

Proyecto 1

Esteganografía por Enmascaramiento con Eco

Fecha de asignación: 3 de octubre, 2018
Grupos: 3

Fecha de entrega: 24 de octubre, 2018

1. Introducción

La esteganografía es una técnica mediante la cual se codifica un mensaje dentro de una señal portadora de manera que éste pase inadvertido en la misma. Típicamente se aprovechan las limitaciones perceptuales del humano para esconder dicho mensaje y así lograr que la señal portadora original sufra mínimas perturbaciones. Con el auge de la tecnología digital es común observar aplicaciones esteganográficas en el campo de las imágenes, video, audio, seguridad y telecomunicaciones. Los sistemas LTI facilitan la implementación y análisis de muchos sistemas esteganográficos.

Audio es una disciplina con gran participación en la esteganografía. Entre las aplicaciones principales destacan la seguridad contra piratería, restricciones geográficas de contenido y adición de metadatos. Asimismo existen gran variedad de técnicas que logran codificar un mensaje en una señal, entre las cuáles se puede mencionar Codificación de Bit, Espectro Disperso, Codificación de Fase y Enmascaramiento con Eco[1, 2].

En particular, la codificación por Enmascaramiento con Eco es de interés para este curso. Sus diferentes componentes internos se pueden representar mediante sistemas LTI como los estudiados hasta el momento. El presente proyecto consiste en la implementación de un sistema esteganográfico mediante la técnica de Enmascaramiento con Eco para señales digitales de audio, utilizando los conceptos y herramientas estudiadas hasta el momento.

2. Objetivo general

Al finalizar el proyecto, el estudiante habrá aplicado los conocimientos obtenidos en lecciones del curso para implementar un sistema esteganográfico para señales digitales de audio.

3. Objetivos específicos

1. Hacer uso de técnicas propias de los sistemas LTI como la convolución para la manipulación de señales.
2. Utilizar técnicas de implementación prácticas como el solapamiento y suma/almacenamiento para el procesado en línea de señales.

3. Hacer uso de técnicas en el dominio de la frecuencia como el cepstro para analizar el estado de un sistema.
4. Expresar el diseño, implementación y resultados obtenidos en un artículo científico.
5. Enfrentar problemas prácticos al implementar sistemas en los lenguajes de programación C o C++.
6. Investigar temas no tratados en clase para demostrar capacidad de integración de conocimientos.

4. Descripción del Sistema

4.1. General

El sistema a implementar está basado en la solución descrita por Gruhl y Bender en [3]. Información adicional puede encontrarse en [4, 1, 2].

En términos generales, el sistema está formado por dos grandes subsistemas: Codificador y Decodificador. Dichos subsistemas se muestran en la Figura 1.

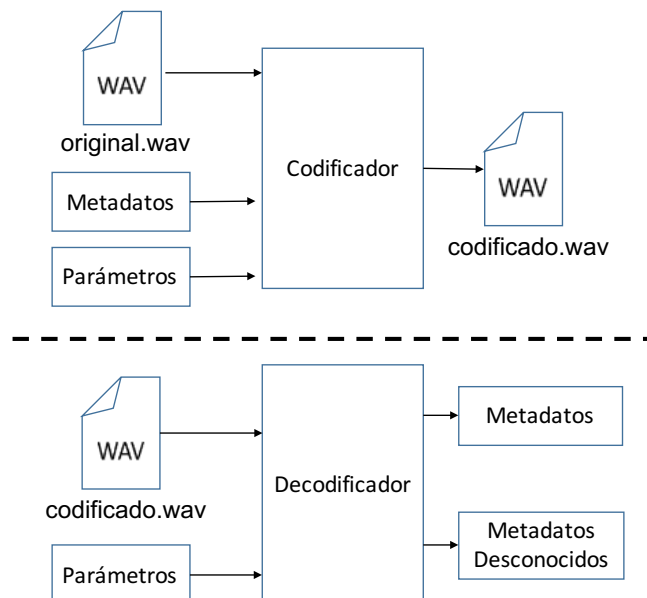


Figura 1: Diagrama general del sistema a implementar

El sistema no requiere ser operado en tiempo real. Tal y como se muestra en la Figura 1, la entrada a ambos subsistemas es un archivo.

Los sistemas se operarán individualmente y de manera no simultánea. Es decir, primero se utiliza el sistema en modo codificador para generar un archivo codificado y posteriormente el mismo se pasa por el sistema en modo decodificador para obtener los metadatos enmascarados en la señal.

A continuación se describen ambos sistemas individualmente.

