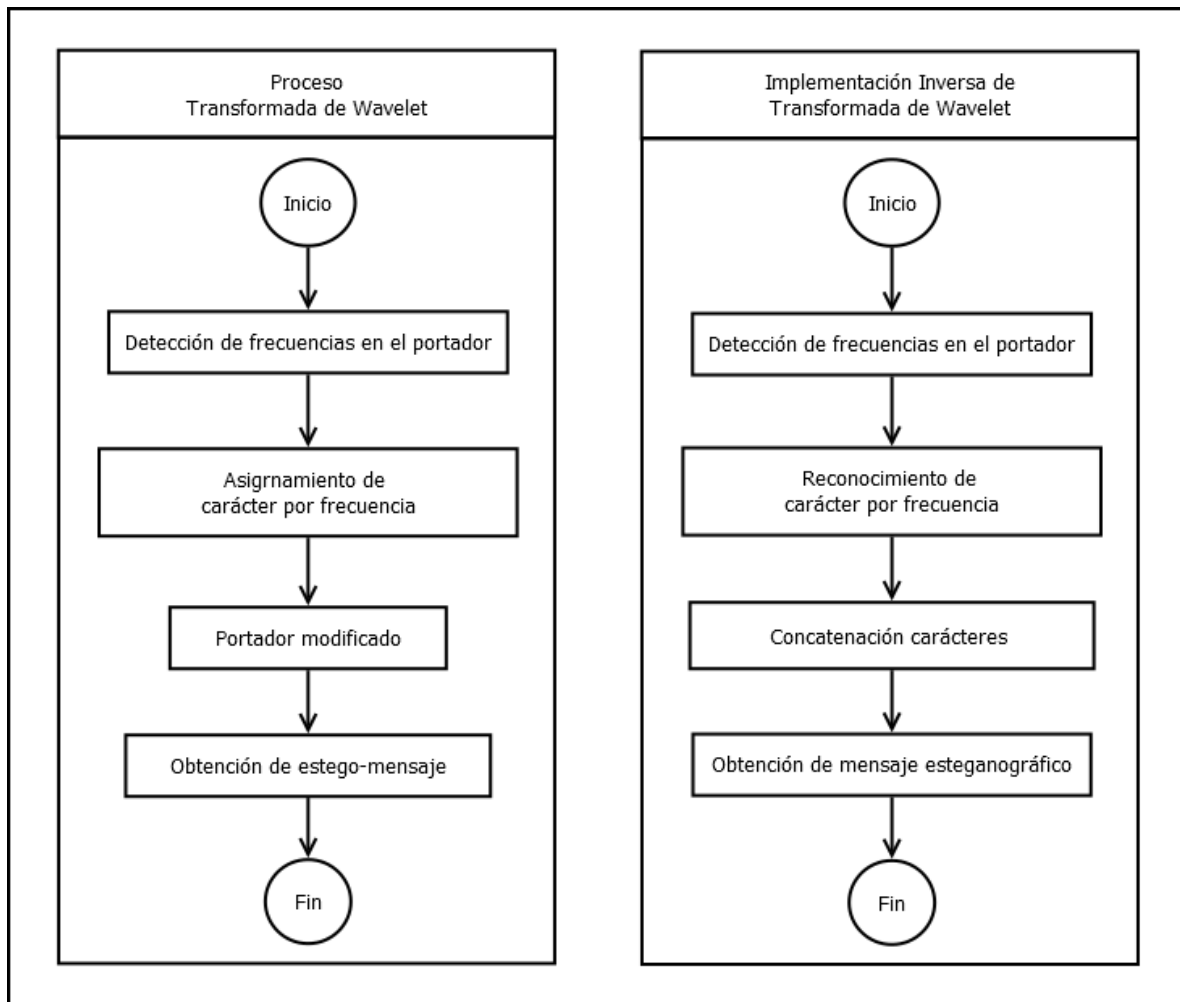


Figura 2.8.1 Diagrama del proceso de transformada de Wavelet(derecha) y diagrama del proceso inverso de transformada de Wavelet.

CAPÍTULO 3

IMPLEMENTACIÓN

3.1 GRABACIÓN DE ARCHIVO DE AUDIO

En el anterior capítulo se mostró la conexión para poder grabar el sonido emitido por la guitarra y mandarlo hacia la computadora, siendo este un proceso de conversión de sonido de analógico a digital (figura 2.3.1) por medio de una interfaz de audio. Recapitulando dicha conexión, el siguiente paso es grabar los sonidos o notas emitidas por la guitarra a través del software DAW *Reaper*, el cual es usado para grabación y producción de audio a nivel profesional. Para poder realizar la grabación es necesario tener diferentes aspectos en cuenta acerca de este tipo de software, debido a que se realizaron ciertas configuraciones dentro del mismo para lograr una correcta grabación e importación del archivo de audio, a su vez, se tiene que realizar la instalación de una herramienta llamada *ASIO4ALL*, la cual se encargara de hacer el enlace entre software y hardware, por medio del software DAW *Reaper* con el hardware de la interfaz de audio. Una vez instalado la herramienta *ASIO4ALL* y conectada la interfaz de audio a la computadora, iniciaremos *Reaper* para posteriormente acceder a *Options* y seleccionar la opción *Preferences*, paso seguido aparecerá una nueva ventana con distintas partes de la configuración (figura 3.1.1), dentro del apartado *Audio* se encuentra sección con nombre *Device* en donde tendremos que realizar la configuración para que el DAW *Reaper* reconozca el dispositivo de entrada, en la sección *Audio System* se debe seleccionar *ASIO*, por lo que las opciones posteriores cambiarán para ese tipo de configuración, una vez realizada esta acción el DAW *Reaper* tendrá seleccionado su canal de entrada y salida, en caso de fallos, cambios dentro del software u más especificaciones concretas es necesario recurrir al manual oficial o guía de usuario de cada herramienta.

