Autenticación de Vídeos MOV usando el Contenedor H.264/AVC

Video MOV Authentication using H.264/AVC Container

TRABAJO FIN DE MÁSTER

MÁSTER EN INGENIERÍA INFORMÁTICA

CURSO 2019–2020

Roberto Edward Gastiaburu Tovar

Director

Luis Javier García Villalba

Colaborador Externo

Ana Lucila Sandoval Orozco

Nota: Sobresaliente (9.0)

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, Junio de 2020

Agradecimientos

Quiero expresar mi agradecimiento a mis directores, Javier y Ana Lucila, quienes me han brindado la oportunidad de realizar este trabajo, sin su ayuda, dedicación y conocimientos, esto no sería viable. Agradecer el apoyo y ayuda frente a la inconvenientes que se han dado mientras se elaboró esta investigación a todo el Grupo de Análisis, Seguridad y Sistemas (GASS).

De manera especial a Dios porque cada día bendice mi vida con la oportunidad de permanecer y gozar al lado de las personas que amo y me aman. A mi querida familia, que lo son todo, mi mujer Erika, mis hijos Iñaki, mi angelita Nerea y mi niña Ainhoa que está por llegar, que son mi fuente de inspiración, impulso y las energías necesarias para la culminación de este trabajo, por su constante amor, apoyo y sacrificio para que esto llegué a buen puerto. A mis padres, Cecilia y Roberto, por haberme forjado como la persona que soy en la actualidad, con reglas, libertades y que me motivaron para alcanzar mis anhelos, muchos de mis logros se los debo a ustedes entre los que se incluye éste. A mis abuelos Amalia, Consuelo, Vicente y Víctor, que desde el cielo velan por mi bienestar y el de mi familia. A mis demás familiares y amigos por formar parte de mi vida, por creer en mí y en la realización de este trabajo.

Este Trabajo Fin de Máster ha sido realizado dentro del grupo de investigación GASS (Grupo de Análisis, Seguridad y Sistemas, grupo 910623 del catálogo de grupos reconocidos por la UCM) como parte de las actividades del proyecto de investigación RAMSES (Internet Forensic Platform for Tracking the Money Flow of Financially-Motivated Malware) financiado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (H2020-FCT-2015/700326-RAMSES).

Índice General

Índice de Figuras

Índice de Tablas

Resumen

En la actualidad la cámara fotográfica más popular del mundo es el teléfono móvil. Estos dispositivos traen nuevas funciones que crecen vertiginosamente y hacen que sean de uso general por la mayoría de personas. Todavía quedan cosas por mejorar pero dada la inversión realizada en ellos, es solo cuestión de tiempo para que se hagan con el mercado. Este marco hace que el análisis a nivel forense de vídeos tome mucho peso y sea necesario y significativo en un sinnúmero de circunstancias como pueden ser pruebas judiciales, pesquisas, abuso de menores, sustitución de identidad, etc.En este trabajo se ha desarrollado un método que analiza el contenido del contenedor de vídeos con formato MOV. La técnica se cimienta en la extracción de información contenida en el contenedor multimedia y operaciones de análisis de datos. El objetivo del método es analizar la información extraída a nivel de dispositivos, modelos y fabricantes de dispositivos móviles para identificar el origen de un vídeo dado. Además, haciendo uso de otra herramienta previamente desarrollada se realiza comparaciones de la información extraída de vídeos procesados por redes sociales y aplicaciones de edición de vídeos. Las herramientas u origen de información usadas para el progreso de la técnica propuesta son por un lado, las especificaciones de los estándares y por otro, las investigaciones previas dedicadas al análisis de otros tipos de contenedores multimedia. Asimismo, para el análisis se hace uso de un dataset de vídeos con características esenciales para que el resultado sea eficiente y lo más preciso posible.

Palabras clave: Metadatos, Consumo de Vídeo, Cámara Móvil, Formato de Película Quick Time, Redes Sociales, Edición de Vídeo, Análisis Forense, Métodos Forenses.

Abstract

Currently the most popular camera in the world is the mobile phone. These devices bring new functions that grow rapidly and make them commonly used by most people. There are still things to improve but given the investment made in them, it is only a matter of time for them to take over the market. This framework makes the forensic analysis of videos take a lot of weight and is necessary and significant in countless circumstances such as judicial evidence, investigations, child abuse, identity substitution, etc. This work has developed a method that analyzes the content of the container of videos in MOV format. The technique is based on the extraction of information contained in the multimedia container and data analysis operations. The objective of the method is to analyze the information extracted at the level of devices, models and manufacturers of mobile devices to identify the origin of a given video. In addition, using another previously developed tool, comparisons are made of the information extracted from videos processed by social networks and video editing applications. The tools or source of information used for the progress of the proposed technique are, on the one hand, the specifications of the standards and, on the other, the previous investigations dedicated to the analysis of other types of multimedia containers. Also, the analysis uses a video dataset with essential characteristics so that the result is efficient and as accurate as possible.

Keywords: Metadata, Video Consumption, Mobile Camera, Quick Time Movie Format, Social Networks, Video Editing, Forensic Analysis, Forensic Methods.

Lista de Acrónimos

AAC	Advanced Audio Coding
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
AVC	Advanced Video Coding
AVI	Audio Video Interleaved
CCD	Charge Coupled Device
CFA	Color Filter Array
DCT	Discrete Cosine Transform
DIP	Digital Image Processor
DSP	Digital Signal Processing
DVD	Digital Versatile Disc
DVD-AR	Digital Versatile Disc Audio grabable
EXIF	Exchangeable Image File Format
FLAC	Free Lossless Audio Codec
FLV	Flash video
FRUC	Frame Rate Up Conversion

GOP	Group of Pictures
GSM	Global System for Mobile Communications
HD	High Definition
HDR	High Dynamic Range
HEVC	High Efficiency Video Coding
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
KNN	K-Nearest Neighbors
LFE	Low Frequency Effects
MCEA	Motion Compensated Edge Artifacts
MIDI	Musical Instrument Digital Interface
MKV	Matroska Video
MOV	Quick Time Movie Format
MP4	Media Player 4
MPEG	Moving Picture Experts Group
MXF	Material Exchange format

PAL	Phase Alternating Line
PIB	Producto Interno Bruto
QTFF	QuickTime File Format
RGB	Red Green Blue
SD	Standard Definition
SVM	Support Vector Machine
VEDR	Video Event Data Recorders
WMA	Windows Media Audio
YUV	Luminance Bandwidth Chrominance

Capítulo 1

Introducción

En [Zif18] se indica que no es ningún secreto que la cámara más popular del mundo en la actualidad es el teléfono. Paralelamente, las ventas de cámaras independientes se han hundido. El énfasis de Apple en las impresionantes nuevas funciones de la cámara en sus últimos iPhones ha subrayado la cuestión de si es solo cuestión de tiempo antes de que el teléfono realmente sea el propietario del mercado de cámaras de uso general. Algunas cosas que mejorar en las cámaras de los dispositivos móviles serían las lentes, los sensores, la funcionalidad de alto rango dinámico (del inglés High Dynamic Range (HDR)) y el zoom o teleobjetivo. Dada la inversión masiva en teléfonos, es solo cuestión de tiempo antes de que reemplacen cada segmento del mercado de cámaras del que son físicamente capaces.

