

Universidad Autónoma del Estado de Hidalgo

Instituto de Ciencias Básicas e Ingeniería

Licenciatura en Ciencias Computacionales

**Prototipo de transmisión de video a través de internet
por medio del protocolo HLS**

Tesis presentada para la obtención del grado de

Licenciado en Ciencias Computacionales

Por

Ivan Felipe Campos Esparza

Director de tesis:

M.T.I Vanessa Esther Ureña Medina

Diciembre 2025

V.º B.º Para presentar
ante comité evaluador



Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería
School of Engineering and Basic Sciences

Mineral de la Reforma, Hgo., a 6 de enero de 2026

Número de control: ICBI-D/005/2026
Asunto: Autorización de impresión.

MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH

Con Título Quinto, Capítulo II, Capítulo V, Artículo 51 Fracción IX del Estatuto General de nuestra Institución, por este medio, le comunico que el Jurado asignado al egresado de la Licenciatura en Ciencias Computacionales **Iván Felipe Campos Esparza**, quien presenta el trabajo de titulación "**Prototipo de transmisión de video a través de Internet por medio del protocolo HLS**", ha decidido, después de revisar fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; dicho trabajo en la reunión de sinodales, **autorizar la impresión del mismo**, una vez realizadas las correcciones acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

Presidente: M.C.C. Gonzalo Alberto Torres Samperio

Secretario: M. en C. Isaías Pérez Pérez

Vocal: M.T.I. Vanessa Esther Ureña Medina

Suplente: M. en A. Víctor Islas Reyes

Sin otro particular por el momento, reciba un cordial saludo.

Atentamente
"Amor, Orden y Progreso"

Mtro. Gabriel Vergara Rodríguez
Director del ICBI

GVR/MMM

Ciudad del Conocimiento, Carretera Pachuca-Tulancingo Km. 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México. C.P. 42184
Teléfono: 771 71 720 00 Ext. 40001
direccion_icbi@uaeh.edu.mx, vergara@uaeh.edu.mx

"Amor, Orden y Progreso"



2025



uaeh.edu.mx

*“En el viaje de la vida no hay caminos llanos;
todos son subidas o bajadas”*

— Arturo Graf

Agradecimientos

A mi mamá por cuidarme y apoyarme siempre, por todo el amor que siempre me ha dado, que con su inteligencia me ha dado enseñanzas y consejos que han sido para mejorar.

A mi papá que siempre nos hacía reír con sus ocurrencias, nos asombraba con su talento musical, el amor al arte y que me brindó lo necesario para seguir estudiando.

A mi abuelita que con su experiencia me aconsejó para tener una vida digna contando sus increíbles historias de juventud y por todo el amor y apoyo que nos tuvo.

A mis hermanas que desde pequeño siempre me apoyaron y estuvieron cuando las necesitaba, por los juegos y aprendizajes que vivimos juntos.

A mi pareja por estar conmigo, por su cariño y el apoyo en cada decisión que hemos tomado juntos.

A mis tíos/as que han estado presentes en mi vida desde que era pequeño que me han apoyado con su cariño y atenciones.

A mi perro pudín por recibirme con cariño cuando voy de visita, por cuidar la casa y ser un fiel compañero.

A mis amistades con las que he pasado buenos momentos y que a pesar del tiempo y la distancia seguimos estando ahí.

A mis profesores y al personal administrativo de la universidad, quienes me brindaron la oportunidad y el apoyo necesario para desarrollar mi formación profesional.

Gracias por todo, lo que he logrado es gracias a ustedes.

Resumen

En este proyecto se trabajará con servidores, codecs, gestores de video y protocolos de transmisión de video para poder hacer el envío sobre internet. Usando principalmente Ubuntu Server como sistema operativo haciendo uso de los procesos en segundo plano y herramientas dedicadas para la gestión de flujo de video tal como lo es FFmpeg y el cómo hacer las configuraciones necesarias para poder transmitirlo.

Esto con el fin de entender y poder decidir cómo se aplica la tecnología dependiendo el contexto en el que se vaya a trabajar.

Esto es de suma importancia ya que hay varias formas de hacerlo, sin embargo, dependiendo el entorno de trabajo se puede optar por una que además de cumplir las necesidades, se adapte mejor a la infraestructura interna.

A su vez se abordan temas relevantes a tener en cuenta al momento de querer transmitir flujos de video sobre internet, desde los detalles más simples y curiosos hasta los más laboriosos y técnicos.

Se espera que al término de esta tesis el lector pueda aplicar de manera práctica el uso de las herramientas y poder comprender y decidir por sí mismo qué alternativa se adecua más a lo que requiere o simplemente hacer la prueba para comprender el funcionamiento de los flujos de video que viajan a través del mundo, siendo esta la principal forma o medio de comunicación que en la actualidad se usa para informarse.

Abstract

This project will involve working with servers, codecs, video managers, and video transmission protocols to enable transmission over the internet. We will mainly use Ubuntu Server as the operating system, leveraging background processes and dedicated tools for video stream management, such as FFmpeg, and configuring the necessary settings to transmit it.

The goal is to understand and decide how to apply the technology depending on the context in which it will be used.

This is extremely important, as there are several ways to do this. However, depending on the work environment, it is possible to choose one that not only meets the needs but also best suits the internal infrastructure.

In turn, relevant issues to consider when transmitting video streams over the internet are addressed, from the simplest and most curious details to the most laborious and technical ones.

It is hoped that by the end of this thesis, the reader will be able to apply the use of the tools practically and be able to understand and decide for themselves which alternative best suits their needs, or simply test it out to understand how video streams travel around the world, this being the main form or means of communication currently used to obtain information.

Índice General

Agradecimientos	6
Resumen.....	8
Abstract.....	9
Índice General	10
Índice de tablas, cuadros y/o figuras	12
Capítulo I. Introducción.....	16
1.1 Introducción	16
1.2 Antecedentes	18
1.3 Problemática	22
1.4 Justificación	23
1.5 Objetivo general.....	24
1.6 Objetivos específicos	24
1.7 Alcances y limitaciones	25
1.8 Marco Teórico.....	26
1.8.1 Elementos del proceso de comunicación de video digital.....	26
1.8.2 Elementos de redes ópticas de alta velocidad	28
1.8.3 Tipos de transmisión y servicios	30
1.8.4 Equipos de transmisión y recepción.....	32
1.8.5 Equipos de redes	33
1.8.6 Equipos de infraestructura	34
1.8.7 Organismos y regulaciones	35
1.8.8 Estándares y tecnologías de transmisión.....	36
1.8.9 Tecnologías y herramientas de procesamiento de video	37
1.9 Estado del arte.....	39
Capítulo II. Metodología	43
2.1 Uso y Justificación de la Metodología	43
2.2 Aplicación de la Metodología	45
2.2.1 Metodología de prototipos	45
2.2.2 Declaración de Uso Académico y Exclusión de Responsabilidad	104
Capítulo III. Resultados	105
Resultados	105

Conclusiones y trabajos futuros	112
Referencias.....	113

Índice de tablas, cuadros y/o figuras

Fig. 1 Diagrama de difusión de video tradicional.....	20
Fig. 2 Representación de consumo de video.....	21
Fig. 3 Elementos de red óptica.....	28
Fig. 4 Diagrama de transmisión.....	30
Fig. 5 <i>Diagrama de infraestructura de recepción de canales.</i>	32
Fig. 6 Infraestructura en rack	34
Fig. 7 Antena.....	36
Fig. 8 <i>Diagrama de componentes de un segmento HLS.</i>	42
Fig. 9 Fases del modelo de prototipado.	45
Fig. 10 Diagrama de infraestructura de procesamiento y transmisión de video.	46
Fig. 11 Página de descarga oficial de FFmpeg en los resultados de búsqueda de Google.....	47
Fig. 12 <i>Sección de descarga de paquetes y archivos ejecutables en la página oficial de FFmpeg.</i>	48
Fig. 13 Link de descarga de proveedor de software gyan.dev	48
Fig. 14 Link de descarga de FFmpeg comprimido en la página de gyan.dev	48
Fig. 15 Archivo comprimido descargado de la herramienta FFmpeg.....	49
Fig. 16 Estructura de carpetas de la herramienta FFmpeg para windows.....	49
Fig. 17 Variables de entorno agregadas al Path de Windows.	50
Fig. 18 Ejecución de FFmpeg mediante terminal de Windows.	50
Fig. 19 Página de ABC News en los resultados de búsqueda de Google.	51
Fig. 20 Herramientas para desarrolladores de Chrome en apartado Network.	51
Fig. 21 Herramientas para desarrolladores de Chrome en apartado Network.	52
Fig. 22 Comando de reproducción de FFmpeg con link de archivo M3U8 como entrada.	52
Fig. 23 Comando de reproducción de FFmpeg con link de archivo M3U8 como entrada.	53
Fig. 24 Comando para generación de lista M3U8 con link de archivo M3U8 como entrada.....	53
Fig. 25 Generación de lista M3U8 con link de archivo M3U8 como entrada.	54
Fig. 26 Generación de lista M3U8 creada.....	54
Fig. 27 Reproducción de lista M3U8 generada.....	55
Fig. 28 Archivo de drivers de NVIDIA para tarjeta gráfica.	59
Fig. 29 Archivo de drivers de NVIDIA para tarjeta gráfica dentro de una carpeta.	60
Fig. 30 Ejecución de comando con el protocolo SCP para envío de archivo.	60
Fig. 31 Ruta donde se encuentra el archivo recibido en Linux.	60
Fig. 32 Pantalla de selección del tipo de módulo del kernel durante la instalación de los drivers NVIDIA.	61
Fig. 33 Advertencia sobre un método alternativo de instalación de controladores NVIDIA detectado.	61
Fig. 34 Advertencia sobre la presencia del controlador libre “Nouveau” durante la instalación del driver NVIDIA.	62
Fig. 35 Mensaje del instalador de NVIDIA solicitando permiso para crear automáticamente los archivos de configuración que desactivan el controlador “Nouveau”.	62
Fig. 36 Aviso del instalador de NVIDIA indicando la necesidad de reiniciar el sistema para aplicar la desactivación del controlador “Nouveau”.	63
Fig. 37 Advertencia del instalador de NVIDIA sobre la imposibilidad de realizar verificaciones de compatibilidad (“sanity checks”).	63

Fig. 38 Proceso de construcción de los módulos del kernel.	64
Fig. 39 Advertencia del instalador de NVIDIA sobre la detección automática de las rutas de bibliotecas del servidor gráfico X.	64
Fig. 40 Advertencia del instalador de NVIDIA sobre la falta de una ruta adecuada para instalar las bibliotecas de compatibilidad de 32 bits.	65
Fig. 41 Mensaje del instalador de NVIDIA ofreciendo registrar el módulo del kernel con DKMS para recompilación automática.	65
Fig. 42 Progreso de registro de los módulos del kernel con DKMS.	66
Fig. 43 Advertencia del instalador de NVIDIA al no detectar la ruta de instalación de las librerías EGL de libglvnd.	66
Fig. 44 Mensaje del instalador de NVIDIA solicitando la reconstrucción del initramfs tras deshabilitar el controlador Nouveau.	67
Fig. 45 Proceso de reconstrucción de initramfs.	67
Fig. 46 Mensaje del instalador de NVIDIA ofreciendo ejecutar la utilidad NVIDIA-xconfig para actualizar automáticamente el archivo de configuración del servidor gráfico Xorg.	68
Fig. 47 Mensaje final del instalador de NVIDIA confirmando la correcta actualización de la configuración de Xorg y la finalización del proceso de instalación.	68
Fig. 48 Mensaje final del instalador de NVIDIA recomendando reiniciar el sistema para aplicar los cambios del controlador gráfico.	69
Fig. 49 Salida del comando NVIDIA-smi mostrando el estado de la GPU y confirmando el funcionamiento del controlador propietario de NVIDIA.	70
Fig. 50 Información de capacidad de codificación y decodificación de la gráfica utilizada en el ejemplo.	71
Fig. 51 Sección de tarjetas Quadro de NVIDIA.	71
Fig. 52 Información de capacidad de codificación y decodificación de una gráfica Quadro NVIDIA RTX A4000.	71
Fig. 53 Configuración de archivo yaml de interfaces de red.	72
Fig. 54 Configuración de archivo “smcroute.conf” para enrutar tráfico multicast sobre una interfaz.	73
Fig. 55 Tráfico sobre interfaces de red.	73
Fig. 56 Prueba de recepción de señal de video.	74
Fig. 57 Metadatos obtenidos de la recepción de señal de video.	74
Fig. 58 Configuración de interfaz de salida del tráfico multicast procesado.	75
Fig. 59 Creación de servicio de procesamiento de video con systemd y FFmpeg.	76
Fig. 60 Servicios creados para transcodificación de video.	76
Fig. 61 Estado de los servicios creados luego de ejecutarse.	77
Fig. 62 Configuración de conexión ssh bloqueando el acceso root.	78
Fig. 63 Activación del firewall.	78
Fig. 64 Agregación de regla en firewall para permitir conexión sobre el puerto 22.	78
Fig. 65 Configuración de recepción de tráfico multicast sobre interfaz de red.	79
Fig. 66 Comando para probar recepción de señal procesada.	80
Fig. 67 Metadatos de la señal procesada recibida.	80
Fig. 68 Estado de servidor Nginx.	81
Fig. 69 Modificación del archivo de configuración de Nginx.	82
Fig. 70 Regla añadida al firewall para permitir conexión de Nginx.	82
Fig. 71 Ruta y carpetas donde se guardan los fragmentos de la lista M3U8.	83
Fig. 72 Creación de archivos que contienen los comandos para generar las listas M3U8.	83

Fig. 73 Ejemplo de servicio recibiendo la señal procesada y generando la lista M3U8.....	83
Fig. 74 Estado de los servicios generados para creación de listas M3U8.....	84
Fig. 75 Archivos generados en las carpetas dentro de la ruta que sirve Nginx.....	85
Fig. 76 Visualización de contenido de lista M3U8.....	85
Fig. 77 Ventana “Abrir medio” del reproductor VLC mostrando la entrada de lista M3U8 apuntando a la IP del servidor con Nginx ya sea pública o local.....	87
Fig. 78 Reproducción del canal Foro TV en VLC Media Player, mostrando la emisión en vivo dentro del reproductor.....	87
Fig. 79 Interfaz principal del programa Raspberry Pi Imager mostrando las opciones de selección de dispositivo, sistema operativo y almacenamiento.....	89
Fig. 80 Pantalla de “Bienvenido a LibreELEC”.....	89
Fig. 81 Selección de add-ons de LibreELEC.....	90
Fig. 82 Configuración de una regla de firewall tipo Destination NAT en MikroTik RouterOS para redirigir tráfico entrante desde la IP pública a un dispositivo local.....	91
Fig. 83 Configuración de la acción de la regla del firewall tipo Destination NAT en MikroTik RouterOS redirigiendo el tráfico a la IP del Raspberry con LibreELEC y Jellyfin.....	91
Fig. 84 Ejemplo de acceso a un servicio mediante la dirección IP pública y puerto desde la barra de direcciones del navegador Google Chrome.....	92
Fig. 85 Pantalla de bienvenida del servidor multimedia Jellyfin mostrando el asistente de configuración inicial.....	92
Fig. 86 Barra lateral del entorno principal de Jellyfin mostrando las secciones de navegación del servidor multimedia.....	93
Fig. 87 Pantalla del asistente de configuración inicial de Jellyfin para añadir bibliotecas.....	93
Fig. 88 Selección del tipo de contenido de Jellyfin para añadir biblioteca.....	94
Fig. 89 Selección de proveedores de metadatos para Jellyfin de películas (The Movie Database y The Open Movie Database).....	94
Fig. 90 Pantalla de Jellyfin mostrando la selección de la carpeta local que contiene los archivos de películas.....	95
Fig. 91 Botón de Jellyfin para escaneo de bibliotecas.....	95
Fig. 92 Sección de “Películas” en Jellyfin mostrando títulos disponibles en la biblioteca multimedia.....	96
Fig. 93 Botón de Jellyfin para identificar títulos y agregar metadatos del vídeo.....	96
Fig. 94 Pantalla de Jellyfin mostrando el resultado de la identificación de una película mediante los metadatos de The Movie Database (TMDb).....	97
Fig. 95 Pantalla de Jellyfin mostrando la vista de detalles de una película con metadatos obtenidos desde The Movie Database (TMDb).....	97
Fig. 96 Estructura de carpetas y nombres de archivo utilizada para la organización de series en Jellyfin.....	98
Fig. 97 Pantalla de Jellyfin mostrando la vista de detalles de una serie con metadatos obtenidos desde The Movie Database (TMDb).....	98
Fig. 98 Botón de Jellyfin para identificar títulos y agregar metadatos del vídeo.....	99
Fig. 99 Pantalla de configuración de Jellyfin para agregar un sintonizador M3U (M3U tuner).....	99
Fig. 100 Pantalla de selección de archivo que muestra la ruta del archivo M3U8 utilizado para configurar un sintonizador en Jellyfin.....	100
Fig. 101 Contenido de un archivo M3U8 utilizado para la configuración de un sintonizador IPTV.....	100
Fig. 102 Pantalla de VLC Media Player abriendo un archivo M3U8.....	101
Fig. 103 VLC Media Player reproduciendo el canal “TNT Series” desde la lista M3U8 generada.....	101

Fig. 104 Pantalla de la sección “Canales” del panel de administración de Jellyfin, donde se listan las fuentes de TV configuradas.	102
Fig. 105 Reproducción de un canal en Jellyfin desde una lista M3U8.	102
Fig. 106 Configuración de proveedor de guía (EPG) en Jellyfin.	102
Fig. 107 Menú lateral de Jellyfin con la sección de usuarios.	103
Fig. 108 Pantalla para agregar un nuevo usuario en Jellyfin.	103
Fig. 109 Gráfica de Satisfacción general usando la escala Likert.	106
Fig. 110 <i>Gráfica de rapidez en reproducción.</i>	107
Fig. 111 Gráfica de reproducción fluida	107
Fig. 112 Gráfica de calidad de imagen.	108
Fig. 113 Gráfica de fluidez de la transmisión.	108
Fig. 114 Gráfica de sincronización de audio.	109
Fig. 115 Gráfica de respuesta de la barra de progreso.	109
Fig. 116 Gráfica de artefactos visuales	110
Fig. 117 Gráfica de estabilidad de la transmisión.	110
Fig. 118 <i>Gráfica de experiencia general de visualización.</i>	111

Capítulo I. Introducción

En este capítulo se propone introducir el tema de investigación, donde se describen los antecedentes, los problemas que aborda, los objetivos que se persiguen y la importancia del estudio. Así mismo se describen los alcances y limitaciones que se presentan en el proceso.

1.1 Introducción

La transmisión de video hoy en día es muy importante para medios informativos, de entretenimiento o incluso video vigilancia. Asegurar un correcto funcionamiento es clave, en este proyecto se pretende abordar los temas principales para la emisión de video a través de internet con el uso de servidores, protocolos de difusión y codecs.

Este proyecto se aplica en un entorno real donde se dispone de los equipos necesarios para su funcionamiento en el campo de telecomunicaciones.

Las soluciones de envío de video son diversas hoy en día, variando tanto en costos como en calidad y tipo de contratación del servicio. El procesamiento de video por su propia naturaleza crea un producto costoso por el tipo de solución que se requiere.

Actualmente existen líderes en este tipo de proyectos como lo es Haivision con sus equipos dedicados con tecnología militar y cifrado, usados ampliamente en transferencias de señal seguras, desde embajadas, proveedores de servicio de televisión entre otro tipo de situaciones donde se requiera la difusión de video.

Se define el tipo de transferencia de video que se implementará. A partir de esto es como se establecen ciertas características necesarias para convertirlo en un tipo de cesión de flujo de video, ya sea RF Overlay en Pon, OTT o IPTV.

El presente trabajo aborda la configuración de un servidor, la instalación de las herramientas necesarias para el procesamiento de audio y video, asimismo el procedimiento para la creación de una distribución de señal que emplea el protocolo HLS para evaluar su desempeño y destacar los beneficios que proporciona en términos de compatibilidad y eficiencia.

1.2 Antecedentes

En este apartado se mencionan los orígenes de la difusión de video en sus orígenes hasta la actualidad para comprender como surgió, evoluciono y la tendencia de la tecnología de emisión de video.

“La historia de las antenas comienza en 1887, cuando Heinrich Hertz, un científico alemán, construyó la primera antena para probar la existencia de ondas electromagnéticas, como había predicho el teórico James Clerk Maxwell” (López, 2025).

Esta hazaña muestra la posibilidad de transmitir datos de manera inalámbrica ampliando el panorama para sus distintos usos siendo aplicada principalmente en estaciones de radio. Posteriormente con el auge de la televisión se recibían canales, pero estaban sujetos a considerable interferencia.

“En 1948, en Estados Unidos, John Walson, dueño de una tienda de electrodomésticos, instaló una antena de 70 pies de altura para captar la señal de televisión y transmitirla por cable directamente a los hogares de sus clientes” (UNCOSU, 2025).

Esto mejoraba considerablemente la señal de los televidentes al usar una antena con mayor capacidad de recepción, fue entonces cuando se dieron los primeros pasos a la televisión por cable.

Según UNCOSU (2025), en 1950, fue cuando oficialmente se lanzó el primer sistema comercial de televisión por cable con cobro de suscripción, evolucionando rápidamente, pasando de sistemas comerciales básicos a la expansión internacional de televisión por antena comunitaria (CATV) en 1970. HBO introdujo el contenido Premium y en 1975 la primera transmisión vía satélite, posteriormente en 1980 la televisión por cable se consolidó y era común en toda América Latina.

Ruether (2023) señala que en 1990 el streaming debutaba como prueba de concepto, una banda llamada Severe Tire Damage fue la primera en transmitir en vivo por internet, dicha banda tenía contacto con la comunidad tecnológica siendo pioneros en la emisión de audio y video por medio de una tecnología llamada MBone, también entraron en escena Adobe Flash Player y protocolos como el protocolo de mensajería en tiempo real (RTMP) y el protocolo de streaming en tiempo real (RTSP).

En la actualidad con la implementación de la fibra óptica sucesora del cable. Se arman cabeceras de distribución de televisión, las cuales incluyen equipos receptores de señal satelital. Dicha señal se distribuye principalmente por los proveedores de servicio de internet y televisión por medio de RF Overlay en PON el cual es un método de conversión de radiofrecuencia a luz para ser transportada a través de fibra óptica convirtiendo la señal análoga a digital, tal transporte se realiza mediante transmisores ópticos configurados para la emisión de la señal de televisión, la cual se combina con la de internet, aunque viaja en una longitud de onda diferente sobre la misma fibra. Finalmente, la recepción se lleva a cabo por medio de ONUs o comúnmente conocidos como módems.

El método tradicional que se conoce hoy día, constituye un gasto considerable para las empresas que buscan ampliar su cobertura hacia zonas más alejadas, ya que requiere infraestructura adecuada, sistemas de climatización y personal capacitado. Actualmente los proveedores de televisión utilizan este método para transmitir el servicio a través de módems con capacidad de televisión por cable. Es una solución sencilla y económica que, hasta la fecha continúa siendo una alternativa viable. Se puede observar el diagrama del método tradicional como se muestra en la Figura 1.

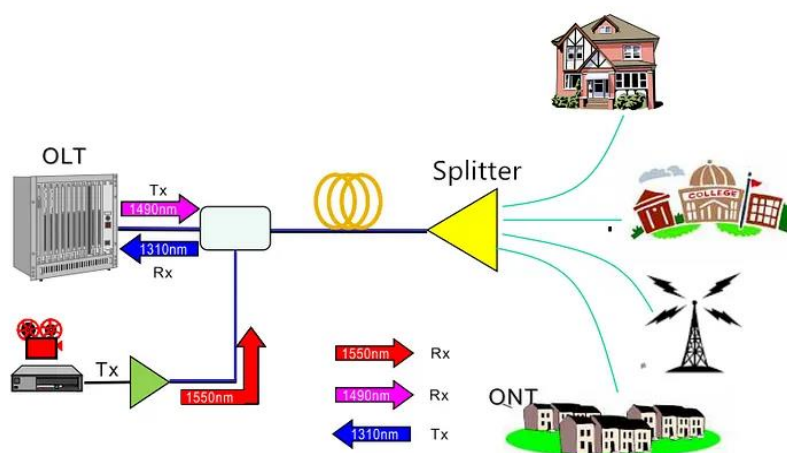


Fig. 1 Diagrama de difusión de video tradicional.

Nota. Tomado de *Visión general de la red de acceso FTTH con GPON*, por Franklin, 2018, Medium.

<https://franklinnn.medium.com/una-visi%C3%B3n-general-de-la-red-de-acceso-ftth-con-gpon-104bc8973d65>

De esta forma se reciben los canales y puede usarse esa señal para transmitir por diferentes protocolos. Es importante que los equipos, conexiones, configuraciones y capacidad de los equipos sea adecuada para evitar artefactos visuales debido a una mala recepción de la señal haciendo que el método tradicional asegure la calidad del servicio dado que es por la red interna de la empresa, sin embargo, no puede ampliar su cobertura tanto como pudiera desearse a lugares más lejanos.

Montoro (2025) indica que por primera vez el streaming superó a la televisión abierta y por cable en Estados Unidos, esto es un indicativo de las preferencias actuales de los usuarios los cuales parecen inclinarse hacia los contenidos de video bajo demanda.

Este análisis nos da un panorama de cómo se creó y evolucionó la difusión de video a lo largo del tiempo, la importancia de conocer su historia es esencial para comprender como el presente desarrollo puede ser de utilidad para el lector al adentrarse al mundo del streaming introduciendo los principios que sostienen este tipo de tecnologías.



Fig. 2 *Representación de consumo de video.*

Nota. Tomado de *Los canales de televisión lineal son una vía de crecimiento en la CEE*, por Prensario Internacional, s.f., Prensario Internacional. <https://www.prensario.net/Los-canales-de-television-lineal-son-una-via-de-crecimiento-en-la-CEE-43563.note.aspx>

1.3 Problemática

La demanda de contenido de video en línea, genera la necesidad de contar con métodos de difusión eficientes y adaptables, actualmente, la forma en que las empresas que dominan el mercado implementan una solución es a través de software propietario, una medida que resulta significativamente costosa, que incluye inversión en infraestructura, equipos especializados, personal capacitado y otras necesidades que deben atenderse para poder transmitir flujos de video hacia ubicaciones lejanas.

Hoy día, existen herramientas que permiten probar de manera fácil y sencilla el beneficio de estas tecnologías y poder considerar cual se adecua mejor, por otro lado, los métodos de transmisión de video evolucionan constantemente, lo que provoca que las soluciones implementadas se vuelvan obsoletas. Esta situación, dificulta ofrecer servicios de emisión eficientes y de calidad que cumplan con la creciente demanda de los usuarios.

Por lo anterior, la problemática detectada implica la existencia de procesos de difusión de video aplicados principalmente por proveedores de servicio de internet y televisión que utilizan técnicas tradicionales costosas y limitadas en cobertura.

1.4 Justificación

Es importante reconocer que hoy por hoy, las transmisiones de video por internet predominan en los medios de información, por lo que resulta fundamental comprender su funcionamiento para ofrecer este servicio de manera eficiente; Conocer las alternativas disponibles permite seleccionar la opción más adecuada según el tipo de difusión que se requiera, considerando el contexto de trabajo y las necesidades específicas del proyecto.

Existen diversas soluciones que pueden adaptarse para garantizar una emisión de video confiable. Por ello, el presente proyecto busca optimizar recursos mediante el uso de herramientas especializadas, la configuración adecuada de los equipos y la implementación de un sistema capaz de ofrecer un producto final confiable y funcional.

Esta solución permite economizar recursos de ancho de banda, hardware, infraestructura necesaria, asimismo permite ampliar la cobertura del servicio de manera confiable y con un nivel de calidad aceptable.

1.5 Objetivo general

Desarrollar un prototipo funcional para transmisión de video a través de internet por medio del protocolo HLS que utilice equipos de cómputo, configuraciones de red y herramientas de procesamiento de video, con el fin de evaluar su desempeño y confiabilidad en transmisiones de larga distancia, y demostrar una solución práctica para las necesidades de emisión actuales.

1.6 Objetivos específicos

- Preparar un escenario de trabajo que incluya la configuración de los equipos de cómputo, redes y herramientas de procesamiento de video para establecer las condiciones necesarias y poder evaluar el desempeño del prototipo de transmisión de video.
- Instalar FFmpeg para el procesamiento de video y asimismo la compatibilidad de los núcleos dedicados de la tarjeta de video haciendo uso de sus drivers oficiales con el fin de aprovechar la tecnología dedicada.
- Crear los servicios en segundo plano en el sistema operativo Linux para poder ejecutar los procesamientos de los diferentes flujos de video en tiempo real y gestionarlos fácilmente.
- Validar el acceso a los flujos de video a través de un sistema middleware que pueda gestionar usuarios para agregar seguridad y control sobre las transmisiones.
- Evaluar los resultados obtenidos con base a la experiencia del usuario mediante una encuesta a través de la escala Likert.

1.7 Alcances y limitaciones

El proyecto de transmisión de video por internet se centra en el desarrollo de un prototipo funcional capaz de transmitir contenido audiovisual utilizando el protocolo HLS, equipos de cómputo robustos, configuraciones de red optimizadas y la herramienta de procesamiento de video FFmpeg. Se implementan ajustes técnicos como bitrate, codecs, puertos y destinos para mejorar el rendimiento del sistema. El prototipo permite realizar transmisiones de video bajo demanda y televisión en vivo, ofreciendo una alternativa eficiente y más económica frente a soluciones comerciales dedicadas.

Si bien el prototipo está diseñado para funcionar correctamente en un entorno controlado, su implementación no está destinada a entornos de producción comercial. Las pruebas se realizan bajo condiciones definidas, lo que permite evaluar de manera efectiva la distribución y la calidad del video, sin cubrir todos los posibles escenarios de tráfico de red real.

El análisis de la calidad del video se realiza mediante observación directa y técnicas básicas de procesamiento de video, demostrando la capacidad del prototipo para entregar contenido de manera confiable. Además, se enfoca en herramientas y protocolos compatibles con FFmpeg, garantizando un sistema funcional y flexible dentro de sus condiciones de prueba. La estabilidad y calidad de la transmisión dependen del ancho de banda disponible y de la red, factores que reflejan situaciones comunes en entornos reales.

1.8 Marco Teórico

El presente marco teórico tiene como objetivo proporcionar una base para los conceptos, principios y tecnologías que están vinculados con la creación de un prototipo para transmitir video a través de internet. Para esto, se detallan los componentes fundamentales que participan en la codificación, emisión y reproducción de contenido audiovisual, además de la infraestructura y las normativas que las respaldan.

1.8.1 Elementos del proceso de comunicación de video digital



Fig. 3 *Elementos del proceso de comunicación de video digital.*

Nota. Tomado de *¿Qué es el streaming de vídeo en directo?*, por Akamai, s.f.
<https://www.akamai.com/es/glossary/what-is-live-video-streaming>

En la transmisión de video digital, los protocolos son una parte fundamental, ya que constituyen un conjunto de reglas que determinan el formato y significado de los mensajes intercambiados entre entidades equivalentes en una misma capa de comunicación. Estas normas aseguran que los dispositivos puedan comunicarse de manera coherente y comprensible, lo cual es especialmente crítico cuando varios equipos deben operar de forma coordinada para enviar contenido audiovisual a través de internet (Tanenbaum & Wetherall, 2012).

