

Audio Compression in mp3 format using php

Nicolas Herrera Rubiano, código: 20171020118 - Alvaro Andres Niño Rincon, código: 20171020139
(Universidad Distrital Francisco José de Caldas, Ingeniería de Sistemas, Redes de Comunicación II)
agosto de 2020

Resumen— La compresión de audio es un tema muy importante del campo de la compresión de datos, ya que no se puede trabajar de la misma manera como se trabajan otros datos en la compresión, recordando que el sonido son oscilaciones que se transmiten por un medio que en general es el aire y para poder hacer uso de este sonido, se procede a lo que es el audio el cual simplemente es una adaptación del sonido a otro medio, el paso a un soporte y para esto, a lo largo del tiempo se han creado diversos soportes de audio, como lo son el vinilo, las cintas magnéticas o pasando a la actualidad con el auge de la tecnología a un formato digital; pero entre más calidad se posea más espacio ocupará en el soporte, bien sea más cinta magnética, un vinilo más grande o en el caso de los formatos digitales más espacio en el disco en el que en ocasiones puede generar ciertos problemas y es por eso que es importante también la compresión de audio la cual es un proceso por el cual se reduce la tasa de bits de una señal digital de audio buscando como fin la reducción de su peso en el disco como se detallará en el presente artículo.

Abstract-- Audio compression is a very important topic in the field of data compression, since you cannot work in the same way as you work with other data in compression, remembering that sound is oscillations that are transmitted through a medium that is generally air and in order to make use of this sound, we proceed to what is the audio which is simply an adaptation of the sound to another medium, the passage to a support and for this, over time have created various audio media, such as vinyl, magnetic tapes or passing to the present with the rise of technology to a digital format; but the more quality you have more space in the support, either more magnetic tape, a larger vinyl or in the case of digital formats more space on the disk which can sometimes generate certain problems and that is why it is also important audio compression which is a process by which the bit rate of a digital audio signal is reduced seeking to reduce its weight on the disk as detailed in this article.

Key words— disk, magnetic tapes, vinyl, audio, compression, data, codec, encoding, decoding, algorithm, software, hardware, storage, bits, process, quality, waves, sound, weight.

I. INTRODUCCIÓN

A lo largo de la historia, se ha evidenciado la necesidad del hombre de estar en constante comunicación, para esto uno de los medios por excelencia es el sonido emitido por la vibración en las cuerdas vocales, a lo que poco a poco se le fue dando forma hasta constituir lo que conocemos como idiomas, los cuales sirven entre muchas de sus características y funciones para establecer una comunicación entre las personas asimismo, se fue descubriendo poco a poco que al poner en contacto dos objetos al momento de tocarse generaban un ruido para el que luego se le dieron múltiples usos según el material y quien los

utilizaba, uno de estos es la creación de la música, la cual inicio por medio del sonido provocado por el contacto entre dos objetos. La forma de comunicación verbal entre las personas fue evolucionando a lo largo del tiempo y siendo utilizada no únicamente para la comunicación entre las mismas sino para distintos aspectos de la vida como lo es el hecho de utilizar la voz para entonar canciones.

El sonido en todas sus formas ha estado inmerso en la vida humana y todo lo que la rodea; pero hay un aspecto esencial en esto y es el hecho de que este sonido no se puede preservar, sino que a medida que se desplazan las ondas que transportan este sonido, el mismo se va perdiendo hasta desaparecer. Este problema a aquejado a la humanidad durante muchísimo tiempo, encontrando como única solución el escribir lo que luego de una interpretación se convertirá en sonido, es decir, los cantantes, escribir sus canciones, los oradores, escribir sus mensaje, los músicos que tocan algún instrumento escribir lo que se interpretará en el mismo por eso se crearon las notas musicales para poder plasmar lo que se iba a interpretar y que de algún manera ese sonido permaneciera, aunque no era como tal el sonido sino una forma de replicarlo. No fue sino hasta mucho tiempo después con el avance de la tecnología y la invención de una técnica mediante la cual se pudiera almacenar sonido en el mismo y posteriormente por medio de otro proceso poder escuchar este sonido almacenado como lo es el caso del vinilo y posteriormente con su evolución llegar a las cintas magnéticas y finalmente al almacenamiento digital de audio como lo tenemos hoy en día que se puede guardar audio en dispositivos de almacenamiento digitales (discos) y modernos y permite reproducir este audio almacenado según el dispositivo en el que se encuentre este disco, ya sea un teléfono celular, un computador, un reproductor exclusivo de audio, etc. además, con el hecho de que se puede enfocar el audio solo a los oídos enfocando las ondas que salen del dispositivo por un cable directo a los oídos. Con esto se solucionó el problema de la preservación del audio; pero acarrió otro problema y es que entre mejor sea el audio, es decir, entre más calidad posea va a ocupar más espacio y saturar el disco. Es por esto, que existe en el campo de la informática la compresión de datos para reducir el tamaño de la información en el disco y de esta manera liberar espacio y se puede realizar este proceso de compresión también para los audios por medio de un proceso mediante el cual se reduce la tasa de bits de una señal digital de audio buscando el mismo propósito de la compresión, el cual es reducir el tamaño del elemento en el disco [1].

II. OBJETIVOS

A. Objetivo general

El objetivo general de este trabajo investigativo es conocer a profundidad todo lo concerniente a la compresión de audio, enfatizando en los trabajos realizados sobre el tema y las formas y métodos de realizar este proceso.

B. Objetivos específicos

- Comprender todas las características y componentes de lo que es el audio.
- Estudiar las diferentes formas de comprensión de audio, ya sea únicamente mediante software por medio de algoritmos o con la ayuda de hardware y la obtención de audio por medio de otras fuentes como lo es un micrófono.
- Realizar una búsqueda exhaustiva sobre los trabajos científicos e investigativos sobre la comprensión de audio.