ASIO4ALL

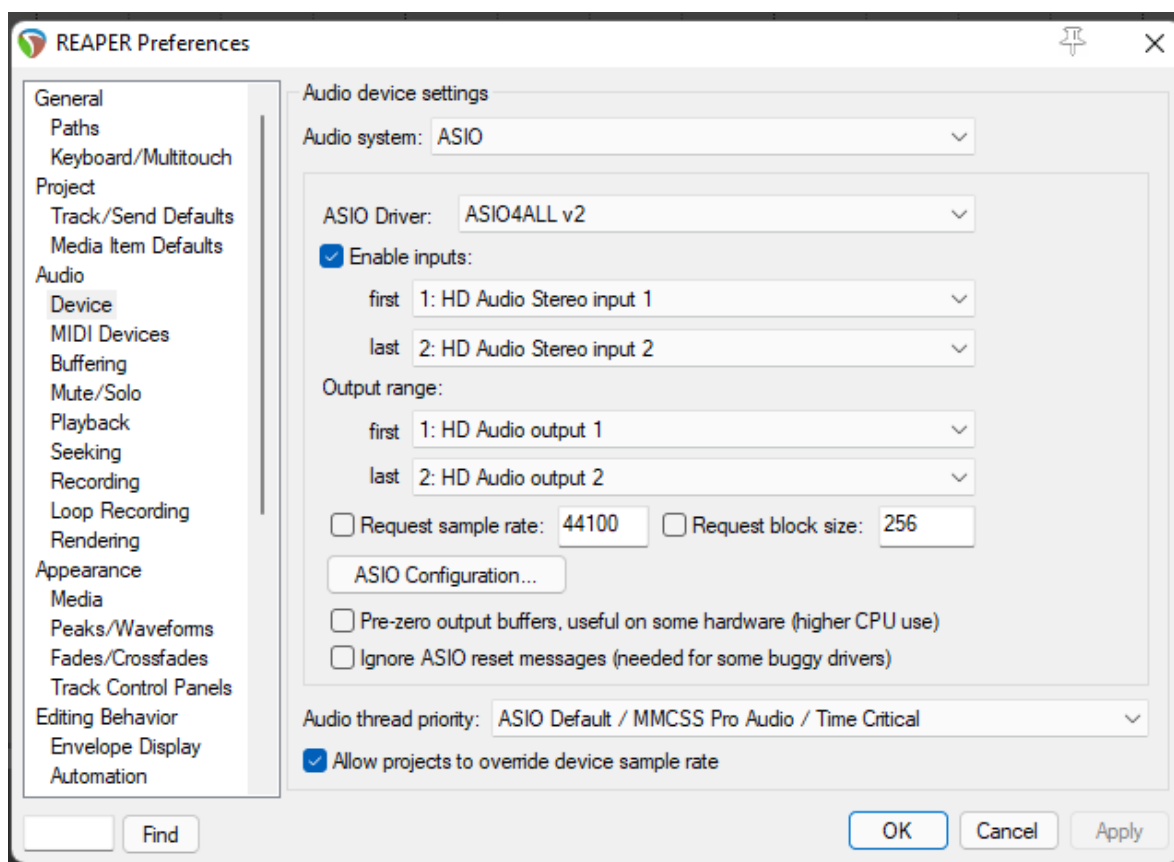


Figura 3.1.1 Ventana de *Preference* en *Reaper* para seleccionar dispositivo de entrada y salida.

Dentro del DAW *Reaper* es necesario crear un nuevo *Track* para después poder comenzar la grabación del instrumento; pulsando doble clic en la parte izquierda se generará un nuevo *Track* y después solo será necesario activarlo por medio del botón de grabación el cual se encuentra en la parte superior izquierda dentro del mismo *Track*, con el cursor encima de el botón aparecerá *Record Arm/Disarm*.

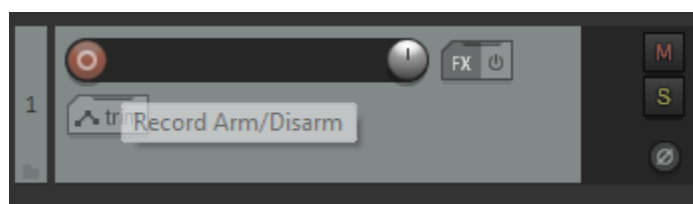
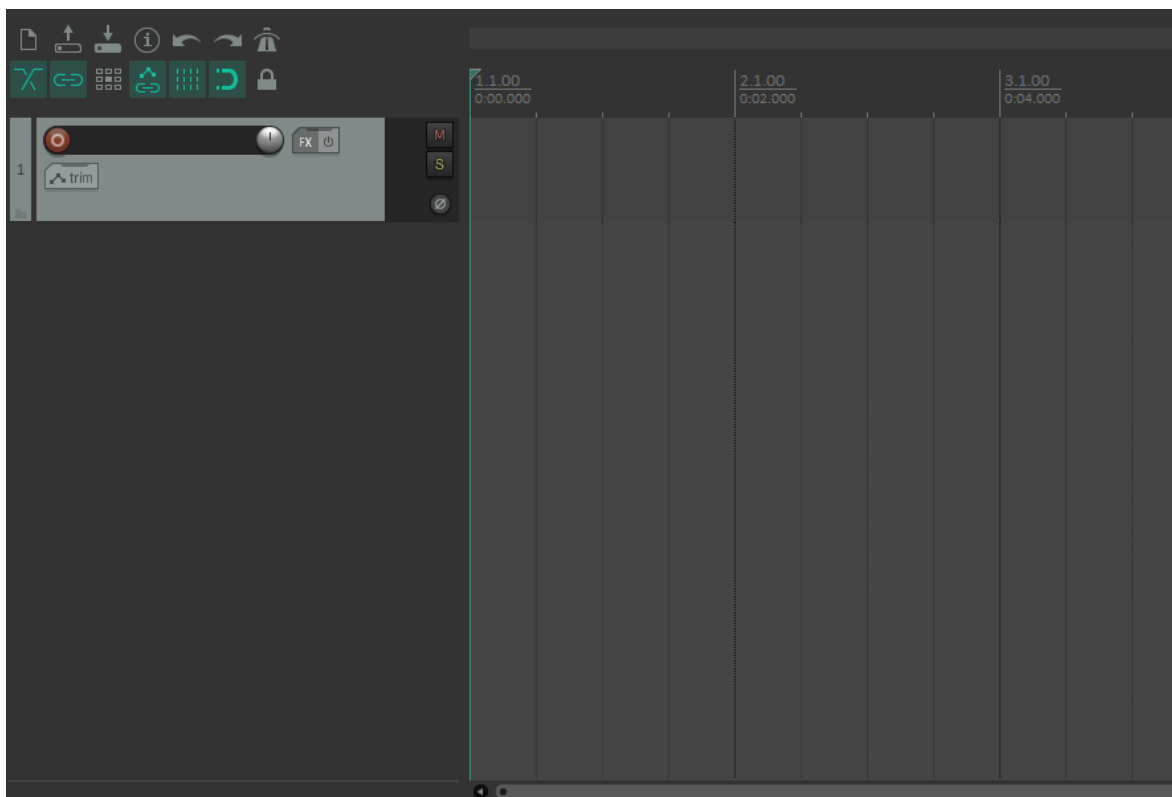


Figura 3.1.2 Ventana de *Preference* en *Reaper* para seleccionar dispositivo de entrada y salida.

Una vez activado el botón podremos escuchar el sonido del instrumento por medio de la interfaz de audio, por lo tanto, si la conexión realizada es correcta junto con la configuración del *DAW*, podremos proceder a la grabación del instrumento por medio del canal seleccionado. Para empezar la grabación será necesario presionar un botón similar al del *Track* pero que en esta ocasión pertenecerá al *Mixer* del *DAW*, con esto empezará a grabar todos los sonidos que se emitan del instrumento hasta que se vuelva a presionar el mismo botón para finalizar la grabación.

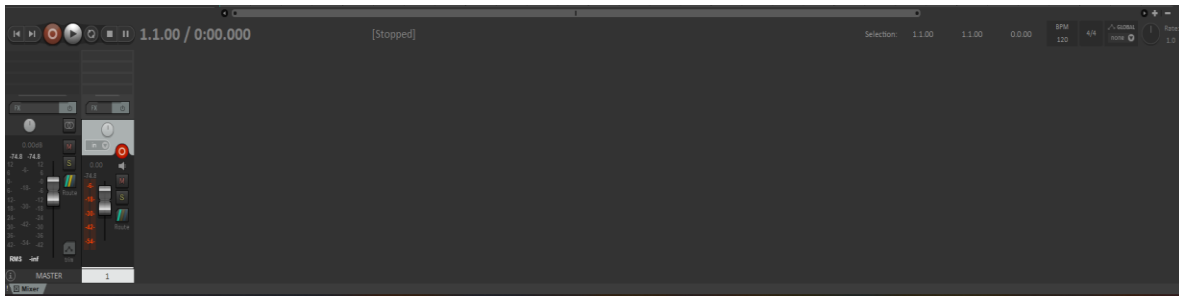


Figura 3.1.3 Componente *Mixer* donde se iniciará o finalizará la grabación.

Una vez realizada la grabación, será necesario exportar el archivo de audio, en este proceso se realizan ciertas configuraciones las cuales ayudaran al proceso de codificación y decodificación dentro del proyecto. Por lo tanto, el archivo de audio exportado en formato .wav tendrá un proceso específico, en este caso, con el *DAW Reaper* es necesario realizar la renderización del archivo de audio para lograr su exportación, en la parte superior izquierda, en las opciones del programa, se seleccionara *File* y siguiente *Render* para posteriormente acceder a la opción *Render* (figura 3.1.4), en este proceso se dará el lugar de almacenamiento del archivo y su nombre, el cual tendrá por nombre Guitarra, pero a su vez, se requiere seleccionar que el archivo de audio este en modo *Stereo* y seleccionar que el archivo tendrá una salida de *16-bits*, estas opciones son imprescindibles, debido a que la mayoría de librerías para lectura de archivos de audio dentro de Python son de 16-bits u 8-bits y la opción de Stereo es la que nos permitirá manipular el archivo de audio cuando se requiera realizar la decodificación, más adelante se detallara cómo funciona este proceso, básicamente el archivo de audio se encuentra con dos canales, los cuales son los que proporcionan sonido para el canal del lado derecho o para el canal del lado izquierdo.