4.2. Codificador

La codificación de los bits de metadatos se realiza agregando ecos controlados en ventanas de la señal original.

El sistema codificador recibe como entrada un archivo de audio y un conjunto de metadatos. El mismo utilizará Enmascaramiento con Eco para esconder dichos datos en la señal original y almacenará el resultado en un archivo de audio resultante. La Figura 2 muestra el diagrama de bloques del subsistema a implementar.

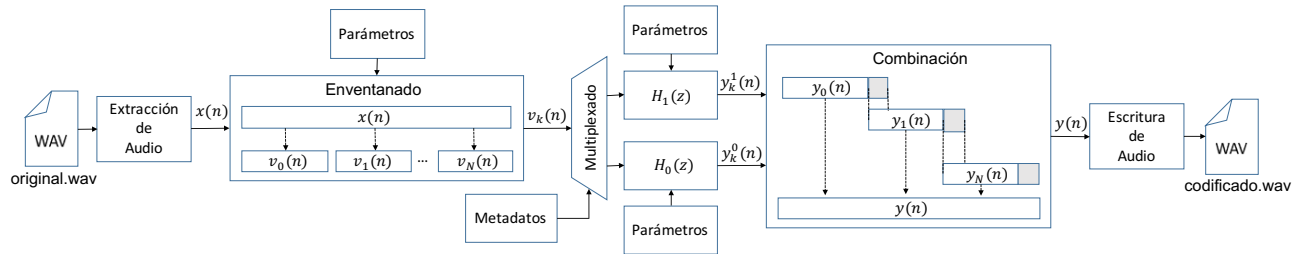


Figura 2: Diagrama de bloques del sistema codificador

4.2.1. Extracción de Audio

El bloque de extracción de audio se encargará de leer el archivo de audio en formato WAV y extraer del mismo la señal cruda a procesar $x(n)$. Estos datos son pasados a las etapas posteriores del codificador.

4.2.2. Enventanado

La entrada a este sistema es la señal original $x(n)$, la cual se segmenta en ventanas de longitud fija $v_k(n)$ de acuerdo a las configuraciones establecidas por el usuario. Dichas ventanas sirven como salida para este sistema.

4.2.3. Multiplexado

Este bloque recibe los metadatos a codificar y los procesa bit por bit. Dichos bits sirven como señal de control para decidir si la ventana segmentada $v_k(n)$ con anterioridad deberá pasarse por

el sistema $H_0(z)$ o $H_1(z)$. Cada bit controla el destino de una ventana individual, por lo que la tasa de transferencia de metadatos es directamente proporcional al tamaño de la ventana elegida.

4.2.4. $H_0(z)$ y $H_1(z)$

Estos bloques son los encargados finalmente de codificar el bit en una ventana. En sistema $H_0(z)$ codificará un bit en 0 y $H_1(z)$ codificará un bit en 1. En (1) se describe la función de transferencia de ambos sistemas:

$$H_b(z) = 1 + a_b Z^{-t_b} \quad (1)$$

Dicho sistema añade un eco a la señal original. Los parámetros a_b y t_b representan la amplitud y el retardo de dicho eco respectivamente. Variando los mismos se obtiene una codificación de un bit b ya sea en 0 o en 1. La Figura 3 muestra, a manera de ejemplo, las respuestas al impulso de (1) para ambos casos.

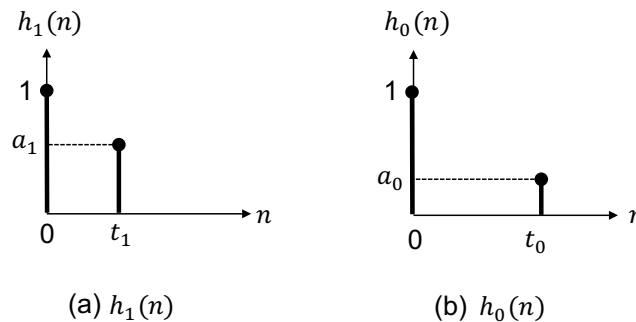


Figura 3: Respuestas al impulso para $H_1(z)$ y $H_0(z)$ respectivamente

La elección de a_b y t_b tiene un impacto directo en el desempeño del sistema. Retardos muy largos podrán disminuir la tasa de transferencia de metadatos, pero retardos cortos podrán causar distorsiones en la señal original y dificultar su decodificación. Similarmente, amplitudes altas pueden facilitar la decodificación de los metadatos pero los ecos podrían empezar a ser perceptibles al oído humano. El comportamiento del sistema ante diferentes configuraciones de $H_b(z)$ deberá estudiarse y mostrarse como resultados en el informe de este proyecto.