En [Inc19] se establece que las imágenes y vídeos utilizados en distintas aplicaciones, redes sociales como son YouTube, Facebook, LinkedIn, WhatsApp y Telegram, además de editores de vídeo como Adobe son generadas por cámaras de dispositivos móviles.

En [Ass19] se describe que a fines de 2018, como se ve en la Figura 1.1 la suscripción a servicios móviles en el mundo representa el 67 % de la población en el mundo, llegando a cifras de 5,1 mil millones de personas. Se han agregado un total de mil millones de nuevos suscriptores en los cuatro años desde 2013 lo que simboliza una contribución del 5 % al crecimiento anual medio, pero la velocidad de crecimiento se está desacelerando. La contribución al crecimiento anual medio de 1.9 % entre 2018 y en 2025 se incrementará a 5.8 mil millones el número total de suscriptores móviles, 71 % de toda la población.

Se espera que se suscriban unos 710 millones de personas a servicios móviles por primera vez en los próximos siete años, la región de Asia Pacífico representará la mitad y el África Subsahariana representará menos de la cuarta parte.

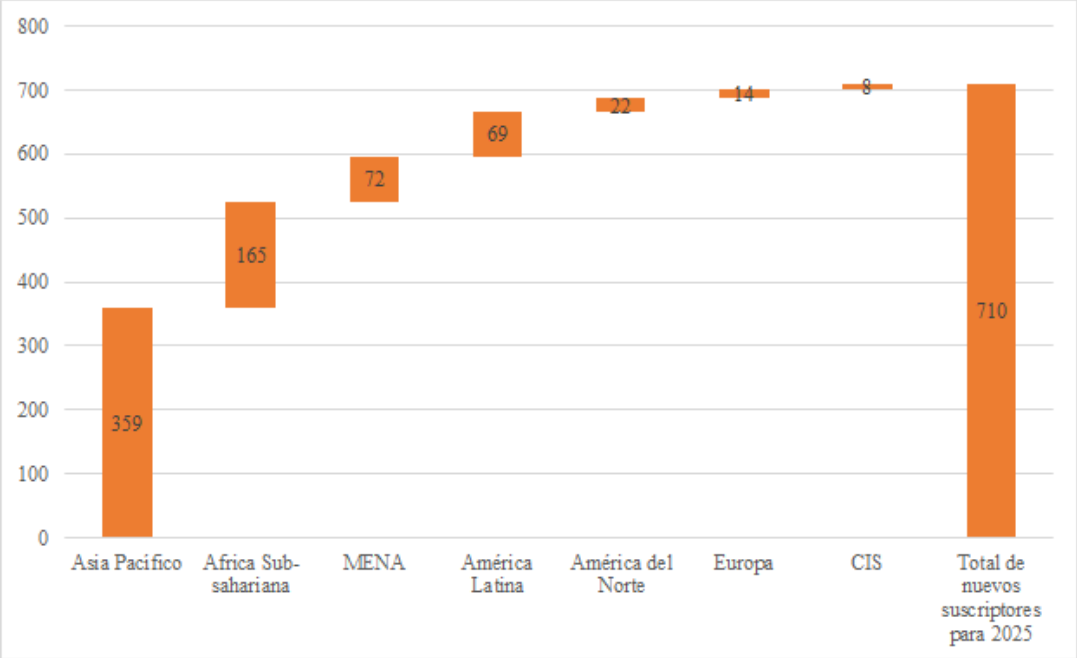


Figura 1.1: Suscriptores para 2025

Mientras tanto, los dispositivos móviles continúan haciendo una aportación valiosa al desarrollo socioeconómico en todo el mundo. Como se ve en la Figura 1.2 en 2018, las tecnologías y servicios móviles generaron $ 3.9 billones de valor económico 4.6 % del Producto Interno Bruto (PIB) a nivel mundial, una aportación que alcanzará $ 4.8 billones, 4.8 % del PIB para 2023, a su vez los países se benefician cada vez de las mejoras en productividad y eficiencia generadas por mayor uso de servicios móviles. Se espera en los próximos 15 años que las tecnologías 5G contribuyan con $2.2 billones a la economía del mundo.

La brecha de conectividad también continúa cerrándose: en los próximos siete años, 1.400 millones de personas usarán Internet móvil por primera vez, lo que traerá que el número total de suscriptores de internet móvil a nivel mundial sea de 5 mil millones para 2025, más del 60 % de la población. Este crecimiento en la conectividad está ayudando a la industria móvil a aumentar su impacto en todas las metas de crecimiento sustentable de las Naciones Unidas y está impulsando la adopción de herramientas y soluciones basadas en dispositivos móviles por ejemplo, en agricultura, educación y atención médica que apuntan a mejorar los medios de subsistencia países de ingresos medios, según el reporte realizado por la Asociación del Sistema Global de Comunicaciones (del inglés Global System for Mobile Communications (GSM)).

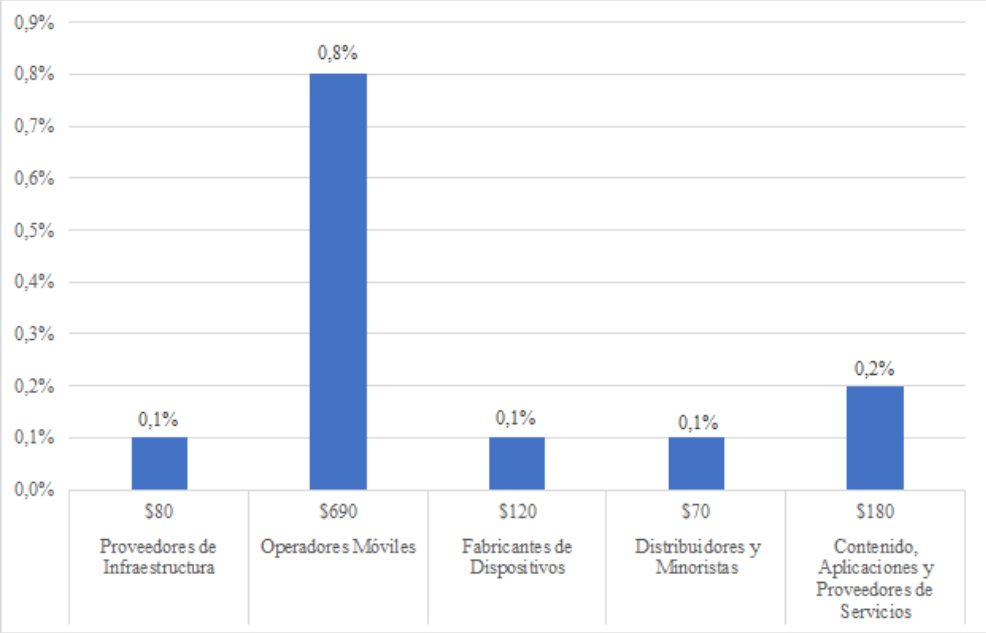


Figura 1.2: Ecosistema móvil económico

1.1. Contexto

El presente Trabajo Fin de Máster se enmarca dentro de un proyecto de investigación titulado RAMSES aprobado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (Convocatoria H2020-FCT-2015, Acción de Innovación, Número de Propuesta: 700326)y en el que participa el Grupo GASS del Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid (Grupo de Análisis, Seguridad y Sistemas, http://gass.ucm.es, grupo 910623 del catálogo de grupos de investigación reconocidos por la UCM).

