1. Objetivo

El objetivo de este estudio es observar las diferencias en la exportación de audios en distintos formatos y utilizar estas diferencias para obtener conclusiones sobre qué formatos pueden dar salidas que afecten a otras etapas, como el preprocesado del audio, extracción de características o la clasificación de las señales. Es por ello por lo que este estudio es muy importante en cuanto al proyecto se refiere, ya que una buena selección del formato del audio puede ser una gran mejora para el clasificador.

Aun así, al no conocer con certeza como los clasificadores usan la información que le mandamos no podemos saber a ciencia cierta si conceptos que se van a hablar en este estudio afectan a la clasificación o en qué medida lo hacen.

2. Conceptos

1. Conceptos de Audio, Códecs, Contenedores.

Bitrate: es una tasa que marca la cantidad de contenido o dato se procesa por unidad de tiempo, es por ello que sus unidades son Bits/Segundos o bps.

Sampling rate o frecuencia de muestreo: es la tasa que nos marca la frecuencia con la que se capturan datos de entrada, en nuestro caso por el micrófono, su unidad es Muestra/Segundo -> 1/s -> Hz.

Códec (Wikipedia): Un códec es un programa o dispositivo hardware capaz de codificar o decodificar una señal o flujo de datos digitales

Códec de audio (Wikipedia): Un códec de audio es un códec que incluye un conjunto de algoritmos que permiten codificar y decodificar los datos auditivos. Sirve para comprimir señales o ficheros de audio con un flujo de datos con el objetivo de que ocupen el menor espacio posible.

Formato contenedor: Un formato contenedor es un tipo de formato de archivo que almacena información de vídeo, audio, metadatos e información de sincronización y corrección de errores siguiendo un formato preestablecido en su especificación técnica.

https://en.wikipedia.org/wiki/Audio_file_format

Con estas definiciones, el objetivo del punto 1 cambia, ya que un contenedor (por ejemplo, 3gp) puede tener uno u otro códec, y lo que afecta a la señal final va a ser el códec.

Aun así, tanto el contenedor como el códec pueden afectar al tamaño final del archivo. El códec porque comprime y el contenedor porque tiene información de corrección de errores, número de pistas de audio, información del códec que se debe usar…

La siguiente página: https://developer.android.com/guide/topics/media/media-formats muestra los códecs que ofrece Android (Al menos de forma nativa, aunque puede que haya formas de incluir de otros códecs pero puede ser costoso, complicado o poco productivo).

Format / Codec	Encoder	Decoder	Details	Supported File Type(s) / Container Formats
AAC LC	•	•	Support for mono/stereo/5.0/5.1 content with standard sampling rates from 8 to 48 kHz.	• 3GPP (.3gp) • MPEG-4 (.mp4, .m4a) • ADTS raw AAC (.aac, decode in Android 3.1+, encode in Android 4.0+, ADIF not supported) • MPEG-TS (.ts, not seekable, Android 3.0+)
HE-AACv1 (AAC+)	• (Android 4.1+)	•		
HE-AACv2 (enhanced AAC+)		•	Support for stereo/5.0/5.1 content with standard sampling rates from 8 to 48 kHz.	
AAC ELD (enhanced low delay AAC)	• (Android 4.1+)	• (Android 4.1+)	Support for mono/stereo content with standard sampling rates from 16 to 48 kHz	
AMR-NB	•	•	4.75 to 12.2 kbps sampled @ 8kHz	3GPP (.3gp)
AMR-WB	•	•	9 rates from 6.60 kbit/s to 23.85 kbit/s sampled @ 16kHz	3GPP (.3gp)
FLAC	• (Android 4.1+)	• (Android 3.1+)	Mono/Stereo (no multichannel). Sample rates up to 48 kHz (but up to 44.1 kHz is recommended on devices with 44.1 kHz output, as the 48 to 44.1 kHz downsampler does not include a low-pass filter). 16-bit recommended; no dither applied for 24-bit.	FLAC (.flac) only
GSM		•	Android supports GSM decoding on telephony devices	GSM(.gsm)
MIDI		•	MIDI Type 0 and 1. DLS Version 1 and 2. XMF and Mobile XMF. Support for ringtone formats RTTTL/RTX, OTA, and iMelody	• Type 0 and 1 (.mid, .xmf, .mxmf) • RTTTL/RTX (.rtttl, .rtx) • OTA (.ota) • iMelody (.imy)
MP3		•	Mono/Stereo 8-320Kbps constant (CBR) or variable bit-rate (VBR)	MP3 (.mp3)
Opus		• (Android 5.0+)		Matroska (.mkv)
PCM/WAVE	• (Android 4.1+)	•	8- and 16-bit linear PCM (rates up to limit of hardware). Sampling rates for raw PCM recordings at 8000, 16000 and 44100 Hz.	WAVE (.wav)
Vorbis		•		• Ogg (.ogg) • Matroska (.mkv, Android 4.0+)

De esta forma, los formatos en los que sería posible grabar son:

ACC LC, HE-AACV1 (AAC+), ACC ELD, AMR-NB, AMR-WB, FLAC y PCM/WAVE

Se va a usar para grabar en Android la clase Media Recorder, que en sus códecs tiene los siguientes:
https://developer.android.com/reference/android/media/MediaRecorder.AudioEncoder

Constants	
int	AAC AAC Low Complexity (AAC-LC) audio codec
int	AAC_ELD Enhanced Low Delay AAC (AAC-ELD) audio codec
int	AMR_NB AMR (Narrowband) audio codec
int	AMR_WB AMR (Wideband) audio codec
int	DEFAULT
int	HE_AAC High Efficiency AAC (HE-AAC) audio codec
int	VORBIS Ogg Vorbis audio codec

(VORBIS es la única que aparece en esta lista y no en la otra, por lo que puede que en alguna actualización se haya añadido y sí que se pueda utilizar)

Y sus contenedores:

https://developer.android.com/reference/android/media/MediaRecorder.OutputFormat

Constants	
int	AAC_ADTS AAC ADTS file format
int	AMR_NB AMR NB file format
int	AMR_WB AMR WB file format
int	DEFAULT
int	MPEG_2_TS H.264/AAC data encapsulated in MPEG2/TS
int	MPEG_4 MPEG4 media file format
int	RAW_AMR *This constant was deprecated in API level 16. Deprecated in favor of MediaRecorder.OutputFormat.AMR_NB*
int	THREE_GPP 3GPP media file format
int	WEBM VP8/VORBIS data in a WEBM container

Usando esta información, los formatos en los que podemos grabar, y vamos a estudiar son:

ACC-LC, ACC-ELD, HE_AAC (Suponiendo que es lo mismo que a HE_ACC1), **AMR_NB, AMR_WB.**

2. Tipos de Códecs.

Se pueden dividir en Lossy (con pérdidas) o Loseless (sin pérdidas)

CODEC	Tipo de cifrado
AAC-LC	Lossy
AAC-ELD	Lossy
HE_AAC	Lossy
AMR_NB	Lossy
AMR_WB	Lossy
PCM (Pulse Code Modulations)	Loseless
FLAC	Loseless

Todos los formatos que permite Media Recorder de Android tienen pérdidas, en comparación con otros códecs, como FLAC o PCM (que es el audio en bruto, sin cabeceras que indiquen la frecuencia de muestreo o el número de canales de audio).

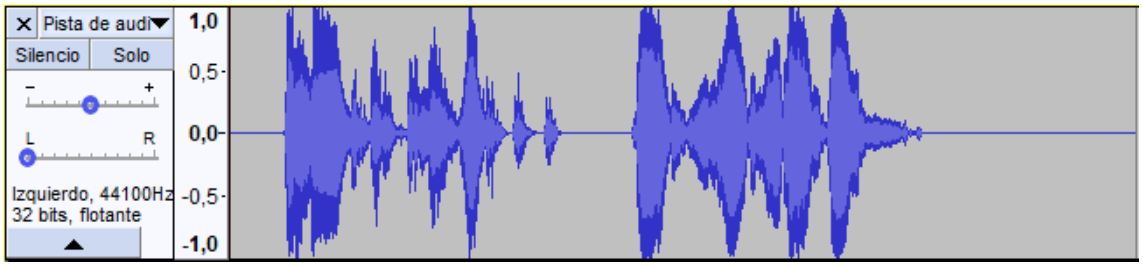
Como todos los códecs que permite Android son con pérdidas, se intentará comparar entre ellos para poder ver el formato que menos pérdida tenga que llegue el mayor número de información a las siguientes fases del proceso.

