UNIVERSIDAD POLITECNICA DE VALENCIA

DEPARTAMENTO DE COMUNICACIONES

Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones

SISTEMA PARA LA INTEGRIDAD DEL VÍDEO EN SISTEMAS DE STREAMING DASH UTILIZANDO BLOCKCHAIN

Autor: Juan Daniel Euceda Pastrana

Director 1: Juan Carlos Guerri Cebollada

Director 2: Pau Arce Vila

Fecha de comienzo: 09/01/2020

Lugar de trabajo: Grupo de Comunicaciones Multimedia del iTEAM

Objetivos — Este trabajo final de máster tiene como objetivo desarrollar un prototipo de sistema para la verificación de la integridad del vídeo utilizando los conceptos de la tecnología blockchain y centrándose especialmente en las características del multimedia *streaming*, lo que genera un método muy innovador de comprobación de vídeos de calidad fija o variable. Para la consecución del objetivo principal, el análisis de resultados y evaluación de prestaciones del sistema se plantean los siguientes objetivos específicos:

- Estudiar las características del vídeo *streaming* en formato DASH.
- Estudiar el concepto y funcionamiento de la tecnología blockchain.
- Conocer el estado del arte de la tecnología blockchain en los diversos tipos de contenido multimedia.
- Desarrollar un sistema de verificación de vídeo adaptativo en lenguaje Python bajo el concepto de bloques referenciados.
- Evaluar las prestaciones del código generado y los resultados obtenidos.
- Ejemplificar escenarios donde el sistema puede ser utilizado y definir líneas de trabajo futuras a través de posibles adaptaciones.

Metodología — El trabajo presentado ha sido desarrollado a través de una metodología clásica de investigación aplicada, donde en primera instancia se realiza un estudio teórico de los temas generales a tratar, subsiguientemente se analizan trabajos o soluciones existentes relacionadas al campo que se ha definido con el fin de tener una perspectiva del estado del arte para que, posteriormente, se planteen mecanismos o estrategias que permitan lograr un objetivo concreto hacia una rama poco tratada para que el trabajo represente una solución innovadora y de utilidad aunque el tipo de ámbito al que es aplicado sea muy concreto, determinado y delimitado.

Desarrollos teóricos realizados — En la elaboración de este trabajo final de máster no hubo desarrollo teórico pues, en la elaboración del prototipo del sistema propuesto para la verificación del vídeo por segmentos, se utilizan conceptos y tecnologías existentes cuya definición se valoró para entender y acoplar las correspondientes características al funcionamiento del programa.

Desarrollo de prototipos y trabajo de laboratorio — Se crearon un conjunto de métodos, desarrollados en lenguaje *Python,* que componen a un prototipo de sistema capaz de analizar los segmentos de audio y vídeo transmitidos en una comunicación *streaming* entre cliente y servidor gracias a un log generado por el explorador utilizado. El sistema está compuesto por tres módulos, uno para el análisis y la creación del blockchain en el emisor, otro para el análisis y la creación del blockchain en el receptor y, por último, un módulo de comparación de las cadenas creadas en los apartados anteriores y que es el encargado de verificar la integridad de estas.

Resultados — Los resultados obtenidos fueron favorables para todas aquellas pruebas en las que se registró una reproducción continua de forma íntegra, también se obtuvo un porcentaje total de acierto en la detección de fallos para aquellas reproducciones continuas premeditadamente alteradas. Aunque el correcto funcionamiento para las reproducciones con un adelanto o retraso no era un objetivo específico de esta tesis, se realizaron pruebas donde se concluye que el sistema no es aplicable en estos escenarios.

Líneas futuras — El presente trabajo final de máster deja abierta la posibilidad de corregir los escenarios en los cuales el sistema no es aplicable actualmente, esto será posible mediante la creación de una herramienta de modelado de audio y vídeo que permita la reconstrucción del contenido de forma exacta, aunque exista un adelanto o retraso en su reproducción, de esa forma se garantizaría una funcionalidad completa, independientemente de la forma en la que se haya percibido el contenido. Como caso ideal de funcionamiento, se podría trabajar directamente en la estructura del reproductor de forma que los módulos realicen el trabajo de comparación de segmentos en tiempo real, aplicando al segmento que llegue al receptor el procedimiento de creación de bloque y hash final para ser comparado inmediatamente como si este fuese el último de la cadena y así no sea necesario la reconstrucción del contenido, lo que traería consigo un mejor rendimiento general y una retroalimentación inmediata. También, una vez finalizado el proceso de defensa, se estableció como línea futura la elaboración de un artículo con el objetivo de que el trabajo realizado sea publicado en alguna revista científica.

Publicaciones — Una lista de vídeos tutoriales fueron publicados en YouTube con el objetivo de dar una mejor explicación del proceso de prueba y muestra de resultados.[1]

Abstract — Since the emergence of the internet in the 90's, the number of people with access to it has been constantly growing, the subsequent appearance of smartphones, social networks and multimedia services has turned networks into a coming and going of image, audio and video. That kind of content currently represents close to 80% of internet traffic, reason that explain why video on demand platforms expect to have nearly of 571 million of users in 2025.

This uncontrollable growing of users and media content, involve the possibility to watch adulterated videos, listen a fake audio or see in a photo something that never happened. Despite of some authenticity multimedia programs existence, no one of them do reference about variable quality videos, a characteristic of streaming communications. Platforms like Netflix, Amazon Prime, HBO Go, YouTube and others implement streaming technology using dynamic adaptative streaming over HTTP or dash standard, a protocol that split videos in segments with same duration and different qualities for offered a fluid's quality of experience to user according his bandwidth.

The objective of this master thesis is proposing a method to verify the integrity of streamed audio and video content over DASH communications, for that, an innovative multimedia blockchain system was development in Python language. This system analyzes every multimedia segment involved in the streaming session, based on this information, a blockchain is constructed for the transmitter and other for the receiver. Both blockchain comparison will give a conclusion about the integrity of the content that the final user watched. This program demonstrated to be assertive for all scenarios where the content was watched in continuous play, also, when premeditatedly a segment was altered, the program showed the specific segment with alteration. For scenarios with forward or regression the system is not applicable.

Autor: Juan Daniel Euceda Pastrana, email: juaeupas@teleco.upv.es
Director 1: Juan Carlos Guerri Cebollada, email: jcguerri@dcom.upv.es
Director 2: Pau Arce Vila, email: paarvi@iteam.upv.es
Fecha de entrega: 10-09-20

ÍNDICE

I. INTRODUCCIÓN

A través de los años hemos sido testigos de cómo el acceso a internet es cada vez más común, la diversidad con la que actualmente se puede tener acceso a una red de datos cableada o inalámbrica hace que, actualmente, sea una herramienta casi indispensable para la ejecución de algunas actividades diarias, sean estas de índole laboral, educativo, medio de comunicación, ocio o cualquier otro ámbito en general y aunque se mantiene una brecha digital bastante amplia en algunos países, según datos del Banco Mundial, desde 2017 cerca del 50% de la población posee acceso a internet, un número que seguramente irá en ascenso tal y como se ha comportado desde su irrupción en la década de los 90's. [2]

La capacidad que nos genera el acceso a internet de tener tanta información a disposición, interconectar millones de personas y empresas entre sí, subir cualquier tipo de contenido a la red con poca o ninguna restricción y a un clic de distancia, nos deja una pregunta intrínseca: ¿Qué es lo que pasa diario en internet?

Sin duda esta pregunta representa un aspecto de mucha relevancia para analizar pues, a partir de tal conocimiento, un proveedor de servicio puede dimensionar su red, un operador de telefonía móvil puede adecuar sus planes prepago o post pago según segmentos de mercado específicos, un emprendedor puede identificar una oportunidad de negocio o un estudiante puede enfocar sus estudios, es por ello que, con el fin de mostrar una idea global a la pregunta antes expuesta, empresas como DOMO Inc., Statista, Hootsuite y otras, nos muestran año con año un resumen de las principales actividades que se realizan en internet cada minuto, donde se destacan las búsquedas globales de información, consultas a repositorios de datos, uso de correo electrónico, servicios en línea, descarga de aplicaciones, el constante uso de redes sociales y servicios de streaming de audio y vídeo tal como se ejemplifica en la Fig. 1.

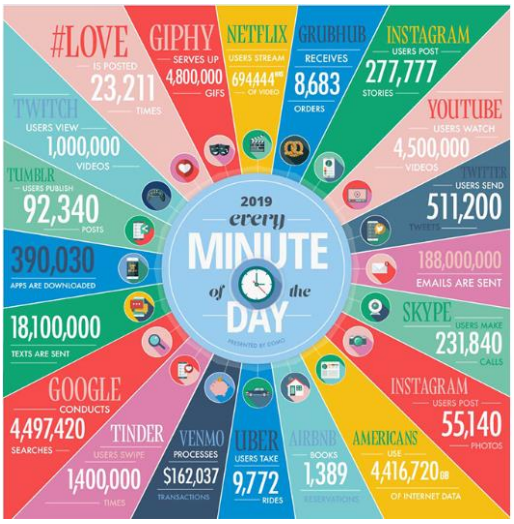

Fig.1. Lo que sucede en internet cada minuto año 2019. [3]

En general, se estima que el 80% del tráfico actual en internet representa a contenido multimedia, una cifra para nada extraña si consideramos la cantidad de vídeos que diariamente se suben y se reproducen en YouTube o la cantidad de fotos y vídeos que segundo a segundo son publicadas en redes como Facebook, Instagram, Snapchat o Tik Tok y qué decir del numeroso contenido existente en plataformas de streaming como Netflix, Amazon Prime Video, HBO Go y la expansión de la IPTV con transmisiones en vivo de canales de televisión o eventos deportivos por internet que hacen de la red un ir y venir constante de audio, imagen y vídeo.

Si bien es cierto, la satisfacción del espectador en cada una de estas plataformas está más ligada a la calidad con la cual recibe el contenido multimedia, la calidad de experiencia del mismo también se basa en la seguridad e integridad de la información que recibe y rara vez se preocupan de ello, lo que puede ser normal para un usuario común pero que es un aspecto de gran importancia para proveedores de servicios *Over The Top* (OTT), noticieros o sistemas de seguridad nacional donde se quiere evitar la desinformación y la viralización de *fake news*, es por ello que cada vez más este tipo de entes han enfocado sus recursos en garantizar la no manipulación del material audiovisual, lo que abre la posibilidad de generar soluciones innovadoras a un dominio de aplicación cada vez más extenso en el mundo del internet.

I.1. *MOTIVACIÓN*

A pesar de la existencia de métodos de verificación de vídeo relativamente útiles como las marcas de agua, huellas y firmas digitales inmersas dentro del propio contenido multimedia, éstas generan un grado de desconfianza al ser vulnerables a la alteración, pues no contienen una referencia original al inicio del contenido. Esa debilidad es debido a que la gran mayoría de cámaras comprimen el vídeo captado a través de algoritmos propios previamente integrados, sin embargo, tal contenido puede ser descomprimido, manipulado y luego comprimido nuevamente generando las marcas o firmas digitales normalmente y pasar desapercibido al no tener una referencia inicial que se rompa al rehacer dicho proceso. [4]

Una forma alternativa e innovadora en la cual se ha venido trabajando en el sector de las comunicaciones multimedia, con el objetivo de fortalecer la veracidad e integridad del contenido, es aplicar los conceptos de la tecnología blockchain a audio y vídeo. A pesar de relacionar blockchain inmediatamente con el sector financiero, por el auge adquirido a través de la aparición de las criptomonedas y demás servicios bancarios basados en él, su concepto como tal está ganando mercado en otros campos tales como el encriptado de datos, *Internet of Things (IoT)* y la transmisión de vídeo streaming, puesto que, con su implementación en la multimedia, se puede demostrar fácilmente la propiedad intelectual y al mismo tiempo proteger la rectitud de los datos. Debido a esa robustez y fiabilidad que ofrece la tecnología blockchain, al realizar una correlación de cada bloque desde el inicio hasta el fin de una cadena de transacciones o acciones, permite que

su adaptación no tenga que velar por los problemas que enfrentan las marcas, sellos y firmas digitales que genera una cámara convencional.

Actualmente se han desarrollado diferentes implementaciones de blockchain aplicado a contenido multimedia con resultados óptimos, sin embargo, ninguno hace referencia al vídeo streaming, un tipo de transmisión donde el mismo vídeo puede llegar al usuario en diversas calidades según lo determine el protocolo de comunicación, lo que agrega una complejidad a la comprobación de la veracidad del vídeo, razón por la cual este trabajo final de máster pretende desarrollar un sistema prototipo que determine la integridad del contenido generado bajo el estándar de streaming adaptativo (DASH) utilizando los conceptos de la tecnología blockchain.

I.2. *OBJETIVOS*

Este trabajo final de máster tiene como objetivo desarrollar un prototipo de sistema para la verificación de la integridad del vídeo utilizando los conceptos de la tecnología blockchain y centrándose especialmente en las características del multimedia streaming, lo que genera un método muy innovador de comprobación de vídeos de calidad fija o variable. Para la consecución del objetivo principal, el análisis de resultados y evaluación de prestaciones del sistema se plantean los siguientes objetivos específicos:

- Estudiar las características del vídeo streaming en formato DASH.
- Estudiar el concepto y funcionamiento de la tecnología blockchain.
- Conocer el estado del arte de la tecnología blockchain en los diversos tipos de contenido multimedia.
- Desarrollar un sistema de verificación de vídeo adaptativo en lenguaje Python bajo el concepto de bloques referenciados.
- Evaluar las prestaciones del código generado y los resultados obtenidos.
- Ejemplificar escenarios donde el sistema puede ser utilizado y definir líneas de trabajo futuras a través de posibles adaptaciones.

I.3. *METODOLOGÍA Y PLAN DE TRABAJO*

El trabajo presentado ha sido desarrollado a través de una metodología clásica de investigación aplicada, donde en primera instancia se realiza un estudio teórico de los temas generales a tratar, subsiguientemente se analizan trabajos o soluciones existentes relacionadas al campo que se ha definido con el fin de tener una perspectiva del estado del arte para que, posteriormente se planteen mecanismos o estrategias que permitan lograr un objetivo concreto hacia una rama poco tratada para que el trabajo represente una solución innovadora de utilidad aunque el tipo de ámbito al que es aplicado sea muy concreto, determinado y delimitado.

El plan de trabajo con el cual se fue desarrollando este proyecto se muestra en el correspondiente diagrama de Gantt en la Fig. 2.

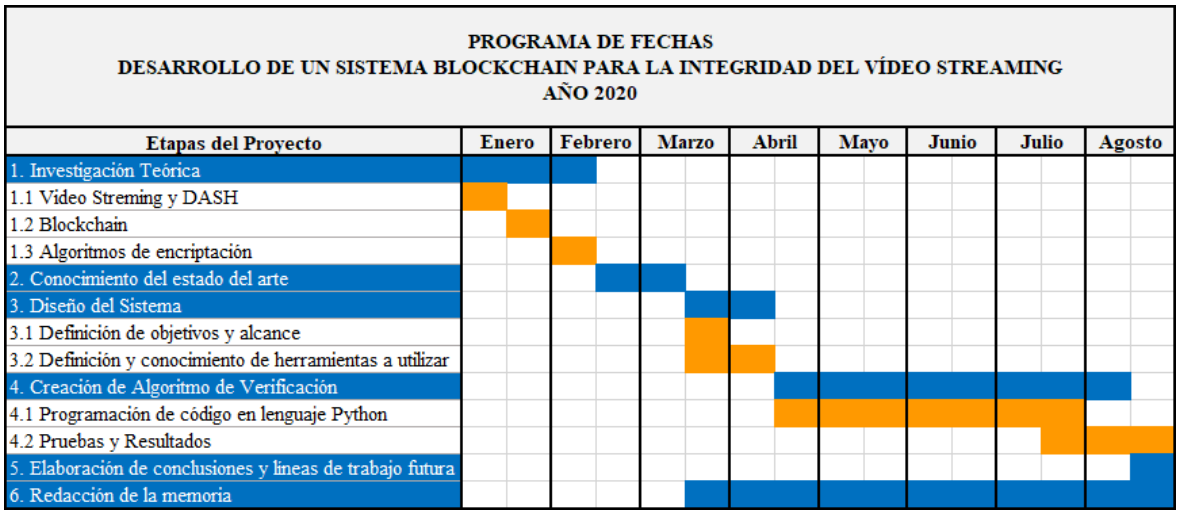

Fig.2. Diagrama de Gantt utilizado para la ejecución del TFM.