Además de la Universidad Complutense de Madrid participan las siguientes entidades:

- Treelogic Telemática y Lógica Racional para la Empresa Europea SL (España)
- Ministério da Justiça (Portugal)
- University of Kent (Reino Unido)
- Centro Ricerche e Studi su Sicurezza e Criminalità (Italia)
- Fachhochschule fur Offentliche Verwaltung und Rechtspflege in Bayern (Alemania)
- Trilateral Research & Consulting LLP (Reino Unido)
- Politecnico di Milano (Italia)
- Service Public Federal Interieur (Bélgica)
- Universitaet des Saarlandes (Alemania)

- Dirección General de Policía - Ministerio del Interior (España)

1.2. Objeto de la Investigación

Estudiar los átomos que la componen como se extraen los metadatos y la elaboración un análisis de la organización de un vídeo digital de forma general, con el propósito de examinar y entender los principales. Para ello se necesita los siguientes objetivos:

- Analizar detalladamente la especificación de almacenamiento de información de vídeos digitales con formato MOV (del inglés Quick Time Movie Format (MOV)).
- Diseñar e implementar un algoritmo que permita la extracción automática de metadatos y datos almacenados en vídeos de dispositivos móviles.
- Analizar la estructura de los vídeos para su posterior identificación y diferenciación con respecto al formato MP4 (del inglés Media Player 4 (MP4)).

1.3. Estructura del Trabajo

Este trabajo tiene una división de 7 capítulos, el presente en su totalidad forma parte del primero.

En el Capítulo 2 se introducen explicaciones esenciales sobre los vídeos, sistemas de compresión de vídeo, sistemas de compresión de audio y los diferentes contenedores multimedia.

En el Capítulo 3 se muestra la bibliografía sobre las técnicas forenses y antiforenses relacionadas con la autenticación del contenido de un vídeo.

En el Capítulo 4 se presenta las aportaciones de este trabajo. La estructura de los metadatos, el algoritmo de extracción de átomos y metadatos y la tecnología utilizada para tal fin.

En el Capítulo 5 se explica la experimentación realizada en este trabajo, la obtención de la muestra y presenta los resultados obtenidos para vídeos sin procesamiento y con procesamiento.

En el Capítulo 6 se exhiben el producto del análisis y las líneas de futura investigación proveniente del presente trabajo.

En el Apéndice A se muestra un anexo con una descripción de los átomos del Contenedor Multimedia MOV y unas capturas de pantalla de las aplicaciones ExifToolGUI y Mp4 Parser a unos vídeos analizados.

Capítulo 2

Vídeos Digitales

Este capítulo examina la estructura y características de las imágenes digitales y las señales de vídeo e introduce conceptos como formatos de muestreo y métricas de calidad que son útiles para una comprensión de la codificación de vídeo. El objetivo del capítulo es dar una visión sobre los vídeos digitales, abordando en la Sección 2.1 el proceso de generación de un vídeo digital, la Sección 2.2 el procesamiento de la escena y del audio, luego en la Sección 2.3 la compresión del vídeo y del audio, asimismo, en la Sección 2.4 revisamos la parte del encapsulamiento, describiendo los principales contenedores de vídeo y por último en la Sección 2.5 se describe el formato de contenedor multimedia MOV.

2.1. Proceso de Generación de un Vídeo Digital

El vídeo digital es una representación de un mundo natural o real, es una escena visual, muestreada espacial y temporalmente. Se muestrea una escena en un punto en el tiempo para producir un marco, una representación de la escena visual se completa en ese momento o un campo que consiste en líneas impares o pares de muestras espaciales. El muestreo se repite en intervalos de 1/25 o 1/30 segundos para producir una señal de vídeo en movimiento. Tres conjuntos de muestras o componentes, generalmente se requieren para representar una escena en color.

Una escena de vídeo típica del mundo real o natural está compuesta de múltiples objetos, cada uno con su propia forma característica, profundidad, textura e iluminación. El color y el brillo de una escena de vídeo natural cambia con diferentes grados de suavidad en toda la escena o tono continuo. Características de una típica escena de vídeo natural, serán relevantes para el procesamiento y la compresión de vídeo, incluyen características espaciales, variación de textura dentro de la escena, número y forma de objetos, color, etc. y características temporales, objeto en movimiento, cambios en la iluminación, movimiento de la cámara o punto de vista, etc.

La sistema fundamental de generación de un vídeo digital se describe en la Figura 2.1, primero la fase de procesamiento de la escena y audio, luego la compresión de vídeo y audio, para finalmente realizar el proceso de encapsulación.

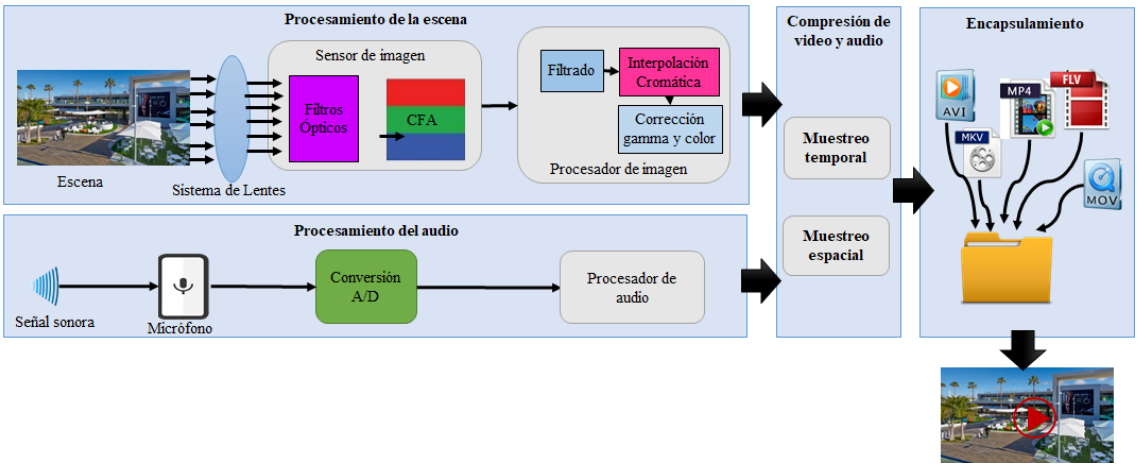


Figura 2.1: Procesamiento de un vídeo digital

2.2. Procesamiento de la Escena y del Audio

El procedimiento para generar una imagen es muy semejante para la mayoría de las cámaras digitales y es capturado junto a una secuencia de audio y en algunos casos datos. La composición de una cámara fotográfica presenta un grupo de lentes, un sistemas de filtros, una matriz de filtro de colores o (del inglés Color Filter Array (CFA)), un sensor de imagen y un procesador de imagen o (del inglés Digital Image Processor (DIP)). Los pormenores de la imagen son testimonio reservado de los fabricantes de cámaras fotográficas. Para crear un vídeo se efectúan dos tareas, el procesamiento de la escena y el procesamiento de audio.