Si bien la aplicación de este sistema puede lograrse de diversas maneras, para el caso específico de este proyecto se utilizará convolución.

4.2.5. Combinación de ventanas

El resultado de convolucionar la ventana de largo M con la respuesta al impulso del sistema de largo N tendrá una longitud resultante de $M + N - 1$. El excedente de $N - 1$ debe tomarse en cuenta en la combinación de las ventanas procesadas en una señal codificada resultante. De

lo contrario, pueden inducirse componentes espectrales parásitas, distorsionar el audio original y corromper los metadatos codificados.

Este bloque se encarga de tomar los excedentes de las ventanas codificadas anteriormente $y_k(n)$ y combinarlos con las ventanas actuales. Para ello se propone el uso de dos técnicas similares:

- Solapamiento y suma (*overlap and add*)
- Solapamiento y almacenamiento (*overlap and sum*)

La solución implementará únicamente una de los dos. Las Figuras 4 y 5 muestran, a grandes rasgos, el funcionamiento del Solapamiento y suma y Solapamiento y almacenamiento respectivamente.

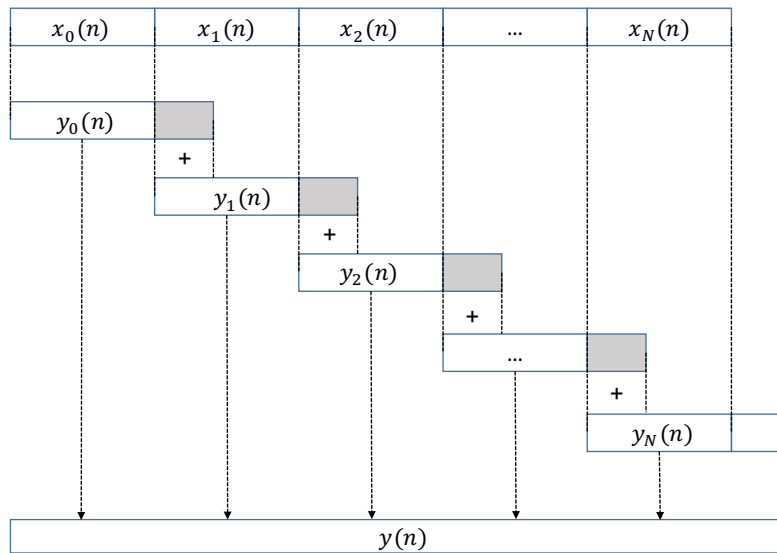


Figura 4: Funcionamiento de Solapamiento y suma

La salida de este bloque, entonces, es una ventana codificada en la cuál se ha considerado el excedente de la ventana anterior. Esto es el equivalente de procesar una señal de mayor extensión (audio original) con los sistemas, para obtener una señal resultante $y(n)$.

4.2.6. Escritura de Audio

Finalmente, el bloque de escritura de audio recibe estas ventanas codificadas y las almacena en el archivo WAV de destino. Este archivo deberá tener el mismo formato que el archivo original y podrá ser escuchado en reproductores de audio convencionales (Amarok, VLC, Totem, MPlayer, etc...).

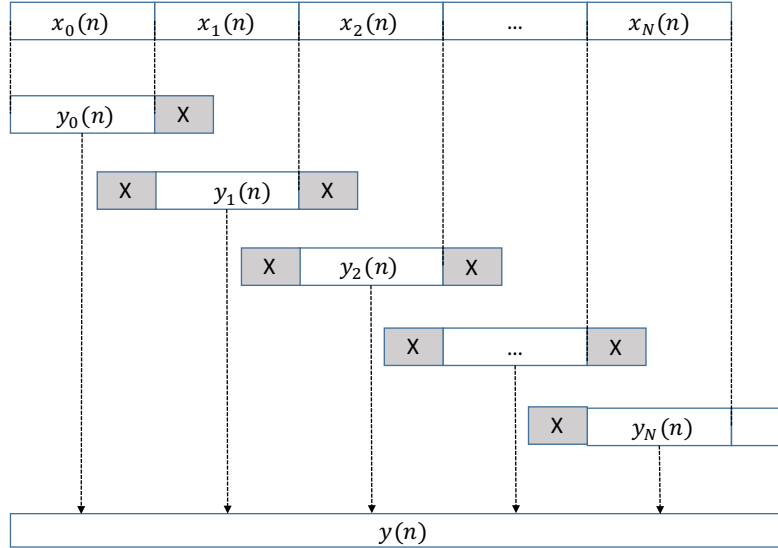


Figura 5: Funcionamiento de Solapamiento y almacenamiento

4.3. Decodificador

La decodificación de los bits de metadatos se realiza encontrando los retardos de los ecos inducidos y evaluando a cuál bit corresponden. Para ello se hace uso de la autocorrelación del Cepstro, ya que tiene la propiedad de destacar los mismos.