I.4. *ORGANIZACIÓN DEL DOCUMENTO*

En la sección II se definen detalladamente los fundamentos teóricos necesarios para comprender el desarrollo y estructura del sistema propuesto. Conceptos como multimedia streaming, protocolo DASH, blockchain y algoritmos de encriptación son abordados en las diferentes subsecciones de este capítulo.

En la sección III se resume el estado del arte en el que se encuentra el uso del blockchain en contenidos multimedia en general, en ella se describen algunas soluciones expuestas en diversas publicaciones y que sirvieron para definir el enfoque innovador del prototipo en esta tesis.

Posterior a una base teórica consolidada y conocimiento de soluciones previas, se desarrolla el capítulo IV con título Estructura del Sistema, donde a lo largo de las diferentes subsecciones se describe la premisa de diseño del sistema, la elaboración de la cadena de bloques referenciados o blockchain, funcionamiento de los módulos de emisor y receptor, además del método de comparación que es el responsable de presentar los resultados del cotejo.

La descripción del entorno en el que se ejecutan las pruebas y los diferentes escenarios comprobados son expuestos en la sección V, mientras que las limitantes y resultados obtenidos son enumerados en la sección VI.

En la séptima y última sección se presentan las respectivas conclusiones y líneas de trabajo futuro que dieron lugar a partir de toda la investigación elaborada.

II. FUNDAMENTOS TEÓRICOS

II.1. *MULTIMEDIA STREAMING*

Llamamos multimedia streaming a todo aquel tipo de contenido de audio y vídeo que es transmitido continuamente por internet en una comunicación cliente y servidor que, a diferencia de las descargas de contenido multimedia normales, donde se debe esperar la descarga total del archivo para su reproducción, en el streaming, el contenido se carga poco a poco en pequeños segmentos que se reproducen a medida van llegando al receptor, con el objetivo de generar una comunicación fluida y sobre todo más eficiente puesto que, podemos consumir el contenido desde nuestro navegador o aplicación sin la necesidad de guardar archivos de gran tamaño para una sola visualización, como puede pasar con la descarga de una película, lo que nos ayuda a mejorar el uso del espacio de almacenamiento en nuestros dispositivos, también si existen problemas en la conexión, evitamos la reanudación total de una descarga y solamente reiniciamos la transmisión al punto donde deseamos además, esta tecnología representa una ventaja de movilidad pues es posible disfrutar el contenido desde cualquier lugar y dispositivo con acceso a internet, mejorando de esa forma la calidad de experiencia del usuario.[5]

El funcionamiento del multimedia streaming se basa en la reproducción de audio o vídeo en el dispositivo de un usuario final, pero tal contenido está guardado en un servidor remoto donde al ser consumido se va dividiendo en pequeños segmentos de datos que se transmiten a través de internet en diferentes formatos y calidades según la negociación cliente-servidor, comúnmente llamada sesión streaming, parámetros como el protocolo de comunicación, formato, codificador y decodificador a utilizar se encuentran definidos en dicha sesión.

Este proceso de transmisión de datos definido por la sesión streaming, típicamente se divide en dos fases, la fase de almacenamiento de búfer y la fase de estado estacionario o *steady state phase*.

Durante la fase de almacenamiento de búfer se envían segmentos de datos a una tasa que es limitada al ancho de banda de la conexión entre el servidor remoto y el dispositivo del usuario final, al llegar a una cantidad de búfer disponible, que es definido en los parámetros de la sesión streaming al inicio de la negociación, se empieza a reproducir el contenido. Una vez que esto sucede, el reproductor comienza a solicitar y consumir los datos previamente almacenados en el búfer independientemente que este tenga o no contenido.

La fase de estado estable es la encargada de solicitar y transmitir segmentos de datos con el objetivo de mantener el búfer con la cantidad adecuada de contenido a ser consumida por el reproductor, para ello es necesario que la tasa de datos de reproducción sea al menos igual a la tasa de descarga de los segmentos así garantiza que el búfer tendrá siempre a disposición uno o más segmentos para reproducir, si esta relación, llamada ratio de acumulación, es superior a uno significa que el tamaño del búfer aumenta en la fase de estado estable, certificando de esa forma

una comunicación fluida aunque la red tuviese pequeñas fluctuaciones producto de congestión, *jitter* u otros factores, sin embargo, tampoco es conveniente utilizar un ratio de acumulación elevado pues esto podría afectar el performance del reproductor, para ello, dentro de esta segunda fase existe el ciclo *ON* y el ciclo *OFF* como medios periódicos de control de transmisión y descarga de bloques para mantener un tamaño de búfer adecuado a una óptima calidad.

En la Fig. 3 podemos observar los diferentes procesos involucrados en una transmisión streaming, iniciando con la fase de búfer que va creciendo en forma lineal creando una pendiente generada a partir del ancho de banda del enlace entre cliente y servidor hasta llegar al punto de reproducción según los parámetros de la sesión, mientras que la fase estacionaria varía entre llegada de paquetes en los ciclos *ON* y tiempos muertos en el ciclo *OFF* que nos indica que no es necesario solicitar un paquete en ese intervalo de tiempo, también se puede observar que estos ciclos son repetitivos durante todo el proceso de la fase estacionaria y la pendiente que genera su alternatividad produce la tasa promedio durante la *steady state*.

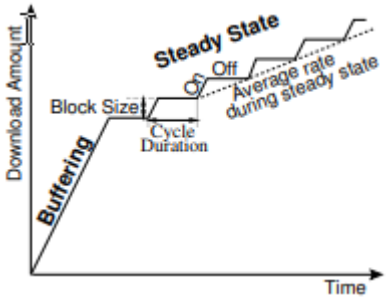

Fig.3. Fases de transmisión de segmentos streaming. [6]

Este funcionamiento trae consigo diversas características que se acentúan cada vez más en los nuevos protocolos de transmisión dedicados a streaming. Primero, la división del archivo multimedia en pequeños fragmentos de corta duración, generalmente entre 3 y 10 segundos, dependiendo del protocolo utilizado, luego cada uno de estos segmentos son transmitidos a través de internet mediante una comunicación *HTTP*, lo que facilita su envío por medio de puertos estándar como el *TCP* no cifrado (80) o el *TCP* cifrado (443).

Otro aspecto importante es que estas sesiones streaming se caracterizan por ser conexiones no persistentes entre el servidor de origen y el cliente final durante una transmisión, pues, con la existencia de los ciclos de encendido y apagado, la petición de un segmento es totalmente independiente de la otra, es decir, que genera comunicación solamente al momento de una solicitud o transmisión de un paquete (ciclo *ON*), lo que lo hace amigable y compatible a la memoria caché de los dispositivos y ayuda al mejor aprovechamiento del ancho de banda.

Por otro lado, una de las características del streaming que da importancia y es base fundamental de este trabajo es la capacidad de reproducción de contenido a velocidades de bits adaptativa o *Adaptative Bit Rate*.

Actualmente los proveedores de contenido (*CDN*) codifican sus vídeos a diferentes calidades para que, según el ancho de banda disponible del cliente, se determine el nivel de calidad en el que se le proporcionarán los segmentos del contenido, esto con el fin de mantener al máximo posible una experiencia de reproducción fluida, de detectarse cambios en los parámetros de performance de la red, ya sea a mejor o peor, se realizan de forma dinámica los ajustes correspondientes en los parámetros de la comunicación para minimizar el almacenamiento en búfer, proporcionar una reproducción de alta calidad y que sea lo menos perceptible para el usuario final. [7]

Grandes compañías como Netflix y YouTube trabajan bajo estos esquemas de transmisión combinando sus estrategias de streaming según las características de las conexión y las fases genéricas antes mencionadas, por ejemplo, existen transmisiones donde todos los segmentos se van almacenando en búfer sin necesidad de una fase estacionaria porque los parámetros de la comunicación lo permiten, así como a su vez existe la combinación de fases de forma estática o con variación en la longitud de ciclos de *ON-OFF*, generado a partir de mezclas de algunos períodos cortos de transmisión y silencio con otros períodos largos de los mismos.[6]

En cuanto a la seguridad en la retransmisión de estos segmentos de vídeo, se ha comprobado que los nodos intermediarios simplemente enrutan los fragmentos hacia su destino final y que en ciertos casos también almacenan en caché tal paquete.[7] Sin embargo, estos fragmento de datos no están exentos de ataques de terceros, por ejemplo de tipo *man in the middle*, que podrían ejecutar códigos especializados para modificar el contenido del paquete.

II.2. *DYNAMIC ADAPTIVE STREAMING OVER HTTP (DASH)*

El streaming de tasa de bits adaptable sobre HTTP o comúnmente conocido por sus siglas *DASH*, es un habilitador capaz de proporcionar contenido multimedia en diferentes formatos y resoluciones para permitir la entrega eficiente y de alta calidad en servicios de streaming por internet y que fue estandarizado en 2012 por el *Moving Picture Experts Group* (*MPEG*). Dicho estándar permite la transmisión de contenido bajo demanda, transmisiones en vivo y servicios de grabación automática, además de ser un esquema de carácter internacional y totalmente abierto.

Para su correcto funcionamiento, el CDN genera distintas copias del contenido en diferentes idiomas, calidades, resoluciones, tasas binarias, etcétera, donde cada una se divide en segmentos almacenados en el servidor HTTP junto con un fichero de índices (MPD) que contiene los metadatos de cada segmento, tal como se puede observar en la parte izquierda de la Fig. 4, mientras que en la parte derecha se tiene el cliente DASH compuesto por un bloque de control DASH, el reproductor multimedia, el interpretador de índices y un cliente HTTP para las comunicaciones con el servidor.

Lo que delimita los parámetros de la comunicación DASH son los bloques marcados en rojo de la Fig.4, en ellos se define el MPD con información de cómo están compuestos los segmentos y la codificación de estos, los cuales pueden ir variando según la dinámica de retroalimentación.

Dicha retroalimentación funciona de la siguiente forma: Primero, vía HTTP el cliente DASH solicita con un GET el archivo MPD, el servidor atiende su solicitud y lo envía. Posteriormente el cliente procesa y gestiona la información recibida en el bloque *MPD Parser*, a partir de dicha información se generan instrucciones al bloque *Segment Parser* donde, según la lógica de selección de calidades, se solicitan los segmentos correspondientes al servidor remoto a través del cliente HTTP, una vez recibido el segmento, este es visualizado en el reproductor y se repite el proceso.[8]

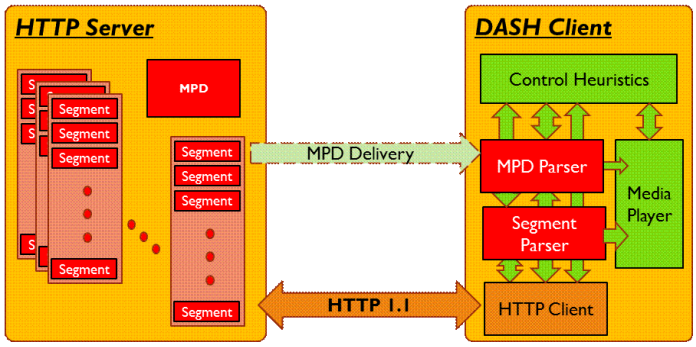

Fig.4. Diagrama de comunicación cliente servidor con DASH. [9]

Luego de comprender el mecanismo de comunicación entre cliente y servidor, se puede observar lo determinante que es la función de los *Media Presentation Description* o índices a lo largo de la transmisión streaming. Como uno de los objetivos de esta tesis de final de máster es trabajar con segmentos de diversas calidades, codificaciones y formatos, resulta útil profundizar en su contenido.

El MPD como tal, es un archivo XML que describe las características y metadatos del contenido multimedia (audio, vídeo, subtítulos, etc.), este es constituido por subelementos jerárquicos que hacen referencia a un concepto específico para ayudar al mejor manejo de segmentos, los cuales están relacionados entre sí como se muestra en la Fig.5. Estos elementos son:

- *Period*: Son etapas temporales en las que se puede dividir el contenido multimedia, por ejemplo, una serie se divide en diversos períodos llamados capítulos, así como una película se divide en escenas y álbumes en canciones.
- *Adaptation Set*: Son componentes del contenido dentro de cada período, por ejemplo, codificadores, pistas de audios, subtítulos, idiomas y contenido de vídeo.
- *Representation*: Es la calidad en la que el contenido del *adaptation set* es ofrecido, orientado al *bit rate* y la resolución.
- *Segment*: Son los fragmentos en los que se divide el contenido del elemento de representación y que poseen una duración determinada.

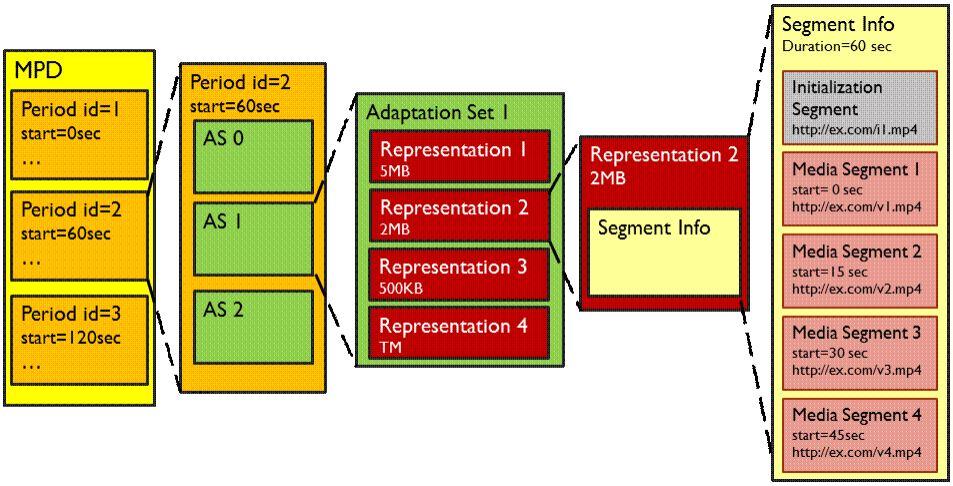

Fig.5. Modelo de datos jerárquico del formato *Media Presentation Description.* [9]

Como se observa en la Fig. 5, el MPD puede contener diversos períodos, en cada uno de ellos se incluye información de su identificativo, el perfil al que se adecua y el tipo de contenido al que apunta, siendo estático para el contenido bajo demanda o dinámico si es contenido en vivo, también incluye otros parámetros como la duración del búfer y opcionalmente valores de tiempo mínimo de actualización del MPD o el URL base. Los flujos multimedia que componen al período son uno o varios sets de adaptación que contienen el identificador, el momento en el que se inicia el período al que pertenecen y la duración de este. Los sets de adaptación generalmente se utilizan para la inserción de anuncios, emisiones programadas o control de idiomas.