El procesamiento de la escena se inicia en una cámara digital, la luz que ingresa a la cámara a través de la lente se filtra primero y se enfoca en una matriz de elementos de dispositivos acoplados a carga Charge Coupled Device (CCD), es decir, píxeles. La matriz CCD es el componente principal y más costoso de una cámara digital. Cada detección de luz, el elemento de la matriz CCD integra la luz incidente sobre todo el espectro y obtiene una representación de la señal eléctrica del paisaje. Dado que cada elemento CCD es esencialmente la captura de imágenes monocromáticas en color requiere matrices CCD separadas para cada componente de color. Debido a los costos, en general las cámaras digitales, se utiliza un solo conjunto CCD disponiéndolos en un patrón donde cada elemento tiene un filtro espectral diferente, típicamente uno de rojo, verde o azul (del inglés Red Green Blue (RGB)). Esta máscara frente al sensor y se llama matriz de filtro de color CFA. En consecuencia, cada elemento CCD solo detecta una

banda de longitudes de onda, y la imagen en bruto recogido de la matriz es una paleta de píxeles rojos, verdes y azules. Dado que el sistema visual humano es más sensibles a la luz verde, los patrones CFA suelen tener más valores verdes que el rojo y el azul. La huella se transforma en una huella digital y se traslada al procesador de imagen que forma una pieza del Digital Signal Processing (DSP). La huella digital producida por sensores es recibida por el procesador de imagen que la expone a diferentes fases, con el fin de consolidar la señal y enmendar modificaciones, como suprimir el ruido y otros fallos. Luego, la huella se consolida pasa a la fase de compresión mediante un códec preciso, luego se encapsula en un depósito que fundamente los datos. En las cámaras de teléfonos se emplea H.264 o MPEG-4 parte 10, que es la regla que define el códec de alta compresión de vídeo, con un contenedor multimedia tipo MOV. Este contenedor tiene la capacidad de guardar o encapsular vídeo y audio. Al final el fichero con extensión .MOV será almacenado en un artefacto para tal fin.

En [QEAL16] se establece que el proceso para generar el audio empieza en el aire cuando la huella sonora, se transfiere y es capturada por el micrófono que trabaja como un sensor electro acústico, que modifica las ondas sonoras en una huella eléctrica para incrementar su potencia y transferir a un conversor analógico digital, que la transforma a una huella digital. La huella digital es capturada por el procesador de audio digital, que la procesa para obtener la calidad de audio antes de la compresión. Luego se comprime para ser encapsulado en un contenedor y guardado en un dispositivo. Las cámaras de los teléfonos emplea el algoritmo de compresión con pérdida (del inglés Advanced Audio Coding (AAC)) que es un formato de huella digital establecido en MPEG-4 parte 3.

2.3. Compresión de Vídeo y Audio

Una escena visual natural es espacial y temporalmente continua. Una escena visual en la forma digital implica muestrear la escena real espacialmente, en una cuadrícula rectangular en el plano de imagen de vídeo y tiempo, como una serie de fotogramas fijos o componentes de fotogramas muestreados a intervalos regulares en el tiempo, tal como se muestra en la Figura 2.2. El vídeo digital es la representación de un vídeo muestreado en una escena en forma digital. Cada muestra espacio-temporal, elemento de imagen o píxel está representada como un número o conjunto de números que describe el brillo “luminancia” y el color de la muestra. Para obtener una imagen muestreada en 2D, una cámara enfoca una proyección en 2D de la escena de vídeo en un sensor, como un conjunto de dispositivos de carga acoplada conjunto CCD. En el caso del color captura de imagen, cada componente de color se filtra por separado y se proyecta en una matriz CCD.

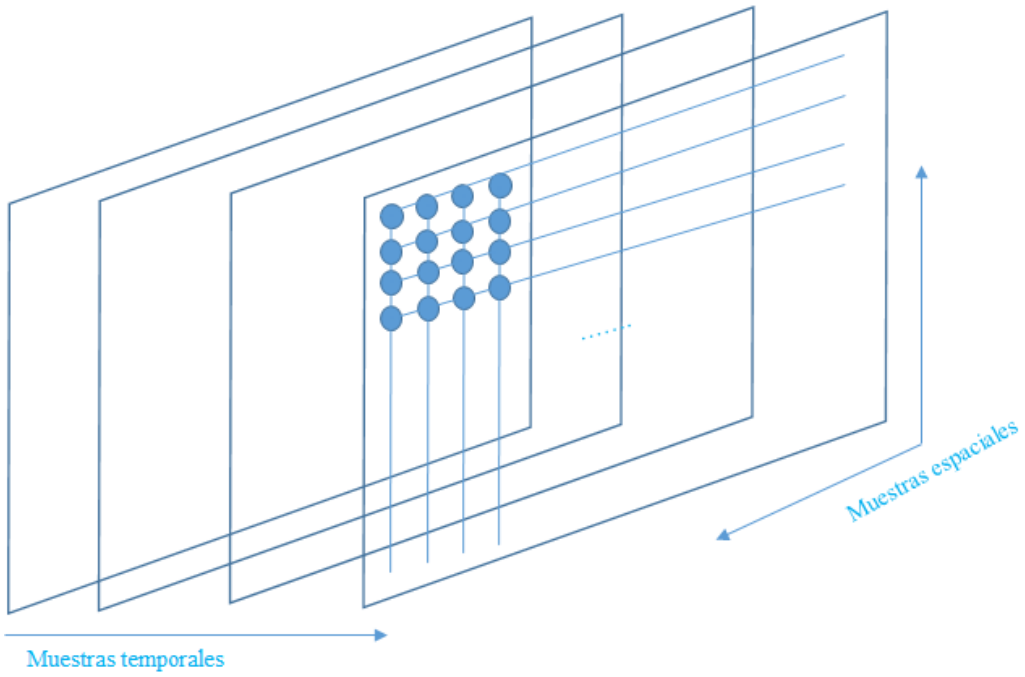


Figura 2.2: Muestreo espacial y temporal de una secuencia de vídeo

2.3.1. Muestreo Espacial

La salida de los sensores CCD es una señal de vídeo analógica, una señal eléctrica variable que representa una imagen de vídeo. El muestreo de la señal en un punto del tiempo produce una imagen o cuadro muestreado, que define valores en un conjunto de puntos de muestreo. El formato más común para una imagen muestreada, es un rectángulo con los puntos de muestreo ubicados en una cuadrícula cuadrada o rectangular. Un marco de tono continuo con dos cuadrículas de muestreo diferentes superpuestas sobre él. El muestreo ocurre en cada uno de los puntos de intersección en la cuadrícula y la imagen muestreada se reconstruirá representando cada muestra como un elemento de imagen cuadrado o píxel. La calidad de la imagen está influenciada por el número de puntos de muestreo. Elegir menos puntos en cuadrícula de muestreo, produce una imagen muestreada de baja resolución, mientras incrementa ligeramente los puntos de muestreo incrementa la resolución de la imagen muestreada.