III. COMPRESIÓN DE AUDIO

La compresión de archivos de audio es un proceso mediante el cual se reduce la tasa de bits de una señal digital de audio buscando reducir el tamaño de archivos de audio con la finalidad de ahorro en la transmisión de los archivos o en su almacenamiento simplemente y esta necesidad de comprimir se creó debido al gran auge de la tecnología con sus nuevos e impresionantes dispositivos y la posibilidad de obtención de música de muchas fuentes como la internet y poder tenerla en grandes cantidades lo que ocasionó que el almacenamiento de los dispositivos se llenara y no tuviera espacio para guardar más y en ocasiones causando daños al funcionamiento del equipo en cuestión, por esto llegó la compresión de archivos y esencialmente en archivos de audio, por el hecho de que entre más duración y/o calidad tenga un audio más pesado será y con la compresión de los mismos, se busca solucionar este problema permitiendo almacenar más archivos de audio en el mismo espacio de memoria [2]. En la comprensión de los archivos de audio, se busca disminuir los canales de audio y asimismo el kilobitrate, es decir, la tasa de bits disminuyendo su tamaño; pero con el agravante que se perdería un poco de calidad. Para comprender mejor este proceso interno de compresión de audio, es bueno saber que el sonido está compuesto por ondas de forma curva con infinitos puntos y cuando se pasa al audio digital, se ubica esta onda en unos ejes coordenados que, al comprimir, se reduce la cuadrícula de los ejes coordenados disminuyendo los puntos de la curva lo que se traduce en una menor cantidad de datos lo que conlleva a un menor peso [3].

Para realizar este proceso hay múltiples formas y tipos de compresión los cuales fundamentalmente son dos: algoritmo de compresión sin pérdida y algoritmo de compresión con pérdida, los cuales de estudiarán a continuación:

- **Algoritmo de compresión sin pérdida:** hace referencia a cualquier proceso de codificación que su objetivo sea representar cierta información en un espacio menor que el original; pero con la posibilidad de tener una reconstrucción exacta de los datos originales, es decir,

que no se pierda nada de información en el proceso. Este tipo de compresión sin pérdida tiene como pilar la teoría de la información para cumplir su propósito, específicamente en aspectos como la redundancia, que es el hecho de repetir y la entropía de los datos que se enfoca en medir la incertidumbre de una fuente de información. Para realizar este proceso, se utilizan dos tipos de modelos: el estático que lee y codifica utilizando la probabilidad de aparición de cada carácter, haciendo uso de la codificación de Huffman por medio del árbol del mismo nombre; pero utilizar un modelo estático tiene sus limitaciones. Si un flujo de entrada no concuerda bien con la estadística previamente acumulada, la relación de compresión se degradaría, posiblemente hasta el punto de que el flujo de datos saliente fuese tan largo como el entrante (o incluso más) [5]. Por tanto, la siguiente mejora obvia fue construir una tabla que se construya conforme se recibe el flujo de entrada. El siguiente modelo es el basado en diccionario (busca coincidencias entre el texto a comprimir y un grupo de cadenas de caracteres en una estructura) el cual usa un código simple para reemplazar cadenas de símbolos; los modelos estáticos generalmente codifican un símbolo a la vez. El esquema de compresión basada en diccionario utiliza un concepto diferente. Lee una entrada de datos y observa por grupos de símbolos que aparecen en el diccionario. Si una cadena concuerda, un indicador o índice en el diccionario puede salir en lugar del código del símbolo. Y para finalizar, algunos algoritmos de compresión sin pérdida son: Lempel-Ziv, que incluyen LZ77, LZ78, LZW y para ciertas clases de formatos de audio existe FLAC y Monkey's Audio [6].

- **Algoritmo de compresión con pérdida:** Este algoritmo, hace referencia a cualquier proceso de codificación, es decir, representar cierta información en una menor cantidad de espacio; pero con el agravante de que sería imposible una reconstrucción de los datos a una forma original. Lo que lo diferencia del anterior método, el algoritmo de compresión sin pérdida genera una réplica exacta mientras que el algoritmo de compresión con pérdida genera una aproximación al archivo original. Esta clase de algoritmo es útil en momentos cuando la reconstrucción exacta no es necesaria para que el resultado tenga sentido, es decir, que aún con la pérdida no pierde el sentido del archivo, aunque sacrificando cierta cantidad de datos para lograr una mejor compresión de los archivos.

Existe tipo de compresión de audio el cual no es el más común, este es el Voice over IP, generalmente usado en la telefonía y en la telefonía en la nube, para esto, la calidad de la voz se determina mediante la medida de la puntuación de la opinión o MOS, cuanto mayor sea este valor, mejor será la calidad de las llamadas y en ciertos casos cuando se necesitan anchos de banda notablemente superiores es donde se introduce el concepto de los códec's [4] y su utilización. Un códec de audio como el desglose de su nombre lo indica ("co" hace referencia a codificar y "dec" a decodificar) es un conjunto de algoritmos que permiten codificar y decodificar datos auditivos,

lo que se traduce en reducir la cantidad de bits que ocupa el fichero de audio, sirve para comprimir señales o ficheros de audio con un flujo de datos con la finalidad de reducir al máximo la ocupación de espacio y luego descomprimiéndolos para reproducirlos o manipularlos es un formato apropiado. Se puede implementar un Códec de audio tanto en software como en hardware o inclusive en una combinación de los dos.

Enfocándose en el hardware y con la idea de mostrar un ejemplo, al conectar una fuente analógica en tu computadora, esta se conecta directamente a un códec de audio para transferir el sonido digital a analógico o viceversa en el caso del micrófono, esto se logra generalmente a través de conectar un dispositivo en tu línea de entrada/salida de conectores del sistema. Estos conectores tienen un aspecto similar al tipo que puedes encontrar en un reproductor de CD portátil o un reproductor de MP3, la toma de los auriculares típicos.

