

Implementación y evaluación de un prototipo de servidor de streaming adaptativo bajo condiciones de red simuladas

Implementation and Evaluation of an Adaptive Streaming Server Prototype under Simulated Network Conditions

Kevin Junior Chocho Quizhpi¹, Juan José Pazmiño Gordillo¹, Lizbeth Rocío Ebla Yerovi²

Cita:

Chocho Quizhpi, K. J., Pazmiño Gordillo, J. J., & Ebla Yerovi, L. R. (2026). Implementación y evaluación de un prototipo de servidor de streaming adaptativo bajo condiciones de red simuladas. *TESLA Revista Científica*, 6(2), e706.
<https://doi.org/10.55204/trc.v6i2.e706>

Recibido: 08/05/2026

Revisado: 18/05/2026 al 30/06/2026

Corregido: 17/07/2026

Aceptado: 24/07/2026

Publicado: 31/07/2026

Licencia:

Los contenidos de este artículo están bajo una licencia Creative Commons Attribution 4.0 International (CC BY 4.0). Los autores conservan los derechos morales y patrimoniales de sus obras.

The contents of this article are under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. The authors retain the moral and patrimonial rights of their works.

¹ Escuela Superior Politécnica de Chimborazo (ESPOCH), Sede Morona Santiago, Macas, Ecuador.

² Escuela Superior Politécnica de Chimborazo (ESPOCH), Facultad de Informática y Electrónica, Riobamba, Ecuador.

¹kevin.chocho@esepoch.edu.ec; ¹jose.pazminio@esepoch.edu.ec; ²lizbeth.ebla@esepoch.edu.ec

¹0009-0009-9026-7932; ¹0000-0001-8887-0788; ²0009-0006-9230-087X

Resumen: Las instituciones educativas que cuentan con una infraestructura de red limitada necesitan alternativas locales para distribuir contenido audiovisual sin depender por completo de plataformas externas. En este estudio se implementó y evaluó un prototipo de servidor de streaming adaptativo basado en MPEG-DASH, sometido a condiciones de red simuladas y controladas. La investigación tuvo un enfoque cuantitativo, con un diseño experimental y comparativo. El prototipo se desplegó en Ubuntu 24.04.4 LTS y utilizó Nginx, FFmpeg, Shaka Packager, Flask y dash.js. Para las pruebas se establecieron cuatro escenarios de ancho de banda: 2,6; 1,4; 0,8 y 0,5 Mbps, con una latencia de 400 ms y una pérdida de paquetes del 0,1 %. Cada escenario se ejecutó diez veces durante cinco minutos. En total, se obtuvieron 12108 registros de telemetría. Los resultados mostraron que el bitrate promedio de video disminuyó de 1805,64 a 483,00 kbps, mientras que el retardo de inicio osciló entre 2,77 y 4,43 segundos. No se presentaron eventos de stalling y el MOS 0.46 se mantuvo entre 4,38 y 4,62. Se concluye que el mecanismo ABR de dash.js respondió de manera estable a las restricciones y mantuvo la continuidad de reproducción. Constituye una base reproducible para futuras evaluaciones institucionales.

Palabras clave: streaming adaptativo; MPEG-DASH; calidad de experiencia; calidad de servicio; redes institucionales.

Abstract: Educational institutions with limited network infrastructure need local alternatives to distribute audiovisual content without relying entirely on external platforms. This study implemented and evaluated an adaptive streaming server prototype based on MPEG-DASH under simulated and controlled network conditions. A quantitative approach with an experimental and comparative design was applied. The prototype was deployed on Ubuntu 24.04.4 LTS using Nginx, FFmpeg, Shaka Packager, Flask, and dash.js. Four bandwidth scenarios were configured: 2.6, 1.4, 0.8, and 0.5 Mbps, with 400 ms latency and 0.1 % packet loss. Each scenario was executed ten times for five minutes. A total of 12,108 telemetry records were collected. The results showed that the average video bitrate decreased from 1,805.64 to 483.00 kbps, while startup delay ranged from 2.77 to 4.43 seconds. No stalling events were recorded, and the 0.46 MOS remained between 4.38 and 4.62. It was concluded that the dash.js adaptive bitrate mechanism responded consistently to network restrictions and maintained playback continuity. Therefore, the developed prototype provides a reproducible foundation for future institutional evaluations and for further studies focused on local adaptive streaming services in educational environments with limited connectivity and restricted technological resources. Its implementation also demonstrates the feasibility of conducting controlled performance tests locally.

Keywords: adaptive streaming; MPEG-DASH; quality of experience; quality of service; institutional networks.

INTRODUCCIÓN

La transmisión de video se ha convertido en uno de los servicios más exigentes de las redes IP debido a la combinación de grandes volúmenes de datos, sensibilidad al retardo y expectativas crecientes de continuidad. La participación del video en el tráfico global aumentó de forma sostenida durante la última década y continúa impulsando la optimización de las cadenas de codificación, distribución y reproducción (Peroni & Gorinsky, 2025; Lorenzi et al., 2024). El video, dentro del contexto educativo, cumple funciones distintas entre sí: se usa en el aula, se aloja en repositorios institucionales, sirve para capacitación y también para difundir resultados científicos. El problema está en que parte de esa distribución termina dependiendo de plataformas externas, y ahí la institución pierde margen de maniobra, pues no decide cómo operan esas plataformas, qué políticas de acceso aplican ni cuánto consumo de Internet le exigen a quien las utiliza.

Estas necesidades se relacionan con el impacto de los servicios de telecomunicaciones y las TIC en la educación superior, así como con la evaluación de requerimientos tecnológicos y la planificación de infraestructura

de red para servicios educativos (Santillán-Lima et al., 2017; Santillán Lima et al., 2016).

El streaming adaptativo sobre HTTP nació como respuesta a un problema concreto: la transmisión a tasa fija no se ajustaba a las condiciones reales de la red, y eso se traducían en cortes o en una calidad pobre para quien tenía poco ancho de banda, por tanto, en lugar de enviar un único flujo, el contenido se codifica en varias representaciones, que luego se fragmenta en segmentos y luego se publica por medio de un servidor. El cliente descarga un manifiesto que describe las alternativas disponibles y, mediante un algoritmo de bitrate adaptativo (Adaptive Bitrate, ABR), decide qué representación solicitar para cada segmento. Esta separación permite el uso de la infraestructura HTTP existente, facilita la interoperabilidad y traslada la decisión de calidad al reproductor (Bentaleb et al., 2018; Timmerer et al., 2025).