Los códecs son otro componente esencial, encargados de comprimir y descomprimir los archivos de video digital, permitiendo que estos sean almacenados, transmitidos y reproducidos de manera eficiente. La elección del códec influye directamente en la calidad de la señal durante la transmisión, siendo determinante para la experiencia del usuario final (Dacast, s.f.).

Según (Rodríguez Solís, 2020). El video digital se entiende como la representación y almacenamiento de imágenes y sonidos mediante sistemas electrónicos que pueden utilizar diferentes formatos, tanto analógicos como digitales. Su función es capturar y reproducir contenido audiovisual de forma que pueda ser manipulado, transmitido y visualizado en diversos dispositivos

En la red de distribución de video, el servidor actúa como un nodo central que proporciona servicios, recursos o información a los dispositivos conectados, recibe solicitudes de los clientes, las procesa y envía las respuestas correspondientes, convirtiéndose en un elemento clave para la operación de la red (GoDaddy, 2024).

Los dispositivos en la red requieren de identificación única para poder localizarse e intercambiar datos, función que cumple la dirección IP. Esta identificación permite la interconexión de los equipos y la correcta entrega de información entre ellos (Fortinet, s.f.).

Acerca de los protocolos de transporte, TCP y UDP estos, cumplen roles distintos según las necesidades de la comunicación. TCP (Protocolo de Control de Transmisión) se encarga de dividir los datos en paquetes, asegurando que lleguen completos y en el orden correcto, lo que proporciona una comunicación confiable y robusta (Hostinger, 2025). Por otro lado, UDP (User Datagram Protocol) prioriza la velocidad sobre la fiabilidad, permitiendo la emisión rápida de datos sin establecer previamente una conexión formal, lo que resulta útil en aplicaciones donde la latencia es crítica, como la reproducción de video en tiempo real (Cloudflare, s.f.).

Finalmente, los métodos de distribución de datos determinan cómo se entrega la información a los receptores. En la comunicación multicast, un emisor puede enviar datos a múltiples receptores simultáneamente, optimizando el uso de la red y reduciendo la carga de tráfico (Hernández Palacios & Hernández Hernández, 2017). Por contraste, la comunicación unicast se limita a la cesión directa de un emisor a un receptor, siendo más simple y adecuada para aplicaciones punto-a-punto o cliente/servidor (Hernández Palacios & Hernández Hernández, 2017).

1.8.2 Elementos de redes ópticas de alta velocidad

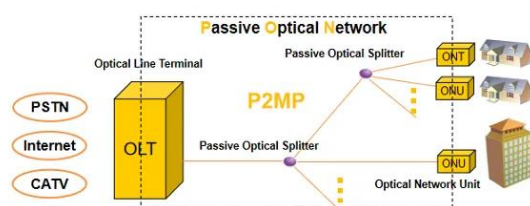


Fig. 3 Elementos de red óptica

Nota. Tomado de ¿Qué es OLT (Optical Line Terminal)? Características y Aplicaciones, por Vsolcn, 2025. <https://es.vsolcn.com/blog/what-is-olt.html>

En las redes de fibra óptica, varios dispositivos y tecnologías trabajan de manera coordinada para garantizar la distribución eficiente de datos y señales de video hacia los usuarios finales. Uno de los elementos centrales es la OLT (Optical Line Terminal), que actúa como el punto principal de la Red Óptica Pasiva (PON). Este equipo convierte señales eléctricas en ópticas para la transferencia descendente hacia los terminales del cliente y realiza la operación inversa para la emisión ascendente. Además, la OLT gestiona la comunicación con los usuarios a través de la infraestructura de fibra, permitiendo la distribución simultánea de datos y televisión mediante distintas frecuencias (VSOL, 2025).

La integración de la señal de televisión en la red PON se realiza mediante técnicas de multiplexación por longitud de onda, conocidas como RF Overlay en PON. En este método, la señal de televisión se transmite a través de un transmisor láser de 1550 nm y se combina con el tráfico de datos de la OLT, que utiliza otras longitudes de onda (1310 y 1490 nm). De esta forma, se logran transmitir simultáneamente datos y señales de radiofrecuencia a través de la misma fibra, distribuyéndolas a los terminales ópticos de los clientes mediante divisores de la red (Premlink, s.f.).

En el extremo del usuario se encuentra la ONU (Unidad de Red Óptica), que convierte las señales ópticas recibidas en señales eléctricas y proporciona múltiples interfaces de servicio. A diferencia de un módem convencional, la ONU está específicamente diseñada para redes de fibra óptica, asegurando que la transmisión sea eficiente y compatible con los servicios proporcionados por la OLT (VSOL, 2025).

En cuanto a la distribución de la señal de televisión, los sistemas CATV (Community Antenna Television) se combinan con la fibra óptica para transmitir señales de televisión por radiofrecuencia, llegando a múltiples usuarios a través de redes de fibra y cables coaxiales. Esta combinación permite continuar utilizando los principios de la televisión por cable tradicional mientras se aprovechan las ventajas de la fibra óptica, ofreciendo una transición hacia tecnologías más modernas como IPTV o VOD (TecnoLibre, 2024).

La infraestructura de fibra también integra dispositivos que facilitan la interconexión y amplificación de la señal. El combinador óptico permite acoplar la salida de energía de una fibra transmisora con la receptora, minimizando las pérdidas y alteraciones en el sistema (OEM-Laser, 2023). Por su parte, los amplificadores de fibra dopada con erbio, conocidos como EDFA, incrementan la potencia de las señales ópticas directamente, compensando las pérdidas en enlaces de larga distancia y manteniendo la calidad de emisión sin necesidad de conversión a señales eléctricas (FiberWDM, 2024).

Por ultimo, los NAP (Network Access Point) funcionan como puntos de acceso intermedios dentro de las redes FTTH (Fiber To The Home). Estos dispositivos permiten distribuir la fibra óptica desde la red troncal hacia las viviendas u otros usuarios finales, gestionando la señal y garantizando que llegue correctamente a múltiples clientes (Óptica Naranjo, 2022). La combinación de estos elementos asegura que la emisión de datos y video sobre fibra óptica sea eficiente, confiable y escalable, adaptándose a las necesidades de cada red.

1.8.3 Tipos de transmisión y servicios

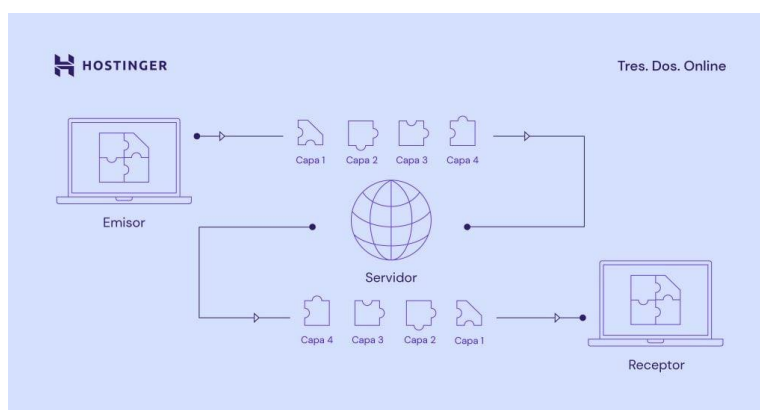


Fig. 4 *Diagrama de transmisión*

Nota. Tomado de Protocolo TCP: definición y funcionamiento, por D. Boada, 2025, Tutoriales Hostinger.

<https://www.hostinger.com/mx/tutoriales/protocolo-tcp>

En la actualidad, la distribución de contenido audiovisual a través de internet se realiza mediante distintos tipos de servicios, cada uno con características y ventajas específicas. Entre ellos, el IPTV (Internet Protocol Televisión) permite ofrecer tanto televisión en directo como contenidos bajo demanda a cualquier dispositivo conectado a internet. Este tipo de distribución se realiza generalmente sobre una red gestionada por el proveedor de servicios, lo que asegura una mayor calidad de señal, baja latencia y menor riesgo de interrupciones durante la visualización (Boada, 2025).

Por otra parte, los servicios OTT, o “Over The Top”, utilizan internet público para entregar contenido audiovisual sin depender de redes gestionadas. Esto incluye cadenas de televisión tradicionales que transmiten sus programas en línea, así como plataformas de streaming que ofrecen contenido exclusivo, en ocasiones sin anuncios. OTT proporciona gran flexibilidad y accesibilidad, ya que permite reproducir contenido en múltiples dispositivos. Sin embargo, a diferencia del IPTV, la calidad del servicio depende directamente de la estabilidad de la conexión a internet del usuario, sin garantías de latencia ni continuidad de la señal (Broadcast Management, s.f.).

Un tercer tipo de servicio ampliamente utilizado es el VOD, o vídeo bajo demanda. Este sistema permite a los usuarios acceder a contenidos multimedia en cualquier momento, eliminando las restricciones de la televisión lineal. La implementación de VOD puede realizarse tanto sobre redes IPTV como OTT, ofreciendo la posibilidad de reproducir videos de manera inmediata según la conveniencia del espectador (Castr, 2020).

El streaming es una tecnología que permite enviar y recibir contenido multimedia como vídeo o audio a través de Internet de forma rápida y eficiente, sin necesidad de descargar por completo el archivo antes de reproducirlo. En este sistema, el contenido almacenado en servidores de streaming se comprime y se divide en pequeños paquetes de datos que pueden transmitirse fácilmente a través de la red. Cuando un usuario solicita acceso al contenido, el servidor envía continuamente fragmentos suficientes como para mantenerse siempre unos segundos por delante de la reproducción, lo que permite una experiencia fluida y sin interrupciones, minimizando el uso de espacio de almacenamiento en el dispositivo del usuario y reduciendo los tiempos de espera (Akamai, s.f.).

En conjunto, estos servicios representan la evolución de la televisión y la distribución de contenido audiovisual, adaptándose a las necesidades de accesibilidad, flexibilidad y calidad que los usuarios modernos requieren.

1.8.4 Equipos de transmisión y recepción

En los sistemas de emisión de video digital, la eficiencia en el envío y la recepción de la señal depende de diversos equipos especializados. La modulación QAM, o modulación de amplitud en cuadratura, permite codificar datos digitales en señales analógicas combinando dos portadoras con distintas amplitudes y fases. Esta técnica posibilita transmitir una mayor cantidad de información sin requerir más ancho de banda, optimizando así la utilización del canal de comunicación (Digi International, s.f.).

Podemos observar en la Figura 5 el diagrama donde se reciben los canales.

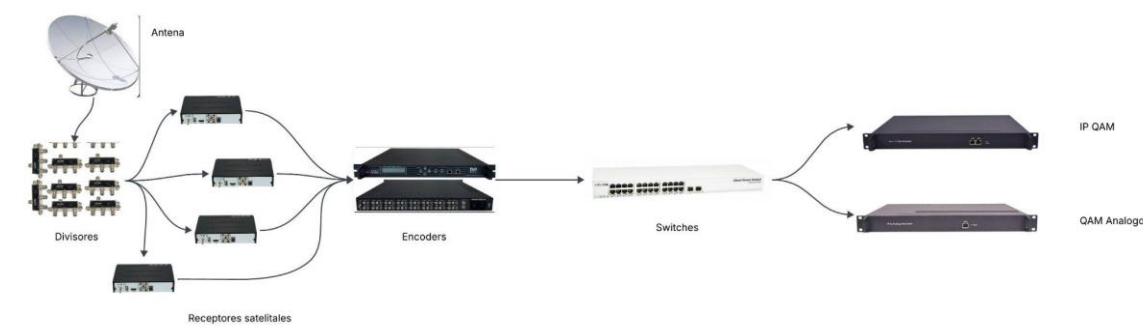


Fig. 5 Diagrama de infraestructura de recepción de canales.

Fuente: Elaboración propia.

El encoder de video es otro componente clave en la transmisión, ya que se encarga de convertir la señal de video original en un formato comprimido. Al eliminar redundancias y datos innecesarios, el encoder permite enviar la señal de manera eficiente, reduciendo la ocupación de ancho de banda y el espacio de almacenamiento sin comprometer significativamente la calidad visual (Outlet Audio Video, s.f.).

Complementariamente, el decoder recibe la señal comprimida y la transforma nuevamente a un formato que puede ser reproducido por los dispositivos de visualización. De esta manera, garantiza que el contenido llegue al usuario final con fluidez y calidad, funcionando como el componente inverso del encoder en el proceso de distribución de video (Outlet Audio Video, s.f.).

De forma colectiva, la combinación de QAM, encoders y decoders asegura que los sistemas de emisión y recepción de video digital sean eficientes, confiables y capaces de mantener la calidad de la señal a lo largo del recorrido desde el emisor hasta el receptor.

1.8.5 Equipos de redes

Dentro de las redes de distribución de video digital, los routers y switches desempeñan un papel fundamental al garantizar que los datos lleguen correctamente a su destino. Los routers actúan como encaminadores del tráfico, dirigiendo los paquetes de datos que contienen información de distintos tipos, desde archivos hasta transmisiones de video y comunicaciones en línea. Esta función asegura que cada paquete llegue de manera eficiente y ordenada a su destino final, optimizando la entrega de contenido audiovisual en la red (Cisco, s.f.).

Por su parte, los switches permiten la interconexión de múltiples dispositivos dentro de una misma red, como computadoras, impresoras, puntos de acceso inalámbricos y servidores. Gracias a ellos, los dispositivos pueden compartir información y recursos de manera eficiente, lo que es especialmente importante para la difusión de video, dado que este tipo de datos requiere un procesamiento rápido y constante para mantener la calidad y continuidad de la señal (Cisco, s.f.).

Routers y switches conforman la base de la infraestructura de red, asegurando que la emisión de video digital se realice de manera confiable, con un flujo de datos organizado y optimizado según las necesidades de la red y los usuarios finales.

1.8.6 Equipos de infraestructura



Fig. 6 Infraestructura en rack

Nota. Tomado de *Aprende qué es un armario rack y qué modelos puedes comprar*, por S. De Luz, 2025, RedesZone. <https://www.redeszone.net/tutoriales/redes-cable/que-es-armario-rack-modelos/>

La infraestructura física de una red de transmisión de video digital incluye diversos equipos diseñados para soportar, proteger y optimizar la operación de los sistemas de comunicación. Entre ellos, las antenas funcionan como conductores metálicos capaces de radiar y recibir ondas electromagnéticas. Estas antenas convierten las ondas en señales eléctricas al recibirlas y, a la inversa, transforman la energía eléctrica en ondas electromagnéticas al transmitirlas, facilitando la propagación de señales de televisión y otros servicios de comunicación (Apuntes de Electrónica – UTN, s.f.).

Otro componente esencial es el rack, un armario metálico que permite alojar y organizar de manera segura todos los equipos informáticos y de telecomunicaciones de una red, incluyendo servidores, routers, switches, sistemas de almacenamiento y cableado. Además de proteger los dispositivos, el rack facilita la ventilación, el mantenimiento y la gestión de la infraestructura, asegurando que los sistemas funcionen de manera ordenada y eficiente dentro de la red (RedesZone, s.f.).

En general, antenas y racks constituyen la base física que soporta la emisión y recepción de video digital, garantizando tanto la eficiencia de las señales como la seguridad y organización de los equipos que permiten su operación.

1.8.7 Organismos y regulaciones

La supervisión y regulación de las telecomunicaciones y la radiodifusión en México ha experimentado cambios recientes significativos. Hasta octubre de 2025, el Instituto Federal de Telecomunicaciones (IFT) fue el organismo encargado de garantizar el uso adecuado de los servicios de radiodifusión y telecomunicaciones, promoviendo la competencia, la conectividad y la protección de los derechos de las audiencias y usuarios en todo el país (Instituto Federal de Telecomunicaciones, s.f.). Sin embargo, a partir del 17 de octubre de 2025, con la extinción del IFT, sus cuentas institucionales en redes sociales quedaron inactivas y también finalizó la atención a usuarios a través de estos canales, marcando un cambio en la supervisión regulatoria de este sector (IFT, 2025, Twitter).

Con la desaparición del IFT, la Agencia de Transformación Digital y Telecomunicaciones (ATDT) asume un papel relevante en la centralización de los recursos tecnológicos del gobierno, buscando unificar capacidades, generar ahorros públicos y optimizar la utilidad de los sistemas digitales del Estado (Agencia de Transformación Digital y Telecomunicaciones, s.f.). Dentro de esta estructura, la Comisión Reguladora de Telecomunicaciones (CRT) funciona como un órgano administrativo desconcentrado con independencia técnica, operativa y de gestión. Su función principal es garantizar el desarrollo eficiente de las telecomunicaciones y la radiodifusión en México, asegurando que la regulación del sector sea profesional y autónoma (Comisión Reguladora de Telecomunicaciones, s.f.).

En conjunto, estos organismos reflejan la evolución de la supervisión del sector de telecomunicaciones en México, pasando de un ente regulador centralizado como el IFT a un modelo más distribuido, con la ATDT coordinando recursos y la CRT garantizando la regulación independiente y técnica de los servicios de radiodifusión y telecomunicaciones.

1.8.8 Estándares y tecnologías de transmisión

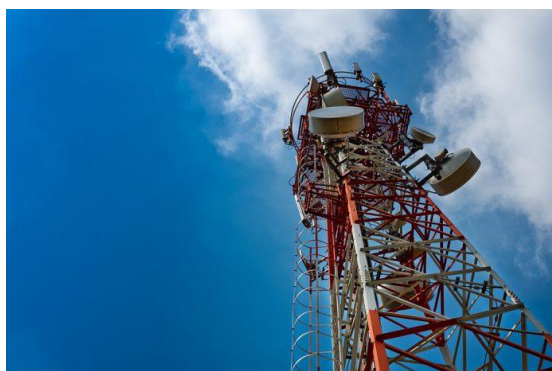


Fig. 7 Antena

Nota. Tomado de *Hablemos de antenas: su historia y evolución*, por T. López, 2025, Radionotas.
<https://radionotas.com/2025/02/25/hablemos-de-antenas-su-historia-y-evolucion/>

La transferencia de video digital se sustenta en una serie de estándares y tecnologías que garantizan compatibilidad, calidad y eficiencia en la distribución de contenido audiovisual. Uno de los principales organismos responsables de estos estándares es la Asociación de Estándares de Radiodifusión (ATSC), una organización internacional sin fines de lucro que desarrolla normas voluntarias para la radiodifusión multimedia. La ATSC integra a miembros de diversas industrias, incluyendo radiodifusión, cine, electrónica de consumo, informática, cable, satélite y semiconductores, con el objetivo de generar estándares técnicos de manera colaborativa y consensuada (ATSC, s.f.).

Otro estándar ampliamente utilizado es DVB (Digital Video Broadcasting), un conjunto de normas internacionales abiertas para la emisión de televisión digital a través de diferentes medios: satélite (DVB-S), cable (DVB-C) y redes terrestres (DVB-T). Este estándar fue desarrollado a principios de la década de 1990 por un consorcio industrial que incluía emisoras, operadores de red y fabricantes, y se ha convertido en una referencia para garantizar interoperabilidad y calidad en la distribución de señal digital (Samim Group, 2024).

En el ámbito de la televisión terrestre, la Televisión Digital Terrestre (TDT) representa un sistema de transferencia digital de señales abiertas, o gratuitas, que moderniza la forma en que los usuarios acceden a este servicio. La TDT permite la recepción de contenido digital sin costo, promoviendo la difusión de información y entretenimiento de manera accesible a la población (TDT, s.f.).

De manera cooperativa, estos estándares y tecnologías proporcionan la base para la transmisión de video digital confiable, interoperable y accesible, facilitando la adopción de nuevas plataformas y servicios mientras se mantiene la calidad de la señal para los usuarios finales.

1.8.9 Tecnologías y herramientas de procesamiento de video

El procesamiento de video digital requiere tecnologías y herramientas que permitan comprimir, codificar, transmitir y reproducir contenido de manera eficiente. Uno de los protocolos más utilizados en este ámbito es HLS (HTTP Live Streaming), desarrollado por Apple para su ecosistema de dispositivos y posteriormente adoptado de manera amplia en la industria. HLS permite dividir los archivos de video en segmentos más pequeños y transmitirlos mediante el protocolo HTTP. Esta característica lo hace especialmente flexible y compatible con la mayoría de dispositivos, funcionando tanto para streaming en directo como para contenidos a la carta (Cloudflare, s.f.).

Otra herramienta fundamental es FFmpeg, un framework multimedia altamente versátil que permite decodificar, codificar, transcodificar, multiplexar, demultiplexar, transmitir, filtrar y reproducir prácticamente cualquier formato de video. Su compatibilidad con estándares antiguos y modernos lo convierte en un recurso indispensable para desarrolladores y profesionales que trabajan con contenido audiovisual, asegurando interoperabilidad entre distintos formatos y protocolos (FFmpeg, s.f.).

Adicionalmente, los CUDA Cores, desarrollados por NVIDIA, constituyen núcleos especializados para la computación en paralelo de propósito general. Gracias a su capacidad para ejecutar múltiples operaciones simultáneamente, estos núcleos son extremadamente útiles para realizar cálculos intensivos, como los requeridos en el procesamiento de gráficos y video de alta complejidad (Sole, 2025).

En unión, HLS, FFmpeg y CUDA Cores proporcionan las bases tecnológicas necesarias para la codificación, transmisión y procesamiento de video digital, asegurando eficiencia, compatibilidad y calidad en los sistemas de streaming modernos.

1.9 Estado del arte

La transmisión de video por internet evolucionó rápidamente debido a las necesidades de comunicación de forma inmediata, de forma estable y con buena calidad. A lo largo de la historia han surgido muchas herramientas que facilitan la codificación, transporte y distribución de video. Este estudio del arte de forma histórica muestra la aparición y evolución de cuatro herramientas similares a lo que se pretende realizar con este proyecto de tesis. Esto con el objetivo de entender cómo dichas tecnologías han contribuido al desarrollo de emisión de video de forma moderna y el cómo se relaciona con el proyecto que se lleva a cabo.

- vMix: vMix es una solución para producción y distribución de video en vivo, fué desarrollado por StudioCoast Pty Ltd, igual que varias soluciones buscaba tener una opción más accesible que los mezcladores de hardware costosos. Permite grabar y editar producciones profesionales en vivo desde una PC con Windows, Permite varias entradas como Cámaras IP, vídeos, imágenes, SRT, títulos, audio, entre otras funciones. Su lanzamiento oficial fué en 2009 y ha evolucionado mucho hasta la fecha (StudioCoast Pty Ltd., s.f.).
- OBS Studio: OBS comenzó como un pequeño proyecto en 2012 creado por Hugh "Jim" Bailey, este creció aceleradamente y con la ayuda de muchos colaboradores que mejoraron la aplicación. Se caracteriza por codificación H.264 y AAC, soporta aceleración por hardware de NVIDIA e Intel, tiene un número ilimitado de escenas y fuentes. Se usa principalmente para streaming en vivo RTMP a Twitch, YouTube, DailyMotion Etc. Genera archivos MP4 o FLV. Es libre y gratuita. Tiene una amplia variedad de herramientas, es fácil de usar y tiene la posibilidad de implementar plugins para expandir la funcionalidad (Ecured, s.f.).

La herramienta actualmente es muy usada por profesionales y aficionados de la transmisión de video ya que es fácil de instalar y configurar, principalmente se usa para streaming de videojuegos o transmisiones de una o varias pantallas personalizando los fondos, capturando audio y video de distintas fuentes.

Esto es relevante ya que nos da la idea de que se puede crear una aplicación que procese el video y lo transmita a través de internet.

- **Larix:** Larix Broadcaster es un software de transmisión móvil gratuito para dispositivos iOS y Android, permite la emisión cómodamente desde un dispositivo móvil, transmite contenido de audio y video en directo. Soporta varios protocolos como SRT, RTMP, RIST, etc. Puede usar tanto la cámara delantera como la trasera del equipo y grabar contenido guardado de capturas de pantalla, permite talkback para audio. Sirve a su vez para transmitir en directo a servicios como Facebook Live, YouTube Live y Twitch entre otros. Compatible con codecs como H.264 y H.265. Pero para habilitar todas las funcionalidades se tiene que pagar una suscripción no muy costosa. Se lanzó entre 2012 y 2013 aunque no hay una fecha de lanzamiento documentada. (Softvelum, s.f.; Softvelum, 2018).

En su versión gratuita, permite hacer una transmisión básica para comprobar la funcionalidad de la misma, la configuración es bastante sencilla ya que, si el transmisor no tiene conocimiento técnico para crear la sesión, basta con que escanee un código QR con la configuración la cual se genera en la página y se anexa al código con los datos necesarios. Esto podría servir como ingesta a un sistema de transmisión ya que también soporta varios protocolos.

- **SRT Gateway:** La puerta de enlace SRT de Haivision es una solución de enrutamiento seguro de transmisiones de diferentes tipos de redes IP, optimiza la transmisión de video en redes impredecibles cuando se enfrenta a pérdida de paquetes, la latencia y un ancho de banda variable. Es una solución costosa ya que sus orígenes fueron militares para transmisión segura y encriptada, fue desarrollado por Haivision en 2013 pero se liberó como código abierto alrededor de 2017 cuando se formó la SRT Alliance.

Este protocolo tiene la principal ventaja de ofrecer baja latencia con recuperación de paquetes por pérdida con el mecanismo (ARQ - Automatic Repeat reQuest). Haciendo muy confiable la transmisión sobre internet. Los equipos dedicados de la marca son el tope de gama y por ende son muy caros por lo que se considera el uso del protocolo.

En resumen, todas las herramientas son útiles para transmitir video con sus diferentes capacidades y limitaciones. Mientras que OBS y vMix son ideales en producciones para PC. Larix es mejor para cesión móvil y por su facilidad de uso. SRT Gateway garantiza confiabilidad y ruteo profesional pero es un equipo bastante costoso.

Al realizar pruebas con distintos sistemas, es posible establecer los requerimientos necesarios para obtener un producto confiable, con las características adecuadas según el entorno de trabajo.

Estas herramientas permiten observar la evolución de las soluciones, desde opciones libres y accesibles hasta equipos profesionales.

Es fundamental considerar la compatibilidad entre formatos, códecs y protocolos de transmisión, ya que no todos son interoperables.

En cuanto a los códecs de video, H.264 es altamente compatible con la mayoría de dispositivos y navegadores, ofreciendo un buen equilibrio entre calidad y compresión. H.265 proporciona mayor compresión, pero requiere más recursos y su compatibilidad es menor.

Respecto a los códecs de audio, AAC es ampliamente utilizado y compatible; MP3 es compatible pero menos eficiente y no recomendado para streaming moderno; AC3 es profesional, multicanal y orientado a sonido envolvente, ofreciendo mayor compatibilidad con sistemas de cine en casa antiguos, aunque menos eficiente que AAC.

Entre los contenedores que soportan estos códecs, MP4 es ampliamente usado, ya que admite H.264, H.265 y AAC, siendo ideal para DASH o HLS con MP4 fragmentado. FLV es tradicional para RTMP, aunque su uso actual es limitado. MPEG-TS es el contenedor estándar para HLS, donde los segmentos .ts permiten transmitir video y audio de manera confiable sobre HTTP.

De acuerdo con Adobe (2024), “la mayoría de los formatos utilizan la compresión para disminuir el tamaño del archivo y la velocidad de bits, para lo cual reducen la calidad de forma selectiva”. Además, mencionan que “La compresión es esencial para reducir el tamaño de las películas de forma que se puedan almacenar, transmitir y reproducir con eficacia”. Esta compresión es un elemento fundamental en cualquier flujo de trabajo de distribución, ya que permite optimizar el uso del ancho de banda y asegurar que los contenidos puedan enviarse de manera eficiente.

Por estas razones, en este proyecto se optó por HLS como protocolo de transmisión usando el formato contenedor MPEG-TS, con H.264 como codec de video y AAC para codec de audio, garantizando compatibilidad y confiabilidad. Se puede observar la composición en la Figura 8 de una manera más visual como están relacionados dichos componentes.

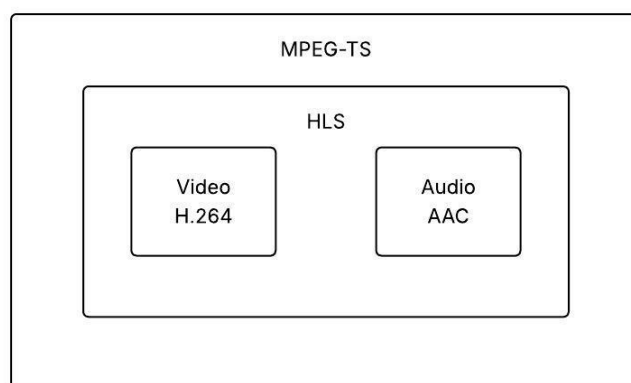


Fig. 8 *Diagrama de componentes de un segmento HLS.*

Fuente: Elaboración propia.

Capítulo II. Metodología

En este capítulo se describe a detalle el enfoque de la investigación, diseño y procesos que se utilizaron para realizar este proyecto con el propósito de describir cómo se desarrolló de manera que cualquier lector pueda comprender, analizar y ejecutar si lo desea.

2.1 Uso y Justificación de la Metodología

Para el desarrollo del proyecto se evaluaron varias metodologías que se adaptan al desarrollo del proyecto, cada una con particularidades diferentes teniendo en consideración la naturaleza experimental del mismo.

“Scrum es una metodología que ayuda a los equipos a colaborar y realizar un trabajo de alto impacto. La metodología Scrum proporciona un plan de valores, roles y pautas para ayudar a tu equipo a concentrarse en la iteración y la mejora continua en proyectos complejos” (Martins, 2025, p. 7).

Scrum por ejemplo usa el desarrollo en sprints con roles definidos como Scrum Master, Product Owner etc. Se centra en entregables funcionales, es bueno para proyectos complejos y equipos de desarrollo grandes. Esta metodología para un prototipo académico puede ser demasiado estructurada y burocrática.

“La metodología Kanban se implementa por medio de tableros Kanban. Se trata de un método visual de gestión de proyectos que permite a los equipos visualizar sus flujos de trabajo y la carga de trabajo” (Martins, 2025, p. 4).

Kanban es un sistema visual de tareas con tarjetas que se mueven por columnas en los estados pendiente, en progreso y terminado. Es flexible, permite ver el flujo de desarrollo y priorización en tiempo real. Es más ligero que Scrum, pero se centra más en gestión de tareas que en validación de funcionalidades concretas.

“El modelo de prototipo es un tipo de enfoque del ciclo de vida del desarrollo de software (SDLC) en el que se crea una versión preliminar del software, denominada prototipo, para demostrar las características principales y recopilar comentarios” (Liyanage, 2025).

En contraste, la metodología de prototipos resulta especialmente adecuada debido a la naturaleza experimental del desarrollo de un sistema de emisión de video. Este tipo de metodología permite construir versiones preliminares del sistema para evaluar rápidamente conceptos técnicos como la configuración de códecs, contenedores, protocolos de transmisión y rendimiento en diferentes escenarios. Cada prototipo funciona como una instancia de validación que permite identificar problemas, ajustar parámetros y mejorar el diseño antes de llegar a una implementación final.