2.3.2. Muestreo Temporal

Se captura una imagen de vídeo en movimiento tomando una instantánea rectangular de la señal en forma periódica por intervalos de tiempo. La reproducción de la serie de fotogramas produce la apariencia de movimiento. Una mayor frecuencia de muestreo temporal o frecuencia de fotogramas, proporciona un movimiento aparentemente más uniforme en la escena del vídeo,

pero requiere que se capturen y almacenen más muestras. Velocidad de fotogramas inferior a 10 fotogramas por segundo, se usan para comunicaciones de vídeo con una tasa de bits muy baja, debido a la cantidad de datos es relativamente pequeño, pero el movimiento es claramente desigual y poco natural a este ritmo. Entre 10 y 20 fotogramas por segundo, es más típico para comunicaciones de vídeo de baja velocidad de bits; puede verse un movimiento más suave pero desigual en las partes de la secuencia que se mueven rápidamente. Muestreo a 25 o 30 fotogramas completos por segundo, son estándar para las imágenes de televisión, con entrelazado para mejorar la apariencia del movimiento; 50 o 60 fotogramas por segundo produce una suavidad movimiento aparente a expensas de una velocidad de datos muy alta.

2.3.3. Marcos y Campos

En [Ric10] se indica que una señal de vídeo puede muestrearse como una serie de fotogramas completos o muestreo progresivo y como secuencia de campos entrelazados o muestreo entrelazado. En una secuencia de vídeo entrelazada, la mitad de los datos de un marco o campo se muestrean en cada intervalo de muestreo temporal. Un campo consiste en líneas impares o pares contenidas en un cuadro de vídeo completo y una serie de vídeo entrelazada contiene una serie de campos, que representa la mitad de la información en un cuadro de vídeo completo. La ventaja de este método de muestreo es que es posible enviar el doble de campos por segundo, el número de fotogramas en una secuencia progresiva equivalente con la misma velocidad de datos, dando la apariencia de movimiento más suave. Por ejemplo, una secuencia de vídeo (del inglés Phase Alternating Line (PAL)) consta de 50 campos por segundo y cuando se reproduce, el movimiento puede aparecer más suave que en un progresivo equivalente secuencia de vídeo que contiene 25 fotogramas por segundo.

2.3.4. Sistemas de Compresión del Vídeo

Los métodos de compresión de vídeo se fundamente en acortar y suprimir referencias iguales del vídeo, para que a través de la red el registro se pueda guardar en algún artefacto, conservando una calidad óptima. Existen dos maneras de comprimir un vídeo:

- **Compresión sin pérdidas** Se refiere a los métodos de compresión en los que la calidad de la señal decodificada es igual a la calidad de la señal de la fuente.
- **Compresión con pérdidas** En la práctica del manejo de imágenes y sonido, son relativamente pocas las aplicaciones en que es necesaria una reconstrucción absolutamente fiel de la información de la fuente, ya que intervienen aspectos perpetúales que son aprovechables, para reducir la cantidad de información que debe transmitirse y que admite, puede ser reconstruida por el decodificador. En los sistemas de compresión con pérdidas es admisible la pérdida de cierta cantidad de información que no es relevante para el

observador final. En estas condiciones, se pretende que la señal tenga la máxima calidad relativa, imperceptible por el público. La compresión con pérdidas no se puede deshacer, no es posible reponer a través de la información comprimida la información inicial. Al no poder deshacerse genera que la huella se degenere rápidamente bajando la calidad, si se enlaza la compresión, mejor dicho se generan muchas compresiones sobre huellas ya codificadas.

Los parámetros de una huella de vídeo son afectados por reiteraciones concurrentes con lo cual se genera la compresión y son:

- **Espacial** Sucede en el momento que los píxeles adyacentes tienen un grado de conexión, es decir, cuando son semejantes.
- **Temporal** Sucede cuando los píxeles en fotogramas contiguos de una huella también están conectados.
- **Psicovisual** Radica en que el ojo no trata toda la comunicación óptica con semejante sensibilidad.
  - **La luminancia** Abarca toda la comunicación encadenada sobre la claridad de la imagen y no presenta comunicación alguna sobre los colores. Representa, asimismo, la imagen en blanco y negro en todas sus gradaciones de grises intermedios.
  - **La crominancia** Abarca en su totalidad sobre el color de los objetos, dividida en los colores esenciales. Las huellas de diferencia de colores básicos rojo, azul y verde menos la claridad (R-Y), (B-Y) y (G-Y) respectivamente producen las huellas de crominancia. De este producto sólo se requieren dos, puesto que con la composición de las otras se genera la última.
- **Codificación** La totalidad de parámetros no suceden con la misma posibilidad en una imagen. La totalidad no precisa idéntico número de bits para codificarlos.

Actualmente, la gran parte de distribuidores de vídeo usan métodos tipo de compresión para vídeos, esto manifiesta la semejanza e interoperabilidad. Los tres métodos más usados son: Del comité de expertos (del inglés Joint Photographic Experts Group (JPEG)) para fotos y (del inglés Moving Picture Experts Group (MPEG)), y H264, para vídeo. Este es el método que se utiliza ahora para la compresión de vídeo. El procesamiento de compresión se basa en adaptar un algoritmo (H264 o MPEG-4) al vídeo único para crear un registro comprimido y dispuesto para ser transferido o almacenado. Los distintos métodos de compresión usan métodos diferentes para acortar los datos y los solución en que grado de frecuencia de bits y latencia son distintos.

En [Ló17] encuentran dos modelos de algoritmos de compresión:

- **Intrafotograma:** Es la que utiliza una procedimiento de codificación que consta en explotar la reiteración espacial que esta entre una imagen y otra en la que interviene un análisis frecuencial. Los datos se acortan a un fotograma de imagen con la conclusión de quitar la comunicación sobrante que puede ser invisible para el ojo humano. El JPEG, es un modelo de este tipo de método de compresión. En un orden JPEG, las imágenes se recopilan o comprimen como imágenes individuales JPEG.
- **Interfotograma:** Para disminuir los datos de vídeo entre la sucesión de fotogramas, la predicción interfotograma se usa por los algoritmos de compresión de vídeo como el MPEG-4 y el H264. Un fotograma se equipara con otro, sólo sí se compilan los píxeles distintos con respecto al primer fotograma, esto se llama método de recopilación distintiva, se acorta el número recopilados de píxeles y luego enviados. Cuando se compila de este modo, las imágenes se presentan como una serie de vídeo único al visualizarla.