MPEG-DASH formaliza este proceso a través del archivo de Descripción de Presentación de Medios (MPD), conjuntos de adaptación y una línea de tiempo de segmentos. El cliente puede cambiar la representación sin necesidad de reiniciar la sesión, siempre que las opciones estén temporalmente alineadas. La decisión se apoya en estimaciones de throughput, nivel del búfer y comportamiento reciente de las descargas. En redes que son estables, el algoritmo permite la incrementación del bitrate para mejorar la nitidez; pero restricciones o caídas, reduce la representación para poder preservar la continuidad de la transmisión (Iren & Kantarci, 2022; Nguyen Viet et al., 2022).

El desempeño de un servicio adaptativo no puede valorarse únicamente mediante métricas de red. La calidad de servicio (Quality of Service, QoS) describe condiciones técnicas como ancho de banda, retardo, pérdida de paquetes y throughput, mientras que la calidad de experiencia (Quality of Experience, QoE) integra la forma en que esas condiciones se reflejan en la sesión de reproducción. Entre los indicadores más utilizados se encuentran el retardo de inicio, el bitrate alcanzado, las conmutaciones de calidad, los eventos de rebuffering y las puntuaciones de opinión media (Mean Opinion Score, MOS) (Kalan & Dulger, 2024; Pant et al., 2025).

La relación entre QoS y QoE no es lineal. Una disminución importante del bitrate no necesariamente produce una reducción proporcional de la valoración global si el reproductor evita interrupciones, mantiene estable el nivel de calidad y limita los tiempos de espera. Por el contrario, una representación de mayor resolución puede ofrecer una experiencia deficiente cuando se acompaña de cambios frecuentes o vaciamiento del búfer. Los modelos paramétricos de la recomendación ITU-T P.1203 permiten estimar esta experiencia a partir de información de audio, video y eventos de reproducción, sin depender exclusivamente de evaluaciones subjetivas con usuarios (International Telecommunication Union, 2017; Robitza et al., 2020).

Aunque la literatura reciente ha propuesto algoritmos ABR avanzados, mecanismos de predicción de ancho de banda y estrategias de codificación adaptativa, gran parte de las evaluaciones se realiza en infraestructuras robustas o en bancos de prueba con capacidad suficiente para múltiples representaciones (Huu et al., 2024; Mueller et al., 2024; Kang & Chung, 2022). Existe menor evidencia sobre el comportamiento de soluciones de código abierto en redes institucionales pequeñas, donde el enlace puede limitarse a valores cercanos al bitrate mínimo requerido y donde la institución necesita conocer previamente el impacto de cada nivel de capacidad.

Esta brecha es relevante en el contexto educativo ecuatoriano. La conectividad limitada, el equipamiento insuficiente y una administración deficiente del tráfico dificultan que las instituciones desplieguen sus propios servicios audiovisuales, lo que mantiene la dependencia de proveedores externos (Collahuazo-Cuases, 2024; Jácome Paredes & Naranjo Pachacama, 2022; Zambrano, 2025). Aunque existen plataformas locales de video bajo demanda, hay una falta de caracterización experimental que vincule restricciones controladas de red con métricas objetivas de experiencia, además de que se pueda reproducir sin infraestructura propietaria (Tenezaca & Paguay, 2023).

La investigación partió de la siguiente pregunta: ¿qué pasa con el rendimiento y con la calidad de experiencia de un servidor de streaming adaptativo cuando el ancho de banda se restringe de forma progresiva, manteniendo fijas la latencia y la pérdida de paquetes? Para responderla no bastaba con simular condiciones ideales, así que se optó por construir un prototipo MPEG-DASH y probarlo sobre una red local controlada, bajo cuatro escenarios distintos. Se midieron los siguientes parámetros: retardo de inicio, bitrate promedio, cuántas veces cambiaba la calidad y con qué frecuencia lo hacía, throughput y, finalmente, la puntuación

MOS O.46.

El aporte de este trabajo se puede resumir en tres partes, siendo la primera la arquitectura, donde se monta el servidor con herramientas como Ubuntu, Nginx, FFmpeg, Shaka Packager, Flask y dash.js. La segunda es la calibración de los escenarios de prueba, esto se realizó a partir del bitrate de cada representación, del audio y de un margen de encapsulación que se deriva del perfil Main de H.264. Mientras que la tercera, es la recolección de datos por medio del cliente con las métricas de QoE, verificando el enlace de forma independiente con iperf3. Con esa combinación no solo se puede decir si el servicio funciona o no, sino entender cómo se degrada, que tan estable puede permanecer y cuál sería la capacidad mínima que una institución debería garantizar antes de poner esto en producción.

MÉTODOS Y RECURSOS

Diseño de la investigación

El estudio se planteó como un diseño cuantitativo, experimental y comparativo, con alcance descriptivo-comparativo. Se tomó como variable independiente el ancho de banda disponible en la interfaz de red del servidor; la latencia, en cambio, se dejó fija en 400 ms y la pérdida de paquetes en 0,1 %, con el fin de que las diferencias observadas entre escenarios pudieran asociarse a la capacidad del enlace y no a otros factores del canal. Para caracterizar el comportamiento del prototipo en cada condición se midieron seis variables de respuesta: el retardo de inicio, el bitrate promedio reproducido, el número de cambios de representación por sesión, la frecuencia de esos cambios por minuto, el throughput efectivo y, por último, la puntuación MOS O.46 calculada a partir del modelo ITU-T P.1203.3.

El tiempo de reproducción por sesión fue de 300 s, con la configuración de red fija. Los escenarios se repitieron diez veces, reiniciando las condiciones y verificando el enlace antes de cada ejecución. Dash.js generó, segundo a segundo, los registros JSON que formaron la población. Cada escenario contribuyó al número total de registros entre 3022 y 3040, que fue ligeramente superior al nominal de 12000 porque cada sesión registró unos segundos adicionales sobre los 300 planificados. Se dividieron los datos recopilados en cuatro estratos diferentes y se compararon en función del escenario.

El estudio no tuvo participantes humanos y no recopiló ningún dato personal. El análisis se basó exclusivamente en la telemetría técnica generada por el reproductor y las herramientas de red. Repetir los procedimientos, aislar el enlace de Internet y documentar cada configuración redujo las fuentes externas de variación.