Los códecs de audio no son únicos, es decir, no existe un único método de codificar audio sino son varios métodos que dependiendo de la reducción de la tasa de bits se pueden clasificar en:

- **Codificadores perceptuales:** estos codificadores se enfocan en explotar o utilizar a favor las limitaciones en la percepción del sistema auditivo del ser humano, específicamente el umbral de audición, enmascaramiento temporal o frecuencial para codificar el flujo de datos. En la codificación de las muestras en formato de audio digital sin comprimir se realiza una transformada al dominio frecuencial y se cuantifican y codifican a partir de un conjunto de datos obtenidos en un modelo psicoacústico (estudio de la relación existente entre las características físicas de un estímulo sonoro y la respuesta de carácter psicológico que el mismo provoca en un sujeto) el cual determina la calidad final.

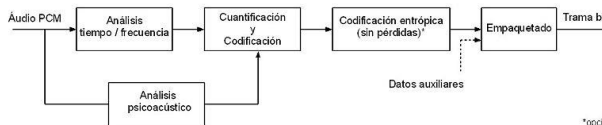


Ilustración 1 – Esquema de un codificador perceptual de audio

- **Codificadores paramétricos:** estos códecs enfatizan en el hecho de que el audio y la voz se pueden representar y sintetizar con todos aislados, patrones armónicos y componentes ruidosas, representándolos mediante parámetros como la amplitud, la frecuencia fundamental o los componentes espectrales los cuales requieren muy pocos bits para representarlos, durante este proceso, en primera medida se extrae información de las muestras de entrada aplicando una transformada de Fourier para luego realizar la estimación de parámetros posteriormente, se codifican basándose en un modelo de la percepción humana y se multiplexan (técnica de combinación de dos o más señales y transmitir las por un solo medio de transmisión) para formar la trama binaria.

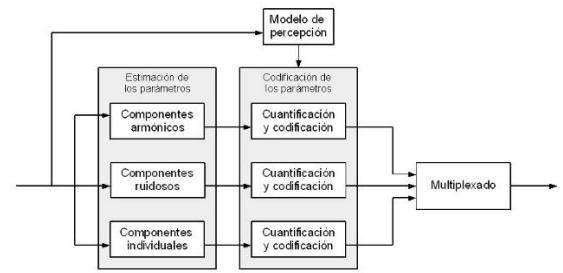


Ilustración 2 - Codificador paramétrico de audio

- **Vocoders:** los vocoders o codificación de voz son codificadores paramétricos específicos para la codificación de la voz, analizando la señal de voz correspondiente a un segmento temporal considerado estacionario para extraer los parámetros del modelo y la excitación y esta es la información que se codifica.

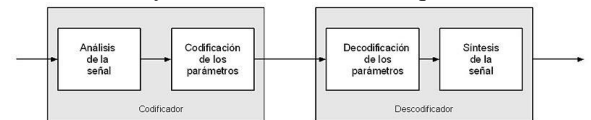


Ilustración 3 - Esquema de un sistema de codificación de voz

- **Codificadores de forma de onda:** estos codificadores se basan en el estudio de la señal de forma que intentan reproducir la forma de la señal de entrada, se aprovecha la redundancia de la señal y, a partir de una predicción lineal, permiten codificar la señal auditiva y consiguiendo tasas de compresión elevadas cuando las señales son redundantes y prácticamente nulas cuando no sea de esta manera y teniendo en cuenta que esta codificación de la señal se puede llevar a cabo tanto en el dominio temporal (modulación por codificación de pulsos, impulsos diferenciales) como en el dominio frecuencial (codificación en sub-bandas y por transformada).
- **Codificadores híbridos:** Estos codificadores híbridos son otra forma de codificación de voz y se conocen como codificadores de análisis por síntesis el cual combinan las técnicas de los codificadores de forma de onda con los vocoders, tiene como finalidad obtener voz de muy buena calidad a unas tasas de bit relativamente bajas y su proceso se basa en analizar un conjunto de muestras en conjunto, es decir, como si fuera una sola para obtener los parámetros de la señal.

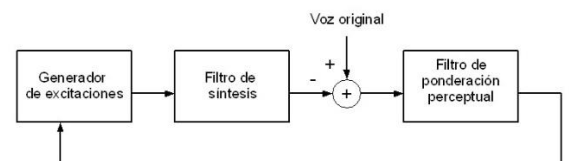


Ilustración 4 - Codificador híbrido

Una parte fundamental de la codificación de audio son los parámetros de caracterización o toma en cuenta para la codificación, unos de estos parámetros para la codificación de audio son:

- **Número de canales:** Este número de canales depende

del número de señales de audio simultáneos que contiene el flujo de datos. Puede ser mono (1 canal), estéreo (2 canales) o multicanal como 5.1 (seis canales) o 7.1 (ocho canales).

- **Frecuencia de muestreo:** esta determina la calidad percibida, es decir, que entre más alta sea esta frecuencia más apegado será el sonido con respecto al original.
- **Número de bits por muestra:** se enfoca en la precisión con la que se reproduce la señal original y el rango dinámico de la misma.
- **Tipo de compresión:** es decir, la compresión con pérdidas y la compresión con pérdidas.
- **Tasa de bits:** Un aspecto muy importante por no decir que es el principal por lo que indica el número de bits de información necesarios por unidad de tiempo y por consiguiente es el que determinará el tamaño final del archivo.

