Universidad Rey Juan Carlos

Escuela Técnica Superior de Ingeniería de Telecomunicación

CODIFICACIÓN DE AUDIO

ESTÁNDARES DE COMUNICACIÓN DE AUDIO Y VÍDEO

Elena Mª del Río Galera & Yolanda Lillo Mata

Grupo: 10

Curso 2019-2020

3er curso

GRADO DE INGENIERÍA EN AUDIOVISUALES Y MULTIMEDIA

EJERCICIO 1

En primer lugar, cada participante del grupo hemos escuchado el archivo *speech.wav* y le hemos dado una puntuación de calidad más conocida como *ACR (Absolute Category Rating)* que como bien hemos visto en el tema, es una forma de evaluar directamente calidad de audio a través de la opinión del usuario. Debemos tener en cuenta que para calificar el audio vamos a utilizar valores comprendidos entre 1 “Malo” y 5 “Excelente”.

Según el enunciado de la práctica tras la escucha del audio hemos procedido a **calificar individualmente la calidad** de este y después hemos compartido los resultados para poder **calcular el promedio**,es decir, nuestro *Mean Opinion Score (MOS).*

Resumimos los datos recopilados en la tabla inferior:

Participantes	Puntuaciones
Yolanda	4.5
Elena	4.0
Media	4.25

A continuación, realizamos lo mismo pero escuchando el archivo *speech_LQ.wav*, y le damos una puntuación pero esta vez en comparación con el audio que hemos escuchado anteriormente (*speech.wav*), esto es lo que llamamos una puntuación del tipo *DCR (Degradation Category Rating),* que es un tipo de evaluación comparativa. Tras calificar individualmente este archivo, hemos calculado el *DMOS (Degradation MOS),* que es el promedio de los valores DCR.

Los datos resultantes se presentan en la tabla inferior.

Participantes	Puntuaciones
Yolanda	1.5
Elena	2.0
Media	1.75

Ahora observamos como el valor de la media disminuye considerablemente respecto al valor de MOS.

Nosotras no hemos tenido discrepancias graves a la hora de dar puntuaciones, pero en el caso de que se hubieran dado tales discrepancias, hubiera sido natural ya que como sabemos de teoría esto es un método puramente cualitativo y subjetivo. Una persona con pérdida de audición, o escuchando el audio en un entorno con ruido, podría dar puntuaciones más pobres, aunque el audio realmente fuera bueno.

En nuestro caso, las condiciones de escucha, han sido buenas y similares,con auriculares y en un entorno tranquilo. Por eso, no existen graves discrepancias entre las puntuaciones que hemos dado a los audios.

A la hora de realizar un estudio a gran escala sobre la calidad de un audio determinado, los resultados obtenidos pudieran no ser del todo fiables.

Este es un método puramente subjetivo, porque no todos tenemos el mismo nivel de audición, por ejemplo, si en el estudio participa, un chico de 13 años, en buen estado de salud, y con un nivel de audición óptimo, y también participa una persona de mediana edad que tiene pérdidas de audición del 20%, las notas que le darían serían muy diferentes por su situación individual y no por la calidad del audio.

EJERCICIO 2

En este apartado se nos pide calcular **calcular el DMOS de forma objetiva**, y esto lo calculamos con el **parámetro PESQ, a través de MATLAB.**

El **PESQ** o evaluación de calidad de voz por percepción es un parámetro para calcular la calidad de un determinado segmento de audio que se basa en dos modelos:

- El **perceptual** que simula la respuesta del oído.
- El **cognitivo** que simula la respuesta del cerebro del oyente.

En otras palabras, este modelo se basa en simular lo que percibimos físicamente a través de nuestro sentido del oído y en lo que interpreta nuestro cerebro.

Por ejemplo, cuando un grupo de personas escucha el mismo archivo de audio pero sus cerebros lo interpretan de diferentes formas, ya sea por factores de edad, por factores físicos, por problemas de audición...