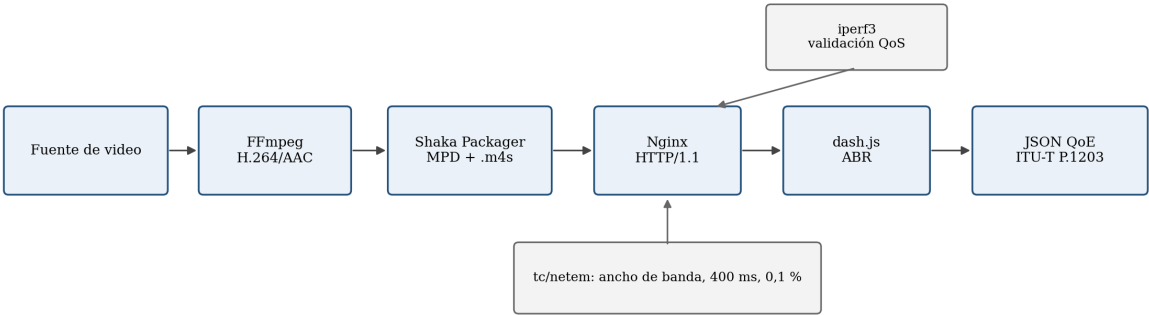
Entorno experimental y arquitectura del prototipo

La topología estuvo compuesta por dos equipos conectados directamente mediante cable Ethernet en una LAN aislada. El servidor utilizó un ASUS TUF Gaming F16 con procesador Intel Core i7-14650HX, 16 GB de RAM DDR5 y almacenamiento de 1 TB, ejecutando Ubuntu 24.04.4 LTS en la dirección 192.168.10.20/24. El cliente fue un ASUS VivoBook X513EP con Intel Core i7-1165G7, 24 GB de RAM DDR4 y SSD NVMe, ejecutando Windows 10 Pro 22H2 en la dirección 192.168.10.30/24.

La distribución se realizó mediante HTTP/1.1 sobre TCP/IP, puerto 80. Nginx publicó el manifiesto MPD y los segmentos; FFmpeg codificó la señal de video en H.264; Shaka Packager produjo la estructura MPEG-DASH; una API REST en Flask coordinó el inicio y la detención de los procesos; y dash.js 5.0.0 ejecutó el algoritmo ABR en el navegador. Traffic Control (tc/netem) aplicó la tasa, el retardo y la pérdida de paquetes sobre la interfaz del servidor mediante una disciplina de cola aplicada a la raíz, que actúa sobre el tráfico de salida.

El manifiesto dinámico tenía cuatro representaciones de video de cada posición de video en el tiempo, 1280 × 720 a 2000 kbps, 854 × 480 a 1000 kbps, 640 × 360 a 500 kbps y 426 × 240 a 250 kbps. El audio estaba codificado en estéreo AAC-LC a 128 kbps y 48 kHz. Los segmentos tenían una duración de cuatro segundos. En una sesión de cinco minutos, el cliente tenía alrededor de 75 oportunidades de selección, suficientes para hacer una comparación de inicio y la convergencia del algoritmo.

Figura 1
Arquitectura funcional del prototipo y flujo de obtención de métricas



Nota. Elaboración propia.

Tabla 1
Componentes principales del prototipo

Componente	Versión / configuración	Función en el experimento
Ubuntu	24.04.4 LTS	Sistema operativo del servidor y base para la configuración de red.
Nginx	1.24.x	Publicación HTTP del MPD, segmentos y aplicaciones web.
FFmpeg	4.4.6 / H.264	Captura y codificación de las representaciones de video y audio.
Shaka Packager	3.7.2	Empaquetado, segmentación y generación del manifiesto MPEG-DASH.
Flask	API REST	Orquestación de procesos, catálogo y control de transmisiones.
dash.js	5.0.0	Reproducción adaptativa y captura de telemetría QoE.
tc/netem	Kernel Linux	Aplicación de ancho de banda, latencia y pérdida de paquetes.

Nota. La arquitectura completa se ejecutó en una LAN sin acceso a Internet. Elaboración propia.

Construcción de los escenarios de red

Los escenarios se definieron a partir del bitrate nominal de cada representación y del flujo de audio. Para evitar que el límite de red coincidiera con la suma elemental de ambos flujos, se incorporó un factor de 1,2. Ese valor corresponde a la relación entre los límites de bitrate de la capa de abstracción de red (NAL) y de la capa de codificación de video (VCL) que establece el perfil Main de H.264, fijados en 1200 y 1000 kbps respectivamente para el nivel considerado; el cociente entre ambos (1200/1000) refleja la sobrecarga máxima de encapsulación prevista por la norma (International Telecommunication Union, 2024, §A.3.1). La capacidad objetivo de cada escenario se calculó mediante la expresión:

$$BW_i = 1,2 \times (R_{video,i} + 128 \text{ kbps})$$

Con este procedimiento, la representación de 720p requirió aproximadamente 2,6 Mbps; la de 480p, 1,4 Mbps; la de 360p, 0,8 Mbps; y la de 240p, 0,5 Mbps. La latencia y la pérdida se mantuvieron en valores fijos, para que la comparación reflejara principalmente las diferencias asociadas a la capacidad disponible. La configuración que se aplicó fue por reglas netem equivalentes a delay 400ms, loss 0.1 % y rate según el escenario. Estos valores corresponden a la configuración impuesta en la disciplina de cola y se mantuvieron constantes en los cuatro escenarios. Como control de las condiciones de QoS, cada repetición incluyó una medición de retardo de ida y vuelta con ping (320 paquetes). El RTT medio aumentó de 424 ms en E1 a 661 ms en E4, incremento atribuible al encolamiento asociado al conformado de tráfico y no a un cambio en la configuración de netem; la pérdida de paquetes medida sobre las 40 repeticiones fue de 0,07 %, consistente con el 0,1 % configurado. Estas mediciones son independientes del retardo de inicio reportado como métrica de QoE en la sección de resultados.

Tabla 2**Configuración de los escenarios de prueba**

Escenario	Representación	Bitrate de video	Ancho de banda	Latencia	Pérdida
E1	720p	2000 kbps	2,6 Mbps	400 ms	0,1 %
E2	480p	1000 kbps	1,4 Mbps	400 ms	0,1 %
E3	360p	500 kbps	0,8 Mbps	400 ms	0,1 %
E4	240p	250 kbps	0,5 Mbps	400 ms	0,1 %

Nota. El ancho de banda incorpora el flujo de audio de 128 kbps y el factor de reserva 1,2. Elaboración propia.