Un aspecto interesante es que las características de tasa de fotogramas, números de canales de audio o tasas de bit no varían en estos flujos multimedia mientras no haya cambio de período, pero, lo que sí puede variar en esto es la representación, que figura como el siguiente nivel de la jerarquía MPD. Cada representación tiene consigo obligatoriamente un identificador propio con ancho de banda y tasa de fotograma definida junto con atributos comunes como la codificación y resolución espacial, estas representaciones son elegidas a partir del algoritmo heurístico de selección del cliente DASH para cada segmento a transmitir pues se encuentran al comienzo de cada fragmento.

Al final de la jerarquía tenemos los segmentos, estos son los elementos que se descargarán en el cliente DASH en los diferentes formatos existentes, se comienza por un segmento de inicialización y posteriormente los segmentos multimedia en contenedores ISO BMFF o MPEG2-TS de pequeñas duraciones y, aunque el estándar como tal no es restrictivo, se recomienda una duración entre dos y cuatro segundos. También, los segmentos como tal suelen estar definidos a través de URL numeradas de manera lógica siguiendo un patrón previamente definido donde cada representación tiene la URL de cada uno de los segmentos que lo componen. [10]

Como el objetivo de este trabajo es verificar la integridad de cada uno de los segmentos streaming, independientemente de su representación, este último elemento de la jerarquía MDP será el más determinante y el cual será manipulado en el prototipo del sistema.

II.3. *BLOCKCHAIN*

El blockchain es una tecnología basada en secuencia de bloques que generan una lista completa de todas las modificaciones hechas en una determinada cadena de objetos, como un registro público convencional, con el objetivo de mantener y verificar la integridad de las transacciones que envuelven un determinado proceso.

La particularidad de este concepto es que cada uno de estos bloques está ligado al bloque anterior a través de un hash cuya referencia generalmente está contenida en la cabecera del bloque actual, de esa forma, cada elemento tiene sólo un bloque padre referenciado, a excepción del primer elemento de la cadena al que se le denomina bloque génesis.

Una estructura general de blockchain se ejemplifica en la Fig.6, donde se observa que cada bloque tiene un lugar específico dentro de la cadena y está compuesto por datos propios y el hash del bloque anterior como método de relación. Cuando ya se han referenciado todos los bloques de interés, la cadena completa se guarda en cada nodo de la red conformando el blockchain y se almacena una copia exacta de ella para todos los participantes de la red, de esa forma, cualquiera de ellos puede verificar las transacciones contenidas y la integridad de estas si así se desea.

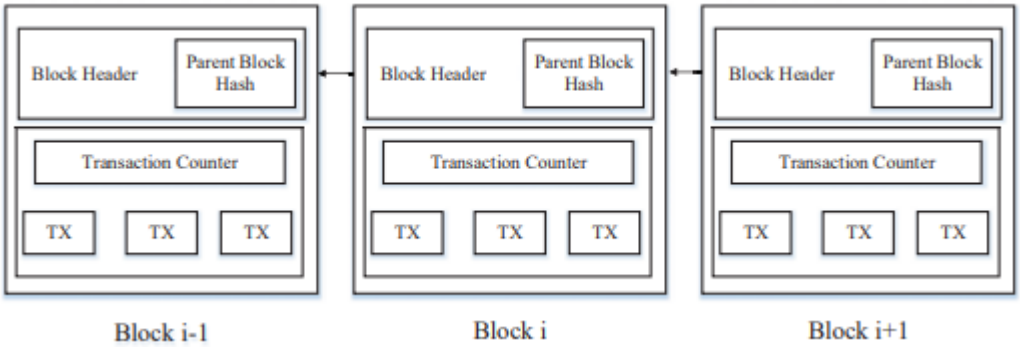

Fig.6. Representación gráfica un proceso blockchain.[11]

Para garantizar la seguridad de los datos, blockchain utiliza un sistema de claves encriptadas o firma digital en el cual cada usuario posee claves públicas y privadas. La clave privada o confidencial se utiliza para firmar (encriptar) las transacciones que se difunden en toda la red mientras los usuarios interesados verifican las transacciones mediante la clave pública, mitigando así la manipulación de los datos.

La estructura en la cual está desarrollado el blockchain trae consigo características importantes para la seguridad de los datos, por ejemplo, su utilización elimina la centralización de las transacciones mediante terceros sin afectar la integridad entre emisores y receptores, esto es posible gracias a algoritmos de consenso previamente definidos entre las partes interesadas para mantener la coherencia de la información, convirtiéndola así en una tecnología persistente, donde se vuelve prácticamente imposible no detectar una transacción ya incluida en un bloque y que haya sido

eliminada, anulada o modificada, permitiendo la validación rápida y la identificación de bloques que contienen transacciones inválidas donde se ha roto la cadena, mejorando de esa forma todos los procesos de auditabilidad, debido a que almacena datos del registro de usuario basados en un modelo de Salida de Transacciones No Gastadas (UTXO) donde toda transacción tiene que referirse a algunas transacciones anteriores no gastadas. Una vez que la transacción es grabada en la cadena, el estado cambia de no gastada a gastada, de esa forma es más fácil hacer la verificación y seguimiento de las transacciones.[11]

Aunque el gran auge de la tecnología blockchain vino acompañada de sus aplicaciones en el sector financiero, por estar ligada al uso criptomonedas, la aplicación del concepto blockchain está ganando mercado en otros dominios de aplicación como *Internet of Things (IoT), security services*, aplicaciones de contenido multimedia donde últimamente se ha trabajado en el desarrollo de servicios como *Video-on-Demand*, *Live Streaming Solutions* y otras soluciones de vídeo basados en esta tecnología como la que se plantea en el capítulo IV de este trabajo.

II.4. *ALGORITMOS DE CIFRADO PARA BLOCKCHAIN*

Un algoritmo de cifrado es un conjunto de instrucciones que permiten la encriptación de datos para prevenir la lectura, alteración o interferencia del mensaje original por parte de terceros a los cuales la información no va dirigida. Estos algoritmos son inevitablemente deterministas pues una entrada dada siempre tendrá la misma salida y cada salida es esencialmente única al momento de descifrarse. Otra cualidad de estos algoritmos es que las posibilidades de que dos entradas separadas tengan el mismo valor de cifrado son prácticamente nulas puesto que, un ligero cambio en la entrada da como resultado una salida totalmente diferente y no relacionada con otra similar. A pesar de que en la tecnología blockchain se utilizan diferentes técnicas de cifrado [12], en este documento se profundiza únicamente el cifrado con funciones HASH debido a su relación con el presente trabajo final de máster.

Los algoritmos de cifrado con funciones HASH se basa en modelos matemáticos que son capaces de transformar los datos en códigos llamados hash. Un hash es una serie de caracteres que, independientemente de la amplitud de la entrada, el tamaño de la salida es de una longitud fija. Una de las ventajas significativas de este tipo de algoritmos es que, como tal, el hash codificado es matemáticamente imposible de descifrar o realizar ingeniería inversa en él, lo que demuestra lo robusto que son los encriptados basados en esta técnica, otra característica importante de mencionar es que sin importar las veces que se coloquen los mismos datos a través del algoritmo, este producirá consistentemente un hash con caracteres idénticos en la cadena así como cualquier cambio mínimo producirá un hash totalmente diferente, por último, la creación de cada hash debe ser un proceso rápido de poco coste computacional para garantizar la fluidez y eficiencia de las operaciones. [13]

En esta categoría de cifrados existen diversos algoritmos que generan hashes de manera óptima y con garantías de seguridad aceptables, sin embargo, los más destacados son el cifrado *Message-Digest Algorithm 5* (MD5) y las diversas variantes pertenecientes a la familia de Algoritmo de Hash Seguro (SHA).

El algoritmo MD5 es un método de cifrado que utiliza el relleno de mensaje para su encriptado de forma que la información final de envío se procesa en segmentos múltiplos de 512 bits cuya salida es la unión de cuatro bloques en la que el algoritmo va dividiendo, tratando y encriptando el mensaje para representarlo como una reducción criptográfica de 128 bits. Los algoritmos SHA por su parte en lugar de dividir la información en cuatro bloques de 128 bits la dividen en 5 bloques de 32 bits cada uno generando hashes de extensión mayor y por consiguiente más seguros que su predecesor. [14]

MD5 y SHA al ser algoritmos de métodos basado en hash poseen un funcionamiento similar y comparten características comunes, por ejemplo, ambos utilizan el relleno de mensaje antes de su cifrado, separan la información en bloques de bits para su tratamiento, además del uso de firma digital para mejorar la verificación de integridad del mensaje.

En cuanto a su comparativa se observa que cada uno de estos mecanismos de cifrado logra su robustez empleando características individuales muy marcadas que distinguen el uno del otro. Como se observa en la Tabla 1, SHA utiliza bloques de longitud mayor para el cifrado de sus mensajes, eso, combinado con el número de iteraciones, hace que MD5 sea más rápido en su ejecución sin embargo, esto repercute en la seguridad de cada método, volviendo a MD5 más vulnerable en cuanto a ataques de fuerza bruta que en algunas ocasiones ya han sido exitosos, mientras que para el descifrado de un mensaje SHA el número de intentos es exponencial y significativamente mayor, volviéndolo casi imposible para las tecnologías actuales.

Comparativa	**MD5**	**SHA**
Longitud del Mensaje Cifrado	128 bits	160 bits
Velocidad de Cifrado	Rápido	Lento
Número de Iteraciones	64	80
Ataques Requeridos para Descifrado	2^128	2^160
Ataques Exitosos Reportados	Sí	No
Seguridad	-	+

Tabla 1: Comparativa entre algoritmo MD5 y SHA.[15]

A pesar de que MD5 tiende a ser más rápido que SHA, esta diferencia suele ser apenas de unos milisegundos, por lo que se ha decidido que la parte práctica de este trabajo final de máster será ejecutada con el cifrado SHA256, por garantizar una máxima seguridad, fluidez y baja carga computacional.

III. ESTADO DEL ARTE

El blockchain, desde su origen a mediados de la década del 2000, fue diseñado especialmente como un sistema electrónico para que dos personas o entidades pudieran tener un método seguro de flujo de dinero de un punto a otro sin necesidad de un intermediario, sin embargo, por el auge obtenido a través de su uso y fiabilidad en operaciones con criptomonedas, la aplicación de su concepto ha trascendido en dominios de aplicaciones diferentes al netamente financiero donde se han hecho varias adaptaciones que van desde soluciones sencillas a muy complejas.[16]

Por ejemplo, una aplicación básica y útil de verificación en contenido multimedia es el análisis de la integridad de una imagen cualquiera, para ello, fue publicado un método innovador basado en blockchain para conseguirlo en imágenes de formatos JPEG. De manera general, el procedimiento transforma una imagen de componentes rojo, verde y azul (RGB) a un espacio de color de luminancias y crominancias (YCbCr). Considerando que la luminosidad o brillo es el estímulo más primario y simple para el ojo humano, el procedimiento trabaja una concatenación con los valores de luminancia, esta componente es dividida en bloques de 8x8 píxeles respetando la codificación del formato JPEG y a cada uno de esos bloques se le aplica la Transformada Discreta de Coseno, el valor resultante es dividido por la matriz de cuantización definida y el redondeo de esa división produce un valor exacto para cada uno de los bloques de la cadena. Posteriormente, ese resultado es almacenado en un vector que finalmente es encriptado y enviado junto a la imagen para que el receptor pueda someter ambos elementos a una correlación para verificar su similitud. [17]

Otro método efectivo para la detección de imágenes manipuladas es el Sistema de Revisión de Hechos Blockchain publicado por el Sistema de Información Multimedia de la facultad de Ciencias de la Computación de la Universidad de Viena, el cual está capacitado para la detección de alteraciones y registro de copyright de una imagen.

Primero, en el algoritmo de registro, una imagen es cargada y dividida en sus diferentes componentes RGB a las cuales se les aplica un hash que es guardado como características propias al igual que los metadatos del archivo, estos son referenciados a un propietario con identificador único y el conjunto de esta información se agrega como un bloque de la cadena. Si la imagen cargada al sistema es codificada y los parámetros característicos resultantes coinciden con algún elemento creado previamente en la cadena, se concluye la existencia de un propietario, negando la posibilidad de generar un nuevo registro. De no encontrar un elemento existente, el sistema no descarta la manipulación de la imagen y lo somete a una verificación uno a uno con cada elemento ya registrado en el blockchain a través del método de vecinos más cercanos evaluando matrices de 8x8 píxeles que, mediante un ratio de similitud concluye la autoridad intelectual o manipulación de la imagen.[18]

Si pasamos de la imagen al vídeo, encontramos una evolución en los métodos de implementación de la verificación del contenido multimedia mediante blockchain puesto que la

comparación por matrices de píxeles resulta totalmente ineficiente y computacionalmente costosa de aplicar para cada imagen que compone al vídeo, por ello, su comprobación se realiza fotograma a fotograma o a través de segmentos de duración específica.

Para el primer caso, se busca la forma de dividir el vídeo en sus respectivos frames para que cada uno pueda ser codificado con un hash propio que es referenciado al de su sucesor, a excepción del cuadro inicial, de esa forma, al existir una modificación en cualquiera de ellos, habrá una alteración en las dependencias que provocará una inconsistencia, la cual será detectada fácilmente al hacerse la verificación final de la integridad del contenido.

Acoplado a esas premisas de funcionamiento existe una solución propuesta por investigadores ucranianos de la *Ternopil National University*, dicha solución está montada en el entorno multiplataforma *Node JS*, en ella, la división de cada cuadro del vídeo se realiza a través de la librería FFMPEG, una herramienta caracterizada por la maleabilidad impregnada a los elementos que componen una grabación pues, independientemente del formato del contenido multimedia, genera una gran cantidad de información útil que ayuda a una integración fluida con los distintos módulos de cualquier sistema de verificación. Luego de la división por cuadros, el módulo *crypto* genera el hash correspondiente al bloque actual, mientras simultáneamente se envía una petición de frames a la librería, una vez recibido el cuadro actual, el módulo *fs ta pach* realiza una interacción física con este para determinar que su posición en el vídeo es la adecuada, garantizando así la secuencia lógica marcada por el contenido original.

Una vez obtenidos los elementos anteriores, por medio de una comunicación HTTP, estos son enviados a la unidad de *Naivechain* que es la interfaz controladora y responsable de la construcción del bloque actual. Dicha interfaz genera una comunicación P2P bastante simple utilizando web sockets para comunicarse con el nodo anterior con el fin de obtener el valor de su hash, cuya incógnita completa las piezas necesarias para generar un nuevo nodo. Una vez homogeneizados los elementos, se genera el bloque actual que posteriormente es adicionado a la cadena existente. Dicho flujo de procesos es representado gráficamente en la Fig. 7 mostrada a continuación.