El proceso óptimo sería realizar la grabación en un formato sin pérdidas, y posteriormente re-codificar el audio usando distintos códecs para obtener el conjunto de datos completo en otro formato de audio. El proceso contrario no se puede realizar: No se puede obtener un audio sin pérdidas a partir de un audio con pérdidas, y recodificaciones de un códec con pérdidas en otros códecs con pérdidas pueden hacer que el audio quede distorsionado y no pueda ser usado para la clasificación.

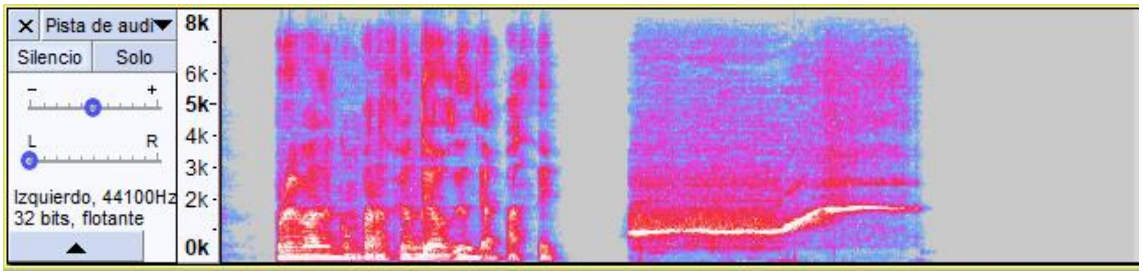
https://en.wikipedia.org/wiki/Comparison_of_audio_coding_formats

3. Representaciones de audio

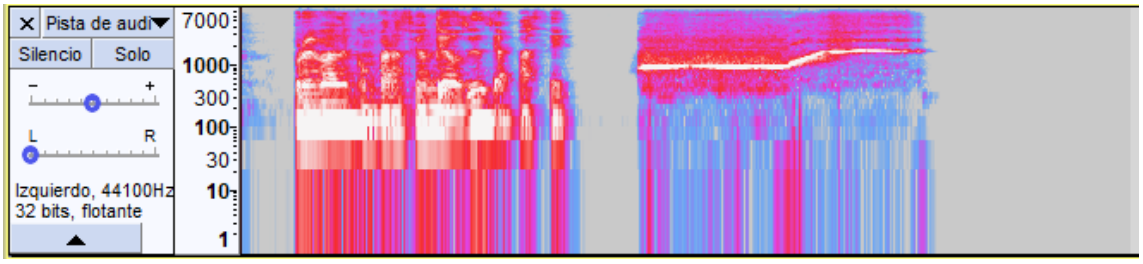
Forma de onda:



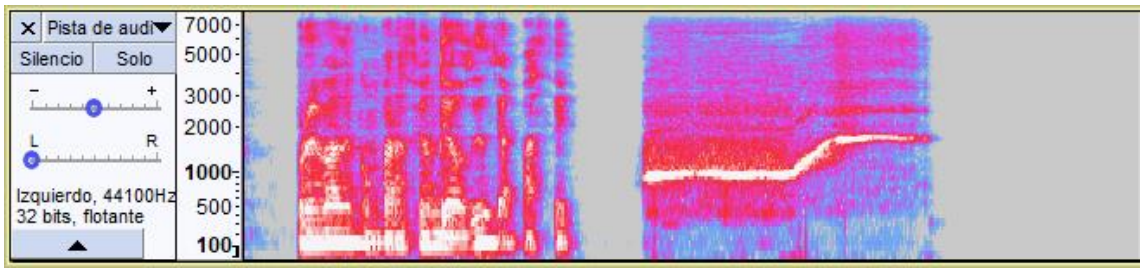
Espectrograma escala lineal:



Espectrograma escala logarítmica:



Espectrograma escala Mel:



Grabación de voz de prueba, constituida de una primera parte hablada y después un silbido.

La forma de onda muestra la amplitud de la onda en cada instante, este tipo de representación no muestra suficiente información para poder realizar la clasificación. Por ello se utiliza el espectrograma, que aplica las transformadas de Fourier para obtener las amplitudes de las distintas ondas que forman la señal de audio.

En las imágenes se ven tres tipos de espectrogramas con los mismos parámetros (por defecto), pero cambiando su escala de medición.

El color rojo indica donde se encuentra la mayor concentración de frecuencias (Hercios/tiempo) **(No estoy seguro de como definirlo)**.

En una representación lineal las frecuencias altas están sobrerrepresentadas, frente a las bajas que son donde se encuentra la mayor parte de la información que define el sonido.

En una representación logarítmica, las frecuencias más bajas también están sobrerrepresentadas, y las frecuencias donde se encuentra la mayor parte de información del audio pueden no estar bien representadas.

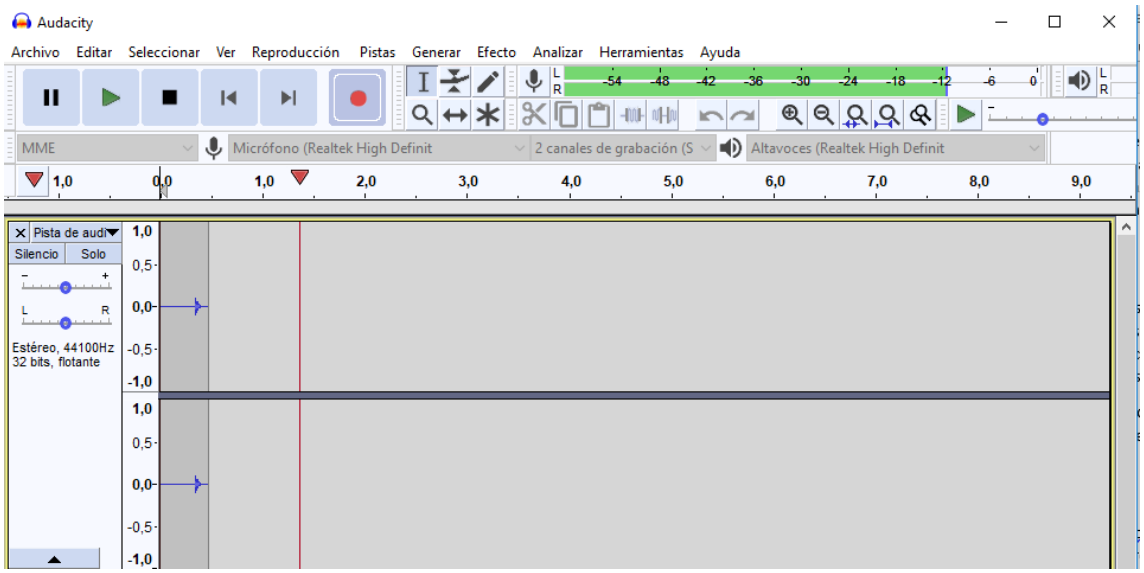
En diversos papers (https://arxiv.org/abs/1709.01922 , apartado 2.2) indican que para conjuntos pequeños de datos, un espectrograma con escala Mel puede ser suficiente para realizar la clasificación. Esta escala hace que las frecuencias más bajas y más altas queden menos representadas, y da más importancia a aquellas frecuencias habituales en la voz humana.