Se nos pide la estimación del DMOS a través de MATLAB entre *speech.wav* y *speech_LQ.wav* y la **comparación con los datos** que obtuvimos de forma cualitativa y subjetiva en el **ejercicio 1**. Lo único que tuvimos que hacer fue cambiar en MATLAB los valores de *ref_audio* y *ev_audio*, tomando como audio de referencia el audio sin ruido(*speech.wav)*, puesto que es el que tiene mayor calidad.

Los resultados de nuestra escucha, y la puntuación individual y subjetiva tanto como la media, se pueden consultar en esta tabla.

En ella observamos que nuestro DMOS es 1,75, lo que se traduce en que subjetivamente hablando la calidad del audio es mala.

En este ejercicio, aplicamos el parámetro objetivo PESQ, para comparar con los resultados de nuestro estudio. El resultado de MATLAB lo podemos ver en la imagen inferior. Siendo el DMOS 1.083.

```
>> test_pesq

Perceptual Evaluation of Speech Quality (PESQ)

done.
done.
 Level normalization...
 IRS filtering...
 Variable delay compensation...
 Acoustic model processing...

P.862.2 Prediction (MOS-LQO):  = 1.083
```

El valor del DMOS tan bajo, como en nuestro caso, significa que la diferencia entre ambos sonidos es muy significativa.

La media de las puntuaciones que nosotras hemos dado, y el valor de DMOS que nos da MATLAB, es similar, esto se debe a que tanto objetiva, como subjetivamente, la diferencia entre los audios es muy notoria y hay degradación.

En cuanto a la comparación del archivo de audio *speech.wav* consigo mismo para calcular el PESQ, vemos que su resultado es bastante alto (DMOS=4.644), y es mayor que la media de nuestras puntuaciones subjetivas.

No hay diferencias significativas entre los dos audios (spoiler, es el mismo audio dos veces). A continuación vemos el resultado de la comparación entre *speech.wav* y él mismo en MATLAB:

```
>> test_pesq

Perceptual Evaluation of Speech Quality (PESQ)

done.
done.
 Level normalization...
 IRS filtering...
 Variable delay compensation...
 Acoustic model processing...

P.862.2 Prediction (MOS-LQO):  = 4.644
>>
```

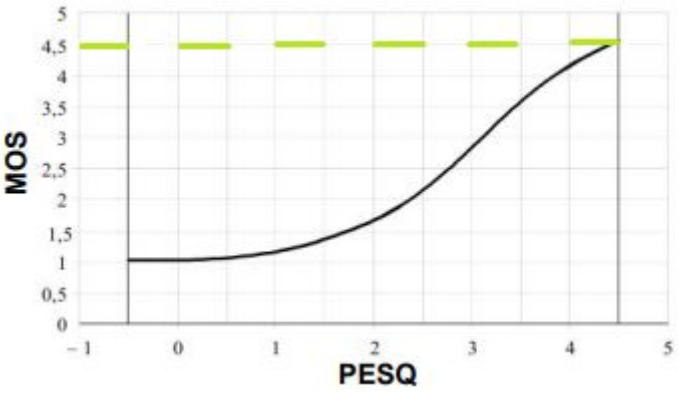
En el código de MATLAB utilizamos *speech.wav* como señal de referencia y señal de evaluación porque utilizamos el algoritmo PESQ, que a diferencia de otros

modelos como por ejemplo 3SQM, necesita una señal de referencia con la que comparar.

Tras evaluar los resultados obtenidos, y pese a que al comparar el audio consigo mismo, esperábamos como DMOS un valor de 5, vemos que el algoritmo PESQ nos da un resultado de 4.644. **¿Por qué ocurre esto? ¿Es incorrecto el resultado que nos da MATLAB?**

No, el resultado que obtenemos del algoritmo de MATLAB no es incorrecto, como nosotras pensamos en un primer momento, y tiene una explicación teórica.

Si repasamos la teoría del PESQ, vemos que los valores van desde -0.5 (degradación) hasta 4.5 (sin degradación), mientras que (como hemos visto arriba), en la escala de MOS los valores van de 1 (mucha degradación) a 5 (sin degradación). PESQ y MOS por lo tanto tienen escalas similares que se relacionan por la función que vemos a la derecha.