Es valioso mencionar que la codificación de audio posee grandes aplicaciones como lo es la reducción de espacio para almacenamiento, algo que se ha mencionado durante todo el documento; pero también posee una gran aplicación en el campo de la transmisión de información por cualquier tipo de red, puesto que a menor tasa de bits más rápido será el envío, se puede utilizar para podcasting en programas de televisión o radio y especialmente en telefonía en sus diferentes formas [7].

IV. ANÁLISIS DE ARTÍCULOS RELACIONADOS

La compresión de audio no es un tema nuevo como tal, a lo largo de los años diversos investigadores han realizado grandes trabajos relacionados y plasmado sus resultados en un documento investigativo y a continuación se presentarán algunos de ellos:

A. *Audio compression using dynamic Huffman and RLE coding [9]*

Este artículo, considera la implementación de la compresión de audio utilizando las técnicas de compresión sin pérdidas como la codificación dinámica de Huffman y la codificación de longitud de carrera (RLE). El archivo de audio se preprocesa en primer lugar para encontrar la frecuencia de muestreo y los bits de datos codificados en el archivo de audio de muestra. Después de eso se aplica la dinámica Huffman y RLE. El diseño de la técnica de codificación dinámica de Huffman implica la evaluación de las probabilidades de que ocurra "sobre la marcha", ya que el conjunto se está transmitiendo y la RLE se basa en la búsqueda de las pistas de los datos, es decir, en la repetición de las cadenas y su sustitución por un solo elemento de datos y su recuento. Estas técnicas trabajan con un objetivo común de obtener la mayor relación de compresión posible y menos tiempo transcurrido para comprimir. La competencia de los métodos propuestos se verifica aplicando estas técnicas a una variedad de datos de audio. El estímulo que subyace a este trabajo es ofrecer un análisis detallado de los métodos de compresión sin pérdidas y encontrar el que mejor se adapte a la compresión de datos multimedia en el entorno de la radio cognitiva.

Debido a problemas en la obtención del documento completo no fue posible acceder a los detalles específicos del proceso de compresión.

B. *Comparison of psychoacoustic principles and genetic algorithms in audio compression [10]*

En este segundo artículo se evidencia que se puede lograr una alta compresión de datos de audio eliminando la información irrelevante de la señal que no es detectable ni siquiera por un oyente bien entrenado o sensible. Los esquemas de codificación de audio contemporáneos como MP3, AAC y Ogg Vorbis identifican la información irrelevante durante el análisis de la señal incorporando en el codificador varios principios psicoacústicos, incluidos los umbrales de audición absolutos, el análisis de banda crítica, el enmascaramiento simultáneo y el enmascaramiento temporal (Painter y Spanias, 2000). El enmascaramiento es el proceso de eliminación de señales sonoras débiles, pero normalmente audibles que se hacen inaudibles por estar muy próximas en frecuencia a los sonidos circundantes o tener amplitudes mucho menores que éstos. Se han realizado numerosos estudios sobre algoritmos genéticos, que resuelven los problemas modelando la evolución darwiniana. Los algoritmos se han aplicado recientemente a la codificación de audio con cierto éxito (Galos et al., 2003). Para lograr la compresión de audio, los algoritmos genéticos analizan un gran número de archivos de sonido para determinar los trozos que tienen más probabilidades de contener señales irrelevantes. La combinación de los trozos irrelevantes, forman una solución que se utilizará para comprimir cualquier archivo de sonido. Presentamos en este trabajo un estudio de la comparación de la aplicación de principios psicoacústicos y algoritmos genéticos para comprimir señales de sonido. Desarrollamos un codificador para realizar el experimento, en el que, al igual que la mayoría de los codificadores de audio conocidos, se utiliza la codificación de Huffman para manejar la compresión sin pérdidas y se utiliza la transformación discreta modificada del coseno (MDCT) para transformar las señales del dominio del tiempo al dominio de la frecuencia. Los resultados se comparan utilizando las relaciones señal-ruido (SNR) y pruebas subjetivas, en las que se pide a dieciocho sujetos (que son estudiantes en el CSUSB) que escuchen y califiquen los archivos descomprimidos por los dos métodos.

Al igual que en el artículo anterior, fue imposible acceder al proceso y los resultados del trabajo realizado para poder mostrarlos a detalle.

C. *LSB Based Audio Steganography Based On Text Compression [14]*

En este artículo, se enfocan en realizar una optimización en la compresión de texto mediante diccionario y comparándolo con otros métodos para evidenciar su amplia superioridad con el fin de reducir el tamaño de los audios a la hora de realizar esteganografía por lo que el algoritmo de compresión es lo que reduce la redundancia de la representación de datos y disminuye la capacidad de almacenamiento de datos. La compresión de datos juega un papel vital en la reducción del costo de comunicación haciendo uso del ancho de banda disponible proponiendo una nueva técnica de compresión de textos basada en diccionarios para textos ASCII con el fin de obtener un buen rendimiento en varios tamaños de documentos. Los bits de compresión basados en el diccionario se ocultan en el bit LSB

de las señales de audio y para calcular la relación señal-ruido (SNR). Esta esteganografía de audio se lleva a cabo para varios algoritmos de compresión con compresión basada en el diccionario evidenciando su superioridad y concluyendo que, en un canal, la reducción del tiempo de transmisión es directamente proporcional a la cantidad de compresión. Si el texto de entrada es sustituido por códigos de longitud variable con una longitud inferior a su tamaño medio, el tamaño del texto de entrada puede reducirse utilizando la compresión basada en el diccionario. Este algoritmo de compresión propuesto logra una buena relación de compresión, reduce los bits por carácter. Esta esteganografía de audio se lleva a cabo para varios algoritmos de compresión con compresión basada en el diccionario. La compresión de texto basada en la esteganografía de audio logra un mejor valor de SNR.