Procedimiento de ejecución y recolección

Antes de cada repetición se comprobaba el estado de Nginx, de la API y de los procesos de codificación y empaquetado. El cliente recibía los MPD desde Nginx, y dash.js arrancaba la reproducción con selección automática de calidad. Cada sesión se mantuvo corriendo durante 300 segundos y al terminar, el módulo de telemetría descargaba el archivo JSON con los datos recabados durante la transmisión.

El registro dejó constancia de la marca de tiempo, el bitrate seleccionado, la representación activa, los cambios de calidad, el retardo inicial y los eventos de interrupción. Por sesión se contabilizó el número de conmutaciones; esa cifra se normalizó con la relación cambios por minuto = cambios totales / 5, precisamente para no confundir el total acumulado de los cinco minutos con una tasa instantánea.

Se calculó la calidad integral utilizando la librería del modelo ITU-T P.1203.3, en modo 0. Se empearon cuatro entradas: audio AAC-LC a 128 kbps, la serie temporal de bitrate de video H.264 a 25 fps, el retardo de inicio y, en esta ejecución, cero eventos de stalling. El resultado (MOS O.46) se interpretó según la escala ACR: 5 es excelente, 4 bueno, 3 regular, 2 pobre y 1 inaceptable (Unión Internacional de Telecomunicaciones, 2017, 2020).

Procesamiento estadístico y criterios de interpretación

Cada sesión se configuró para 300 s y el módulo de datos recibió entre 302 y 305 muestras por sesión porque el temporizador de la sesión tardó en abrir y cerrar la exportación, y por eso el número total de registros es 12108 en comparación con los 12000 nominales. Los registros se emplearon únicamente para obtener las medias por sesión y las series temporales.

La unidad de análisis fueron las diez sesiones de cada escenario ($n = 10$). Para cada métrica se calcularon la media, la desviación estándar y el intervalo de confianza al 95 %, estimado con la distribución t de Student y nueve grados de libertad:

$$IC_{95\%} = \bar{x} \pm t_{0,975;9} \times \left(\frac{s}{\sqrt{10}} \right)$$

La dispersión entre ejecuciones midió la estabilidad entre repeticiones, algo distinto de la estabilidad del flujo, que se refleja en la frecuencia de cambios. El análisis solo se limitó a una estadística descriptiva, sin pruebas de contraste de hipótesis.

Los intervalos de confianza se interpretaron como indicadores de precisión de las medias obtenidas en las diez repeticiones. La lectura de la QoE se complementó con la configuración de red, las mediciones de control y los antecedentes sobre gestión de bitrate, estabilidad y prevención del rebuffering (Kalan & Dulger, 2024; Huu et al., 2024; Uddin et al., 2025).

RESULTADOS**Funcionamiento del prototipo y consistencia del experimento**

Las 40 sesiones se completaron sin una sola caída del servicio. Nginx entregó de forma continua el manifiesto dinámico, los archivos de inicialización y los segmentos .m4s, siempre con estado HTTP 200 y tipo application/dash+xml. La API mantuvo bajo control los procesos de FFmpeg y Shaka Packager en todo momento, y el cliente accedió sin inconvenientes a las cuatro representaciones descritas en el MPD. Además, no hubo eventos de stalling en las sesiones usadas para calcular el MOS.

Las diez repeticiones por escenario generaron entre 3022 y 3040 registros, lo que suma un total de 12108 observaciones. En E1, con media 4,62 con una desviación aproximada a 0,025, mientras que el intervalo fue [4,602; 4,638]. En E4, con una dispersión aproximada de 0,003, el margen del intervalo fue de $\pm 0,002$. Estos intervalos son reducidos en comparación con la escala de cinco puntos y demuestran que las medias fueron coherentes entre repeticiones.

Comportamiento de las métricas QoE

Tabla 3

Estadísticos descriptivos de las métricas QoE por escenario (media $\pm$ desviación estándar)

Escenario	Retardo de inicio (s)	Bitrate promedio (kbps)	Cambios por sesión	Cambios por minuto
E1	3,041 $\pm$ 0,535	1805,64 $\pm$ 240,45	41,1 $\pm$ 12,91	8,22 $\pm$ 2,59
E2	2,767 $\pm$ 0,304	1044,92 $\pm$ 180,03	22,4 $\pm$ 10,53	4,48 $\pm$ 2,11
E3	2,910 $\pm$ 0,414	769,63 $\pm$ 23,20	4,6 $\pm$ 3,13	0,92 $\pm$ 0,63
E4	4,427 $\pm$ 0,730	483,00 $\pm$ 7,19	2,3 $\pm$ 0,48	0,46 $\pm$ 0,10

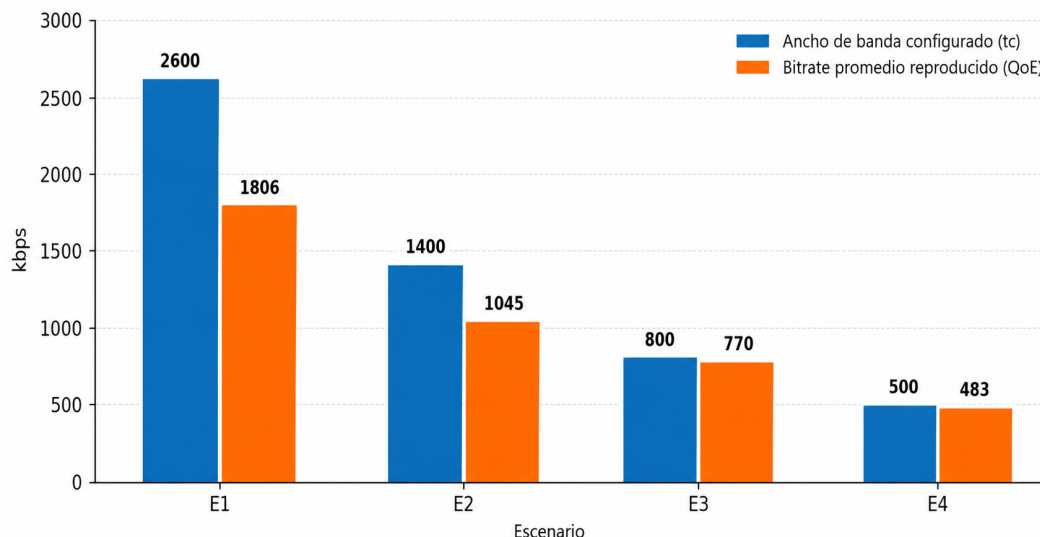
Nota. Los cambios por sesión corresponden a cada ensayo de cinco minutos. La última columna expresa la frecuencia normalizada. Elaboración propia.