En [Lim19] se muestra la Tabla 2.1 presenta la cronología de estándares y codificación de vídeos, entre los que destacan los comités (del inglés International Telecommunication Union (ITU)), (del inglés International Electrotechnical Commission (IEC)) e (del inglés International Organization for Standardization (ISO)).

Tabla 2.1: Sistemas de compresión cronológicamente

Año	Desarrollado	Formato
1988	ITU-T	H.261
1993	ISO-IEC	MPEG-1 part-2
1995	ITU-T	H.262
1995	ISO-IEC	MPEG-2 part-2
1996	ITU-T	H.263 / H.263+
1999	ISO-IEC	MPEG-4 part-2 (Visual)
2003	ITU-T	H.264 Advanced Video Coding
2003	ISO-IEC	MPEG-4 part-10 (Advanced video coding)
2004	Xiph	Theora
2006	Microsoft SMPTE	VC-1 or WMV-9
2011	Google	VP8
2013	Google	VP9
2013	ITU-T	H.265 / HEVC
2013	Xiph	Daala

En [Tek15] se muestra MPEG1 como una estructura de datos jerárquica que consta de las siguientes seis capas, que permite al decodificador interpretar los datos inequívocamente:

1. Las secuencias están formadas por varios grupos de imágenes.

2. Grupo de las imágenes (del inglés Group of Pictures (GOP)) están compuestas de imágenes.

3. Las imágenes consisten en rebanadas o slices. Hay cuatro tipos de imágenes que indican el respectivo modo de compresión: I-pictures, P-pictures, B-pictures y D-pictures.

   Las imágenes I son intratramas codificadas (del inglés Discrete Cosine Transform (DCT)) usando un algoritmo similar a JPEG. Ellos sirven como puntos de acceso aleatorio a la secuencia. Hay dos tipos de inter-estructura imágenes codificadas, imágenes P y B.

   En estas imágenes, el movimiento compensado los errores de predicción está codificado con DCT. Solo la predicción hacia adelante se usa en P-imágenes, que siempre están codificadas en relación con las imágenes I o P anteriores. La predicción de las imágenes B puede ser hacia adelante, hacia atrás o bidireccional en relación con otras imágenes I o P. Las imágenes D contienen solo el componente DC de cada bloque, y sirve para navegar a velocidades de bits muy bajas. El número de I, P y B en un GOP depende de la aplicación, por ejemplo, depende del tiempo de acceso y requisitos bitrate.

   La composición de un GOP se ilustra mediante el siguiente ejemplo:

   Un GOP se muestra en la Figura 2.3 que se compone de nueve imágenes. Nota que el primer fotograma de cada GOP es siempre una I-imagen. En MPEG el orden en que se procesan las imágenes no es necesariamente el mismo que su orden secuencial de tiempo. Las imágenes de la Figura 2.3 pueden codificarse en uno de los siguientes:

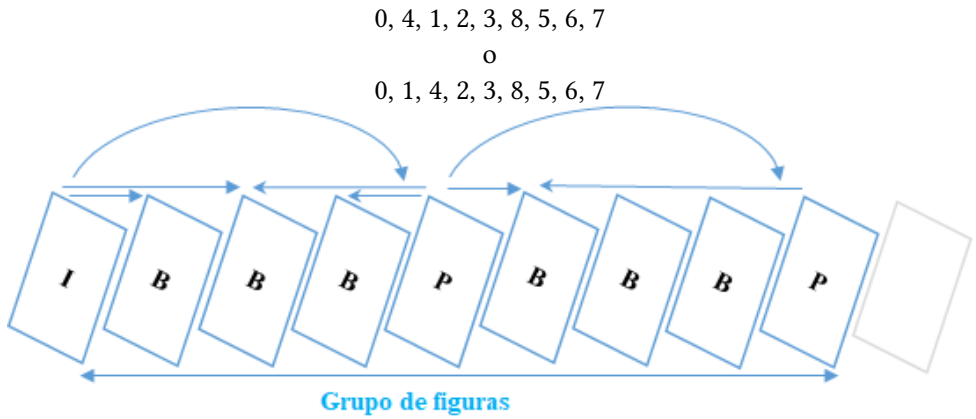


Figura 2.3: Grupo de figuras en MPEG1

   ya que la predicción para las imágenes P y B debería basarse en imágenes que ya están transmitidos.

4. Rebanadas o slices se componen de macrobloques que introducen principalmente por recuperación de error.

5. La composición de macrobloques (MEGABYTE) son los mismos que en la norma H.261

estándar. Algunos parámetros de compresión se pueden variar en base a MB. Los tipos de MB dependiendo de la elección de estos parámetros se enumeran en la Tabla 2.2.

6. Bloques son 8 X Matrices de 8 píxeles. Ellos son la unidad DCT más pequeña.

En [11193] se refiere a los encabezados se definen por secuencias, GOP, fotos, rebanadas y MBs que solo especifica los datos que siguen. Para extender el conocimiento sobre el estándar MPEG-1.

Tabla 2.2: Macrobloques tipo en MPEG1

I-Pictures	P-Pictures	B-Pictures
Intra	Intra	Intra
Intra-A	Intra-A	Intra-A
	Inter-D	Inter-F
	Inter-DA	Inter-FD
	Inter-F	Inter-FDA
	Inter-FD	Inter-B (Visual)
	Inter-FDA	Inter-BD
	Skipped	Inter-BDA
		Inter-I
		Inter-ID
		Inter-IDA
		Skinned

2.3.5. Sistemas de Compresión del Audio

En [Wika] se define como la reducción del tamaño de los archivos de audio como una forma de compresión de datos. Los códecs de audio son normalmente identificados como algoritmos de compresión de audio. En [Wik19] se muestran los principales formatos de compresión de audio que existen actualmente y son: AAC, Codec de audio libre sin perdidas (del inglés Free Lossless Audio Codec (FLAC)), Monkey's Audio, MP3, Vorbis, WavPack, codec de audio de windows (del inglés Windows Media Audio (WMA)) y se describen en la Tabla 2.3.