3. Proceso del estudio

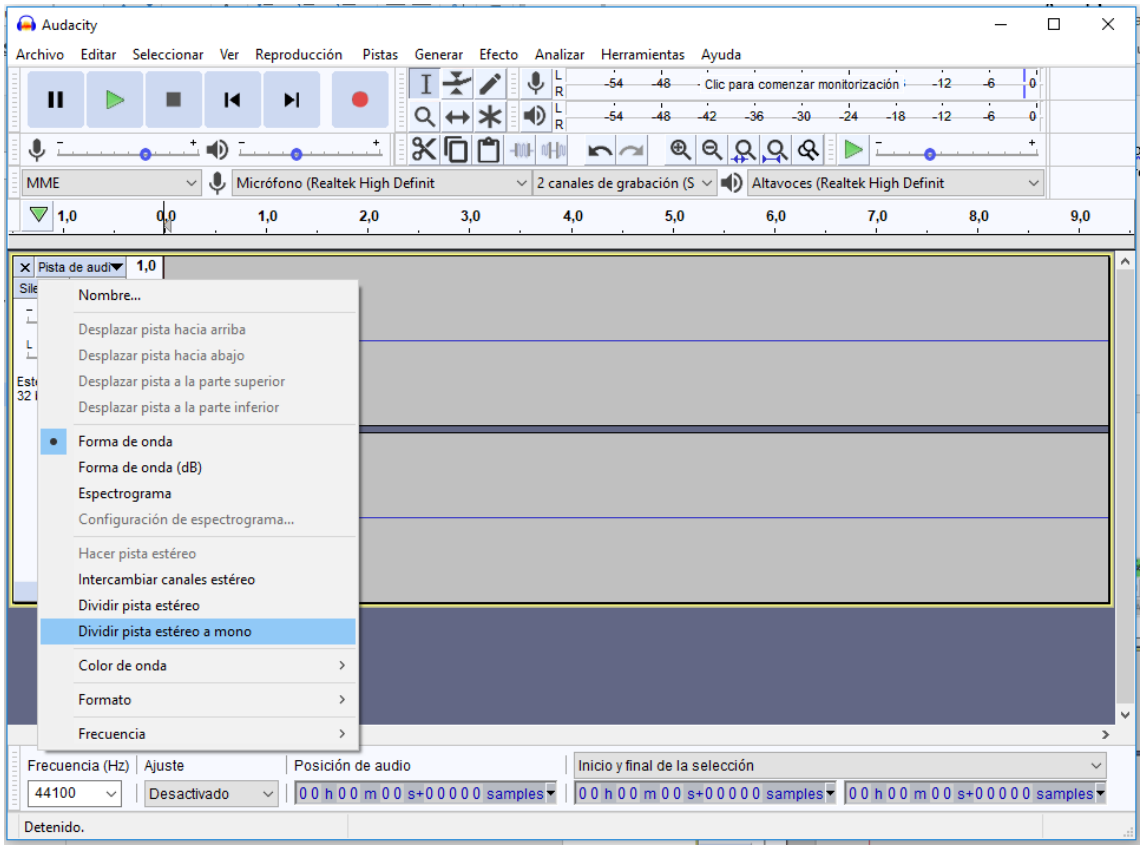
El programa Audacity permite tanto exportar en distintos formatos como con distintos códecs de forma fácil. A parte permite importar distintas pistas de audio, realizar distintas operaciones sobre ellas, mostrar espectrogramas de forma sencilla, etc. Por lo tanto, se puede utilizar de forma fácil y rápida para replicar los resultados de este estudio.

**Se necesita descargar e instalar la librería dll ffmpeg para poder exportar en los formatos necesarios, al intentar exportar en uno de estos formatos aparece información de cómo descargar e importar la librería.

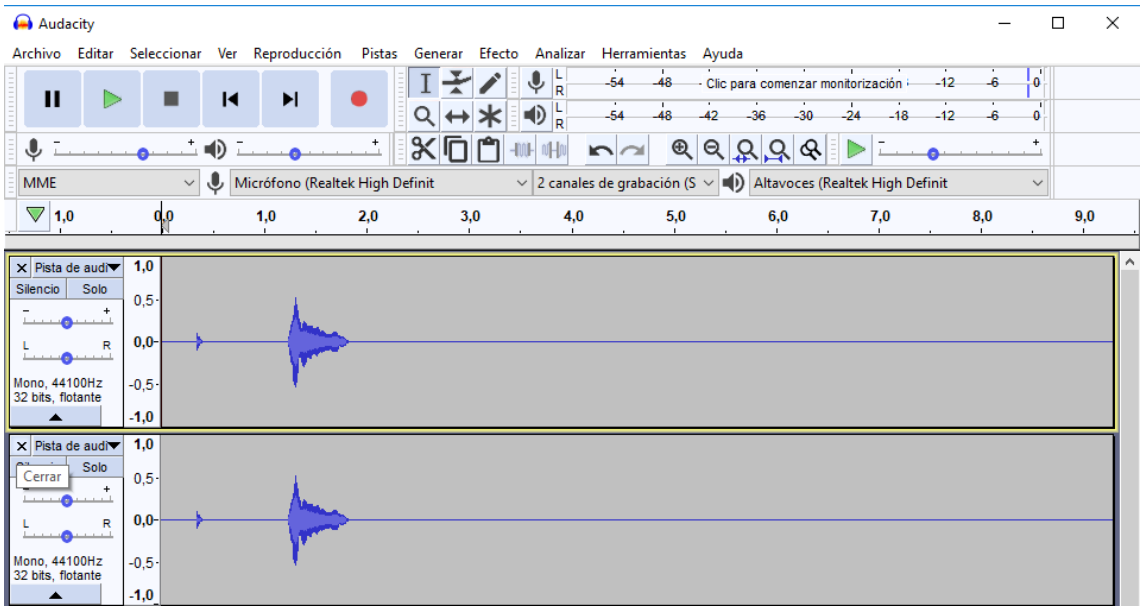
Paso 1: Grabación de audio de prueba (O importación, pista mono para simplificar, si se graba en estéreo se puede hacer click en la opción “dividir pista estéreo” y eliminar una de las pistas)