Asimismo, la metodología de prototipos facilita la obtención de retroalimentación temprana, lo que es fundamental en proyectos donde la calidad de experiencia del usuario como la fluidez de reproducción, latencia o estabilidad del stream es un aspecto clave. Esta retroalimentación continua reduce significativamente el riesgo de avanzar hacia una solución que no cumpla con los requerimientos. Por estas razones, la metodología de prototipos ofrece un enfoque iterativo, flexible y centrado en la experimentación, lo cual coincide con las necesidades del proyecto.

Las retroalimentaciones nos permiten comprender mejor el cómo se puede mejorar la construcción del prototipo para implementar la versión final con ayuda de la participación activa del usuario. Scrum o Kanban son útiles para proyectos grandes o gestión de tareas, pero no proporcionan un enfoque tan directo y rápido en la validación de funcionalidades como lo hace la metodología de prototipos.

La retroalimentación del usuario y la posibilidad de hacer ajustes para mejorar el algoritmo a lo largo del refinamiento de las iteraciones convierten a la metodología de prototipos en la más adecuada para el desarrollo del proyecto, reduciendo el riesgo de no cumplir con lo requerido de manera inicial.

2.2 Aplicación de la Metodología

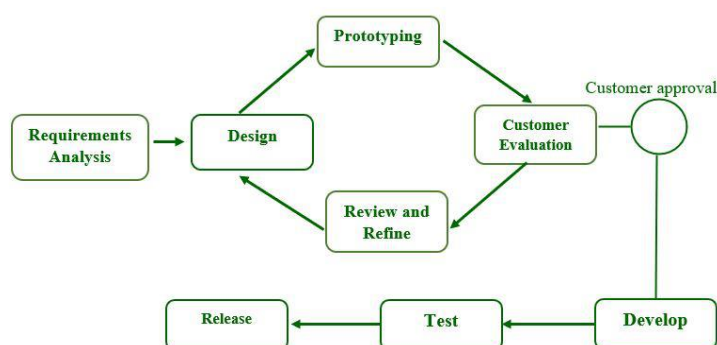


Fig. 9 Fases del modelo de prototipado.

Nota. Tomado de Software Prototyping Model and Phases, por GeeksforGeeks, 2023. Recuperado de <https://www.geeksforgeeks.org/software-engineering/software-prototyping-model-and-phases/>

2.2.1 Metodología de prototipos

En este capítulo, se muestra la planificación de manera teórica enfocada en el desarrollo práctico del desarrollo, aplicando la metodología de prototipos desde el diseño, la estructura inicial del proyecto, su construcción y evaluación de resultados obtenidos en las pruebas donde posteriormente se hace el refinamiento concluyendo en la implementación final del prototipo funcional.

Fase 1: Definición de requerimientos

En la presente fase se realiza el análisis de la estructura interna del centro de recepción y control de transmisiones de video donde se hacen las observaciones pertinentes para definir los requisitos iniciales que cumplan las demandas de una emisión de calidad. Siendo así las bases para que el proyecto pueda lograr su objetivo principal que es la distribución de video a través de internet.

- Transmisión de bajo peso y poca latencia.
- Servidor de video basado en Linux con NGINX.
- Transcodificación de vídeo con FFmpeg.
- Reproducción de video desde PCs y móviles.

En la Figura 10 se puede observar la estructura del proyecto con sus componentes.

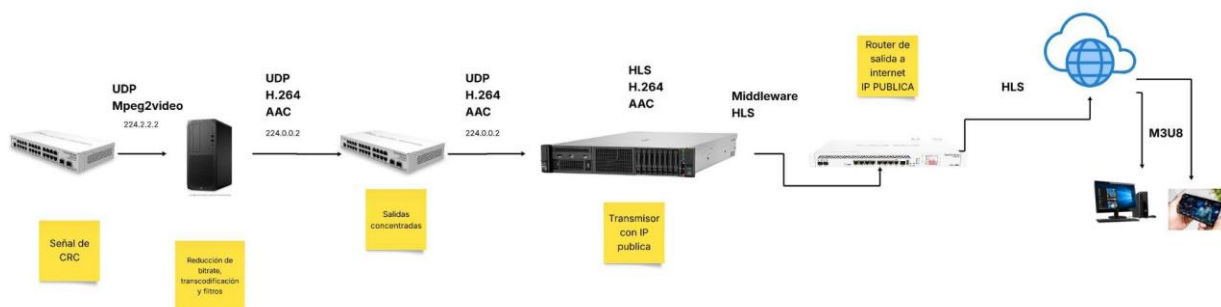


Fig. 10 *Diagrama de infraestructura de procesamiento y transmisión de video.*

Fuente: Elaboración propia.

El switch de red concentra los flujos UDP multicast en formato MPEG-2 Video, un códec con menor eficiencia de compresión frente a estándares actuales. Aunque es un formato antiguo, todavía se utiliza en ciertos proveedores de servicios y sistemas de televisión, especialmente cuando se trabaja con infraestructura heredada. Estos flujos provienen de canales recibidos por antena en definición estándar (SD) 720×480 píxeles bajo el formato NTSC, típico de transmisiones destinadas a televisores antiguos y generalmente entregado en señal entrelazada (480i).

Posteriormente, dichos flujos son enviados al equipo o estación de trabajo encargada del procesamiento, donde se realiza el transcoding de MPEG-2 Video a H.264 también es conocido como Advanced Video Coding (AVC), esto con el propósito de mejorar la compatibilidad con dispositivos actuales y reducir el ancho de banda necesario para la transferencia. Este proceso aprovecha la capacidad de cómputo de los núcleos CUDA para acelerar la conversión y optimizar el rendimiento del sistema.

Una vez procesadas, las salidas de video se envían hacia un segundo switch que concentra los flujos transcodificados. Desde ahí, se redirigen a un servidor que funciona como transmisor y generador de listas M3U8, fundamentales para la distribución de contenido mediante el protocolo HLS. Este servidor también actúa como middleware, empleando software de código abierto para gestionar la seguridad, autenticación y estabilidad del servicio de transmisión de video, asegurando así un flujo confiable y accesible para los usuarios finales.

Fase 2: Desarrollo del prototipo inicial

Se construye la versión inicial en base a las funcionalidades que requiere el sistema para lo cual se puede hacer una pequeña prueba en base a las investigaciones realizadas anteriormente con el software FFmpeg, y una transmisión pública sobre internet.

Dado que FFmpeg es compatible tanto con sistemas Windows y Linux, procedemos con la prueba rápida sobre windows, para lo cual primero se tiene que instalar la herramienta.

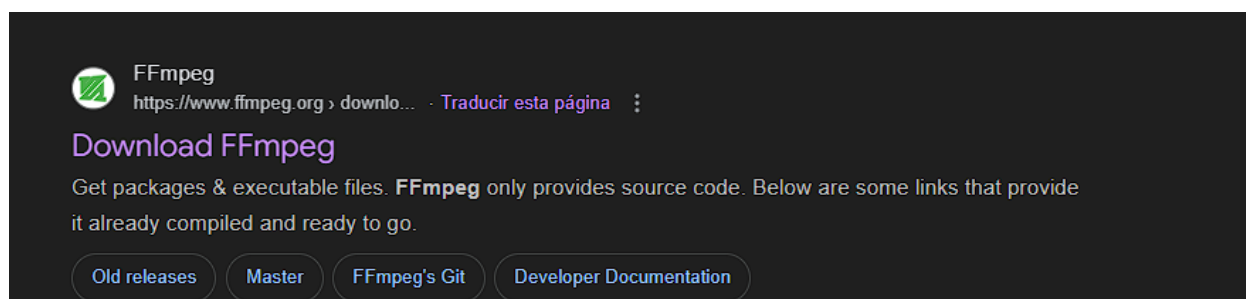


Fig. 11 *Página de descarga oficial de FFmpeg en los resultados de búsqueda de Google.*

Nota. Captura de pantalla de una búsqueda realizada por el autor en Google (2025).

Se da click en Download FFmpeg.

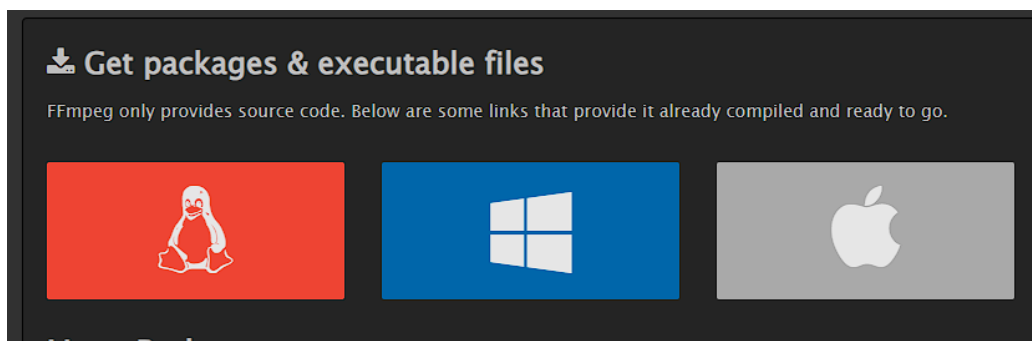


Fig. 12 Sección de descarga de paquetes y archivos ejecutables en la página oficial de FFmpeg.

Nota. Captura de pantalla tomada del sitio web oficial de FFmpeg.

Seleccionando el sistema operativo Windows donde se nos redirecciona a un link de descarga de la herramienta.

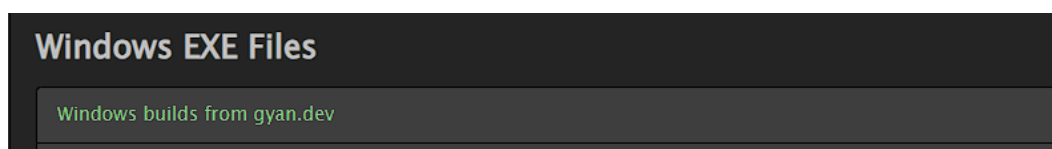


Fig. 13 Link de descarga de proveedor de software gyan.dev

Nota. Captura de pantalla tomada del sitio web oficial de FFmpeg.

Posteriormente podemos encontrar dos formas de descarga de archivos comprimidos, un archivo con las herramientas esenciales y el otro siendo la versión completa.



Fig. 14 Link de descarga de FFmpeg comprimido en la página de gyan.dev

Nota. Captura de pantalla tomada del sitio web oficial de gyan.dev.

Donde se puede elegir entre la opción completa quedando un archivo comprimido.

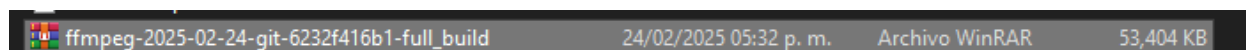


Fig. 15 Archivo comprimido descargado de la herramienta FFmpeg.

Nota. Captura de pantalla realizada por el autor mostrando la descarga comprimida de la herramienta FFmpeg.

Al descomprimir dicho archivo se encuentra la estructura de carpetas de la herramienta FFmpeg.

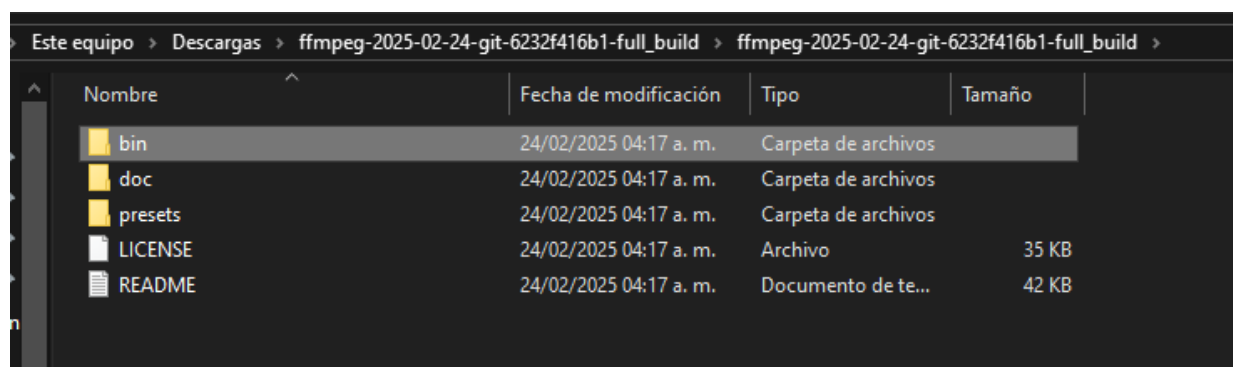


Fig. 16 Estructura de carpetas de la herramienta FFmpeg para windows.

Nota. Captura de pantalla realizada por el autor mostrando el archivo descomprimido de FFmpeg.

Se observa una carpeta bin que contiene las herramientas principales de la herramienta que servirán para ejecutarlas sobre windows y hacer las pruebas pertinentes.

Se agrega la ruta de esa carpeta a las variables de entorno fijándose que tenga permisos de ejecución (.exe) como se muestra en la Figura 10.

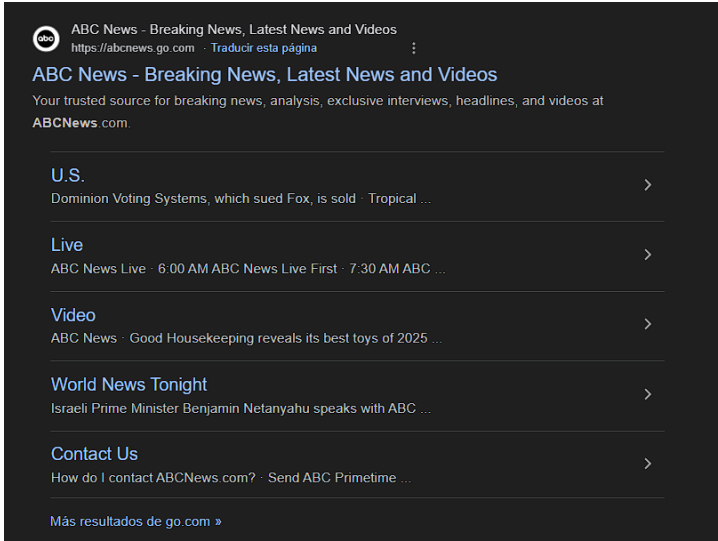


Fig. 19 *Página de ABC News en los resultados de búsqueda de Google.*

Nota. Captura de pantalla de una búsqueda realizada por el autor en Google (2025).

ABC News transmite en vivo, por lo cual podemos usar como una prueba simple la recepción de aquella emisión entrando a su página web (véase la Figura 19).

Para ello, es necesario abrir la página y analizar el tráfico de red, identificando dentro de las solicitudes la URL del archivo M3U8 como se observa en la Figura 20 y 21.

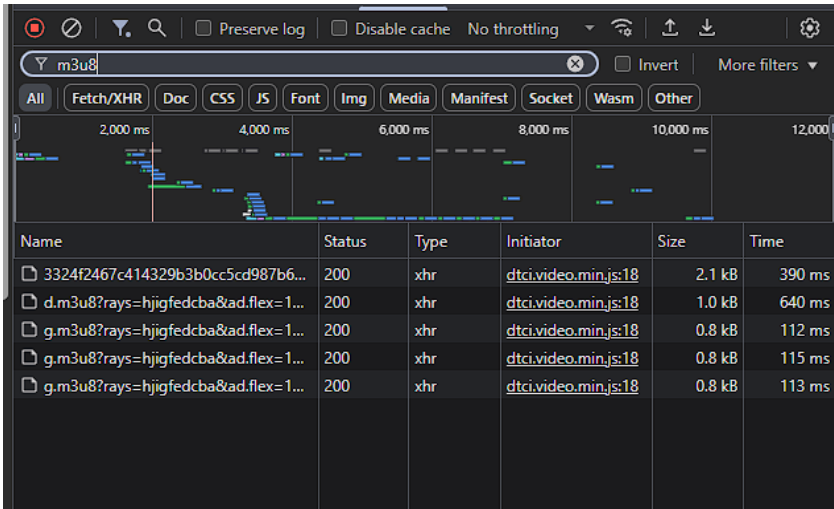


Fig. 20 *Herramientas para desarrolladores de Chrome en apartado Network.*

Nota. Captura de pantalla de inspección de tráfico en la página abcnews.go.com/Live por el autor en Google (2025).

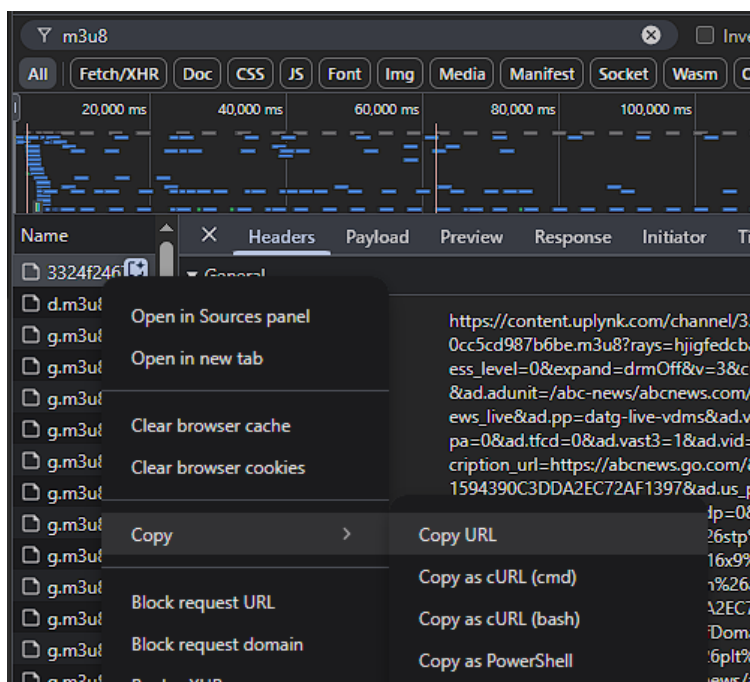


Fig. 21 Herramientas para desarrolladores de Chrome en apartado Network.

Nota. Captura de pantalla de archivo M3U8 tomado de ABC News (2025).

abcnews.go.com/Live

Y se puede verificar la recepción usando el comando `ffplay -i URL`, como se muestra en la Figura 22 y 23.



Fig. 22 Comando de reproducción de FFmpeg con link de archivo M3U8 como entrada.

Nota. Captura de pantalla de archivo M3U8 tomado de ABC News (2025).

abcnews.go.com/Live



Fig. 23 Comando de reproducción de FFmpeg con link de archivo M3U8 como entrada.

Nota. Captura de pantalla de reproducción del archivo M3U8 tomado de ABC News (2025). abcnews.go.com/Live

Se puede observar que se obtiene la reproducción de video correspondiente.

Ahora bien, teniendo el flujo de datos necesario se puede usar FFmpeg para generar una lista M3U8 que servirá para prueba del uso del protocolo HLS.

Creamos una carpeta y nos ubicamos en esa dirección dentro de la terminal, y procederemos a ejecutar el siguiente comando: `ffmpeg -i URL -hls master.m3u8`

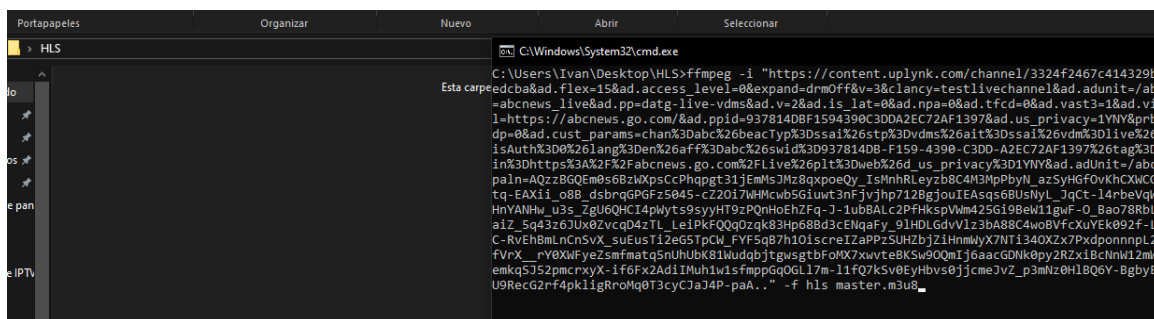


Fig. 24 Comando para generación de lista M3U8 con link de archivo M3U8 como entrada.

Nota. Captura de pantalla generación de archivo M3U8 con M3U8 tomado de ABC News (2025). abcnews.go.com/Live

En consecuencia al ejecutar se generará una lista M3U8 interminable hasta que se cancele el comando con la combinación de teclas Ctrl + C, véase la Figura 25.

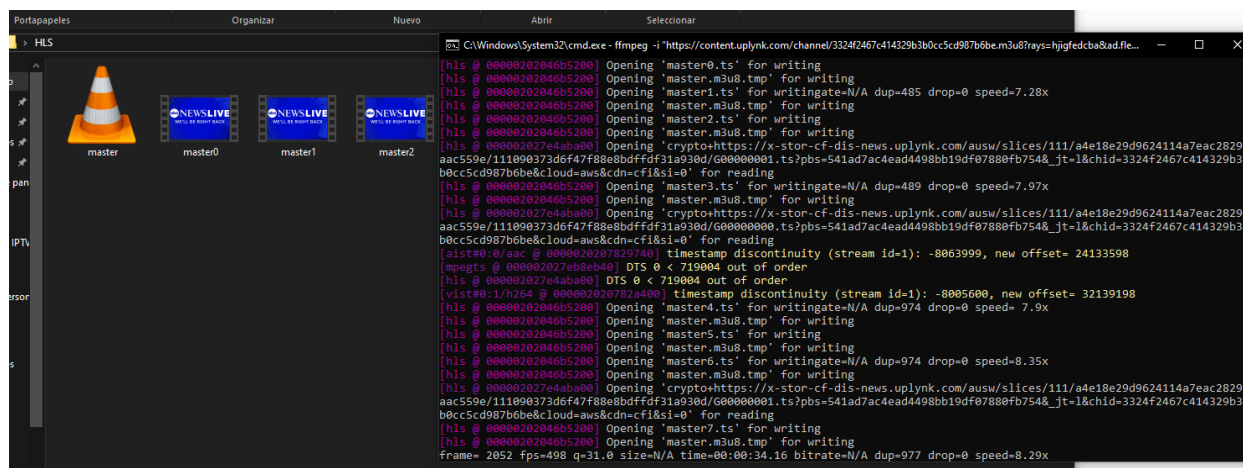


Fig. 25 Generación de lista M3U8 con link de archivo M3U8 como entrada.

Nota. Captura de pantalla generación de archivo M3U8 con M3U8 tomado de ABC News (2025). abcnews.go.com/Live

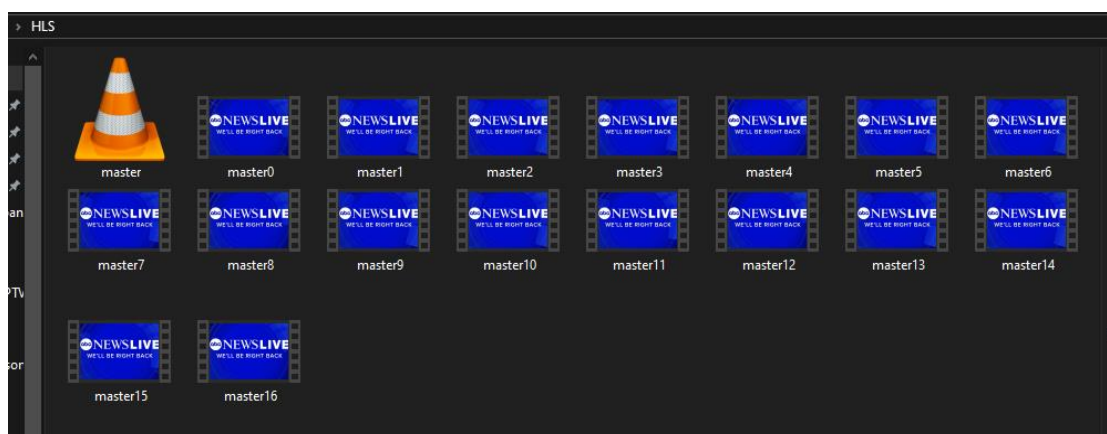


Fig. 26 Generación de lista M3U8 creada.

Nota. Captura de pantalla de generación de lista M3U8 generada con archivo M3U8 tomado de ABC News (2025). abcnews.go.com/Live

Quedándonos el archivo nombrado master el cual es la lista M3U8 donde contiene la información de reproducción de los segmentos .ts que se generaron. Al abrir la lista con VLC podremos observar el video generado como se muestra en la Figura 26.

Posteriormente se puede observar la reproducción del video al abrir la lista con el reproductor VLC.

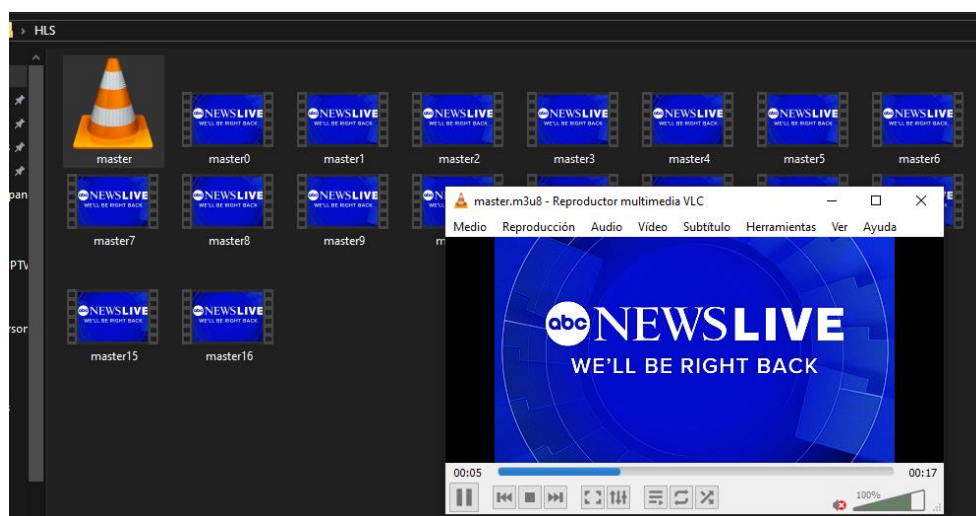


Fig. 27 Reproducción de lista M3U8 generada.

Nota. Captura de pantalla de reproducción de lista M3U8 generada con archivo M3U8 tomado de ABC News (2025). abcnews.go.com/Live

Dicho esto, la carpeta puede ser compartida por red y al acceder a la lista M3U8 podrá observar el video de la misma forma.

Al concluir la prueba, se observa que la herramienta funciona correctamente, lo que demuestra que esta opción es factible para su implementación.

Fase 3: Revisión y evaluación

Para la fase mencionada se hacen las observaciones pertinentes y se comienza a revisar posibles problemas que puedan surgir al usar este modo de procesamiento.

Primeramente, no se tiene control sobre el tamaño o bitrate de la transmisión pudiendo saturar el ancho de banda disponible en caso de no controlarse.

La lista M3U8 genera archivos de manera infinita pudiendo saturar el disco duro del equipo en donde se ejecute.

Es incierta la compatibilidad con todos los dispositivos a los que se requiera transmitir dado que no todos soportan ciertos formatos o codecs de vídeo.

No hay control sobre el contenido disponible como por ejemplo categorías, tipos de video como series, películas o tv en vivo.

No se tiene una forma fácil de acceder a los streams en caso de requerir más.

En caso de restricciones de usuario no hay ninguna ya que cualquiera que tenga acceso a la lista podría reproducirla.

Se sabe que el procesamiento de video es una labor bastante demandante para los equipos computacionales por lo que se requiere un equipo con la suficiente capacidad para poder procesar varios flujos de video.

En general el flujo de video es bueno, pero requiere ajustes y gestión para poder enviarlo sobre internet de la mejor manera.

Fase 4: Refinamiento

En la frase descrita se empieza a trabajar con el equipo y herramientas que mejor se adaptan al tipo de proyecto respecto a su funcionalidad.

Teniendo como referencia el prototipo inicial, se procede a desarrollar un nuevo prototipo que cumpla con los requerimientos y evite las posibles fallas anteriormente mencionadas.

En este punto se opta por implementar equipos dedicados a tareas específicas atendiendo las necesidades que anteriormente se mencionaron.

FFmpeg cuenta con parámetros que pueden controlar el flujo de bits, teniendo como opciones el bitrate constante y el bitrate variable. Esto asegura un aproximado que podemos estimar para controlar el tráfico dentro de la red.

FFmpeg tiene la posibilidad de controlar el bitrate haciéndolo predecible para los equipos ya que puede que algunos equipos no soportan el flujo variable de una emisión de video, este flujo depende de la complejidad de la escena que se esté transmitiendo, un claro ejemplo de complejidad son los canales de deportes que tienen mucho movimiento, cambio de luces, colores entre otras características que vuelven el video bastante pesado, por lo que controlar el flujo de bits se vuelve indispensable en este punto, se considera que pueden tener una configuración independiente dado que algunos canales como caricaturas que son trazos muy simples sin tanto movimiento pueden no requerir la limitación del bitrate tanto como otros.

A su vez FFMpeg tiene opciones en el protocolo HLS para evitar que haya demasiados archivos .ts sin control que saturen el disco de memoria, desde el nombre del segmento hasta el número y tiempo de estos mismos.

Se tiene que habilitar un equipo que sea capaz de convertir la señal recibida en un formato compatible con la mayoría de dispositivos, siendo este el formato H.264. Para lo cual se investigó que hay dos formas de hacer este proceso conocido como transcodificación. El pasar de un codec a otro es bastante pesado y se tiene la opción de hacerlo por CPU o GPU, la mejor forma de hacerlo sería con una gráfica dedicada como las de la marca NVIDIA dado que son líderes en procesamiento de video.

Las tarjetas de NVIDIA pueden hacer el proceso de transcodificación sin problema con sus núcleos dedicados, pruebas simples se pueden hacer con el procesador pero es necesario el poder de una unidad gráfica para sacar todo el provecho del video.

Se requiere de un software middleware que permita entre lo principal, administrar el contenido de manera eficiente para el espectador, y que permita limitar el acceso a usuarios no autorizados esto con el fin de mantener la seguridad del sistema y no gastar recursos innecesarios como ancho de banda o hardware de los equipos.

También se requiere un equipo que haga el proceso de distribución de datos a través de internet, en este caso debe ser un servidor dedicado. Puede tanto generar las listas M3U8 como tanto obtenerlas localmente y mandarlas a internet como podría dedicarse únicamente a generarlas y otro equipo recibir las listas para su distribución.

Fase 5: Iteración

Después de crear el modelo con los equipos necesarios para este proyecto, el equipo dedicado a la transcodificación se instaló con Windows 10 para hacer una prueba rápida del funcionamiento. Por lo que la instalación de drivers de la tarjeta fue bastante sencilla. El cual después de su funcionamiento se migra a Linux Ubuntu Server 24.04 LTS.

Se instaló FFmpeg y se recibió el flujo solo en una interfaz, se aplicaron las configuraciones necesarias para transcodificar a H.264 y bajar el bitrate, es flujo se envía y recibe localmente para comprobar el peso de la transferencia procesada, por lo que al hacer la prueba se observa que comprime el video y lo mantiene constante manteniendo la calidad.