Tabla 2.3: Sistema de compresión de audio

Codec	AAC	FLAC	Monkey's Audio	MP3	Vorbis (Ogg)	WavPack	WMA
Creador	ISO/IEC MPEG	Xiph.org Ancho Committe	Matthew T. Foundation, Josh Coalson	ISO/IEC MPEG Audio Ashland	(Xiph.Org Committe	Conifer Software Foundation)	Microsft
Fecha lanzamiento público	1997	20/07/2001	2000	1993	11/05/2000	1998	1999
Última versión estable	ISO/IEC 14496-3	1.4.2	4.33	ISO/IEC 11172-3, ISO/IEC 13818-3	(1.4.6)	5.1.0	11.0
Tipo de compresión	Con pérdidas,	Sin pérdidas Híbrido	Sin pérdidas	Con pérdidas	Con pérdidas	Con pérdidas, Sin	Con pérdidas, Sin pérdidas pérdidas, Híbrido
Frecuencia de muestreo	8 kHz a 192 kHz	1 Khz a 1.04857 Mhz	8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48 kHz	8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48 kHz	1 Hz a 200 kHz	1 Hz a 13.777216 MHz	8, 11.025, 12, 16, 22.05, 32, 44.1, 48, 96 kHz
Tasa de bits	8-529 Kbps (estéreo)	Variable	Variable	8, 16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320 Kbps	Variable	Variable (modo sin pérdidas) / superior 196 kbps (modo con pérdidas)	4-768 kbps / variable (sin pérdidas)
Bits por muestra	Cualquiera	4, 8, 16, 24, 32	?	Cualquiera	Cualquiera	Variable (modo sin pérdidas) / superior a 2.2 (modo con pérdidas)	16, 24 (modo sin pérdidas) / Cualquiera (modo con pérdidas)
CBR	Si	No	No	Si	Si	Si	Si
VBR	Si	Si	Si	Si	Si	Si	Si
Multicanal	hasta 28 canales	hasta 8 canales	No	No	hasta 255 canales	hasta 16 canales	hasta 8 canales (WMA Professional) / hasta 6 canales (WMA sin pérdidas)

El estándar MPEG-2 es ampliamente utilizado, ya que es el vehículo para el disco versátil digital (del inglés Digital Versatile Disc (DVD)) y también tiene un componente de audio el estándar MPEG-2 (del inglés AAC) que estaba dirigido a la reproducción de sonido transparente para teatros. Puede reproducir a 320 kbps para cinco canales, de modo que el sonido se pueda reproducir desde cinco direcciones: izquierda, derecha, centro, semáforo, y derecho-envolvente. Los llamados sistemas de 5.1 canales también incluyen una frecuencia canal de mejora (del inglés Low Frequency Effects (LFE)) (un "woofer"). Por otro lado, MPEG-2 AAC también es capaz de ofrecer sonido estéreo de alta calidad a velocidades por debajo de 128 kbps. La tecnología de codificación de audio para el formato (del inglés Digital Versatile Disc Audio grabable (DVD-AR)) y también es adoptada por XlvI Radio, uno de los dos servicios de radio por satélite en América del Norte. Tal como se muestra en la Tabla 2.4.

Tabla 2.4: MP3 Rendimiento de compresión

Calidad de sonido	Ancho de banda	Modo	Índice de compresión
Telefonía	3.1 kHz	Mono	96:1
Mejor que onda corta	4.5 kHz	Mono	4 8:1
Mejor que radio AM	7.5 kHz	Mono	24:1
Similar a radio FM	11 kHz	Estereo	26:1 a 24:1
Near-CD	15 kHz	Estereo	16:1
CD	>15 kHz	Estereo	14:1 a 12:1

En [LDL14] se muestra MPEG-4 audio integra diferentes componentes de audio dentro de un estándar, compresión, perceptualmente basado en codificadores, texto a voz y Musical Instrument Digital Interface (MIDI). El estándar MPEG-4 AAC, es similar al estándar MPEG-2 AAC, con algún cambio menor.

2.4. Formato de Contenedor Multimedia

El formato de contenedor multimedia o también llamado archivo contenedor, almacena información de vídeo, audio y metadatos. Además, se puede almacenar información específica del fabricante de un dispositivo, p. ej. subtítulos. Asimismo, en algunos casos solo contiene información de vídeo comprimido y metadatos, p. ej. cuando se captura el vídeo con un dispositivo que tiene deshabilitado el micrófono. En la Figura 2.4 se muestra el contenido del contenedor multimedia.

Para integrar toda la información de vídeo y audio, el dispositivo realiza un proceso llamado encapsulamiento y el resultado es la extensión del fichero multimedia. La mayoría de estos vídeos se pueden reproducir en plataformas o sistemas operativo de terceros. Por ejemplo, MOV se genera y reproduce en dispositivos de marca Apple con sistema operativo iOS, pero también se

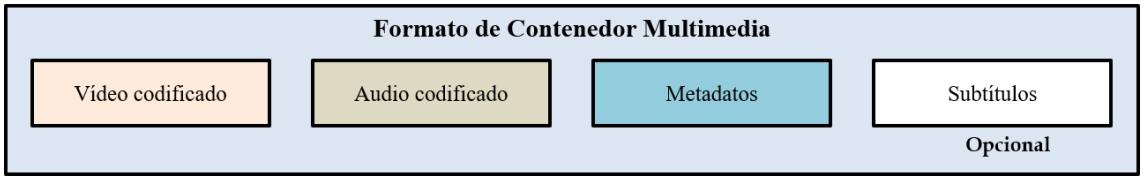


Figura 2.4: Contenido del formato de contenedor multimedia

reproduce en dispositivos con sistema operativo Android, Windows, Linux, etc. No obstante, el formato Audio Video Interleaved (AVI) solo se crea y reproduce en dispositivos específicos que en su mayoría son cámaras digitales convencionales u ordenadores con sistema operativo Windows.

En la Tabla 2.5, se describe los principales contenedores multimedia de vídeos.