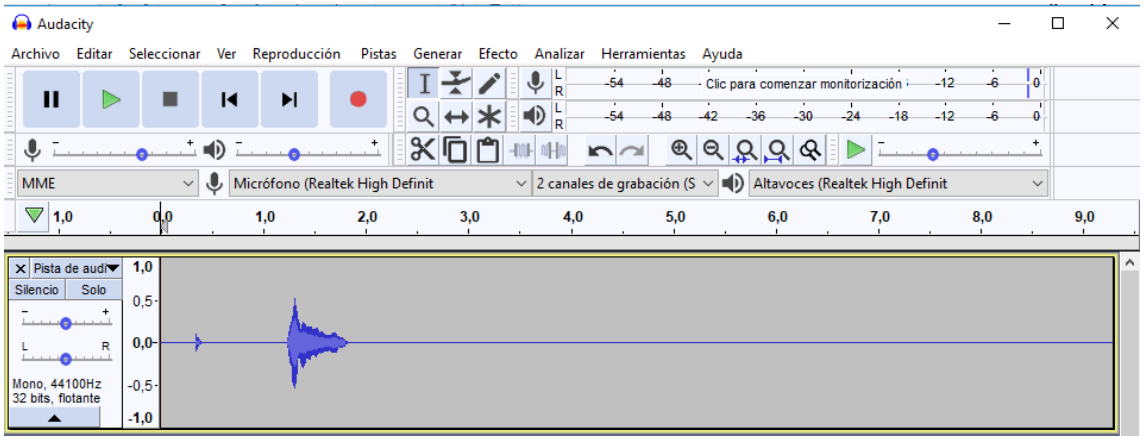
Grabación estéreo



División de estéreo a mono

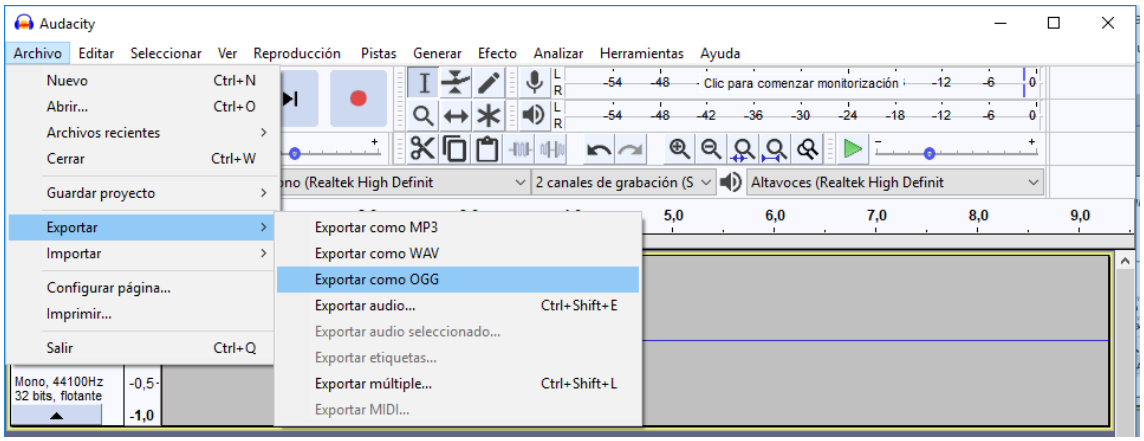


Cierre de una de las pistas para trabajar sobre uno de los dos canales



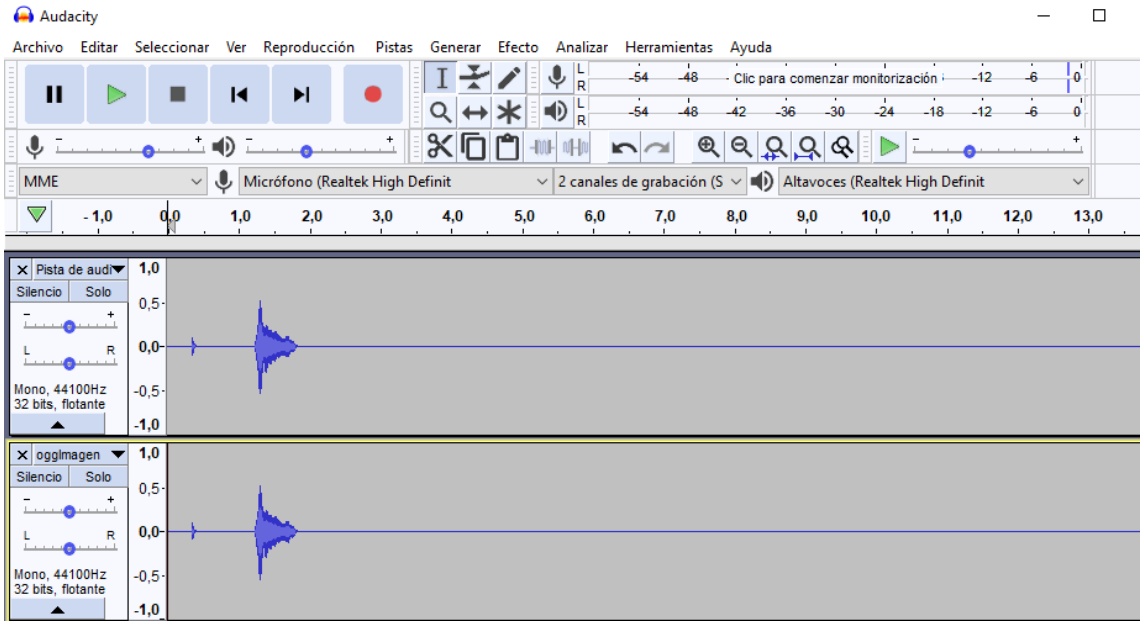
Baseline del audio (Mono)

Paso 2: Exportación en distintos formatos usando Archivo -> Exportar -> Exportar Audio.



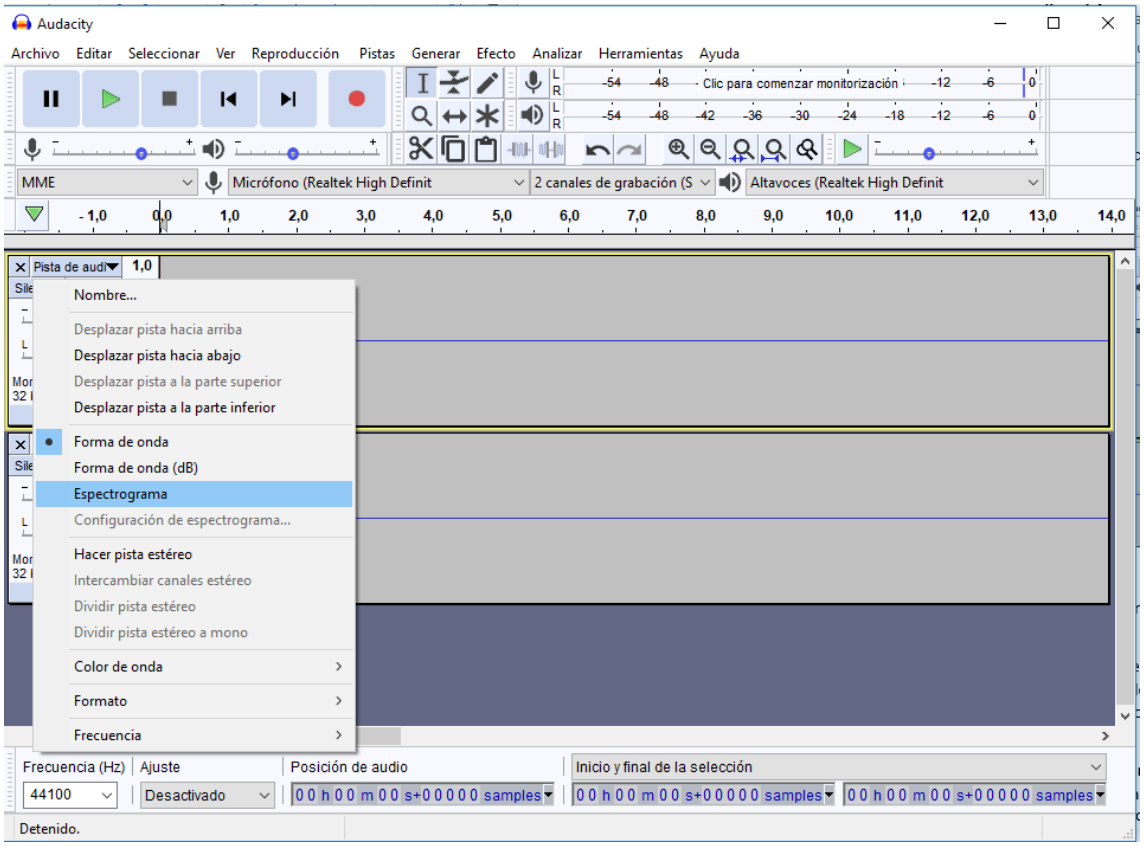
Exportación de Audio, OGG Vorbis es un formato con pérdidas

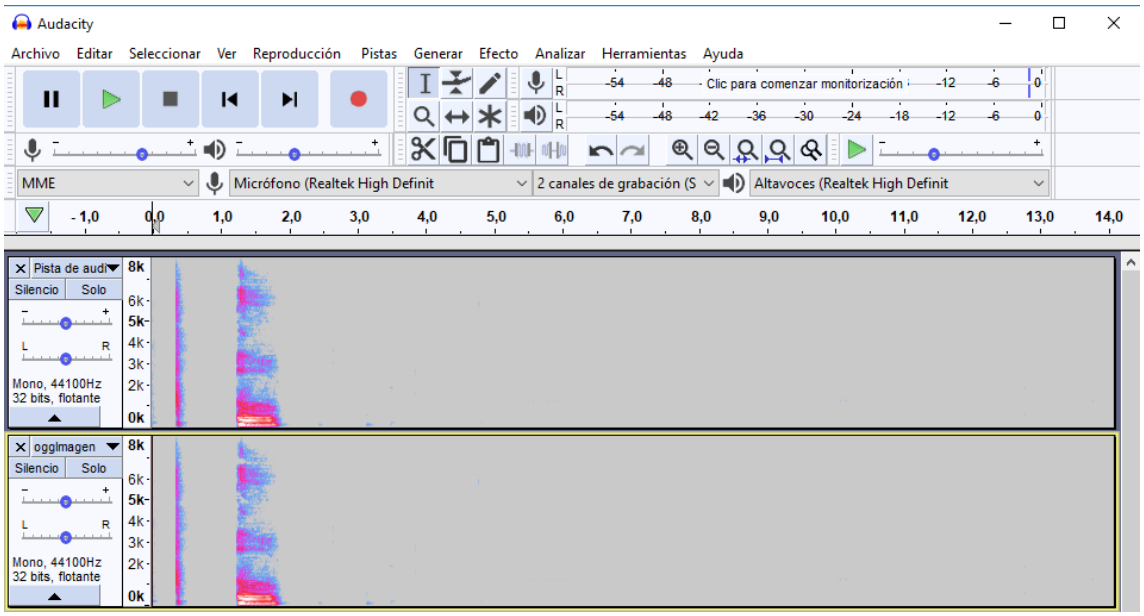
Paso 3: Importación de los distintos audios, se irán añadiendo como pistas distintas.