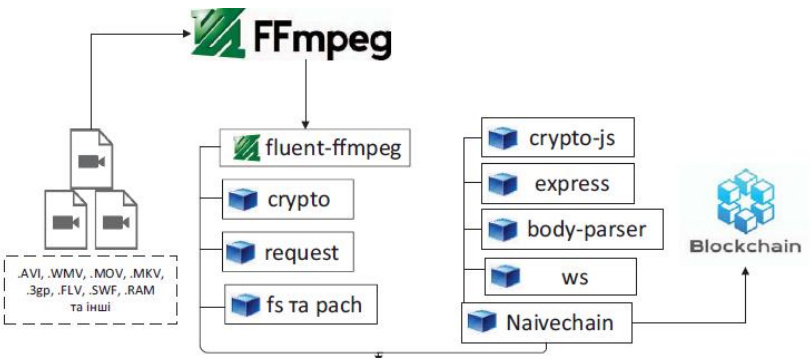

Fig.7. Diagrama de verificación de vídeo con blockchain basado Node JS. [19]

El resultado de este flujo de procesos se puede observar en la Fig. 8, estructuralmente cada uno de estos bloques están compuestos por metadatos, el valor del hash del cuadro actual, el valor del hash del cuadro anterior que, combinados con el contenido pictográfico y de audio, producen elementos de arquitectura totalmente diferente a las producidas en una codificación de imágenes simples, pero, a pesar de que esta solución es muy competente para el tratamiento de vídeos y se adapta a cabalidad al funcionamiento teórico del blockchain, trae consigo problemas de performance, especialmente ligados a contenido de alta definición pues en tales condiciones su tasa de codificación varía entre uno y tres frames por segundo, lo que impacta notablemente su tiempo de ejecución en elementos de larga duración.[19]

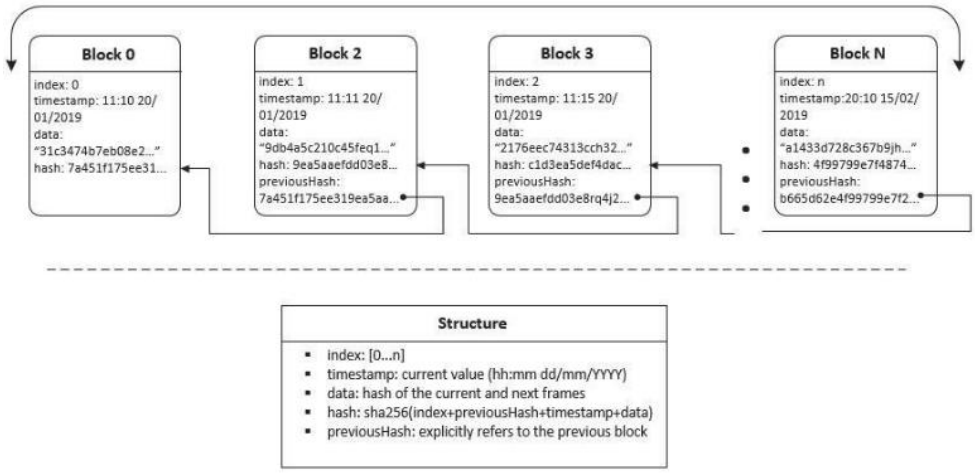

Fig.8. Estructura de vídeo blockchain basada en frames. [19]

Ante la problemática de performance producto de realizar un bloque por cada cuadro que compone a un vídeo, se buscaron alternativas viables que redujeran los tiempos de procesamiento al momento de crear los nodos que conformarán el blockchain, una de las opciones destacadas es la división del contenido en grupos de imágenes.

Se conoce que las cámaras emplean diversas técnicas o estándares de compresión al momento de capturar un vídeo para reducir el espacio físico de almacenamiento, sin tener afectar la calidad del contenido, una de estas técnicas es la codificación por diagrama de bloques. En este prototipo de codificaciones existen al menos tres tipos de imágenes:

- *Capa Intra (I)*: Imagen fija e independiente que representa una referencia para los otros dos tipos de cuadros y su ubicación determina el inicio de un grupo de imágenes.
- *Capa de Predicción (P)*: Conocida como capa de movimiento, representa los cambios producto de movimientos entre frames, tal diferencia puede ser entre una capa intra y una de predicción o dos capas de predicción.
- *Capa Bidireccional (B)*: Son cuadros que contienen información de la imagen precedente y la siguiente, la cual puede ser de tipo I o P.

Esta división por GOP fue aprovechada para crear un sistema de comprobación de integridad del vídeo con blockchain donde en lugar de crear nodos por cada cuadro se crean por cada grupo, aprovechando la ventaja generada por la codificación, lo que disminuye considerablemente el número de nodos totales y el tiempo de ejecución. Una variante adicionada a esta solución es la implementación de un acuerdo mutuo de carácter digital entre dos partes interesadas en el intercambio verificable de contenido multimedia. Este convenio llamado *Smart Contract* es el encargado de realizar la división del vídeo y la generación del blockchain en un servidor previamente consensuado entre emisor y receptor que sirve como punto intermedio de fiabilidad de ambas partes, tal como muestra el diagrama en la Fig. 9. [20]

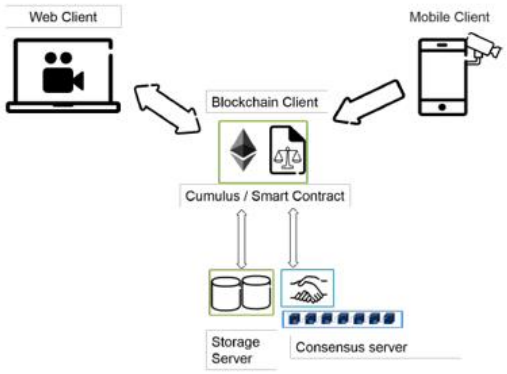

Fig.9. Esquema de vídeo blockchain con uso de Contratos Inteligentes. [20]

Con relación a los resultados, este esquema es capaz de realizar múltiples codificaciones de GOP por segundo en servidores de capacidades similares a las de un ordenador portátil actual, lo que nos muestra la mejora sustancial del performance al momento de crear la cadena de bloques, sin embargo, el uso de estas soluciones requieren la contratación de servicios terciarios, trayendo consigo costos que pueden evitarse aplicando algoritmos de encriptado local, si la sensibilidad del contenido lo permite y aunque existen servidores de consenso de carácter gratuito, no garantizan la calidad y fiabilidad necesaria para este tipo de aplicaciones.

Otro punto negativo de esta solución es que su aplicabilidad se limita a vídeos ya finalizados, su descomposición en grupos no es ajustable mientras el contenido está siendo grabado pues esto depende de la existencia de distintas capas intra que se vayan generando y tales condiciones de aparición no se alcanzan a modelar para cualquier escenario o tipo de codificación.

Ante la falta de una solución que optimizara la comprobación de archivos de vídeo combinado a la idea de poder firmar digitalmente contenido que se genera en vivo, se desemboca la necesidad de crear aplicaciones basadas en codificación de audio y vídeo con segmentos de duración determinada.

El sistema de integridad de vídeo descentralizado para la captura de accidentes automovilísticos es un ejemplo de este tipo de aplicaciones, su método de funcionamiento es mostrado en la Fig. 10,

primero el usuario activa la captura de vídeo en su teléfono móvil, desde ese momento y cada cien milisegundos los valores de aceleración son consultados en los ejes tridimensionales en el acelerómetro incorporado del teléfono inteligente, cuando existen cambios bruscos en ellos y la posición del GPS se encuentra estática, se genera una detección de colisión que inmediatamente ejecuta la orden de firmar digitalmente la captura, con el objetivo de generar la cadena de bloques correspondiente, por último, la sucesión de hash resultante se envía a un servidor blockchain para su posterior comparación. Todo esto resulta muy útil para la documentación de accidentes además de ser un apoyo para generar un veredicto justo según sean las condiciones.

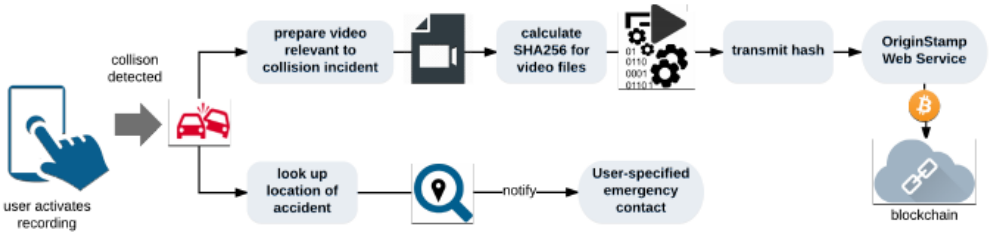

Fig.10. Funcionamiento sistema de integridad de vídeo para la captura de accidentes automovilísticos.[21]

En cuanto al tratamiento del vídeo, la aplicación está grabando continuamente mientras esta se encuentre en ejecución, sin embargo, la mayoría de las ocasiones puede resultar innecesario almacenar recorridos sin accidente alguno, por eso la aplicación se encarga de documentar y firmar solamente segmentos específicos, para ese control, la captura del contenido es dividida en fragmentos de diez segundos de duración.

El primer segmento es grabado y almacenado en la memoria del móvil mientras el segundo fragmento está siendo grabado, si al finalizar la grabación del segundo ninguna colisión es detectada, el primero trozo de vídeo es eliminado mientras el tercer segmento es grabado, tal proceso es repetido de forma que al no existir incidencia, la documentación no es de un tamaño mayor a veinte segundos pero, si en caso contrario, un accidente es detectado, los trozos de grabación no son descartados y se comienza el proceso de firma digital mediante hash para cada fragmento, documentando únicamente lo concerniente al accidente, lo que demuestra la importancia del manejo de vídeo verificable a través de pedazos de poca duración. [21]

A lo largo de esta sección hemos percibido que la utilización del concepto de la tecnología blockchain aplicado al contenido multimedia ha sufrido una constante evolución de carácter positivo, pues cada vez es más común encontrar soluciones que nos brinden la seguridad necesaria para comprobar la integridad de lo que vemos.

Este constante progreso ha permitido adaptar un mecanismo, ideado inicialmente como un método financiero, a la comprobación de la integridad de una imagen y su autor, la verificación de un vídeo existente u otro que está siendo grabado ahora mismo, sin embargo, con el surgimiento de nuevas formas de proveer y transmitir contenido multimedia en streaming, como lo hacen

actualmente Netflix, YouTube, Amazon, Facebook y otras compañías, ha dejado un espacio aún no cubierto y que puede significar el siguiente eslabón en la cadena de evolución del blockchain en el área multimedia, pues todas las soluciones mencionadas anteriormente están diseñadas para contenido que no varía de calidad a lo largo del tiempo, un fenómeno que sí se encuentra presente al ver una película o serie por internet, en las comunicaciones por videollamada o transmisiones en vivo donde la calidad puede ser variante según los parámetros de conexión tal como se mencionó en la sección II.

La posibilidad de poder verificar la integridad de un contenido multimedia de calidad variante y no variante en el tiempo es lo que da vida a la idea de crear un programa único e innovador capaz de conjuntar ambos escenarios y que podría dar respuesta a la pregunta: *¿Qué sigue para el blockchain en el dominio de aplicaciones multimedia?*

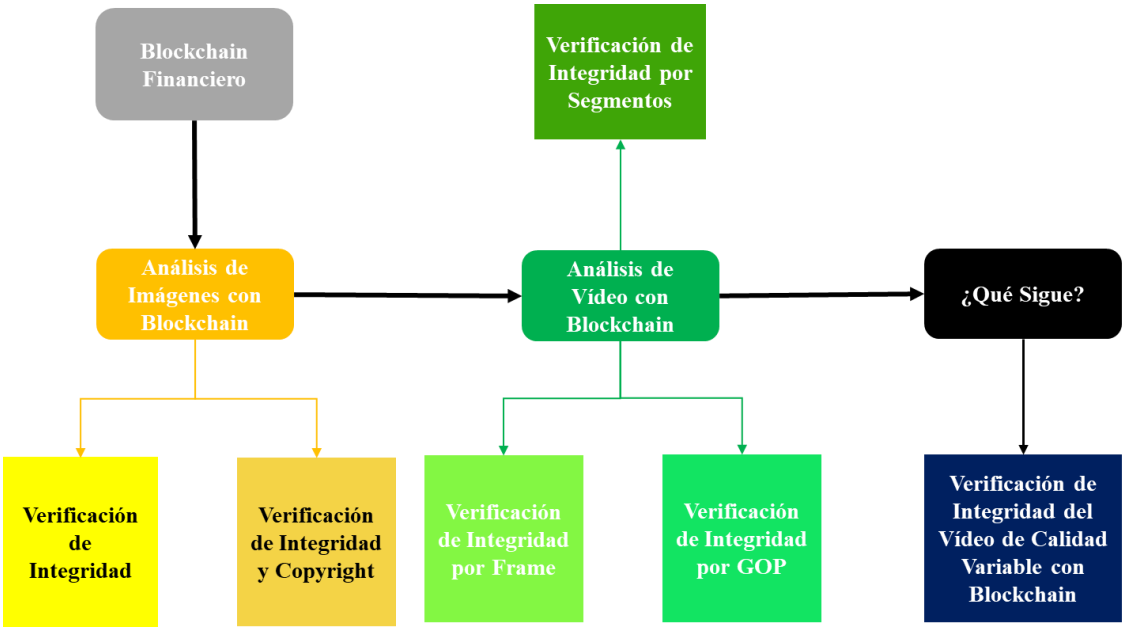

Fig.11. Evolución del blockchain aplicado a contenido multimedia.

IV. ESTRUCTURA DEL SISTEMA

IV.1. *PREMISA Y DISEÑO DEL SISTEMA*

Para el diseño de un sistema de verificación de la integridad de un vídeo de calidad variable es necesario considerar que un sistema de comunicación humano o informático básicamente está compuesto por un emisor, un mensaje, un medio de transmisión y un receptor, bajo esas condiciones el objetivo principal de la solución es comparar de forma fiable, eficaz y eficiente los segmentos de vídeo enviados por el emisor con el vídeo resultante recibido por el usuario final.

Las propiedades deterministas que una cadena de bloques referenciados impregna a una serie de transacciones o elementos relacionados entre sí, a través de la concatenación de la información

contenida en cada bloque vía un algoritmo de encriptación, nos brinda un método ágil para una comprobación directa o detallada entre dos cadenas, ya sea verificando el hash del último bloque o comprobando la información o el hash de cada bloque individualmente.

En ese sentido, la estructura propuesta para los bloques está compuesta por tres atributos: primero un valor de Índex que hace referencia a la posición del segmento de audio o vídeo en la cadena, segundo, un hash correspondiente al segmento identificado que nos asegura la integridad de éste, pues, de existir una alteración en su contenido, el hash resultante es totalmente distinto y tercero, el valor del hash del bloque anterior, a excepción del bloque génesis donde dicho valor se inicializa en "0" al ser el origen de la cadena, la representación gráfica del blockchain resultante se muestra a continuación en la Fig. 12.

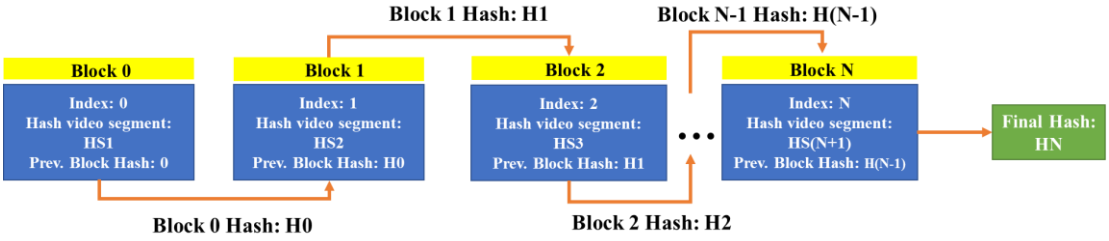

Fig.12. Estructura del blockchain para vídeos de calidad variables.