D. Digital Audio Compression [15]

Este artículo, proporciona una visión amplia sobre la compresión de audio digital desde lo que es la digitalización del audio hasta su compresión. La conversión de lo analógico a lo digital comienza por muestrear la entrada de audio en intervalos regulares y discretos de tiempo y cuantificar los valores aplicados en un número discreto de niveles espaciados uniformemente. Los datos de audio digital consisten en una secuencia de valores binarios que representan el número de niveles de cuantificación para cada muestra de audio. El método de representar cada muestra con una palabra de código independiente se denomina modulación por código de pulsos (PCM). Según la teoría de Nyquist, una señal muestreada en el tiempo puede representar fielmente señales de hasta la mitad de la tasa de muestreo. Las tasas de muestreo típicas van de 8 kilohercios (kHz) a 48 kHz. La tasa de 8 kHz cubre un rango de frecuencia de hasta 4 kHz y así cubre la mayoría de las frecuencias producidas por la voz humana. La tasa de 48 kHz cubre un rango de frecuencia de hasta 24 kHz y cubre de forma más que adecuada todo el rango de frecuencia auditiva, que para los humanos típicamente extiende a sólo 20 kHz. En la práctica, la gama de frecuencias es algo inferior a la mitad de la tasa de muestreo debido a las limitaciones prácticas del sistema. En comparación con la mayoría de los tipos de datos digitales (excluido el vídeo digital), las tasas de datos asociadas con el audio digital no presionado son sustanciales. Por ejemplo, los datos de audio de un disco compacto (2 canales de audio muestreados a 44,1 kHz con 16 bits por muestra) requieren una velocidad de datos de alrededor de 1,4 megabits por segundo. Es evidente que se necesita alguna forma de compresión para permitir un almacenamiento y una transmisión más eficientes de estos datos.

La transformación de ley es una técnica básica de compresión de audio especificada por la Recomendación G.711 del Comité Consultivo Internacional de Telecomunicaciones (CCITT). La transformación es de naturaleza esencialmente logarítmica y a la baja los códigos de salida de 8 bits por muestra para cubrir un rango dinámico equivalente a 14 bits de valores linealmente cuantificados. Esta transformación ofrece una relación de compresión de (número de bits por muestra fuente) 8 a 1. A diferencia de la cuantificación lineal, los espaciamientos de pasos logarítmicos representan *samples* de audio de baja amplitud con mayor precisión que los valores de amplitud más altos. Así, la relación señal-ruido de la salida formada por el

transductor es más uniforme en el rango de amplitudes de la señal de entrada.

El siguiente enfoque es la Modulación de código de pulso diferencial adaptativo, el codificador ADPCM puede adaptarse a las características de la señal de audio cambiando el tamaño del paso de *either* el cuantificador o el predictor, o cambiando ambos. El método de cálculo del valor predicho y la forma en que el predictor y el cuantificador se adaptan a la señal de audio varían entre los diferentes sistemas de codificación ADPCM.

El siguiente algoritmo de compresión de audio es el algoritmo de compresión de audio del Grupo de Expertos en Cinematografía (MPEG) es una norma de la Organización Internacional de Normalización (ISO) para la compresión de audio de alta fidelidad. Es una parte de un estándar de compresión de tres partes. Con las otras dos partes, video y sistemas, el estándar compuesto se ocupa de la compresión de video y audio sincronizados a una velocidad de bits total de aproximadamente 1,5 megabits por segundo. Al igual que la ley y el ADPCM, la compresión MPEG/audio tiene pérdidas; sin embargo, el algoritmo MPEG puede lograr una compresión transparente y sin pérdidas perceptivas. El comité de MPEG/audio realizó pruebas de escucha subjetivas extensas durante el desarrollo del estándar. Las pruebas mostraron que incluso con una relación de compresión de 6 a 1 (estéreo, 16 bits por de audio de muestra muestreada a 48 kHz comprimida a 256 kilobits por segundo) y bajo condiciones óptimas de escucha, los oyentes expertos fueron incapaces de distinguir entre audios originales y codificados. El alto rendimiento de este algoritmo de compresión se debe a la explotación del enmascaramiento auditivo. Este enmascaramiento es una debilidad perceptiva del oído que se produce cuando la presencia de una señal de audio fuerte hace imperceptible una vecindad espectral de señales de audio más débiles. Este fenómeno de enmascaramiento de ruido ha sido observado y corroborado a través de una variedad de experimentos psicoacústicos. Los resultados empíricos también muestran que el oído tiene una selectividad de frecuencia limitada que varía en agudeza desde menos de 100 Hz para las frecuencias audibles más bajas hasta más de 4 kHz para las más altas. Por lo tanto, el espectro auditivo puede dividirse en bandas críticas que reflejan el poder de resolución del oído como función de la frecuencia.

E. DSP audio compression system based on G.723.1 [11]

Este artículo presenta un sistema de compresión de audio basado en el algoritmo G.723.1. El sistema es capaz de trabajar en patrón dúplex completo ya que puede comprimir y descomprimir los datos recibidos al mismo tiempo. El sistema funciona con el TMS320C5509A de TI, que es un procesador de señal digital de punto fijo. El chip DSP utiliza MCBSP (Multi-Channel Buffered Serial Port) para recibir y transferir datos entre el chip de códec AIC23 y DMA (Direct Memory Access). El algoritmo G.723.1 se traslada al chip DSP y se optimiza para cumplir con los requisitos del sistema. En el presente documento se introducen cuatro métodos para optimizar el algoritmo. Tras la optimización, el tiempo de compresión y descompresión se reduce de más de 310ms a 11ms. Este sistema utiliza el RS232 para comunicarse con otro monitor superior.