Por esta relación, el resultado que nos da MATLAB a través del PESQ, del DMOS, es 4.644, y no 5, como se podría esperar sin tener en cuenta la información detallada y reflejada en la imagen.

EJERCICIO 3

En este ejercicio debemos comprobar las **características técnicas de un archivo de audio** que hemos grabado nosotros desde el ordenador, en el cual decimos una frase del tema.

```
Duration: 00:00:08.78, start: 0.000000, bitrate: 261 kb/s
  Stream #0:0: Audio: mp3, 44100 Hz, stereo, fltp, 259 kb/s
```

Utilizamos el comando *ffprobe -i recorded_audio.mp3* ya que nuestro archivo tiene una extensión MP3 y obtenemos lo siguiente:

A partir de esta imagen podemos identificar los parámetros más importantes, como la **duración del stream** de **08.78 segundos**, el **bitrate o tasa de bits** que tiene un valor de **261kb/s**, y el valor de **sampling rate o frecuencia de muestreo** que es de **44100 Hz.**

Para identificar estos parámetros no hemos tenido mayor problema, sin embargo para conocer el **número de canales** que en este caso es **1** (*stream #0:0:*)

hemos querido hacer una comprobación práctica, probando el mismo comando (*ffprobe -i recorded_audio.mp3)* pero con un archivo de video con extensión *.mp4*.

Podemos observar, como en el caso de un vídeo, tenemos un canal para audio (*stream #0:0:*) y otro para vídeo (*stream #0:1:*). Lo que confirma nuestra hipótesis de que *recorded_audio.mp3* tienen un solo canal.

```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'always.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 1
    compatible_brands: mp41mp42isom
    creation_time   : 2020-04-15T15:53:09.000000Z
  Duration: 00:00:38.96, start: 0.000000, bitrate: 5786 kb/s
    Stream #0:0(spa): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp,
 125 kb/s (default)
    Metadata:
      creation_time   : 2020-04-15T15:53:09.000000Z
      handler_name    : Core Media Audio
    Stream #0:1(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p(tv, bt709, p
rogressive), 1280x720 [SAR 1:1 DAR 16:9], 5643 kb/s, 23.98 fps, 23.98 tbr, 24k tbn
, 48k tbc (default)
    Metadata:
      creation_time   : 2020-04-15T15:53:09.000000Z
      handler_name    : Core Media Video
```

Para identificar el codec utilizado en nuestro archivo original, podemos observar que en la primera imagen del ejercicio 3, pone *Audio:mp3*, por tanto, su **codec utilizado** es **MP3**. Para estar seguras de nuestra conclusión hemos comprobado los parámetros de *recorded_audio.wav*, y vemos como aquí el codec utilizado es PCM (*Audio:pcm_s24le*).

```
Input #0, wav, from 'recorded_audio.wav':
  Duration: 00:00:08.68, bitrate: 2307 kb/s
    Stream #0:0: Audio: pcm_s24le ([1][0][0][0] / 0x0001), 48000 Hz,
stereo, s32 (24 bit), 2304 kb/s
```

De forma análoga al ejercicio 1 hemos **calculado el MOS** y hemos recogido los resultado en la siguiente tabla.

Participantes	Puntuaciones
Yolanda	4.0
Elena	4.5
Media	4.25

La **metodología de grabación** utilizada ha sido a través de la grabadora del ordenador, con unos auriculares con micrófono y en un entorno sin ruido.

Los parámetros obtenidos con el comando *ffprobe*, los hemos diseccionado más arriba, simplemente comentar, lo atípico de que una grabación rudimentaria como es el caso tenga una frecuencia de muestreo tan grande (44100 Hz) cuando lo normal sería en torno a 16000 Hz, esto nos podrá ocasionar problemas en ejercicios posteriores. En cuanto al bitrate observamos un valor que no es lo habitual en archivos con extensión MP3, ya que es demasiado grande.