La elección de estos únicos tres valores obedece, en primer lugar, a un principio de eficacia del sistema, es claro que mientras menos atributos compongan el bloque, más rápido será su ensamblaje, procesado y comparación, también es cierto que otros atributos comúnmente utilizados como la versión del bloque, número de bits por bloque, el contador de transacciones y el *nonce*, resultan innecesarios incluirlos porque la secuencia lógica del vídeo determina el número de bloques, su tamaño y encadenamiento, a su vez, los segmentos son las entradas que producen un hash único al pasarlo por el algoritmo de cifrado, lo que hace prescindible la utilización de un *nonce*. Sin embargo, el único atributo deliberadamente omitido es la marca de tiempo o *timestamp* porque por muy eficaz que sea la conexión entre emisor y receptor, habrá una diferencia de tiempo entre el envío y recepción lo que produciría como resultado una disparidad en cada uno de los bloques aunque estos no sean alterados, al obviar dicho parámetro se garantiza una cadena de bloques determinista tanto en emisión como en recepción y cuya comparación genera una conclusión positiva o negativa de la veracidad entre mensaje original y mensaje recibido.

Una vez definida la composición de la cadena de bloques, la estructura de la solución fue diseñada partir de tres módulos principales, cuyos funcionamientos son profundizados en las siguientes subsecciones. Tal y como lo muestra la Fig. 13, los nombres correspondientes a cada módulo son: módulo blockchain en el emisor, módulo blockchain en el receptor y módulo de comparación.

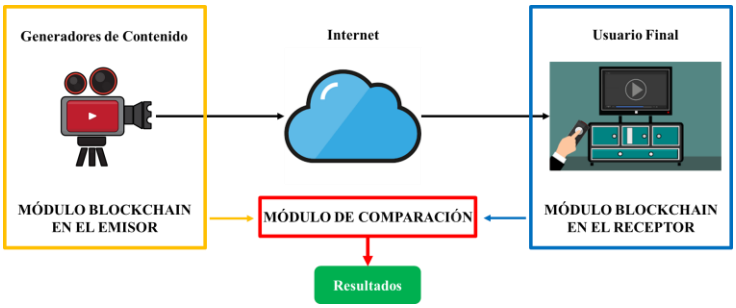

Fig.13. Diagrama de módulos para el sistema de verificación de vídeo streaming.

IV.2. *MÓDULO BLOCKCHAIN EN EL EMISOR*

El módulo de blockchain en el emisor tiene como objetivo principal la creación de la cadena de bloques a partir de los segmentos identificados en el log generado por el navegador que se utilice para la reproducción del contenido.

Al ejecutarse el módulo en el emisor, el archivo de extensión HAR pasa inicialmente por dos métodos de filtrado, el primer método se encarga de encontrar el segmento MP4 de referencia para la reconstrucción del audio y vídeo mientras que el segundo método identifica los segmentos M4S seleccionados según las condiciones de la sesión streaming, sin embargo, al resultado de este segundo filtro se le aplica una depuración de segmentos repetidos, pues el log contiene cada elemento enviado por el transmisor y cada elemento recibido por el receptor, lo que genera un doble registro del mismo segmento. Una vez realizada esta tercera etapa de depuración, se procede a crear de manera separada una lista con los segmentos MP4 y M4S de vídeo y otra idéntica para los segmentos de audio, esta división se realiza para una correcta construcción del blockchain de forma individual, a su vez, el módulo cuenta con la función de salvar en un archivo de texto cada una de las listas si así se desea.

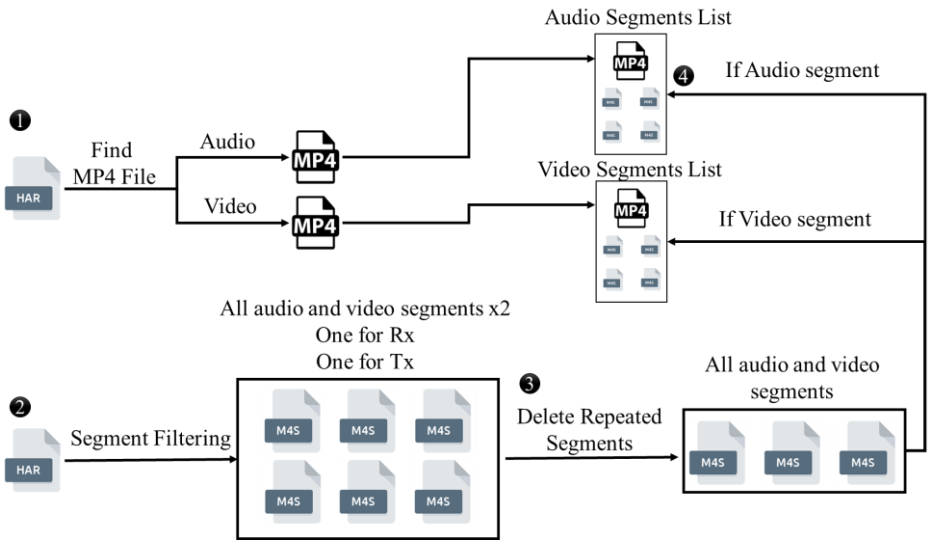

Fig.14. Filtrado de archivo HAR y construcción de lista de segmentos.

Habiendo separado las listas según el tipo de segmento, cada una es enviada al método de creación del blockchain donde elemento a elemento se van creando los bloques. Primero se agrega el valor índex correspondiente a la posición del segmento, en segundo lugar, se agrega una cadena de texto que es el resultado de cifrar con SHA-256 el contenido del segmento M4S actual, como tercer y último componente, se adiciona el valor de referencia del bloque previo, para su consecución y como al objeto bloque como tal no es posible cifrarlo directamente, fue necesario definir una estructura JSON donde se detalla el nombre del parámetro y su valor en el bloque, el resultado de cifrar dicha información es añadido al bloque en creación. Este último paso es aplicable a todos los bloques excepto al bloque génesis donde el parámetro se inicializó con el valor 0. El orden lógico de aplicación es mostrado en la Fig. 15.

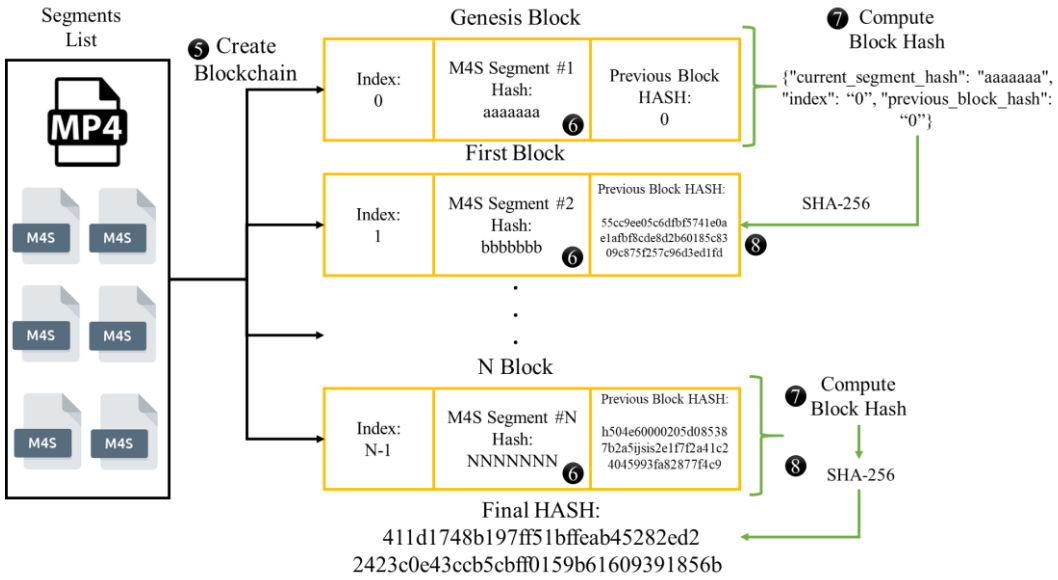

Fig.15. Creación de bloques y ensamblaje de la cadena referenciada.

Ejecutado todo el proceso antes mencionado, se cuenta con la funcionalidad de salvar la información de la cadena, si se realiza, en un archivo de texto se guarda la información de cada bloque creado y en la última línea se adiciona el valor del hash correspondiente al último bloque que servirá para una comparación directa entre cadenas.

Como la finalidad de estudio en este trabajo final de máster no radica en el modelado de un canal de transmisión streaming y con objeto de reutilizar el código, se define un método de reconstrucción del audio y vídeo con el cual se simula el contenido final recibido por el usuario, esto se fundamenta en los segmentos ya reconocidos en el archivo HAR, pues como se mencionó anteriormente, en él se encuentra el registro de transmisión y recepción de la sesión. Con ayuda de la herramienta FFMPEG, el módulo aplica el comando necesario para la concatenación según las condiciones de sistema operativo y tipo de segmento.

IV.3. *MÓDULO BLOCKCHAIN EN EL RECEPTOR*

El segundo módulo del sistema es el encargado del análisis y construcción de la cadena de bloques referenciados a partir del audio y vídeo recibido por el usuario final, los cuales fueron simulados en el último proceso del módulo anterior.

Al ejecutarse el módulo del receptor, se solicita la ruta donde está alojado el archivo de audio o vídeo concatenado, posteriormente se debe ingresar el nombre del fichero incluyendo su extensión MP4 y, por último, se requiere el número de segundos en los que se dividió cada segmento de la sesión streaming. Habiendo definido todos los parámetros antes mencionados y con ayuda de la herramienta MP4Box, el sistema realiza una separación del archivo ingresado con el fin de recrear los segmentos originales que lo conformaron. Una vez generada la división, los segmentos resultantes son la base para crear la cadena blockchain del receptor de la misma forma que se explicó y ejemplificó en la Fig. 15 de la sección anterior.

Por último, el módulo del receptor también cuenta con la finalidad de guardar la información de los bloques y hash final de la cadena en un archivo de texto cuya generación es fundamental para la comparación entre emisión y recepción.

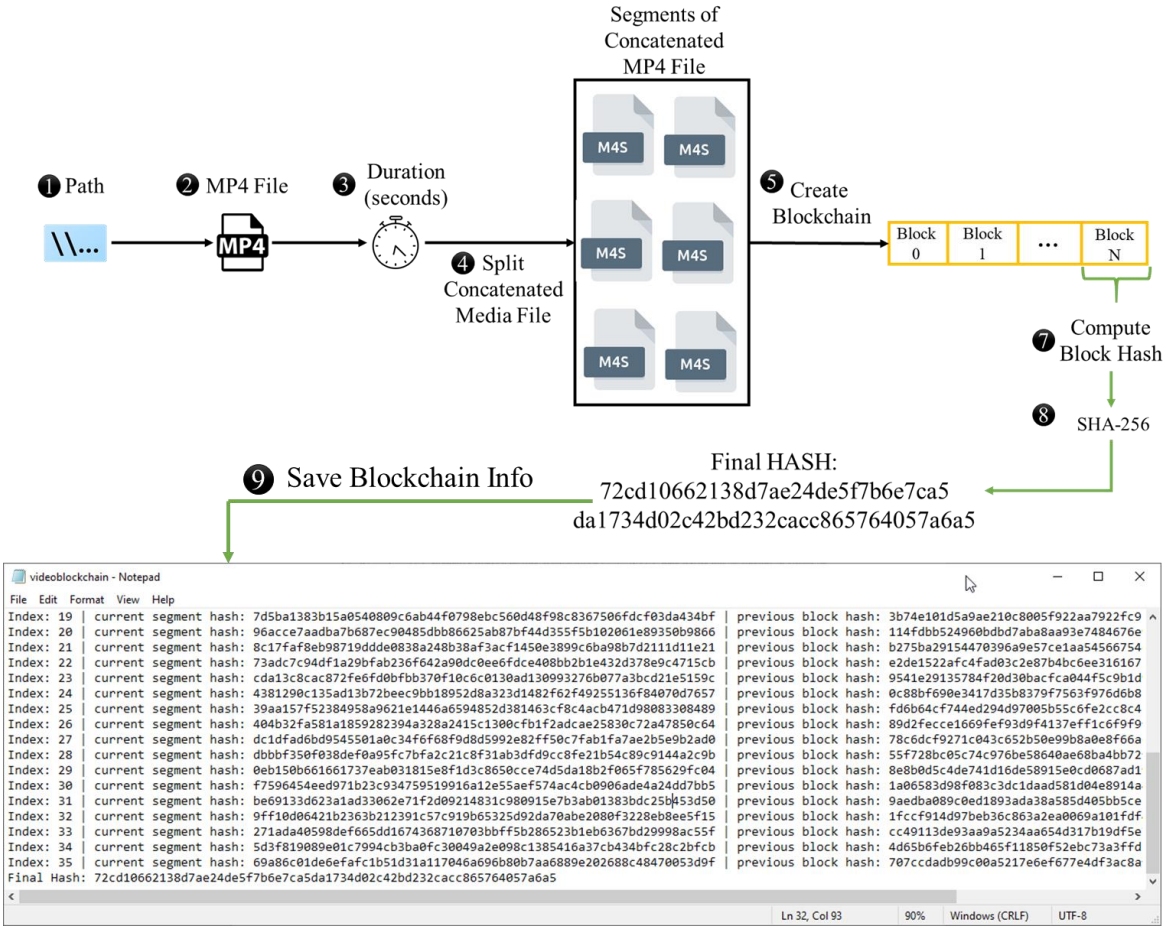

Fig.16. Funcionamiento del módulo blockchain en el receptor.

IV.4. *MÓDULO DE COMPARACIÓN BLOCKCHAIN*

El tercer y último módulo es el encargado de comparar la cadena identificada en el emisor con la cadena generada en el receptor a partir de los archivos de audio y vídeo concatenados.

Asentado en una estructura bastante simple, en primera fase, el módulo solicita el nombre del archivo blockchain de emisión y luego el de recepción, si estos no se encuentran en el directorio donde se está ejecutando el módulo, se debe ingresar el directorio donde están alocado y el nombre de estos, luego, verificada la existencia de los ficheros, se abren y leen cada uno individualmente, donde a su vez, las líneas contenidas en ellos son añadidas como elementos de una lista.

Completadas las listas, la segunda fase del módulo entra en acción enviándolas a un método de comparación donde, en primer lugar, se analiza el último elemento de cada una, pues como se fundamentó anteriormente, con el concepto de blockchain bastará la comparación del hash final para determinar la integridad de todos los bloques, si ambos elementos finales son iguales, se concluye una correcta integridad del vídeo streaming, de lo contrario, se deduce una alteración entre el envío y la recepción.

Para finalizar, independientemente del resultado de la comparación del último elemento, que corresponde al hash final de la cadena, el módulo permite un análisis detallado bloque por bloque donde se despliega toda la información contenida en cada uno de ellos, lanzando un estado “OK” para aquellos bloques iguales en el emisor y receptor, y un estado “ERROR” para aquellos con algún tipo de desigualdad, tal y como se muestra en la Fig. 17 mostrada a continuación.

