

# Universidad Internacional del Ecuador

## Interfaces y Multimedia

### Actividad 1.3 – Gestión del Trabajo Autónomo

**Estudiante:** Fabian Campoverde

**Docente:** Mgs. Richard Armijos

**Proyecto:** EcoAlerta Loja

**Fecha:** Noviembre 2025

## 1. Introducción

En el desarrollo de sistemas web orientados a la atención ciudadana, como EcoAlerta Loja, el audio se convierte en un elemento clave para reforzar la interacción y mejorar la accesibilidad. Un sonido bien diseñado aporta claridad, reduce la incertidumbre del usuario y genera una experiencia más intuitiva al momento de reportar incidentes ambientales. Sin embargo, trabajar con audio para la web implica retos específicos: mantener un peso adecuado, garantizar compatibilidad entre navegadores y asegurar niveles de volumen consistentes en distintos dispositivos.

La presente práctica consistió en grabar y procesar un audio breve de notificación, optimizarlo bajo estándares actuales de sonido digital (-16 LUFS y -1 dBFS de pico), y generar versiones en WAV, MP3 y OGG listas para incorporación en producción. Además, se documentó el uso de un tutor de IA como apoyo técnico, validando siempre cada recomendación con fuentes confiables. Este informe reúne las decisiones adoptadas, la metodología aplicada y los resultados obtenidos.

## 2. Marco Teórico

El audio digital se estructura a partir de tres conceptos fundamentales. La **frecuencia de muestreo** determina cuántas veces por segundo se captura la señal; en este proyecto se usó 44.1 kHz, estándar común en aplicaciones web y móviles. La **profundidad de bits** influye directamente en el rango dinámico; se emplearon 16 bits, suficientes para un recurso de interfaz sin incrementar el peso innecesariamente.

En cuanto al volumen, la diferencia entre **dBFS** y **LUFS** es determinante. Los picos en dBFS indican el máximo nivel alcanzado, mientras que LUFS evalúa la sonoridad percibida, algo indispensable para mantener consistencia entre plataformas. Para notificaciones web se recomienda trabajar alrededor de -16 LUFS, evitando la necesidad de que el usuario ajuste el volumen manualmente.

Respecto a los formatos, el archivo **WAV** se utiliza como máster sin pérdidas; **MP3** es ideal para compatibilidad general, aunque con compresión perceptual, y **OGG Vorbis** ofrece mejor eficiencia a bitrates similares, siendo ampliamente adoptado en entornos HTML5. Estas diferencias permiten seleccionar el formato adecuado según el contexto de uso.

### 3. Metodología de Producción

La grabación se realizó con el micrófono interno de un dispositivo Android, capturando la locución en una habitación cerrada para reducir eco y ruido ambiental. Posteriormente, el archivo se procesó utilizando Audacity, siguiendo una secuencia de edición centrada en limpieza y control dinámico.

Primero se aplicó un recorte fino para eliminar silencios innecesarios, seguido de una reducción de ruido moderada basada en una muestra de ambiente. Luego se usó un compresor suave para estabilizar el rango dinámico, definiendo un ataque de 10 ms y un release de 120 ms. Para la normalización se ajustó el pico a -1 dBFS, mientras que el loudness se aproximó a -16 LUFS mediante un limitador ligero.

La exportación se hizo en tres formatos: WAV 44.1 kHz/16-bit como archivo maestro; MP3 a 128 kbps CBR para máxima compatibilidad; y OGG Vorbis en calidad aproximada Q4 para priorizar eficiencia. El tutor de IA me ayudó a resolver dudas puntuales sobre parámetros de compresión y normalización, pero cada recomendación fue verificada con documentación técnica antes de aplicarla.

### 4. Desarrollo

Durante la edición se generaron capturas del antes y después, observándose una onda más uniforme y sin picos abruptos tras el procesamiento. Para documentar los

parámetros principales, se elaboró la siguiente tabla de referencia:

Proceso	Parámetro aplicado
Reducción de ruido	Umbral moderado (-12 dB)
Compresor	Ratio 2:1, ataque 10 ms, release 120 ms
Normalización	Pico a -1 dBFS
Loudness	Aproximado a -16 LUFS

En la comparativa final, el archivo WAV resultó en un tamaño aproximado de 2.4 MB, mientras que el MP3 ocupó cerca de 290 KB y el OGG alrededor de 260 KB, sin pérdidas perceptibles en el contexto de una notificación corta.

Para la integración web, se implementó un componente HTML con múltiples fuentes, preload en modo *metadata* y una etiqueta accesible, asegurando compatibilidad entre navegadores:

```
<audio controls preload="metadata" aria-label="Notificación EcoAlerta">  
  <source src="ecoalerta.ogg" type="audio/ogg">  
  <source src="ecoalerta.mp3" type="audio/mpeg">  
</audio>
```

## 5. Resultados y Discusión

La edición permitió mejorar significativamente la claridad del audio, eliminando ruidos sutiles que afectaban la percepción del mensaje. La normalización también logró un nivel de volumen más consistente, adecuado para usuarios que consumen la aplicación desde dispositivos móviles.

La comparativa entre códecs mostró que OGG conserva mejor calidad en relación al tamaño, por lo que se recomendó como primera opción dentro del HTML. El proceso reafirmó la importancia de evaluar el peso de los recursos, ya que influye directamente en tiempos de carga y métricas de rendimiento.

El apoyo del tutor de IA resultó útil especialmente para revisar parámetros y confirmar buenas prácticas, pero siempre se complementó con documentación oficial, manteniendo un criterio propio en cada decisión.

## 6. Conclusiones

1. La normalización por LUFS fue esencial para garantizar un sonido uniforme, evitando diferencias abruptas entre dispositivos.
2. El formato OGG demostró ser el más eficiente para la web en términos de peso y calidad, superando a MP3 en escenarios HTML5.
3. El flujo de edición permitió obtener un archivo claro y estable, adecuado para notificaciones dentro de EcoAlerta Loja.
4. El tutor de IA aceleró el proceso de toma de decisiones, pero siempre requirió verificación técnica independiente.
5. La optimización del audio contribuye directamente a mejorar la experiencia del usuario y la accesibilidad de la plataforma.

## 7. Referencias

EBU. (2020). *EBU R 128 – Loudness Normalisation and Permitted Maximum Level of Audio Signals*.

MDN Web Docs. (2024). *HTML Audio Element*.

W3C. (2021). *Web Audio API Recommendation*.