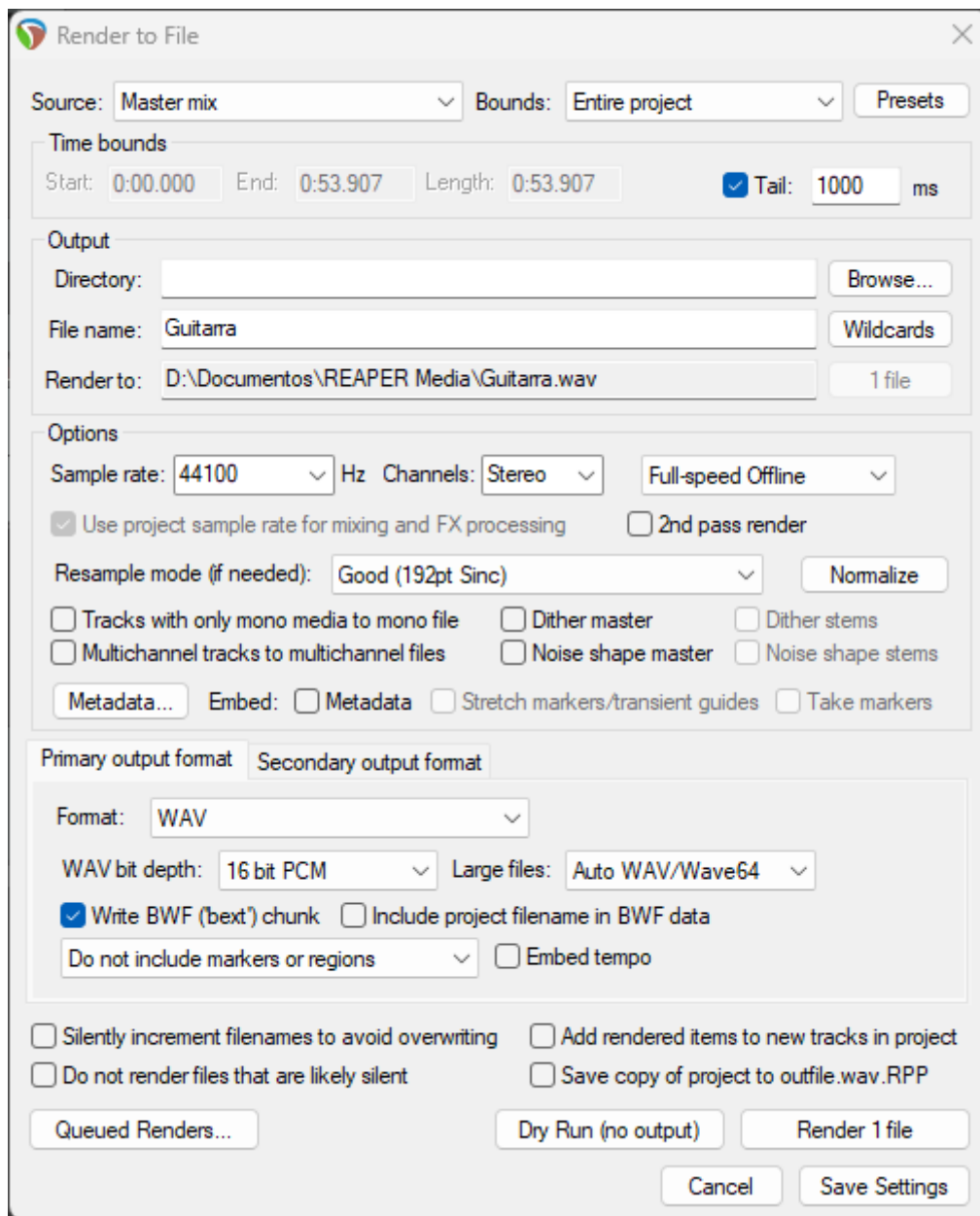


Figura 3.1.4 Ventana de renderización para exportación del audio emitido por el instrumento.

En caso de ser requerido, el software *Reaper* proporciona ciertas herramientas de edición de audio, de ser necesario, podría ser modificado para lograr que las notas sean un poco más limpias o eliminar errores dentro de la grabación en caso de que existieran.

3.2 Transformada de Wavelet general y manera de uso

Dada la explicación del capítulo anterior donde se mencionan todas las características del audio que se renderizó y exportó, en este capítulo se detallará como se hace uso de dicho archivo de audio junto con sus características y el funcionamiento general de la Transformada de Wavelet para este proyecto. Retomando información anterior, el archivo de audio se exportó de manera *Stereo*, esto significa que el archivo de audio tendrá dos canales de salida de sonido, de manera concreta, es que existirá sonido tanto del lado izquierdo como del derecho, todo debido a la Transformada de Wavelet y la manera en la que trabaja, con dicha consideración se explicará cómo se usó la Transformada de Wavelet y como se reparte por los canales del archivo de audio en modo *Stereo*.

Cuando se aplica la Transformada de Wavelet con algunas de las funciones, como por ejemplo Haar, se obtienen los coeficientes de aproximación (cA) y los coeficientes de detalle (cD) de acuerdo con la señal con la que se esté trabajando (figura 3.1.4), en este caso el audio que será la señal, por el momento solo se mencionarán estos dos componentes (cA y cD). Lo que significa que se obtendrán dos señales diferentes o en el caso de este proyecto a manera computacional, dos arreglos, en uno se encuentran los coeficientes de aproximación (cA) y en el otro se encuentran los coeficientes de detalle (cD). Una vez definido esto, para emplear la transformada de Wavelet con el audio en modo *Stereo* se obtendrán la señal de ambos canales, seleccionar alguno de los canales, sea izquierdo o derecho, es indiferente en la parte de la codificación, siempre y cuando no se manipule de distinta manera el audio y se altere alguno de los dos canales, por lo que, obtendremos los mismos datos tanto en el canal derecho e izquierdo que contienen el audio para posteriormente obtener sus coeficientes de aproximación y sus coeficientes de detalle por medio de la Transformada de Wavelet, sin tener en consideración aún los cambios que las funciones de la Transformada de Wavelet le pueden generar al mismo, como el caso de la función de Haar o Daubechies. Una vez obtenidos los coeficientes, aplicando la Transformada de Wavelet, en manera de arreglos será necesario realizar la unión de estos arreglos dentro de uno mismo

para poder generar otra vez un archivo de audio, el cual será totalmente distinto al audio original, en este caso se tiene que realizar la elección de en qué canal se decide poner los coeficientes de aproximación y los coeficientes de detalle, sea coeficientes de aproximación en canal derecho y coeficientes de detalle en canal izquierdo o viceversa. Una vez realizada dicha decisión se podrá importar un nuevo archivo de audio con la Transformada de Wavelet aplicada en el mismo.

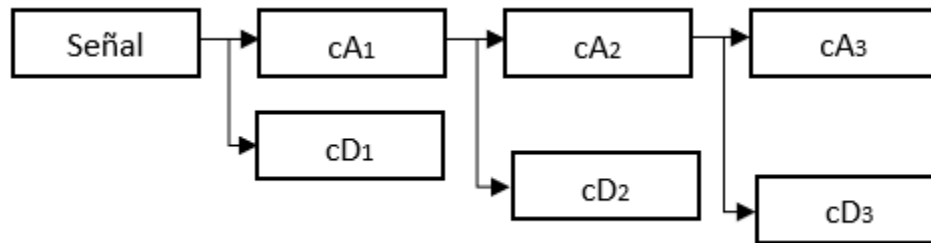


Figura 3.2.1 Representación gráfica de la Transformada de Wavelet y obtención de coeficientes.