El retardo de inicio se mantuvo similar en E1, E2 y E3: 3,041; 2,767 y 2,910 s, respectivamente. En E4 subió a 4,427 s, con la mayor desviación estándar de las cuatro pruebas. El incremento es razonable: con el escenario de 0,5 Mbps, el margen, durante la descarga inicial disminuye y esto extiende el tiempo requerido para generar el búfer. Aun así, el retardo se mantuvo por debajo de cinco segundos en promedio y no derivó en interrupciones posteriores.

El bitrate promedio descendió de manera escalonada con la restricción. E1 alcanzó 1805,64 kbps, E2 1044,92 kbps, E3 769,63 kbps y E4 483,00 kbps. Entre E1 y E4 la reducción fue de aproximadamente 73,3 %. La trayectoria temporal mostró una fase de incremento durante los primeros 30s y una posterior convergencia hacia niveles característicos de cada escenario. En E1 la dispersión fue mayor porque el algoritmo dispuso de margen para probar representaciones; en E4 el bitrate fue casi constante y su desviación se redujo a 7,19 kbps.

Figura 2

Comparación entre ancho de banda configurado y bitrate promedio reproducido



Nota. Elaboración propia.

Las conmutaciones de calidad repitieron el mismo patrón que el resto de las métricas: en E1 hubo 41,1 cambios por sesión, es decir, 8,22 por minuto; en E2 la cifra cayó a 22,4 y 4,48; en E3, a 4,6 y 0,92, y en E4, a apenas 2,3 y 0,46. Un número alto no significa mejor adaptación; solo indica más actividad del algoritmo, y esa actividad puede impactar la estabilidad visual. Bajo las circunstancias más restrictivas, el reproductor se mantuvo en la misma representación: la variabilidad disminuyó, la experiencia fue más consistente y la resolución fue más baja.

Al comparar entre repeticiones, E1 y E4 están bien separados: el primer grupo tiene alto bitrate, con retrasos de 2,4 a 4,0 segundos, mientras que el segundo grupo tenía 483 kbps y retrasos de 4,1 a 5,1 segundos. Dentro de cada condición las pequeñas variaciones del retardo no guardaron relación sistemática con el bitrate medio. Lo que marcó la diferencia fue el nivel de ancho de banda impuesto, no el comportamiento puntual de cada repetición.

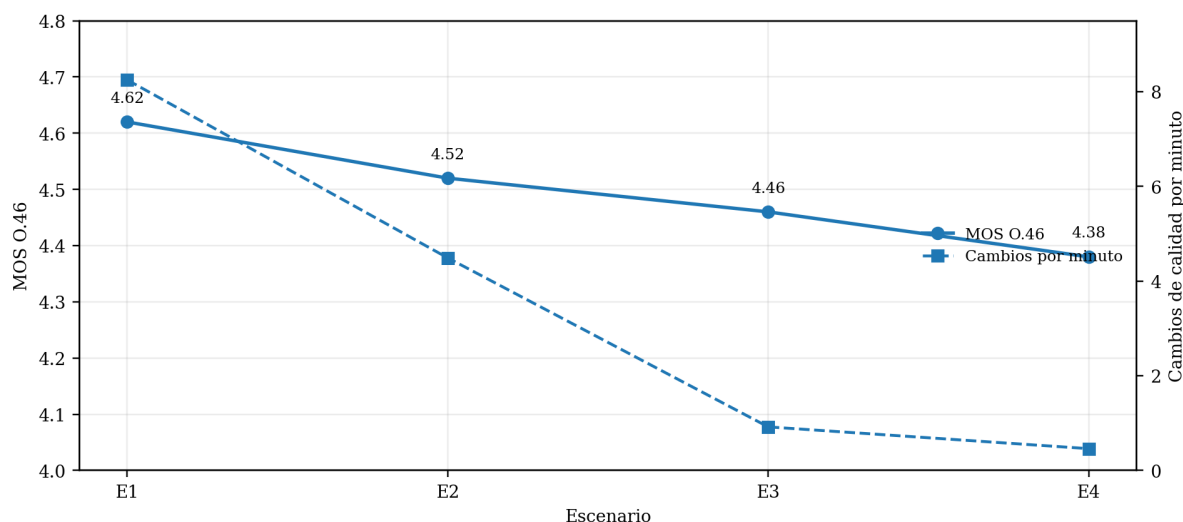
Relación entre capacidad disponible, bitrate y MOS

El bitrate reproducido se mantuvo siempre por debajo del límite configurado en cada escenario, con valores que se reducen conforme disminuye la capacidad. En E1, entre el límite de 2600 kbps y el promedio reproducido hubo una diferencia cercana a 794 kbps; en E4, entre 500 kbps y 483,00 kbps, la diferencia se redujo a 17 kbps. La escala de bitrate reproducida según el límite establecido en cada caso, así como la convergencia de E4 a solo 17 kbps desde el máximo configurado y una desviación de 7,19 kbps entre repeticiones, muestran que la restricción de tc funcionó bien y de manera reproducible en la transmisión de video. Este margen decreciente indica que el algoritmo es cauteloso cuando se le da la opción entre diferentes representaciones y demasiado cauteloso si solo se mantiene la representación mínima.

La puntuación MOS integró el efecto de la calidad audiovisual, el retardo inicial y la ausencia de stalling. Los resultados se presentan en la Tabla 4. Las cuatro medias superaron 4,0, por lo que se ubicaron en la categoría buena o superior. La diferencia total entre E1 y E4 fue de 0,24 puntos, equivalente a una reducción aproximada del 5,2 %, muy inferior a la caída de bitrate.

Figura 3

Evolución de la puntuación MOS O.46 y de las conmutaciones de calidad



Nota. Elaboración propia.

Tabla 4

Puntuación MOS O.46 y relación con la capacidad de red

Escenario	BW configurado	Bitrate reproducido	Retardo de inicio	MOS O.46	Desv. estándar
E1	2600 kbps	1805,64 kbps	3,041 s	4,62	0,025
E2	1400 kbps	1044,92 kbps	2,767 s	4,52	0,030
E3	800 kbps	769,63 kbps	2,910 s	4,46	0,005
E4	500 kbps	483,00 kbps	4,427 s	4,38	0,003

Nota. MOS calculado con ITU-T P.1203.3, modo 0. Todos los escenarios se ubicaron en la categoría buena de la escala ACR. Elaboración propia.