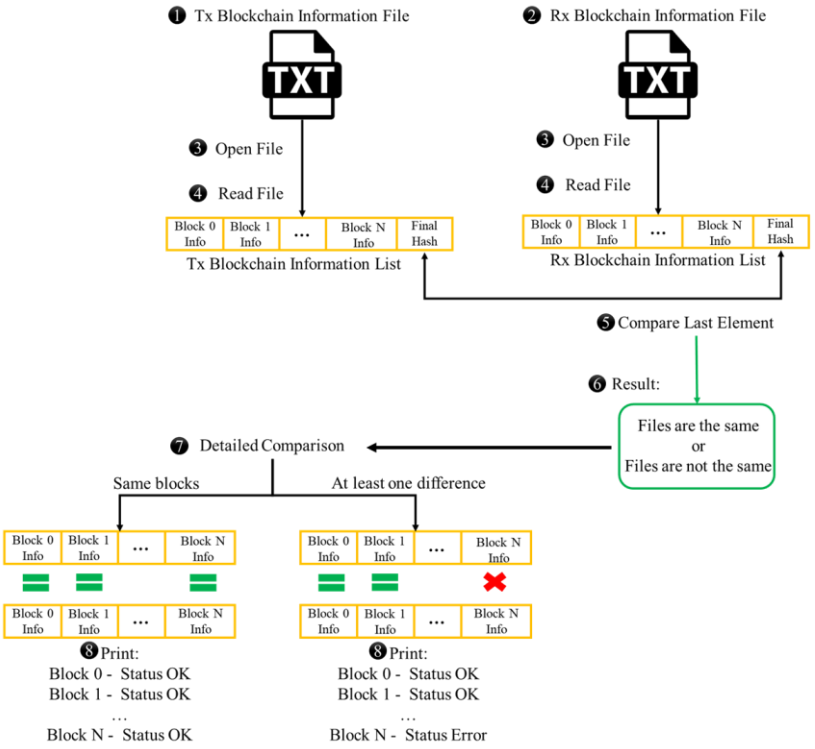

Fig.17. Funcionamiento del módulo de comparación blockchain.

V. PRUEBAS

V.1. *ENTORNO DE EJECUCIÓN DE PRUEBAS*

El entorno de ejecución de pruebas comienza con la codificación y preparación del contenido de audio y vídeo, así como la configuración necesaria para su transmisión utilizando la tecnología de streaming adaptativo.

Inicialmente se destina una carpeta en el cliente local donde se despliegan los archivos necesarios para el funcionamiento del reproductor multimedia, posteriormente, con el uso de Docker Desktop se prepara un entorno virtual entre la máquina local y el servidor web que servirá como origen de la sesión streaming.

Fig.18. Tipología y versiones utilizadas para el montaje de la sesión streaming.

Establecido el entorno virtual se procede a la codificación del vídeo original con la herramienta FFMPEG, con ella se separa el audio del vídeo con el objetivo de codificar solamente la componente de vídeo en diferentes calidades mientras la componente de audio será la misma, pues variar su calidad no representa un consumo considerable de ancho de banda. Teniendo las diferentes calidades en las que se ofrecerá el vídeo, se procede al empaquetado con la herramienta MP4Box, esto se realiza con el fin de segmentar las diferentes calidades del vídeo y la componente de audio que serán referenciadas en un único archivo MPD, el cual será consultado en la sesión streaming.

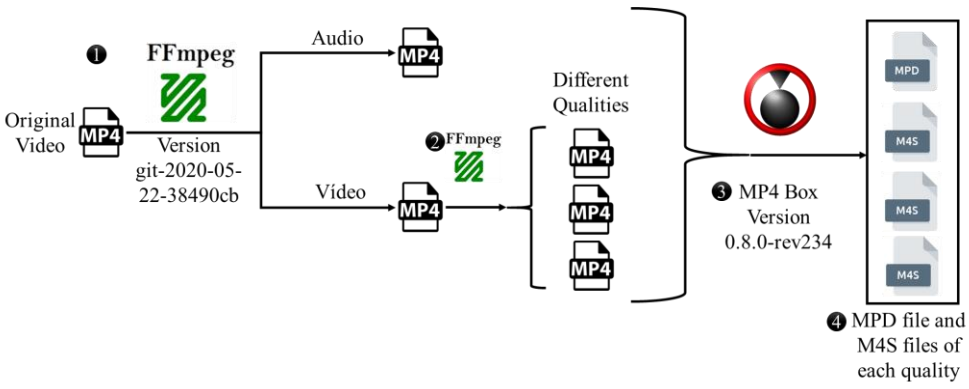

Fig.19. Versiones de las herramientas de codificación y empaquetado del vídeo.

V.2. *EJECUCIÓN DE PRUEBAS*

Explicado el proceso del montaje lógico e infraestructural, se definieron diversos escenarios para la obtención del archivo HAR, que sirve como base para la ejecución de los módulos del sistema creado. Para ello se ejecutaron pruebas en los navegadores Google Chrome, Microsoft Edge y Mozilla Firefox, en cada una de las pruebas se utiliza el reproductor Shaka Player versión 2.4.7 puesto que, en comparación a las versiones Shaka superiores, el log resultante de la versión antes mencionada se acopla de mejor forma al funcionamiento de las herramientas FFMPEG y MP4 Box.

Para las pruebas se utilizaron dos vídeos de diferente duración, cada uno fue empaquetado y codificado en tres resoluciones: 360, 720 y 1080 píxeles por pulgadas, de esa forma cualquier variación de ancho de banda en las herramientas de desarrollador de los navegadores permitirá al reproductor elegir la opción de segmento que mejor se acople al ancho de banda que experimenta al momento de solicitar un paquete, garantizando el escenario de vídeos con calidad variable deseado. Los recursos utilizados son los siguiente:

- *Vídeo tos.mp4* [22]: Fragmento del cortometraje *Tears of Steal* de *Blender Foundation* a resolución original de 4K y de duración 00:03:00.
- *Vídeo sintel.mp4* [23]: Tráiler del cortometraje *Sintel* de *Blender Foundation* a resolución original de 1080 p.p. y de duración 00:00:52.

También, para una mejor puesta a prueba de los diversos módulos, se realizó un conjunto de ensayos heterogéneos que combinan los vídeos y navegadores con segmentos de 3, 5 y 7 segundos de duración, como se muestra en la Tabla 2, a cada uno de estos escenarios se les aplica el procedimiento de prueba con la versión Python 3.8.3 tanto en el sistema operativo Windows 10 del cliente local y el cliente Ubuntu 18.03 del servidor HTTP.

S.O.	Navegador	Google Chrome	Microsoft Edge	Mozilla Firefox
Windows	Versión	84.0.4147.135	81.0.416.88	80.0
	Video	tos.mp4	tos.mp4	tos.mp4
	Segmentos 3 segundos	Reproducción Normal	Reproducción con Retroceso	Reproducción Normal
	Segmentos 5 segundos	Reproducción Normal	Reproducción Normal	Reproducción con Adelanto
	Video	sintel.mp4	sintel.mp4	sintel.mp4
	Segmentos 5 segundos	Reproduccion con Adelanto	Reproducción Normal	Reproducción con Retroceso
	Segmentos 7 segundos	Reproducción Normal	Reproducción Normal	Reproducción Normal
Ubuntu	Video	tos.mp4	tos.mp4	tos.mp4
	Segmentos 3 segundos	Reproducción Normal	Reproducción con Retroceso	Reproducción Normal
	Segmentos 5 segundos	Reproducción Normal	Reproducción Normal	Reproducción con Adelanto
	Video	sintel.mp4	sintel.mp4	sintel.mp4
	Segmentos 5 segundos	Reproduccion con Adelanto	Reproducción Normal	Reproducción con Retroceso
	Segmentos 7 segundos	Reproducción Normal	Reproducción Normal	Reproducción Normal

Tabla 2: Resumen de pruebas ejecutadas para diversos navegadores, vídeos y duración de segmentos.

Por último, se presentan escenarios con un segmento aleatorio intencionalmente alterado con el fin de probar la correcta funcionalidad del sistema ante la existencia de modificaciones en la comunicación, los resultados de cada ensayo son mostrados en la sección VI a continuación.

VI. RESULTADOS

Para un mejor análisis y compresión de los resultados, la sección se ha dividido de forma que se pueda explicar el porqué del éxito o fallo del sistema según el tipo de ensayo al que se ha sometido el vídeo en el reproductor. También se presenta una subsección con los resultados de distintas alteraciones premeditadas de un segmento cualquiera en una reproducción continua, como forma de verificación del algoritmo y, por último, una subsección donde se detalla el tiempo promedio de ejecución de cada método creado como parámetro de alusión a la eficiencia del código.

VI.1. *RESULTADOS PARA REPRODUCCIÓN CON ADELANTO O RETRASO*

El término reproducción con adelanto o retraso hace referencia a una reproducción continua que, en algún punto cualquiera de la comunicación, el usuario final decide dar un salto de uno o varios segundos fuera de búfer del contenido que se está recibiendo o ya fue recibido, alterando así el orden lógico del vídeo y provocando que al menos un segmento sea retransmitido o en su defecto no transmitido.

Aunque el correcto funcionamiento de este tipo de escenarios no formaba parte de un objetivo general de esta tesis, se realizaron ensayos para conocer el comportamiento del sistema en tales circunstancias, los resultados obtenidos en este tipo de pruebas fueron desfavorables, como se muestra en la Fig. 20 y Fig. 21.

Fig.20. Resultado del ensayo en explorador Google Chrome, vídeo Sintel y segmentos de 5 segundos.

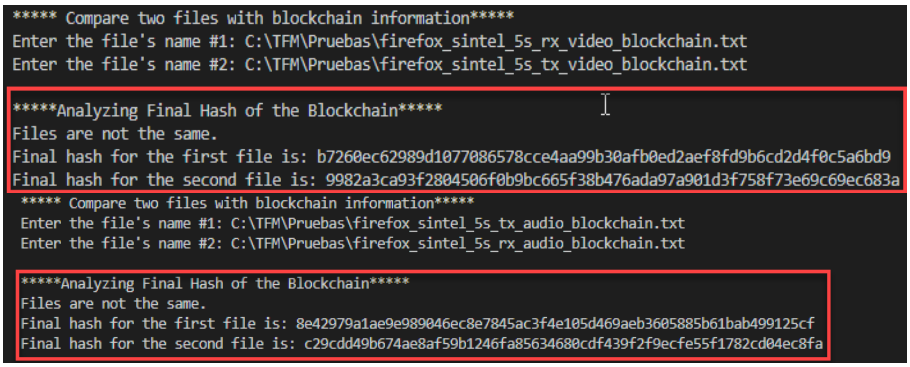

Fig.21. Resultado del ensayo en Mozilla Firefox para audio y vídeo Sintel con segmentos de 5 segundos.

La razón por la cual el sistema ha mostrado una falla es porque la herramienta FFMPEG, utilizada para el proceso de concatenación, arroja como resultado archivos de la misma duración del vídeo original independientemente del tipo de reproducción, como se muestra en la Fig. 22 y Fig. 23.

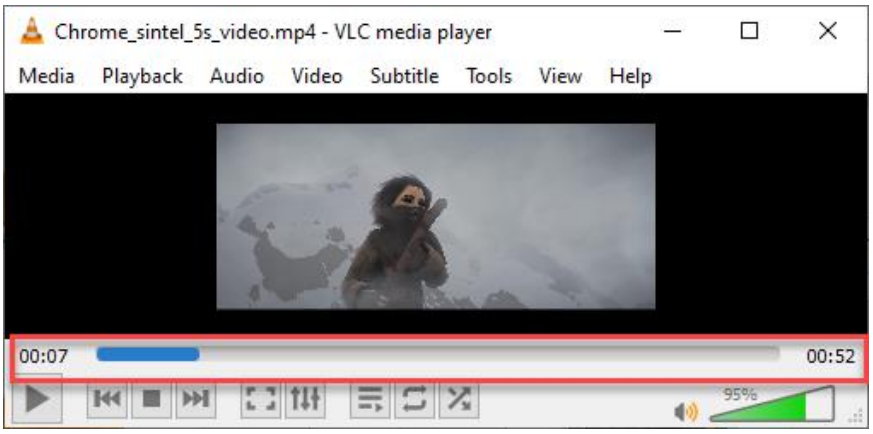

Fig.22. Reproducción resultante de concatenación del vídeo Sintel con segmentos de 5 segundos y adelantos.

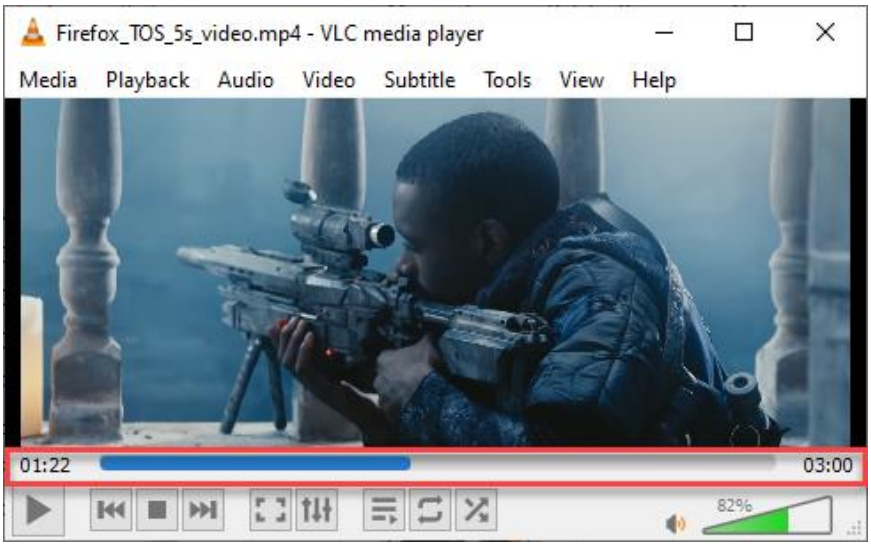

Fig.23. Reproducción resultante de concatenación del vídeo TOS con segmentos de 5 segundos y retrasos.

En el caso de detectarse un adelanto, los segmentos ausentes permanecen congelados en imagen y mudos en sonido, lo que provoca nuevos segmentos inexistentes cuando el archivo concatenado es fraccionado y por ende nuevos bloques que producirán disparidad en la comparación con la cadena obtenida en el transmisor. Para los escenarios con uno o más puntos de retroceso, la retransmisión de un segmento provoca la sobreescritura del o los segmentos repetidos, esto se traduce en la alteración de las referencias previamente marcadas en la cadena y por consiguiente otra disparidad en la comparación con el blockchain del transmisor.

Aunque estas limitantes están más ligadas a las capacidades de las herramientas externas utilizadas para la manipulación del vídeo, el perfeccionamiento en estos tipos de reproducción puede tomarse como una línea de trabajo futuro.

VI.2. *RESULTADOS PARA REPRODUCCIÓN CONTINUA*

Se entiende por reproducción continua a la comunicación donde el usuario final es capaz de ver todo el contenido sin necesidad de adelantarlo o retrocederlo, provocando que todos los segmentos que lo componen sean transmitidos en la secuencia lógica del vídeo.

Para este tipo de prueba, los resultados obtenidos en la comparación del blockchain del emisor y receptor fueron totalmente acertados en cada una de las pruebas emitidas, con ello, se puede afirmar la consecución de un método fiable para la verificación de vídeos transmitidos en comunicaciones de tipo streaming, sean estos de una sola calidad o de calidad variable, objetivo principal de este trabajo final de máster. Algunos de estos resultados positivos, son mostrados desde la Fig. 24 hasta la Fig. 29.

Un punto que vale la pena aclarar, es el hecho que el resultado final del análisis de una sesión streaming es totalmente ajeno al resultado de otra, es decir, que a partir del resultado de la creación de cadenas y su respectiva comparación para la sesión streaming de un usuario A, no puede ser utilizado para concluir el resultado de la sesión de un usuario B, pues las condiciones de ancho de banda que experimenta cada uno de ellos serán independientes y muy probablemente diferentes.