El sistema decodificador recibe como entrada un archivo de audio codificado previamente $x(n)$ y un conjunto de metadatos. La Figura 6 muestra el diagrama de bloques del subsistema a implementar.

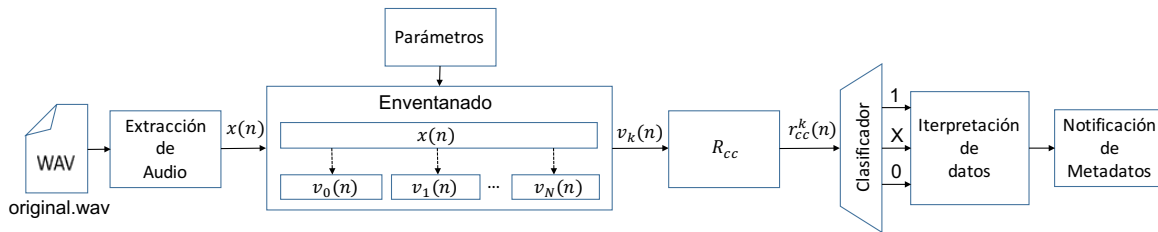


Figura 6: Diagrama de bloques del sistema decodificador

Los bloques de extracción de audio y de enventanado son equivalentes a los descritos en la sección del codificador.

4.3.1. Autocorrelación del Cepstro - R_{cc}

El Cepstro de una señal está dado por (2).

$$c(n) = \mathcal{F}^{-1} \{ \ln(\mathcal{F} \{x(n)\}) \} \quad (2)$$

Utilizando las propiedades de la transformada de Fourier se puede deducir que la autocorrelación está dada por (3).

$$\begin{aligned} R_{cc} &= c(n) * c(n) \\ R_{cc} &= \mathcal{F}^{-1} \{ \ln(\mathcal{F} \{x(n)\})^2 \} \end{aligned} \quad (3)$$

La entrada a este bloque será una ventana de la señal codificada $v_k(n)$ y la salida será el resultado de la autocorrelación del Cepstro para la misma $r_{cc}^k(n)$. La salida tendrá los retardos de los ecos pronunciados. Es de gran importancia que la longitud de estas ventanas sean congruentes con la longitud de las ventanas utilizadas durante la codificación.

4.3.2. Clasificador

El bloque clasificador es el que finalmente decodifica el bit de metadatos enmascarado en la ventana. Para ello, el bloque debe evaluar la magnitud de la autocorrelación del Cepstro tanto en los retardos t_1 y t_0 y decidir si se ha codificado un 1, un 0 o no se ha codificado nada del todo.

Las Figuras 7 y 8 muestran dos ejemplos de autocorrelación del Cepstro para el cual el clasificador ha decodificado un 1 y un 0 respectivamente. Ambas figuras fueron tomadas de [3].

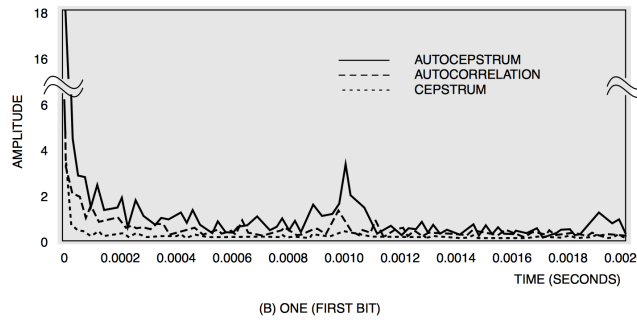


Figura 7: Autocorrelación del Cepstro para un 1 codificado

Es posible que, por la naturaleza de la señal original y los parámetros de codificación elegidos, el sistema falle el proceso de decodificación. A la tasa de bits decodificados correctamente sobre la cantidad de bits codificados en total se le conoce como *tasa de recuperación*. Este es un resultado de gran interés para caracterizar el sistema implementado y, por tanto, se deben presentar resultados del mismo en el informe del proyecto.