Para el momento de la decodificación, se deberá tener especial atención en que canal se realizó cada asignación de los coeficientes, debido a que para realizar la inversa de la Transformada de Wavelet es necesario reconocer cuales son los valores de los coeficientes de aproximación y los coeficientes de detalle, ya que como al momento de aplicar la Transformada de Wavelet se generan los coeficientes, para obtener la señal original serán necesarios esos mismos coeficientes. Conociendo esto, la inversa de la Transformada de Wavelet reintegrará los datos del archivo de audio original basándose en los coeficientes, en este caso, saber que arreglo de datos pertenece a cada coeficiente (cA o cD) y poder trabajar con ellos, una vez realizado el proceso obtendremos el audio/señal original, posteriormente, será necesario reconocer la frecuencia de cada una de las notas encontradas dentro del audio e identificar que letra tiene asignada cada nota, contemplado la octava en la que se encuentre para poder mostrar el mensaje que se desea conocer.

3.3 Detección de notas musicales y su uso

Para conocer el mensaje final emitido por las notas musicales encontradas en el audio, se tiene que hacer conocimiento de estas y cómo funcionan en este proyecto de manera computacional. Por lo que, empezando de manera general cualquier sonido es medido por medio de frecuencias en Hercios (Hz) y esto por ende aplica para las notas musicales, cada nota musical tiene un rango de frecuencia en donde se maneja. De esta manera se puede reconocer que nota se está tocando en base a su frecuencia. Pero, para entender un poco más como se realizó este proyecto se detallarán cosas dentro de la teoría musical. La mejor representación de las notas musicales en un instrumento es por medio de un piano, para esto, se deben tener ciertos conceptos en claro, en la parte izquierda del piano el sonido será más grave, mientras, que en la parte derecha del piano el sonido será más agudo. Para poder definir esto musicalmente hay que tener en cuenta que un piano estándar cuenta con 88 teclas, las cuales estarán divididos por octavas, en cada octava se encontrarán las notas musicales Do, Re, Mi, Fa, Sol, La y Si. Para poder empezar a conocer el piano, se realiza desde un punto medio en el piano o una nota media, que se le denomina Do central, el cual se encuentra en la octava 4, por lo que se le denomina que es un Do₄. Teniendo en consideración que la nota Do central se encuentra en la octava 4 se puede empezar a realizar el análisis de las notas por medio de frecuencia, para lograr esto, se utiliza la nota de La₄ debido a que esta nota su frecuencia es de 440 Hz, lo que significa que se encuentra a nueve y medios semitonos del Do central, un semitono es el espacio que hay entre tecla y tecla en el piano, lo que se comprueba de la manera matemática de la siguiente forma, que $Do_4 = 440 * 2^{-9/12}$ y Do₀ es, lógicamente, cuatro octavas más abajo, por lo tanto, es $2^{-4} = 1/16$ de tono de Do₄. Por lo que, para saber cuál es la nota basándose desde Do₀, es el número de semitonos desde Do₀ hasta la nota. De manera representativa a forma de ecuación, con toda la información se muestre a forma de variables, en su forma en inglés, como semitonos como *h* (*half steps*), nota como *p* (*Pitch*) y Do₀ como C₀ (nomenclatura musical en inglés) (figura 3.3.1).

$$h = 12 \log_2(p/C0)$$

Figura 3.3.1 Ecuación para detección de semitonos dado D_{00}

Una vez obtenida la variable h , se obtiene la mayoría de información que se necesita, pues solo se necesitan dos operaciones más para poder conocer el valor de la nota junto con su octava. Para poder obtener el valor de la octava de la nota correspondiente, basándonos en los semitonos (h) se realizara una operación de división entera, en donde se realizara un redondeo en la parte de la división para obtener un número entero como nuestro cociente, esta operación será el número de semitonos (h) entre 12 para obtener la octavas correspondiente, mientras que para saber cuál es la nota correspondiente se obtendrá el residuo de la misma operación, el cual se definirá dentro de un arreglo con cada una de las notas dependiendo el número obtenido.

3.3.1 Uso de la guitarra eléctrica

Para este proyecto se hace uso del instrumento de la guitarra eléctrica, por lo que la estructura, musicalmente hablando, es diferente. Por ejemplo, que la guitarra eléctrica no cuenta con todas las notas musicales como un piano (visto en el subcapítulo 3.3) y en este episodio se demostrara como trabaja de manera breve el funcionamiento de las notas de la guitarra eléctrica para posteriormente poder explicar el funcionamiento en la creación del mensaje para su codificación y decodificación por medio del código de programación en Python.

Una guitarra tradicional cuenta con seis cuerdas, las cuales son afinadas de una manera armónica para poder generar acordes con la misma, esto quiere decir, que cada nota con la que se afine la guitarra presentara una secuencia diferente de notas, a su vez, dependiendo del grosor de la cuerda se obtendrán notas más graves o agudas, dando paso a las octavas dentro de la guitarra. Por lo tanto, el sistema para generar un mensaje por medio de notas musicales mediante una guitarra eléctrica conlleva el colocar las letras que se ocuparan en cierta nota y corroborar que no se sobre encime con la misma nota, pero en diferente cuerda.

Para poder demostrar esto de manera más grafica se mostrará una tabla (tabla 3.3.1.1) de las notas de un mástil de una guitarra eléctrica, la cual contiene veintidós trastes, tomando en cuenta que por la disposición de las notas en el traste doce se repiten las notas solo que una octava más abajo, los trastes son las secciones en las que se tienen división el brazo de la guitarra que se delimitan por finas líneas de metal, acero o níquel y esto genera que se puedan definir notas entre cada uno de ellos. En la tabla se muestra cada uno de ellos y su respectiva nota con la afinación estándar de la guitarra, el traste 0 significa que la cuerda se toca al aire.

Para poder conceptualizar de manera más sencilla las notas que eran necesarias tocar se hizo uso de colores dentro de la tabla para diferenciar octavas dentro de cada cuerda. La manera en cómo se consiguieron las octavas fue con la ayuda de un afinador dentro del software *DAW Reaper* tocando ciertas notas para conocer su octava y tener referencia así de las otras notas, teniendo en cuenta que las notas se repiten en cada cuerda se realizó una sección donde se asignaran cada nota a una letra para poder así generar una oración y no colocar dos letras en una nota (figura 3.3.1.2), consiguiendo con esto a su vez que se visualice cuantas notas se tienen disponibles para poder emplear en la oración.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
E4	F4	F#4	G4	G#4	A4	A#4	B4	C5	C#5	D5	D#5	E5	F5	F#5	G5	G#5	A5	A#5	B5	C6	C#6	D6	
B3	C4	C#4	D4	D#4	E4	F4	F#4	G4	G#4	A4	A#4	B4	C5	C#5	D5	D#5	E5	F5	F#5	G5	G#5	A5	
G3	G#3	A3	A#3	B3	C4	C#4	D4	D#4	E4	F4	F#4	G4	G#4	A4	A#4	B4	C5	C#5	D5	D#5	E5	F5	
D3	D#3	E3	F3	F#3	G3	G#3	A3	A#3	B3	C4	C#4	D4	D#4	E4	F4	F#4	G4	G#4	A4	A#4	B4	C5	
A2	A#2	B2	C3	C#3	D3	D#3	E3	F3	F#3	G3	G#3	A3	A#3	B3	C4	C#4	D4	D#4	E4	F4	F#4	G4	
E2	F2	F#2	G2	G#2	A2	A#2	B2	C3	C#3	D3	D#3	E3	F3	F#3	G3	G#3	A3	A#3	B3	C4	C#4	D4	

Figura 3.3.1.1 Tabla de las notas de guitarra eléctrica con sus respectivas octavas.

