



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Evaluación del uso de diferentes códecs de audio y video  
con el fin de mejorar la eficiencia en retransmisiones en  
vivo de la empresa PAXUA PRODUCTORA  
AUDIOVISUAL, en Ibiza

Trabajo Fin de Grado

Grado en Ingeniería de Sistemas de Telecomunicación, Sonido e  
Imagen

AUTOR/A: Alonso Maciel, Daniel

Tutor/a: Boronat Segui, Fernando

Cotutor/a externo: Alcántara Manzanares, Pablo

CURSO ACADÉMICO: 2023/2024

## Resumen

Este Trabajo de Fin de Grado tiene como objetivo optimizar los códecs y configuraciones de transmisión en vivo utilizados por la empresa PAXUA Films S.C. para lograr la mejor calidad posible de vídeo y audio sin exceder los límites del ancho de banda disponible ni afectar negativamente al rendimiento.

A través de un enfoque metodológico práctico y analítico, se evalúan diferentes códecs de vídeo y audio, con un interés particular en H.264, VP9, AV1 para vídeo, y Opus para audio. Se establecen experimentos controlados que midan la calidad de la imagen, la fidelidad del audio, y el rendimiento de la transmisión utilizando el protocolo RTMP y el software WireCast. Las pruebas se realizan en un entorno que simula las condiciones reales de retransmisión de la empresa.

Se seleccionan herramientas analíticas avanzadas para garantizar mediciones objetivas, incluyendo PSNR y VMAF para la calidad de vídeo, y PESQ o POLQA para la calidad de audio. Se presta especial atención a la correlación entre la configuración del códec y el rendimiento en varios escenarios de red. Los resultados proporcionan una base de datos empírica para formular recomendaciones específicas sobre la selección del códec y ajustes de configuración para la empresa.

**Palabras clave:** Retransmisión en directo; Códecs de vídeo y audio; Protocolo RTMP; Optimización de ancho de banda; Calidad de transmisión.

## **Abstract**

The following thesis aims to optimize the codecs and live broadcast settings used by PAXUA Films S.C. to achieve the best possible video and audio quality without exceeding the available bandwidth limits or negatively impacting performance.

Through a practical and analytical methodological approach, various video and audio codecs are evaluated, with a particular focus on H.264, VP9, AV1 for video, and Opus for audio. Controlled experiments are established to measure image quality, audio fidelity, and broadcast performance using the RTMP protocol and WireCast software. Tests are conducted in an environment that simulates the real broadcasting conditions of the company.

Advanced analytical tools are selected to ensure objective measurements, including PSNR and VMAF for video quality, and PESQ or POLQA for audio quality. Special attention is paid to the correlation between codec configuration and performance across various network scenarios. The results provide an empirical database to formulate specific recommendations on codec selection and configuration settings for the company.

**Keywords:** Live broadcasting; Video and audio codecs; RTMP protocol; Bandwidth optimization; Broadcast quality.

# Índice

<b>Resumen</b> .....	1
Palabras clave: .....	1
<b>Abstract</b> .....	2
Keywords: .....	2
<b>1. Introducción</b> .....	6
1.1. Contexto y presentación del proyecto .....	6
1.2. Objetivos .....	6
1.3. Metodología.....	7
1.4. Desarrollo del proyecto.....	7
1.5. Problemas encontrados.....	8
<b>2. Marco teórico</b> .....	10
2.1. Fundamentos de la transmisión de vídeo y audio en directo .....	10
2.1.1. Historia y evolución del <i>streaming</i> de vídeo y audio.....	10
2.1.2. Principios básicos del <i>streaming</i> en directo .....	11
2.1.2.1. Compresión de datos .....	11
2.1.2.2. Transmisión y buffering .....	11
2.1.2.3. Protocolos de transmisión .....	12
2.1.2.4. Streaming adaptativo.....	12
2.2. Tecnologías utilizadas en el proyecto.....	13
2.2.1. Descripción del protocolo RTMP.....	13
2.2.2. Software de transmisión .....	13
2.2.2.1. Open Broadcast Software.....	13
2.2.2.2. WireCast.....	14
2.2.3. Códecs de vídeo y audio.....	14
2.2.3.1. FFmpeg .....	14
2.2.3.2. Detalles técnicos y aplicaciones de H.264, VP9 y AV1 .....	14
2.2.3.3. Detalles técnicos y aplicaciones de Opus.....	15
2.2.3.5. Comparación de códecs en términos de eficiencia y calidad .....	16
2.2.3.5.1. Comparativa entre H.264, VP9 y AV1 .....	16
2.2.3.5.2. Comparativa entre Opus y AAC .....	17
2.2.4. Métricas de calidad para vídeo y audio .....	18
2.2.4.1. Descripción de PSNR y VMAF .....	18
2.2.4.2. Descripción de PESQ y POLQA.....	19

<b>3. Procedimiento</b> .....	19
3.1. Software y herramientas utilizadas.....	19
3.1.1. Justificación de la selección de herramientas.....	19
3.1.2. Configuración y personalización de las herramientas .....	20
3.1.2.1. Hacer stream a YouTube con OBS .....	20
3.1.2.2. Transcodificar archivos con FFmpeg .....	22
3.1.3. Proceso de obtención de métricas .....	24
3.2. Detalle de los métodos de codificación y las configuraciones de códecs .....	25
3.2.1. Configuraciones de códecs y parámetros técnicos .....	25
3.2.2. Justificación de la selección de parámetros.....	26
3.3. Escenarios de prueba para la transmisión .....	27
3.3.1. Descripción de los escenarios de prueba.....	27
3.3.2. Relevancia de cada escenario en la evaluación de códecs .....	28
3.4. Método de recolección y análisis de datos.....	29
3.4.1. Técnicas estadísticas utilizadas para el análisis de datos.....	29
3.4.2. Herramientas de software utilizadas para el análisis de datos .....	30
<b>4. Resultados</b> .....	31
4.1. Presentación de los datos obtenidos en las pruebas .....	31
4.1.1. Visualización de los datos obtenidos .....	31
4.1.2. Contexto e interpretación de los datos obtenidos .....	35
4.1.2.1. Relevancia de los resultados en escenarios de uso real.....	37
4.1.2.2. Implicaciones de los datos para futuras configuraciones .....	38
4.2. Comparativas gráficas .....	39
4.2.1. Análisis de tendencias en los datos .....	39
4.2.2. Ventajas y desventajas de cada códec.....	40
4.2.2.1. Resumen de los mejores resultados basado en los datos obtenidos	40
4.2.2.2. Recomendaciones para la selección de códecs y configuraciones según el tipo de contenido .....	41
4.2.2.3. Configuraciones que evitar y su explicación.....	41
<b>5. Conclusiones y recomendaciones</b> .....	41
5.1. Resumen de los hallazgos más relevantes.....	41
5.2. Limitaciones del estudio .....	42
5.3. Recomendaciones basadas en los resultados del estudio .....	43
5.3.1. Ajustes específicos para la infraestructura existente en la empresa. ....	43
<b>6. Bibliografía</b> .....	45

<b>7. Listado de Anexos</b> .....	48
<b>ANEXO 1: Carpeta de programas y herramientas utilizadas</b> .....	48
<b>ANEXO 2: Carpeta de fragmentos de audio utilizados</b> .....	48
<b>ANEXO 3: Carpeta de fragmentos de vídeo utilizados</b> .....	48
<b>ANEXO 4: Análisis de métricas y resultados</b> .....	49
<b>ANEXO 5: Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2030</b> .....	49

## **Índice de Imágenes**

Imagen 1: Esquema de funcionamiento de VMAF [29]. .....	18
Imagen 2: Interfaz de OBS Studio y ventana de Ajustes. ....	21
Imagen 3: Ventana de Emisión en Directo de YouTube Studio. ....	21

## **Índice de Figuras**

Figura 1: Gráfica de PSNR en la prueba de streaming .....	32
Figura 2: Gráfica de VMAF en la prueba de streaming .....	32
Figura 3: Gráfica de PSNR en la prueba de transcodificación .....	33
Figura 4: Gráfica de VMAF en la prueba de Transcodificación .....	33
Figura 5: Gráfica del tamaño de los archivos en la prueba de transcodificación.....	34
Figura 6: Gráfica de PESQ en la prueba de audio .....	34
Figura 7: Gráfica del tamaño de los archivos en la prueba de audio.....	35

## 1. Introducción

### 1.1. Contexto y presentación del proyecto

En la era digital actual, la cantidad de retransmisiones en directo de contenidos multimedia ha aumentado exponencialmente debido a los avances tecnológicos y un aumento en la demanda de consumo de medios en tiempo real. Este proyecto se centra en la optimización de códecs de audio y vídeo para mejorar la eficiencia en las retransmisiones en directo, donde la calidad del contenido y la eficiencia del ancho de banda son parámetros muy importantes a tener en cuenta.

La motivación para este estudio surge de la necesidad de las empresas de producción audiovisual, como PAXUA Films S.C., de realizar retransmisiones de alta calidad sin verse afectadas por las limitaciones debidas al uso de ancho de banda. Este proyecto busca identificar configuraciones que maximicen la calidad visual y auditiva mientras minimizan el uso de recursos de red. Los códecs para evaluar son H.264, VP9, AV1 para vídeo y Opus para audio.

### 1.2. Objetivos

El objetivo general es optimizar la selección y configuración de códecs para retransmisiones en directo, con el fin de mejorar la eficiencia en el uso del ancho de banda y la calidad de transmisión para la empresa PAXUA Films S.C. Los objetivos específicos se describen a continuación:

- Evaluar el desempeño actual de los códecs de vídeo y audio utilizados en las retransmisiones en directo de PAXUA.
- Comparar la eficiencia de estos códecs en términos de calidad de transmisión y consumo de ancho de banda bajo diferentes configuraciones y condiciones de red, con otros códecs como VP9, AV1 y Opus.
- Identificar las mejores prácticas y configuraciones que equilibren de manera óptima la calidad y el uso de ancho de banda.
- Desarrollar recomendaciones basadas en datos obtenidos de forma empírica que puedan ser implementadas por los técnicos de PAXUA para mejorar las retransmisiones en directo.

### 1.3. Metodología

Los códecs que se analizan se seleccionan teniendo en cuenta su relevancia en la industria audiovisual. Se configuran diferentes tasas de bits y resoluciones para cada códec con el fin de evaluar su rendimiento en una variedad de condiciones. Cada códec se prueba en dos resoluciones (HD y Full HD) y en tres configuraciones de tasa de bits (baja, media y alta) para determinar su eficacia a la hora de optimizar el ancho de banda y la calidad de la transmisión.

En cuanto al software, inicialmente se planeó utilizar Wirecast para la transmisión y grabación de los vídeos, dado que es el software utilizado por la empresa. Sin embargo, debido a que Wirecast es un software de pago y el acceso a una licencia no estaba disponible durante el período de realización del proyecto, se opta por utilizar OBS (*Open Broadcaster Software*) como alternativa para configurar los códecs en las configuraciones especificadas y transmitir la señal. La calidad de vídeo se mide usando VMAF y PSNR, y la calidad de audio con PESQ. Se decide no utilizar POLQA para la evaluación de la calidad de audio debido a que, al igual que el software Wirecast, es un software de pago y no se dispone de una licencia para su uso.

Por último, se presentan y comparan los datos por medio de gráficas, utilizando el software Microsoft Excel, para determinar diferencias significativas en el rendimiento entre los códecs bajo diferentes configuraciones y condiciones.

### 1.4. Desarrollo del proyecto

El desarrollo de este proyecto se organiza en varias fases claramente definidas, con el objetivo de garantizar una ejecución fluida y sistemática del estudio. El proyecto tiene una duración estimada de dos meses, comenzando con una fase de revisión de literatura y terminando con la redacción y revisión del documento final. Se utilizan diversos recursos tecnológicos, así como plataformas de análisis estadístico para el tratamiento de los datos. Los recursos son revisados y actualizados regularmente para garantizar su adecuación a los requisitos del proyecto. A continuación, se definen las fases del proyecto:

- Fase 1: revisión de literatura y definición de la metodología.
- Fase 2: configuración y realización de pruebas experimentales.
- Fase 3: recopilación y análisis de datos.



- Fase 4: redacción de la memoria y preparación para la defensa.

Los retrasos en la recopilación de datos a causa de fallos técnicos o problemas de software se abordan mediante la realización de pruebas preliminares de los equipos y software. Además, se contemplan alternativas para cada paso del proceso, de manera que, si una idea no se puede ejecutar, se busca otra forma de obtener los datos necesarios u otra forma de realizar las evaluaciones necesarias.

### 1.5. Problemas encontrados

A lo largo del desarrollo del proyecto, se encontraron varios problemas técnicos que requirieron ajustes en la metodología y en las herramientas utilizadas. A continuación, se describen los principales desafíos y las soluciones implementadas:

Inicialmente, se planificó utilizar el software OBS Studio (pues Wirecast requiere de una licencia) para realizar transmisiones en directo a YouTube usando los diferentes códecs para evaluar el resultado final. Sin embargo, se encontró que OBS Studio no permite la transmisión en directo utilizando el códec de vídeo VP9 ni el códec de audio Opus. Estos códecs solo están disponibles para realizar grabaciones. Esto limitó las opciones de códecs disponibles para la evaluación en transmisión en directo a únicamente H.264 y AV1.

Durante las pruebas de transmisión en directo con OBS Studio, se observó que las transmisiones a 1080p utilizando el códec AV1 presentaban cortes y problemas de rendimiento. Este problema no se presentaba al utilizar una resolución de 720p ni el códec H.264, pero afectó significativamente a la calidad de las pruebas a mayor resolución.

Al realizar las métricas de PSNR y VMAF comparando los directos descargados de YouTube con los archivos de grabación (grabados simultáneamente con la transmisión para disponer de los archivos originales), se obtuvieron valores anormalmente bajos. Tras investigar, se determinó que la falta de sincronización exacta entre los archivos originales y los descargados de YouTube provocaba que las métricas no fueran precisas ni útiles.