En el caso del servidor de transmisión se instala directamente con Linux Ubuntu Server 24.04.2 LTS y se hace la prueba de comunicación con el equipo transcodificador, al mandar por la interfaz conectada sobre el mismo segmento a la IP se pudo recibir sin problema la señal.

A su vez se le instala Nginx de manera básica para poder comprobar que pueda servir los archivos, al observar la interfaz de Nginx desde otro computador conectado a la misma red que el servidor por donde reciben internet ambos, se concluye que puede servir de manera correcta.

Fase 6: Implementación final

Una vez validados los requisitos mediante el prototipo, se construye el sistema completo con la mejor calidad posible.

Se instala el workstation o equipo transcodificador con Ubuntu Server 24.04.2 LTS y se instalan los controladores de la tarjeta gráfica, para poder usar los núcleos CUDA para la hacer el cambio de códec de audio y video con la GPU y su aceleración por hardware de NVENC.

Instalación:

Primero se identifica la tarjeta gráfica del equipo puede ser con los comandos:

```
sudo update-pciids
```

`sudo lspci -v | less` (Con esto se puede ver el modelo de la tarjeta y se busca el .run en la página de drivers oficiales de NVIDIA).

Luego de encontrarlo se descarga quedando un archivo como se observa en la Figura 28.

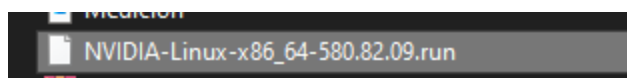


Fig. 28 Archivo de drivers de NVIDIA para tarjeta gráfica.

Nota. Captura de pantalla de archivo descargado de drivers de tarjeta gráfica tomado de NVIDIA (2025). www.NVIDIA.com

Puede enviarse al equipo por el comando SCP que sirve para transferir archivos de manera segura.

Se crea una carpeta ParaWS en el escritorio.

```
scp -r -P 22 C:\Users\Usuario\Desktop\ParaWS\* usuarioLinux@IP:/home/usuario
```

El ejemplo se puede observar en la Figura 29.

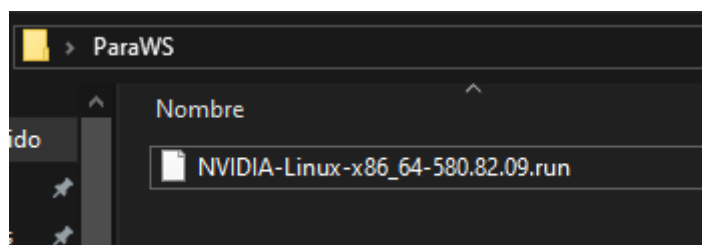


Fig. 29 Archivo de drivers de NVIDIA para tarjeta gráfica dentro de una carpeta.

Nota. Captura de pantalla de archivo descargado de drivers de tarjeta gráfica tomado de NVIDIA (2025). www.NVIDIA.com

Si se accede al equipo al que se mandó se observa en dicha dirección como se muestra en la Figura 30.

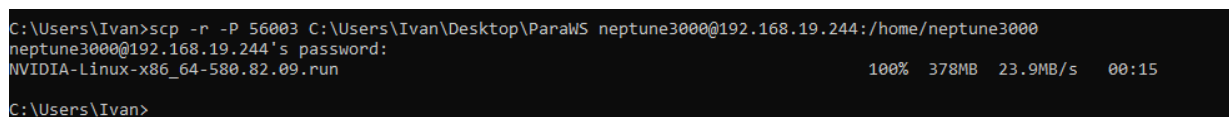


Fig. 30 Ejecución de comando con el protocolo SCP para envío de archivo.

Nota. Captura de pantalla realizada por el autor.

De manera que se envía al equipo el archivo dentro de la carpeta que corresponda, véase en la Figura 31.

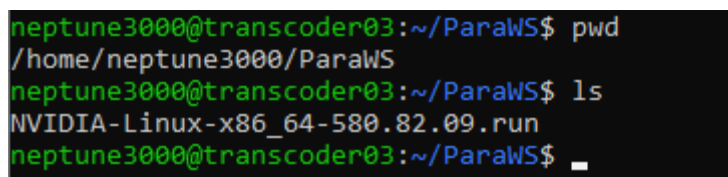


Fig. 31 Ruta donde se encuentra el archivo recibido en Linux.

Nota. Captura de pantalla realizada por el autor.

Al ser un archivo .run puede ejecutarse directamente en Linux con:

```
sh ./NVIDIA-Linux-x86_64-580.82.09.run
```

Posterior a la ejecución del comando muestra los pasos de la instalación seleccionando que queremos usar los gráficos de NVIDIA, véase en la Figura 32.

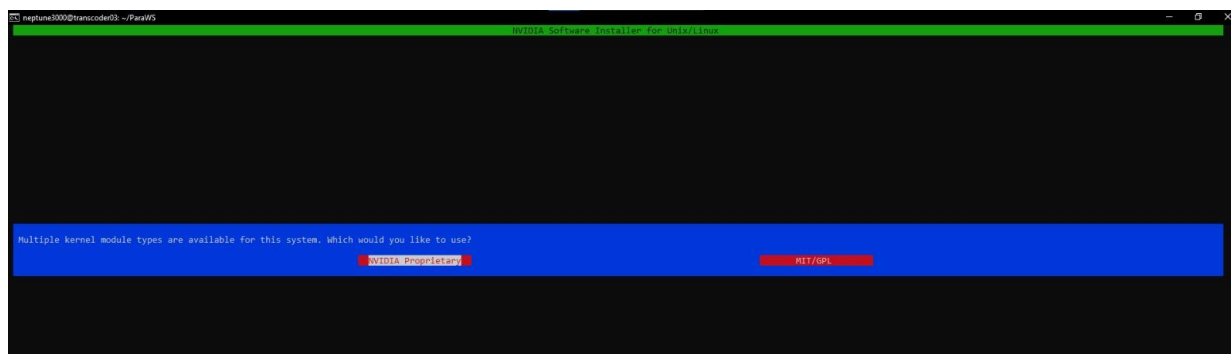


Fig. 32 Pantalla de selección del tipo de módulo del kernel durante la instalación de los drivers NVIDIA.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

Posteriormente se mostrarán una serie de advertencias, es importante conocer cómo afecta la elección que se toma en cada paso de la instalación.

Como se puede observar en la Figura 33 indica que hay una forma alternativa de instalación lo cual se debe ignorar y proceder con la instalación.

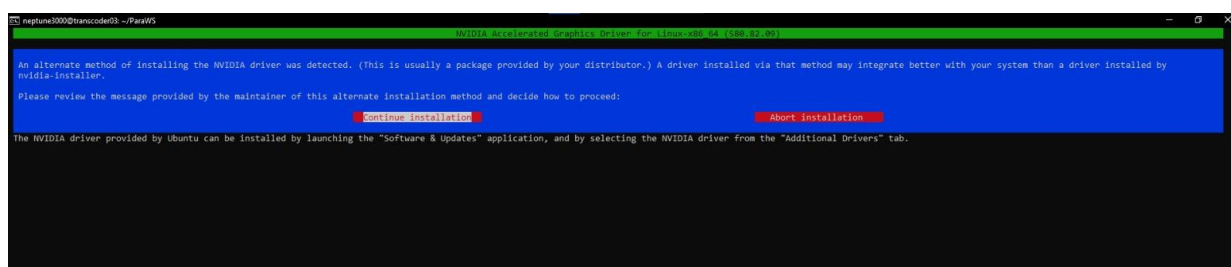


Fig. 33 Advertencia sobre un método alternativo de instalación de controladores NVIDIA detectado.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

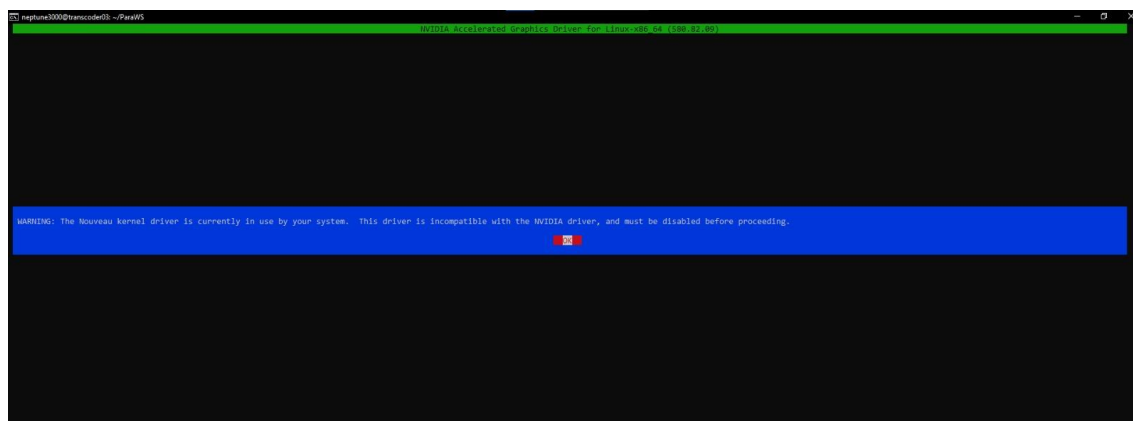


Fig. 34 *Advertencia sobre la presencia del controlador libre “Nouveau” durante la instalación del driver NVIDIA.*

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

En la advertencia que se observa en la Figura 34, se indica la presencia del controlador por defecto informando que se encuentra en uso y será deshabilitado para completar la instalación.

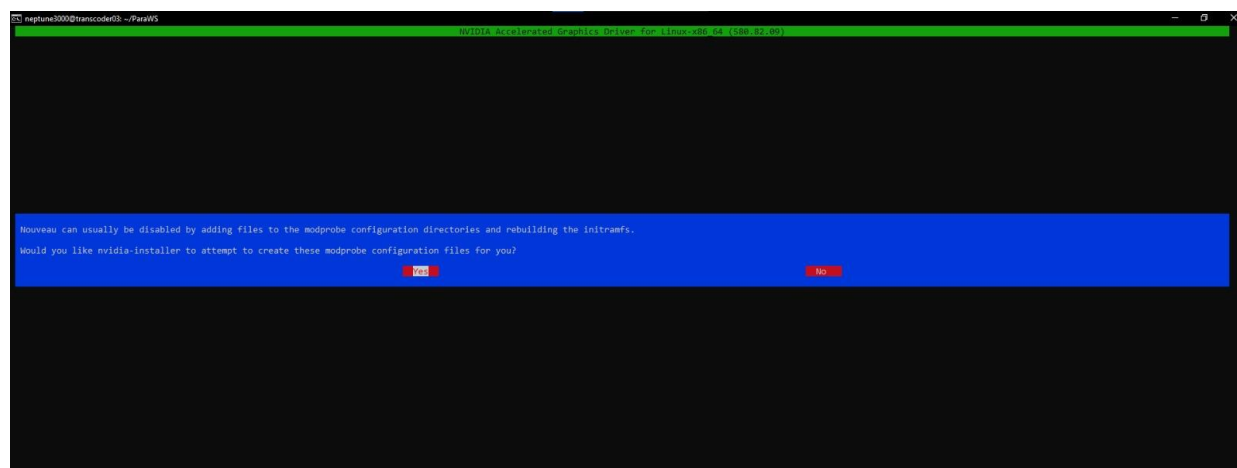


Fig. 35 *Mensaje del instalador de NVIDIA solicitando permiso para crear automáticamente los archivos de configuración que desactivan el controlador “Nouveau”.*

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

Se crean los archivos de configuración de modprobe para deshabilitar Nouveau que es el controlador por defecto. Y Rebuild del initramfs para que se apliquen al reinicio los cambios.

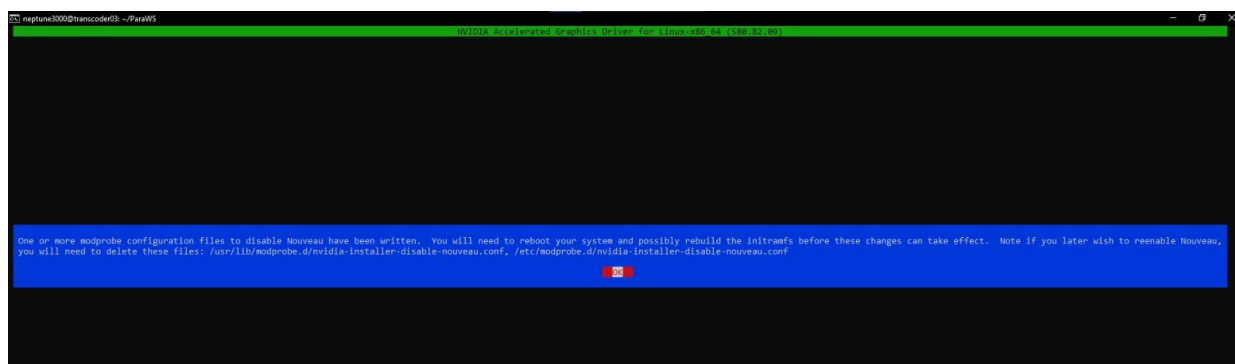


Fig. 36 Aviso del instalador de NVIDIA indicando la necesidad de reiniciar el sistema para aplicar la desactivación del controlador “Nouveau”.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

En la advertencia que se presenta en la Figura 36 se indica la necesidad de aplicar un reinicio para aplicar los cambios que deshabilitan el controlador por defecto.

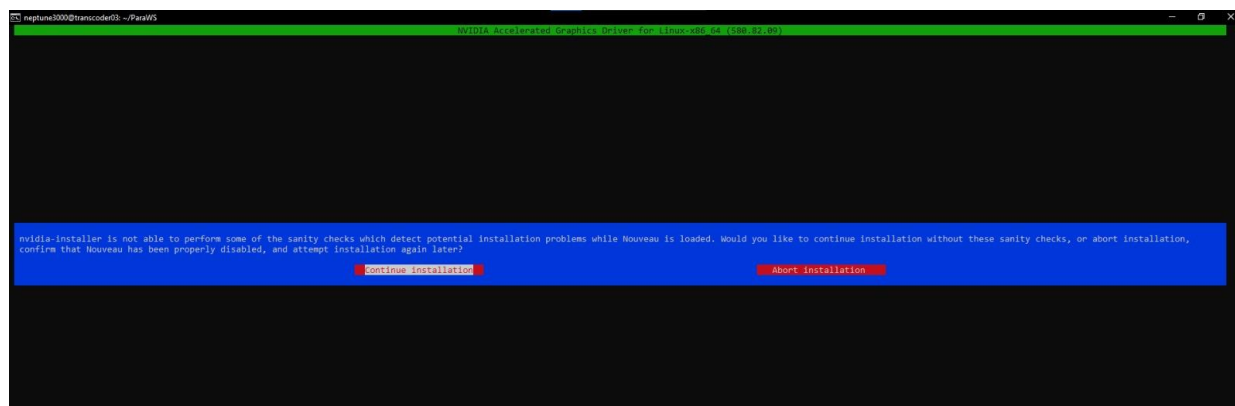


Fig. 37 Advertencia del instalador de NVIDIA sobre la imposibilidad de realizar verificaciones de compatibilidad (“sanity checks”).

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

Se indica que no se puede realizar verificaciones de compatibilidad, pero como el módulo Nouveau todavía se encuentra activo no puede garantizar que la instalación sea segura. Por lo que se recomienda verificar que el módulo Nouveau haya sido deshabilitado apropiadamente antes de proceder.

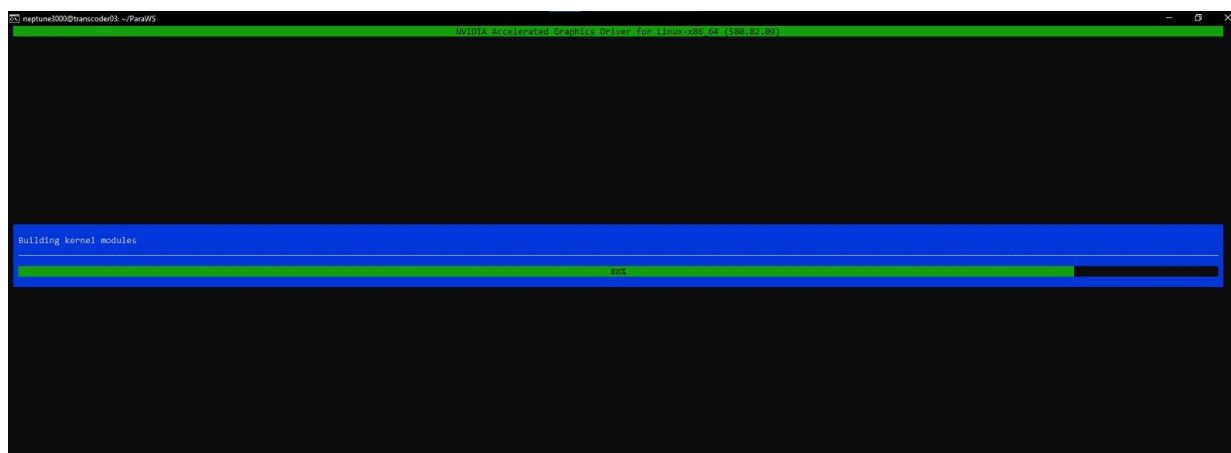


Fig. 38 *Proceso de construcción de los módulos del kernel.*

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

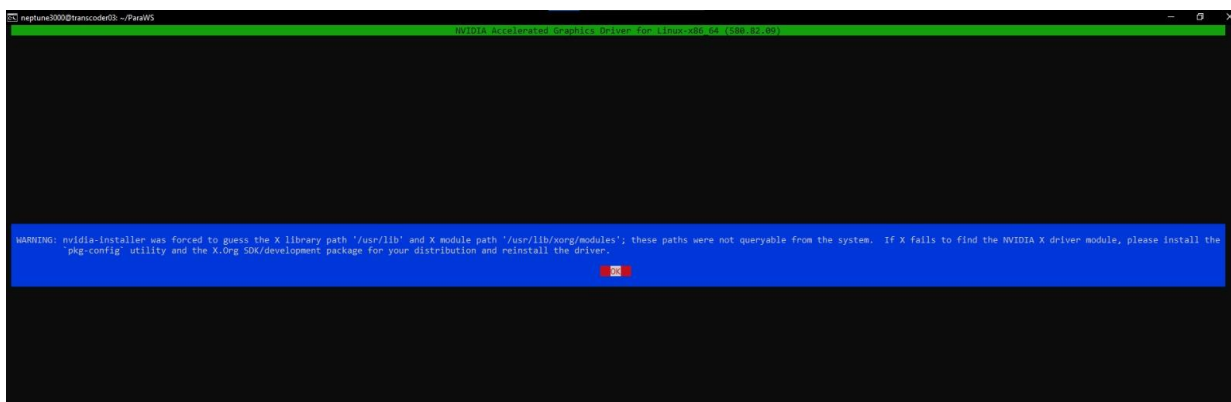


Fig. 39 *Advertencia del instalador de NVIDIA sobre la detección automática de las rutas de bibliotecas del servidor gráfico X.*

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

La advertencia mostrada en la Figura 39 puede resolverse con: `sudo apt install pkg-config xserver-xorg-dev` ya que NVIDIA no puede detectar automáticamente dónde se encuentra Xorg en el sistema el cual es un estándar para la gestión de gráficos.

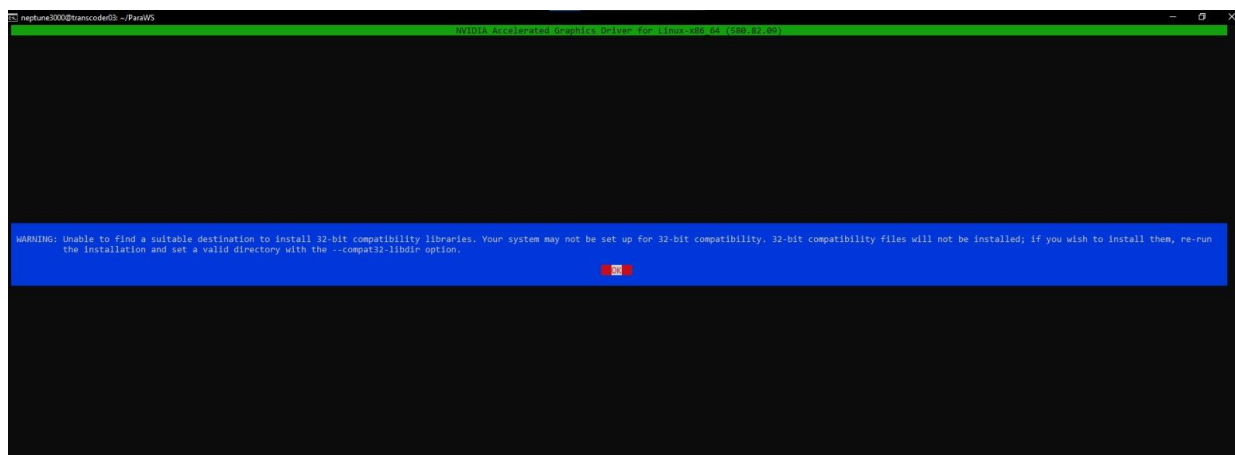


Fig. 40 Advertencia del instalador de NVIDIA sobre la falta de una ruta adecuada para instalar las bibliotecas de compatibilidad de 32 bits.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

Dicha advertencia mostrada en la Figura 40 se ignora si solo queremos que NVIDIA funcione en Linux de 64 bits.

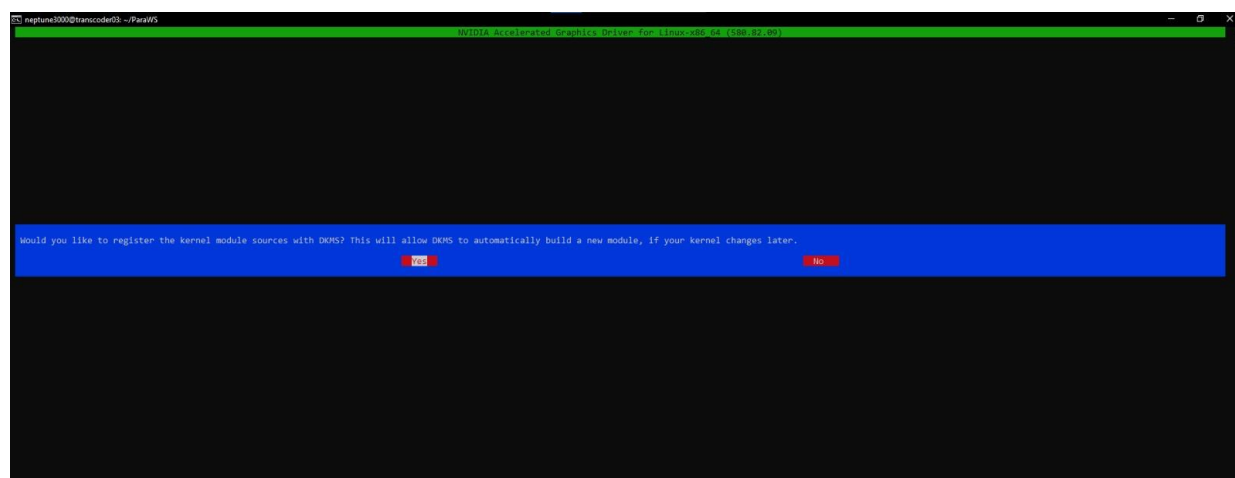


Fig. 41 Mensaje del instalador de NVIDIA ofreciendo registrar el módulo del kernel con DKMS para recompilación automática.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

La advertencia que se muestra en la Figura 41 indica que DKMS es un sistema que recompila los módulos del kernel automáticamente cuando este se actualiza. Evitando que cada vez que se actualice el sistema se tenga que volver a ejecutar el procedimiento manualmente.

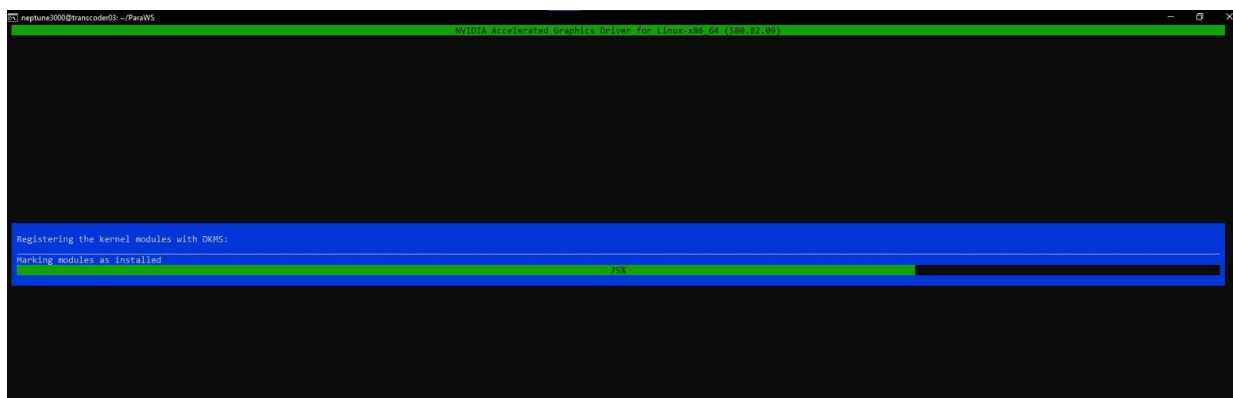


Fig. 42 Progreso de registro de los módulos del kernel con DKMS.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

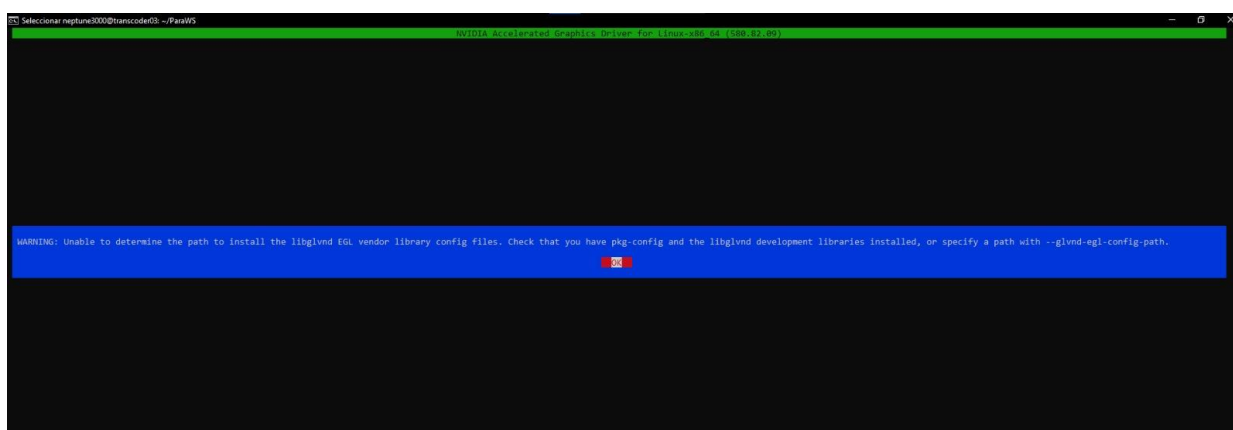


Fig. 43 Advertencia del instalador de NVIDIA al no detectar la ruta de instalación de las librerías EGL de libglvnd.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

La advertencia que se muestra en la Figura 43 se puede solucionar ejecutando el comando `sudo apt install pkg-config libglvnd-dev` ya que al igual que lo anterior no se pudo determinar dónde se colocan los archivos de configuración de EGL para NVIDIA, libglvnd es una biblioteca que permite que Linux use múltiples implementaciones de OpenGL/EGL en paralelo. Esto causa problemas si alguna aplicación busca específicamente esos archivos de configuración y no los encuentra y es relativamente seguro de ignorar.

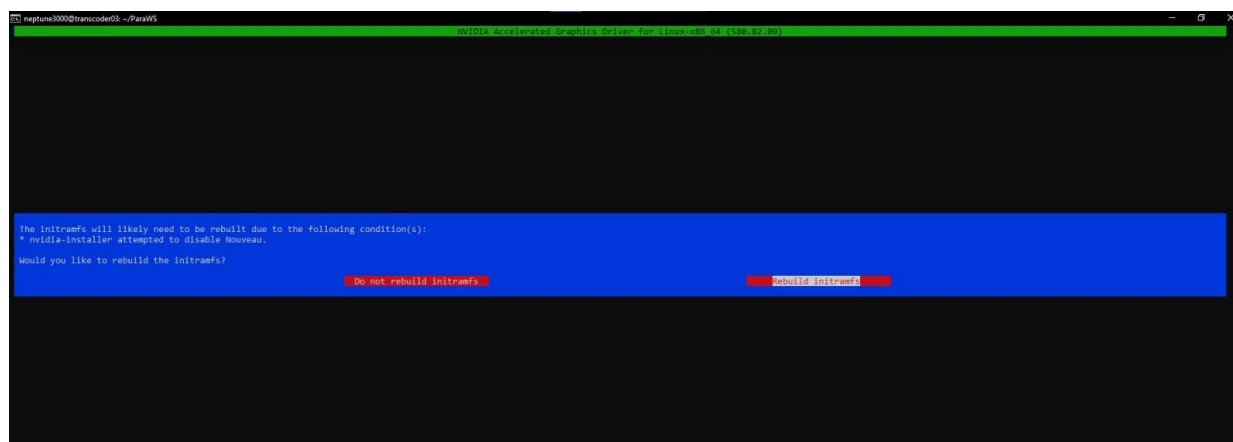


Fig. 44 Mensaje del instalador de NVIDIA solicitando la reconstrucción del initramfs tras deshabilitar el controlador Nouveau.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

Aquí se deshabilitó Nouveau pero para que el cambio surta efecto se tiene que reconstruir el initramfs, la cual es la imagen que carga los módulos al arrancar Linux.

Se selecciona Rebuild initramfs para que actualice la imagen del arranque con Nouveau deshabilitado y permitirá que el driver de NVIDIA funcione correctamente.