E-2	F-2	F#-2	G-2	G#-2	A-2	A#-2	B-2				
C-3	C#-3	D-3	D#-3	E-3	F-3	F#-3	G-3	G#-3	A-3	A#-3	B-3
C-4	C#-4	D-4	D#-4	E-4	F-4	F#-4	G-4	G#-4	A-4	A#-4	B-4
C-5	C#-5	D-5	D#-5	E-5	F-5	F#-5	G-5	G#-5	A-5	A#-5	B-5
C-6	C#-6	D-6									

Figura 3.3.1.2 Todas las notas disponibles en la guitarra eléctrica con sus respectivas octavas.

3.3.2 Generación de oración

Para realizar esteganografía y ocultar el mensaje esteganográfico dentro de las notas de la guitarra eléctrica o portador (audio), es necesaria una oración que cumpla con una cantidad total de caracteres y otra cantidad de caracteres específicos, siendo estos caracteres específicos los que determinaran la cantidad de notas que se utilizaran, por lo tanto, se realizó la selección de una oración popular que pudiera cumplir con dichos parámetros. La oración seleccionada para este proceso y la cual se estará implementando es *“EL SOL SIEMPRE SALE POR EL ESTE”*, esta oración cuenta con 31 caracteres en total, incluyendo espacios, pero específicamente son 11 caracteres los que se encuentran en la oración, por lo que, da la posibilidad de abarcar 11 notas que seleccionar para poder generar una melodía que interpretada generen esta oración, también se destaca que la oración será solo en mayúsculas con todos los caracteres para no utilizar dos notas para una misma letra y cambie su número ASCII por la diferencia entre mayúscula o minúscula.

Para la generación de un audio (portador) con las notas de la guitarra, tratando de ser un audio con una melodía pequeña, tomando en cuenta que esta tesis busca la esteganografía por medio de notas musicales de una guitarra eléctrica y, por ende, tiene que ser un audio que no genere sospechas al momento de escucharlo, como el caso de ser un audio que no genera una melodía. Por esto, se empleó mínimamente teoría musical para generar dicha melodía, en este proceso se optó por generar el mensaje basándose en acordes, concretamente en acordes de séptima, los cuales fueron Em7, Fmaj7 y Gm7, estos acordes fueron seleccionados por decisión personal.

Por lo que, teniendo en cuenta los anteriores acordes se empezó a asignar letras a las notas de dichos acordes e ir generando dicha melodía, cuidando que no se colocaran dos letras sobre la misma nota. Generando así, que en la tabla donde se encontraban todas las notas de la guitarra eléctrica su fondo sea de color azul de las notas que serán usadas (figura 3.3.2.1).

5	6	7	8	9	10	11	12	13
A-4	A#-4	B-4	C-5	C#-5	D-5	D#-5	E-5	F-5
E-4	F-4	F#-4	G-4	G#-4	A-4	A#-4	B-4	C-5
C-4	C#-4	D-4	D#-4	E-4	F-4	F#-4	G-4	G#-4
G-3	G#-3	A-3	A#-3	B-3	C-4	C#-4	D-4	D#-4
D-3	D#-3	E-3	F-3	F#-3	G-3	G#-3	A-3	A#-3
A-2	A#-2	B-2	C-3	C#-3	D-3	D#-3	E-3	F-3

Figura 3.3.2.1 Todas las notas disponibles en la guitarra eléctrica con sus respectivas octavas.

Una vez teniendo las notas que se emplearan junto con la oración que se desea utilizar, se graba la secuencia de audio que será nuestro portador y se pasa por medio del software para embeber por medio de la transformada de Wavelet al portador el mensaje esteganográfico (visto en capítulo 2). Se asigno un carácter a cada una de las notas, en este caso contando con las 12 notas se optó por utilizar una nota extra para tener un carácter de espacio " " extra, debido a que es un carácter que se repite demasiado en el mensaje esteganográfico, siendo esta la nota C#-4. (figura 3.3.2.1).

5	6	7	8	9	10	11	12	13
A-4	A#-4	B-4	C-5	C#-5	D-5	D#-5	E-5	F-5
E-4	F-4	F#-4 "I"	G-4	G#-4 "O"	A-4 "P"	A#-4	B-4	C-5
C-4	C#-4 " "	D-4	D#-4 "S"	E-4 "R"	F-4	F#-4	G-4	G#-4
G-3	G#-3	A-3	A#-3 "T"	B-3 "E"	C-4	C#-4 " "	D-4	D#-4
D-3	D#-3 "A"	E-3 "L"	F-3	F#-3 "M"	G-3	G#-3	A-3	A#-3
A-2	A#-2	B-2	C-3	C#-3	D-3	D#-3	E-3	F-3

3.4 Embebecimiento y extracción con herramientas y programación

Para realizar la programación de este proyecto se realizó mediante el lenguaje de programación de Python, cuya versión es Python 3.8, el cual tiene apoyo de muchas librerías para poder realizar diferentes trabajos de manera más rápido y eficaz, al

ser usado como por ejemplo para tareas de inteligencia artificial, machine learning o deep learning. La manera de trabajar con Python fue mediante la herramienta de *Anaconda*, la cual proporciona muchas herramientas para el uso del lenguaje de programación de Python, como el caso de *Spyder* como el editor seleccionado para esta tesis, usando la versión de *Spyder* 4.2.5. Como se comentó, existen distintas librerías o bibliotecas de terceros que permiten agilizar ciertos procesos y se ocupan en otros distintos softwares, las librerías que se emplearon son las siguientes:

- Numpy
 - Esta librería es muy usada cuando se utiliza Python, ya que esta librería otorga ciertas herramientas computacionales como es el uso de arreglos y la manipulación de estos, para esta tesis su uso fue para poder utilizar arreglos.
- Wavio
 - Es un módulo de Python el cual realiza lectura y escritura de archivos WAV usando arreglos de numpy.
- IPython,display
 - Este módulo ayuda a crear un objeto de audio, usando sus parámetros nos devuelve un objeto de audio que se generó por medio de arreglos numpy, listas o unicodes.
- Matplotlib
 - Ayuda a desplegar las gráficas de los audios generados después de aplicar la Transformada de Wavelet.
- Crepe
 - Para este trabajo logra detectar el tono que se generó por medio de las notas, al ser un rastreador monofónico que utiliza redes neuronales para su uso, por lo tanto, es dependiente tensorflow un módulo de aprendizaje automático para Python.
- Pywt
 - Iniciales de Python Wavelet Transform, esta librería es usada para emplear la Transformada de Wavelet dentro del proyecto.
- Math

- Este módulo proporciona funciones matemáticas, para este caso se hizo uso en algunas ecuaciones donde eran necesarias operaciones matemáticas.
- Collections
 - El módulo proporciona alternativas a tipos de datos y estructuras de datos de las que tiene Python, como son los diccionarios y las listas.

Una vez explicadas las librerías que se utilizaron para esta tesis, se mencionara el proceso que llevo el código para su implementación, tanto del uso de la Transformada de Wavelet como de su Inversa.