Debido a las limitaciones encontradas en el método original (imposibilidad de evaluar ciertos códecs y obtención de datos no útiles para otros), se decidió cambiar la metodología. En lugar de realizar transmisiones en directo, se optó por utilizar la herramienta FFmpeg para codificar el archivo de vídeo original directamente a cada uno de los códecs y evaluar su desempeño y eficiencia. Este cambio permitió obtener métricas reales y coherentes, además de analizar la eficiencia de compresión basada en el tamaño de los archivos resultantes.

Este enfoque alternativo resultó en una recopilación de datos más precisa y útil para la evaluación de los códecs, permitiendo cumplir con los objetivos del proyecto y evaluar los 4 códecs mencionados anteriormente.

Tanto los datos obtenidos del análisis como los obtenidos mediante la transcodificación con FFmpeg serán incluidos y discutidos en este proyecto. De esta manera, se proporcionará una visión completa del proceso, destacando las dificultades encontradas y cómo fueron superadas para lograr los objetivos principales.

## 2. Marco teórico

### 2.1. Fundamentos de la transmisión de vídeo y audio en directo

#### 2.1.1. Historia y evolución del *streaming* de vídeo y audio

El concepto de *streaming* (transmisión en tiempo real) hace referencia a la distribución digital de contenido multimedia a través de la red, de manera que el usuario reproduce el contenido en cuanto empiece a recibir datos y el resto de contenido se va descargando simultáneamente [1].

Este concepto comenzó a aparecer en la década de 1990 con el desarrollo de tecnologías clave como el *Real-Time Messaging Protocol* (RTMP), un protocolo de red desarrollado por Adobe utilizado para transmitir contenidos multimedia entre un servidor y Flash Player y el *Real-Time Streaming Protocol* (RTSP), un protocolo que permite establecer y controlar sesiones con uno o varios flujos multimedia y que utiliza *Real-time Transport Protocol* (RTP) para la transmisión de datos y *RTP Control Protocol* (RTCP) para el control de calidad [2]. Estos protocolos facilitaron la transmisión de contenido en tiempo real, marcando los primeros pasos significativos hacia el *streaming* moderno.

No fue hasta la llegada de Adobe Flash y su capacidad para gestionar vídeos, que el *streaming* de vídeo realmente ganó popularidad en la web [3]. Esto abrió el camino a los servicios de *streaming* que hoy en día dominan el consumo de medios, como YouTube, lanzado en 2005, que marcó un hito importante al popularizar el *streaming* de vídeo [4] al permitir que los usuarios subieran y compartieran vídeos fácilmente.

A medida que avanzamos hacia la década de 2010, tecnologías como HTML5 comenzaron a reemplazar a Flash en la entrega de contenido multimedia, debido a su mayor seguridad y eficiencia [5]. Este cambio fue acompañado por una adopción más amplia del *streaming* adaptativo, como se evidencia en el desarrollo de MPEG-DASH, que permite una experiencia de usuario más suave y de mayor calidad.

En los primeros días, el *streaming* multimedia enfrentaba limitaciones significativas debido a las restricciones de ancho de banda y la capacidad de procesamiento de los ordenadores de la época. Sin embargo, con el incremento de velocidad que recibió la banda ancha y las mejoras en las tecnologías de

compresión, como el códec H.264, el *streaming* de vídeo se volvió más accesible y eficiente [6].

A medida que pasan los años, la industria ha visto una transición hacia códecs más avanzados como H.265/HEVC y AV1, que ofrecen una mayor compresión y, por lo tanto, reducen el costo del ancho de banda, a la vez que mantienen o incluso mejoran la calidad de vídeo. Estos avances han sido complementados por la adopción de HTML5 en los navegadores, lo que ha facilitado una experiencia de usuario más rica y accesible sin la necesidad de *plugins* adicionales.

### 2.1.2. Principios básicos del *streaming* en directo

El *streaming* en directo se fundamenta en varios principios técnicos que aseguran la entrega eficaz de contenido multimedia a través de Internet para conseguir una transmisión eficiente y estable. Estos principios que se describen a continuación son esenciales para comprender cómo se logra una transmisión fluida y de alta calidad.

#### 2.1.2.1. Compresión de datos

Uno de los aspectos más importantes del *streaming* en directo es la compresión de datos. Hay que tener en cuenta que los archivos de vídeo y audio sin comprimir ocupan demasiado espacio, por tanto, la compresión es indispensable para reducir el ancho de banda requerido durante la transmisión sin sacrificar significativamente la calidad del contenido. Los códecs como H.264, entre otros, facilitan este proceso, pues permiten una alta calidad de imagen a tasas de bits relativamente bajas al estar diseñados para reducir el tamaño de los archivos de vídeo de manera eficiente. [7]. Cada códec tiene diferentes características y niveles de eficiencia en la compresión y calidad del vídeo.

#### 2.1.2.2. Transmisión y *buffering*

La transmisión de contenidos multimedia en directo implica enviar contenido de manera continua desde un servidor a un dispositivo receptor a través de Internet. Este proceso se basa en protocolos específicos que gestionan la transferencia eficiente de datos. El *buffering* (almacenamiento en búfer) permite la reproducción fluida del contenido. Su funcionamiento consiste en precargar datos en la memoria del dispositivo antes de que se reproduzcan, de

este modo se consiguen mitigar los problemas causados por la variabilidad en la velocidad de la conexión a Internet, asegurando que el vídeo o el audio continúe reproduciéndose sin interrupciones, aunque la conexión se vuelva temporalmente más lenta o inestable [8].

Otro aspecto importante del *buffering* es permitir que los vídeos se comiencen a reproducir antes de que todo el contenido se haya cargado completamente, facilitando la continuidad de la reproducción durante las variaciones en la velocidad de conexión. Además, el *buffering* adecuado puede mejorar significativamente la experiencia del usuario, evitando las pausas frecuentes e interrupciones que resultan frustrantes.

#### 2.1.2.3. Protocolos de transmisión

Los protocolos de transmisión se requieren para la entrega eficiente de contenido multimedia en *streaming*. Estos protocolos se encargan de gestionar la transferencia de datos a través de redes y llevar a cabo tareas como la fragmentación de datos para facilitar su manejo y transmisión, la detección y corrección de errores, la sincronización del audio y el vídeo, el ajuste de la calidad de la transmisión en función de las condiciones de la red, y asegurar que la transmisión mantenga una calidad óptima. Los tres protocolos de transmisión más relevantes y utilizados actualmente son:

- RTPM (*Real-Time Messaging Protocol*), desarrollado por Adobe.
- HLS (*HTTP Live Streaming*), desarrollado por Apple.
- MPEG-DASH.

#### 2.1.2.4. *Streaming* adaptativo

Con el fin de optimizar el uso del ancho de banda disponible, reducir el *buffering* y mejorar la calidad de la reproducción se inventó el *streaming* adaptativo, en concreto el *streaming* de tasa de bits adaptable (*Adaptive Bitrate Streaming*, ABR), una técnica diseñada para mejorar la experiencia de visualización en condiciones de red variables. Esta técnica funciona mediante la evaluación continua del rendimiento de la red y la adaptación de la calidad del contenido multimedia para coincidir con el ancho de banda disponible [9]. Durante la reproducción, el reproductor selecciona la versión más adecuada de cada segmento en tiempo real, basándose en la velocidad de la conexión y la

capacidad del dispositivo, lo que asegura una reproducción continua y de la mejor calidad posible sin interrupciones, incluso en redes inestables.

## 2.2. Tecnologías utilizadas en el proyecto

### 2.2.1. Descripción del protocolo RTMP

El protocolo RTMP fue desarrollado inicialmente por Macromedia, que luego fue adquirida por Adobe, y se diseñó para transmitir vídeo, audio y otros datos a través de Internet, optimizado para baja latencia, esencial para aplicaciones de *streaming* en directo. Aunque Adobe anunció el fin de Flash en 2020, RTMP sigue siendo ampliamente utilizado en aplicaciones que requieren baja latencia [10].

La principal preocupación de la mayoría de los usuarios respecto a una solución de *streaming* es la calidad de la entrega del contenido. Si la resolución es baja, los consumidores no la aceptarán. De igual manera, una solución con alta latencia y tiempos de búfer prolongados antes de la reproducción tendrá un rendimiento deficiente y el mismo impacto negativo entre los consumidores [11]. Por esta razón se utiliza RTMP, pues desde su aparición ha demostrado disponer de una latencia baja y un *buffering* muy reducido, entregando una gran calidad siempre y cuando la red sea lo suficientemente robusta y capaz de proporcionar una velocidad suficiente.

RTMP también destaca por su capacidad para soportar *streaming* masivo simultáneo. Sin embargo, al ser un protocolo no encriptado, es muy vulnerable y cualquier transmisión RTMP puede ser recibida y escuchada sin requerir autorización. Aun así, plataformas dedicadas al vídeo y al *streaming* como YouTube pueden utilizar esta tecnología. Esto se debe a que estas plataformas utilizan una variante denominada *Real-Time Messaging Protocol Server* (RTMPS), una extensión segura del protocolo RTMP que ofrece cifrado y conexión de seguridad en la capa de transporte (TLS/SSL) [12].

### 2.2.2. Software de transmisión

#### 2.2.2.1. Open Broadcast Software

OBS Studio (*Open Broadcast Software Studio*) es un software muy popular y de código abierto diseñado para la transmisión y grabación de vídeo. Destaca por su flexibilidad y la amplia gama de funcionalidades que soporta, incluyendo

diversas opciones de personalización y compatibilidad con múltiples tipos de códecs y formatos de transmisión [13]. Este software soporta la transmisión en vivo a servicios populares como YouTube, Facebook Live y Twitch, así como la capacidad de transmitir a servidores RTMP personalizados [14].

#### 2.2.2.2. WireCast

WireCast es un software robusto utilizado para la producción de vídeos en directo. Este software dispone de mezcla de vídeo, inserción de múltiples fuentes de audio y vídeo, y soporte para transmisión a múltiples plataformas simultáneamente [15]. Es especialmente conocido por su interfaz intuitiva y su conjunto de herramientas profesionales que facilitan la producción de transmisiones de alta calidad.

#### 2.2.3. Códecs de vídeo y audio

##### 2.2.3.1. FFmpeg

FFmpeg es un proyecto de software libre que produce bibliotecas y programas para manejar datos multimedia. Se utiliza para, codificar, decodificar, transcodificar, grabar y transmitir audio y vídeo, filtrar, y reproducir prácticamente cualquier archivo multimedia [16]. Este software incluye herramientas como *ffmpeg* para conversión de archivos, *ffplay* para reproducción y *ffprobe* para análisis de medios. FFmpeg soporta una amplia variedad de formatos y códecs, lo que lo hace extremadamente versátil y popular en la industria audiovisual para tareas de edición, *streaming* y transcodificación de contenido multimedia.

##### 2.2.3.2. Detalles técnicos y aplicaciones de H.264, VP9 y AV1

H.264, también conocido como MPEG-4 parte 10 o H.264/MPEG-4 AVC (*Advanced Video Coding*) es una norma que define un códec de vídeo conocido por su eficiencia en la compresión, es uno de los códecs más ampliamente utilizados, perfecto para transmitir vídeo de alta calidad a tasas de bit reducidas. El ITU-T *Video Coding Experts Group* (VCEG) y el ISO/IEC *Moving Picture Experts Group* (MPEG) desarrollaron conjuntamente este proyecto con la intención de crear un estándar capaz de proporcionar una buena calidad de imagen a tasas binarias inferiores a los estándares previos sin complicar mucho más su diseño [17]. Este códec utiliza técnicas avanzadas como la predicción y compensación de movimiento, y la transformada de coseno discreta, lo que

resulta en una gran compresión sin pérdidas perceptibles de calidad [18]. El códec H.264 es ampliamente utilizado para aplicaciones que van desde vídeos en Internet (como YouTube o Vimeo), transmisión de televisión, videoconferencias e incluso vigilancia por vídeo. El hecho de ser compatible con casi todos los dispositivos y navegadores lo hace extremadamente versátil y popular.

VP9 fue desarrollado por Google como una alternativa abierta y libre de regalías a H.264. VP9 ofrece mejor compresión que su predecesor y permite reducir el uso del ancho de banda hasta en un 50% en comparación con H.264, sin degradar la calidad del vídeo. Este códec utiliza un modelo de predicción basado en bloques y una mejor segmentación, lo que permite una adaptación más granular a las condiciones del contenido de vídeo [19]. Se utiliza principalmente en plataformas de Google como YouTube debido a su eficiencia en situaciones de ancho de banda limitado. También es utilizado en dispositivos que soportan decodificación por hardware de VP9, mejorando la reproducción de vídeo de alta resolución.

AV1 es el códec más reciente de los tres y ofrece una compresión aún más eficiente que VP9 y H.264. Fue desarrollado por la *Alliance for Open Media* con el fin de minimizar el uso de ancho de banda y mejorar la calidad del vídeo [20], ofreciendo una mejora de hasta el 30% en la compresión [21], aunque requiere más recursos computacionales para su codificación y decodificación. AV1 introduce nuevas técnicas como la codificación por bloques adaptativos, la restauración de cuadros y el modelado de segmentación mejorado, dando lugar a una compresión superior sin comprometer la calidad. Su adopción ha sido más lenta debido a la mayor complejidad computacional para la codificación. Aun así, está ganando terreno en servicios de *streaming* que buscan la máxima eficiencia, como Netflix y Vimeo [22].

#### 2.2.3.3. Detalles técnicos y aplicaciones de Opus

Opus es un códec de audio muy versátil y eficiente, diseñado con la capacidad de adaptarse a una gran variedad de aplicaciones de audio en Internet. Es un códec de código abierto, desarrollado conjuntamente por la IETF (*Internet Engineering Task Force*) para proporcionar una calidad de audio superior en una variedad de condiciones de red [23].