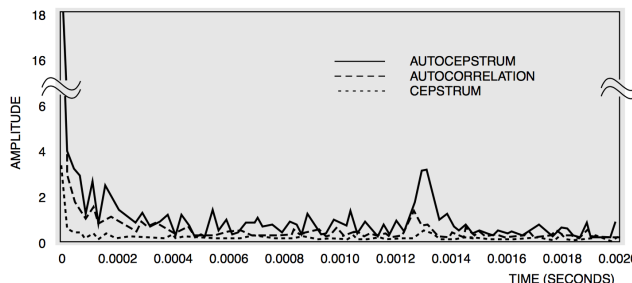


Figura 8: Autocorrelación del Cepstro para un 0 codificado

4.3.3. Interpretación de Datos

Una vez se ha decodificado la ventana recibida, el bit obtenido se pasa como entrada al bloque de interpretación de datos. En éste, los bits se acumulan para formar bytes, que a su vez se acumulan para formar símbolos que representen algún sentido para la aplicación en cuestión. Por ejemplo, los bits codificados podrían representar códigos ASCII los cuales combinados representan un texto legible al usuario.

La implementación de este bloque se deja abierto a diseño del estudiante.

4.3.4. Notificación de Metadatos

Finalmente, una vez que los bits decodificados se han interpretado en símbolos y agrupado en contenidos conceptualmente válidos, el bloque de Notificación de Metadatos se encarga de notificar al usuario que un nuevo metadato ha sido encontrado. Para el caso del presente proyecto dicha notificación consistirá en imprimir un mensaje a la salida estándar del sistema.

Similarmente, podría darse el caso en el que se ha decodificado un símbolo desconocido (ya sea por corrupción o por incoherencia entre ambos subsistemas). En este caso el bloque notificador notificará al usuario acerca del símbolo encontrado más una alerta de que el mismo es desconocido. Para el caso del presente proyecto dicha notificación consistirá en imprimir un mensaje a la salida de error del sistema.

5. Requerimientos del Sistema

5.1. Generales

R001. El sistema podrá ser implementado en los lenguajes de programación C o C++ o en un lenguaje de prototipado rápido como GNU/Octave o Python. El lenguaje de prototipado es recomendado.

R002. En entregable contará con todo el código fuente de autoría propia desarrollado para el sistema.

R003. El sistema deberá ser implementado para un sistema operativo basado en GNU/Linux.

R004. El entregable contará con un Makefile o script suficiente para construir la aplicación a partir del código fuente.

R005. El entregable deberá contar con el historial que demuestre el uso de un sistema de control de versiones durante el desarrollo del sistema.

5.2. Parámetros del Sistema

El comportamiento del sistema podrá ser configurado por el usuario mediante parámetros de ejecución.

R006. El usuario especificará los parámetros del sistema mediante argumentos al programa en la línea de comando.

R007. El sistema abortará su ejecución ante parámetros desconocidos.

En general, el usuario especificará parámetros de metadatos y de enmascaramiento.

5.2.1. Metadatos

Los parámetros de metadatos le indican al sistema qué tipo de información codificar en la señal. El usuario podrá elegir qué metadatos codificar en cada corrida, así el resultado podría contener uno, varios o todos los metadatos empotrados.

R008. Los parámetros de metadatos serán opcionales para la operación del sistema.

R009. Los parámetros de metadatos duplicados serán ignorados con excepción del último.

Como mínimo, el sistema deberá ser capaz de codificar y decodificar los siguientes metadatos:

Título de la canción

R010. El sistema brindará un parámetro para codificar el título de la canción.

Nombre del artista

R011. El sistema brindará un parámetro para codificar el nombre del artista interprete de la canción.

Autor original

R012. El sistema brindará un parámetro para codificar el nombre del autor original de la canción.

Nombre del álbum

R013. El sistema brindará un parámetro para codificar el nombre del álbum al que pertenece la canción.

Año de publicación

R014. El sistema brindará un parámetro para codificar el año en que la canción fue publicada.

Artistas invitado

R015. El sistema brindará un parámetro para codificar el artista invitado en la interpretación de la canción.

Productor

R016. El sistema brindará un parámetro para codificar el productor de la canción.

5.2.2. Enmascaramiento

Los parámetros de enmascaramiento le indican al sistema la naturaleza del eco a utilizar para codificar y decodificar los bits de metadatos. Un archivo codificado con ciertos valores de enmascaramiento solo podrá ser decodificado utilizando los mismos valores.

R017. El sistema operará con valores de enmascaramiento por defecto en caso que el usuario no especifique los mismos.