3.4.1 Funcionamiento de código para la Transformada de Wavelet aplicada al audio

La estructura básica para hacer uso de la Transformada de Wavelet y generar un archivo de audio modificado con sus respectivos coeficientes para su posterior inversión y obtener el mensaje dentro del portador es la siguiente:

- Importación de archivo de Audio
- Obtención de audio con Crepe
- Filtro de audio para agregar solo frecuencias concretas
- Filtrado de audio para no tener frecuencias repetidas
- Eliminación de frecuencias que no sean iguales, pero se encuentren dentro de un mismo rango de una nota musical
- Obtención de frase dependiendo la letra que se encontró después del filtrado de las frecuencias
- Se obtienen los canales derecho e izquierdo
- Aplicación de transformada de Wavelet y obtención de coeficientes
- Unión de coeficientes para la generación de un nuevo archivo de audio
- Generación y exportación de un archivo de audio

De todo el proceso mencionado anteriormente se detallarán algunos de los puntos para profundizar mejor en ellos. El primer punto por explicar será la obtención de

audio con Crepe (figura 3.4.1.1), en este caso, al llamar a la función nos retorna los valores del tiempo, frecuencia y confianza, adicional se puede obtener su matriz de activación para poder graficarla. Estos valores están relacionados, ya que el tiempo que regrese el milisegundo en donde se encuentre esa frecuencia y la confianza es la exactitud de que esa frecuencia sea o no correcta. Debido a que es muy alta la cantidad de información que se puede tener cada segundo, para no obtener información extra ni repetida, en lugar de que Crepe haga una evaluación del audio cada 10 ms., valor por default, se le asignó que haga una evaluación cada 30 ms., los cuales nos ayuda a obtener información correcta y concisa de las notas.

```
time, frequency, confidence, activation = crepe.predict(wav.data, wav.rate, viterbi=True, step_size=30)
```

Figura 3.4.1.1 Función predict de la librería Crepe para evaluación de Hercios.

Después de haber obtenido estos valores, se realiza un análisis de la confianza de dichas frecuencias obtenidas, el cual, para mantener un valor correcto, se obtienen solo las frecuencias que tienen un 92% de confianza, con esto, los datos que tenemos se reducen aún más y podemos definir que si es la frecuencia que estamos buscando, después de esto, también se eliminan las frecuencias repetidas, por lo que los datos que tenemos son muchos menos para finalmente como último proceso de filtración, se eliminan las frecuencias que ya se hayan encontrado y estén dentro de un rango de ± 8 Hercios, por lo que, si encontramos una frecuencia como en el caso de La₄ que su valor es de 144 Hz, si se encuentra otra frecuencia cuyo valor sea 149 Hz, no se añadirá dentro de los datos a inspeccionar.

Todo este filtro ayudara para que queden solo las frecuencias que fueron emitidas en el audio y así, poder conocer las notas que fueron tocadas en la guitarra eléctrica, las cuales, dependiendo de la nota tocada se obtendrá la letra asignada a cada una de ellas para poder así generar la frase que se describió en el capítulo 3.3.2 “*EL SOL SIEMPRE SALE POR EL ESTE*”.

La manera de obtener las notas musicales en el código fue realizando la fórmula que se mencionó en la sección *Detección de notas musicales y su uso* (3.3), el cual, después de obtener la nota y la octava en la que se encuentra dicha nota musical

se realiza la asignación de la letra de la oración correspondiente mediante un switch con el que se podrá formar la frase mencionada previamente (figura 3.4.1.2).

```
A4 = 440
C0 = A4 * pow(2, -4.75)
name = ["C", "C#", "D", "D#", "E", "F", "F#", "G", "G#", "A", "A#", "B"]

h = round(12 * log2(p1[i] / C0))
octave = h // 12
n = h % 12
note = name[n] + str(octave)
print("note: "+note)
print()
if note == 'B3':
    val = "E"
    print("E")
if note == 'E3':
    val = "L"
    print("L")
if note == 'D#3':
    val = "A"
    print("A")
if note == 'F#3':
    val = "M"
    print("M")
-----
if note == 'A#3':
    val = "T"
    print("T")
if note == 'D#4':
    val = "S"
    print("S")
if note == 'G#4':
    val = "O"
    print("O")
if note == 'F#4':
    val = "I"
    print("I")
if note == 'E4':
    val = "R"
    print("R")
if note == 'A4':
    val = "P"
    print("P")
if note == 'C#4':
    val = " "
    print(" ")
print(p1[i])
frase.append(val)
```

Figura 3.4.1.2 Asignación de letra para oración dependiendo de nota musical.

Para la poder generar el portador, se realiza la Transformada de Wavelet por medio de la librería de PyWavelets la cual será necesario mandar por parámetros alguno de los dos canales (derecho o izquierdo) para que realice la transformación (figura 3.4.1.3), en el caso de esta tesis se hará uso del canal izquierdo y también se deberá añadir alguna de las familias de Wavelet, en este caso se ocupara la familia de Daubechies y nos retornara ambos coeficientes respectivamente.

```
cA, cD = pywt.dwt(left_channel, 'db1')
```

Figura 3.4.1.3 Función de PyWavelets para obtenciones de coeficientes.

Por lo tanto, la manera en cómo funciona la inversa de la Transformada de Wavelet para obtener dicha información después de que los valores de las frecuencias del portador fueran cambiadas por el uso de la transformada de Wavelet siguen las siguientes fases:

- Importación de archivo de Audio
- Aplicación de inversa de transformada de Wavelet y obtención de coeficientes
- Obtención de audio con Crepe
- Filtro de audio para agregar solo frecuencias concretas
- Filtrado de audio para no tener frecuencias repetidas
- Eliminación de frecuencias que no sean iguales, pero se encuentren dentro de un mismo rango de una nota musical
- Obtención de frase dependiendo la letra que se encontró después del filtrado de las frecuencias

Lo más relevante en este cambio es que a diferencia del proceso en cuando se aplica la transformada de Wavelet al final del proceso, en este caso, se realiza al principio para poder obtener así, el audio original con el que se trabajó en primer lugar para poder obtener exitosamente la frase *“EL SOL SIEMPRE SALE POR EL ESTE”*.

Las frecuencias obtenidas sin ningún tratamiento se muestran de una manera uniforme de ambos lados (figura 3.4.1.4 y figura 3.4.1.5), mientras que las

frecuencias tratadas con la transformada de Wavelet representan una variación en su frecuencia (figura 3.4.1.6 y figura 3.4.1.7).

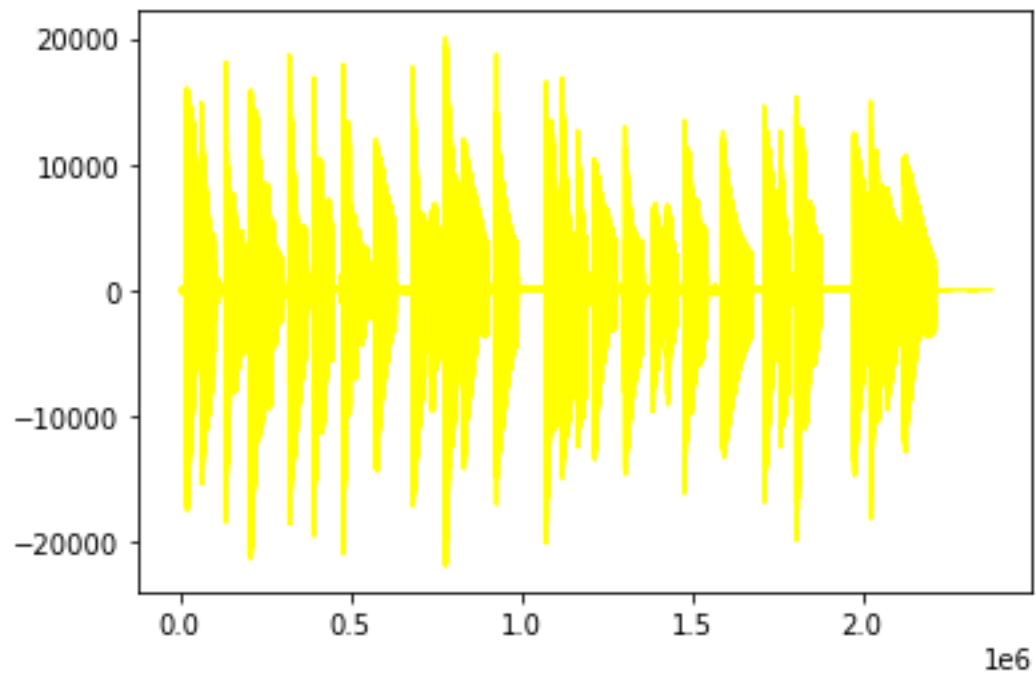


Figura 3.4.1.4 Frecuencias de canal derecho sin ningún tratamiento.