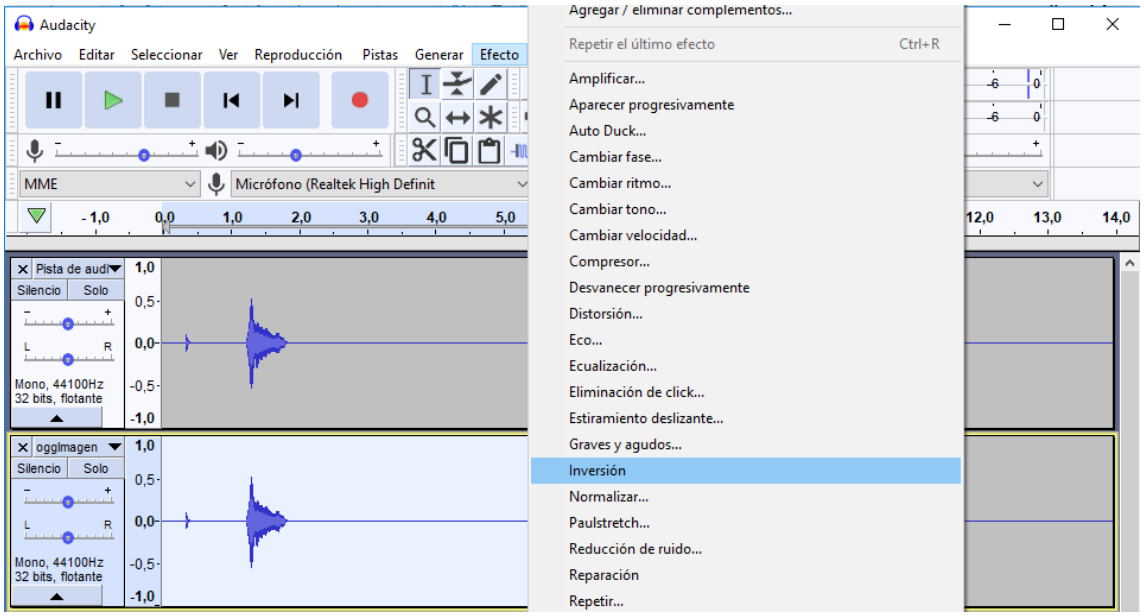
Audio original (Superior), Audio en OGG Vorbis (Inferior)

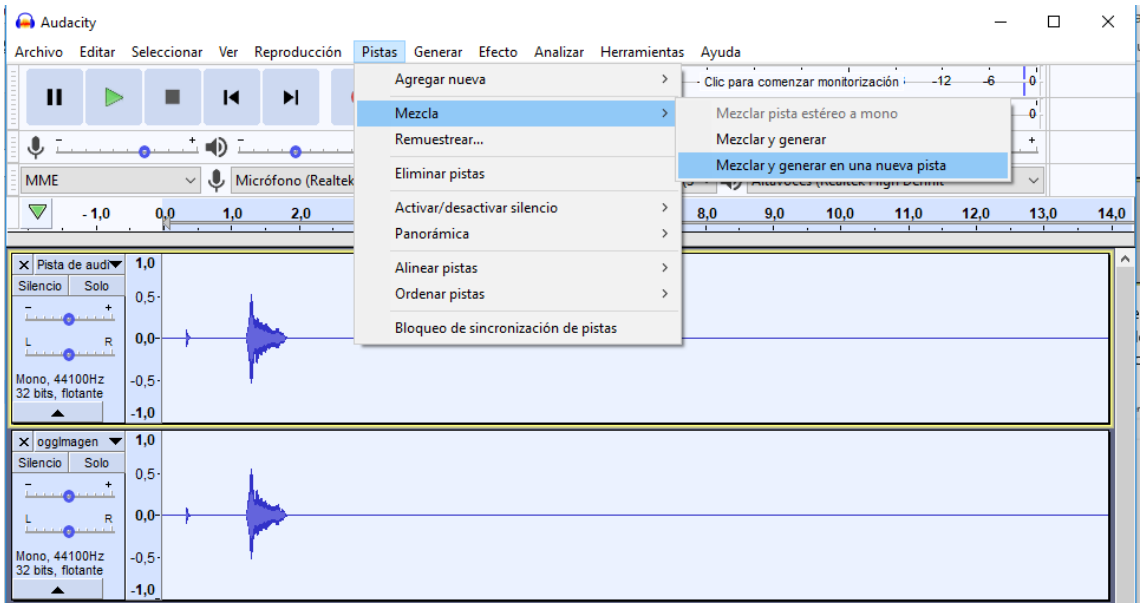
Paso 4: Se puede usar la vista de espectrograma para ver diferencias (Aunque seguramente no se aprecie ninguna)



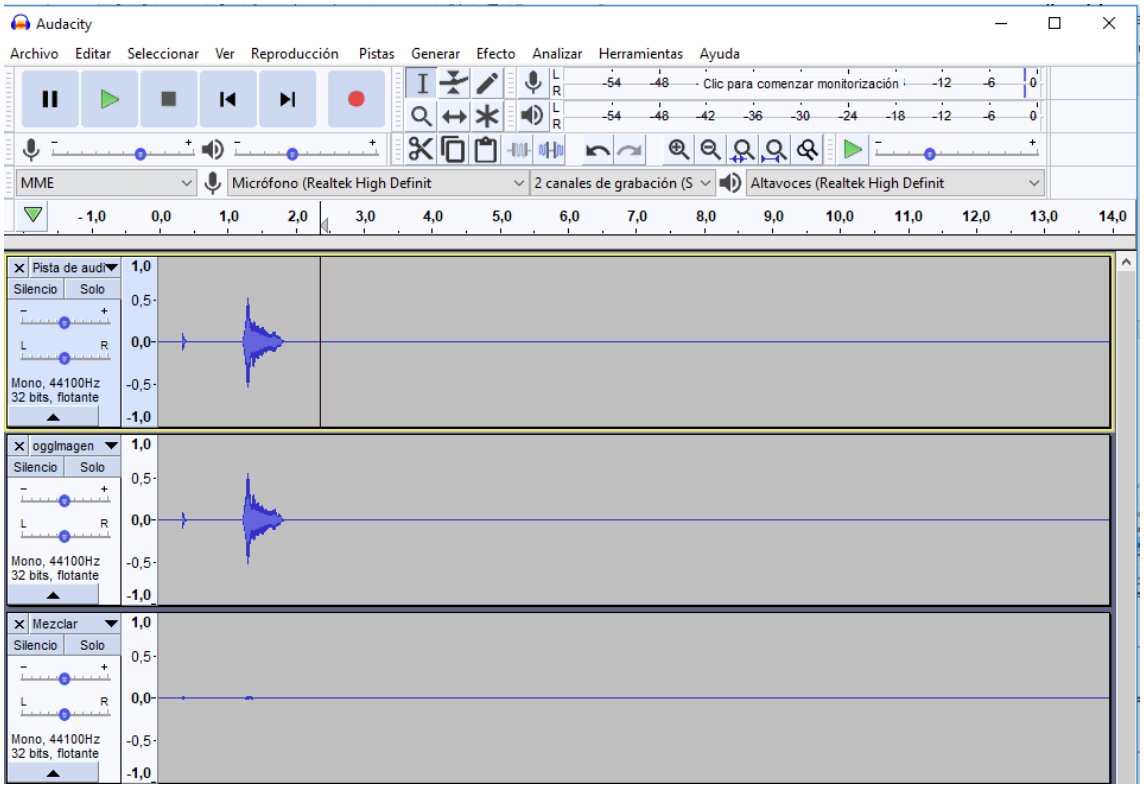


Paso 5: Para ver la diferencia entre dos audios, se realiza la operación Inversión de una de las pistas (Pista -> Invertir) esto hace que, al sumar dos señales iguales, una invertida y la otra no, se resten y se anulen. Por eso haciendo esta acción, sobre el audio original y los audios exportados se pueden ver las diferencias entre los sonidos.