R018. Los parámetros por defecto serán aquellos que hayan presentado el mejor desempeño durante la implementación del sistema.

Específicamente, el usuario podrá configurar los siguientes parámetros de enmascaramiento.

Retardos

R019. El usuario especificará los retardos en microsegundos.

R020. El sistema brindará un parámetro para especificar el retardo de encodificación/decodificación de un bit en 0.

R021. El sistema brindará un parámetro para especificar el retardo de encodificación/decodificación de un bit en 1.

Atenuación

R022. El usuario especificará las atenuaciones en porcentajes con respecto a la señal original.

R023. El sistema brindará un parámetro para especificar la atenuación de encodificación/decodificación de un bit en 0.

R024. El sistema brindará un parámetro para especificar la atenuación de encodificación/decodificación de un bit en 1.

5.3. Formato de Audio

El sistema en general operará con archivos de audio en formato WAV. Este formato agrega unos encabezados a la señal en crudo por lo que es posible tanto reproducir el audio utilizando reproductores convencionales (Amarok, VLC, Totem, MPlayer...) como acceder al contenido de la señal original sin necesidad de un decodificador externo. De manera similar, el archivo

resultante con los metadatos empotrados será en formato WAV, usando la misma cabecera. En específico, se trabajará con archivos WAV con características de CD.

R025. El sistema operará a 44.100kHz, 16bps Little Endian, estéreo.

Al ser una señal estéreo, el usuario podrá elegir un canal al cuál codificarle los metadatos. De manera similar, el usuario podrá elegir el canal a decodificar. Así, un archivo podría contener metadatos diferentes en cada canal, y los mismos podrán ser decodificados independientemente.

R026. El usuario podrá especificar mediante un argumento en la línea de comandos el canal a codificar/decodificar.

R027. Si el usuario no especifica el argumento de canal, se trabajará por defecto en el canal izquierdo.

5.4. Codificador

Cuando el sistema opera en modo codificador, los metadatos son enmascarados en la señal original para su posterior decodificación.

R028. El sistema entrará en modo de codificación si el usuario proporciona al menos un parámetro de metadatos.

R029. El sistema codificador recibirá la ruta al archivo WAV original mediante el primer argumento posicional en la línea de comandos.

R030. El sistema codificador recibirá la ruta al archivo WAV codificado de destino en el segundo argumento posicional en la línea de comandos.

R031. El sistema terminará su ejecución cuando la totalidad del archivo haya sido codificado.

R032. Con los parámetros por defecto, la señal codificada será perceptualmente indistinguible de la original.

La elección del tamaño de la ventana de codificación incide directamente la caracterización de tasa de transferencia del sistema. Con el fin de estandarizar dicha caracterización, la elección del tamaño de la ventana deberá ser estricta.

R033. El sistema codificador segmentará el audio en ventanas del menor tamaño posible basado en el mayor retardo configurado para el sistema.

Si bien, la aplicación del sistema codificador puede resolverse utilizando diferentes mecanismos (ecuación de diferencias, dominio de la frecuencia, etc. . .) el sistema aplicará el mismo haciendo uso de la convolución.

R034. El sistema codificador será capaz de codificar los bits de metadatos mediante la convolución de una ventana con el sistema $h1(n)$ o $h0(n)$ respectivamente.

Debe tenerse especial cuidado con la longitud de la convolución. Si una ventana de señal de longitud M se convoluciona con una respuesta al impulso de longitud N , el resultado será de longitud $N+M-1$. Al haber truncado una señal de mayor duración en una ventana de menor duración, este excedente de $N-1$ deberá considerarse en la ventana siguiente. De no ser así, la señal se corromperá con componentes espectrales parásitas afectando la calidad del audio y, posiblemente, la integridad de la codificación. Únicamente uno de los dos requerimientos posteriores aplica.

R035. a. El sistema codificador combinará la codificación de las ventanas mediante solapamiento y almacenamiento

R036. b. El sistema codificador combinará la codificación de las ventanas mediante solapamiento y suma.

Finalmente el archivo resultante, luego de combinar todas las ventanas, se almacenará en el archivo WAV especificado por el usuario utilizando el mismo formato establecido.

R037. El encabezado del archivo WAV codificado deberá ser equivalente al encabezado del archivo WAV resultante.

R038. El archivo codificado podrá escucharse en reproductores de audio estándar (Amarok, VLC, MPlayer, Totem...).

5.5. Decodificador

Cuando el sistema entra en modo de decodificación, se extraerán los metadatos de un archivo WAV previamente codificado.