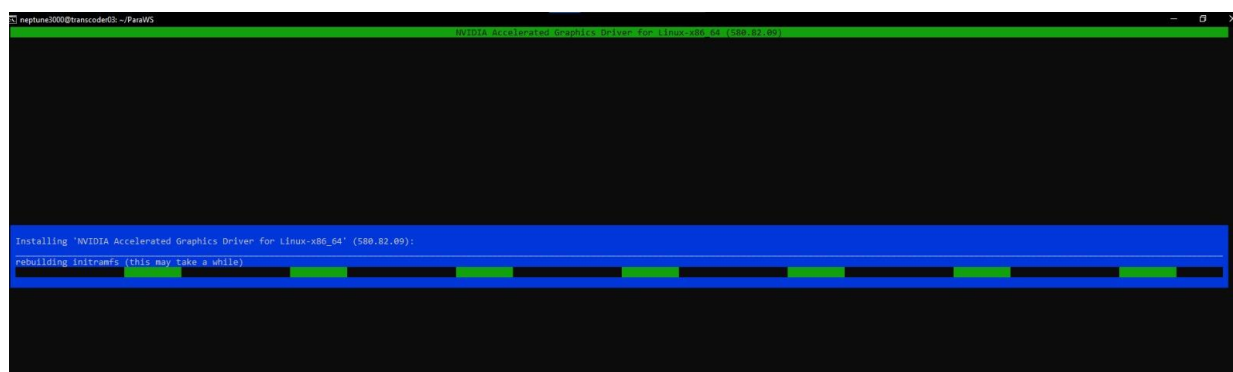


Fig. 45 Proceso de reconstrucción de initramfs.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

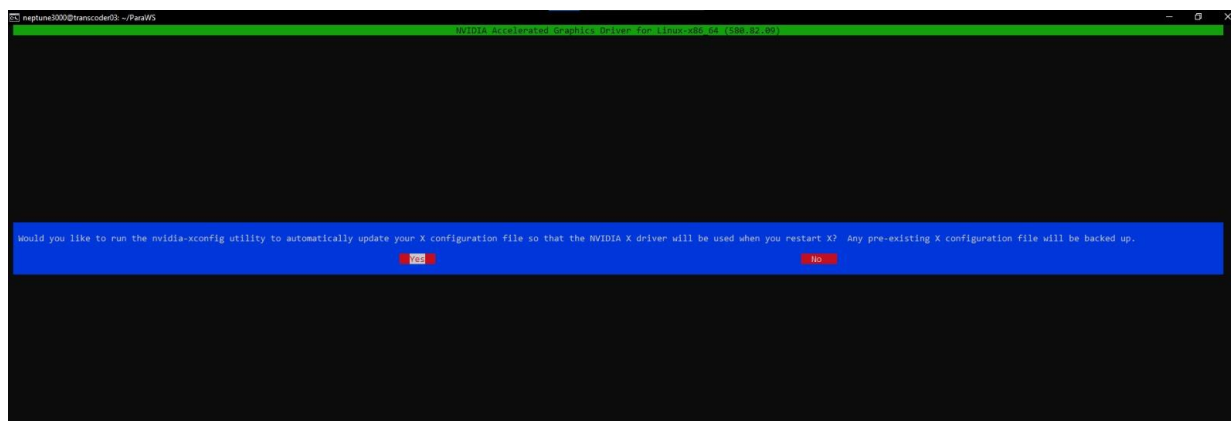


Fig. 46 Mensaje del instalador de NVIDIA ofreciendo ejecutar la utilidad NVIDIA-xconfig para actualizar automáticamente el archivo de configuración del servidor gráfico Xorg.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

En la Figura 46 se muestra la opción de ejecutar la utilidad que actualiza automáticamente el archivo de configuración del servidor gráfico Xorg. Se selecciona Yes para que el driver de NVIDIA sea usado automáticamente en el próximo inicio de sesión gráfica, ya que se actualiza el archivo de configuración de Xorg. Y cualquier archivo de configuración existente se respaldará automáticamente así no se pierde la configuración actual.

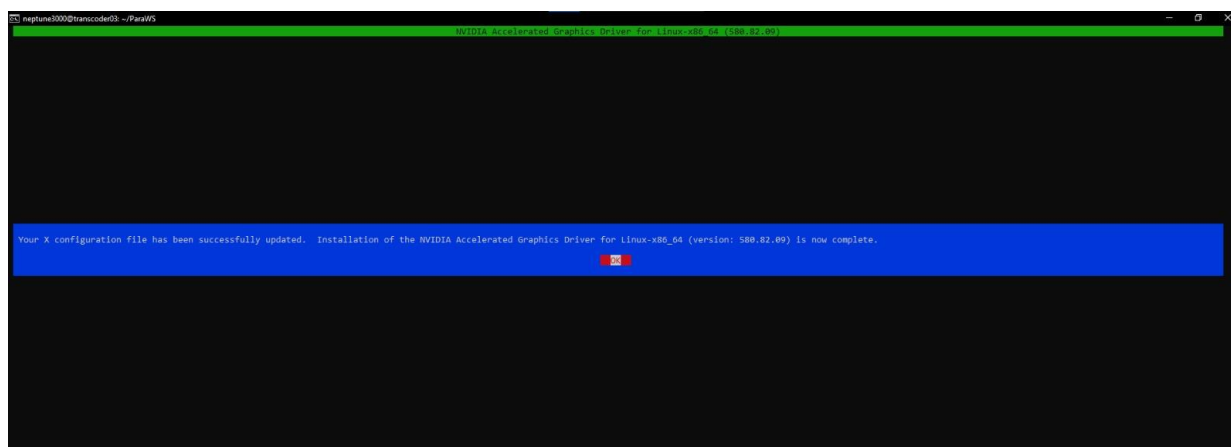


Fig. 47 Mensaje final del instalador de NVIDIA confirmando la correcta actualización de la configuración de Xorg y la finalización del proceso de instalación.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

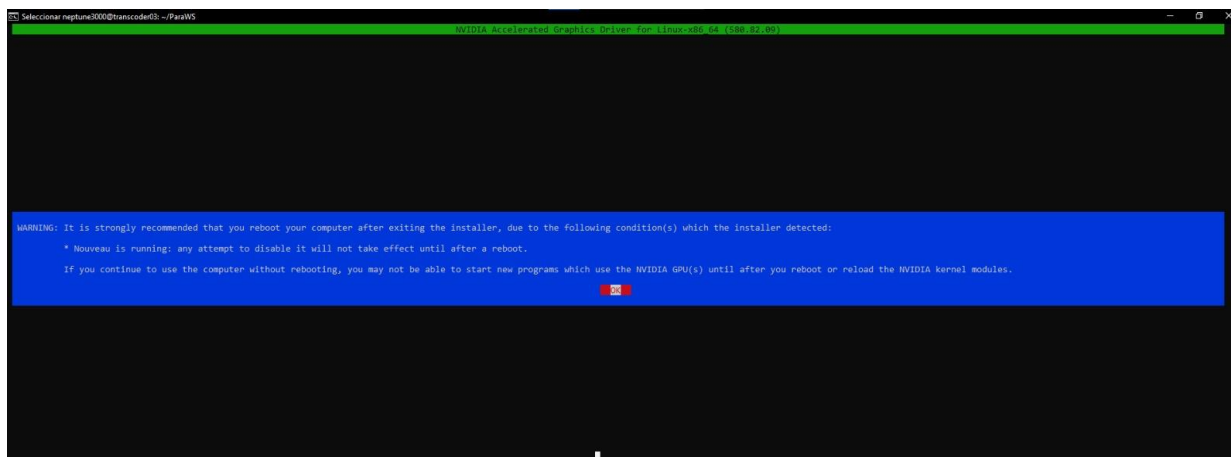


Fig. 48 *Mensaje final del instalador de NVIDIA recomendando reiniciar el sistema para aplicar los cambios del controlador gráfico.*

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

Al terminar la instalación se tiene que reiniciar el equipo con: `sudo reboot`

Puede que al verificar con el comando `nvidia-smi` salga este mensaje: NVIDIA-SMI has failed because it couldn't communicate with the NVIDIA driver. Make sure that the latest NVIDIA driver is installed and running.
`a-smi`

Puede que el kernel sea muy nuevo y el driver que se instaló no lo soporta, el módulo no se compila por lo que lo más seguro es usar el driver más reciente disponible desde PPA oficial de Ubuntu Server en lugar del `.run` de NVIDIA. Asegurando la compatibilidad del kernel.

Se añade el PPA de drivers gráficos de la siguiente manera con los comandos:
`sudo add-apt-repository ppa:graphics-drivers/ppa`
`sudo apt update`

Y se instala el driver más reciente compatible por ejemplo el 535 o 550 según la disponibilidad.

`sudo apt install NVIDIA-driver-535`

Al volver a iniciar se puede verificar la instalación de los drivers con: nvidia-smi

```
neptune3000@transcoder03:~$ nvidia-smi
Fri Sep 26 23:38:50 2025
```

NVIDIA-SMI 535.247.01			Driver Version: 535.247.01			CUDA Version: 12.2		
GPU	Name	Perf	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC	
Fan	Temp		Pwr:Usage/Cap		Memory-Usage	GPU-Util	Compute M.	
							MIG M.	
0	NVIDIA GeForce RTX 4080	P8	Off	00000000:01:00.0	Off	0%	N/A	
31%	32C		12W / 320W		1MiB / 16376MiB		Default	
							N/A	

```

Processes:
GPU  GI  CI  PID  Type  Process name  GPU Memory
ID   ID  ID              Usage
-----
No running processes found
neptune3000@transcoder03:~$
```

Fig. 49 Salida del comando NVIDIA-smi mostrando el estado de la GPU y confirmando el funcionamiento del controlador propietario de NVIDIA.

Nota. Captura de pantalla realizada por el autor durante el proceso de instalación de drivers.

Como se observa en la Figura 49 se concluye la instalación con éxito para poder aprovechar los núcleos CUDA de NVIDIA dedicados.

Una parte importante es verificar el número de sesiones concurrentes de la tarjeta gráfica, esto se traduce en el número máximo de transcodificaciones que puede hacer simultáneamente.

Se puede verificar en Video Encode and Decode Support Matrix página que muestra las capacidades de las diferentes tarjetas gráficas de NVIDIA.

↓ BOARD	↓ FAMILY	↓ NVENC Generation	Desktop/ Mobile	# OF CHIPS	Total # of NVENC	Max # of concurrent sessions	H.264 (AVC) YUV 4:2:0	H.264 (AVC) YUV 4:2:2	H.264 (AVC) YUV 4:4:4	H.264 (AVC) Lossless	H.265 (HEVC) 4K YUV 4:2:0
GeForce RTX 4080 16GB	Ada Lovelace	8th Gen	D	1	2	8	YES	NO	YES	YES	YES

Fig. 50 Información de capacidad de codificación y decodificación de la gráfica utilizada en el ejemplo.

Nota. Captura de pantalla de tabla de matriz de compatibilidad de codificación y decodificación de vídeo de tarjetas NVIDIA tomado de Video Encode and Decode Support Matrix (2025).

developer.NVIDIA.com/video-encode-decode-support-matrix

En la parte de Max # of concurrent sessions que se muestra en la Figura 50 se puede observar el número máximo de sesiones simultáneas. Pero hay gráficas que esto no es un impedimento como las Quadro, véanse en las Figura 51 y 52.

Professional (NVIDIA RTX / Quadro)

Fig. 51 Sección de tarjetas Quadro de NVIDIA.

Nota. Captura de pantalla de tarjetas de NVIDIA Quadro tomado de Video Encode and Decode. Support Matrix (2025). developer.NVIDIA.com/video-encode-decode-support-matrix

NVIDIA RTX A4000	Ampere	7th Gen	D	1	1	Unrestricted	YES	NO	YES	YES
------------------	--------	---------	---	---	---	--------------	-----	----	-----	-----

Fig. 52 Información de capacidad de codificación y decodificación de una gráfica Quadro NVIDIA RTX A4000.

Nota. Captura de pantalla de tabla de matriz de compatibilidad de codificación y decodificación de vídeo de tarjeta NVIDIA Quadro tomado de Video Encode and Decode Support Matrix (2025).

developer.NVIDIA.com/video-encode-decode-support-matrix

Se observa que algunas gráficas tienen Unrestricted, básicamente limitada al hardware del equipo siendo de las mejores para este tipo de aplicaciones.

Una vez instalada la gráfica se procede a configurar las interfaces para recibir y enviar el tráfico multicast.

Como se muestra en la Figura 53, podemos recibir tráfico de red que es generado por un equipo encoder en la dirección 224.2.2.2 el cual se recibe en la interfaz enp2s0f3.

Se puede instalar una herramienta llamada smcroute que sirve para controlar el enrutamiento multicast estático, véase en la Figura 54 o bien se pueden definir las rutas estáticas en las interfaces como se puede observar en la Figura 53.

```
neptune3000@transcoder03:/etc/netplan$ sudo cat 50-cloud-init.yaml
[sudo] password for neptune3000:
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp2s0f3:
      dhcp4: false
      addresses: [192.169.0.2/24]
    enp2s0f2:
      dhcp4: false
      addresses: [192.169.101.2/24]
      nameservers:
        addresses: [192.169.101.1]
      routes:
        - to: 224.2.2.4/32
          via: 192.169.101.1
```

Fig. 53 Configuración de archivo yaml de interfaces de red.

Nota. Captura de pantalla realizada por el autor mostrando el archivo de configuración de red en Ubuntu Server.

enp2s0f3 recibe el tráfico 224.2.2.2 se usa con smcroute. Dicha herramienta se instala con el comando `sudo apt install smcroute`


```
neptune3000@transcoder03:/etc/netplan$ sudo cat /etc/smcroute.conf
phyint enp2s0f3 enable
mgroup from enp2s0f3 group 224.2.2.2
```

Fig. 54 Configuración de archivo “smcroute.conf” para enrutar tráfico multicast sobre una interfaz.

Nota. Captura de pantalla realizada por el autor mostrando el contenido del archivo de configuración.

enp2s0f2 recibe el tráfico 224.2.2.4 y se define en la interfaz modificando el archivo de configuración de red, pero es propenso a desconexión.

Ambas formas funcionan, queda a criterio del lector la aplicación.

Se comprueba que se reciba tráfico en las interfaces. Se puede usar la herramienta ifstat.

enp2s0f2				enp2s0f3			
KB/s	in	KB/s	out	KB/s	in	KB/s	out
65512.50		0.00		103700.5		0.00	
65542.80		0.00		103748.1		0.00	
65614.42		0.00		103960.5		0.00	
65625.16		0.00		103882.4		0.00	

Fig. 55 Tráfico sobre interfaces de red.

Nota. Captura de pantalla realizada por el autor mostrando el tráfico de red sobre las interfaces con la herramienta ifstat.

Ahora se instala la herramienta FFmpeg la cual sirve entre muchas cosas para verificar las entradas de vídeo de tráfico UDP multicast.

Se instala con: `sudo apt install ffmpeg`

Usando el comando `ffprobe -i [protocolo://IP:Puerto]`

```

neptune3000@transcoder03:/etc/netplan$ ffprobe -i udp://224.2.2.2:10001
ffprobe version 6.1.1-3ubuntu5 Copyright (c) 2007-2023 the FFmpeg developers
  built with gcc 13 (Ubuntu 13.2.0-23ubuntu3)
  configuration: --prefix=/usr --extra-version=3ubuntu5 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir
=/usr/include/x86_64-linux-gnu --arch=amd64 --enable-gpl --disable-stripping --disable-omx --enable-gnutls --enable-liba
om --enable-libass --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libdav1d --enable-libf
ite --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libglslang --enable-libgme --enable-libgsm
--enable-libharfbuzz --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus -
--enable-librubberband --enable-libshine --enable-lisnappy --enable-libsoxr --enable-lispeex --enable-libtheora --enabl
e-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwebp --enable-libx265 --enable-libxml2 -
--enable-libxvid --enable-libzimg --enable-openal --enable-opengl --enable-opengl --disable-sndio --enable-libvpl --disab
le-libmfx --enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --enable-frei0r --enable-ladspa -
--enable-libbluray --enable-libjack --enable-libpulse --enable-librabbitmq --enable-librist --enable-lisrt --enable-lis
sh --enable-lisvta --enable-libx264 --enable-libzmq --enable-libzvb --enable-lv2 --enable-sdl2 --enable-libplacebo -
--enable-librav1e --enable-pocketsphinx --enable-lisrsvg --enable-libjxl --enable-shared
  libavutil      58. 29.100 / 58. 29.100
  libavcodec     60. 31.102 / 60. 31.102
  libavformat    60. 16.100 / 60. 16.100
  libavdevice    60.  3.100 / 60.  3.100
  libavfilter     9. 12.100 /  9. 12.100
  libswscale     7.  5.100 /  7.  5.100
  libswresample  4. 12.100 /  4. 12.100
  libpostproc    57.  3.100 / 57.  3.100
[mpeg2video @ 0x5c5f90d694c0] Invalid frame dimensions 0x0.
Last message repeated 12 times

```

Fig. 56 Prueba de recepción de señal de video.

Nota. Captura de pantalla realizada por el autor mostrando la prueba de recepción de video con la herramienta ffprobe de FFmpeg.

```

Last message repeated 12 times
Input #0, mpegts, from 'udp://224.2.2.2:10001':
  Duration: N/A, start: 49231.576789, bitrate: 4692 kb/s
  Program 1003
  Metadata:
    service_name      : ESTRELLAS
    service_provider : TV-Provider
  Stream #0:0[0x27]: Video: mpeg2video (Main) ([2][0][0][0] / 0x0002), yuv420p(tv, bottom first), 720x480 [SAR 8:9 DAR 4
:3], Closed Captions, 4500 kb/s, 29.97 fps, 29.97 tbr, 90k tbn
  Side data:
    cpb: bitrate max/min/avg: 4500000/0/0 buffer size: 1835008 vbv_delay: N/A
  Stream #0:1[0x28]: Audio: ac3 ([129][0][0][0] / 0x0081), 48000 Hz, stereo, fltp, 192 kb/s

```

Fig. 57 Metadatos obtenidos de la recepción de señal de video.

Nota. Captura de pantalla realizada por el autor mostrando la prueba de recepción de video con la herramienta ffprobe de FFmpeg.

Se puede observar que la emisión se recibe correctamente, así como la información asociada a los datos obtenidos durante la inspección con la herramienta ffprobe, como se muestra en las Figuras 56 y 57.

Ahora se requiere definir la interfaz de salida de video procesado.

```
enp2s0f0:
  dhcp4: false
  addresses: [192.168.12.2/24]
  nameservers:
    addresses: [192.168.12.1]
  routes:
    - to: 224.0.0.8/32
      scope: link
```

Fig. 58 Configuración de interfaz de salida del tráfico multicast procesado.

Nota. Captura de pantalla realizada por el autor mostrando la configuración de la interfaz de salida de tráfico multicast modificando el archivo .yaml de red.

En este caso será la enp2s0f0 donde se asigna la dirección estática y en la ruta se define una ruta estática hacia la dirección multicast 224.0.0.8/32 máscara /32 (una única IP).

En el caso de scope: link significa que esa ruta no pasa por un gateway, sino que se trata como directamente alcanzable en el enlace local (la propia red conectada a la interfaz).

En resumen, al poner una ruta estática scope: link se le indica al sistema que si quiere enviar tráfico multicast al grupo 224.0.0.8 debe hacerlo por la interfaz directamente, sin gateway.

Posteriormente se crea un servicio en segundo plano o también conocido como demonios de Linux, sirven para ejecutar servicios en segundo plano.

Se encuentran en la ruta: /etc/systemd/system

Con el comando `sudo nano nombredelarchivo.service` se crea un archivo vacío de tipo .service donde se ingresa el proceso correspondiente.

En este caso se creó un servicio de prueba donde se recibe la señal UDP, se procesa usando comandos de FFmpeg que trabajan en conjunto con NVIDIA para transcodificar el video.

Usando los núcleos dedicados de NVIDIA en conjunto con FFmpeg. Recordando que estaba en MPEG-2 el cual es un códec de video antiguo.

Nota: Se puede hacer la transcodificación de MPEG-2 a H.264 pero no al revés ya que NVIDIA no tiene soporte para MPEG-2 por lo mismo de que es un códec antiguo.

```
neptune3000@transcoder03:/etc/systemd/system$ sudo cat prueba.service
[Unit]
Description=Servicio ffmpeg para IPTV prueba Ligero
After=network.target

[Service]
Type=simple
ExecStart=/usr/bin/ffmpeg -fflags +genpts+discardcorrupt -i "udp://224.2.2.2:10002" -c:v h264_nvenc -c:a aac -vf yadif=1 -aspect 16:9 -f mpegts "udp://224.0.0.8:40000?pkt_size=1316"
Restart=always
RestartSec=5
User=root
WorkingDirectory=/var/www/udp_simple/prueba
Environment=HOME=/root

[Install]
WantedBy=multi-user.target
neptune3000@transcoder03:/etc/systemd/system$
```

Fig. 59 Creación de servicio de procesamiento de video con systemd y FFmpeg.

Nota. Captura de pantalla realizada por el autor mostrando la configuración de un archivo .service donde se usan los parámetros de FFmpeg para transcodificar la señal y enviarla.

Se puede observar en la Figura 59 la estructura básica de un servicio en segundo plano en Linux.

Dicho servicio ejecuta el programa FFmpeg, descarta los paquetes corruptos, recibe la transmisión UDP multicast, transcodifica el video a el códec H.264 y además transcodifica el audio a el códec AAC. Desentrelaza la señal y duplica los frames. Modifica la relación aspecto y manda la salida por UDP a la dirección que de salida por la interfaz enp2s0f0.

Se crea la carpeta del WorkingDirectory ya que define la carpeta desde la cual se ejecutará el servicio. Sirve para que rutas relativas en scripts o binarios funcionen bien, y todo se ejecute en un contexto adecuado ya que de otra forma se usaría la raíz como carpeta base.

Posterior a eso, se crean los servicios necesarios como se observa en la Figura 60.

```
neptune3000@transcoder03:/etc/systemd/system$ ls prueba*
prueba1.service prueba2.service prueba3.service prueba4.service prueba5.service prueba6.service prueba7.service prueba8.service prueba9.service prueba.service
```

Fig. 60 Servicios creados para transcodificación de video.

Nota. Captura de pantalla realizada por el autor mostrando los archivos generados de diferentes señales.

Una vez creados se ejecutan los siguientes comandos para iniciar.

`sudo systemctl daemon-reload` (systemd vuelve a leer y a actualizar la configuración).

`sudo systemctl start prueba.service` (inicia el servicio, si se modifica entonces se usa `restart` y si se quiere parar se usa `stop`).

Se puede observar que se ejecutan correctamente si el estado dice `running` como se muestra en la Figura 61.

```
neptune3000@transcoder03:/etc/systemd/system$ sudo systemctl status prueba*
[sudo] password for neptune3000:
* prueba1.service - Servicio ffmpeg para IPTV prueba1 Ligero
  Loaded: loaded (/etc/systemd/system/prueba1.service; disabled; preset: enabled)
  Active: active (running) since Thu 2025-10-02 06:45:38 UTC; 6 days ago
    Main PID: 44795 (ffmpeg)
      Tasks: 55 (limit: 76708)
     Memory: 70.1M (peak: 74.3M)
        CPU: 1d 14h 38min 41.583s
    CGroup: /system.slice/prueba1.service
            └─44795 /usr/bin/ffmpeg -fflags +genpts+discardcorrupt -i udp://224.2.2.4:10040 -c:v h264_nvenc -c:a aac -vf yadif=1 -aspect 16:9 -f mpegts "udp://224.0.0.8:4000?pkt_size=1316"

Oct 08 19:03:52 transcoder03 ffmpeg[44795]: [48.0K blob data]
Oct 08 19:07:26 transcoder03 ffmpeg[44795]: [47.9K blob data]
Oct 08 19:11:01 transcoder03 ffmpeg[44795]: [48.0K blob data]
Oct 08 19:14:35 transcoder03 ffmpeg[44795]: [48.0K blob data]
Oct 08 19:18:10 transcoder03 ffmpeg[44795]: [48.0K blob data]
Oct 08 19:21:45 transcoder03 ffmpeg[44795]: [48.0K blob data]
Oct 08 19:25:19 transcoder03 ffmpeg[44795]: [48.0K blob data]
Oct 08 19:28:53 transcoder03 ffmpeg[44795]: [48.0K blob data]
Oct 08 19:32:28 transcoder03 ffmpeg[44795]: [48.0K blob data]
Oct 08 19:36:02 transcoder03 ffmpeg[44795]: [47.9K blob data]

* prueba2.service - Servicio ffmpeg para IPTV prueba2 Ligero
  Loaded: loaded (/etc/systemd/system/prueba2.service; disabled; preset: enabled)
  Active: active (running) since Mon 2025-10-06 17:56:23 UTC; 2 days ago
    Main PID: 76815 (ffmpeg)
      Tasks: 55 (limit: 76708)
     Memory: 69.6M (peak: 76.3M)
        CPU: 12h 19min 7.125s
    CGroup: /system.slice/prueba2.service
            └─76815 /usr/bin/ffmpeg -fflags +genpts+discardcorrupt -i udp://224.2.2.2:10001 -c:v h264_nvenc -c:a aac -aspect 16:9 -vf yadif=1 -f mpegts "udp://224.0.0.8:4000?pkt_size=1316"

Oct 08 18:58:24 transcoder03 ffmpeg[76815]: [47.9K blob data]
Oct 08 19:02:33 transcoder03 ffmpeg[76815]: [48.0K blob data]
Oct 08 19:06:43 transcoder03 ffmpeg[76815]: [47.9K blob data]
Oct 08 19:10:53 transcoder03 ffmpeg[76815]: [48.0K blob data]
Oct 08 19:15:03 transcoder03 ffmpeg[76815]: [48.0K blob data]
Oct 08 19:19:13 transcoder03 ffmpeg[76815]: [48.0K blob data]
Oct 08 19:23:23 transcoder03 ffmpeg[76815]: [48.0K blob data]
Oct 08 19:27:32 transcoder03 ffmpeg[76815]: [48.0K blob data]
Oct 08 19:31:43 transcoder03 ffmpeg[76815]: [48.0K blob data]
Oct 08 19:35:53 transcoder03 ffmpeg[76815]: [48.0K blob data]

* prueba3.service - Servicio ffmpeg para IPTV prueba3 Ligero
  Loaded: loaded (/etc/systemd/system/prueba3.service; disabled; preset: enabled)
  Active: active (running) since Thu 2025-10-02 06:45:38 UTC; 6 days ago
    Main PID: 44784 (ffmpeg)
```

Fig. 61 Estado de los servicios creados luego de ejecutarse.

Nota. Captura de pantalla realizada por el autor del estado de los servicios.

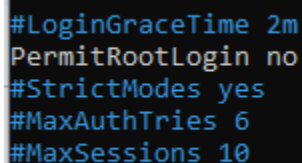
Opcionalmente se puede habilitar con `systemctl enable prueba.service` para que cada que encienda la máquina se ejecute automáticamente y no se tenga que ejecutar el `start`.

Se puede comprobar que funcione correctamente si en la salida de la interfaz del equipo conectamos una máquina e intentamos recibir el tráfico ya sea con FFmpeg o con un reproductor como VLC.

Posteriormente se instala el Servidor de transmisión con Ubuntu Server 24.04.2 LTS y se configura de la siguiente manera.

Modificar puerto ssh con `sudo nano /etc/ssh/sshd_config` donde se des comenta el puerto y se escribe el que se quiera asignar dentro de los puertos permitidos (por cuestiones de seguridad).

Se cancela el logeo root: `PermitRootLogin` no `#disabled` como se muestra en la Figura 62.

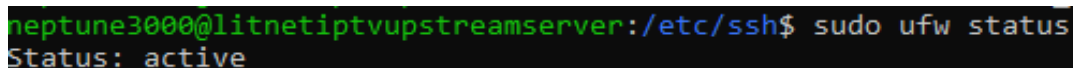


```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Fig. 62 Configuración de conexión ssh bloqueando el acceso root.

Nota. Captura de pantalla realizada por el autor bloqueando el acceso root al servidor.

Activar firewall con `sudo ufw enable` y permitir puerto SSH, véase en las Figuras 63 y 64.



```
neptune3000@litnetiptvupstreamserver:/etc/ssh$ sudo ufw status
Status: active
```

Fig. 63 Activación del firewall.

Nota. Captura de pantalla realizada por el autor de comando para activar firewall integrado.

Se permite el puerto de conexión SSH con `sudo ufw allow [Puerto]` en este caso es el puerto por defecto.



22	ALLOW	Anywhere
----	-------	----------

Fig. 64 Agregación de regla en firewall para permitir conexión sobre el puerto 22.

Nota. Captura de pantalla realizada por el autor de regla añadida en el firewall.

Con el comando `sudo ufw reload` se aplican los cambios en el firewall.

Se agregan políticas de seguridad básicas para denegar o permitir tráfico.

```
sudo ufw default deny incoming
sudo ufw default allow outgoing
```

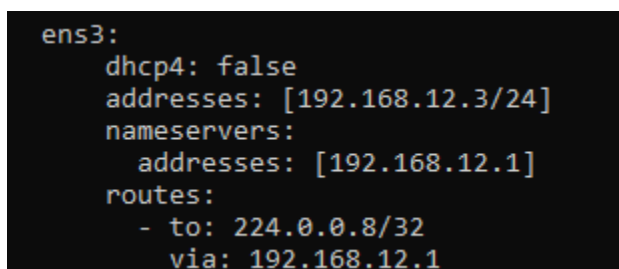
Luego se instala fail2ban para prevenir intrusos.

```
sudo apt install fail2ban
sudo systemctl enable fail2ban
sudo systemctl start fail2ban
```

Se agrega el o los usuarios con privilegios (dependiendo cuantos administren el servidor).

```
sudo adduser USUARIO
sudo usermod -aG USUARIO root
sudo usermod -aG sudo USUARIO
```

Y ahora se configura la interfaz donde se recibirán las transmisiones procesadas, como se muestra en la Figura 65.

A screenshot of a terminal window showing the configuration for the network interface 'ens3'. The configuration is as follows:

```
ens3:
  dhcp4: false
  addresses: [192.168.12.3/24]
  nameservers:
    addresses: [192.168.12.1]
  routes:
    - to: 224.0.0.8/32
      via: 192.168.12.1
```

Fig. 65 Configuración de recepción de tráfico multicast sobre interfaz de red.

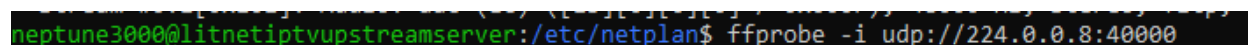
Nota. Captura de pantalla realizada por el autor de la configuración de interfaz de red del servidor de transmisión para recibir el tráfico multicast.

En dicha interfaz se usa la dirección que termina con 12.3 ya que la 12.1 es del gateway, la 12.2 es del Workstation que procesa las señales haciendo la transcodificación.

En las rutas se indica que reciba el tráfico de la 224.0.0.8 por el gateway de la interfaz emisora.

Se instala FFmpeg con `sudo apt install ffmpeg`.

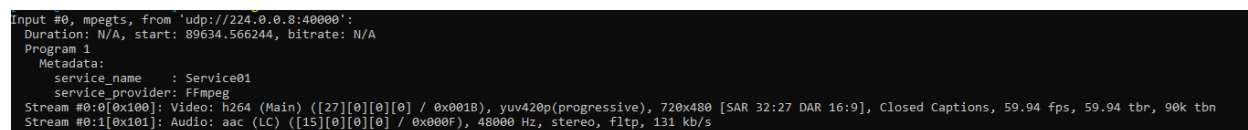
Se hace la prueba de recepción con la herramienta de FFmpeg ejecutando `ffprobe -i udp://IP:Puerto`, como se muestra en la Figura 66.



```
neptune3000@litnetiptvupstreamserver:/etc/netplan$ ffprobe -i udp://224.0.0.8:40000
```

Fig. 66 *Comando para probar recepción de señal procesada.*

Nota. Captura de pantalla realizada por el autor probando la recepción de la señal procesada con la herramienta `ffprobe` de FFmpeg.