EJERCICIO 4

Vamos a estudiar la influencia del bitrate y el sampling rate en el tamaño y calidad del stream de audio.

En primer lugar, tenemos que convertir nuestro archivo original en un audio MP3 con bitrate 320k y sampling rate 16000Hz, para ello utilizamos el comando que se enuncia en la práctica que es el siguiente:
ffmpeg -i recorded_audio.mp3 -codec:a libmp3lame -b:a 320k -ar 16000 nuevo320.mp3.

Recordemos que nuestro audio original también estaba en formato MP3, aunque podríamos utilizar el formato que quisiéramos, y que tenía un bitrate de 261 kb/s y un sampling rate de 44100 Hz. El archivo *nuevo.mp3* es el que utilizaremos para todas las comparaciones a la hora de estudiar la calidad de nuestro demás sonidos.

Ahora que ya tenemos el archivo *nuevo320.mp3*, que es el que vamos a modificar en adelante, consultamos su información, en comparación con *recorded_audio.mp3*, y vemos que si bien nuestra frecuencia de muestreo si se ve modificada a 16000 Hz, la tasa de bits, también se ve modificada, pero no al valor que le hemos especificado, si no que está a un valor de 128Kbps, más abajo, profundizaremos en el por qué de esto.

Vamos a estudiar la **influencia del bitrate**, variando el mismo desde 320k hasta 64k, nosotras le hemos dado los valores de **64k, 128k y 256k** y dejamos el sampling rate fijo en valor 16000 Hz.
Hemos resumido los datos recopilados para compararlos con mayor facilidad en la siguiente tabla:

Bitrate entrada (Kbps)	Bitrate salida (Kbps)	Tamaño (Kb)	Calidad	Sampling rate (Hz)
320	160	178	4.644	16000
256	160	178	4.644	16000
128	128	142	4.629	16000
64	64	71	4.502	16000
Señal original	261	279	-	44100

Vamos a discutir los resultados de la **influencia del bitrate** expuestos en la tabla superior.

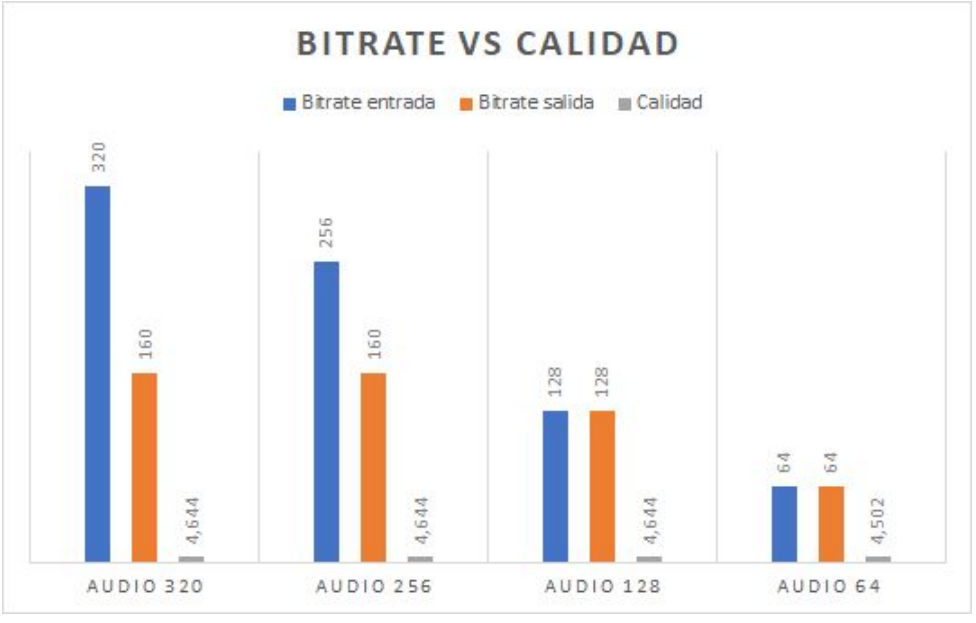
Primero observamos que algunos de los datos del bitrate de salida no coinciden con los de entrada. Observamos que esto ocurre cuando el bitrate de entrada es mayor que 128, y que el bitrate de salida nunca es superior a 160.