Entre las características del códec Opus destaca su capacidad para manejar tanto la música como el habla de manera eficiente sin necesidad de cambiar los códecs para diferentes tipos de audio. Opus puede ajustar dinámicamente las tasas de bits, la resolución del audio y el ancho de banda desde 6 kbps hasta 510 kbps [24], lo que lo hace muy adaptable a las variaciones en la disponibilidad del ancho de banda. Este códec soporta tanto la codificación de audio con baja latencia para aplicaciones en tiempo real como la codificación de alta fidelidad para aplicaciones de música y, además, tiene la capacidad de integrar características de resiliencia ante pérdidas de paquetes, haciéndolo ideal para aplicaciones sobre redes inestables.

Opus es muy utilizado en aplicaciones de comunicación en tiempo real, como VoIP [25]. También se utiliza en aplicaciones de transmisión de música en línea y podcasts. Además, es el códec preferido para juegos en línea y transmisiones en vivo, debido a sus características.

#### 2.2.3.4. *Advanced Audio Coding (AAC)*

*Advanced Audio Coding (AAC)* es un estándar de compresión de audio desarrollado como sucesor del MP3, ofreciendo mejor calidad de sonido a tasas de bits similares. Este códec utiliza técnicas avanzadas como codificación perceptual y segmentación temporal para lograr una mayor eficiencia de compresión [26]. AAC es ampliamente utilizado en aplicaciones de *streaming* de música, transmisión de radio digital y almacenamiento de audio en dispositivos móviles debido a su capacidad para proporcionar una alta calidad de audio en un tamaño de archivo reducido.

#### 2.2.3.5. Comparación de códecs en términos de eficiencia y calidad

En esta sección se comparan tres códecs de vídeo ampliamente utilizados: H.264, VP9 y AV1, junto con los códecs de audio Opus y AAC.

##### 2.2.3.5.1. Comparativa entre H.264, VP9 y AV1

En primer lugar, H.264 sigue siendo el más extendido debido a su compatibilidad universal y excelente equilibrio entre calidad y ancho de banda. Sin embargo, requiere tasas de bits relativamente altas para mantener calidad en resoluciones superiores.

En cuanto a VP9, este códec proporciona una mejora significativa en la eficiencia de compresión sobre H.264, permitiendo hasta un 50% de reducción en la tasa de bits comparado con su predecesor (VP8) [27].

Por último, AV1 ofrece una reducción de hasta el 30% en la tasa de bits en comparación con VP9, aunque es más exigente computacionalmente.

Todos los códecs manejan bien la calidad de vídeo a altas tasas de bits, pero AV1 y VP9 proporcionan una mejor calidad en tasas más bajas comparado con H.264, siendo más robustos ante limitaciones de ancho de banda.

En términos de latencia, H.264 y VP9 tienen latencias comparables cuando se configuran adecuadamente, adecuadas para la mayoría de las aplicaciones de *streaming*, mientras que AV1, aunque ofrece mejor calidad y eficiencia, puede introducir una mayor latencia debido a su complejidad de decodificación, lo que podría ser un problema para el *streaming* en tiempo real.

#### 2.2.3.5.2. Comparativa entre Opus y AAC

En cuanto a eficiencia, Opus es altamente eficiente y adaptable, ideal para una variedad de aplicaciones, desde llamadas VoIP hasta *streaming* de música. AAC también es conocido por su eficiencia en la compresión y es comúnmente utilizado en plataformas de *streaming* debido a su capacidad para proporcionar alta calidad de audio a tasas de bits moderadas.

Teniendo en cuenta la calidad, Opus se destaca en proporcionar una calidad de audio excepcional en todas las tasas de bits, tanto en música como en habla, ajustándose dinámicamente a las condiciones de la red. AAC también ofrece una buena calidad de audio y es ampliamente aceptado en la industria del entretenimiento y en las transmisiones en directo, siendo el estándar en muchos dispositivos y plataformas.

En términos de latencia, Opus tiene una ventaja significativa, soportando modos de ultra baja latencia que son esenciales para aplicaciones en tiempo real como juegos en línea y comunicaciones interactivas. Por otro lado, AAC, aunque adecuado para *streaming* de música y vídeo, generalmente no soporta las latencias extremadamente bajas necesarias para aplicaciones de tiempo real crítico [28].

## 2.2.4. Métricas de calidad para vídeo y audio

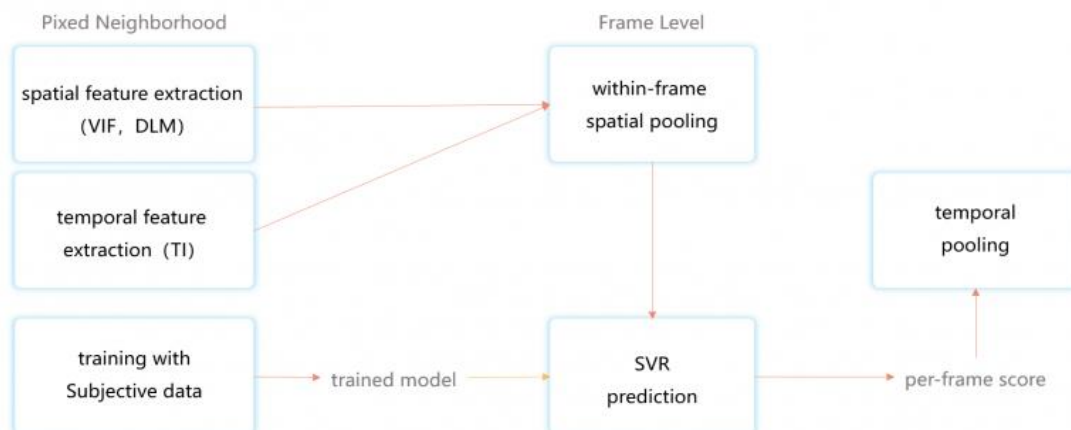
### 2.2.4.1. Descripción de PSNR y VMAF

La Proporción Máxima de Señal a Ruido o PSNR (*Peak Signal to-Noise Ratio*) es una métrica utilizada en ingeniería para medir la calidad de reconstrucción de imágenes y vídeos en comparación a la fuente original. Se calcula usando el logaritmo de la relación entre la máxima energía de una señal (representada generalmente por la intensidad máxima de píxel) y la energía del ruido que afecta a la fidelidad de su representación. PSNR es ampliamente utilizado en el campo del procesamiento de imágenes y vídeo para evaluar la eficacia de los algoritmos de compresión y otras transformaciones en mantener la calidad original. Esta métrica sigue la fórmula descrita en la Ecuación 1.

$$\text{PSNR} = 10 \log_{10} \left( \frac{\text{MAX}^2}{\text{MSE}} \right)$$

*Ecuación 1: Proporción Máxima de Señal a Ruido (PSNR) [29].*

VMAF (*Video Multimethod Assessment Fusion*) es una métrica de evaluación de calidad de vídeo desarrollada por Netflix que utiliza un modelo de aprendizaje automático basado en entrenamiento con datos de percepción visual humana para predecir la calidad visual de un vídeo. Esta métrica considera factores como la nitidez, la textura y los errores de movimiento, entre otros, y es útil para comparar la calidad de diferentes códecs de vídeo o ajustes de calidad dentro del mismo códec, proporcionando una medida más cercana a la percepción visual humana que otras métricas como PSNR. A continuación, en la Imagen 1 se muestra el esquema de esta métrica.



*Imagen 1: Esquema de funcionamiento de VMAF [29].*

#### 2.2.4.2. Descripción de PESQ y POLQA

PESQ (*Perceptual Evaluation of Speech Quality*) es una norma internacional para la evaluación automática de la calidad de voz. Mide la calidad del habla comparando la señal de habla antes y después de la transmisión o compresión, proporcionando una puntuación que refleja la calidad percibida [29]. Es ampliamente utilizada en la industria de las telecomunicaciones para evaluar la calidad de las transmisiones de voz en redes móviles y VoIP, ayudando a determinar la eficacia de los algoritmos de compresión de audio y ajustes de red.

POLQA (*Perceptual Objective Listening Quality Analysis*) es la evolución de PESQ y se utiliza para evaluar la calidad de audio en redes de banda ancha modernas y tecnologías de audio avanzadas. POLQA puede evaluar una mayor gama de distorsiones y una calidad de sonido superior en comparación con PESQ [30], proporcionando una medida precisa de la experiencia auditiva del usuario final.

### 3. Procedimiento

#### 3.1. Software y herramientas utilizadas

##### 3.1.1. Justificación de la selección de herramientas

Para este proyecto, se han elegido las siguientes herramientas debido a sus características y capacidades específicas:

- Wirecast. Originalmente considerado por su interfaz intuitiva y robustas características de transmisión profesional y capacidades de producción en tiempo real. Sin embargo, debido a su costo, se decidió utilizar OBS Studio.
- OBS Studio. Una solución gratuita y de código abierto, ampliamente utilizada para grabación y transmisión en vivo. Ofrece flexibilidad y personalización a través de *plugins* y configuraciones avanzadas, lo que lo convierte en una alternativa viable y accesible a Wirecast.
- YouTube. Utilizado como plataforma de transmisión debido a su amplia accesibilidad y capacidades de *streaming* en tiempo real con alta fiabilidad.
- FFmpeg. Seleccionado por sus avanzadas herramientas de procesamiento de audio y vídeo. Permite la conversión, grabación, y transmisión de contenidos multimedia, además de soportar una gran variedad de formatos y códecs.

### 3.1.2. Configuración y personalización de las herramientas

Como se ha mencionado en un apartado anterior, este proyecto se divide en dos secciones:

- Comparar fragmentos de un vídeo transmitidos a YouTube por OBS con los archivos originales grabados también con OBS.
- Transcodificar estos fragmentos mediante FFmpeg y comparar la calidad con los archivos originales

A continuación, se procede a exponer y explicar el procedimiento de ambas secciones.

#### 3.1.2.1. Hacer *stream* a YouTube con OBS

Para realizar una transmisión en directo a YouTube mediante OBS Studio hay que descargar el software de la página oficial (<https://obsproject.com/es/download>), iniciar sesión en la página web de YouTube (<https://www.youtube.com/>) y efectuar las siguientes configuraciones:

- En OBS, crear una nueva escena y añadir una nueva fuente multimedia que contenga el vídeo que se desea transmitir.
- En el menú de ajustes, en el apartado de emisión, configurar la emisión para transmitir directamente al canal de YouTube, mediante una clave de transmisión cuyo método de obtención se explica más adelante.
- En el mismo menú, en el apartado de salida, modificar los parámetros de salida (códecs, ratio de bits, resolución) a los valores que se pretende evaluar en cada caso. Esto se deberá modificar y verificar antes de cada transmisión para asegurar que se eligen los parámetros adecuados.

Siguiendo todos los pasos y efectuando las configuraciones mencionadas, el resultado debería ser similar al que se puede observar en la imagen 2.

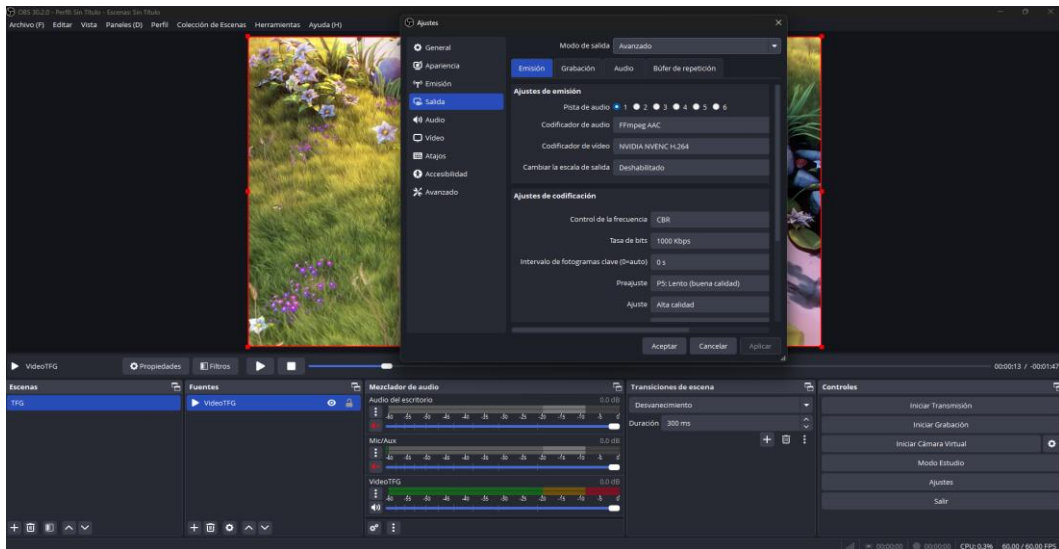


Imagen 2: Interfaz de OBS Studio y ventana de Ajustes.

Para obtener la clave de retransmisión de YouTube solo es necesario iniciar sesión, crear un nuevo evento de emisión en directo y copiar la clave de emisión que se encuentra en el apartado de configuración de la emisión. En la imagen 3 se puede observar el menú de Emisión de YouTube Studio y la clave de acceso utilizada (una vez finalizado este proyecto la clave quedará inservible), así como el título configurado y el vídeo puesto en oculto, de este modo se controla qué transmisión se está realizando y se evita que cualquier otra persona pueda visualizarla.