F. Improved detection of MP3 double compression using content-independent features [12]

Con el auge de los teléfonos inteligentes y las redes inalámbricas de alta velocidad en los últimos años, la transmisión y el intercambio de audio se vuelven convenientes y baratos, de modo que los medios digitales están reemplazando gradualmente a los medios físicos. Esta tendencia también ha provocado más ataques al audio digital y su aplicación. La doble compresión de MP3, que se logra descomprimiendo y recomprimiendo el audio con una relación de compresión diferente, es una manipulación típica del audio con fines malignos. En este documento, se propone un enfoque para detectar los archivos de audio MP3 con codificación ascendente y descendente y revelar la calidad real de la compresión sobre la base de patrones estadísticos extraídos de los coeficientes cuantificados de MDCT y sus derivados. Para minimizar la falsa predicción causada por las características individuales de los clips de audio diversificados, generamos señales de audio de referencia recomprimiendo y calibrando el audio, y medimos las diferencias entre las características basadas en la señal y las basadas en la referencia. Se aplicaron máquinas de vectores de apoyo y sistemas de inferencia neural-fuzzy en evolución dinámica para clasificaciones binarias y multiclase. Los resultados experimentales muestran que nuestro enfoque detecta eficazmente la doble compresión de MP3 y expone el historial de procesamiento de audio para la ciencia forense digital.

G. Music files compression based on time-frequency representation of audio signal [13]

En este último artículo analizado, se ofrece un algoritmo de compresión de datos de audio basado en la presentación de la frecuencia de tiempo de la señal de música. Para el cálculo de la representación tiempo-frecuencia de una señal de audio se desarrolló el algoritmo, basado en la transformada rápida de Fourier (SFFT). El resultado del trabajo del algoritmo son los archivos que tienen como es posible el tamaño más pequeño a condición de que se mantenga el sensor principal de la música, para tareas de identificación de una señal, búsqueda de piezas de música, etc.

V. PARTE PRÁCTICA

Como se ha mencionado a lo largo del presente trabajo, la codificación de audio consiste en un proceso que busca reducir la tasa de bits de un archivo con la finalidad de disminuir en lo amplio de su posibilidades el espacio que ocupa en el disco de almacenamiento y en este caso, se realizó un programa en ambiente web con el lenguaje de programación php con la finalidad de comprimir archivos de audio en formato mp3 variando los canales de audio y la tasa de bits y comprobando que efectivamente reduce el tamaño del espacio que ocupa el archivo de audio en formato mp3 en disco.

En primera medida se muestra la interfaz de HTML5 en donde se construyó la parte gráfica del proyecto con la ayuda de Bootstrap para ayudar en ciertas cosas de estilización de la página. Seguidamente se muestra la creación del formulario para la captura del archivo de audio en formato mp3, haciendo claridad que no se envía por el método normal de un formulario, es decir, por “action” sino por “post” para evitar que se recargue la página, luego de adjuntar el archivo de audio, se recogerá la

información por medio de un ajax como se detallará posteriormente.

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Compresor de audio</title>
7   <link rel="icon" href="img/1.png" type="icon/png" />
8   <link rel="stylesheet" href="css/bootstrap.min.css">
9   <link rel="stylesheet" href="css/styles.css">
10  <script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
11  <script src="//code.jquery.com/jquery-1.11.3.min.js"></script>
12  <script src="js/bootstrap.min.js"></script>
13  <script type="text/javascript" src="js/bootstrap-filestyle.min.js"></script>
14 </head>
15
16 <body>
17   <div class="container">
18     <form method="post" action="" enctype="multipart/form-data" id="myform" enctype="multipart/form-data">
19       <div class="form-group col-md-8">
20         <input type="file" name="file" id="cancion" class="filestyle" data-input="false" data-classIcon="icon-plus" />
21       </div>
22       <div class="form-group col-md-4">
23         <input type="submit" value="Comprimir canción" class="btn-compress">
24       </div>
25     </form>
26   </div>
27 </body>
28 </html>

```

Ilustración 5 - HTML (primera parte)

Al momento de oprimir el botón de “comprimir canción” el usuario verá una ilustración en movimiento asemejando la funcionalidad de una barra de progreso con el fin de dar a entender que se está realizando la compresión.

```

26 </div>
27 </div>
28 </form>
29 <div class="row">
30   <div class="col-md-12 text-center" id="loader"></div>
31 </div>
32 </div>
33 </div>
34 </div>
35 </div>
36 </div>

```

Ilustración 6 - HTML (segunda parte)

Al mismo tiempo de oprimir ese botón, internamente el flujo de proceso se traslada al script en donde se captura el formulario y se crea un elemento de HTML para que al momento que termine el proceso del programa se muestre el contenido según el resultado obtenido, si el servidor da una respuesta positiva entonces indica que se comprimió correctamente o por el contrario ocurrió algún error.

```

37 <script>
38   $(function() {
39     $('#file').Filestyle({ text: 'Seleccionar una canción', btnClass: 'btn-primary' });
40     $('#formmyform').submit(function (e) {
41       e.preventDefault();
42       var formData = new FormData(this);
43       var elem = document.createElement("div");
44
45       $.ajax({
46         url: 'server.php',
47         type: 'POST',
48         data: formData,
49         beforeSend: function () {
50           $('#loader').show();
51           $('#btn-compress').prop('disabled', true);
52         },
53         success: function (data) {
54           if (data.success == true) {
55             data.originalSong + "<strong> 1 </strong>" + data.compressedSong + "<strong>";
56             swal({
57               title: 'Compresión Terminada!',
58               content: elem,
59               button: 'Cerrar'
60             });
61             $('#btn-compress').prop('disabled', false);
62             $('#cancion').val(null);
63             $('#loader').hide();
64           } else {
65             swal("Oops..!",
66               "La canción " + data.error + " ya existe!",
67               "error");
68             $('#btn-compress').prop('disabled', false);
69             $('#cancion').val(null);
70             $('#loader').hide();
71           }
72         },
73         cache: false,
74         contentType: false,
75         processData: false
76       });
77     });
78   });
79 </script>
80 </html>

```

Ilustración 7 - HTML (tercera parte)

Para que la parte visual del proyecto se vea de manera óptima se necesitan ayudas como la utilización de Bootstrap, aunque en algunas ocasiones se hace necesario estilizar los componentes de manera manual con CSS como se muestra a continuación:

```
css > # styles.css > .container
1 body {
2   background-image: url(../img/rupixen-com-oVnCYVs_6TK-unsplash.jpg);
3   background-repeat: no-repeat;
4   -webkit-background-size: cover;
5   -moz-background-size: cover;
6   -o-background-size: cover;
7   background-size: cover;
8 }
9
10 .container {
11   margin-top: 100px;
12   background-color: rgba(17, 17, 17, 0.849);
13 }
14
15 .container-buttons {
16   padding-top: 20px;
17 }
18
19 input[type="file"] {
20   color: white;
21   padding-left: 30px;
22 }
23
24 #loader {
25   display: none;
26 }
27
28 .btn-compress {
29   background-color: #54ca6d;
30   padding: 10px;
31   font-weight: 900;
32   border-radius: 17px;
33   border: 1px solid white;
34 }
35
36 .btn-compress:hover {
37   background-color: #02ff39;
38   transition: ease 1s;
39 }
40
```

Ilustración 8 - CSS (primera parte)

```
41 .custom-file-input {
42   color: transparent;
43   opacity: 1;
44 }
45
46 .custom-file-input::-webkit-file-upload-button {
47   visibility: hidden;
48 }
49
50 .custom-file-input::before {
51   content: 'Selecciona una canción';
52   color: black;
53   display: inline-block;
54   background: -webkit-linear-gradient(top, #f9f9f9, #e3e3e3);
55   border: 1px solid #999;
56   border-radius: 3px;
57   padding: 5px 8px;
58   outline: none;
59   white-space: nowrap;
60   -webkit-user-select: none;
61   cursor: pointer;
62   text-shadow: 1px 1px #fff;
63   font-weight: 700;
64   font-size: 10pt;
65   margin-right: 30px;
66 }
67
68 .custom-file-input:hover::before {
69   border-color: black;
70 }
71
72 .custom-file-input:active {
73   outline: 0;
74 }
75
76 .custom-file-input:active::before {
77   background: -webkit-linear-gradient(top, #e3e3e3, #f9f9f9);
78 }
```

Ilustración 9 - CSS (segunda parte)

Finalmente, pasamos a la parte lógica del proyecto en donde se realiza el proceso de compresión del archivo de audio programado en PHP en donde se recibe lo enviado del formulario, extrayendo la ruta de la canción así como su nombre y la extensión del archivo de audio para posteriormente hacer la búsqueda de la canción en la carpeta de los archivos originales para verificar que no se haya comprimido antes y luego sino no se ha comprimido antes se mueve el archivo a la carpeta de archivos originales y ya está, se da la indicación de que ya se ha comprimido ese archivo.

```
<?php
1
2 require './vendor/autoload.php';
3 header('Content-type: application/json');
4 $song = $_FILES['file']['name'];
5 $extensions = array("mp3","wav");
6 $extension = pathinfo($_FILES['file']['name'], PATHINFO_EXTENSION);
7 $songName = $_FILES['file']['name'];
8 if (($extension == "mp3") || $extension == "wav" && in_array($extension, $extensions)) {
9   if ($_FILES['file']['error'] > 0) {
10     echo "Return Code: " . $_FILES['file']['error'] . "<br />";
11   } else {
12     if (file_exists("upload/" . $songName)) {
13       $originalSong = $songName;
14       echo json_encode(array('error' => $originalSong, 'success' => false));
15       return;
16     } else {
17       move_uploaded_file(
18         $_FILES['file']['tmp_name'],
19         "upload/" . $songName
20       );
21       $originalSong = "Cancion original guardada en: " . "upload/" . $songName;
22       $compressedSong = "Cancion comprimida guardada en: compressedSongs/" . $songName;
23     }
24   }
25 } else {
26   echo "Invalid file";
27 }
28
29
30
```

Ilustración 10 - PHP obtención de datos y ubicación de archivo

Luego, se pasa a la parte de la compresión del archivo de audio, creando una variable con el nombre de la librería utilizada para la compresión que es “FFmpeg” [8] para posteriormente llamarla y crear una primera instancia de la misma con los ejecutables de la librería, luego se abre la canción original mencionando el formato mediante la librería libmp3lame y finalmente se toma la canción y se realiza la compresión teniendo en cuenta dos factores de los mencionados anteriormente que son los canales y la tasa de bits, se mencionan los nuevos valores y por último luego de la compresión se guarda el archivo nuevo en la carpeta de comprimidos y por medio de un json se envía un mensaje al HTML para desplegar en pantalla que se realizó correctamente el proceso.