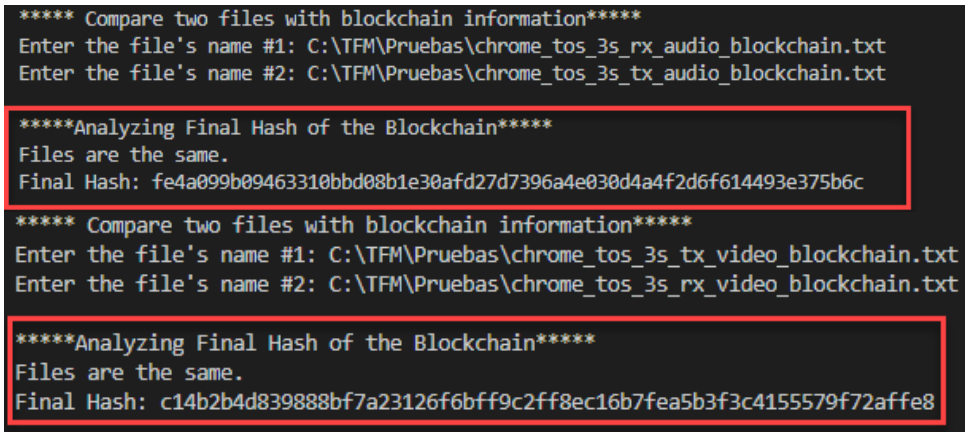

Fig.24. Resultados del ensayo en Google Chrome para audio y vídeo tos con segmentos de 3 segundos.

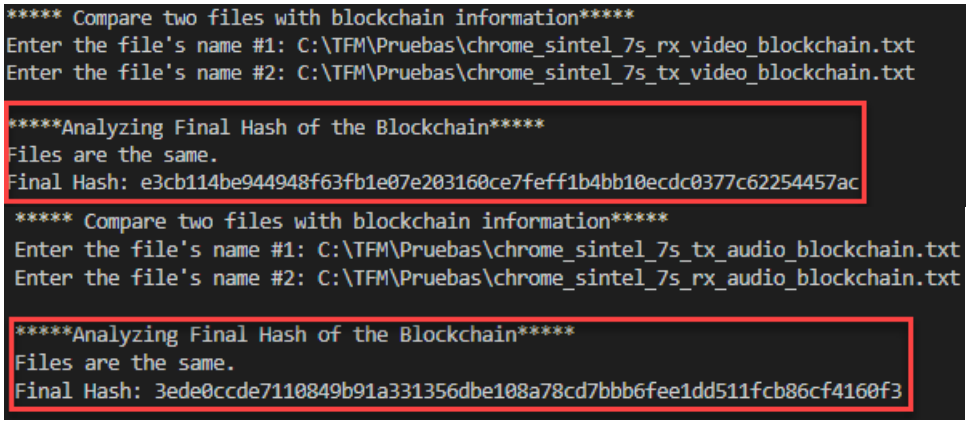

Fig.25. Resultados del ensayo en Google Chrome para audio y vídeo Sintel con segmentos de 7 segundos.

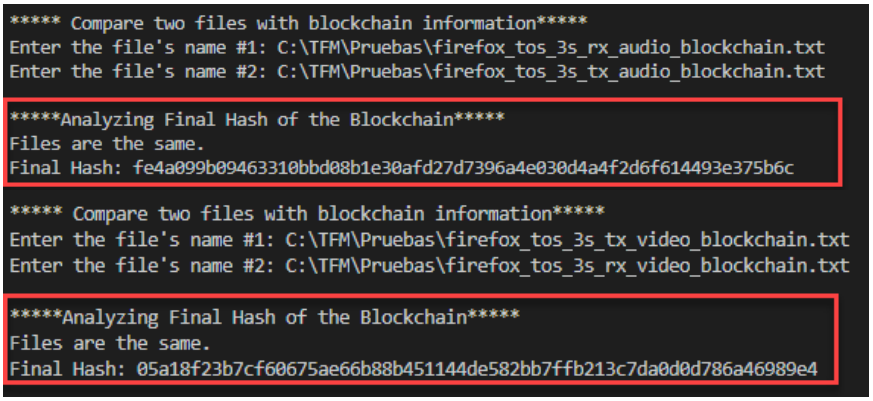

Fig.26. Resultados del ensayo en Mozilla Firefox para audio y vídeo tos con segmentos de 3 segundos.

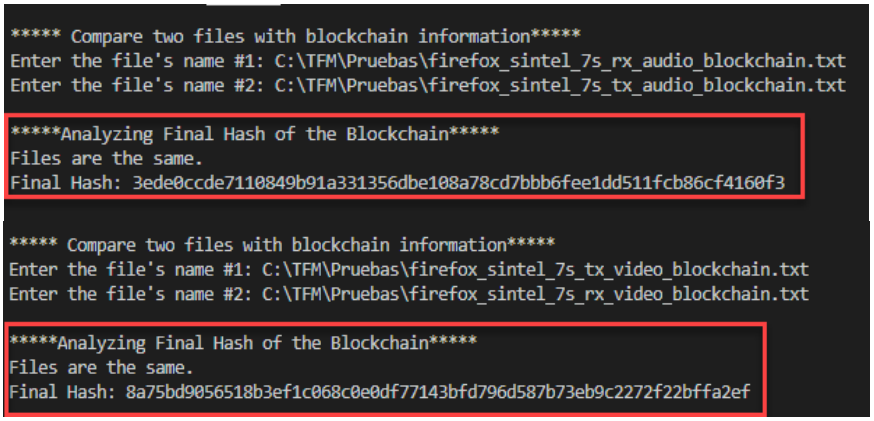

Fig.27. Resultados del ensayo en Mozilla Firefox para audio y vídeo Sintel con segmentos de 7 segundos.

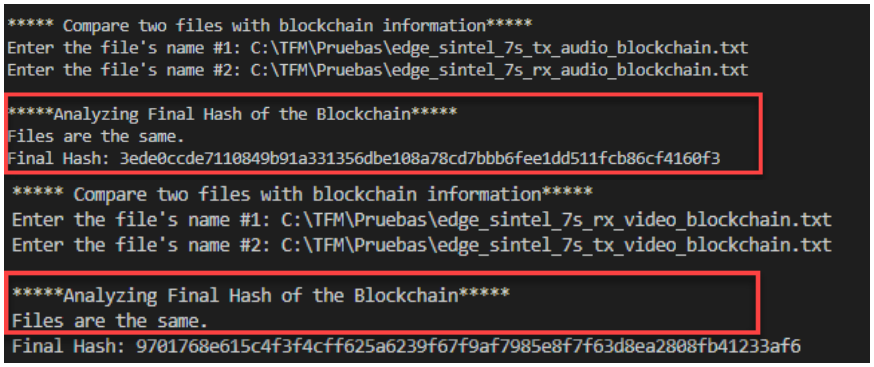

Fig.28. Resultados del ensayo en Microsoft Edge para audio y vídeo Sintel con segmentos de 7 segundos.

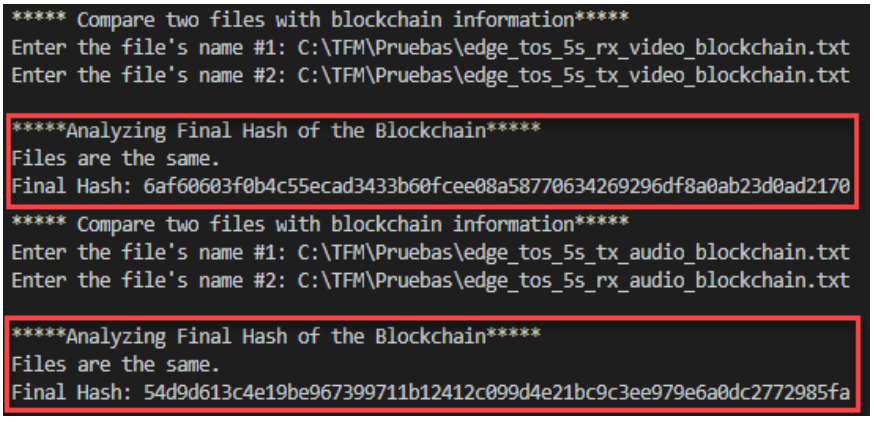

Fig.29. Resultados del ensayo en Microsoft Edge para audio y vídeo tos con segmentos de 5 segundos.

VI.3. *RESULTADOS PARA REPRODUCCIÓN CON ALTERACIONES*

Una vez descartado el uso del programa para los escenarios de reproducción con retraso o adelanto, faltaba corroborar el correcto funcionamiento del algoritmo al momento de modificar un segmento cualquiera de audio o vídeo en una reproducción continua. Para ello, será necesario el cambio de al menos un segmento cuando se esté ejecutando el módulo del receptor, específicamente antes de la instrucción mostrada en la Fig. 30, en ese momento elegimos cualquier otro segmento diferente al resultante de la fragmentación del vídeo, a este archivo sustituto es necesario alterarle el nombre según el patrón generado para que la generación de la cadena se realice sin inconvenientes, se salva el resultado y en el módulo de comparación deberá mostrar la discordancia desde el punto alterado.

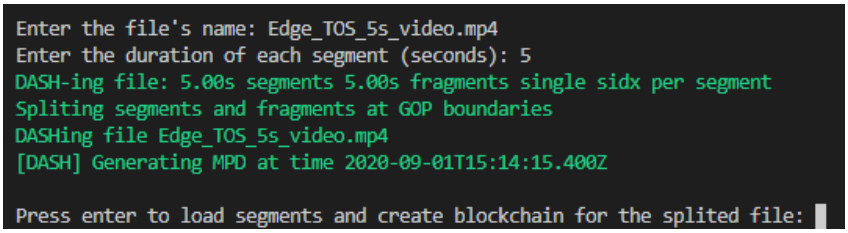

Fig.30. Instrucción para la creación del blockchain en el módulo del receptor.

La tabla 3 muestra las pruebas realizadas y su correspondiente resultado, estos a su vez son respaldados por la Fig. 31 hasta la Fig. 35.

Número de Prueba	Descripción	Tipo de Segmento	Segmento Original	Descripción	Segmento Alterado	Descripción	Discordancia Esperada	Resultado
1	Navegador Chrome Video Tos Segmentos 5 segundos	Video	video_360_21.m4s	Segmento #21 de calidad 360 pp de video TOS.	video_720_9.m4s	Segmento #9 de calidad 720 pp de video Sintel.	Bloque 20 en adelante	OK
2	Navegador Edge Video Sintel Segmentos 5 segundos	Video	video_720_6.m4s	Segmento #6 de calidad 720 pp de video Sintel.	video_1080_1.m4s	Segmento #1 de calidad 1080 pp de video TOS.	Bloque 5 en adelante	OK
3	Navegador Firefox Video Tos Segmentos 3 segundos	Video	video_1080_54.m4s	Segmento #54 de calidad 1080 pp de video TOS.	video_360_54.m4s	Segmento #54 de calidad 360 pp de video TOS.	Bloque 53 en adelante	OK
4	Navegador Chrome Video Sintel Segmentos 7 segundos	Video	video_360_2.m4s	Segmento #2 de calidad 360 pp de video Sintel.	video_1080_2.m4s	Segmento #2 de calidad 1080 pp de video Sintel.	Bloque 1 en adelante	OK
5	Navegador Edge Video Tos Segmentos 5 segundos	Audio	video_audio_12.m4s	Segmento de audio #12 de video TOS.	video_audio_2.m4s	Segmento de audio #2 de video Sintel.	Bloque 11 en adelante	OK

Tabla 3: Resumen de pruebas de alteración de segmentos y su resultado.

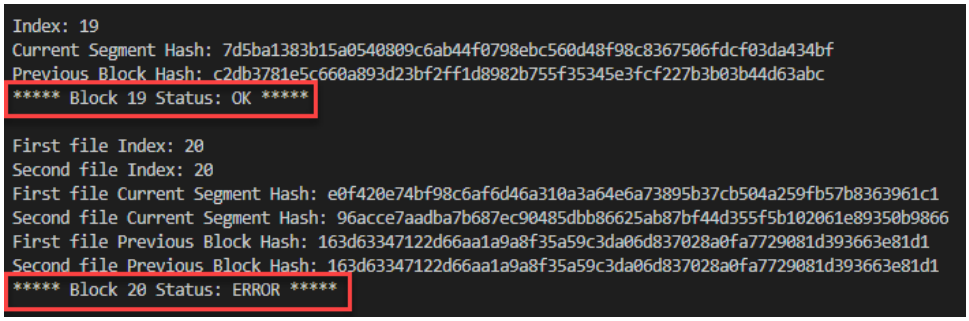

Fig.31. Resultado de la prueba 1 de la tabla 3.

Fig.32. Resultado de la prueba 2 de la tabla 3.

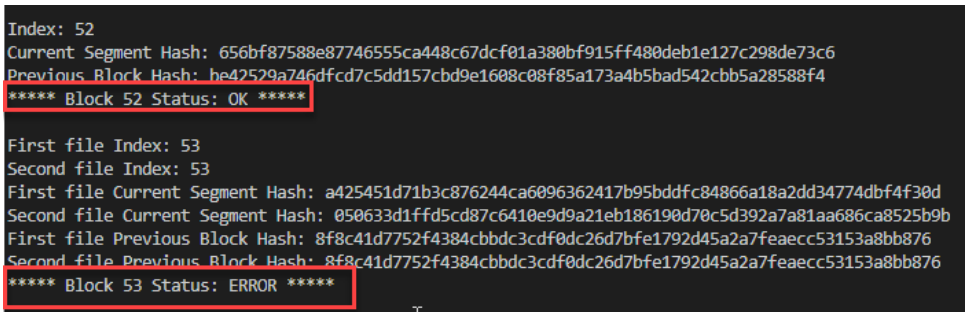

Fig.33. Resultado de la prueba 3 de la tabla 3.

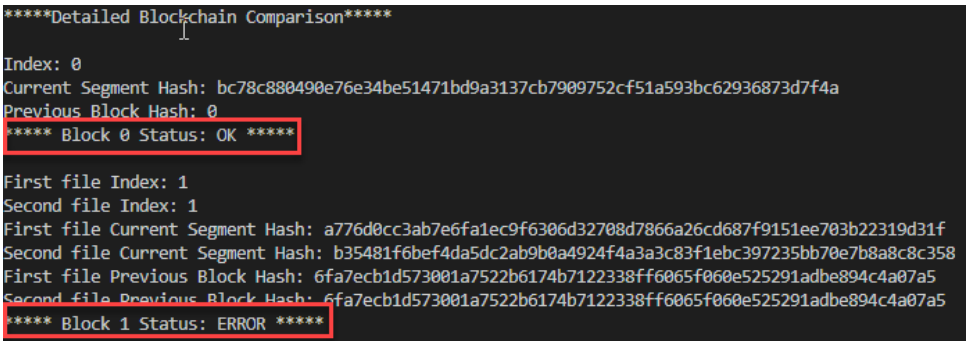

Fig.34. Resultado de la prueba 4 de la tabla 3.

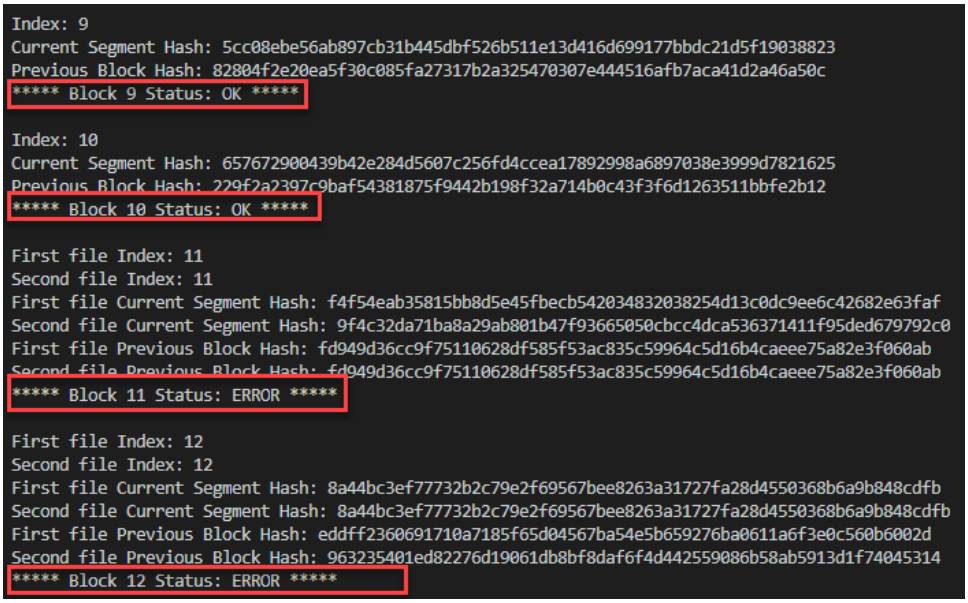

Fig.35. Resultado de la prueba 5 de la tabla 3.