R039. El sistema entrará en modo de decodificación si el usuario no proporciona ningún parámetro de metadatos.

R040. El sistema decodificador recibirá la ruta al archivo WAV codificado mediante el primer argumento posicional en la línea de comandos.

R041. El sistema terminará su ejecución cuando la totalidad del archivo haya sido decodificado

Similar al encodificador el tamaño de la ventana será definido en términos de los retardos.

R042. El sistema decodificador segmentará el audio en ventanas del menor tamaño posible basado en el mayor retardo configurado para el sistema.

El proceso de decodificación se da encontrando el eco que fue agregado a la respectiva ventana durante el proceso de codificación. Si bien existen múltiples maneras de clasificar dicho eco, el sistema deberá hacer uso de la autocorrelación del Cepstro.

R043. El sistema decodificador clasificará cada ventana como un 1 o 0 según el eco encontrado.

R044. El sistema decodificador deberá utilizar la autocorrelación del Cepstro para clasificar el bit de la ventana.

R045. La autocorrelación del cepstro debe evaluarse en el dominio espectral y no evaluar la autocorrelación temporal.

R046. Con los parámetros por defecto, el decodificador tendrá una tasa de recuperación igual o mayor al 90 %.

Es posible que no todas las ventanas contengan bits encodificados. El sistema de decodificación deberá ser lo suficientemente robusto para reconocer ventanas carentes de bit (sin eco).

R047. El sistema decodificador deberá reconocer ventanas no codificadas.

Finalmente, el usuario es notificado acerca de los metadatos encontrados.

R048. El sistema deberá imprimir los metadatos encontrados en la salida estándar.

R049. Los metadatos irán acompañados de su respectiva categoría.

R050. El sistema deberá imprimir los metadatos no reconocidos en la salida de error.

R051. Los metadatos desconocidos irán acompañados con una advertencia.

6. Artículo Científico

6.1. Formato

El diseño, la implementación y los resultados del proyecto se deben resumir en un artículo científico. Éste se apegará al formato de la IEEE Transactions cuyas especificaciones y plantilla se puede encontrar en [5].

A001. El artículo científico seguirá el formato de IEEE Transactions (contenido y forma).

A002. El artículo científico tendrá una extensión no mayor a 7 páginas.

6.1.1. Consideraciones de Forma

A003. Sólo se deben utilizar palabras en español.

A004. No se deben utilizar siglas sin antes introducirlas.

A005. Tablas, imágenes, ecuaciones y demás auxiliares deben ser introducidos previo a su aparición.

A006. Los nombres de las tablas, imágenes, ecuaciones y demás auxiliares inician con mayúscula. E.g.: Tabla 1. Ejemplo.

A007. Con excepción de las tablas, todos los auxiliares llevan el título debajo del mismo.

A008. Las tablas llevan el título por encima de la misma.

A009. Las tablas no llevan divisiones verticales.

A010. Las tablas únicamente llevan divisiones horizontales al inicio, final y entre los encabezados y los datos.

A011. No se debe poner código textual. Se utilizan diferentes diagramas para representar el código en cuestión.

A012. Las palabras clave van en orden alfabético.

A013. El resumen se escribe en inglés.

6.1.2. Consideraciones de Contenido

A014. El resumen debe proveer información necesaria para que un lector pueda decidir si el artículo le será de utilidad o no.

A015. Las palabras claves debes ser tales que representen el contenido principal de proyecto desarrollado.

A016. La introducción brinda contexto, trabajos relacionados y estado del arte con respecto al tema.

A017. La introducción finaliza describiendo la estructura del documento.

A018. Las secciones posteriores exponen la teoría necesaria para entender la solución propuesta en el proyecto.

A019. La solución propuesta describe cada módulo con detalle suficiente para que el lector pueda reproducir la implementación.

A020. El análisis de resultados presenta las diferentes respuestas del sistema ante distintas configuraciones y escenarios con el fin de caracterizar el mismo.

A021. El análisis de resultados relaciona los resultados obtenidos con la teoría que respalda el diseño.

A022. Se deben analizar resultados para los diferentes componentes que conforman el sistema.

A023. Se brindarán conclusiones únicamente para resultados presentados en el artículo.

A024. Se brindará una conclusión por cada resultado analizado.

A025. Las conclusiones finalizan con recomendaciones para trabajos futuros.

A026. El artículo debe contener las citas a las diferentes referencias bibliográficas consultadas.