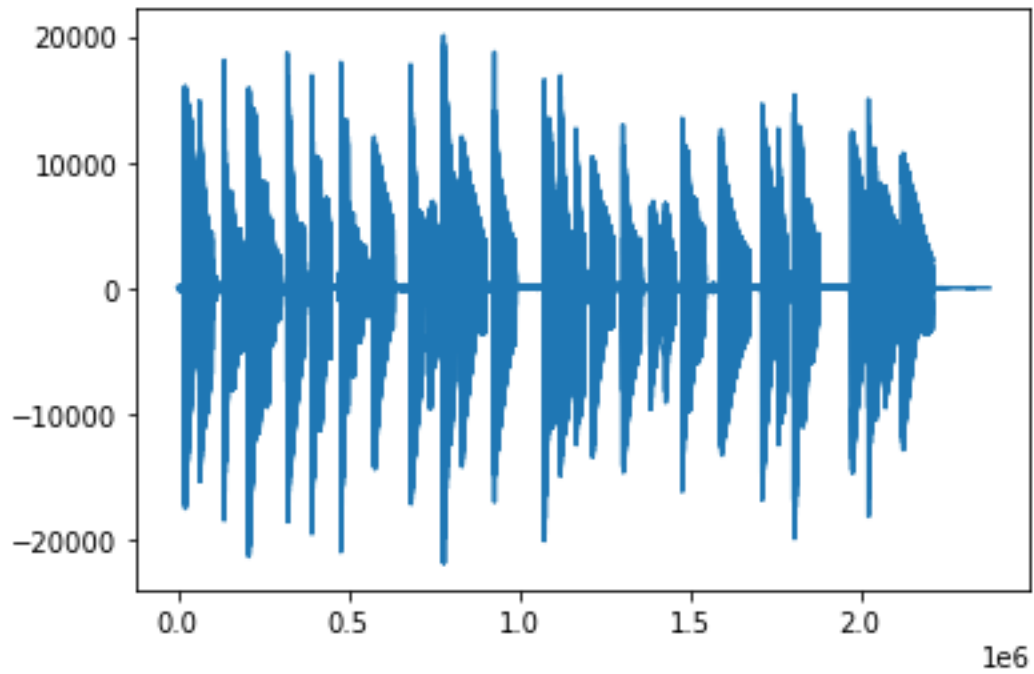


Figura 3.4.1.5 Frecuencias de canal izquierdo sin ningún tratamiento.

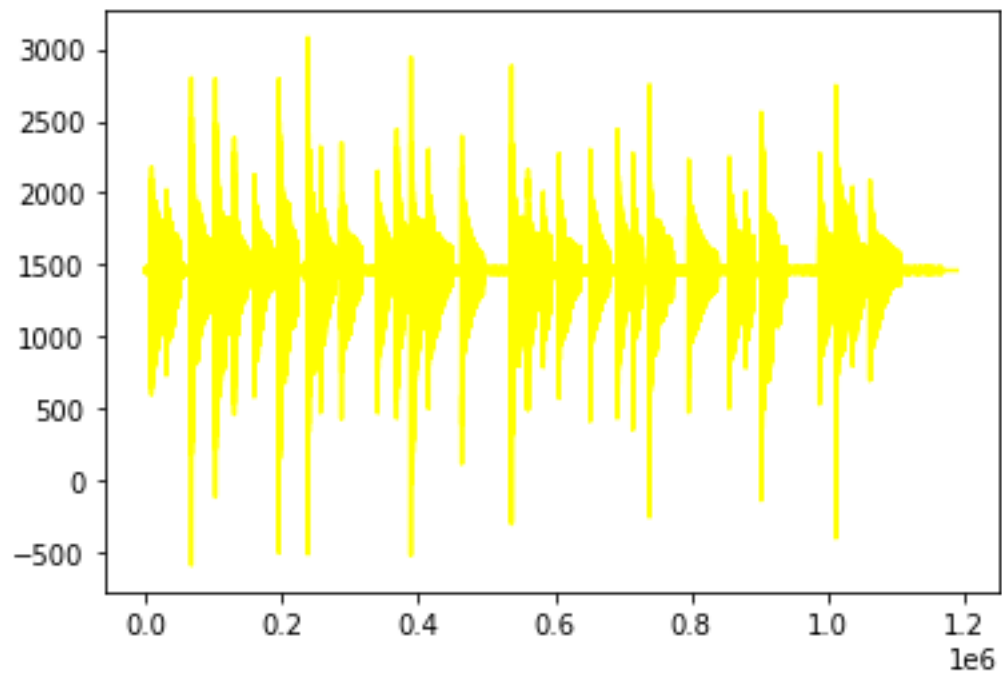


Figura 3.4.1.6 Frecuencias de canal izquierdo siendo portador.

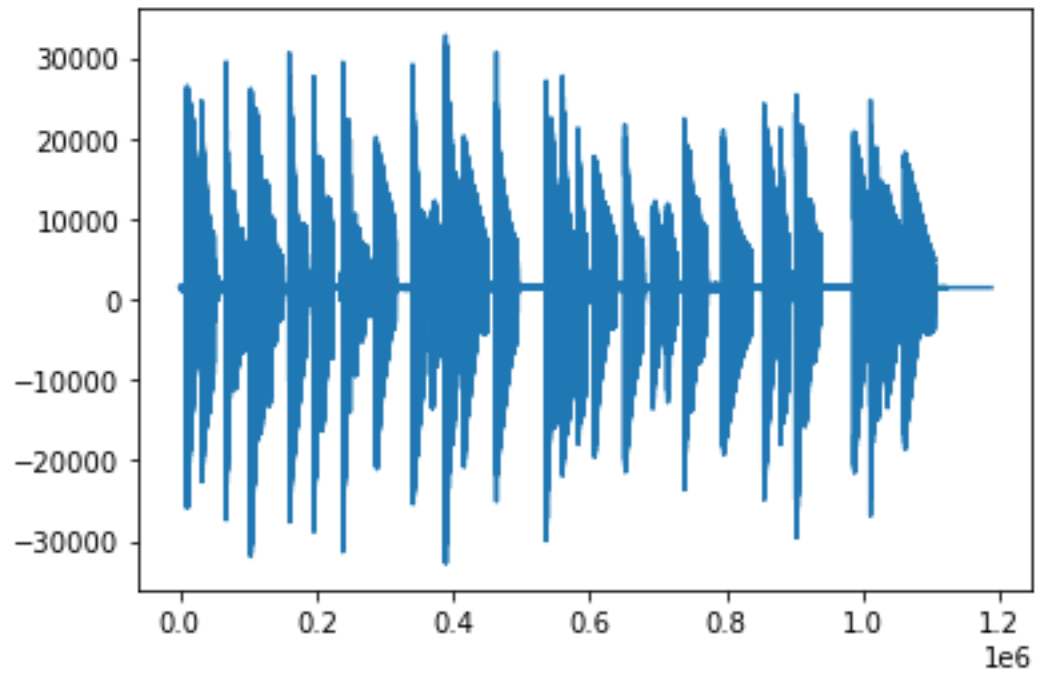


Figura 3.4.1.7 Frecuencias de canal derecho siendo portador.

CAPÍTULO 4

Conclusiones

En esta tesis se demuestra que se pueden manipular frecuencias para poder almacenar un mensaje o un conjunto de caracteres dependiendo las mismas, esto puede dar como resultado otra forma de seguridad extra al momento de querer no mostrar información tan concreta y/o directa, así como existen diferentes algoritmos que nos ayudan a encriptar contraseñas, se podría ocupar lo que se planteó esta tesis en dichos procesos, así como, puede ser en un pin de banco o una contraseña, en la cual sea necesario un archivo para poder obtener acceso a información de alta seguridad o generando un código sabiendo las frecuencias para poder tener acceso, claro que esta forma de seguridad requeriría más pasos a los convencionales, pero de ser posible se puede generar una capa extra de seguridad.