Siendo la frecuencia de muestreo, la cantidad de muestras de audio que tomas en un segundo, y el bitrate, la tasa de bits o datos que son procesados por unidad de tiempo. Si tengo 16000 muestras por segundo, existe un máximo de bitrate para esa frecuencia de muestreo, y ese máximo es 160.

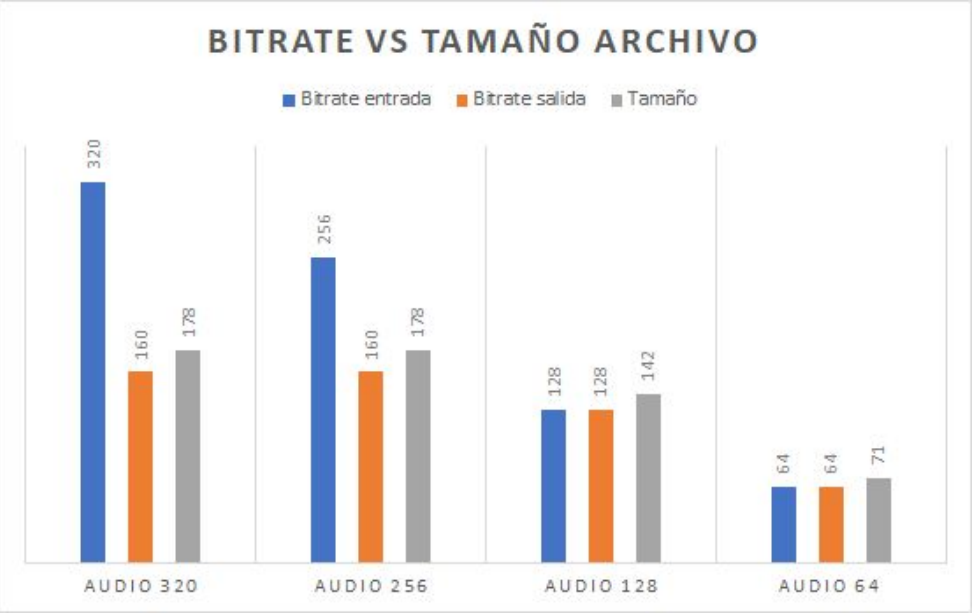
Para llegar a esta conclusión hemos comprobado nuestra hipótesis generando a partir de nuestro audio original, que tiene como frecuencia de muestreo 44100 Hz, y por lo tanto si podrá tener una tasa de bits mayor que 320,(hasta 441).

En cuanto a la **calidad** y el **tamaño**, según lo que acabamos de explicar, y dada la relación bitrate-sampling rate, tiene todo el sentido, que con un bitrate de entrada superior a 160, la calidad sea máxima y no varíe, y el tamaño, sea igualmente el máximo y tampoco varíe. (A mayor calidad, mayor tamaño).

En las siguiente gráfica, vemos claramente lo que hemos comentado arriba, que a mayor **bitrate**, mayor **calidad**, siendo bitrate máximo en este caso 160:



Y en la siguiente gráfica vemos una relación directamente proporcional del **bitrate** y el **tamaño de archivo**:



Ahora, vamos a estudiar la **influencia del sampling rate**, dando los valores de 16000Hz, 11025Hz y 8000Hz y dejamos el bitrate en un valor fijo 128k.
Hemos resumido los datos recopilados en la siguiente tabla:

Bitrate entrada (Kbps)	Bitrate salida (Kbps)	Tamaño (Kb)	Calidad	Sampling rate (Hz)
128	128	142	4.629	16000
128	64	72	3.917	11025
128	64	73	2.718	8000
Señal original	261	279	-	44100

Como hemos visto en el apartado anterior existe una relación entre bitrate y sampling rate, si disminuye el sampling rate, como es el caso de este ejercicio, se ve afectado el bitrate. Aunque queramos forzar un bitrate mayor, al disminuir el número de muestras de estudio, el bitrate se adapta al máximo disponible según las muestras que tenemos, y que además se ve afectado por tratarse del encoder MP3

y estar en estéreo, todo esto se puede comprobar en la tabla inferior, y ahí podemos ver como nuestro bitrate, está entre los valores normales para cada frecuencia.

