

MODELOS Y CLASIFICACION

Técnicas para el Análisis del Audio Práctica

1. Analizar el audio AnalisisTextos.mp3 con MedialInfo

Informar: (AnalisisTextos.txt)

- Formato: MPEG Audio (MP3)
- Tasa de bits: 256 kb/s (constante - CBR)
- Canales: 1 canal (mono)
- Frecuencia de muestreo: 48.0 kHz

2. Realizar el sampleo con ffmpeg

```
ffmpeg -i C:\ffmpeg\AnalisisTextos.mp3 -ar 16000 -ac 1  
C:\ffmpeg\AnalisisTextos_rsampld.wav
```

3. Analizar el audio nuevamente con MedialInfo

Informar: (AnalisisTextos_rsampld.txt)

- Formato: WAV (PCM)
- tasa de bits: 256 kb/s
- canales: 1 canal (mono)
- formato de muestreo: 16.0 kHz

4. Con Python → en el script

- ✓ Mostrar el Vector de la señal segmentada
- ✓ Mostrar la cantidad de elementos de la muestra
- ✓ Mostrar la Frecuencia de Muestreo
- ✓ Mostrar la duración en segundos del audio

5. Imprimir la señal sonora → en el script

6. Reproducir la señal original → en el script

7. Modificar la frecuencia de muestreo para que dure más y menos tiempo. Explicar que sucede con el sonido

```
Audio(audio, rate=sr*2)
```

- ✓ Se reproduce en la mitad del tiempo.
- ✓ Suena más agudo y más rápido

```
Audio(audio, rate=sr*0.3)
```

- ✓ El audio dura más tiempo, se interpreta más lentamente.
- ✓ Suena más grave, más profundo.

8. Bajar la calidad del audio y reproducir la señal. Explicar cuál es el proceso.

Para bajar la calidad, reduje la profundidad de bits de 16 a 8, lo que significa menos niveles para representar el sonido. Esto hace que el audio pierda detalle y suene más distorsionado o con ruido.

Lo logré cambiando la señal a números enteros entre -128 y 127, lo que degrada la calidad pero mantiene el sonido reconocible.

```
# Escalar la amplitud al rango [-128, 127] y convertir a enteros de 8 bits
```

```
# audio_8bit = (audio * 2**7).astype(np.int8)
```

```
# Reproducir el audio con menor resolución
```

```
# print("Audio con profundidad de 8 bits:")
```

```
# Audio(audio_8bit, rate=sr)
```