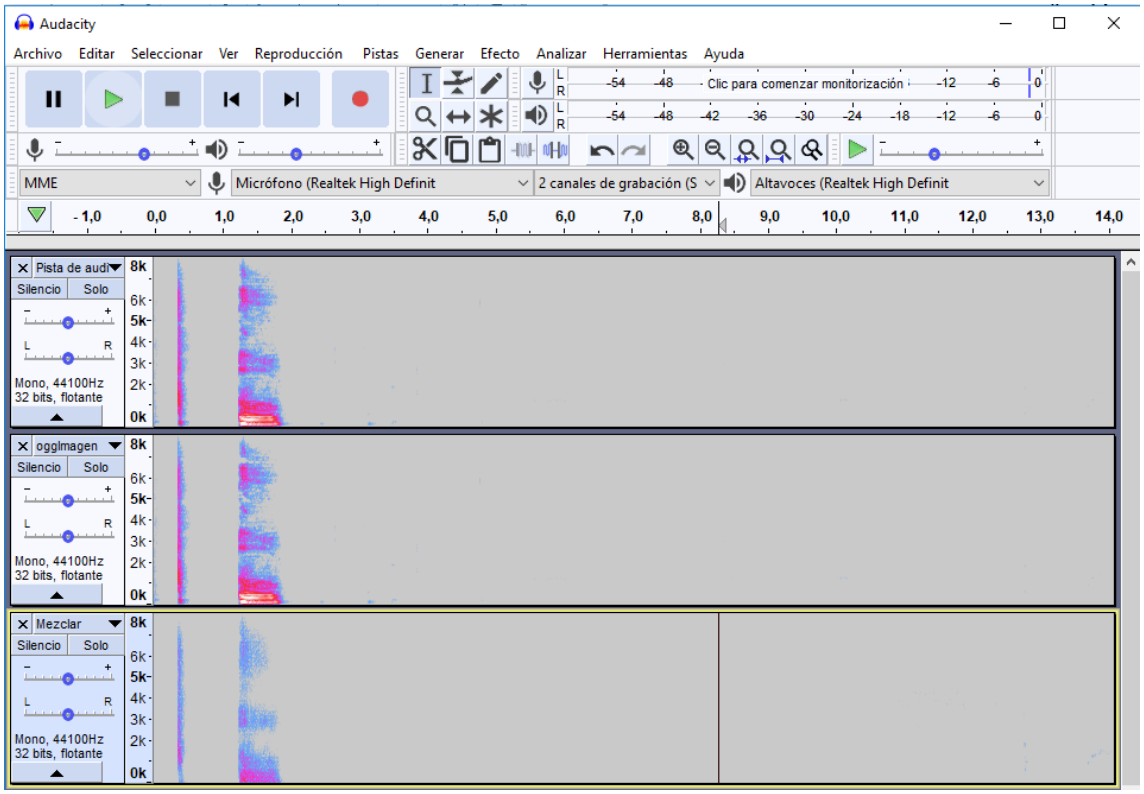




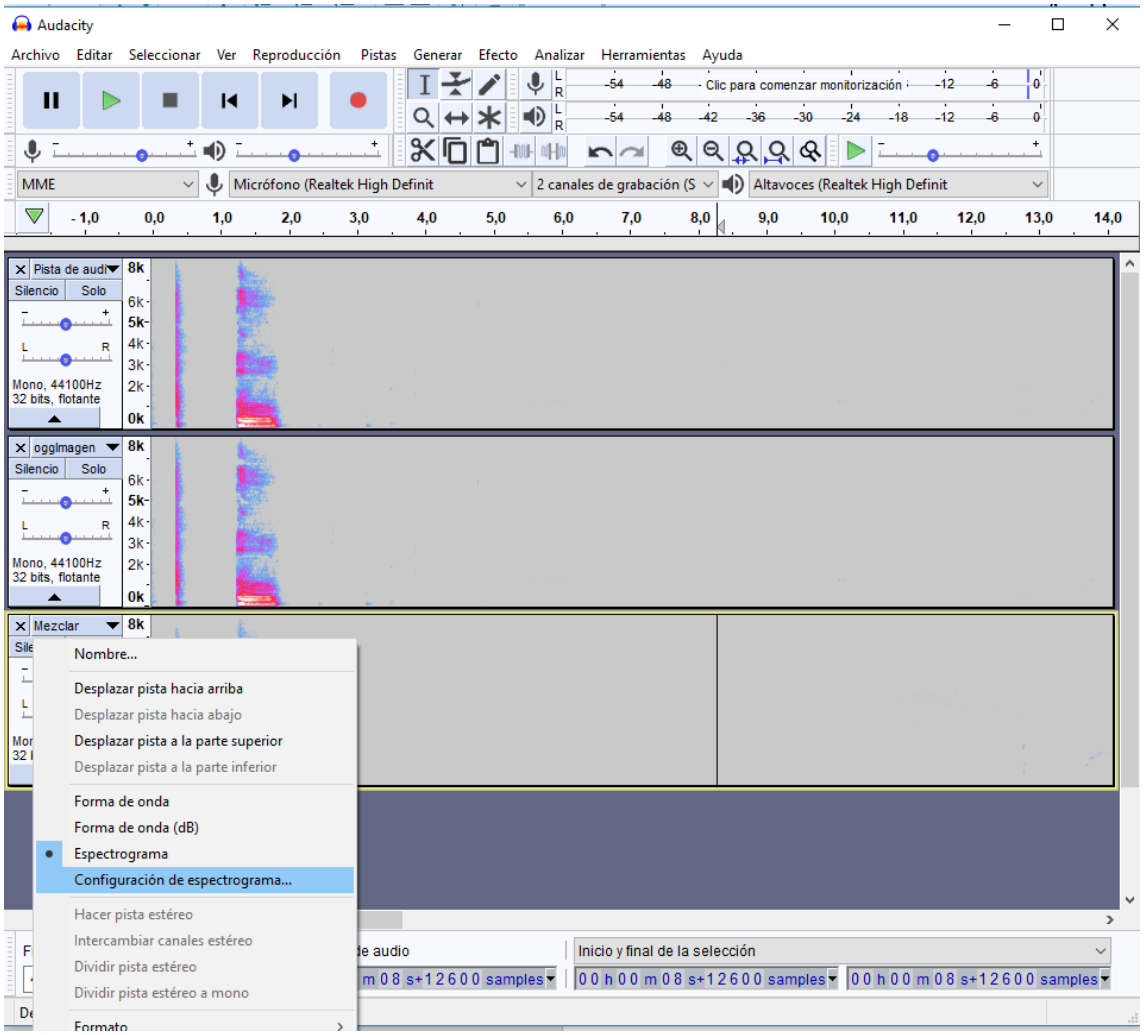
Si son Loseless debería haber silencio, si tiene pérdida, se debería ver la diferencia tanto en la forma de onda como con en el espectrograma.