```
Input #0, mpegts, from 'udp://224.0.0.8:40000':
  Duration: N/A, start: 89634.566244, bitrate: N/A
  Program 1
  Metadata:
    service_name       : Service01
    service_provider   : FFmpeg
  Stream #0:0[0x100]: Video: h264 (Main) ([27][0][0][0] / 0x001B), yuv420p(progressive), 720x480 [SAR 32:27 DAR 16:9], Closed Captions, 59.94 fps, 59.94 tbr, 90k tbn
  Stream #0:1[0x101]: Audio: aac (LC) ([15][0][0][0] / 0x000F), 48000 Hz, stereo, fltp, 131 kb/s
```

Fig. 67 *Metadatos de la señal procesada recibida.*

Nota. Captura de pantalla realizada por el autor visualizando metadatos de la recepción de la señal procesada recibida en el servidor de transmisión.

Se puede verificar que se transcodifica el video a H.264 y AAC, a su vez se desentrelaza y los frames ya no son 29.97 si no 59.94.

La cadencia de cuadros es un aspecto relevante, ya que la mayoría de los monitores actuales operan a 60 Hz. Cuando el contenido se encuentra a 29.97 fps, pueden presentarse desajustes en la sincronización que generan el efecto conocido como judder, el cual produce una ligera vibración o irregularidad en el movimiento perceptible para el espectador.

Es bastante extensa la documentación de FFmpeg y las opciones que tiene para manejar el video, pero con las opciones básicas se puede tener una emisión confiable.

Se procede a generar las listas M3U8 para su distribución a través de Nginx que funge como servidor HTTP.

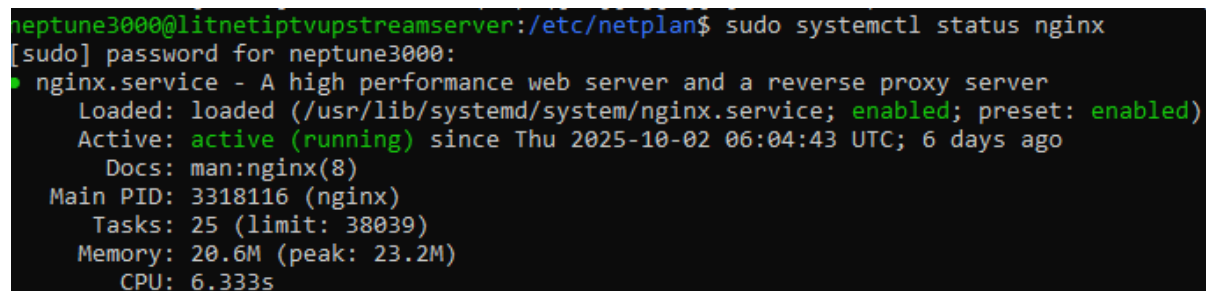
Se instala Nginx de la siguiente manera.

```
sudo apt install nginx
```

```
sudo systemctl start nginx (se inicia).
```

```
sudo systemctl enable nginx (lo habilita para arrancar con el sistema).
```

Se verifica el estado con `systemctl status Nginx`, véase en la Figura 68.



```
neptune3000@litnetiptvupstreamserver:/etc/netplan$ sudo systemctl status nginx
[sudo] password for neptune3000:
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-10-02 06:04:43 UTC; 6 days ago
     Docs: man:nginx(8)
    Main PID: 3318116 (nginx)
      Tasks: 25 (limit: 38039)
    Memory: 20.6M (peak: 23.2M)
       CPU: 6.333s
```

Fig. 68 *Estado de servidor Nginx.*

Nota. Captura de pantalla realizada por el autor observando el estado del servicio nginx.

Se modifica el archivo de configuración de Nginx. El cual se encuentra en la ruta `/etc/nginx/conf.d`

Al final se agrega del archivo se agrega la configuración del servidor lo que implica el puerto de escucha, y la ruta que se sirve, véase en la Figura 69.

```
server {
    listen 8080;
    server_name localhost;
    location / {
        root /var/www/hls/;
        add_header Cache-Control no-cache;
        add_header Access-Control-Allow-Origin *;
    }
}
```

Fig. 69 *Modificación del archivo de configuración de Nginx.*

Nota. Captura de pantalla realizada por el autor observando la modificación necesaria para el servidor Nginx.

En la ruta /var/www/hls/ es donde se encuentra la carpeta donde están disponibles los fragmentos de video que se sirven por el puerto 8080 vía HTTP.

Se agrega el puerto 8080 en UFW para que permita las conexiones.

8080/tcp	ALLOW	Anywhere
----------	-------	----------

Fig. 70 *Regla añadida al firewall para permitir conexión de Nginx.*

Nota. Captura de pantalla realizada por el autor de la regla añadida al firewall sobre el puerto que se escucha en el servidor Nginx.

Se crea la carpeta que se servirá por HTTP con `sudo mkdir hls`.

Luego dentro de esa carpeta se crearán varias subcarpetas para las transmisiones, como se muestra en la Figura 71.

```
neptune3000@litnetiptvupstreamserver:/var/www/hls$ pwd
/var/www/hls
neptune3000@litnetiptvupstreamserver:/var/www/hls$ ls
prueba prueba1 prueba2 prueba3 prueba4 prueba5 prueba6 prueba7 prueba8 prueba9
neptune3000@litnetiptvupstreamserver:/var/www/hls$
```

Fig. 71 Ruta y carpetas donde se guardan los fragmentos de la lista M3U8.

Nota. Captura de pantalla realizada por el autor de la carpeta raíz que se comparte y las subcarpetas que contienen las listas y fragmentos M3U8.

Se procede a crear los servicios que generarán las listas M3U8, véanse la Figura 72 y 73.

```
pruebahlsWS1.service
pruebahlsWS2.service
pruebahlsWS3.service
pruebahlsWS4.service
pruebahlsWS5.service
pruebahlsWS6.service
pruebahlsWS7.service
pruebahlsWS8.service
pruebahlsWS9.service
pruebahlsWS.service
```

Fig. 72 Creación de archivos que contienen los comandos para generar las listas M3U8.

Nota. Captura de pantalla realizada por el autor de la creación de servicios para generar listas M3U8.

```
neptune3000@litnetiptvupstreamserver:/etc/systemd/system$ sudo cat pruebahlsWS.service
[Unit]
Description=Servicio ffmpeg para IPTV HLS
After=network.target

[Service]
Type=simple
ExecStart=/usr/bin/ffmpeg -i "udp://224.0.0.8:40000" -c copy -f hls -hls_time 3 -hls_list_size 10 -hls_flags delete_segments -y /var/www/hls/prueba/master.m3u8
Restart=always
User=root
WorkingDirectory=/var/www/working/prueba
Environment=HOME=/root

[Install]
WantedBy=multi-user.target
```

Fig. 73 Ejemplo de servicio recibiendo la señal procesada y generando la lista M3U8.

Nota. Captura de pantalla realizada por el autor de generación de la lista M3U8 con la señal procesada hacia la ruta de la carpeta servida por Nginx.

Se puede observar que se recibe la señal procesada, se copia el códec ya que no se necesita modificar, y la salida será en el protocolo HLS, donde serán diez fragmentos de tres segundos, al final se indica que cuando sean diez segmentos se vaya borrando los segmentos anteriores para no saturar el disco con los fragmentos de video. Y se escriben en la ruta que se indicó anteriormente, dentro de la carpeta correspondiente.

Se muestra la creación del archivo master.m3u8 generalmente llamado master pero también el archivo puede llamarse de otra forma como por ejemplo stream.m3u8.

De la misma manera, se recargan los servicios, y se inician de la siguiente forma:

sudo systemctl restart pruebahlsWS*.service (de este modo se ahorra algo de tiempo al iniciar todos a la vez).

Se verifica el estado con: sudo systemctl status pruebahlsWS*.service

```
neptune300@litnetiptvupstreamserver:/etc/systemd/system$ sudo systemctl status pruebahlsWS*.service
● pruebahlsWS1.service - Servicio ffmpeg para IPTV HLS
   Loaded: loaded (/etc/systemd/system/pruebahlsWS1.service; disabled; preset: enabled)
   Active: active (running) since Thu 2025-10-02 06:04:43 UTC; 6 days ago
     Main PID: 3318196 (ffmpeg)
       Tasks: 4 (limit: 30039)
    Memory: 53.4M (peak: 59.8M)
         CPU: 1h 13min 28.162s
    CGroup: /system.slice/pruebahlsWS1.service
            └─3318196 /usr/bin/ffmpeg -i udp://224.0.0.8:40001 -c copy -f hls -hls_time 3 -hls_list_size 10 -hls_flags delete_segments -y /var/www/hls/prueba1/master.m3u8

Oct 08 20:52:54 litnetiptvupstreamserver ffmpeg[3318196]: [5768 blob data]
Oct 08 20:52:54 litnetiptvupstreamserver ffmpeg[3318196]: [hls @ 0x5d925513de00] Opening '/var/www/hls/prueba1/master.m3u8.tmp' for writing
Oct 08 20:52:58 litnetiptvupstreamserver ffmpeg[3318196]: [5218 blob data]
Oct 08 20:52:58 litnetiptvupstreamserver ffmpeg[3318196]: [hls @ 0x5d925513de00] Opening '/var/www/hls/prueba1/master.m3u8.tmp' for writing
Oct 08 20:53:02 litnetiptvupstreamserver ffmpeg[3318196]: [5218 blob data]
Oct 08 20:53:02 litnetiptvupstreamserver ffmpeg[3318196]: [hls @ 0x5d925513de00] Opening '/var/www/hls/prueba1/master.m3u8.tmp' for writing
Oct 08 20:53:06 litnetiptvupstreamserver ffmpeg[3318196]: [5218 blob data]
Oct 08 20:53:06 litnetiptvupstreamserver ffmpeg[3318196]: [hls @ 0x5d925513de00] Opening '/var/www/hls/prueba1/master.m3u8.tmp' for writing
Oct 08 20:53:11 litnetiptvupstreamserver ffmpeg[3318196]: [5768 blob data]
Oct 08 20:53:11 litnetiptvupstreamserver ffmpeg[3318196]: [hls @ 0x5d925513de00] Opening '/var/www/hls/prueba1/master.m3u8.tmp' for writing

● pruebahlsWS2.service - Servicio ffmpeg para IPTV HLS
   Loaded: loaded (/etc/systemd/system/pruebahlsWS2.service; disabled; preset: enabled)
   Active: active (running) since Mon 2025-10-06 17:55:12 UTC; 2 days ago
     Main PID: 3996381 (ffmpeg)
       Tasks: 4 (limit: 30039)
    Memory: 43.8M (peak: 53.1M)
         CPU: 22min 54.100s
    CGroup: /system.slice/pruebahlsWS2.service
            └─3996381 /usr/bin/ffmpeg -i udp://224.0.0.8:40002 -c copy -f hls -hls_time 3 -hls_list_size 10 -hls_flags delete_segments -y /var/www/hls/prueba2/master.m3u8

Oct 08 20:52:56 litnetiptvupstreamserver ffmpeg[3996381]: [512B blob data]
Oct 08 20:52:56 litnetiptvupstreamserver ffmpeg[3996381]: [hls @ 0x64b9bbe8bcc0] Opening '/var/www/hls/prueba2/master.m3u8.tmp' for writing
Oct 08 20:53:00 litnetiptvupstreamserver ffmpeg[3996381]: [512B blob data]
Oct 08 20:53:00 litnetiptvupstreamserver ffmpeg[3996381]: [hls @ 0x64b9bbe8bcc0] Opening '/var/www/hls/prueba2/master.m3u8.tmp' for writing
Oct 08 20:53:04 litnetiptvupstreamserver ffmpeg[3996381]: [512B blob data]
Oct 08 20:53:04 litnetiptvupstreamserver ffmpeg[3996381]: [hls @ 0x64b9bbe8bcc0] Opening '/var/www/hls/prueba2/master.m3u8.tmp' for writing
Oct 08 20:53:08 litnetiptvupstreamserver ffmpeg[3996381]: [512B blob data]
Oct 08 20:53:08 litnetiptvupstreamserver ffmpeg[3996381]: [hls @ 0x64b9bbe8bcc0] Opening '/var/www/hls/prueba2/master.m3u8.tmp' for writing
Oct 08 20:53:12 litnetiptvupstreamserver ffmpeg[3996381]: [512B blob data]
Oct 08 20:53:12 litnetiptvupstreamserver ffmpeg[3996381]: [hls @ 0x64b9bbe8bcc0] Opening '/var/www/hls/prueba2/master.m3u8.tmp' for writing
```

Fig. 74 Estado de los servicios generados para creación de listas M3U8.

Nota. Captura de pantalla realizada por el autor del estado de los servicios que generan la lista M3U8 con la señal procesada hacia la ruta de la carpeta servida por Nginx.

Como se observa en la Figura 74, se están escribiendo los segmentos en las rutas correspondientes lo cual se puede validar de manera sencilla entrando a la ruta.

```

neptune3000@litnetiptvupstreamserver:/var/www/hls$ ls
prueba prueba1 prueba2 prueba3 prueba4 prueba5 prueba6 prueba7 prueba8 prueba9
neptune3000@litnetiptvupstreamserver:/var/www/hls$ cd prueba
neptune3000@litnetiptvupstreamserver:/var/www/hls/prueba$ ls
master10.ts master11.ts master12.ts master2.ts master3.ts master4.ts master5.ts master6.ts master7.ts master8.ts master9.ts master.m3u8
neptune3000@litnetiptvupstreamserver:/var/www/hls/prueba$ ls
master10.ts master11.ts master12.ts master2.ts master3.ts master4.ts master5.ts master6.ts master7.ts master8.ts master9.ts master.m3u8
neptune3000@litnetiptvupstreamserver:/var/www/hls/prueba$ ls
master10.ts master11.ts master12.ts master2.ts master3.ts master4.ts master5.ts master6.ts master7.ts master8.ts master9.ts master.m3u8
neptune3000@litnetiptvupstreamserver:/var/www/hls/prueba$

```

Fig. 75 Archivos generados en las carpetas dentro de la ruta que sirve Nginx.

Nota. Captura de pantalla realizada por el autor de las listas M3U8 generadas en las carpetas correspondientes que se comparten con Nginx.

Se observa la generación de los segmentos .ts que conforman la lista M3U8 en las distintas carpetas, y cómo, al alcanzar un total de diez archivos, los segmentos más antiguos se eliminan automáticamente.

La lista M3U8 contiene la información necesaria para que el reproductor de video procese y reproduzca los segmentos de forma ordenada, como se muestra en la Figura 76.

```

neptune3000@litnetiptvupstreamserver:/var/www/hls/prueba$ sudo cat master.m3u8
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:4
#EXT-X-MEDIA-SEQUENCE:38
#EXTINF:4.170833,
master38.ts
#EXTINF:4.170833,
master39.ts
#EXTINF:4.170833,
master40.ts
#EXTINF:4.170833,
master41.ts
#EXTINF:4.170833,
master42.ts
#EXTINF:4.170833,
master43.ts
#EXTINF:4.170833,
master44.ts
#EXTINF:4.170833,
master45.ts
#EXTINF:4.170833,
master46.ts
#EXTINF:4.170833,
master47.ts

```

Fig. 76 Visualización de contenido de lista M3U8.

Nota. Captura de pantalla realizada por el autor del contenido del archivo M3U8 que hace referencia a los .ts que son los fragmentos de vídeo producidos.

El encabezado #EXTM3U indica que es un archivo M3U8 válido.

#EXT-X-VERSION indica la versión del protocolo HLS.

#EXT-X-TARGETDURATION indica la duración máxima en segundos de cada segmento .ts

#EXT-X-MEDIA-SEQUENCE indica el número de secuencia del primer segmento.

#EXTINF indica la duración de ese segmento.

Esto indica que archivos de vídeo reproducir y en qué orden.

Por último, en el proceso de instalación del servidor se requiere configurar una interfaz con una dirección IP pública que garantice el acceso desde cualquier ubicación mediante un dispositivo con conexión a Internet, de la siguiente manera:

Nombre de la interfaz:

dhcp4: false

addresses: [IP Pública/Máscara de red]

nameservers:

addresses: [8.8.8.8, 1.1.1.1]

routes:

- to: 0.0.0.0/0

via: Gateway del proveedor

metric: 100

Se puede usar el reproductor de video VLC para comprobar que se estén publicando de manera correcta las listas M3U8, véase la Figura 77.

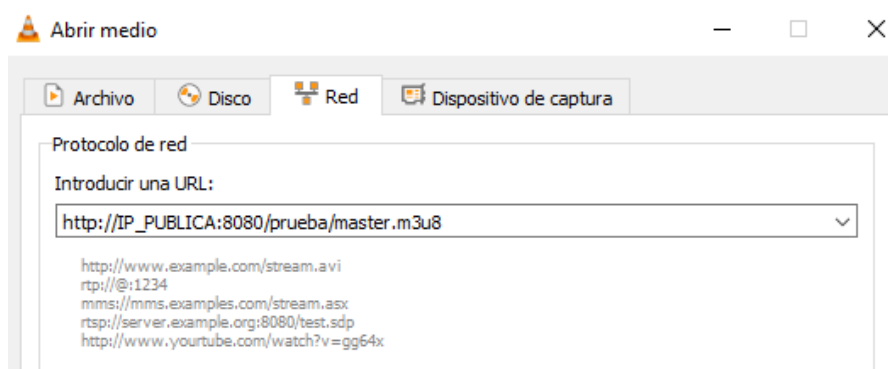


Fig. 77 Ventana “Abrir medio” del reproductor VLC mostrando la entrada de lista M3U8 apuntando a la IP del servidor con Nginx ya sea pública o local.

Nota. Captura de pantalla realizada por el autor en el programa VLC Media Player durante la ejecución del proceso de demostración.

Se puede observar la correcta visualización como se muestra en la Figura 78.



Fig. 78 Reproducción del canal Foro TV en VLC Media Player, mostrando la emisión en vivo dentro del reproductor.

Nota. Captura de pantalla realizada por el autor durante la reproducción del canal Foro TV en el programa VLC Media Player. El contenido audiovisual mostrado pertenece a Foro TV.

Comprobando que se están sirviendo los segmentos de video por medio de la lista M3U8 generada de manera correcta.

Una vez instalada la parte básica, se procede a generar el middleware. Este puede implementarse en el mismo servidor o bien puede ser en un equipo diferente. En este caso, se instala en un equipo sencillo, un Raspberry Pi 5, con el propósito de simular un servidor de middleware dedicado.

El Raspberry Pi 5 incorpora opciones que permiten instalar el sistema operativo directamente desde Internet, sin necesidad de cargar manualmente una imagen. Para este caso, se consideran dos alternativas de instalación.

La primera opción consiste en instalar LibreELEC, un sistema operativo basado en Linux diseñado para ejecutar Kodi, un software de centro multimedia. Este sistema convierte al Raspberry Pi en un media center dedicado, con un rendimiento eficiente y un proceso de configuración rápido. Asimismo, permite la instalación de complementos como Jellyfin, un servidor de medios de código abierto que posibilita organizar, transmitir y acceder a contenidos tales como películas, series y televisión en vivo.

La segunda opción implica instalar Ubuntu Server en el Raspberry Pi e implementar Jellyfin como un servicio independiente. Aunque esta alternativa ofrece mayor flexibilidad, requiere una configuración más detallada y puede resultar ligeramente más compleja.

Ambas opciones hacen lo mismo sin embargo la gestión es diferente, las rutas cambian, la forma de administrar los procesos como activar o desactivar el servicio también lo hace.

En este caso, debido al número reducido de flujos a manejar, se opta por la primera alternativa, ya que ofrece una implementación más rápida. Al iniciar el Raspberry Pi, dentro del asistente de instalación se presenta la categoría Media Center, donde es posible seleccionar la opción LibreELEC, véase la Figura 79.



Fig. 79 Interfaz principal del programa Raspberry Pi Imager mostrando las opciones de selección de dispositivo, sistema operativo y almacenamiento.

Nota. Captura de pantalla realizada por el autor durante el proceso de preparación de instalación del sistema operativo en Raspberry Pi.

Al iniciar el Raspberry se presentan opciones de configuración preliminares de carácter básico.

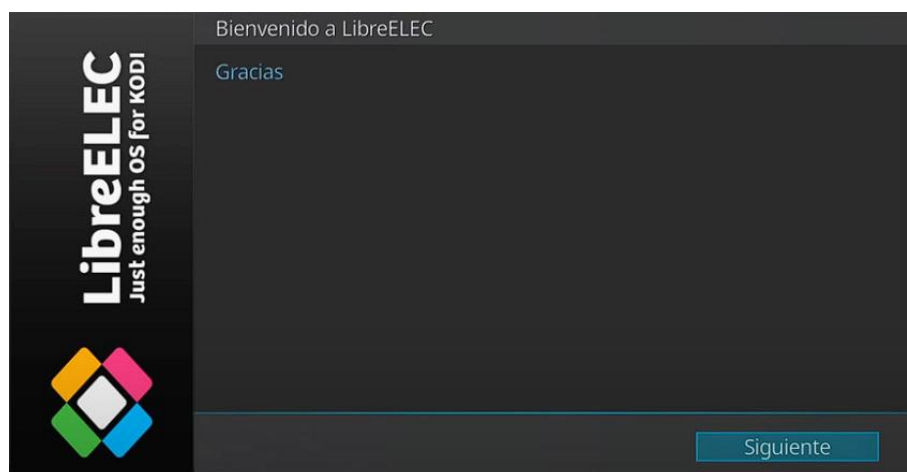


Fig. 80 Pantalla de “Bienvenido a LibreELEC”

Nota. Captura de pantalla realizada por el autor durante la configuración inicial del sistema LibreELEC.

Al concluir la configuración inicial, se pueden instalar los add-on como Jellyfin en el apartado de repositorios, como se muestra en la Figura 81.

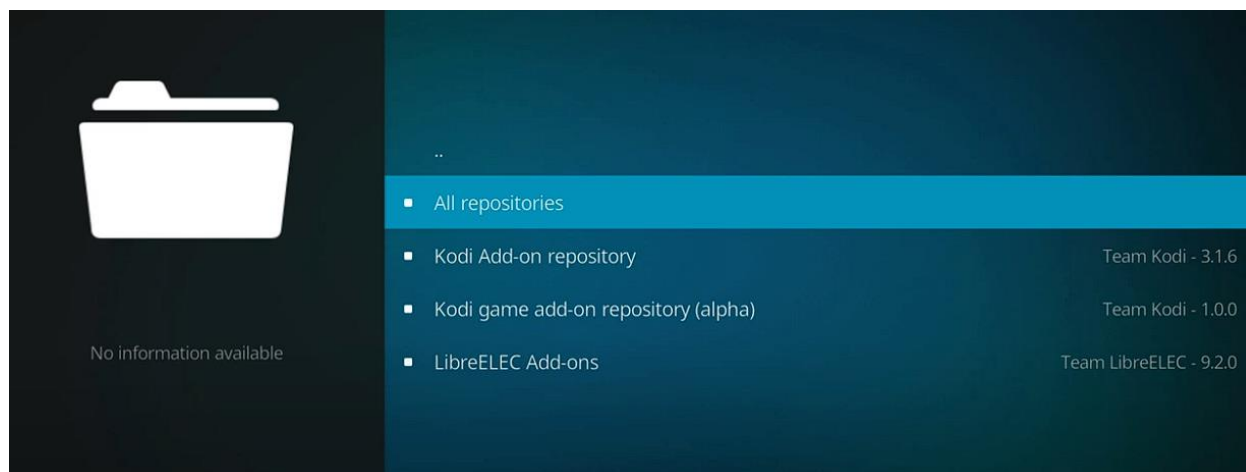


Fig. 81 Selección de add-ons de LibreELEC

Nota. Captura de pantalla realizada por el autor durante la configuración inicial del sistema LibreELEC.

Una vez instalado el add-on de Jellyfin, este puede activarse y su interfaz es accesible mediante la dirección IP asignada al equipo, utilizando el puerto 8096 ya que se encuentra habilitado de manera predeterminada y puede modificarse en caso de ser necesario.

Para permitir el acceso externo, es posible realizar un reenvío de puertos (port forward), de modo que las solicitudes dirigidas a la IP pública en el puerto 8096 se redirijan hacia el Raspberry.

En dispositivos MikroTik, la configuración del port forwarding se implementa en IP > Firewall mediante la adición de la regla necesaria, según se observa en las Figuras 82 y 83.

The screenshot shows the 'General' tab of a MikroTik Firewall Rule configuration window. The 'Chain' is set to 'dstnat'. The 'Src. Address' field is empty. The 'Dst. Address' field is set to 'IP PUBLICA'. The 'Protocol' is set to '6 (tcp)'. The 'Src. Port' field is empty. The 'Dst. Port' field is set to '8096'. The 'Any. Port' field is empty. The 'In. Interface' field is empty. The 'Out. Interface' field is empty. On the right side, there are buttons for 'OK', 'Cancel', 'Apply', 'Disable', 'Comment', 'Copy', 'Remove', 'Reset Counters', and 'Reset All Counters'.

Fig. 82 Configuración de una regla de firewall tipo Destination NAT en MikroTik RouterOS para redirigir tráfico entrante desde la IP pública a un dispositivo local.

Nota. Captura de pantalla realizada por el autor durante el proceso de configuración del firewall en MikroTik RouterOS.

The screenshot shows the 'Action' tab of the same MikroTik Firewall Rule configuration window. The 'Action' is set to 'dst-nat'. The 'Log' checkbox is checked. The 'Log Prefix' field is empty. The 'To Addresses' field is set to '192.168.70.248'. The 'To Ports' field is set to '8096'. On the right side, there are buttons for 'OK', 'Cancel', 'Apply', 'Disable', 'Comment', 'Copy', 'Remove', 'Reset Counters', and 'Reset All Counters'.

Fig. 83 Configuración de la acción de la regla del firewall tipo Destination NAT en MikroTik RouterOS redirigiendo el tráfico a la IP del Raspberry con LibreELEC y Jellyfin.

Nota. Captura de pantalla realizada por el autor durante el proceso de configuración del firewall en MikroTik RouterOS.

192.168.70.248 es la IP que se le asignó por DHCP al Raspberry.

De igual manera, el acceso puede realizarse mediante un navegador web, como Chrome, desde cualquier equipo con conexión a Internet, tal como se observa en las Figuras 84 y 85.



Fig. 84 *Ejemplo de acceso a un servicio mediante la dirección IP pública y puerto desde la barra de direcciones del navegador Google Chrome.*

Nota. Captura de pantalla realizada por el autor como demostración del acceso a un servicio por IP y puerto en el navegador web.

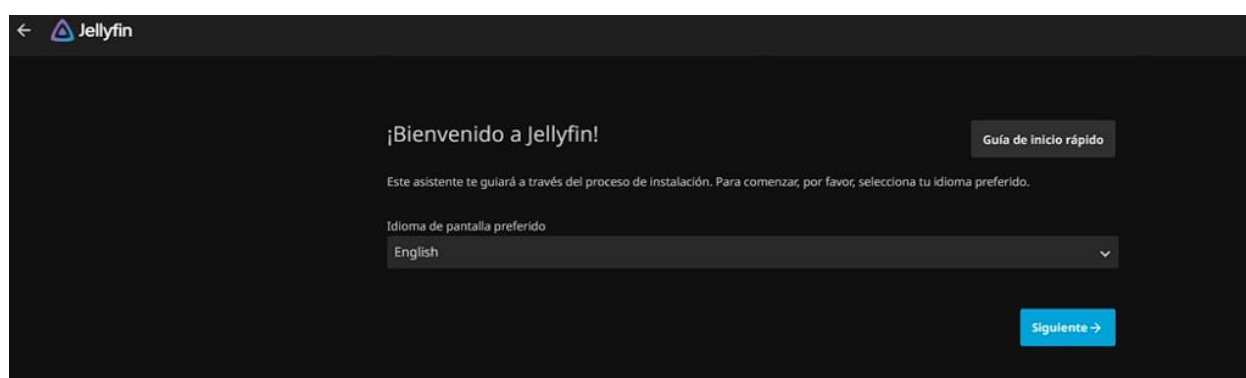


Fig. 85 *Pantalla de bienvenida del servidor multimedia Jellyfin mostrando el asistente de configuración inicial.*

Nota. Captura de pantalla realizada por el autor durante el proceso de configuración inicial del servidor Jellyfin.

En esta etapa se realizará la configuración inicial, que incluye la selección del idioma, la creación del usuario administrador, la definición de la biblioteca inicial, la elección del lenguaje de la interfaz y la habilitación de los permisos de conexión remota.

Una vez completada esta configuración, será posible iniciar sesión de manera habitual.

Desde el Panel de control es posible gestionar la configuración del contenido, así como el acceso de los usuarios y los permisos asociados a su contenido.

Asimismo, en el Panel de control se pueden agregar bibliotecas según el tipo de contenido que se desee incluir, tal como se observa en las Figuras 86 a 87.

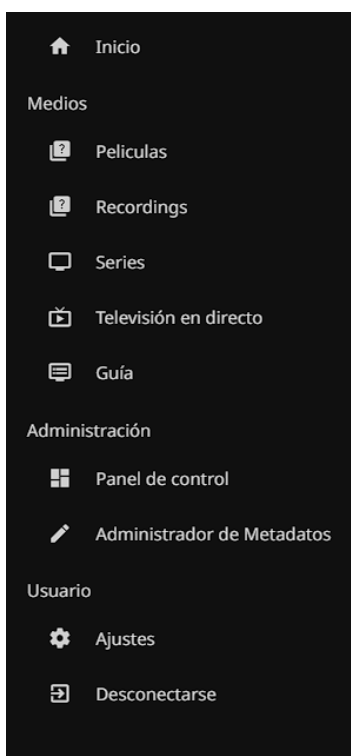


Fig. 86 Barra lateral del entorno principal de Jellyfin mostrando las secciones de navegación del servidor multimedia.

Nota. Captura de pantalla realizada por el autor durante la exploración de la interfaz de Jellyfin.

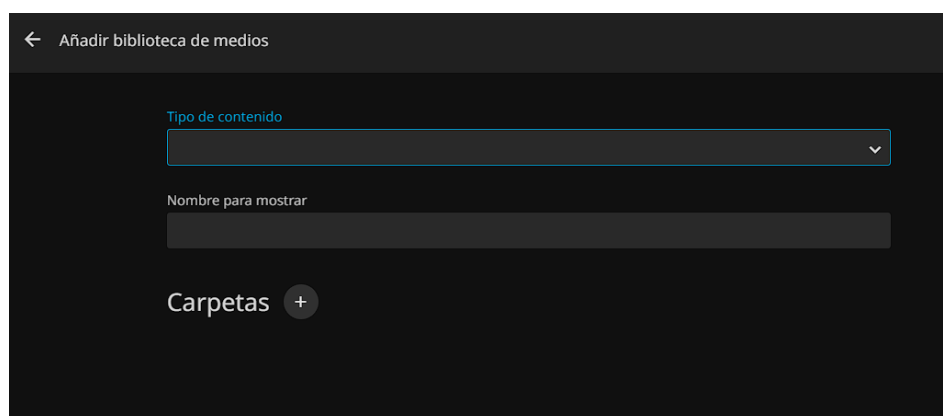


Fig. 87 Pantalla del asistente de configuración inicial de Jellyfin para añadir bibliotecas.

Nota. Captura de pantalla realizada por el autor durante la exploración de la interfaz de Jellyfin.

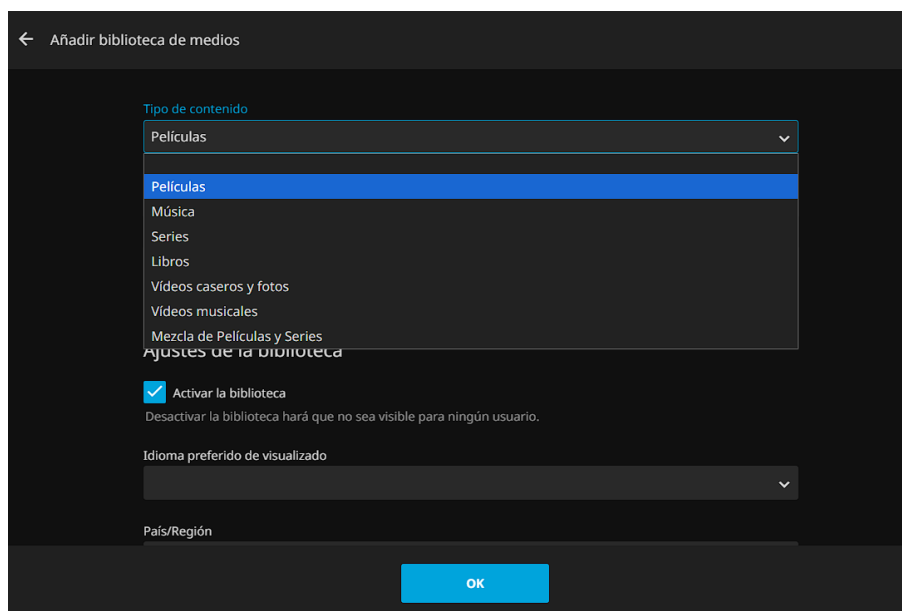


Fig. 88 Selección del tipo de contenido de Jellyfin para añadir biblioteca.

Nota. Captura de pantalla realizada por el autor durante la exploración de la interfaz de Jellyfin.

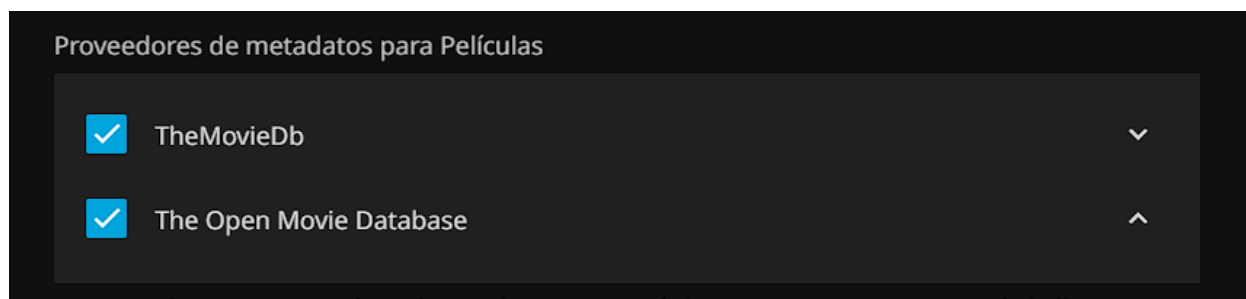


Fig. 89 Selección de proveedores de metadatos para Jellyfin de películas (The Movie Database y The Open Movie Database).

Nota. Captura de pantalla realizada por el autor durante la exploración de la interfaz de Jellyfin.

Dentro del apartado podemos ver cosas interesantes como los proveedores de metadatos, donde se aprecia su uso al agregar una película o serie.

Luego de crear la biblioteca se define la ruta de donde tomará los videos en este caso películas, véase la Figura 90.

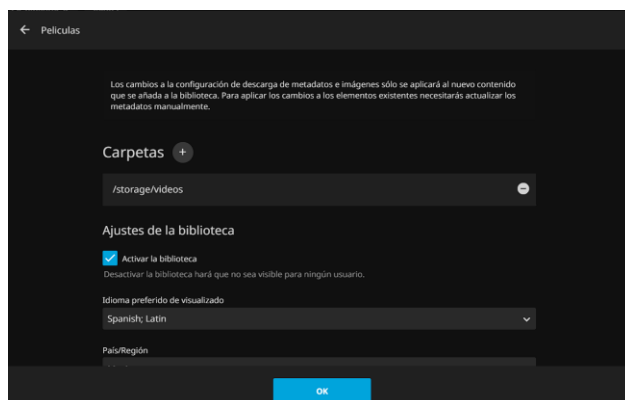


Fig. 90 *Pantalla de Jellyfin mostrando la selección de la carpeta local que contiene los archivos de películas.*

Nota. Captura de pantalla realizada por el autor durante la exploración de la interfaz de Jellyfin.

Para enviar películas como se vio anteriormente se puede usar el comando SCP a la ruta marcada.

Después de definir la ruta se escanean todas las rutas para actualizarlas, como se muestra en la Figura 91.

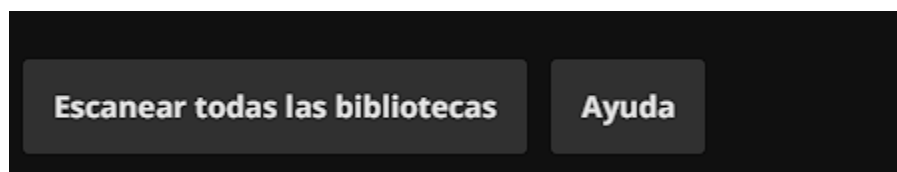


Fig. 91 *Botón de Jellyfin para escaneo de bibliotecas.*

Nota. Captura de pantalla realizada por el autor durante la exploración de la interfaz de Jellyfin.

Si el video tiene nombre de la película, puede intentar agregar los metadatos automáticamente, como se muestra en la Figura 92.

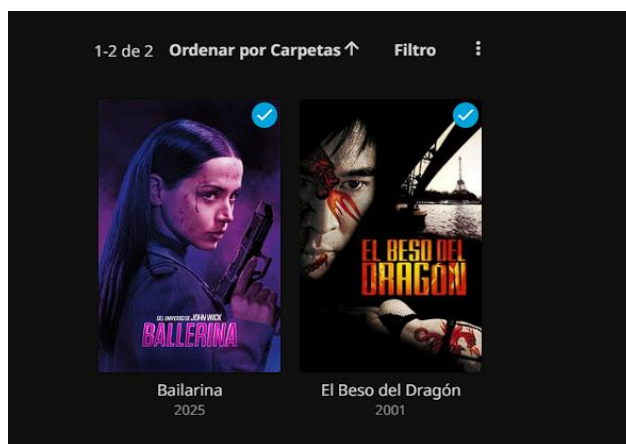


Fig. 92 Sección de “Películas” en Jellyfin mostrando títulos disponibles en la biblioteca multimedia.

Nota. Captura de pantalla realizada por el autor durante la visualización de la biblioteca de películas en Jellyfin. Los títulos mostrados (Bailarina, 2025; El beso del dragón, 2001) se presentan únicamente con fines ilustrativos.

Asimismo, es posible hacer clic en Identificar y Buscar con el fin de mejorar la precisión en la ubicación del título y asegurar que los metadatos correspondan correctamente al contenido, tal como se observa en las Figuras 93 y 94.

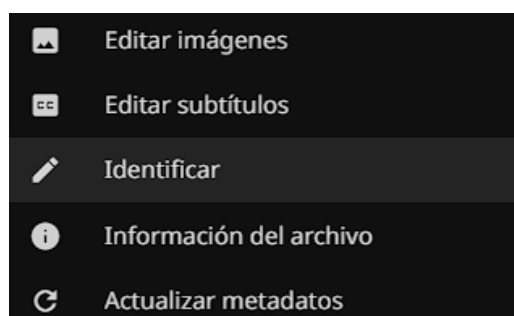


Fig. 93 Botón de Jellyfin para identificar títulos y agregar metadatos del vídeo.

Nota. Captura de pantalla realizada por el autor durante la exploración de la interfaz de Jellyfin.

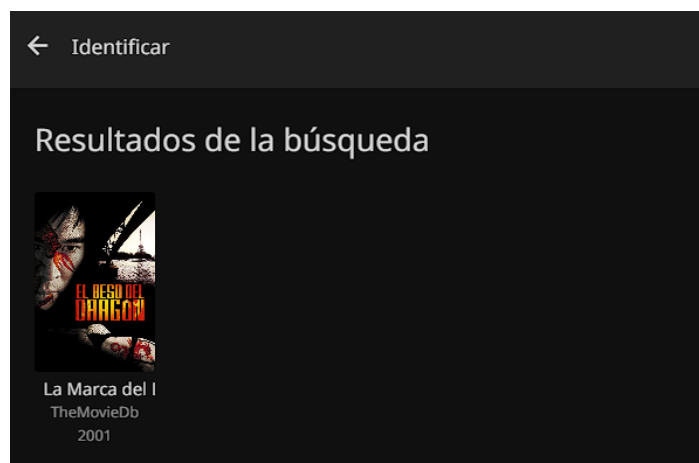


Fig. 94 Pantalla de Jellyfin mostrando el resultado de la identificación de una película mediante los metadatos de The Movie Database (TMDb).

Nota. Captura de pantalla realizada por el autor durante el proceso de identificación automática de contenido en Jellyfin.

De tal forma se actualizan los metadatos de la película sin tener que intervenir en descripción e imágenes.



Fig. 95 Pantalla de Jellyfin mostrando la vista de detalles de una película con metadatos obtenidos desde The Movie Database (TMDb).

Nota. Captura de pantalla realizada por el autor durante la visualización de los metadatos de una película en Jellyfin.

Los datos se presentan en una interfaz amigable, tal como se muestra en la Figura 95.

El procedimiento es similar para las series, sin embargo, en este caso, se debe respetar un orden específico al nombrar los episodios.