Las desviaciones MOS bajas de E3 y E4 se deben a su extrema estabilidad, el cliente se mantuvo en una representación casi constante durante toda la sesión, por consiguiente, el modelo recibió secuencias de bitrate prácticamente idénticas. En E1 y E2 sucedió lo opuesto pues el reproductor exploró distintas calidades, y esa exploración resultó en una dispersión algo mayor, aunque siempre limitada. Lo que parece atenuar más el impacto en la percepción de la disminución de calidad es la continuidad, es decir, la estabilidad del flujo, por encima de la resolución misma.

DISCUSIÓN

Los hallazgos confirman que el cliente ajusta la calidad a la capacidad estimada, priorizando la continuidad por encima del uso máximo del enlace. El bitrate bajó de forma progresiva y coherente con los límites impuestos. El reproductor, sin embargo, no agotó la capacidad disponible en ningún escenario, algo que va en línea con lo reportado por la literatura sobre algoritmos ABR: “*son, ante todo, mecanismos que negocian entre calidad, estabilidad y riesgo de rebuffering*” (Bentaleb et al., 2018; Kalan & Dulger, 2024).

En E1, el margen entre throughput y bitrate permitió alternar entre representaciones. El promedio de 8,22 cambios por minuto evidencia una adaptación activa, pero también una menor estabilidad visual. En E3 y E4, la frecuencia cayó por debajo de un cambio por minuto y las desviaciones del bitrate fueron pequeñas. Este resultado es compatible con los estudios que señalan que, ante restricciones severas, los clientes tienden a adoptar una estrategia conservadora y a mantenerse en un nivel sostenible (Huu et al., 2024; Nguyen Viet et al., 2022).

La interpretación de las conmutaciones requiere distinguir adaptación y calidad de experiencia. Un algoritmo que cambia constantemente puede responder rápido a la red, pero también introducir oscilaciones visibles. Por esta razón, las investigaciones actuales no buscan maximizar únicamente el bitrate; también consideran la amplitud de los cambios, el nivel del búfer y el tiempo sin interrupciones (Timmerer et al., 2025; Uddin et al., 2025). La disminución de conmutaciones en E3 y E4 en el prototipo ayudó a que la experiencia fuera uniforme y a que los MOS se mantuvieran por encima de 4,0.

El aumento del retardo de inicio en E4 fue el efecto negativo más claro de la restricción. La media de 4,427 s indica que el cliente necesitó más tiempo para iniciar con un enlace cercano al mínimo de servicio. Los valores de E1-E3, en cambio, permanecieron alrededor de tres segundos. La literatura identifica el retardo inicial y el rebuffering como factores críticos de abandono; por tanto, un despliegue real debería reservar capacidad adicional o ajustar el objetivo de búfer para evitar que nuevas cargas concurrentes eleven este valor (Naik, 2025; Pant et al., 2025).

Uno de los resultados más relevantes es justo esa asimetría entre bitrate y MOS. La capacidad cayó de 2,6 a 0,5 Mbps, el bitrate promedio bajó 73,3 %, pero la puntuación MOS 0.46 apenas se movió: solo 5,2 %. Esto tiene una explicación técnica. El modelo P.1203 no juzga el bitrate por sí solo, sino que combina calidad audiovisual, cambios temporales y eventos de reproducción; en ausencia de stalling y con estabilidad bajo las condiciones restrictivas, el deterioro global quedó contenido (International Telecommunication Union, 2017; Robitza et al., 2020).

En el plano técnico, el estudio deja algo claro: una cadena de software libre alcanza para cubrir las funciones esenciales de un servicio institucional, desde la codificación multirrepresentación y el empaquetado MPEG-DASH hasta la publicación HTTP, el control mediante API, la reproducción adaptativa y la recolección de telemetría. Los componentes están desacoplados; eso permite sustituirlos o actualizarlos sin rediseñar toda la arquitectura. Y trabajar con HTTP/1.1 y Nginx tiene otra ventaja: no ata el sistema a protocolos ni servidores propietarios.

Los resultados se sostienen en la coherencia del banco de pruebas. El bitrate reproducido se adaptó de manera escalonada a cada límite configurado y, en la situación más restrictiva, se acercó a 17 kbps del máximo con una dispersión mínima entre repeticiones, el retardo medido en el canal, superior al valor nominal por efecto del encolamiento, confirma asimismo que la disciplina de cola operaba durante las sesiones. Estas observaciones señalan que tc/netem creó condiciones que se pueden repetir, de acuerdo con los principios de las pruebas de throughput y el empleo de shaping para simular escenarios determinados (Constantine et al., 2011; Raussi et al., 2023).

Estos resultados ofrecen a las instituciones educativas un primer parámetro de: 0,5 Mbps por sesión bastaron para sostener la representación de 240p y llegar a un MOS de 4,38, siempre que no hubiera tráfico concurrente. Pero este valor no implica una recomendación de producción ya que no considera solicitudes simultáneas, tráfico de fondo o fluctuaciones dinámicas de la red. La metodología en sí es muy relevante para cómo una institución puede analizar su catálogo de representaciones y calcular márgenes operativos para ofrecer servicios.

La principal contribución radica en el desarrollo de un entorno reproducible para redes con recursos limitados. La literatura se enfoca en las mejoras algorítmicas que se evalúan en infraestructuras con mayor capacidad de procesamiento y conectividad. Los datos sobre bitrate, retraso, conmutación y MOS están disponibles para comparar futuras versiones del algoritmo, diferentes duraciones de segmentos o esquemas de gestión de colas; y hay que decirlo, el trabajo también contribuye al objetivo 9.5 del ODS 9 al fortalecer las capacidades de investigación y desarrollo tecnológico en infraestructura digital.

El estudio tiene sus límites, pues se trabajó con un solo cliente y una conexión de cable directo, por lo que no hubo una forma de evaluar la escalabilidad o la competencia por el enlace. La latencia se fijó en 400 ms y la pérdida en 0,1 % durante todo el experimento lo que dejó fuera cualquier estimación sobre cómo interactúan esas variables entre sí. Las redes se mantuvieron estáticas cuando las redes reales no funcionan así estas fluctúan. Y se usó un único algoritmo ABR con un conjunto fijo de representaciones, y el MOS se estimó con un modelo paramétrico sin contraste frente a una prueba subjetiva.