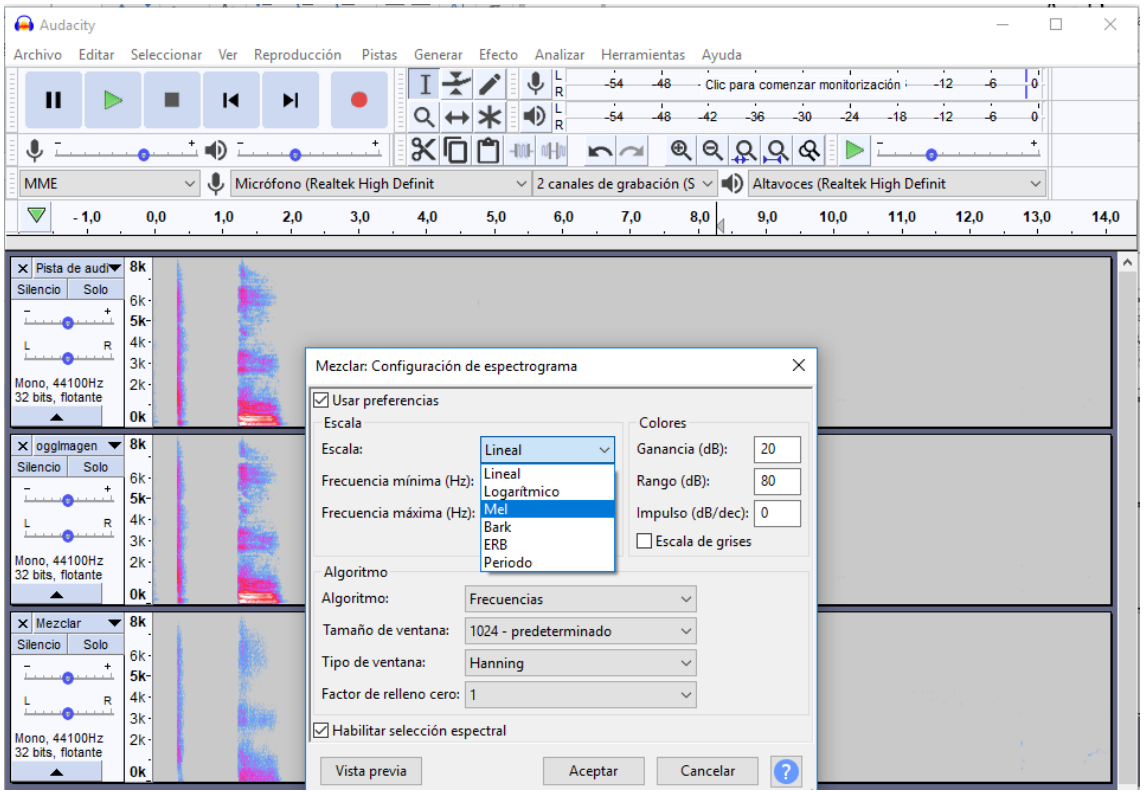


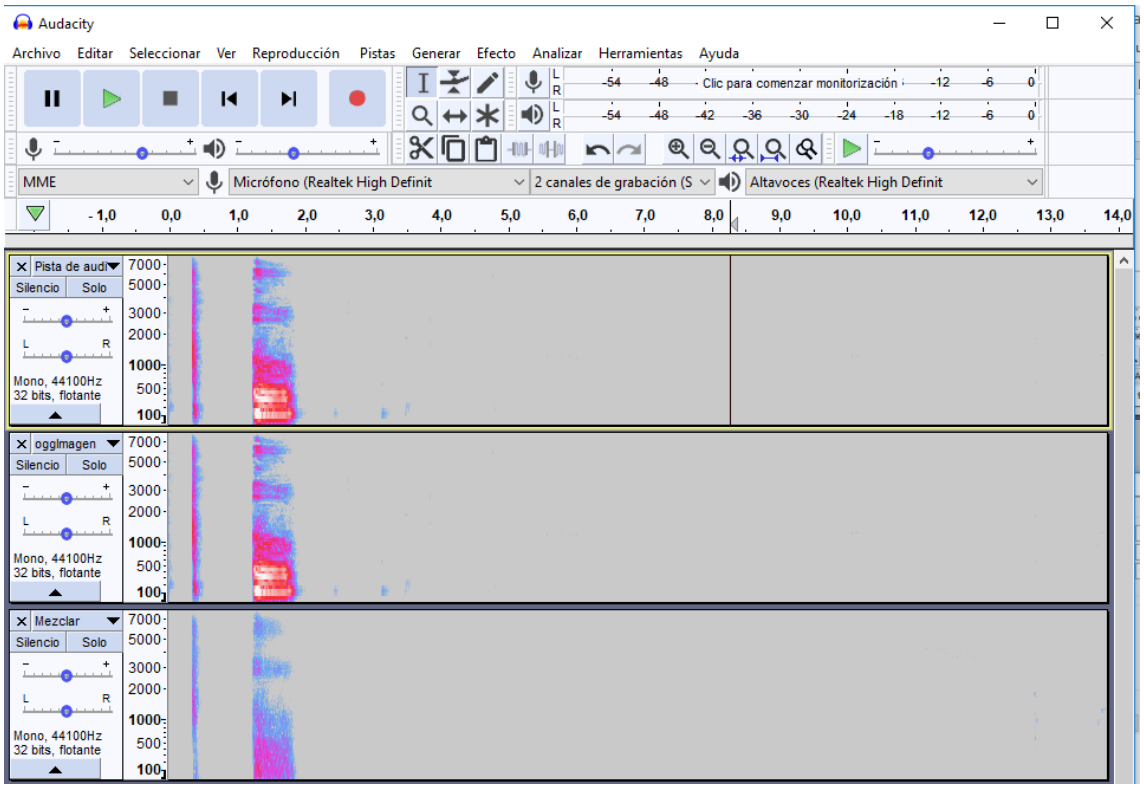
Resta de ambas ondas en el tercer canal, más visible si se usa el espectrograma.



Para cambiar la unidad de medida del espectrograma, se selecciona la configuración del espectrograma.


Audacity
Archivo Editar Seleccionar Ver Reproducción Pistas Generar Efecto Analizar Herramientas Ayuda
Clic para comenzar monitorización
MME
Micrófono (Realtek High Definit
2 canales de grabación (S
Altavoces (Realtek High Definit
Pista de audi
Silencio Solo
Mono, 44100Hz
32 bits, flotante
oggImagen
Mezclar
Nombre...
Desplazar pista hacia arriba
Desplazar pista hacia abajo
Desplazar pista a la parte superior
Desplazar pista a la parte inferior
Forma de onda
Forma de onda (dB)
Espectrograma
Configuración de espectrograma...
Hacer pista estéreo
Intercambiar canales estéreo
Dividir pista estéreo
Dividir pista estéreo a mono
Formato
Inicio y final de la selección