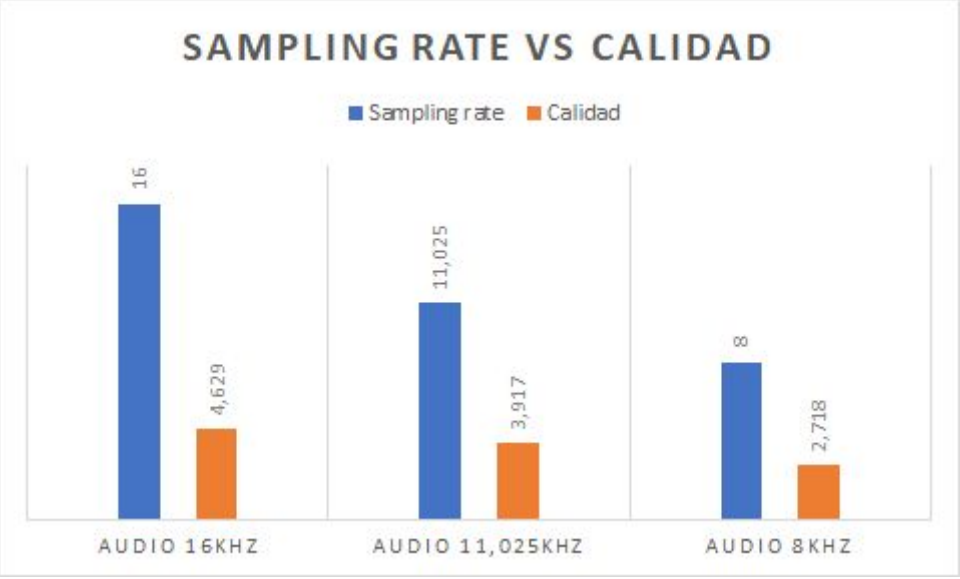
MP3 tiene soporte para numerosas frecuencias que oscilan entre 8000 Hz y 44100Hz.

Por tanto concluimos, que como el número de muestras disminuye nuestra calidad también es menor y de forma análoga el tamaño también va disminuyendo

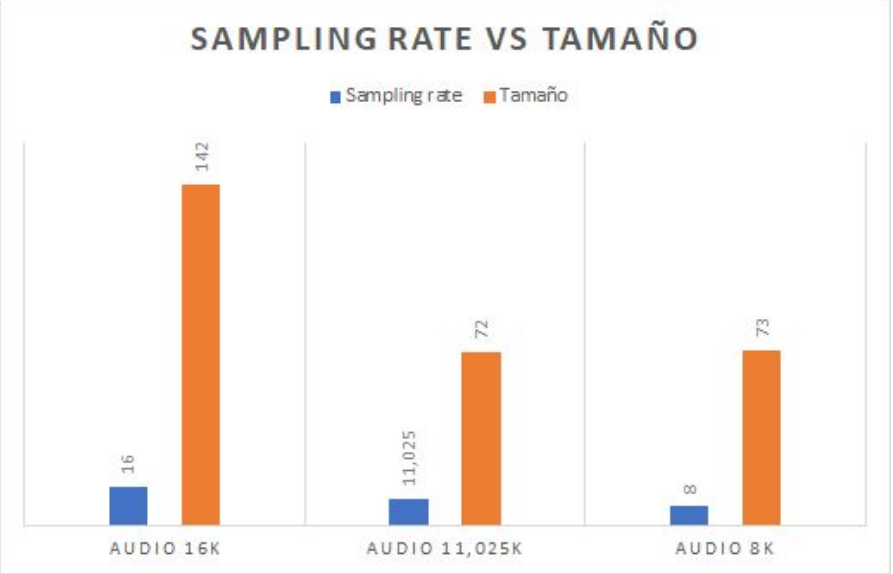
Frecuencia (Hz)	Mínimo bitrate estéreo	Máximo bitrate estéreo
16000	48	160
11025	48	160
8000	32	160