6.2. Puntos a Analizar

Entre otros, se debe presentar análisis sobre los siguientes puntos:

A027. Se utilizará PEAQ [6, 7] para evaluar la calidad perceptual de archivo codificado ante diferentes configuraciones del sistema.

A028. Se medirá la tasa de recuperación del sistema ante diferentes configuraciones del sistema.

A029. Se correlacionará la tasa de recuperación con la calidad perceptual del archivo codificado.

A030. Se correlacionará la tasa de transferencia de metadatos con la calidad perceptual del archivo codificado.

A031. Se experimentará con archivos que contengan diferentes tipos de géneros musicales.

A032. El estudiante propondrá el análisis de otros resultados obtenidos.

7. Trabajo opcional

Se asignará un máximo de 10 % extra sobre la calificación del Proyecto 1 (aplicable sobre la calificación global, en el rubro de proyectos únicamente) al los siguientes ítemes:

- O001.** Codificación de la imagen de la portada del disco.
- O002.** Sistema de corrección de errores.
- O003.** Ejecución del sistema en línea (vía parlante y micrófono).

8. Evaluación

El proyecto será evaluado haciendo uso de la evaluación presentada en la Tabla 1.

Tabla 1: Evaluación del proyecto

Rubro	Item	(%)
Requerimientos (65 %)	R001	—
	R002	—
	R003	—
	R004	—
	R005	2.5
Parámetros	R006	—
	R007	0.5
Metadatos (12 %)	R008	0.5
	R009	1
	R010	1.5
	R011	1.5
	R012	1.5
	R013	1.5
	R014	1.5
	R015	1.5
	R016	1.5
Enmascaramiento (5 %)	R017	5
	R018	
	R019	
	R020	
	R021	
	R022	
	R023	

	R024	
Formato (5 %)	R025	4
	R026	0.5
	R027	0.5
Codificación (20 %)	R028	0.5
	R029	0.5
	R030	0.5
	R031	0.5
	R032	2.5
	R033	0.5
	R034	7.5
	R035	7.5
	R036	7.5
	R037	—
	R038	—
Decodificación (20 %)	R039	0.5
	R040	0.5
	R041	0.5
	R042	0.5
	R043	4
	R044	7.5
	R045	—
	R046	—
	R047	3.5
	R048	1
	R049	0.5
	R050	1
	R051	0.5
Artículo Científico (35 %)	A001	—
	A002	—
Forma (5 %)	A003	5
	A004	
	A005	
	A006	
	A007	
	A008	
	A009	
	A010	

	A011	
	A012	
	A013	
Contenido (16 %)	A014	1
	A015	0.5
	A016	2
	A017	0.5
	A018	4
	A029	4
	A020	—
	A021	—
	A022	—
	A023	0.5
	A024	2
	A025	0.5
	A026	1
Análisis (14 %)	A027	3
	A028	3
	A029	2
	A030	2
	A031	2
	A032	2
Trabajo Opcional (15 %)	O001	5
	O002	5
	O003	5

Referencias

- [1] T. Kim P. Dutta, D. Bhattacharyya. Data hiding in audio signal: a review. *International Journal of Database Theory and Application*, 2, 2009. Available from: http://www.sersc.org/journals/IJDTA/vol2_no2/1.pdf.
- [2] S. Behal N. Kaur. Audio steganography techniques - a survey. *International Journal of Engineering Research and Applications*, 4:94–100, 2014. Available from: http://www.ijera.com/papers/Vol4_issue6/Version%205/P0460594100.pdf.

- [3] A. Lu D. Gruhl, W. Bender. Echo hiding. Technical report, Massachusetts Institute of Technology Media Laboratory. Available from: <https://pdfs.semanticscholar.org/b964/4dbf393117b4c4bb91122cf7b08f71b71fbd.pdf>.
- [4] V. Korzhik. Audio watermarking based on echo hiding with zero error probability. *International Journal of Computer Science and Applications*, 10:1–10, 2013. Available from: <http://www.tmrfindia.org/ijcsa/v10i11.pdf>.
- [5] IEEE. Article templates and instructions, 2017. Available from: https://www.ieee.org/publications_standards/publications/authors/author_templates.html.
- [6] ITU Radiocommunication Assembly. Method for objective measurements of perceived audio quality. Technical report, ITU, 1998-2001. Available from: https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1387-1-200111-I!!PDF-E.pdf.
- [7] ITU Radiocommunication Assembly. Bs.1387 : Method for objective measurements of perceived audio quality, 1998-2001. Available from: <https://www.itu.int/rec/R-REC-BS.1387-1-200111-I/en>.