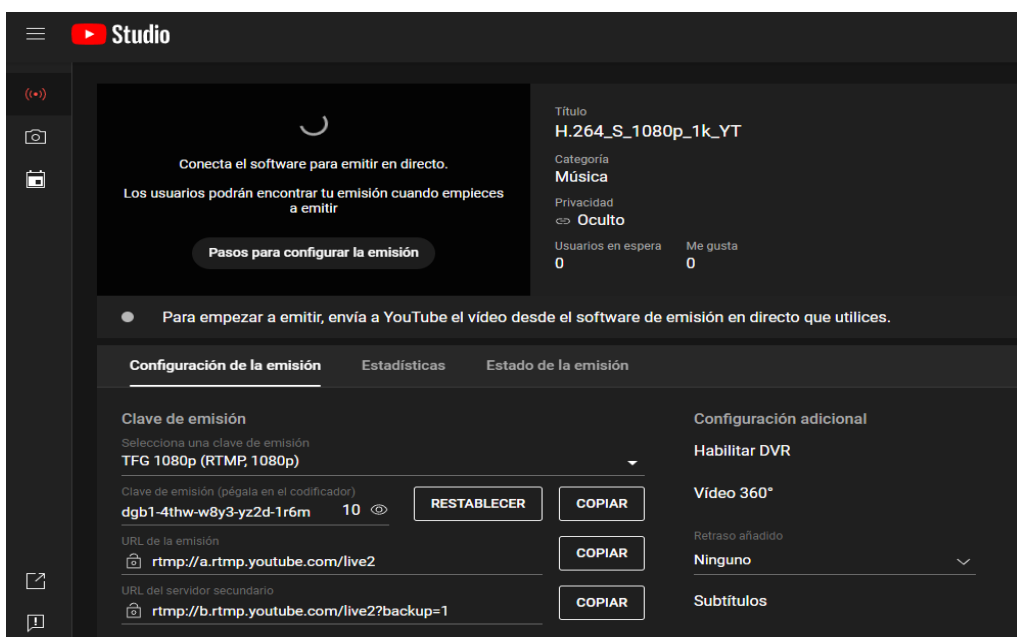


Imagen 3: Ventana de Emisión en Directo de YouTube Studio.

### 3.1.2.2. Transcodificar archivos con FFmpeg

Para utilizar FFmpeg, primero es necesario instalarlo. Se puede descargar desde su web oficial (<https://ffmpeg.org/download.html>). Después de la descarga, se deben seguir las instrucciones específicas para el sistema operativo que se esté utilizando para completar la instalación.

FFmpeg se ejecuta a través de la consola de comandos (cmd) en Windows o la terminal en macOS y Linux. Como este proyecto se desarrolla en Windows, la consola se abre presionando las teclas Win + R o escribiendo “cmd” en la barra inferior de búsqueda.

Si no se ha trabajado con FFmpeg con anterioridad, es recomendable utilizar un tutorial a modo de apoyo. El tutorial consultado durante la realización de este proyecto para aprender a utilizar FFmpeg se encuentra en la web El Terminal de Linux [30].

Para convertir un archivo de vídeo de un formato a otro, por ejemplo, de MP4 a MKV, se utiliza el siguiente comando

```
“ffmpeg -i input.mp4 output.mkv”
```

que contiene los siguientes componentes:

- “ffmpeg”: llama al programa FFmpeg.
- “-i input.mp4”: indica el archivo de entrada.
- “output.mkv”: indica el archivo de salida y su formato.

Las pruebas de vídeo se realizan con un fragmento de 2 minutos de un cortometraje, por tanto, es necesario recortar el vídeo original a solo los primeros 180 segundos. Esto se consigue con el siguiente comando:

```
“ffmpeg -i input.mkv -ss 00:00:00 -t 00:02:00 -c copy output_2min.mkv”
```

- “-ss 00:00:00”: establece el punto de inicio en 0 segundos.
- “-t 00:02:00”: establece la duración final en 2 minutos.
- “-c copy”: copia el audio y el vídeo sin recodificación. De este modo no se pierde calidad al extraer un fragmento del vídeo original.

En cuanto a la resolución, se puede ajustar utilizando el siguiente comando

```
“ffmpeg -i input.mp4 -vf scale=1280:720 output_720p.mkv”
```

cuyos componentes adicionales son los siguientes:

- “-vf scale=1280x720”: un filtro de vídeo que escala la resolución a 720p.
- “output\_720p.mkv”: indica el archivo de salida, su formato y especifica la resolución en su nombre. Nombrar los archivos con sus características relevantes es muy útil a la hora de manejar múltiples archivos y evitar errores o pérdida de datos obtenidos.

Una vez se obtiene el vídeo en el formato y resolución deseados, el siguiente comando muestra cómo transcodificar el archivo a 60 fotogramas por segundo usando el códec H.264 a 1000kbps y el códec AAC a 16kHz:

```
“ffmpeg -i input_720p.mkv -c:v libx264 -b:v 1000k -c:a aac -ar 16000 -r 60  
output_720p_H.264_AAC.mkv”
```

- “-c:v libx264”: define el códec de vídeo a usar. En caso de querer usar VP9 o AV1 se sustituye “libx264” por “libvpx-vp9” o “libaom-av1”
- “-b:v 1000k”: establece el bitrate del vídeo a 1000kbps.
- “-c:a aac”: define el códec de audio a usar, en este caso es AAC. Para Opus se sustituye “aac” por “libopus”.
- “-ar 16000”: establece la frecuencia de muestreo del audio en 16kHz.
- “-r 60”: establece la tasa de fotogramas a 60 fps.

Como en este proyecto no solo se evalúa el vídeo, sino también el audio, y para ello es necesario tener los archivos únicamente de audio. El siguiente comando permite extraer el audio de un archivo de vídeo y guardarlo como WAV:

```
“ffmpeg -i input.mkv -q:a 0 -map a output.wav”
```

- “-q:a 0”: ajusta la calidad del audio. La opción 0 indica la calidad más alta posible, es decir, sin introducir pérdidas.
- “-map a”: selecciona solo el audio para extraer y procesar, no el vídeo.

Con el fin de evaluar la eficiencia de compresión y calidad de los códec de audio AAC y Opus, se utiliza el siguiente comando para transcodificar el audio de los vídeos originales:



```
“ffmpeg -i input.wav -vn -c:a aac -b:a 16k -ac 1 output.aac”
```

- “-vn”: excluye el vídeo del archivo de salida.
- “-ac 1”: establece un solo canal de audio (mono).
- Para poder obtener la métrica PESQ es necesario que ambos audios, el original y el degradado, estén en formato .wav, así que es necesario emplear el primer comando explicado en este apartado para convertir el archivo .aac en .wav.

Los comandos que se han mencionado y explicado son suficientes para realizar todas las pruebas de transcodificación y comparación de los diferentes códecs de vídeo y audio.

### 3.1.3. Proceso de obtención de métricas

Para evaluar la calidad de vídeo y audio en este proyecto, se han utilizado tres métricas principales: PSNR, VMAF y PESQ. A continuación, se describe el proceso y los comandos utilizados para obtener cada una de estas métricas.

#### 3.1.3.1. Método de obtención de PSNR

Para obtener PSNR, se utilizó la herramienta FFmpeg con el siguiente comando:

```
“ffmpeg -i video_original.mkv -i video_degradado.mkv -lavfi  
psnr=stats_file=psnr_logfile.txt -f null -“
```

- “-lavfi psnr=stats\_file=psnr\_logfile.txt”: aplica el filtro PSNR y guarda las estadísticas en un archivo de texto.
- “-f null -”: indica que no se va a generar un archivo de salida, solo se calculan las métricas y se muestran en la misma consola.

Este comando toma el vídeo original y el vídeo degradado, calcula el PSNR y lo muestra en la consola.

#### 3.1.3.2. Método de obtención de VMAF

Para calcular VMAF, se utilizó el modelo VMAF de FFmpeg con el siguiente comando:

```
“ffmpeg -i video_degradado.mkv -i video_original.mkv -lavfi  
libvmaf=model='path=vmaf_v0.6.1.json' -f null -“
```

- “-lavfi libvmaf="log\_path=vmaf\_logfile.json"”: Aplica el filtro VMAF y guarda las estadísticas en un archivo .json.
- “-f null -“: al igual que con PSNR, indica que no se va a generar un archivo de salida, solo se calculan las métricas.

Este comando calcula la puntuación VMAF y la muestra en la consola. Para el correcto funcionamiento del comando es necesario que en “path=” se especifique la ruta en la que se encuentra el archivo “vmaf\_v0.6.1.json”.

### 3.1.3.3. Método de obtención de PESQ

Para calcular PESQ, se utilizó la herramienta PESQ con los siguientes comandos, según la frecuencia de muestreo:

Para 8 kHz:

```
“pesq +8000 audio_original_8k.wav audio_degradado_8k.wav”
```

Para 16 kHz:

```
“pesq +16000 audio_original_16k.wav audio_degradado_16k.wav”
```

Para 16 kHz en modo de banda ancha:

```
“pesq +16000 +wb audio_original_16k.wav audio_degradado_16k.wav”
```

El modo de banda ancha de PESQ está diseñado para evaluar la calidad de las señales de voz en un rango de frecuencia más amplio, de 50 Hz a 7000 Hz, a diferencia del modo normal de banda estrecha que solo evalúa en un rango de 300 Hz a 3400 Hz. Es más adecuado para aplicaciones de voz de alta calidad, como la VoIP y las transmisiones de audio en alta definición.

## 3.2. Detalle de los métodos de codificación y las configuraciones de códecs

### 3.2.1. Configuraciones de códecs y parámetros técnicos

Para la transmisión en directo y la transcodificación de archivos, se seleccionan diferentes códecs y configuraciones específicas para evaluar su rendimiento y calidad en diversos escenarios. A continuación, se detallan las configuraciones utilizadas.

Transmisión en directo:

- Códecs de vídeo: H.264 y AV1.
- Códecs de audio: AAC (debido a las limitaciones del software de transmisión, OBS solo puede transmitir audio en vivo con AAC).
- Resoluciones: 1080p y 720p.
- Tasas de bit: 1000 kbps, 4000 kbps y 8000 kbps.

Si se suman todas las combinaciones, el resultado es de 12 pruebas. Sin embargo, solo es posible obtener métricas de 9 debido a que las transmisiones que se realizan con AV1 a 1080p no resultan consistentes y contienen errores y retardos debidos a la complejidad del códec.

Transcodificación de archivos con FFmpeg:

- Códecs de vídeo: H.264, VP9, AV1.
- Códecs de audio: AAC, Opus.
- Resoluciones: 1080p y 720p.
- Bitrates: 200kps, 500kps y 1000kbps, para los archivos de vídeo.
- Frecuencia de muestreo de audio: 8kHz, 16kHz, 48kHz (la original).

Es importante destacar que para la evaluación del audio se utiliza un fragmento de vídeo diferente al empleado para la evaluación de vídeo. El vídeo que se emplea en las pruebas de vídeo carece de diálogos y contiene únicamente efectos de sonido, lo cual puede afectar la fiabilidad de las pruebas de calidad de audio. Por lo tanto, se selecciona un fragmento de vídeo con diálogos y efectos de sonido para garantizar la precisión de las pruebas de calidad de audio. Los vídeos en cuestión son Big Buck Bunny [31] y Tears of Steel [32].

### 3.2.2. Justificación de la selección de parámetros

La selección de los códecs y parámetros utilizados en las pruebas se fundamenta en varios factores clave que aseguran una evaluación exhaustiva y precisa del rendimiento en diferentes escenarios. En primer lugar, se considera la compatibilidad y el uso común de los códecs. H.264 se elige debido a su amplia adopción y compatibilidad con la mayoría de las plataformas de transmisión y dispositivos. AV1, por otro lado, se selecciona por su capacidad para ofrecer una

mejor compresión y calidad en comparación con H.264, y su creciente adopción en la industria. VP9 también se incluye debido a su uso en plataformas como YouTube, donde ofrece un equilibrio favorable entre calidad y compresión. En cuanto a los códecs de audio, AAC se utiliza por ser el más comúnmente soportado en plataformas de transmisión en directo, mientras que Opus se elige por su alta calidad de audio a bajas tasas de bits, ideal para la transmisión de voz y música en tiempo real.

Las resoluciones de 720p y 1080p se seleccionan por ser estándares comunes en la transmisión en directo y la creación de contenido. Estas resoluciones proporcionan un buen balance entre calidad visual y requerimientos de ancho de banda, permitiendo evaluar el rendimiento en condiciones reales de uso. En términos de *bitrate*, se decide utilizar 1000 kbps, 4000 kbps y 8000 kbps para la transmisión en directo, representando diferentes niveles de calidad y ancho de banda, calidad baja, media y alta. Para la transcodificación de archivos, se utilizan tasas de bits de 200 kbps, 500 kbps y 1000 kbps, lo que permite evaluar la eficiencia de compresión y la calidad visual en diferentes escenarios, desde condiciones de bajo ancho de banda hasta alta calidad.

Se incluyen diferentes frecuencias de muestreo del audio (8 kHz, 16 kHz y 48 kHz) para evaluar la calidad de audio en diversas condiciones. Las frecuencias de 8 kHz y 16 kHz son comunes para la transmisión de voz en tiempo real y son adecuadas para evaluar la calidad de audio en condiciones de ancho de banda limitado. La frecuencia de 48 kHz, estándar para audio de alta calidad, proporciona un punto de referencia para la evaluación comparativa.

### 3.3. Escenarios de prueba para la transmisión

#### 3.3.1. Descripción de los escenarios de prueba

Con el fin de evaluar el rendimiento de los diferentes códecs de vídeo y audio en escenarios de transmisión en directo, se diseñan una serie de pruebas que simulan configuraciones técnicas variadas. Las pruebas se realizan utilizando el software OBS Studio para transmitir fragmentos de vídeo a la plataforma YouTube a dos resoluciones diferentes, tres tasas de bits diferentes y dos códecs diferentes como ya se ha mencionado en el apartado anterior.

Se utiliza una conexión estable para asegurar que las variaciones en los resultados sean principalmente debidas a los códecs y no a las condiciones fluctuantes de la red.