```
31 $ffmpeg = FFmpeg\FFmpeg::create(array(
32   'ffmpeg.binaries' => 'C:\ffmpeg\bin\ffmpeg.exe',
33   'ffprobe.binaries' => 'C:\ffmpeg\bin\ffprobe.exe'
34 ));
35
36
37 $audio = $ffmpeg->open('upload/' . $song);
38 $format = new FFmpeg\Format\Audio\Wav();
39
40 $format->on('progress', function ($audio, $format, $percentage) {
41 });
42
43 $format
44   ->setAudioChannels(2)
45   ->setAudioKiloBitrate(32);
46
47 $audio->save($format, 'C:\xampp\htdocs\compressor-project\compressedSongs/' . $song);
48
49 echo json_encode(array('originalSong' => $originalSong, 'compressedSong' => $compressedSong, 'success' => true));
50
```

Ilustración 11 - PHP compresión y ubicación de archivo final

VI. RESULTADOS

En la realización de la parte práctica del tema, creando un programa para comprimir archivos de audio en formato mp3 se obtuvieron grandes resultados al comprobar que a medida que se disminuía la tasa de bits la compresión era significativamente más grande logrando en ocasiones reducir en más de la mitad el peso del archivo de audio, aunque teniendo en cuenta que entre más se comprimía un archivo más se deformaba el audio y poco a poco iba perdiendo el sentido alejándose más del original, es decir, que entre más compresión se escuchaba un sonido muy distinto al original por medio de una distorsión muy grande del mismo, lo que da a entender que es un proceso de compresión con pérdida porque luego por más que se aumentara la tasa de bits al archivo comprimido y resultara de un peso aún superior al original, la calidad del audio jamás regresaba a lo que es el

archivo original además, se logró evidenciar de manera práctica que la verdadera influencia a la hora de comprimir los archivos de audio es la tasa de bits y con muy poca relevancia los canales de audio.

VII. CONCLUSIONES

El desarrollo de este proyecto investigativo y práctico acarreo grandes enseñanzas, hallazgos, ayudas e incentivó un espíritu investigativo y curioso respecto a los temas planteados, debido al hecho de que se dio una visión muy amplia sobre lo que es el sonido, el audio, los archivos de audio digitales para posteriormente llegar a lo que es la comprensión de estos archivos, entendiendo que hay muchas formas de realizar este proceso tanto en software como en hardware y es muy utilizado en ciertas ramas se podría decir que vital para ciertos trabajos, asimismo con la realización de la parte práctica se acentuaron todos estos conocimientos, definiciones, fórmulas y demás experimentando de manera vivenciada la comprensión de archivos de audio, logrando en cierta manera manipular y jugar con su peso.

VIII. REFERENCIAS

- [1] "Compresión de audio", Es.wikipedia.org, 2020. [Online]. Available: https://es.wikipedia.org/wiki/Compresi%C3%B3n_de_audio. [Accessed: 03- Aug- 2020].
- [2] "Como Comprimir un Audio MP3 sin Perder Calidad | Comprime tu Música y Canciones | Mira Cómo Se Hace", Mira Cómo Se Hace, 2020. [Online]. Available: <https://miracomosehace.com/comprimir-audio-mp3-calidad-musica/>. [Accessed: 14- Aug- 2020].
- [3] "Compresión de audio - Multimedia", Pulso.uniovi.es, 2020. [Online]. Available: http://www.pulso.uniovi.es/wiki/index.php/Compresi%C3%B3n_de_audio. [Accessed: 14- Aug- 2020].
- [4] "Código de audio", Es.wikipedia.org, 2020. [Online]. Available: https://es.wikipedia.org/wiki/C%C3%B3dec_de_audio. [Accessed: 02- Aug- 2020].
- [5] "Cómo comprimir tus archivos de audio fácilmente con estas herramientas", SoftZone, 2020. [Online]. Available: <https://www.softzone.es/2016/07/09/como-comprimir-tus-archivos-de-audio-facilmente-con-estas-herramientas/>. [Accessed: 14- Aug- 2020].
- [6] "Encode/MP3 – FFmpeg", Trac.ffmpeg.org, 2020. [Online]. Available: <https://trac.ffmpeg.org/wiki/Encode/MP3>. [Accessed: 08- Aug- 2020].
- [7] ¿. ES, "¿En qué consiste la Compresión de Audio? | NFON Base de Conocimientos ES", Nfon.com, 2020. [Online]. Available: <https://www.nfon.com/es/servicio/base-de-conocimiento/base-de-conocimiento-destacar/compresion-de-audio>. [Accessed: 14- Aug- 2020].
- [8] "ffmpeg Documentation", Ffmpeg.org, 2020. [Online]. Available: <https://ffmpeg.org/ffmpeg.html#Audio-Options>. [Accessed: 03- Aug- 2020].
- [9] R. B. Patil and K. D. Kulat, "Audio compression using dynamic Huffman and RLE coding," 2017 2nd International Conference on Communication and Electronics Systems (ICES), Coimbatore, 2017, pp. 160-162, doi: 10.1109/CESYS.2017.8321256.
- [10] H. Chen and T. L. Yu, "Comparison of psychoacoustic principles and genetic algorithms in audio compression," 18th International Conference on Systems Engineering (ICSEng'05), Las Vegas, NV, USA, 2005, pp. 270-275, doi: 10.1109/ICSENG.2005.28.
- [11] P. Li, X. Yu and G. Chen, "DSP audio compression system based on G.723.1," 2015 International Conference on Smart and Sustainable City and Big Data (ICSSC), Shanghai, 2015, pp. 160-164, doi: 10.1049/cp.2015.0269.
- [12] M. Qiao, A. H. Sung and Q. Liu, "Improved detection of MP3 double compression using content-independent features," 2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013), KunMing, 2013, pp. 1-4, doi: 10.1109/ICSPCC.2013.6664121.
- [13] A. V. Shekhirev and E. V. Rabinovich, "Music files compression based on time-frequency representation of audio signal," 2008 Third International Forum on Strategic Technologies, Novosibirsk-Tomsk, 2008, pp. 340-342, doi: 10.1109/IFOST.2008.4602938.
- [14] M. Baritha Begum and Y. Venkataramani, "LSB Based Audio Steganography Based On Text Compression", Procedia Engineering, vol. 30, pp. 703-710, 2012. Available: 10.1016/j.proeng.2012.01.917.
- [15] M. Baritha Begum and Y. Venkataramani, "LSB Based Audio Steganography Based On Text Compression", Procedia Engineering, vol. 30, pp. 703-710, 2012. Available: 10.1016/j.proeng.2012.01.917.