También se puede hacer su uso para el envío de mensajes, como lo que se ha mantenido en esta tesis, con esteganografía, los cuales se pueden ocultar en caso de envíos de mensajes de alta confidencialidad.

A su vez, esta tesis representa una manera de solucionarlo, la cual puede ser mejorada o pueden existir mejoras al mismo, como una ejecución mucho más rápida del programa, ya que actualmente se hace de manera lineal y se podría generar de manera asíncrona para poder obtener de manera más rápida el nuevo audio generado. Igual, se tiene presente un solo problema con el código, que es que en este momento no resuelve el problema si encuentra dos frecuencias seguidas, debido a que, al actual análisis de frecuencias, elimina las frecuencias repetidas, pero no hay manera de saber si se tocó dos veces la misma nota, lo cual esto generaría un problema que podría cambiar la estructura o coherencia del mensaje.

Bibliografía

REAPER | User Guide. (s. f.). <https://www.reaper.fm/userguide.php>

The MathWorks. (n.d.). *Introduction to wavelet families*. MATLAB & Simulink. <https://www.mathworks.com/help/wavelet/gs/introduction-to-the-wavelet-families.html>

PyWavelets developers. (n.d.). *PyWavelets — Wavelet transforms in Python*. Read the Docs. <https://pywavelets.readthedocs.io/en/latest/>

IPython developers. (n.d.). *IPython.display — API documentation*. Read the Docs. <https://ipython.readthedocs.io/en/8.16.0/api/generated/IPython.display.html>

Python Software Foundation. (n.d.). *CREPE* [Computer software]. PyPI. <https://pypi.org/project/crepe/>

Weckesser, W. (n.d.). *wavio* [Computer software]. PyPI. <https://pypi.org/project/wavio/>

NumPy developers. (n.d.). *NumPy documentation* (stable version). NumPy. <https://numpy.org/doc/stable/>

Momento Guitarra. (n.d.). *Trastes: ¿Qué son y cómo se llaman?* Momento Guitarra. <https://momentoguitarra.com/que-es/trastes/>

John D. Cook. (2009, 30 septiembre). *Perfect fifths, equal temperament* [Blog post]. JohnDCook.com. <https://www.johndcook.com/blog/2009/09/30/perfect-fifths-equal-temperament/>

John D. Cook. (n.d.). *Convert pitch between Hz and music notation*. John D. Cook. https://www.johndcook.com/frequency_pitch.html

John D. Cook. (2016, February 10). *Musical pitch notation from frequency with Python code*. JohnDCook.com. <https://www.johndcook.com/blog/2016/02/10/musical-pitch-notation/>

Piano key frequencies. (n.d.). In *Wikipedia*. https://en.wikipedia.org/wiki/Piano_key_frequencies

Miles. (2020, October 23). *Turning frequencies into notes in Python* [Online forum post]. Stack Overflow. <https://stackoverflow.com/questions/64505024/turning-frequencies-into-notes-in-python>

Frecuencia de las notas musicales. (n.d.). GuiasDeApoyo.net. https://www.guiasdeapoyo.net/guias/cuart_fis_c/Frecuencia%20de%20las%20notas%20musicales.pdf

El Club del Autodidacta. (2012, August 13). *Cálculo de la frecuencia de nuestras notas musicales.* <https://elclubdelautodidacta.es/wp/2012/08/calculo-de-la-frecuencia-de-nuestras-notas-musicales/>

Gómez, P. (n.d.). *S11-Wav-P2* [Presentación PDF]. INAOEP. <http://ccc.inaoep.mx/~pgomez/cursos/pds/slides/S11-Wav-P2.pdf>

Referencias

- [1] Ribagorda, Arturo; Estévez-Tapiador, Juan M.; Hernández-Castro, Julio C.. (2007). *Esteganografía y esteganálisis: Hitos históricos, casos prácticos, resultados de investigación y aplicaciones a Internet*. Biblioteca de la Universidad Carlos III de Madrid: Instituto Juan Velázquez de Velasco de Investigación en Inteligencia para la Seguridad y la Defensa.
- [2] Stefan Katzenbeisser Fabien A. P. Petitcolas. (2000). *Information Hiding Techniques for Steganography and Digital Watermarking*. Norwood, MA 02062: ARTECH HOUSE, INC..
- [3] Kristie Macrakis. (2014). *Prisoners, Lovers, and Spies: The Story of Invisible Ink from Herodotus to al-Qaeda*. Yale University: Yale University Press.
- [4] David Chaum. (1984). *Advances in Cryptology: Proceedings of Crypto 83*. Boston, MA: Springer.
- [5] Jesús Díaz Vico. (2010). *ESTEGANOGRAFÍA Y ESTEGOANÁLISIS: OCULTACIÓN DE DATOS EN STREAM EN AUDIOS VORBIS*. Universidad Politécnica de Madrid: Facultad de Informática.
- [6] Nedeljko Cvejic. *Algorithms for audio watermarking and steganography*. PhD thesis, Department of Electrical and Information Engineering, University of Oulu, 2009.
- [7] N.A. Saleh, H.N. Boghdady, S.I. Shaheen, and A.M. Darwish. An efficient lossless data hiding technique for palette-based images with capacity optimization. In *Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services. 14th International Workshop on*, pages 241 –244, 27-30 2007.
- [8] Nan iWu andMin shiang Hwang. Data hiding: Current status and key issues abstract, 2007.
- [9] J. Chou, K. Ramchandran, and A. Ortega. Next generation techniques for robust and imperceptible audio data hiding. In *ICASSP '01: Proceedings of the Acoustics, Speech, and Signal Processing, 2001. on IEEE International Conference*, pages 1349–1352, Washington, DC, USA, 2001. IEEE Computer Society.
- [10] Ted Painter and Spanias Andreas. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–513, Abril 2000.

[11] W. Bender, D. Gruhl, N. Morimoto, and Aiguo Lu. Techniques for data hiding. *IBM Syst. J.*, 35(3-4):313–336, 1996.

Glosario

La rama de la esteganografía, como cualquier otra, cuenta con un glosario específico en el cual se definen los distintas fases y parámetros con los cuales se cuentan dentro de todo un proceso esteganográfico, debido a que su denotación es primordial para el uso de toda esta tesis, ya que todas estas definiciones serán utilizadas para la explicación de la misma.

Esquema esteganográfico: Se denota a todo el procedimiento esteganográfico, desde el almacenamiento del mensaje hasta la lectura del mismo, en concreto, la realización de codificación y decodificación.

Portador: Será todo aquel objeto en el cual se transporte el mensaje que se desea ocultar, estos medios de manera digital pueden ser imágenes, videos, texto plano o como en el caso de esta tesis, audio.

Mensaje esteganográfico: Es el mensaje que se desea almacenar dentro del portador.

Estego-algoritmo: Algoritmo esteganográfico que realizara el proceso de unión entre el mensaje esteganográfico y el portador, en este caso, el uso de la Transformada de Wavelet como estego-algoritmo.

Empotrar/Embeber (Codificación): Nombre de la acción de la unión del mensaje esteganográfico con el portador.

Estego-mensaje: Se denota así al mensaje resultante de la acción de empotrar.

Extraer (Decodificación): Nombre de la acción de recuperación del mensaje esteganográfico dentro del estego-mensaje.

Embebedor: Usuario encargado de realizar la acción de empotrar.

Extractor: Usuario encargado de realizar la acción de extraer.

Estegoanalista: Usuario terciario, encargado de revisar si existe un mensaje esteganográfico dentro del portador o estego-mensaje.