```
#####
# LibreELEC #
# https://libreelec.tv #
#####

LibreELEC (official): 12.0.2 (RPi5.aarch64)
LibreELEC:~ # ls
backup      emulators  lost+found  picons     recordings  tvshows
downloads   logfiles   music       pictures    screenshots videos
LibreELEC:~ # cd recordings/
LibreELEC:~/recordings # ls
Kono Subarashii Sekai ni Bakuen wo!
LibreELEC:~/recordings # cd Kono\ Subarashii\ Sekai\ ni\ Bakuen\ wo!/
LibreELEC:~/recordings/Kono Subarashii Sekai ni Bakuen wo! # ls
Season 01
LibreELEC:~/recordings/Kono Subarashii Sekai ni Bakuen wo! # cd Season\ 01/
LibreELEC:~/recordings/Kono Subarashii Sekai ni Bakuen wo!/Season 01 # ls
Episode S01E01.mp4 Episode S01E04.mp4 Episode S01E07.mp4 Episode S01E10.mp4
Episode S01E02.mp4 Episode S01E05.mp4 Episode S01E08.mp4 Episode S01E11.mp4
Episode S01E03.mp4 Episode S01E06.mp4 Episode S01E09.mp4 Episode S01E12.mp4
LibreELEC:~/recordings/Kono Subarashii Sekai ni Bakuen wo!/Season 01 #
```

Fig. 96 Estructura de carpetas y nombres de archivo utilizada para la organización de series en Jellyfin.

Nota. Captura de pantalla realizada por el autor mostrando la disposición de directorios y archivos empleados para la correcta identificación de series en Jellyfin.

Se recomienda identificar el nombre de la carpeta principal con el título de la serie, creando dentro de ella una carpeta por temporada. Cada temporada debe contener los episodios nombrados siguiendo el formato tal como se observa en la Figura 96.

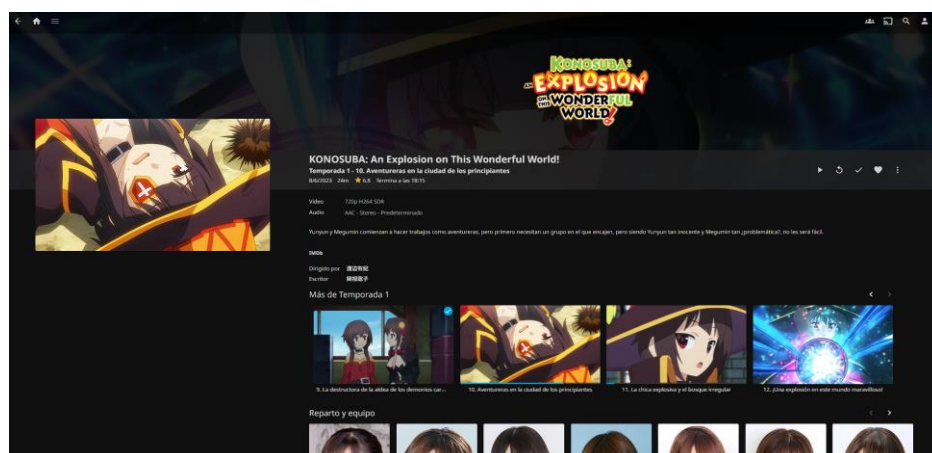


Fig. 97 Pantalla de Jellyfin mostrando la vista de detalles de una serie con metadatos obtenidos desde The Movie Database (TMDb).

Nota. Captura de pantalla realizada por el autor durante la visualización de los metadatos de una serie en Jellyfin.

Esta organización resulta de gran ayuda para subir contenido de manera rápida y eficiente.

Además del video bajo demanda, correspondiente a películas y series, es posible gestionar contenido de transmisión en vivo, el cual se administra desde el apartado de Televisión en vivo, véase la Figura 98.

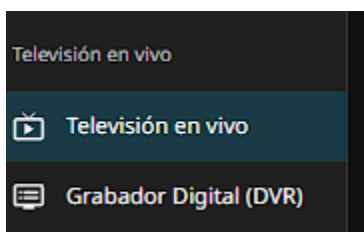


Fig. 98 Botón de Jellyfin para identificar títulos y agregar metadatos del video.

Nota. Captura de pantalla realizada por el autor durante la exploración de la interfaz de Jellyfin.

En este apartado se carga la lista M3U8 en Jellyfin, lo que permite integrar y gestionar los canales de televisión en vivo dentro del sistema, como se muestra en la Figura 99.

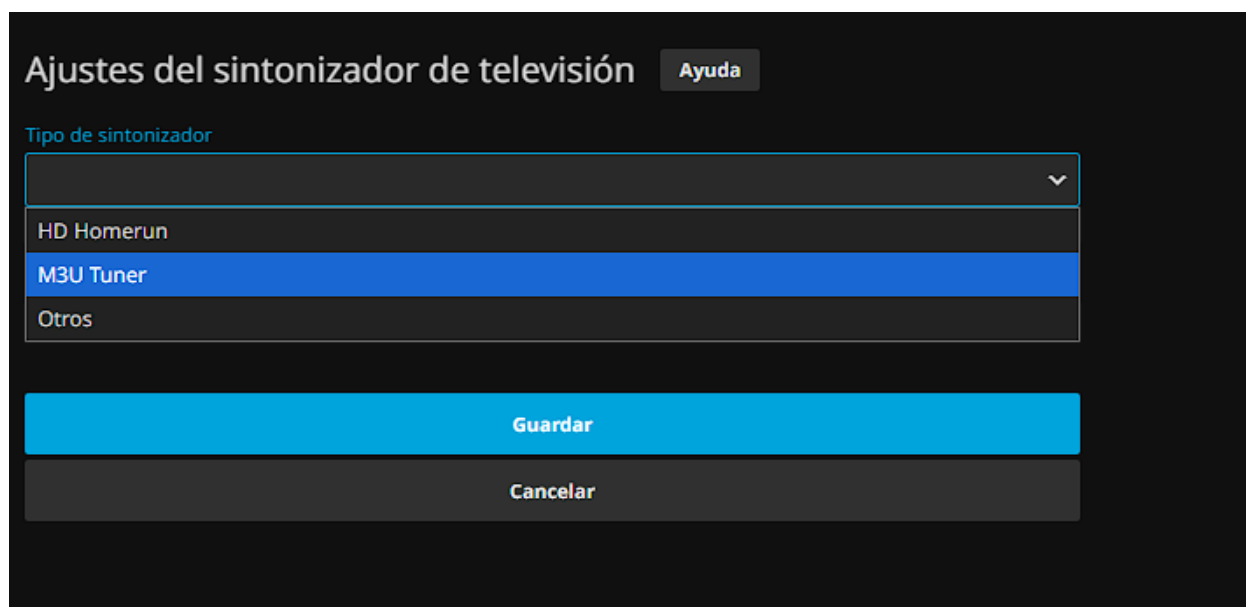


Fig. 99 Pantalla de configuración de Jellyfin para agregar un sintonizador M3U (M3U tuner).

Nota. Captura de pantalla realizada por el autor durante el proceso de configuración del sintonizador M3U en Jellyfin.

En este apartado es posible agregar una ruta o un archivo, lo que permite incluir la lista M3U8 correspondiente para la emisión de los canales en vivo, véase la Figura 100.

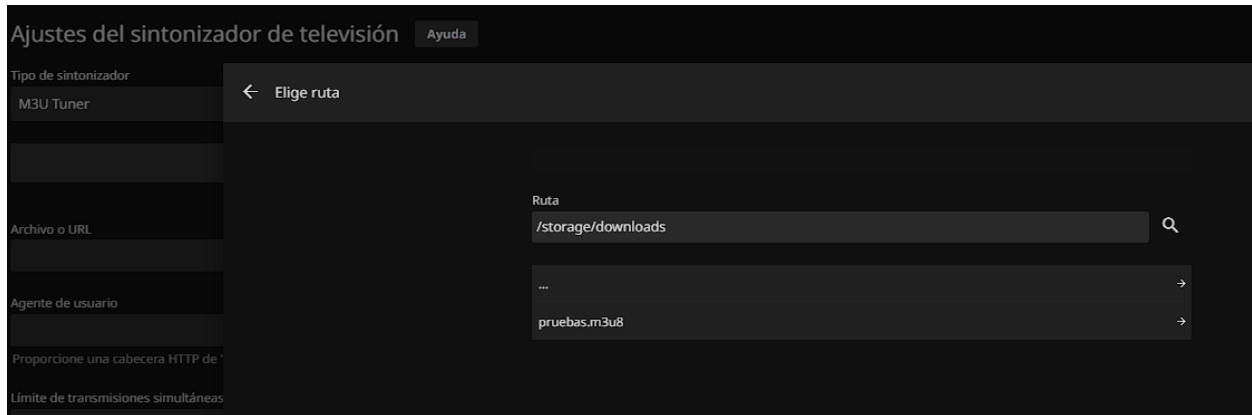


Fig. 100 Pantalla de selección de archivo que muestra la ruta del archivo M3U8 utilizado para configurar un sintonizador en Jellyfin.

Nota. Captura de pantalla realizada por el autor durante el proceso de configuración del sintonizador M3U en Jellyfin.

En este caso, se crea un archivo M3U8 que contiene las URL generadas previamente. Para ello, se elabora un archivo de texto en el cual se incluyen las direcciones correspondientes, véase la Figura 101.

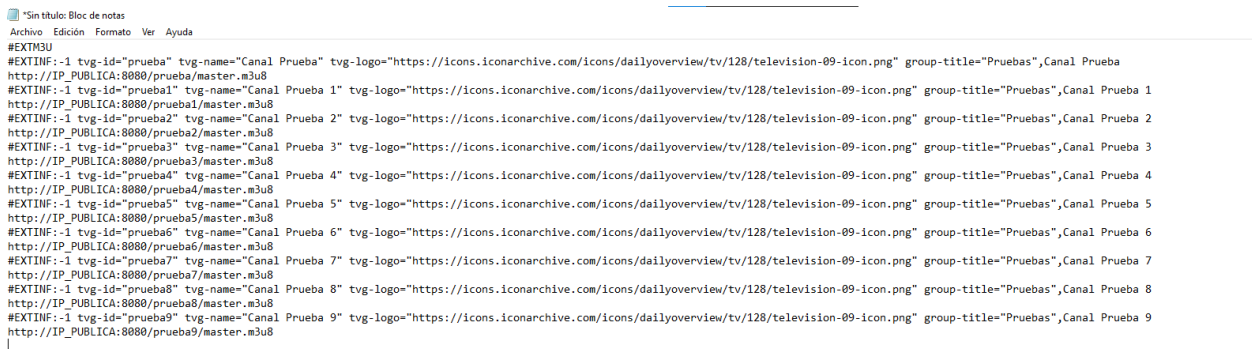


Fig. 101 Contenido de un archivo M3U8 utilizado para la configuración de un sintonizador IPTV.

Nota. Captura de pantalla realizada por el autor mostrando el contenido de un archivo M3U8 con fines ilustrativos.

El archivo se guarda con la extensión .m3u8 y se transfiere al equipo mediante SCP. Asimismo, es posible abrir la lista M3U8 en el reproductor VLC para verificar su correcto funcionamiento, como se muestra en las Figuras 102 y 103.

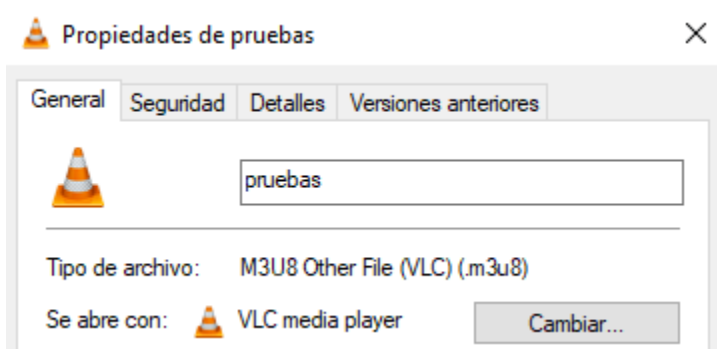


Fig. 102 Pantalla de VLC Media Player abriendo un archivo M3U8.

Nota. Captura de pantalla realizada por el autor durante el proceso de apertura de un archivo M3U8 en VLC.

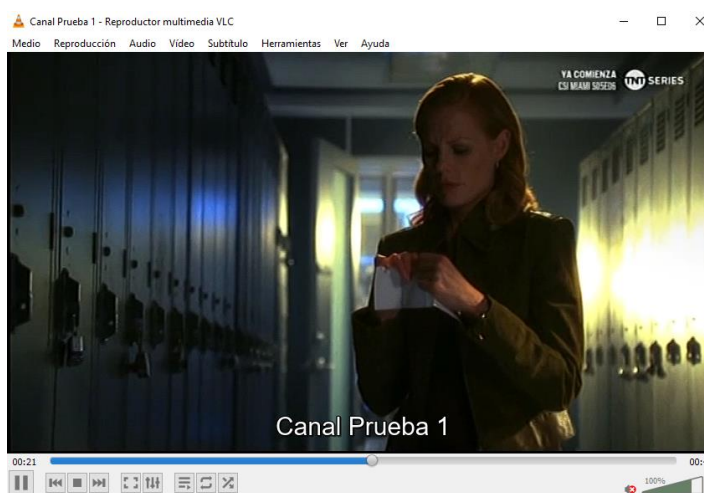


Fig. 103 VLC Media Player reproduciendo el canal “TNT Series” desde la lista M3U8 generada.

Nota. Captura de pantalla realizada por el autor durante la reproducción del canal TNT Series en el programa VLC Media Player, el contenido audiovisual mostrado pertenece a TNT Series y a sus respectivos titulares de derechos. La imagen se presenta únicamente con fines demostrativos.

Una vez agregada la lista M3U8 se puede visualizar dentro de la aplicación en el apartado de canales, véanse las Figuras 104 y 105.

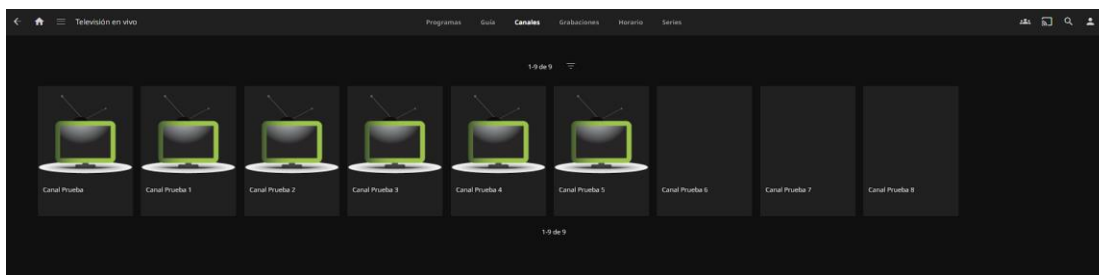


Fig. 104 Pantalla de la sección “Canales” del panel de administración de Jellyfin, donde se listan las fuentes de TV configuradas.

Nota. Captura de pantalla realizada por el autor durante la exploración de la interfaz de Jellyfin.

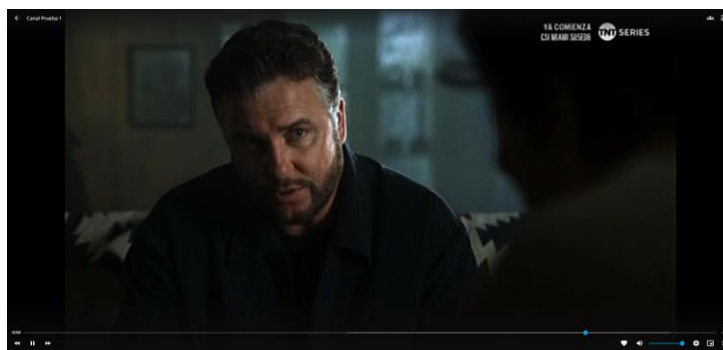


Fig. 105 Reproducción de un canal en Jellyfin desde una lista M3U8.

Nota. El contenido audiovisual mostrado pertenece a TNT Series y a sus respectivos titulares de derechos.

La imagen se presenta únicamente con fines demostrativos para mostrar la interfaz del reproductor de Jellyfin.

En estos casos, es posible asociar la lista M3U8 con metadatos, tales como la programación y un EPG (Electronic Program Guide), los cuales generalmente se encuentran en archivos XML, como se muestra en la Figura 106.

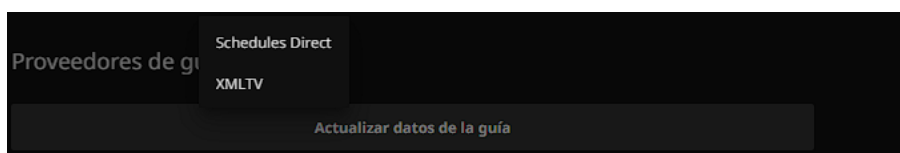


Fig. 106 Configuración de proveedor de guía (EPG) en Jellyfin.

Nota. Captura de pantalla realizada por el autor durante la configuración de la fuente de guía de programación en Jellyfin.

Asimismo, es posible agregar usuarios y establecer un control de acceso al contenido, permitiendo gestionar los permisos según el perfil de cada usuario, como se muestra en las Figuras 107 y 108.

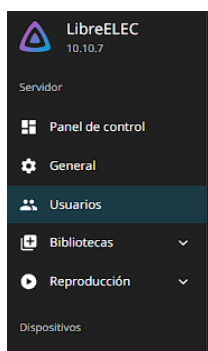


Fig. 107 Menú lateral de Jellyfin con la sección de usuarios.

Nota. Captura de pantalla realizada por el autor durante la navegación por el menú lateral de Jellyfin.

Fig. 108 Pantalla para agregar un nuevo usuario en Jellyfin.

Nota. Captura de pantalla realizada por el autor durante el proceso de creación de usuarios en Jellyfin.

De manera sencilla, es posible otorgar o revocar el acceso al contenido disponible, incluyendo tanto bibliotecas predeterminadas como contenido personalizado, tales como películas, series y otros medios.

2.2.2 Declaración de Uso Académico y Exclusión de Responsabilidad

El presente proyecto se desarrolló exclusivamente con fines académicos y de investigación tecnológica. El prototipo de transmisión de video desarrollado en este trabajo se empleó únicamente para pruebas técnicas y para evaluar el comportamiento de las herramientas y protocolos involucrados en el proceso de codificación y distribución de video sobre redes IP.

Las señales de televisión utilizadas durante las pruebas fueron empleadas únicamente como material de referencia para validar la funcionalidad del sistema, sin perseguir fines de retransmisión pública, explotación comercial o distribución no autorizada de contenido protegido por derechos de autor.

El autor de este trabajo no promueve ni respalda el uso del prototipo para actividades que vulneren la legislación vigente, los derechos de propiedad intelectual, las concesiones de radiodifusión ni los lineamientos de los proveedores de contenido. Cualquier uso distinto al estrictamente académico descrito en este documento es responsabilidad exclusiva del usuario final.

Asimismo, este proyecto no tiene como propósito reemplazar, replicar o intervenir en servicios de distribución audiovisual autorizados, sino demostrar de manera controlada el funcionamiento de tecnologías como protocolos y codificadores de video en un entorno de aprendizaje.

Capítulo III. Resultados

Resultados

Los resultados obtenidos con base a los objetivos planteados de este proyecto han sido favorables, dado que se logró desarrollar un sistema funcional, que ofrece una experiencia satisfactoria al usuario, permitiéndole acceder al contenido requerido desde cualquier ubicación con conexión a internet en un dispositivo compatible.

Los equipos utilizados se integraron adecuadamente y trabajaron de manera continua, cumpliendo con el propósito inicial del proyecto. Se identificaron algunas interrupciones externas, como fallas en el origen de los canales o interferencias solares, que afectan la transmisión, sin embargo, estos factores no resultan críticos y pueden considerarse normales.

El ámbito del streaming es amplio y comprende numerosas especializaciones, por lo que alcanzar un nivel experto requiere profundizar en múltiples áreas. No obstante, con un conocimiento básico y la implementación de conceptos esenciales, es posible desarrollar un sistema funcional que cumpla con los requerimientos necesarios.

Otro aspecto a considerar es el marco legal de las transmisiones, ya que estas se encuentran reguladas y los contenidos están protegidos por derechos de autor, requiriendo autorización o pago para su difusión. Asimismo, el uso de software de código abierto implica cumplir con los términos de la licencia correspondiente especialmente en casos de uso comercial.

Aunque existen herramientas comerciales y de código abierto que ofrecen funcionalidades similares e incluso más avanzadas, el prototipo desarrollado es adecuado para introducirse en el mundo de la distribución de video a través de internet, comprender su complejidad y de manera potencial aportar a soluciones existentes.

Para evaluar la eficiencia del protocolo HLS en la visualización de contenido, se diseñó un formulario que consideró distintos aspectos de la reproducción de video. Las listas M3U8 fueron elaboradas por el autor y Jellyfin se utilizó como plataforma de distribución y reproducción de los flujos de video. Los datos obtenidos fueron empleados únicamente con fines académicos.

Las respuestas se evaluaron por medio de la escala Likert de cinco puntos, con nueve afirmaciones siendo aplicadas a doce usuarios. Donde el valor de uno corresponde a total insatisfacción y cinco a totalmente satisfecho.

Los aspectos evaluados fueron el inicio rápido del video, fluidez de reproducción, calidad de imagen, estabilidad de la transmisión, sincronización de audio, respuesta de la barra de progreso, ausencia de artefactos visuales y experiencia general de visualización.

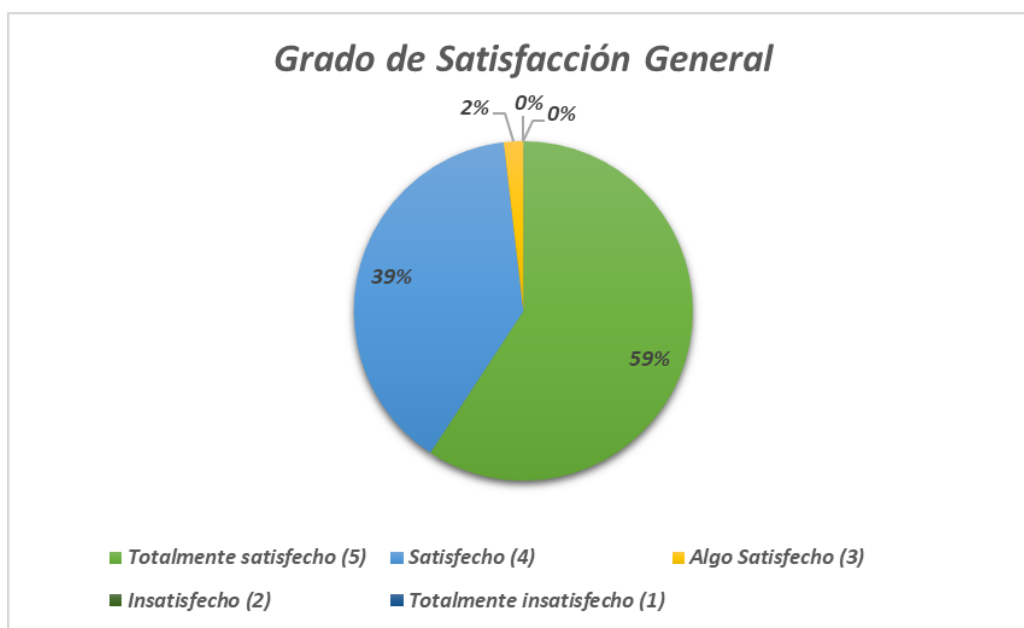


Fig. 109 Gráfica de Satisfacción general usando la escala Likert.

Fuente: Elaboración propia.

Se puede observar que la mayoría de usuarios se encuentra totalmente satisfecho en la gráfica general como se muestra en la Figura 109.

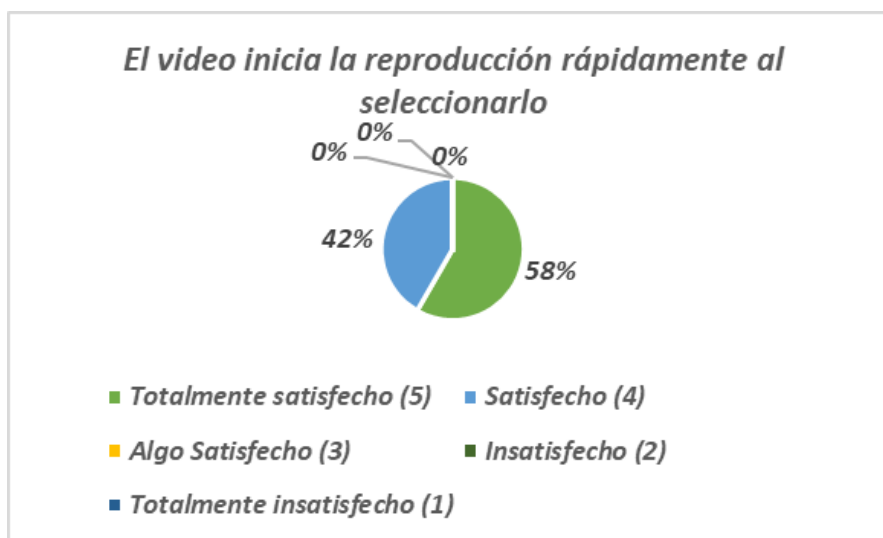


Fig. 110 Gráfica de rapidez en reproducción.

Fuente: Elaboración propia.

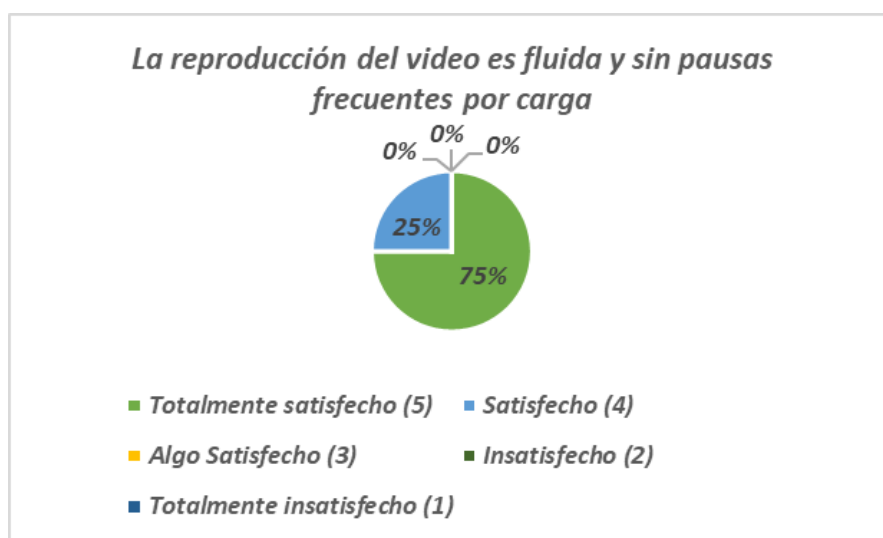


Fig. 111 Gráfica de reproducción fluida

Fuente: Elaboración propia.

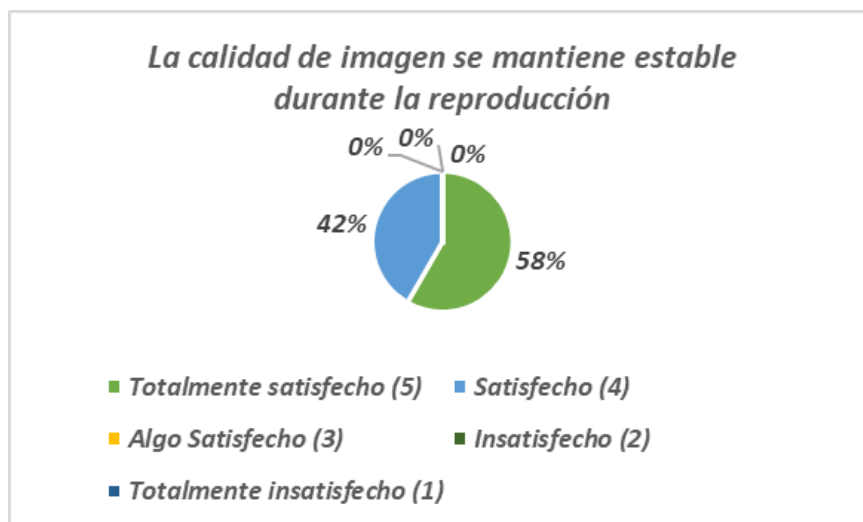


Fig. 112 Gráfica de calidad de imagen.

Fuente: Elaboración propia.

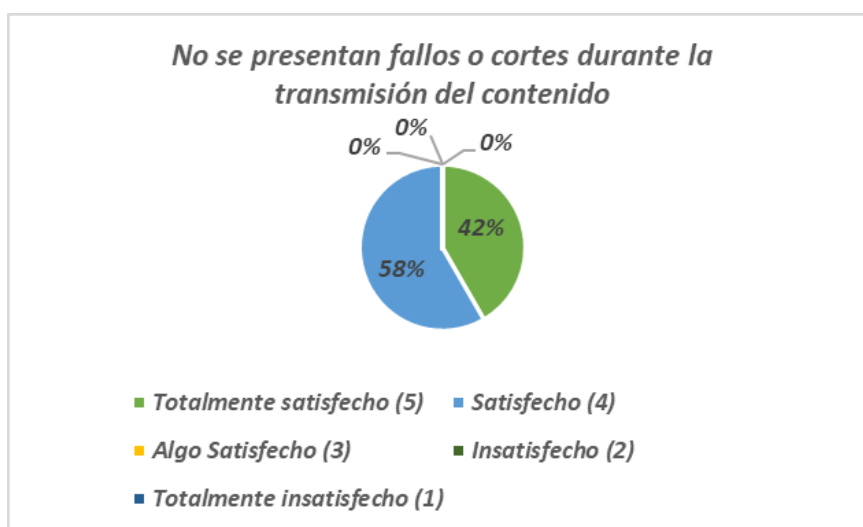


Fig. 113 Gráfica de fluidez de la transmisión.

Fuente: Elaboración propia.

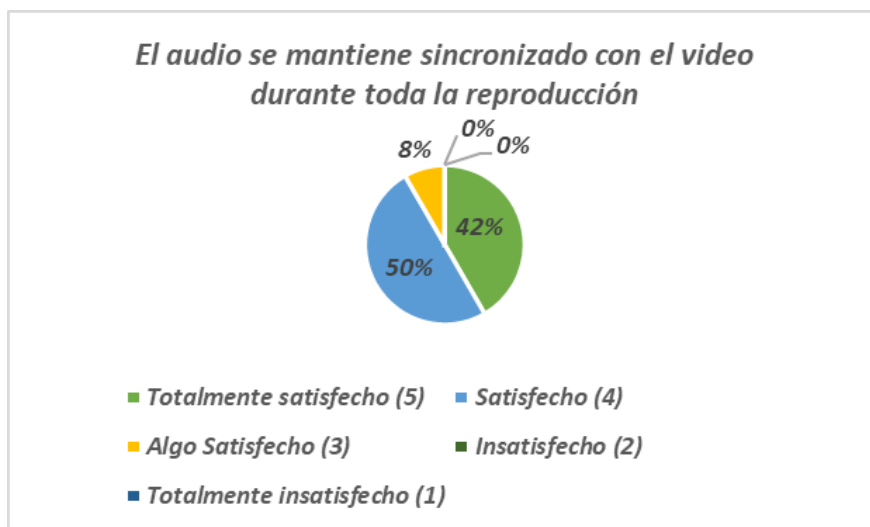


Fig. 114 Gráfica de sincronización de audio.

Fuente: Elaboración propia.

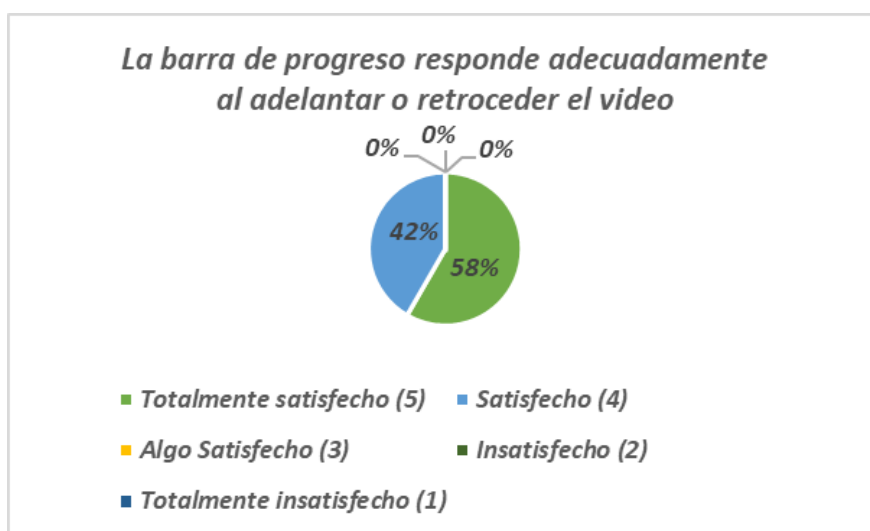


Fig. 115 Gráfica de respuesta de la barra de progreso.

Fuente: Elaboración propia.

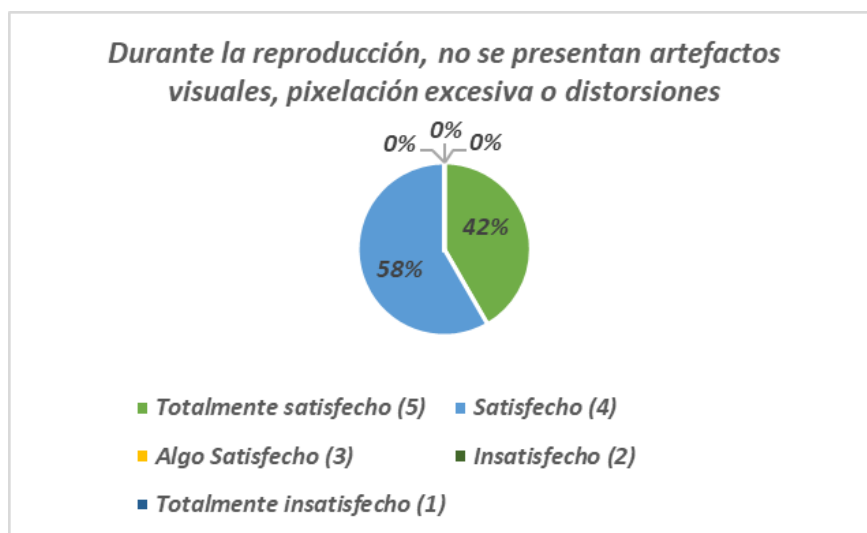


Fig. 116 Gráfica de artefactos visuales

Fuente: Elaboración propia.

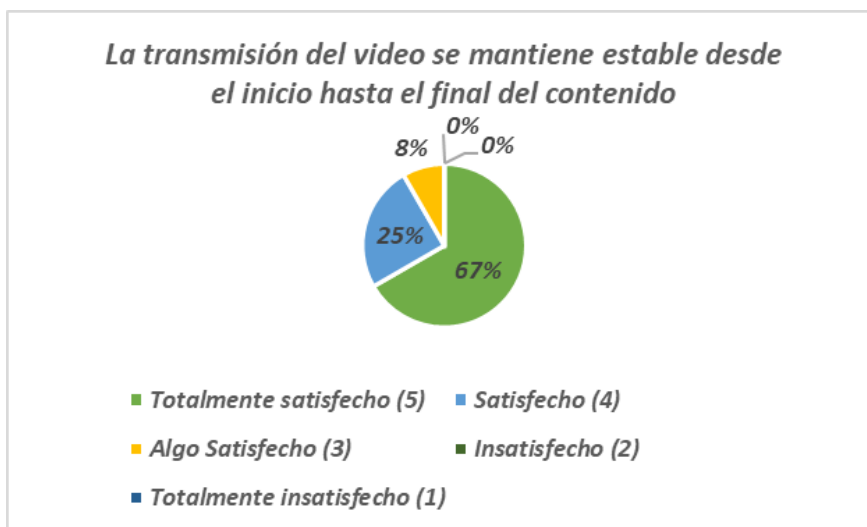


Fig. 117 Gráfica de estabilidad de la transmisión.

Fuente: Elaboración propia.

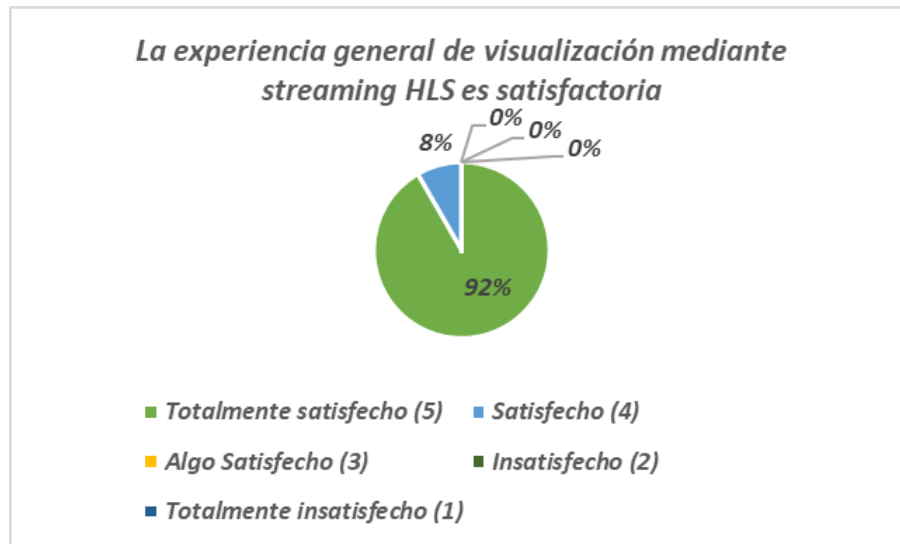


Fig. 118 Gráfica de experiencia general de visualización.

Fuente: Elaboración propia.

Los resultados muestran que la mayoría de los usuarios percibieron que el video inicia rápidamente, la calidad de imagen se mantiene estable, la reproducción es fluida, el audio se encuentra correctamente sincronizado, la respuesta de la barra de progreso es adecuada, no se presentan demasiados artefactos visuales, la transmisión se mantuvo estable de inicio a fin y en general la experiencia del usuario es positiva como se muestra en las Figuras 110-118.

En general, la mayoría de los usuarios calificó la experiencia de visualización como totalmente satisfactoria.

La evaluación mediante la escala Likert y el análisis de la experiencia de usuario reflejan que la interfaz es intuitiva, la reproducción es fluida, la calidad de imagen y audio se mantienen estables, y la navegación es satisfactoria, confirmando que el sistema ofrece una experiencia positiva para el usuario final.