CONCLUSIONES

El entorno experimental permitió caracterizar, de forma controlada, el comportamiento del prototipo ante distintos niveles de ancho de banda. Los cuatro escenarios se ejecutaron con latencia de 400 ms y pérdida de 0,1 %, diez repeticiones de cinco minutos cada una. El bitrate reproducido se ajustó de forma escalonada a los límites configurados en cada escenario, lo que confirma que las condiciones eran reproducibles y que los cambios observados respondieron, sobre todo, a la capacidad disponible.

Como parte de la arquitectura implementada se integró FFmpeg, Shaka Packager, Nginx, Flask y dash.js sobre una LAN aislada. El sistema generó y distribuyó cuatro representaciones MPEG-DASH; mantuvo los procesos activos a lo largo de las 40 sesiones y no registró ni interrupciones de servicio ni eventos de stalling en las pruebas analizadas. Construir un prototipo funcional con herramientas de código abierto es, entonces, técnicamente viable.

El algoritmo ABR ajustó el bitrate según la restricción de cada escenario: 1805,64 kbps en E1, 1044,92 en E2, 769,63 en E3 y 483,00 en E4. Los cambios disminuyeron de 8,22 a 0,46 cambios por minuto, indicando un cambio que pasó de buscar muchas representaciones a una elección firme. El peor efecto fue el retraso de inicio, que aumentó a 4,427 s en E4.

La calidad de la experiencia seguía siendo aceptable en los cuatro casos. Las puntuaciones MOS 0.46 oscilaron entre 4,38 y 4,62, lo cual es bueno en la escala ACR. Esa caída en MOS fue mucho menos pronunciada que la del bitrate, y había una razón muy simple: la reproducción no se ralentizó y se mantuvo estable en representaciones sostenibles. Al final, la calidad de la experiencia fue más importante que cualquier otro factor.

El prototipo y el procedimiento pueden usarse como base para nuevas investigaciones y para futuras experimentaciones institucionales. Pero estos son resultados preliminares; no prueban la escalabilidad en producción. Los valores obtenidos describen el comportamiento de una sesión única sobre un enlace controlado y no deben leerse como parámetros de dimensionamiento.

RECOMENDACIONES Y LÍNEAS FUTURAS

La siguiente etapa de evaluación debe ampliar el banco de pruebas hacia condiciones dinámicas y hacia el uso concurrente. Cada sesión mantuvo aquí un nivel de ancho de banda constante, decisión que permitió aislar la variable, pero que no reproduce las fluctuaciones de una red institucional; se recomienda aplicar perfiles con caídas, recuperaciones y ráfagas de capacidad para determinar con qué velocidad el algoritmo detecta el cambio, reduce la calidad y vuelve a una representación superior sin oscilaciones innecesarias. Simultáneamente, agregar varios clientes requiere de una prueba de escalabilidad progresiva, ósea agregar sucesivamente reproductores hasta que el rebuffering, el MOS o el retardo inicial superen el umbral establecido, debido a que las conexiones adicionales comparten recursos o compiten por la conexión.

En lo que respecta al algoritmo es recomendable comparar el comportamiento determinado del algoritmo ABR del reproductorred in dash.js con estrategias basadas en QoS y en modelos híbridos o de aprendizaje por

refuerzo sobre los mismos videos, representaciones y perfiles de red para que las diferencias no se atribuyan solo al contenido. La evaluación no debería apoyarse solo en el MOS, sino también debería analizarse el tiempo en cada representación, la amplitud de las conmutaciones y la frecuencia con que se vacía el búfer, para que aporten información que el promedio oculta. La duración de cada segmento también conviene analizarla desde otra perspectiva que permitan fijar el equilibrio entre velocidad de reacción y sobrecarga de solicitudes.

Finalmente, la validación del prototipo en una red institucional real, con Wi-Fi, tráfico académico, diferentes dispositivos y contenidos de complejidad visual variable, es lo que posibilitaría determinar hasta qué punto se pueden generalizar estos descubrimientos. Para no perder la capacidad de reproducir el estudio, se debe conservar la trazabilidad de las configuraciones y a su vez recopilar los datos promedio de un panel que monitoree en tiempo real y cruce las métricas QoS del servidor con la telemetría QoE de cada cliente, así como con alertas ante niveles críticos.

IMPLICACIONES PARA EL DESPLIEGUE INSTITUCIONAL

Estos resultados permiten ver una forma de implementar servicios de streaming en instituciones que necesiten distribuir clases, eventos o repositorios audiovisuales dentro de su propia red. No se trata solo de comparar la infraestructura, sino también de definir el contenido y los usuarios esperados, para identificar qué resoluciones ofrecen un valor real, qué bitrate de audio, cuánto dura el segmento y cuántas sesiones simultáneas van a tener lugar. La capacidad total se calcula por lo tanto sobre esa base, que, como siempre, es mayor que la demanda nominal porque las solicitudes de manifiesto, los encabezados HTTP, las retransmisiones TCP y los picos de descarga añaden un consumo no contabilizado por la suma de las tasas de bits.

Esta arquitectura facilita una implementación por fases gracias a Nginx, FFmpeg y Shaka Packager que operan como servicios separados y la API centraliza el control, lo que permite a una institución empezar a generar contenido. La autonomía tecnológica no significa dejar de lado las plataformas externas, pues un servidor local resulta apropiado para contenidos internos, materiales de acceso restringido o transmisiones que conviene no enviar por el enlace de Internet. El valor del prototipo está en que es una alternativa que otorga control sobre el streaming a las instituciones, conservando el catálogo, las políticas de retención y el ajuste de las representaciones a su propia red.

Es recomendable establecer criterios operativos sobre las mismas métricas que se reportan en este estudio antes de iniciar la producción, determinando un valor límite aceptable para cada una según el contexto de uso. En su conjunto, estos elementos constituyen un protocolo de aceptación que se repite cada vez que la red, el hardware o la codificación cambian. Esto hace que la evaluación deje de ser una única prueba y empiece a funcionar como control constante de la calidad del servicio. La solución presenta más componentes como lo son el configurar la cadena de codificación, analizar el manifiesto, gestionar procesos en Linux y leer en conjunto QoS y QoE que entran en distintas áreas de la tecnología.

FINANCIACIÓN

El estudio no contó con financiamiento externo. Se desarrolló con recursos propios en el marco de un trabajo de titulación de la Escuela Superior Politécnica de Chimborazo, Sede Morona Santiago.

CONFLICTO DE INTERESES

Los autores declaran no tener conflictos de interés relacionados con esta investigación.