### 3.3.2. Relevancia de cada escenario en la evaluación de códecs

Cada escenario de prueba se selecciona para abarcar una variedad de condiciones que reflejan situaciones comunes en el *streaming* de vídeo y audio y en la transcodificación de archivos multimedia. La evaluación se centra en dos resoluciones (720p y 1080p) y tres tasas de bits para los códecs de vídeo, así como tres frecuencias de muestreo para los códecs de audio en la transcodificación. Esta variedad de configuraciones permite una comparación detallada del rendimiento de los códecs en diferentes condiciones.

Las pruebas a baja tasa de bits sirven para evaluar la eficiencia de los códecs en situaciones donde el ancho de banda es limitado. Es relevante para aplicaciones móviles y áreas con infraestructuras de red deficientes. En este caso, se prioriza la reducción del uso de ancho de banda mientras se mantiene una calidad aceptable.

Las pruebas a media tasa de bits representan una configuración estándar que equilibra calidad y ancho de banda. Es adecuada para la mayoría de las transmisiones en directo, ofreciendo una buena calidad de imagen sin requerir un ancho de banda excesivo.

Las pruebas de alta tasa de bits simulan condiciones de transmisión de alta calidad, donde la prioridad es la calidad visual. Es útil para eventos especiales donde la calidad de imagen es lo más importante, como eventos deportivos o conciertos en vivo.

En cuanto a la resolución, 720p es adecuada para la mayoría de las transmisiones en línea y es menos exigente en términos de ancho de banda y capacidad de procesamiento. Es ideal para transmisiones móviles y situaciones donde el ancho de banda es limitado. Evaluar los códecs a 720p permite comprender su eficiencia en condiciones donde se debe equilibrar la calidad y el uso de recursos. Por otro lado, la resolución 1080p proporciona una calidad de imagen significativamente mejor y es adecuada para pantallas más grandes y transmisiones de alta calidad. Una resolución mayor requiere más ancho de

banda y capacidad de procesamiento. Evaluar los códecs a 1080p permite analizar su rendimiento en condiciones donde la calidad de imagen es prioritaria y el ancho de banda es suficiente para soportar la transmisión de alta definición.

Las frecuencias de muestreo de audio elegidas se deben a diferentes calidades de audio. 8 kHz es la frecuencia de muestreo típicamente utilizada en aplicaciones de voz, como llamadas telefónicas. Proporciona una calidad de audio básica, adecuada para la transmisión de voz, pero insuficiente para la música o vídeos con contenido de audio complejo. Este escenario permite evaluar cómo los códecs manejan la baja calidad de audio en condiciones de ancho de banda extremadamente limitado.

16 kHz representa una calidad de audio intermedia, mejor que la utilizada en llamadas telefónicas y adecuada para voz y música con baja fidelidad. Es un equilibrio entre la calidad de audio y el uso de ancho de banda. Evaluar los códecs a esta frecuencia de muestreo ayuda a entender su eficiencia en aplicaciones donde la calidad de audio es importante pero el ancho de banda es limitado.

Por último, 48 kHz es estándar en la industria para audio de alta calidad, utilizada en música, vídeos y transmisiones de alta fidelidad. Proporciona la mejor calidad de audio y es adecuada para contenido multimedia que requiere una experiencia auditiva rica. Este escenario evalúa cómo los códecs mantienen la calidad de audio superior bajo condiciones óptimas de ancho de banda.

### 3.4. Método de recolección y análisis de datos

#### 3.4.1. Técnicas estadísticas utilizadas para el análisis de datos

La recolección de datos para este proyecto se realiza mediante la anotación sistemática de los resultados obtenidos en cada prueba de transmisión y transcodificación. Cada prueba se ejecuta bajo condiciones controladas y se registran las métricas de calidad de vídeo (PSNR y VMAF) y de audio (PESQ) para cada combinación de códec, resolución y tasa de bits.

Una vez recolectados todos los datos, se procede a su análisis de manera descriptiva y visual mediante el uso de tablas y gráficas. Esto se utiliza para evaluar las diferencias en la calidad de vídeo y audio obtenidas bajo las diferentes configuraciones de prueba.

Los análisis se realizan en varios niveles:

- Comparación entre códecs: evaluando las diferencias de rendimiento entre H.264, VP9 y AV1, y AAC y Opus.
- Comparación entre resoluciones: analizando el impacto de la resolución (720p y 1080p) en la calidad de transmisión y compresión.
- Comparación entre tasas de bits: determinando cómo diferentes tasas de bits (1000 kbps, 4000 kbps y 8000 kbps para transmisión y 200 kbps, 500 kbps y 1000 kbps para transcodificación) afectan a la calidad de los vídeos y audios.

Estas comparaciones permiten identificar patrones y tendencias en el rendimiento de los códecs bajo diversas condiciones, proporcionando una base sólida para las conclusiones del proyecto.

#### 3.4.2. Herramientas de software utilizadas para el análisis de datos

El software y herramientas que se utilizan para la recolección y análisis de datos se seleccionan por su capacidad para manejar grandes volúmenes de datos y realizar análisis estadísticos complejos.

Todos los datos obtenidos de las pruebas de transmisión y transcodificación se registran en Microsoft Excel. Excel facilita la organización de los datos en tablas, permitiendo una visualización clara y estructurada de los resultados. Además, se hace uso de su capacidad de gráficos para generar visualizaciones preliminares de los datos.

Para realizar el análisis, se utiliza la opción de crear gráficos a partir de los datos almacenados en las tablas de Excel. Esto proporciona las herramientas necesarias para ejecutar este análisis, facilitando la identificación de diferencias significativas entre los datos y permitiendo una interpretación estadística clara de los resultados.

Aunque FFmpeg no se utiliza directamente para el análisis estadístico, resulta indispensable en el proceso de transcodificación y en la recolección de las métricas de calidad de vídeo y audio. Se utiliza para convertir los archivos de vídeo y audio a los diferentes códecs y tasas de bits especificadas, así como para extraer las métricas PSNR, VMAF y PESQ.

En cuanto al *streaming*, YouTube sirve como plataforma para recibir y almacenar las transmisiones en directo. Las transmisiones se realizan mediante OBS y se almacenan en YouTube, desde donde se descargan posteriormente para realizar el análisis comparativo con los archivos de grabación originales. Esta plataforma permite una gestión eficiente de los vídeos transmitidos, facilitando el acceso y descarga de los mismos para su análisis.

## 4. Resultados

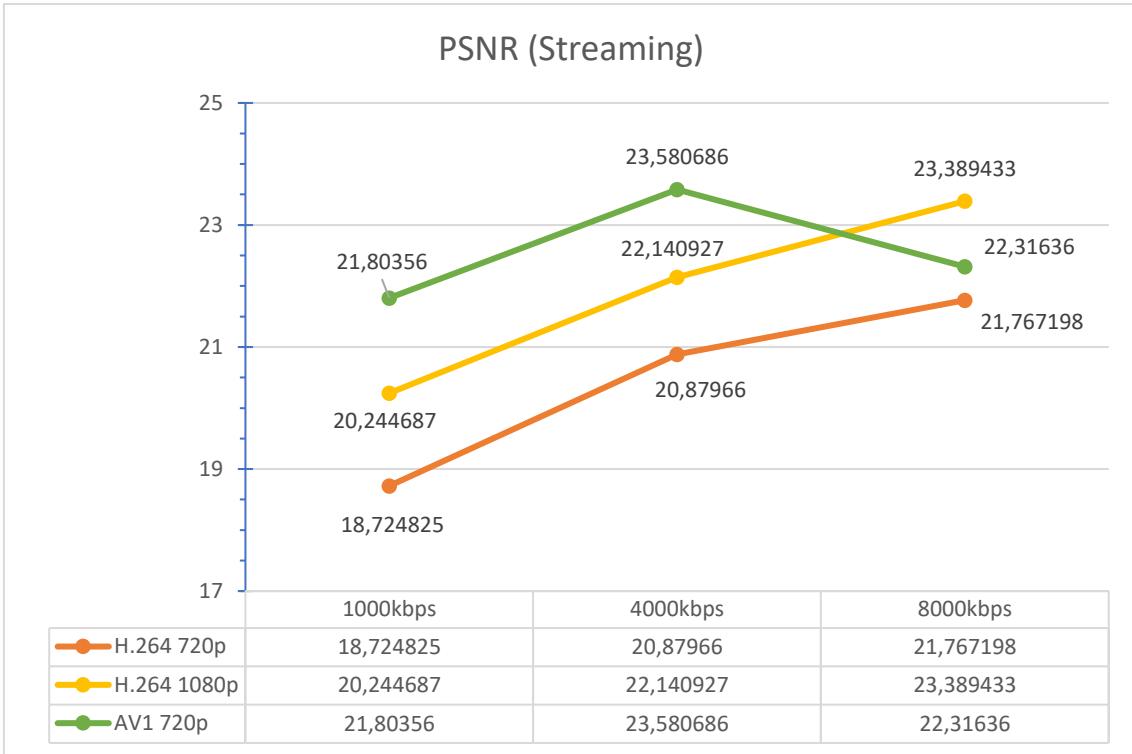
### 4.1. Presentación de los datos obtenidos en las pruebas

#### 4.1.1. Visualización de los datos obtenidos

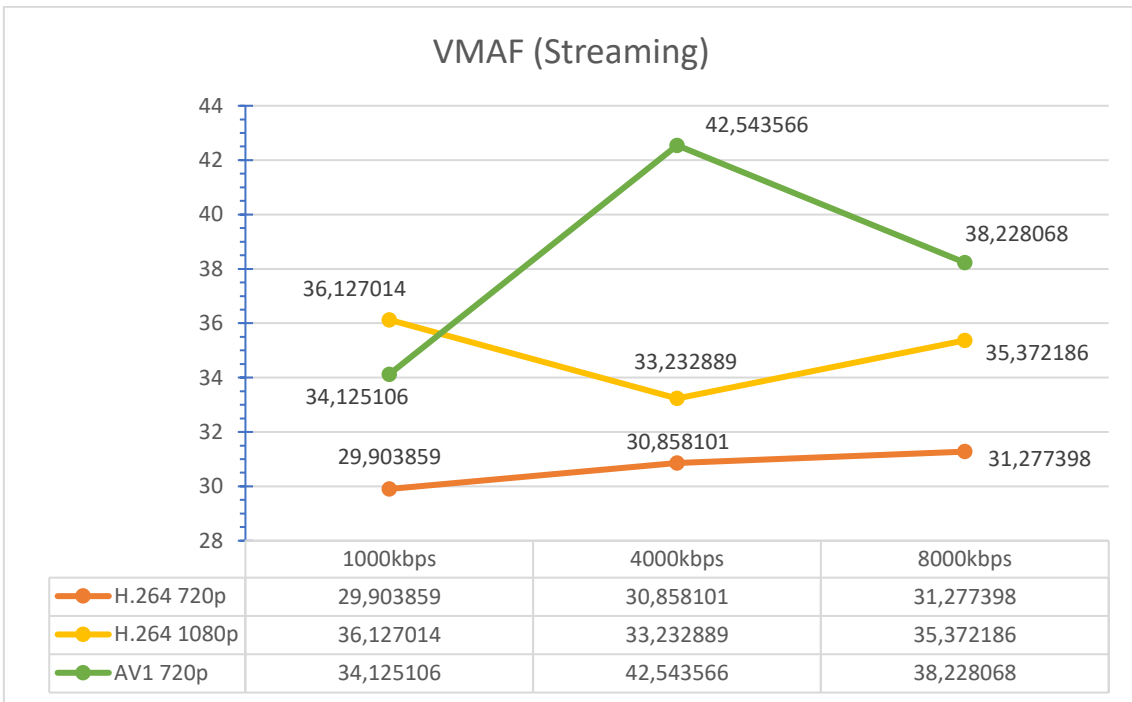
En este apartado se presentan las gráficas y tablas resultantes del análisis de los datos obtenidos durante las pruebas de transcodificación y transmisión de vídeo y audio. Las siguientes figuras ilustran el rendimiento de los códecs H.264, VP9 y AV1 en términos de PSNR y VMAF, así como la evaluación de calidad de audio utilizando PESQ para los códecs AAC y Opus a diferentes frecuencias de muestreo.

Las figuras 1 y 2 muestran los resultados del desempeño de los códecs de vídeo H.264 y AV1 durante la prueba de transmisión. Las figuras 3, 4 y 5 muestran los resultados del desempeño de los códecs de vídeo H.264, VP9 y AV1 durante la prueba de transcodificación, así como el tamaño total de cada uno de los archivos resultantes. Por último, las figuras 6 y 7 muestran los resultados de la prueba de audio realizada con los códecs de audio AAC y Opus, así como el tamaño de cada uno de los archivos resultantes.