VI.5. *ANÁLISIS DE EFICIENCIA*

Un algoritmo puede evaluar su eficiencia principalmente a través del estudio de su complejidad temporal y su complejidad espacial [24], para el primer caso, cada método del sistema ha registrado tiempos de ejecución bastante bajos, como lo muestra la tabla 4, donde apenas el proceso de concatenación de segmentos supera el orden de los milisegundos tanto en el cliente local como en el servidor HTTP.

Estos resultados de la complejidad temporal están estrechamente ligados a las cualidades lineales con las que se ideó cada método, pues se evitó totalmente el uso de ciclos anidados que aumentaran el orden de operaciones en los procesos. De esa forma, y gracias a la utilización de instrucciones y ciclos simples, se garantiza una complejidad espacial polinómica de grado 1, lo que conlleva a un aumento lineal en el consumo de recursos a medida aumente el tamaño del vídeo y el número de bloques en los que se ha divido la cadena.

Escenario	Proceso	Tiempo Windows	Tiempo Ubuntu	Escenario	Proceso	Tiempo Windows	Tiempo Ubuntu
Vídeo: TOS Segmentos: 3 segundos	Filtrado archivo HAR	341 ms.	144 ms.	Vídeo: Sintel Segmentos: 5 segundos	Filtrado archivo HAR	75 ms.	61 ms.
	Creación Audio Blockchain	325 ms.	118 ms.		Creación Audio Blockchain	43 ms.	41 ms.
	Creación Video Blockchain	379 ms.	181 ms.		Creación Video Blockchain	47 ms.	50 ms
	Concatenado	6 seg.	5 seg.		Concatenado	2 seg.	1.66 seg.
	Separación	529 ms.	507 ms.		Separación	362 ms.	246 ms.
	Comparación Final	9 ms.	0.04 ms.		Comparación Final	12 ms.	0.03 ms.
	Comparación Detallada	314 ms.	1 ms.		Comparación Detallada	112 ms.	0.3 ms.
Vídeo: TOS Segmentos: 5 segundos	Filtrado archivo HAR	219 ms.	127 ms.	Vídeo: Sintel Segmentos: 7 segundos	Filtrado archivo HAR	54 ms.	67 ms.
	Creación Audio Blockchain	168 ms.	80 ms.		Creación Audio Blockchain	25 ms.	24 ms.
	Creación Video Blockchain	189 ms.	156 ms.		Creación Video Blockchain	32 ms.	40 ms.
	Concatenado	5 seg.	3.8 seg.		Concatenado	3 seg.	1. 36 seg.
	Separación	498 ms.	399 ms.		Separación	323 ms.	155 ms.
	Comparación Final	14 ms.	0.07 ms.		Comparación Final	3 ms.	0.1 ms.
	Comparación Detallada	149 ms.	1.5 ms.		Comparación Detallada	44 ms.	0.3 ms.

Tabla 4: Mediciones de tiempo por módulo en reproducción continua según su ensayo.

Por último, es importante mencionar que, en las pruebas ejecutadas, la creación de un bloque individual llevo consigo un tiempo entre 2 y 5 milisegundos según el caso, valores muy aceptables para el tipo de aplicación. Dichos resultados se pueden observar en la tabla 5.

Descripción		Número Bloques	Tiempo Windows (ms)	Tiempo Ubuntu (ms)	Prom. Bloque Windows (ms)	Prom. Bloque Ubuntu (ms)
AUDIO	TOS - 3 seg	60	325	118	5.42	1.97
	TOS - 5 seg	36	168	80	4.67	2.22
	SINTEL - 5 seg	11	43	41	3.91	3.73
	SINTEL - 7 seg	8	25	24	3.13	3.00
VIDEO	TOS - 3 seg	60	379	181	6.32	3.02
	TOS - 5 seg	36	189	156	5.25	4.33
	SINTEL - 5 seg	11	47	50	4.27	4.55
	SINTEL - 7 seg	8	32	40	4.00	5.00

Tabla 5: Mediciones de tiempo en creación de blockchain según número de bloques y sistema operativo.

VII. CONCLUSIONES Y LÍNEAS FUTURAS

El porcentaje de tráfico generado por los servicios de multimedia streaming en internet es sustancialmente significativo, prueba de ello es el creciente número de usuarios en servicios de plataformas de audio o vídeo bajo demanda donde se estima que para el año 2025 habrán cerca de 571 millones de usuarios con acceso a este tipo de contenido. [25]

Esto, trae consigo una intrínseca necesidad de proteger la integridad de los datos multimedia que recibimos y aunque existen muchos tipos de soluciones cuya finalidad es preservar la rectitud de imágenes, audio y vídeo, actualmente se han difundido pocas o ninguna solución que sea capaz de garantizar la seguridad en servicios streaming.

En ese sentido, el sistema de verificación de la integridad del vídeo en sistemas DASH con blockchain desarrollado en este trabajo de fin de máster, representa un modelo base de carácter versátil e innovador para dar seguridad al audio y vídeo streaming, ya que la premisa en la cual se fundamentó cada uno de los módulos, demostró ser totalmente funcional para reproducciones continuas con o sin alteración dando un resultado correcto según el caso, de hecho, las limitaciones mostradas en los escenarios con adelanto o retraso, obedecen a las restricciones de concatenación y división que presentan las herramientas FFMPEG y MP4BOX utilizadas, razón que no influye en absoluto al objetivo general planteado originalmente.

Con respecto a su utilidad práctica, los módulos del sistema o la lógica de ellos, pueden ser utilizados por proveedores de contenido multimedia bajo demanda como IPTV, Amazon, Netflix, YouTube e incluso Spotify, pues se demostró que el audio por sí solo, aunque carezca de calidad variable, es perfectamente verificable en el sistema, de esa forma, dichas compañías pueden garantizar que el usuario recibe el contenido correcto, en la calidad correcta y sin interferencia de terceros o incluso podrían comparar contenidos entre sí, con el objetivo de identificar vídeos duplicados que usuarios maliciosos copian con el fin de monetizar en nombre de otra persona.

En cuanto a las líneas de trabajo futuro, el presente trabajo final de máster deja abierta la posibilidad de corregir los escenarios en los cuales el sistema no es aplicable actualmente, esto será posible mediante la creación de una herramienta de modelado de audio y vídeo que permita la reconstrucción del contenido de forma exacta aunque exista un adelanto o retraso en su reproducción, de esa forma se garantiza una correcta funcionalidad, independientemente de la forma en la que se haya percibido el contenido.

Como caso ideal de funcionamiento, se podría trabajar directamente en la estructura del reproductor de forma que los módulos realicen el trabajo de comparación de segmentos en tiempo real, aplicando al segmento que llegue al receptor el procedimiento de creación de bloque y hash final para ser comparado inmediatamente como si este fuese el último de la cadena y así no sea necesario la reconstrucción del contenido, lo que traería consigo un mejor rendimiento general y una retroalimentación inmediata.

Por último, una vez finalizado el proceso de defensa, se estableció como línea de trabajo futura la elaboración de un artículo con el objetivo de que la investigación realizada en la elaboración del sistema y sus funcionalidades sean publicadas en alguna revista científica.

AGRADECIMIENTOS

La vida, una montaña rusa de constantes subes y bajas, un camino de vivencias impredecibles que puede llevarte a destinos impensados en un tiempo preciso. Parece que fue ayer cuando todo quedó en pausa a ocho mil kilómetros de distancia mientras cargaba mi vida en dos maletas y un sueño. No había por qué sufrir, esa era la consigna, pues al final, el embalaje de la ilusión lo selló papá con aquel último abrazo en la sala del aeropuerto deseándome lo mejor y que vino a mi memoria en este momento. Hoy, a escasos días de terminar esta fantástica aventura, abundan las vivencias que marcaron el camino de este curso.

Por ello, quiero agradecer primeramente a Dios, que a lo largo de mi carrera educativa y profesional me ha llevado a todos los lugares donde he deseado, siendo fiel a su palabra de conceder todo aquello que pidamos en su nombre. Agradezco a mi padre Juan Euceda y mi madre Gloria Pastrana que a pesar de sus limitaciones creyeron y apoyaron cada una de mis aspiraciones, aunque estas parecieran imposibles, la consecución de cada logro es más mérito suyo que mío. A mis hermanos Claudia, Ceily y Víctor que a lo largo de los años han sabido ser una guía, una fuerza y una motivación para soñar alto, a mis sobrinos Valery, Waleska, Rodrigo, Odalys, Andrew, Paula y Valentina que con cada ocurrencia me enseñan a ver las cosas de una manera simple, como solo un niño podría hacerlo. Agradezco a Ámbar, Carlos y Brolin, mis hermanos por elección cuya amistad me hace dar siempre lo mejor de mí pues han demostrado su incondicionalidad desde hace más de 15 años, a Ana Raquel por estar pendiente en cada día de este episodio de mi vida que deseo culminar con un abrazo suyo. A todos mis amigos, compañeros de máster y demás familiares que sería difícil de mencionar y que mostraron su apoyo de principio a fin, al programa de becas Honduras 2020que, sin su ayuda, la consecución de este sueño no hubiera sido posible, muchas gracias por dejarme representar dignamente a nuestro amado país.

Por último, agradezco a la Universidad Politécnica de Valencia por la posibilidad brindada al momento de aceptar mi postulación a la maestría, sin duda el haber venido ha sido una de las mejores decisiones de mi vida y espero haber estado a la altura de las circunstancias, también agradecer a mis asesores Juan Carlos Guerri y Pau Arce que confiaron en mis capacidades desde el primer minuto para desarrollar esta tesis de la mejor manera.

Un hombre no anda solo en su caminar, siempre habrá personas que harán de la marcha un recuerdo igual de valioso que la meta misma.

BIBLIOGRAFÍA

[1] J. Euceda, "SISTEMA PARA LA INTEGRIDAD DEL VÍDEO EN SISTEMAS DE STREAMING DASH UTILIZANDO BLOCKCHAIN - YouTube," 2020. [Online]. Available: https://www.youtube.com/playlist?list=PLPGtBL3IVBH7hq8cf_eLWq-s0SdJMFQNi. [Accessed: 09-Sep-2020].

[2] The World Bank Group, "Individuals using the Internet (% of population)," 2019. [Online]. Available: https://data.worldbank.org/indicator/it.net.user.zs. [Accessed: 30-Mar-2020].

[3] DOMO Inc., "Data Never Sleeps 7.0 Infographic," 2020. [Online]. Available: https://www.domo.com/learn/data-never-sleeps-7?utm_source=wire&utm_medium=pr&utm_campaign=ABM_Other_FY20_Global_PR&campid=701f2000001C3kNAAS. [Accessed: 30-Mar-2020].

[4] O. I. Al-Sanjary and G. Sulong, "Detection of video forgery: A review of literature," *J. Theor. Appl. Inf. Technol.*, 2015.

[5] K. Bouraqia, E. Sabir, M. Sadik, and L. Ladid, "Quality of Experience for Streaming Services: Measurements, Challenges and Insights," *IEEE Access*, vol. 8, pp. 13341–13361, 2020.

[6] A. Rao, A. Legout, Y. S. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," *Proc. 7th Conf. Emerg. Netw. Exp. Technol. Conex.*, 2011.

[7] Panopto, "7 Things That Define Modern Video Streaming & How It Scales," 2019. [Online]. Available: https://www.panopto.com/blog/just-what-is-modern-streaming-seven-characteristics-that-define-the-next-shift-in-video-technology/. [Accessed: 07-Apr-2020].

[8] Universidad Politécnica de Valencia, "MPEG-DASH (streaming adaptativo basado en HTTP) | | UPV - YouTube." [Online]. Available: https://www.youtube.com/watch?v=d3vN7zkNGBY. [Accessed: 08-Apr-2020].

[9] I. Sodagar, "White paper on MPEG-DASH Standard." Communication Group, 2012.

[10] Universidad Politécnica de Valencia, "Componentes del fichero índice o MPD de MPEG-DASH | | UPV - YouTube." [Online]. Available: https://www.youtube.com/watch?v=fon46LL5GT8. [Accessed: 08-Apr-2020].

[11] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, 2017, pp. 557–564.

[12] HIMSS, "Cryptography in Blockchain." [Online]. Available: https://www.himss.org/resources/cryptography-blockchain. [Accessed: 09-Apr-2020].

[13] JAXenter, "How cryptographic algorithms and hashing keep blockchain secure." [Online]. Available: https://jaxenter.com/cryptographic-hashing-secure-blockchain-149464.html. [Accessed: 09-Apr-2020].

[14] R. Rivest, "Computer and Network Security," 1997. [Online]. Available: http://web.mit.edu/6.857/OldStuff/Fall97/lectures/lecture9.pdf. [Accessed: 12-Apr-2020].

[15] P. Gupta and S. Kumar, "A Comparative Analysis of SHA and MD5 Algorithm," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 3, pp. 4492–4495, 2014.

[16] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[17] R. A. Dobre, R. O. Preda, C. C. Oprea, and I. Pirnog, "Authentication of JPEG Images on the Blockchain," *Proc. - 2018 Int. Conf. Control. Artif. Intell. Robot. Optim. ICCAIRO 2018*, pp. 211–215, 2018.

[18] M. Harran, W. Farrelly, and K. Curran, "A method for verifying integrity & authenticating digital media," *Appl. Comput. Informatics*, vol. 14, no. 2, pp. 145–158, 2018.

[19] V. Yatskiv, N. Yatskiv, and O. Bandrivskyi, "Proof Video Integrity Based on Blockchain," 2019.

[20] A. Hemlin Billström and F. Huss, "Video Integrity through Blockchain Technology," 2017.

[21] B. Gipp, J. Kosti, and C. Breitinger, "Securing Video Integrity Using Decentralized Trusted Timestamping on the Bitcoin Blockchain," *Proc. 10th Mediterr. Conf. Inf. Syst.*, no. September, p. 51, 2016.

[22] I. Hubert, T. Roosendaal, and J. Kerbosch, *Tears of Steel - 4k version (in HD) - Blender Foundation channel - YouTube*. .

[23] Blender Foundation, *"Sintel" Trailer, Durian Open Movie Project - YouTube*. .

[24] A. De Leon, "Complejidad Algorítmica – Ejemplos Python, PHP – Mi Camino Master," 2018. [Online]. Available: http://micaminomaster.com.co/grafo-algoritmo/complejidad-algoritmica-ejemplos-python-php/. [Accessed: 02-Sep-2020].

[25] Statista, "Video-on-Demand - Users worldwide 2025 | Statista," 2020. [Online]. Available: https://www.statista.com/forecasts/456771/video-on-demand-users-worldwide-forecast. [Accessed: 03-Sep-2020].