CONTRIBUCIÓN DE AUTORÍA

Kevin Junior Chocho Quizhpi: Metodología, implementación del prototipo, investigación, análisis formal, procesamiento de datos, visualización y edición. Juan José Pazmiño Gordillo: Conceptualización, metodología, supervisión, validación, revisión crítica, edición y aprobación de la versión final. Lizbeth Rocío Ebla Yerovi: Validación, revisión crítica, edición y aprobación de la versión final.

REFERENCIAS

- Bentaleb, A., Taani, B., Begen, A. C., Timmerer, C., & Zimmermann, R. (2018). A survey on bitrate adaptation schemes for streaming media over HTTP. *IEEE Communications Surveys & Tutorials*, 21(1), 562-601. <https://doi.org/10.1109/COMST.2018.2862938>
- Collahuazo-Cuases, J. D. (2024). Integración de las tecnologías de la información y las comunicaciones en el sistema de educación pública ecuatoriano: Una revisión sistemática. *Polo del Conocimiento*, 9(10), 1716-1736. <https://doi.org/10.23857/pc.v9i10.8203>
- Constantine, B., Forget, G., Geib, R., & Schrage, R. (2011). Framework for TCP throughput testing (RFC 6349). Internet Engineering Task Force.
- Huu, T. V., Pham, V. S., Huong, T. N. T., & Le, H.-C. (2024). QoE aware video streaming scheme utilizing GRU-based bandwidth prediction and adaptive bitrate selection for heterogeneous mobile networks. *IEEE Access*, 12, 45785-45795. <https://doi.org/10.1109/ACCESS.2024.3382155>
- International Telecommunication Union. (2017). Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport (Recommendation ITU-T P.1203).
- International Telecommunication Union. (2020). Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport - Quality integration module (Recommendation ITU-T P.1203.3, Amendment 1).
- International Telecommunication Union. (2024). Advanced video coding for generic audiovisual services (Recommendation ITU-T H.264, V15).
- Iren, E., & Kantarci, A. (2022). Content aware video streaming with MPEG-DASH technology. *TEM Journal*, 11(2), 763-771.
- Jácome Paredes, D. A., & Naranjo Pachacama, J. P. (2022). Diseño de una red para el uso de aplicaciones de video mediante redes definidas por software (SDN) [Trabajo de titulación, Universidad Politécnica Salesiana].
- Kalan, R., & Dulger, I. (2024). A survey on QoE management schemes for HTTP adaptive video streaming: Challenges, solutions, and opportunities. *IEEE Access*, 12, 170803-170817. <https://doi.org/10.1109/ACCESS.2024.3491613>
- Kang, J., & Chung, K. (2022). HTTP adaptive streaming framework with online reinforcement learning. *Applied Sciences*, 12(15), 7423. <https://doi.org/10.3390/app12157423>
- Lorenzi, D., Tashtarian, F., Hellwagner, H., & Timmerer, C. (2024). MEDUSA: A dynamic codec switching approach in HTTP adaptive streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 20(10), Article 319. <https://doi.org/10.1145/3656175>
- Mueller, C., Munteanu, A., Timmerer, C., & Hellwagner, H. (2024). Content-adaptive video coding for HTTP adaptive streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(9), 5790-5803. <https://doi.org/10.1109/TCSVT.2024.3387164>
- Naik, H. M. (2025). Quantifying multimedia streaming quality: A practical analysis using PIE and Flow Queue PIE (arXiv:2511.17525). *arXiv*.
- Nguyen Viet, H., Trinh Dac, C., Pham Ngoc, N., & Truong Thu, H. (2022). Flexible HTTP-based video adaptive streaming for good QoE during sudden bandwidth drops. *TechRxiv*. <https://doi.org/10.36227/techrxiv.21755879.v2>
- Pant, P. K., Joshi, M. C., & Tiwari, N. (2025). User quality of experience (QoE) in IEEE 802.11 WLAN: A comprehensive review. *International Journal of Latest Technology in Engineering, Management & Applied Science*, 14(8), 692-700. <https://doi.org/10.51583/IJLTEMAS.2025.1408000086>
- Peroni, L., & Gorinsky, S. (2025). An end-to-end pipeline perspective on video streaming in best-effort networks: A survey and tutorial. *ACM Computing Surveys*, 57(12). <https://doi.org/10.1145/3742472>
- Raussi, P., Kokkonen, H., Ahola, K., Heikkinen, A., & Uitto, M. (2023). Prioritizing protection communication in a 5G slice: Evaluating HTB traffic shaping and UL bitrate adaptation for enhanced reliability. *The Journal of Engineering*, 2023, e12309. <https://doi.org/10.1049/tje2.12309>
- Robitza, W., Ramachandra Rao, R. R., Göring, S., & Raake, A. (2020). ITU-T standardized bitstream-based video quality models (Technical Report). Technische Universität Ilmenau.

- Santillán Lima, J. C., Llanga Vargas, A., Mayorga, W., & Valdiviezo, S. (2016). Evaluación del requerimiento tecnológico para servicios educativos, propuesta de infraestructura de red inalámbrica del Campus La Dolorosa de la Universidad Nacional de Chimborazo. En *II Congreso Internacional Educación Contemporánea, Calidad Educativa y Buen Vivir*.
- Santillán-Lima, J. C., Rocha-Jacome, C., Guerrero-Morejón, K., Llanga-Vargas, A., Váscquez-Barrera, F., & Molina-Granja, F. (2017). El impacto de los servicios de telecomunicaciones y las TIC en las necesidades de la educación superior. En *IV Congreso Internacional de Ciencia, Tecnología, Innovación y Emprendimiento (CITE)*.
- Tenezaca, K. F. C., & Paguay, N. O. C. (2023). Desarrollo e implementación de un sistema web de streaming de video bajo demanda para la empresa ADS Publicidad [Trabajo de titulación, Escuela Superior Politécnica de Chimborazo].
- Timmerer, C., Amirpour, H., Tashtarian, F., Afzal, S., Rizk, A., Zink, M., & Hellwagner, H. (2025). HTTP adaptive streaming: A review on current advances and future challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 21(7), Article 198. <https://doi.org/10.1145/3736306>
- Uddin, S., Grega, M., Leszczuk, M., & Rahman, W. (2025). Evaluating HAS and low-latency streaming algorithms for enhanced QoE. *Electronics*, 14(13), 2587. <https://doi.org/10.3390/electronics14132587>
- Zambrano, E. R. M. (2025). Mejoramiento de la infraestructura TI en instituciones educativas: Impacto en las competencias académicas de estudiantes. *Revista Ingenio Global*, 4(1), 237-249. <https://doi.org/10.62943/rig.v4n1.2025.227>