*Figura 1: Gráfica de PSNR en la prueba de streaming*



*Figura 2: Gráfica de VMAF en la prueba de streaming*

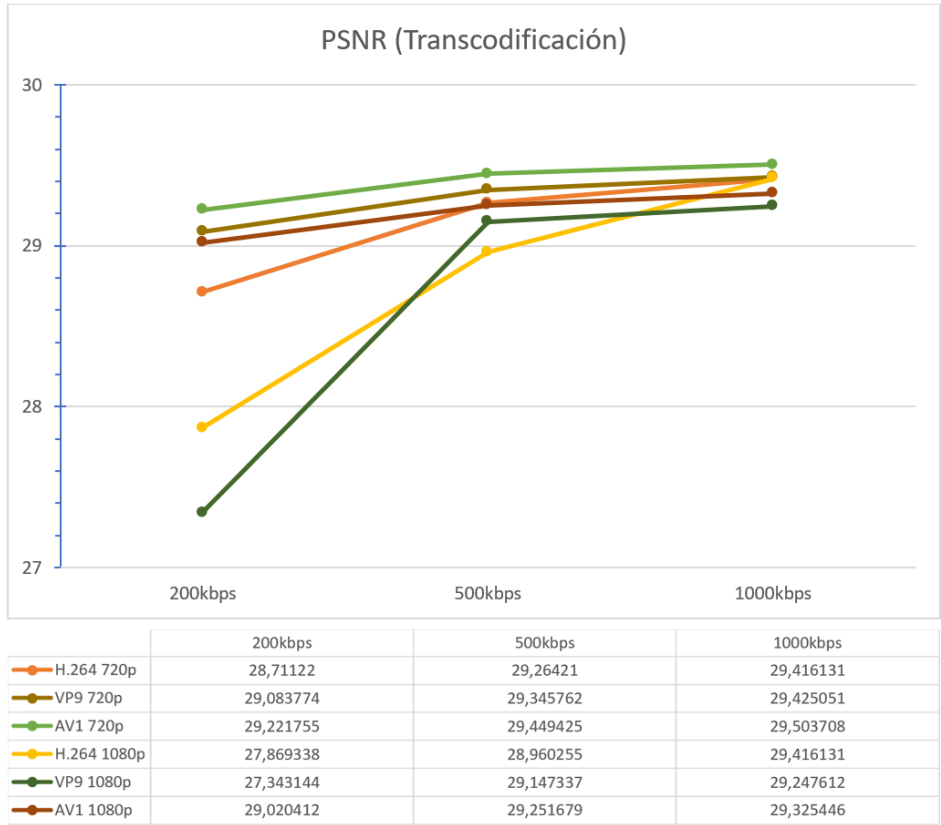


Figura 3: Gráfica de PSNR en la prueba de transcodificación

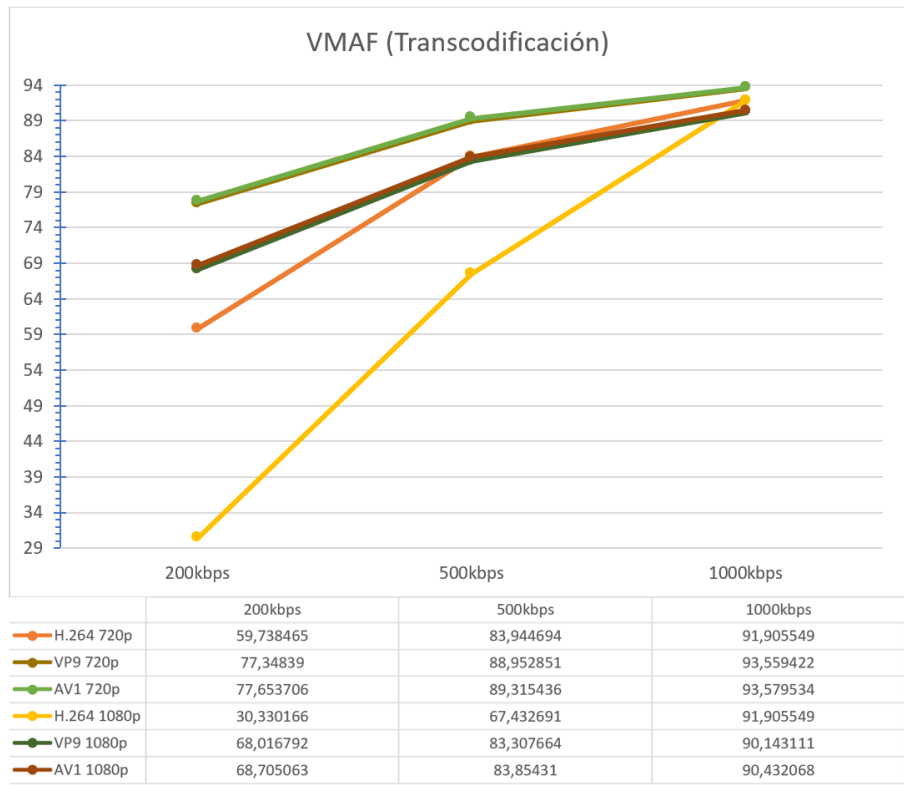


Figura 4: Gráfica de VMAF en la prueba de Transcodificación

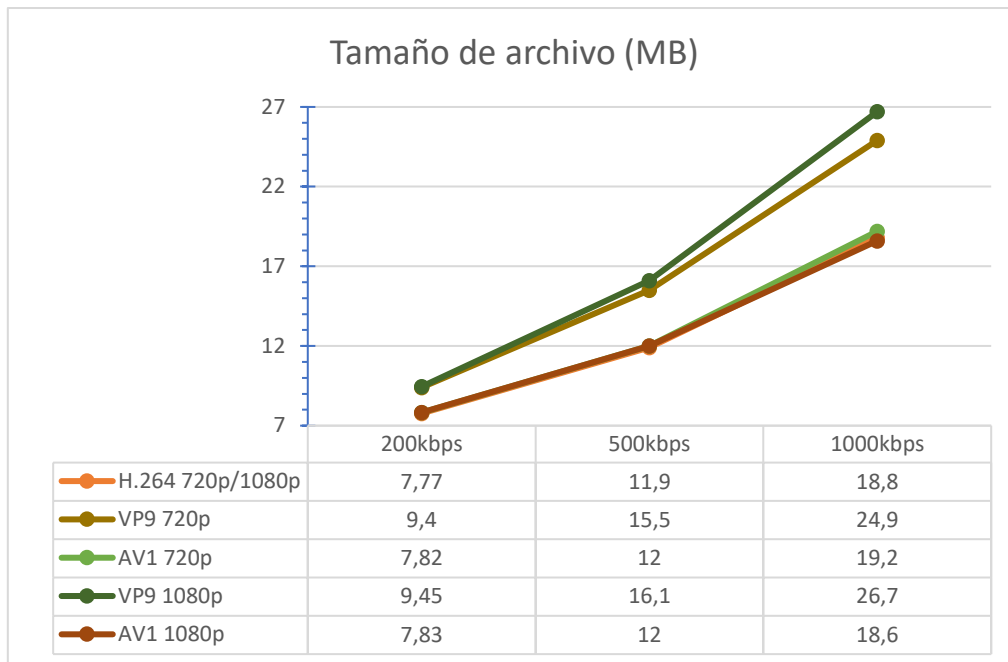


Figura 5: Gráfica del tamaño de los archivos en la prueba de transcodificación

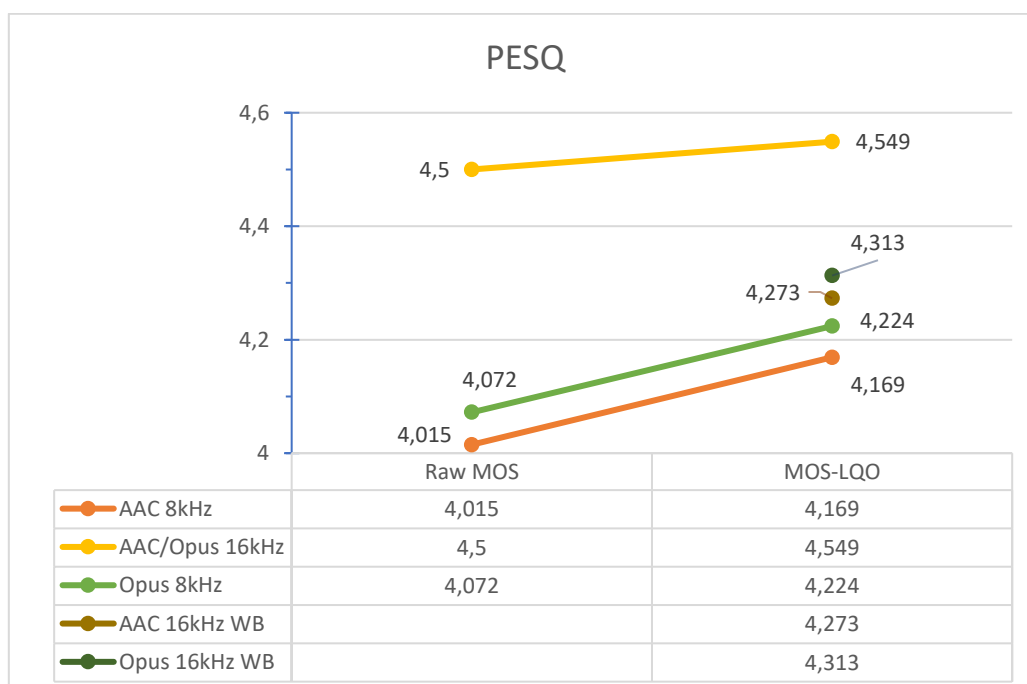


Figura 6: Gráfica de PESQ en la prueba de audio

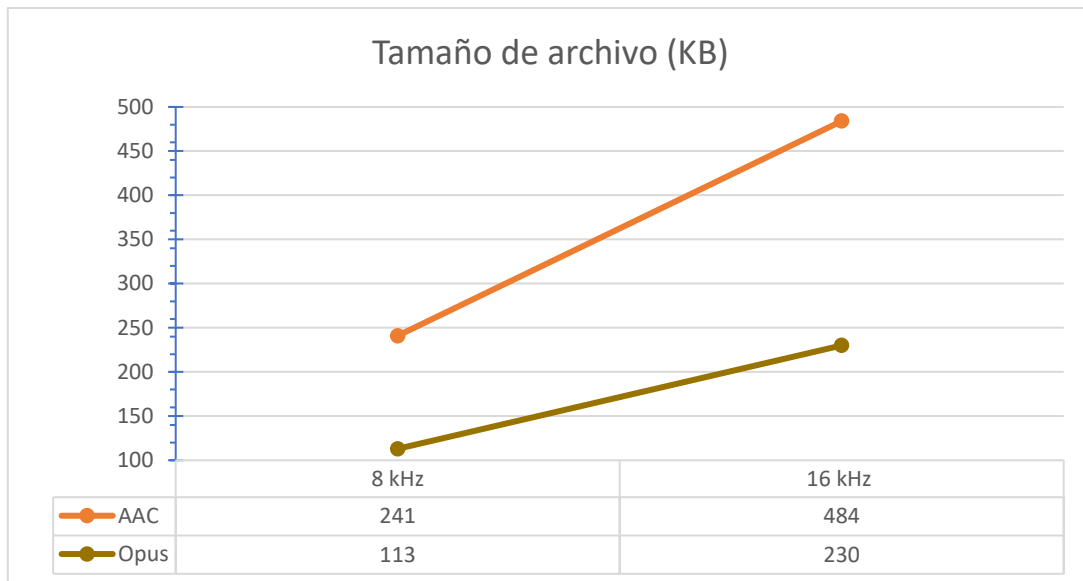


Figura 7: Gráfica del tamaño de los archivos en la prueba de audio

#### 4.1.2. Contexto e interpretación de los datos obtenidos

En primer lugar, se presentan y analizan los resultados de las pruebas de *streaming* realizadas utilizando OBS para transmitir en directo a YouTube, grabando simultáneamente. Las métricas de calidad obtenidas se derivan de la comparación entre el archivo grabado y el archivo descargado de YouTube después de la transmisión.

En la Figura 1 se observa que el códec H.264 a 720p muestra una mejora gradual con el aumento del *bitrate*, pasando de 18.724825 a 21.767198. Con el incremento de la resolución a 1080p también mejora al aumentar el *bitrate*, aunque en menor medida, desde 20.244687 hasta 23.389433. Además, el códec AV1 presenta un comportamiento similar al de H.264, pero con valores iniciales más altos, alcanzando un pico en 4000kbps y disminuyendo ligeramente a 8000kbps.

La Figura 2 indica que el códec H.264 a resolución 720p mejora con el aumento del *bitrate*, aunque los valores absolutos son más bajos de lo esperado, variando de 29.903859 a 31.277398. El mismo códec a resolución 1080p comienza con valores más altos que 720p y muestra una mejora continua con el aumento del *bitrate*. En cuanto al códec AV1, a resolución 720p alcanza un pico en 4000kbps con 42.543566, pero disminuye a 38.228068 en 8000kbps.

Estos resultados reflejan que el aumento del *bitrate* mejora la calidad de la transmisión. Sin embargo, los valores bajos indican problemas de sincronización entre los archivos comparados, lo que sugiere que no estaban perfectamente alineados en el tiempo, como ya se ha comentado en un apartado anterior.

En segundo lugar, se presentan y analizan los resultados de las pruebas de transcodificación, realizadas mediante el uso de la función que tiene FFmpeg para transcodificar a tasas de bits elegibles, presentados en las figuras 3 y 4.

Usando un *bitrate* de 200 kbps se observa que los valores de PSNR y VMAF son considerablemente más bajos para la resolución de 1080p en comparación con la resolución de 720p. Esto sugiere que, a tasas de bits bajas, es más difícil mantener una calidad de vídeo aceptable empleando resoluciones más altas. Por ejemplo, el códec H.264 en 1080p muestra el valor más bajo de PSNR de 27.869338, mientras que AV1 en 720p muestra un valor relativamente alto (29.221755). AV1 en 720p obtiene el valor más alto de VMAF, 77.653706, mientras que H.264 en 1080p muestra el valor más bajo, obteniendo una puntuación de 30.30166. Esto indica que AV1 maneja mejor las condiciones de bajo *bitrate*, especialmente en resoluciones más bajas.

Al incrementar la tasa de bits a 500 kbps, todos los códecs muestran una mejora notable en los valores de PSNR y VMAF. Sin embargo, AV1 sigue mostrando un rendimiento superior en comparación con H.264 y VP9 en ambas resoluciones. En 720p, AV1 alcanza un PSNR de 29.449425, que es el más alto entre los códecs evaluados. AV1 sigue siendo el códec con mejor desempeño, alcanzando un VMAF de 89.315436 en 720p, seguido de VP9 y H.264 en la misma resolución.