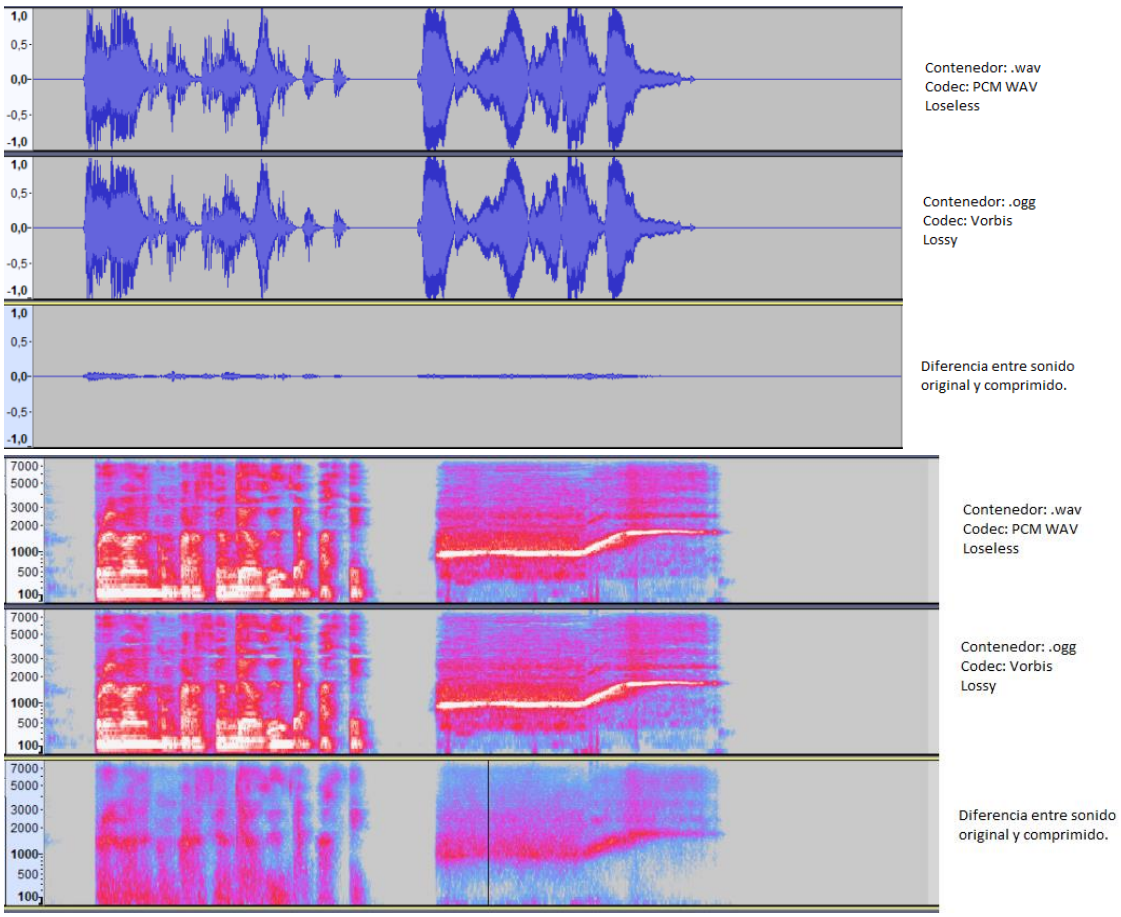
Para obtener resultados hemos elegido comparar los formatos WAV (Códec PCM WAV), MP4 (Códec AAC_LC, HE_AAC y AAC_ELD), 3GP (Códec ARM_NB, ARM_WB, AAC_LC, HE_AAC y AAC_ELD), OGG (Códec Vorbis) y MP3 (Códec LAME) aunque estos dos últimos formatos solo

permiten decodificar audios en Android y no codificar, por lo que solo se puede escuchar pero no grabar y por ende estos formatos no nos sirven para la aplicación final.

4. Resultados del estudio

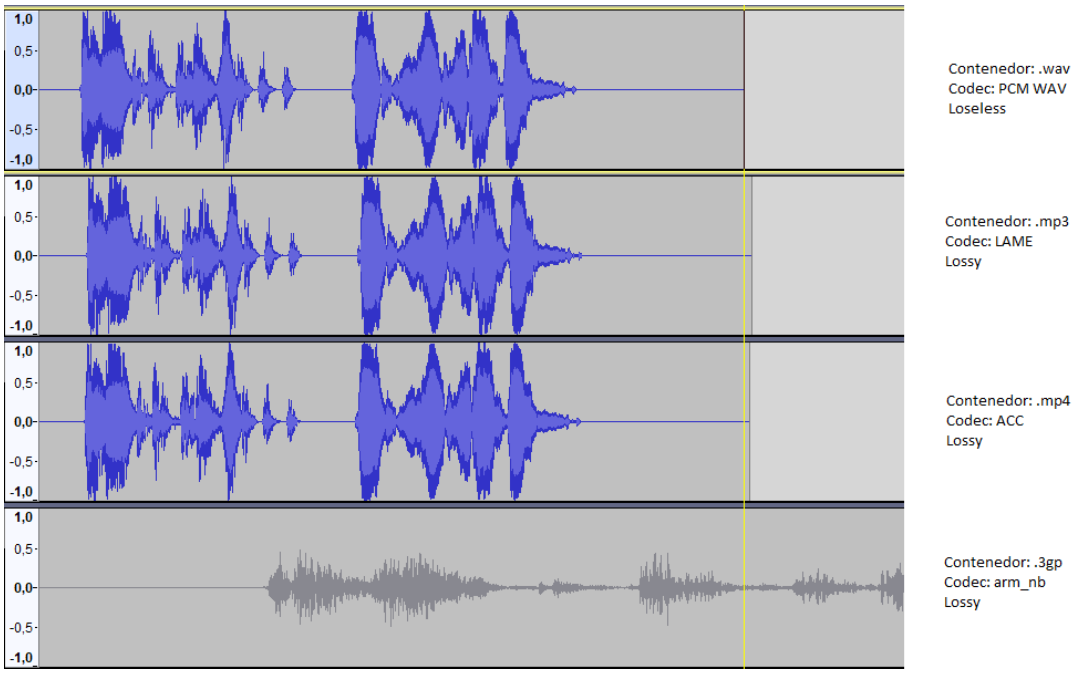
A continuación, se va a mostrar los resultados de las operaciones comentados con anterioridad para obtener las pérdidas de un formato sobre una pista mono original (WAC).

Vamos a empezar el formato OGG, aunque sabemos que tiene pérdida queremos saber que pérdida tiene, y si para el oído humano es representativo.



Como se puede observar, al realizar el proceso de inversión y suma con el original vemos que tiene una pequeña pérdida. Al escuchar esta pérdida solo se obtienen algunos ruidos imposibles de identificar para el oído humano, pero como hemos explicado al principio del estudio, no sabemos si para el clasificador esta pérdida es importante o no.

Ahora vamos a mostrar los resultados de los formatos que sabemos que podemos usar tanto en Android como en Media Recorder, además hemos añadido MP3 para mostrar y explicar el error que ocurre al comparar estos formatos.



Como se puede observar los tres formatos (MP3, MP4 y 3GP) tienen un desplazamiento al principio del audio que hace imposible comparar con el original para obtener la pérdida que tienen (porque sabemos que tienen pérdida).

Al encontrarnos ante este impedimento tenemos que encontrar otra forma para elegir el formato entre 3GP y MP4, además de elegir el mejor códec para ese formato. Para ello nos vamos a centrar en otros aspectos del audio que son la calidad y el tamaño que ocupa.

El contenedor o formato 3GP está orientado a dispositivos móviles por lo que el tamaño de estos archivos suele ser reducido y de una menor calidad. En contraposición está MP4 que es un contenedor más general y compatible con más dispositivos lo que hace que pueda tener una mayor calidad que 3GP que es lo que nos interesa en este estudio, más que el tamaño que puede tener este audio.

En cuanto al códec, al haber elegido el formato contenedor MP4 en Android podemos usar los códecs AAC (que realmente es AAC_LC), HE_AAC y AAC_ELD.

HE_ACC está diseñado para transmitir audio con baja calidad de forma rápida utilizando un bitrate bajo. El algoritmo que utiliza el códec es la replicación de la banda espectral, mediante la cual solo se codifican las frecuencias bajas y medias, y mediante los armónicos de estas frecuencias se calculan las frecuencias altas. Este algoritmo se “inventa” las frecuencias altas, por lo cual puede afectar a la información que llega al clasificador, y por ello descartamos este códec.

ACC_ELD es un códec “Low Delay”, es decir, para ofrecer una velocidad de codificación muy alta, pensada para aplicaciones críticas. Este necesita una configuración especial, que en media Recorder para Android no ha dado problemas, acelerándose la velocidad de audio y sonando de forma ininteligible.

Al no tener un sistema donde la velocidad de codificación sea un factor esencial, y al poder usar el códec AAC (AAC-LC) de forma fácil y con buena calidad (modificando el bitrate a

128kbps y el sampling rate a 48kHz) usaremos este último como la elección para la grabación de los audios de este proyecto.

5. Enlaces de interés para la documentación final

http://poseidon2.feld.cvut.cz/conf/poster/proceedings/Poster_2017/Section_EI/EI_042_Smutny.pdf