Conclusiones y trabajos futuros

El desarrollo de este proyecto permitió aplicar el conocimiento adquirido en muchas áreas ya que va desde redes, multimedia, programación, sistemas operativos entre otras asignaturas que han sido de mucha ayuda para comprender la complejidad de un sistema de transmisión de video.

A lo largo del proceso, se adquirieron aprendizajes adicionales mediante prueba y error, los cuales habrían sido difíciles de obtener de otra manera, por lo que desarrollar este proyecto contribuyó significativamente a mi experiencia tanto profesional como personal.

El desarrollo del prototipo permitió comprobar que la infraestructura implementada es capaz de manejar transmisiones de video en tiempo real de manera estable y continua. La correcta instalación y funcionamiento de FFmpeg, junto con la configuración de servicios en segundo plano y la validación del acceso a los flujos mediante una aplicación con autenticación básica, demuestra que los objetivos planteados fueron cumplidos y que el enfoque propuesto es viable para escenarios de distribución de video a través de internet.

A partir de los resultados obtenidos, se identificaron áreas de oportunidad y de mejora para futuros desarrollos. Entre los trabajos futuros se incluyen la personalización de la interfaz de usuario para el gusto del espectador, la personalización de listas M3U8 con programación electrónica y de este modo el televidente pueda saber que programa sigue. El uso de un dominio personalizado para la facilidad de conexión del usuario y por último la mejora de hardware para poder soportar una mayor cantidad de transmisiones de manera simultánea.

Referencias

ABC News. Recuperado el 7 de noviembre de 2025, de ABC news live reports <https://abcnews.go.com/Live>

Adobe. (2024, mayo 30). *Acerca de la compresión y codificación de vídeo y de audio*. Adobe.com. Recuperado el 17 de noviembre de 2025, de <https://helpx.adobe.com/la/media-encoder/using/video-audio-encoding-compression.html>

ATDT. (s/f). *Agencia de Transformación Digital y Telecomunicaciones (ATDT)*. gob.mx. Recuperado el 7 de noviembre de 2025, de <https://datos.gob.mx/organization/about/atdt>

Apuntes de Electrónica – UTN. (s.f.). *Antenas*. Recuperado el 4 de noviembre de 2025, de <http://www1.frm.utn.edu.ar/comunicaciones/antenas.html>

Akamai.com. (s.f) *¿Qué es el streaming de vídeo en directo?* Recuperado el 5 de noviembre de 2025, de <https://www.akamai.com/es/glossary/what-is-live-video-streaming>

ATSC. (s. f.). *About ATSC*. ATSC. Recuperado el 8 de noviembre de <https://www.atsc.org/about/>

AWS. (s. f.). *Tablas de referencia de códecs y contenedores admitidos*. Recuperado el 8 de noviembre de 2025, de https://docs.aws.amazon.com/es_es/mediaconvert/latest/ug/supported-containers-codecs-details.html

Boada, D. (2025, febrero 9). *Protocolo TCP: definición y funcionamiento*. Tutoriales Hostinger. <https://www.hostinger.com/mx/tutoriales/protocolo-tcp>

Boada, D. (2025, junio 5). *¿Qué es la IPTV, cómo funciona y cuáles son sus ventajas y desventajas?* Tutoriales Hostinger. <https://www.hostinger.com/es/tutoriales/que-es-la-iptv>

Broadcast Management Group. (2022, diciembre 28). *What is OTT?*
<https://broadcastmgmt.com/ott-networks/what-is-ott/>

Optica Naranjo. (2022, septiembre 30). *Caja NAP de Fibra Óptica: ¿Qué es y para qué sirve?* <https://opticanaranjo.com.ar/caja-para-fibra-optica/>

Cisco. (s.f.). *¿Cómo funciona un switch?* Recuperado el 7 de noviembre de 2025, de https://www.cisco.com/c/es_mx/solutions/small-business/resource-center/networking/network-switch-how.html

Cisco. (s.f.). *¿Qué es un router?* Recuperado de el 9 de noviembre de 2025, de https://www.cisco.com/c/es_mx/solutions/small-business/resource-center/networking/what-is-a-router.html

Cloudflare. (s.f.). *What is HTTP Live Streaming (HLS).* Recuperado de el 9 de noviembre de 2025, de <https://www.cloudflare.com/es-es/learning/video/what-is-http-live-streaming/>

Cloudflare. (s.f.). *What is a protocol? | Network protocol definition.* Recuperado el 3 de noviembre de 2025, de <https://www.cloudflare.com/es-es/learning/network-layer/what-is-a-protocol/>

Cloudflare. (s.f.). *¿Qué es el UDP?* Recuperado el 4 de noviembre de 2025, de <https://www.cloudflare.com/es-es/learning/ddos/glossary/user-datagram-protocol-udp/>

Comisión Reguladora de Telecomunicaciones. (s/f). *¿Qué hacemos?* Recuperado el 8 de noviembre de 2025, de <https://www.gob.mx/crt/es/que-hacemos>

De Luz, S. (2025, marzo 4). *Aprende a crear tus propios servicios en Linux usando Systemd.* RedesZone. Recuperado el 10 de noviembre de 2025, de <https://www.redeszone.net/tutoriales/servidores/administrar-servicios-Linux-systemd/>

De Luz, S. (2025, mayo 25). *Aprende qué es un armario rack y qué modelos puedes comprar*. Recuperado el 18 de noviembre de 2025, de RedesZone. <https://www.redeszone.net/tutoriales/redes-cable/que-es-armario-rack-modelos/>

DIGI International. (s.f) *QAM (Modulación de amplitud en cuadratura)*. Digi.com. Recuperado el 5 de noviembre de 2025, de <https://es.digi.com/resources/definitions/qam>

Ecured. (s.f.). *OBS Studio*. Recuperado el 9 de noviembre de 2025, de: https://www.ecured.cu/OBS_Studio

Outlet Audio Video. (s.f). *Encoder / Decoder Video*. Recuperado el 8 de noviembre de 2025, de <https://outletaudiovideo.com/collections/encoder-decoder-video>

FFmpeg. (s. f.). *About FFmpeg*. Recuperado el 9 de noviembre de 2025, de: <https://ffmpeg.org/about.html>

Fortinet (s/f). *¿Qué es una dirección IP? ¿Cómo funciona?* Recuperado el 5 de noviembre de 2025, de <https://www.fortinet.com/lat/resources/cyberglossary/what-is-ip-address>

FiberWDM. (2024, junio 13). *What is EDFA, how does EDFA work and what are the types of EDFA?* FiberWDM. Recuperado el 5 de noviembre de 2025, de https://es.fiberwdm.com/blog/what-is-edfa-how-does-edfa-work-and-what-are-the-types-of-edfa_b156

GeeksforGeeks. (2023, marzo 30). *Software Prototyping Model and Phases*. Recuperado el 14 de noviembre de 2025, de <https://www.geeksforgeeks.org/software-engineering/software-prototyping-model-and-phases/>

GoDaddy, E. de C. (2024, octubre 25). *Qué es un servidor y cómo funciona: Tipos, características y usos principales*. GoDaddy Resources - LATAM; GoDaddy. <https://www.godaddy.com/resources/latam/digitalizacion/servidor-que-es-tipos>

Haivision. (s.f.). *Haivision SRT Gateway*. Recuperado el 08 de noviembre de 2025, de: <https://www.haivision.com/products/srt-gateway/>

Hernández Palacios, R., & Hernández Hernández, G. (2017). *Comunicaciones multicast*. Ciencia Huasteca Boletín Científico De La Escuela Superior De Huejutla, 5(9). Recuperado el 5 de noviembre de 2025, de <https://doi.org/10.29057/esh.v5i9.2216>

IFT. (s.f.). *¿Qué es el IFT?* ift.org.mx. Recuperado el 5 de noviembre de 2025, de <https://www.ift.org.mx/conocenos>

Kodi. (s.f.). *Acerca de Kodi*. Recuperado el 08 de noviembre de 2025, de <https://kodi.tv/about/>

Jellyfin (s.f.). *The free software media system*. Recuperado el 5 de noviembre de 2025, de <https://jellyfin.org/>

Castr. (2020, octubre 19). *What is VOD? All you need to know about video on demand*. *Castr's Blog*; Castr. Recuperado el 7 de noviembre de 2025, de <https://castr.com/blog/what-is-vod/>

LibreELEC. (s/f.). *About*. Recuperado el 8 de noviembre de 2025, de <https://libreelec.tv/about/>

Liyanage, A. (2025, mayo 30). *El modelo de prototipo en SDLC: construir antes de construir*. Medium. Recuperado el 14 de noviembre de 2025, de <https://medium.com/@nld.anuradha/the-prototype-model-in-sdlc-building-before-building-5226ab37a2c7>

López, T. (2025, febrero 25). *Hablemos de antenas: su historia y evolución*. Radionotas. Recuperado el 11 de noviembre de 2025, de <https://radionotas.com/2025/02/25/hablemos-de-antenas-su-historia-y-evolucion/#:~:text=La%20historia%20de%20las%20antenas,el%20te%C3%B3rico%20James%20Clerk%20Maxwell>.

Martins, J. (2025, enero 19). *¿Qué es la metodología Kanban y cómo funciona?* Asana. Recuperado el 10 de noviembre de 2025, de <https://asana.com/es/resources/what-is-kanban>

Martins, J. (2025, febrero 15). *Scrum: conceptos clave y cómo se aplica en la gestión de proyectos*. Asana. Recuperado el 8 de noviembre de 2025, de <https://asana.com/resources/what-is-scrum>

Medium.com (2018, noviembre 13). *Una visión general de la red de acceso FTTH con GPON*. Recuperado el 5 de noviembre de 2025, de <https://franklinnn.medium.com/una-visi%C3%B3n-general-de-la-red-de-acceso-ftth-con-gpon-104bc8973d65>

Montoro, R. (2025, junio 18). *El auge del streaming: por primera vez, la transmisión online supera al cable y a la señal abierta, según Nielsen*. Infobae. <https://www.infobae.com/tecno/2025/06/19/el-auge-del-streaming-por-primera-vez-la-transmision-online-supera-al-cable-y-a-la-senal-abierta-segun-nielsen/>

Muguirra, A. (s. f.). *¿Qué es la escala de Likert y cómo utilizarla?* QuestionPro. Recuperado el 20 de noviembre de 2025, de <https://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/>

Nielsen, F. (2025, junio 24). *Qué es un códec de vídeo: Todo lo que debe saber*. Dacast; Dacast - OTT streaming platform. Recuperado el 5 de noviembre de 2025, de <https://www.dacast.com/es/blog-es/que-es-un-codec-de-video/>

NVIDIA (s.f) *Video encode and decode support matrix*. Developer. Recuperado el 5 de noviembre de 2025, de <https://developer.NVIDIA.com/video-encode-decode-support-matrix>

OEM-Laser. (2023, marzo 15). *What is fiber combiner?* OEM-Laser. Recuperado el 5 de noviembre de 2025, de <https://www.oem-laser.com/info/what-is-fiber-combiner-83349995.html>

Open Broadcaster Software. (s/f). *OBS Studio*. Recuperado el 10 de noviembre de 2025, de <https://obsproject.com/>

Prensario Internacional. (s.f.). *Los canales de televisión lineal son una vía de crecimiento en la CEE*. Recuperado el 12 de noviembre de 2025, de <https://www.prensario.net/Los-canales-de-television-lineal-son-una-via-de-crecimiento-en-la-CEE-43563.note.aspx>

Premlink. (s.f.). *Superposición de RF en PON*. Recuperado el 4 de noviembre de 2025, de <https://www.premlink.net/es/superposicion-de-rf-en-pon/>

Raspberry Pi Foundation. (s. f.). *Raspberry Pi*. Recuperado el 15 de noviembre de 2025, de <https://www.raspberrypi.com/>

Rodríguez Solís, A. N. (2020, mayo 31). *Creación de videos digitales*. Universidad Autónoma del Estado de Hidalgo. Recuperado el 7 de noviembre de 2025, de <https://repository.uaeh.edu.mx/bitstream/items/3dbc30e2-987d-4149-b566-97f551b37e8f/creacion-videos-digitales.pdf>

Ruether, T. (2023, febrero 21). Historia de la transmisión multimedia. Wowza Media Systems. <https://www.wowza.com/blog/history-of-streaming-media#h-1990-2000-streaming-debuts-as-a-proof-of-concept>

Samim Group. (2024, julio 24). *What is Digital Video Broadcasting (DVB)*. Recuperado el 8 de noviembre de 2025, de <https://www.samimgroup.com/blog/what-is-digital-video-broadcasting/>

Softvelum. (s.f.). *Larix Broadcaster*. Recuperado el 5 de noviembre de 2025, de <https://softvelum.com/larix/>

Sole, R. (2025, junio 16). *Tu gráfica NVIDIA tiene cientos, ¿sabes qué son los núcleos CUDA?* HardZone. <https://hardzone.es/marcas/NVIDIA/nucleos-cuda/>

StudioCoast Pty Ltd. (s.f.). *vMix – Live Production Software*. Recuperado el 8 de noviembre de 2025, de <https://www.vmix.com/software/features.aspx>

Tanenbaum, A. S., & Wetherall, D. J. (2012). *Redes de computadoras (5.ª ed.)*. Pearson. Recuperado el 3 de noviembre de 2025.

TecnoLibre. (2024, julio 19). *¿Qué es CATV?*. TecnoLibre. Recuperado el 5 de noviembre de 2025, de <https://tecnolibre.net/2024/07/19/que-es-catv/>

TDT (s.f.). *¿Qué es la TDT?*. Recuperado el 8 de noviembre de 2025, de https://www.tdt.mx/que_es_tdt.php

Themoviedb.org (s.f.). *The Movie Database (TMDB)*. Recuperado el 15 de noviembre de 2025, de <https://www.themoviedb.org/>

UNCOSU. (s.f.). *Del tejado al satélite: la historia del cable que conectó al mundo*. Recuperado el 11 de noviembre de 2025, de <https://uncosu.gob.gt/2025/06/25/del-tejado-al-satelite-la-historia-del-cable-que-conecto-al-mundo/>

Vsolcn.com (2025, enero 13) *¿Qué es OLT (Optical Line Terminal)? Características y Aplicaciones*. Recuperado el 7 de noviembre de 2025, de <https://es.vsolcn.com/blog/what-is-olt.html>

VSOL. (2025, enero 14). *¿Qué es ONU? Características, beneficios y tipos explicados*. Recuperado el 8 de noviembre de 2025, de <https://es.vsolcn.com/blog/what-is-onu.html>

Whitehead, J. (2025, agosto 19). *Guía completa de los archivos M3U8: Comprender, Crear y Utilizar*. Dacast. Recuperado el 8 de noviembre de 2025, de <https://www.dacast.com/es/blog-es/M3U8-file/>