En las siguiente gráfica, lo que venimos comentando, que a menor **sampling rate**, menor **calidad**, siendo el sampling rate mínimo en MP3 8000 Hz.

En nuestra gráfica, hemos puesto la frecuencia de muestreo en kHz para que se vea mejor la diferencia.



En esta última gráfica vemos una relación del **sampling rate** y el **tamaño**, cuando disminuye uno, el otro también lo hace.



EJERCICIO 5

En este apartado se trata de probar diferentes encoders de audio y contrastar sus diferencias, lo hacemos partiendo de *recorded_audio.mp3*.

Hemos utilizado, el **encoder libopus**, cuyo archivo resultante tiene una extensión **.ogg**, el **encoder aac**, con una extensión de archivo generado **.m4a**, y el **encoder flac**, con extensión de archivo **.flac**.

Cuando ya tenemos nuestros nuevos audios, con diferentes códecs, vamos comparando la calidad de cada uno de ellos con el audio original (*recorded_audio.mp3*) y observamos que MATLAB no nos lo permite.

Esto es debido a las limitaciones del PESQ, que no procesa frecuencias de muestreo tan grandes como la de nuestra grabación (44100Hz), el máximo que puede tratar el algoritmo en MATLAB es 16000 Hz. Por eso a efectos de calidad, vamos a comparar con el archivo *nuevo320.mp3,* que si que tiene sampling rate 16000Hz.

En la tabla de la siguiente página vemos las variaciones de calidad y tamaño según el encoder de audio utilizado.

Encoder	Calidad	Tamaño (Kb)	Bitrate (Kbps)	Sampling Rate (Hz)
OPUS	4.492	138	328	48000
AAC	4.430	86	78	16000
FLAC	4.464	249	228	16000
MP3 ORIGINAL	-	287	261	44100

Con la información que se muestra en la tabla podemos concluir que elegir un encoder u otro, no es indiferente a efectos de calidad, o tamaño.

Respecto a los encoders que nosotras hemos probado, podemos decir con seguridad, que no todos son igual de eficientes. Si bien, respecto a la calidad de audio, no observamos grandes diferencias de utilizar un encoder u otro, si vemos esas diferencias en el tamaño.

Por la poca variación de la calidad, y la gran diferencia de tamaño de archivo, nos parece que el encoder más eficiente de los aquí probados, sería el ACC. Es cierto que respecto a los otros es el que tiene peor calidad, pero en relación a la calidad-tamaño, es el mejor. El archivo .flac o el .ogg, ocupan más del doble de Kb que el archivo .m4a, y su calidad, no es el doble de buena.

Aunque en el ejercicio se pide que todos los nuevos archivos con encoders diferentes tengan mismo sampling rate y mismo bitrate, por mucho que en la conversión se lo hemos especificado,

ffmpeg -i recorded_audio.mp3 -codec:a libopus -b:a 128k -ar 16000 salida.ogg

ffmpeg -i recorded_audio.mp3 -codec:a acc -b:a 128k -ar 16000 salida.m4a

ffmpeg -i recorded_audio.mp3 -codec:a flac -b:a 128k -ar 16000 salida.flac.

vemos que la realidad es diferente, y el bitrate y el sampling rate sufren modificaciones, por las limitaciones propias de cada encoder.

En el ejercicio 4 hemos visto como no podemos forzar determinados valores de frecuencia ni de tasa de bits para MP3, y que **los archivos de audio dependen de varios factores que necesitan estar en consonancia** unos con otros, porque de otra manera no tendría sentido.