En la configuración de alto *bitrate* (1000 kbps) las diferencias en los valores de ambas métricas se reducen significativamente y todos los códecs muestran una calidad muy similar, con valores alrededor de 29.5 para PSNR y 92 para VMAF. AV1 a 720p sigue liderando con un VMAF de 93.579534, pero la diferencia con los otros códecs es menor. Esto sugiere que, a tasas de bits más altas, los códecs son capaces de mantener una alta calidad de vídeo de manera más consistente, independientemente de la resolución.

En cuanto al tamaño de archivos de las pruebas de transcodificación, presentadas en la Figura 5, los datos muestran que, a tasas de bits más bajas, todos los códecs mantienen un tamaño de archivo relativamente manejable, pero a medida que aumenta el *bitrate*, VP9 tiende a generar archivos más grandes, especialmente en resoluciones más altas como 1080p. Esto indica que, aunque VP9 puede ofrecer una calidad visual superior, lo hace a costa de un mayor tamaño de archivo. Por otro lado, H.264 y AV1 muestran un tamaño de archivo más consistente y eficiente en todas las configuraciones de tasa de bits y resolución, lo que los hace más adecuados para escenarios donde el espacio de almacenamiento es limitado.

Por último, observando la Figura 7, se ve que los resultados de PESQ para las pruebas de 8 kHz y 16 kHz muestran que Opus tiene una ligera ventaja sobre AAC en términos de calidad de audio percibida. Utilizando una frecuencia de muestreo de 8 kHz, Opus obtiene un MOS-LQO de 4.224, mientras que AAC se queda en 4.169. A 16 kHz, tanto en modo normal como en banda ancha, Opus sigue superando a AAC. En modo banda ancha (WB), Opus alcanza un MOS-LQO de 4.313 comparado con los 4.273 de AAC.

La eficiencia de compresión de Opus se refleja en los tamaños de archivo más pequeños en comparación con AAC. A 8 kHz, los archivos Opus son casi la mitad del tamaño de los archivos AAC (113 KB frente a 241 KB). A 16 kHz, Opus sigue siendo más eficiente con un tamaño de archivo de 230 KB, mientras que AAC alcanza los 484 KB. Estas observaciones sugieren que Opus no solo ofrece una calidad de audio ligeramente superior, sino que también es más eficiente en términos de almacenamiento, lo que lo convierte en una mejor opción para aplicaciones de transmisión de audio con limitaciones de ancho de banda.

#### 4.1.2.1. Relevancia de los resultados en escenarios de uso real

Los resultados obtenidos de la prueba de *streaming* tienen varias implicaciones prácticas. En primer lugar, los datos muestran que el códec AV1 es capaz de ofrecer una calidad de vídeo superior a H.264, aunque el rendimiento varía significativamente dependiendo del *bitrate*, y debido a la desincronización no se aprecian diferencias mayores en las gráficas de esta primera prueba. Para servicios de transmisión en vivo, donde el ancho de banda puede ser limitado y las condiciones de red pueden fluctuar, los resultados

indican que AV1 es una opción superior en comparación con H.264, especialmente a bajo *bitrate*. AV1 ofrece una mejor calidad visual a tasas de bits más bajas, lo que permite a las plataformas de *streaming* ofrecer una experiencia de usuario de alta calidad incluso en redes con limitaciones de ancho de banda. Sin embargo, esta prueba también determina que el uso del códec AV1 es mucho más exigente y a resoluciones superiores puede no ser una opción, pues requiere de muchos más recursos que H.264.

En términos de almacenamiento y distribución de contenido, los resultados del tamaño de los archivos obtenidos en la prueba de transcodificación destacan la eficiencia de AV1 y H.264 en comparación con VP9. Para servicios que manejan grandes volúmenes de contenido y necesitan optimizar el espacio de almacenamiento, AV1 ofrece una ventaja significativa al mantener una alta calidad de vídeo con tamaños de archivo más manejables. Esto se traduce en menores costos de almacenamiento y una distribución más eficiente del contenido. Sin embargo, al ser el códec más complejo de los tres, puede ser preferible optar por VP9, una opción más compatible con los dispositivos que AV1 y cuya calidad es muy similar, aunque su eficiencia de compresión no sea tan alta.

#### 4.1.2.2. Implicaciones de los datos para futuras configuraciones

Los datos obtenidos proporcionan una base para futuras configuraciones y optimizaciones en la transmisión y transcodificación de vídeo. En primer lugar, los resultados sugieren que, para optimizar la calidad de vídeo en *streaming*, es recomendable utilizar el códec AV1 a tasas de bits bajas y medias. Esto es especialmente relevante para plataformas que operan en mercados con infraestructura de red limitada, donde la eficiencia de la compresión puede mejorar significativamente la experiencia del usuario. Para resoluciones mayores, debido al costo computacional, utilizar el códec H.264 garantiza evitar cualquier tipo de problema en la transmisión en cuanto a la codificación, aunque se requiera un *bitrate* mayor para asegurar una buena calidad de imagen.

Para configuraciones de transcodificación, los datos indican que AV1 debe ser la opción preferida cuando se busca maximizar la calidad del vídeo mientras se minimiza el tamaño del archivo. Esta eficiencia permite una mejor gestión del almacenamiento y puede reducir los costos operativos a largo plazo.

En el ámbito del audio, los resultados de PESQ y el análisis del tamaño de los archivos de audio sugieren que Opus es una opción superior a AAC, especialmente en escenarios donde la eficiencia del ancho de banda y el almacenamiento son relevantes. Opus no solo ofrece una calidad de audio ligeramente superior, sino que también es más eficiente en términos de compresión, por tanto, es una mejor opción para aplicaciones de transmisión de audio.

## 4.2. Comparativas gráficas

### 4.2.1. Análisis de tendencias en los datos

Las comparativas gráficas permiten identificar patrones y tendencias en los datos recopilados. A continuación, se presenta un análisis detallado de las tendencias observadas:

En la prueba de *streaming* las gráficas reflejan mejoras en la calidad con el aumento del *bitrate*, pero también destacan problemas de sincronización en las transmisiones. A pesar de estos problemas, AV1 sigue mostrando una ligera ventaja en calidad sobre H.264, especialmente a 4000kbps. Sin embargo, a 8000kbps, la calidad parece disminuir ligeramente para AV1, lo que sugiere que podría haber otros factores causantes de errores, como la capacidad de procesamiento en tiempo real.

En la prueba de transcodificación las gráficas de PSNR y VMAF muestran una clara tendencia de mejora con el aumento del *bitrate*, como es lógico. En todos los códecs evaluados, tanto a 720p como a 1080p, los valores de PSNR y VMAF también aumentan, indicando una mejor calidad visual. Sin embargo, AV1 muestra una mayor eficiencia en la compresión, especialmente a *bitrates* bajos y medios, superando consistentemente a H.264 y VP9 en ambas resoluciones. La tendencia en los tamaños de archivo indica que, a mayores tasas de bits, VP9 tiende a generar archivos más grandes, especialmente en 1080p. H.264 y AV1 mantienen tamaños de archivo más consistentes y eficientes.

Las gráficas de PESQ muestran que Opus supera a AAC en términos de calidad de audio percibida a frecuencias de muestreo de 8kHz y 16kHz. Además, la eficiencia de compresión de Opus es claramente superior, resultando en archivos de menor tamaño sin comprometer la calidad.



#### 4.2.2. Ventajas y desventajas de cada códec

En base a la información obtenida y los resultados de las pruebas, a continuación se presentan las ventajas y desventajas de cada códec.

H.264:

- Ventajas: amplia compatibilidad, buena calidad a *bitrates* altos, menor tamaño de archivo en comparación con VP9.
- Desventajas: menor eficiencia de compresión en *bitrates* bajos y medios, menor calidad visual comparado con AV1 y VP9.

VP9:

- Ventajas: buena calidad visual, especialmente a *bitrates* medios y altos.
- Desventajas: genera archivos más grandes, mayor demanda de procesamiento, menor eficiencia en *bitrates* bajos que AV1.

AV1:

- Ventajas: mayor eficiencia de compresión, mejor calidad visual a todos los *bitrates*, especialmente en condiciones de ancho de banda limitado.
- Desventajas: mayor demanda de procesamiento, aún no tan ampliamente soportado como H.264.

AAC:

- Ventajas: buena calidad de audio percibida, amplia compatibilidad.
- Desventajas: menor eficiencia de compresión en comparación con Opus.

Opus:

- Ventajas: excelente calidad de audio percibida, alta eficiencia de compresión, menor tamaño de archivo.
- Desventajas: menos compatibilidad en algunas plataformas más antiguas.

##### 4.2.2.1. Resumen de los mejores resultados basado en los datos obtenidos

De acuerdo con los datos obtenidos, el mejor códec de vídeo en términos de calidad y eficiencia de compresión resulta ser AV1 y el mejor códec de audio en términos de calidad percibida y eficiencia de compresión, tanto a 8 kHz como

a 16kHz, resulta ser Opus. Teniendo en cuenta únicamente los códecs analizados en este proyecto.

#### 4.2.2.2. Recomendaciones para la selección de códecs y configuraciones según el tipo de contenido

Para transmisiones en directo con limitaciones de ancho de banda se recomienda utilizar AV1 a resoluciones de 720p y *bitrates* de entre 500kbps y 1000kbps, junto con Opus a 8kHz para optimizar la calidad de vídeo y audio.

Para contenido almacenado o transmitido en condiciones de ancho de banda amplio se recomienda el uso de AV1 o VP9 a resoluciones de 1080p y *bitrates* de 1000kbps, con audio Opus a 16kHz para maximizar la calidad visual y auditiva.

Para escenarios de compatibilidad amplia lo mejor es utilizar el códec H.264 a *bitrates* altos en combinación con el códec de audio AAC para asegurar una calidad aceptable y compatibilidad con la mayoría de las plataformas y dispositivos.

#### 4.2.2.3. Configuraciones que evitar y su explicación

Estas son dos recomendaciones que evitar en cuanto a configuraciones de códecs y casos de uso:

- Evitar los códecs H.264 y VP9 a *bitrates* bajos para resoluciones altas. Los resultados indican una significativa pérdida de calidad visual y un tamaño de archivo no optimizado.
- Evitar AAC a bajas frecuencias de muestreo cuando la eficiencia de compresión es indispensable. Opus ofrece una mejor calidad percibida y tamaños de archivo más pequeños.

## 5. Conclusiones y recomendaciones

### 5.1. Resumen de los hallazgos más relevantes

Durante la realización de este proyecto, se llevaron a cabo una serie de pruebas exhaustivas para evaluar el rendimiento de diferentes códecs de vídeo y audio en diversas configuraciones.

Se ha identificado que AV1 y Opus son los códecs más eficientes para vídeo y audio, respectivamente. AV1 ofrece una mayor calidad de vídeo a tasas de bits más bajas, mientras que Opus proporciona una calidad de audio superior con un menor impacto en el almacenamiento.

Basado en los resultados obtenidos, se han proporcionado recomendaciones claras para la selección de códecs y configuraciones de *bitrate* y resolución. Esto permite a los usuarios optimizar la calidad de transmisión y almacenamiento según sus necesidades específicas.

Los hallazgos sugieren que, para maximizar la calidad de transmisión, es importante elegir configuraciones que equilibren la eficiencia de compresión y la calidad visual, especialmente en condiciones de ancho de banda limitado.

## 5.2. Limitaciones del estudio

A pesar de los hallazgos significativos y las recomendaciones prácticas proporcionadas, existen varias limitaciones que deben ser reconocidas. Estas limitaciones pueden haber influido en los resultados y sugieren áreas para futuras investigaciones.

El hardware utilizado para las pruebas, un ordenador personal y la configuración de red, puede haber limitado la capacidad para realizar transmisiones de alta calidad, especialmente en resoluciones más altas y con códecs más exigentes como AV1. Los recursos de procesamiento limitados pueden haber afectado el rendimiento del codificador y la calidad de la transmisión.

Algunos códecs y configuraciones no fueron compatibles con el software de transmisión utilizado, lo que limitó la posibilidad de realizar pruebas con todos los códecs deseados, como VP9 y Opus para transmisión en directo.

Las pruebas se realizaron en un entorno controlado con condiciones de red relativamente estables. No se exploraron adecuadamente escenarios de red variables, como conexiones móviles, redes públicas, o condiciones de congestión de red, las cuales pueden tener un impacto significativo en la calidad de la transmisión y la eficiencia de los códecs.

El contenido utilizado para las pruebas se limitó a un fragmento específico de vídeo y audio. Diferentes tipos de contenido, como deportes en vivo, conferencias, o vídeos con alta complejidad de movimiento y detalle, pueden mostrar diferentes comportamientos en términos de eficiencia de códecs y calidad de transmisión.

Las pruebas se llevaron a cabo con fragmentos de vídeo de corta duración. Las transmisiones de larga duración pueden presentar desafíos adicionales, como la acumulación de errores de sincronización, fluctuaciones en la calidad de la red, y variaciones en la carga de trabajo del codificador, que no fueron abordados en este proyecto.

Durante las pruebas de transmisión, se observaron problemas de sincronización entre los archivos grabados y descargados de YouTube, lo que afectó las métricas de calidad obtenidas. La falta de sincronización precisa introdujo errores en los valores de PSNR y VMAF, disminuyendo la precisión de los resultados.

La evaluación de la calidad de audio se limitó a las frecuencias de muestreo de 8kHz y 16kHz, y no se incluyó la evaluación de frecuencias más altas como 48kHz debido a las limitaciones del software PESQ. Esto puede haber limitado la capacidad de comparar de manera exhaustiva la eficiencia y calidad de los códecs de audio en diferentes escenarios.

### 5.3. Recomendaciones basadas en los resultados del estudio

#### 5.3.1. Ajustes específicos para la infraestructura existente en la empresa.

Los datos obtenidos a partir de este estudio establecen un punto de partida para realizar recomendaciones específicas que pueden ser implementadas por la empresa PAUXA Films para optimizar su infraestructura de codificación y transmisión de vídeo y audio. A continuación, se detallan las recomendaciones clave basadas en los resultados del análisis de códecs y las pruebas realizadas.

- **Implementar el códec AV1** para las transmisiones y almacenamiento de contenido a *bitrates* medios y altos ( $\geq 1000$  kbps). Los resultados mostraron que AV1 ofrece una calidad superior en comparación con H.264 y VP9 a tasas de bits medias y altas, especialmente en resoluciones de 720p y 1080p. Aunque AV1 puede requerir más poder de procesamiento, la calidad

de vídeo mejorada y la eficiencia de compresión justifican su uso en producciones de alta calidad.

- **Mantener el uso de H.264** para *streaming* en tiempo real y entornos donde la compatibilidad y el rendimiento son críticos. H.264 sigue siendo ampliamente compatible y ofrece un rendimiento sólido con una calidad aceptable a través de diversas configuraciones de red. Este códec es especialmente útil en escenarios donde la baja latencia es un factor clave, como en las transmisiones en vivo.
- **Adoptar el códec Opus** para las transmisiones y grabaciones de audio, especialmente en entornos de ancho de banda limitado. Los resultados de las pruebas de PESQ mostraron que Opus ofrece una calidad de audio ligeramente superior a la de AAC, además de ser más eficiente en términos de tamaño de archivo. Esto es particularmente beneficioso para reducir el consumo de ancho de banda y el espacio de almacenamiento sin sacrificar la calidad del audio.
- **Realizar pruebas periódicas y monitorear el rendimiento** de los códecs y configuraciones utilizadas para adaptarse a los cambios tecnológicos y las necesidades de la audiencia. La tecnología de codificación y las condiciones de la red evolucionan constantemente. Mantenerse actualizado con las mejores prácticas y ajustar las configuraciones según sea necesario permitirá a la empresa PAXUA Films ofrecer siempre la mejor calidad de transmisión y grabación posible.

En conclusión, las recomendaciones específicas para la infraestructura existente en PAXUA Films proporcionan un marco claro para optimizar sus procesos de codificación y transmisión de vídeo y audio. La implementación del códec AV1 para *bitrates* medios y altos garantizará una calidad superior, mientras que el uso continuado de H.264 asegurará compatibilidad y rendimiento en *streaming* en tiempo real. La adopción del códec Opus mejorará la eficiencia y calidad del audio en entornos de ancho de banda limitado. Además, la realización de pruebas periódicas y la monitorización constante del rendimiento permitirán a PAXUA Films mantenerse a la vanguardia de la tecnología, asegurando siempre la mejor calidad en sus producciones audiovisuales.

## 6. Bibliografía

- [1] Latto, N. (28 de julio de 2021). *¿Qué es el streaming y cómo funciona?* AVG. Consultado el 20 de julio de 2024, de <https://www.avg.com/es/signal/what-is-streaming>
- [2] Anitua Valluerca, A. (s.f.). *Protocolos de streaming*. Telefónica Servicios Audiovisuales. Consultado el 20 de julio de 2024, de <https://www.telefonicaserviciosaudiovisuales.com/articulos-de-divulgacion/protocolos-de-streaming/>
- [3] Warren, C. (19 de noviembre de 2012). *The history of Flash: From multimedia hero to security liability*. Mashable. Consultado el 20 de julio de 2024, de <https://mashable.com/archive/history-of-flash>
- [4] Volle, A. (s.f.). *Streaming media*. Encyclopaedia Britannica. Consultado el 20 de julio de 2024, de <https://www.britannica.com/technology/streaming-media>
- [5] Vogel, J. (14 de marzo de 2011). *A History of Media Streaming and the Future*. Streaming Media. Consultado el 20 de julio de 2024, de <https://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=74306>
- [6] Potter, D. (27 de junio de 2011). *Streaming Media: Past, Present, and Future*. Streaming Media. Consultado el 20 de julio de 2024, de <https://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=74735>
- [7] Espinal, E. (4 de abril de 2024). *Understanding Video Compression*. Shrinkit Blog. Consultado el 20 de julio de 2024, de <https://shrinkit.app/blog/understanding-video-compression>
- [8] Cloudflare. (s.f.). *¿Qué es el buffering?* Cloudflare. Consultado el 20 de julio de 2024, de <https://www.cloudflare.com/es-es/learning/video/what-is-buffering/>
- [9] Ortiz, J. (s.f.). *Streaming de vídeo adaptivo*. Glosario de conceptos de vídeo online. Consultado el 20 de julio de 2024, de <https://javierortiz.mx/glosario/conceptos-video-online/streaming-de-video-adaptivo/>
- [10] Krings, E. (21 de octubre de 2022). *What is RTMP?* Maestro Blog. Consultado el 21 de julio de 2024, de <https://www.maestro.io/blog/what-is-rtmp/>
- [11] Skavish, K. (s.f.). *RTMP guide: What is RTMP and how does it work?* Wave.video. Consultado el 21 de julio de 2024, de <https://wave.video/es/blog/rtmp-guide/>
- [12] YouTube. (s.f.). *Cifrar emisiones en directo con RTMPS*. Google Support. Consultado el 21 de julio de 2024, de <https://support.google.com/youtube/answer/10364924?hl=es>

- [13] Elurnet. (s.f.). *¿Qué es y para qué sirve OBS?* Elurnet. Consultado el 21 de julio de 2024, de <https://elurnet.net/que-es-y-para-que-sirve-obs/>
- [14] Bradley University. (s.f.). *IT Tools: OBS*. Bradley University. Consultado el 21 de julio de 2024, de <https://www.bradley.edu/sites/it/departments/ldt/it-tools/obs/>
- [15] Wikipedia. (s.f.). *Wirecast*. Wikipedia. Consultado el 21 de julio de 2024, de <https://en.wikipedia.org/wiki/Wirecast>
- [16] FFMpeg. (s.f.). *About FFMpeg*. FFMpeg. Consultado el 21 de julio de 2024, de <https://www.ffmpeg.org/about.html>
- [16] ITU. (s.f.). *Recomendación ITU-T H.264: Codificación de video avanzada para la comunicación visual*. ITU. Consultado el 21 de julio de 2024, de <https://web.archive.org/web/20210727123623/https://www.itu.int/rec/T-REC-H.264/es>
- [17] Cloudflare. (s.f.). *¿Qué es H.264 (AVC)?* Cloudflare. Consultado el 21 de julio de 2024, de <https://www.cloudflare.com/es-es/learning/video/what-is-h264-avc/>
- [18] Google. (25 de mayo de 2023). *VP9*. Google Developers. Consultado el 21 de julio de 2024, de <https://developers.google.com/media/vp9?hl=es-419>
- [19] Clouinary. (25 de abril de 2024). *Open-Source and Royalty-Free AV1 Compresses Video Efficiently and Effectively*. Clouinary Guides. Consultado el 21 de julio de 2024, de <https://clouinary.com/guides/video-formats/open-source-and-royalty-free-av1-compresses-video-efficiently-and-effectively>
- [20] Gómez Bizcocho, R. (s.f.). *Por qué debes comenzar a usar el códec AV1*. Contenidos Digitales. Consultado el 21 de julio de 2024, de <https://contenidos-digitales.es/por-que-debes-comenzar-a-usar-el-codec-av1/>
- [21] Suárez, J. (27 de febrero de 2023). *Nuevos códec V1, HEVC y VVC*. AVPasión. Consultado el 21 de julio de 2024, de <https://www.avpasion.com/2023-nuevos-codecs-v1-hevc-vvc/>
- [22] Opus. (s.f.). *Opus Interactive Audio Codec*. Opus Codec. Consultado el 21 de julio de 2024, de <https://www.opus-codec.org>
- [23] Internet Engineering Task Force (IETF). (2012). *RFC 6716: Definition of the Opus Audio Codec*. IETF Datatracker. Consultado el 21 de julio de 2024, de <https://datatracker.ietf.org/doc/html/rfc6716>
- [24] Martínez, G. (25 de mayo de 2019). *¿Qué es el códec Opus?* GoToConnect. Consultado el 21 de julio de 2024, de <https://pbx.lat/que-es-el-codec-opus/>

- [25] Rouse, M. (20 de noviembre de 2012). *Advanced Audio Coding (AAC)*. Technopedia. Consultado el 21 de julio de 2024, de <https://www.techopedia.com/definition/218/advanced-audio-coding-aac>
- [26] Clouinary. (1 de mayo de 2024). *VP9 vs H.264*. Clouinary Guides. Consultado el 21 de julio de 2024, de <https://clouinary.com/guides/video-formats/vp9-vs-h-264>
- [27] DigiMad. (s.f.). *AAC (Advanced Audio Coding)*. DigiMad. Consultado el 21 de julio de 2024, de <https://www.digimad.es/aac-advanced-audio-coding.html>
- [28] Liu, Z. (s.f.). *VMAF, SSIM, PSNR: Quality Metrics*. Visionular. Consultado el 22 de julio de 2024, de <https://visionular.ai/vmaf-ssim-psnr-quality-metrics/>
- [29] Jork. (28 de marzo de 2023). *POLQA vs PESQ: Calculando el MOS*. JBlázquez. Consultado el 22 de julio de 2024, de <https://jblazquez.es/polqa-vs-pesq-calculando-el-mos/>
- [30] Manz. (s.f.). *Tutorial de ffmpeg*. Terminal de Linux. Consultado el 16 de mayo de 2024, de <https://terminaldelinux.com/terminal/multimedia/ffmpeg/>
- [31] Goedegebure, S. (2008). *Big Buck Bunny*. Blender.org. Consultado el 14 de mayo de 2024, de <https://peach.blender.org/>
- [32] Hubert, I. (2012). *Tears of Steel*. Blender.org. Consultado el 14 de mayo de 2024, de <https://mango.blender.org/>



## **7. Listado de Anexos**

La carpeta principal de los ficheros anexos se denomina "Ficheros Anexos" y está organizada en varias subcarpetas y un archivo Excel, cada uno conteniendo los materiales y datos utilizados en este proyecto.

### **ANEXO 1: Carpeta de programas y herramientas utilizadas**

La carpeta "Programas y Herramientas" incluye todo el software utilizado: FFmpeg, PESQ, VMAF y el instalador de OBS.

### **ANEXO 2: Carpeta de fragmentos de audio utilizados**

La carpeta llamada "Pruebas Audio" se divide en tres subcarpetas: "AAC", "Opus" y "Audios Originales".

Las carpetas "AAC" y "Opus" contienen archivos de audio transcodificados a dos frecuencias de muestreo, en formatos .aac o .opus y .wav. Dentro de "Audios Originales" están los archivos de audio original a ambas frecuencias de muestreo.

### **ANEXO 3: Carpeta de fragmentos de vídeo utilizados**

La carpeta "Pruebas Vídeo" está estructurada en "Streaming", "Transcodificación" y "Vídeos Originales".

La carpeta "Streaming" incluye carpetas para AV1 y H.264, con archivos de vídeo a diferentes resoluciones y tasas de bits, y una subcarpeta "YT" con los vídeos descargados de YouTube.

La carpeta "Transcodificación" contiene subcarpetas para AV1, VP9 y H.264, cada una con vídeos transcodificados a diferentes resoluciones y tasas de bits.

La carpeta "Vídeos Originales" contiene los vídeos originales utilizados en las pruebas.

## ANEXO 4: Análisis de métricas y resultados

El archivo "Medidas.xlsx" ubicado en la carpeta principal denominada "Ficheros Anexos" contiene hojas con datos y gráficas de las pruebas de *streaming*, transcodificación y audio realizadas.

## ANEXO 5: Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2030

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				x
ODS 2. Hambre cero.				x
ODS 3. Salud y bienestar.			x	
ODS 4. Educación de calidad.			x	
ODS 5. Igualdad de género.				x
ODS 6. Agua limpia y saneamiento.				x
ODS 7. Energía asequible y no contaminante.				x
ODS 8. Trabajo decente y crecimiento económico.	x			
ODS 9. Industria, innovación e infraestructuras.	x			
ODS 10. Reducción de las desigualdades.				x
ODS 11. Ciudades y comunidades sostenibles.			x	
ODS 12. Producción y consumo responsables.		x		
ODS 13. Acción por el clima.			x	
ODS 14. Vida submarina.				x
ODS 15. Vida de ecosistemas terrestres.				x
ODS 16. Paz, justicia e instituciones sólidas.				x
ODS 17. Alianzas para lograr objetivos.		x		

A continuación, se describe la alineación de este TFG con los ODS con un grado de relación más alto.

### ODS 8: Trabajo decente y crecimiento económico.

La implementación de tecnologías avanzadas en la transmisión y compresión de vídeos puede generar nuevas oportunidades laborales en el sector tecnológico y audiovisual. Además, optimizar el uso del ancho de banda y los recursos tecnológicos puede contribuir al crecimiento económico, al hacer las transmisiones más accesibles y eficientes.

### **ODS 9: Industria, innovación e infraestructuras.**

Este proyecto fomenta la innovación en el sector de las tecnologías de la información y comunicación al evaluar y recomendar el uso de códecs de última generación. Además, apoya el desarrollo de infraestructuras digitales eficientes, esenciales para el avance de las industrias creativas y de entretenimiento.

### **ODS 12: Producción y consumo responsables.**

Al optimizar los métodos de compresión y transmisión de vídeo y audio se promueve un uso más responsable de los recursos tecnológicos y la infraestructura de internet. Esto puede ayudar a reducir el desperdicio digital y mejorar la sostenibilidad de las producciones audiovisuales.

### **ODS 17: Alianzas para lograr los objetivos.**

El desarrollo y la implementación de estándares abiertos y eficientes para la compresión y transmisión de contenidos multimedia pueden fomentar alianzas entre diferentes actores de la industria tecnológica y audiovisual, esenciales para lograr avances significativos y sostenibles en el sector.