Tabla 2.5: Principales contenedores multimedia de vídeos

Contenedor	Descripción
AVI	Introducido por Microsoft en noviembre de 1992, es un formato de contenedor multimedia como parte de su vídeo para la tecnología Windows [Qui12]. El formato AVI tiene muchas características pero raramente se utilizan además soporta múltiples flujos de audio y vídeo. Los formatos de audio y vídeo AC3, DivX, MP3 y Xvid son soportados por el contenedor AVI, admite varios flujos de datos de audio, lo que en la práctica significa que puede contener varias bandas sonoras en distintos idiomas.
FLV	Anteriormente Macromedia Flash, es un formato contenedor de Adobe Flash, usado para divulgar a través de Internet los contenidos de vídeo [JP09]. Flash video (FLV) reduce el tamaño de sus archivos hasta un 60 % más que otros formatos, las velocidades de descarga son altas y favorecen la reducción de la capacidad de almacenamiento de los servidores. Los bits de vídeo de este tipo de archivos son una variante propia del estándar H.263. Tiene como desventaja que requiere de una aplicación propia para la reproducción de los archivos FLV como es Adobe Flash Player.
MXF	Material Exchange format (MXF) es un formato de fichero versátil que guarda datos con cualquier formato de compresión con los metadatos, guarda ficheros de streaming que se visualicen durante la transferencia, contiene un listado de ficheros y guarda la información sincronizada. Es un formato contenedor que facilita la interoperabilidad de contenidos entre distintas aplicaciones utilizadas en la cadena de producción de televisión [Rom13].
MKV	Matroska se asocia con la extensión *.Matroska Video (MKV), que contiene vídeo con subtítulos y audio, pero también se emplean las extensiones *.mka para archivos que sólo contienen audio, y *.mks para los que contienen exclusivamente subtítulos. Es un formato de contenedor estándar de código abierto que permite contener un conjunto ilimitado de vídeo, audio, imagen o pistas de subtítulos en un único archivo. Se creó con la intención de ser un formato universal de almacenamiento de contenidos multimedia comunes, pero siendo totalmente abierto. [RDS10].
MP4	Es un formato contenedor desarrollado por MPEG y está definido como MPEG-4 parte 14. Integra dentro de su formato tanto audio y vídeo comprimido. Por otro lado se incluye información adicional como metadatos, que es información sobre el audio y vídeo del vídeo contenido.
3GP	Este formato de archivo es una versión simplificada del contenedor MP4, almacena el vídeo como MPEG-4 o H.263 y el audio se almacena en los formatos AMR-NB o AAC-LC. Es un contenedor para almacenar información de audio y vídeo usado por teléfonos móviles.
MOV	Para comprimir usa el códec MPEG-4 y para poder almacenar películas y demás archivos de vídeo utiliza varias pistas. Fue desarrollado por Apple Inc. en 1998.

2.5. Formato de Contenedor Multimedia MOV

Es un formato de contenedor multimedia multiplataforma que fue creado por Apple en 1998 y está definido en el estándar QuickTime File Format (QTFF) [Inc16]. Este formato es ideal para el intercambio de medios digitales entre dispositivos, aplicaciones y sistemas operativos, ya que se puede utilizar para describir casi cualquier estructura de medios. Todos los dispositivos de la marca Apple hacen uso de este formato y sus versiones actuales permiten interactuar con películas en 3D y realidad virtual. También, QTFF es compatible con el codificador H.264 y se ha utilizado como base del estándar MPEG-4 y JPEG-2000, desarrollado por ISO. Aunque estos tipos de archivos tienen estructuras similares y contienen muchos elementos funcionalmente idénticos, son tipos de archivos distintos.

El formato contenedor multimedia MOV almacena determinados tipo de datos, ya sea audio, vídeo, efectos y texto y los guarda en una o más pistas. Cada pista tiene codificado digitalmente un determinado códec o una mención a datos almacenado en otro fichero a esto denominamos flujo de información. Las pistas contienen objetos denominados átomos ubicados en una estructura jerárquica de datos. Un átomo puede ser el padre o contener otros átomos con su respectiva información, pero no puede cumplir las dos funciones.

Las películas QuickTime se almacenan en el disco, utilizando dos estructuras básicas para almacenar información: átomos (átomos simples o átomos clásicos) y átomos QT. Un archivo QuickTime almacena la descripción de sus medios por separado de los datos multimedia.

2.5.1. Átomos

Un átomo es la unidad básica de datos del formato de contenedor MOV. Cada átomo contiene campos de tamaño y tipo que preceden a cualquier otro dato. El campo de tamaño indica el número total de bytes en el átomo, incluidos los campos de tamaño y tipo. El campo de tipo especifica el tipo de datos almacenados en el átomo y, por implicación, el formato de esos datos. En algunos casos, los campos de tamaño y tipo son seguidos por un campo de versión y un campo de indicadores.

Los tipos de átomo se especifican mediante un entero sin signo de 32 bits, normalmente interpretado como un código estándar (del inglés American Standard Code for Information Interchange (ASCII)) de cuatro caracteres. Además, se reserva todos los códigos de cuatro caracteres que consisten enteramente en letras minúsculas. Todos los datos de una película de QuickTime se almacenan en el orden de bytes big-endian, también conocido como ordenamiento de bytes de red, en el que los bytes más significativos se almacenan y transmiten primero salvo que se indique que no.

Un átomo hoja es el que no contiene otros átomos, y normalmente tiene datos con uno o más

campos o tablas. Algunos átomos de hoja actúan como banderas o marcadores de posición, sin embargo, y no contienen datos más allá de su tamaño y campos tipo. En la Figura 2.5 se muestra el diseño y organización de los átomos en un vídeo con formato MOV. Cada átomo lleva su propio tamaño y tipo de información, así como sus datos. Un átomo de hoja simplemente contiene una serie de campos de datos accesibles por desplazamientos. Los átomos no tienen que estar en un orden particular, a menos que sea específicamente mencionado en l especificación.

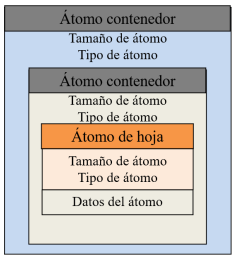


Figura 2.5: Diseño y organización de un átomo

2.5.2. Estructura del Átomo

Los átomos se componen de un encabezado, seguido por los datos del átomo. El encabezado contiene los campos del tamaño en bytes y del tipo del átomo. Puede tener un campo de tamaño extendido, que da el tamaño de un átomo grande compuesto por un entero de 64 bits. Si un campo de tamaño extendido está presente se establece en 1. El tamaño real de un átomo no puede ser menor a 8 bytes los que corresponde al tamaño y al tipo. Algunos átomos también contienen campos de versión y flag y estos no se tratan como parte del encabezado de átomo; se tratan como campos de datos específicos para cada tipo de átomo que los contiene. Tales campos siempre destan a cero, salvo que se diga lo contrario. Un encabezado de átomo consta de los siguientes campos:

1. **Tamaño del átomo:** Está representado por un entero de 32 bits que indica el tamaño del átomo, en el que se incluye el encabezado y el contenido del átomo, incluidos los átomos contenidos. Normalmente, el campo de size contiene el tamaño real del átomo, en bytes, expresado como un entero sin signo de 32 bits. Sin embargo, el campo de size puede contener valores especiales que indican un método alternativo para determinar el tamaño del átomo. Estos valores especiales se utilizan normalmente sólo para datos de medios (‘mdat’). Dos valores especiales son válidos para el campo de size:
   - **0**, que sólo está permitido para un átomo de nivel superior, designa el último átomo del archivo e indica que el átomo se extiende hasta el final del archivo.
   - **1**, lo que significa que el tamaño real se da en el campo de extended size extendido, un campo opcional de 64 bits que sigue el campo de type.

Esto organiza átomos de datos de medios que contienen más de 2^32 bytes.

2. **Tipo:** Está representado por un entero de 32 bits que contiene el tipo del átomo. Usualmente es tratado como un campo de cuatro caracteres con un valor nemotécnico, como 'moov' (0x6D6F6F76) para un átomo de película o 'trak' (0x7472616B) para un átomo de pista, pero valores no ASCII (como 0x00000001) también se usan.

   Si se encuentra un átomo con un tipo desconocido, no hay que intentar interpretar los datos del átomo. Utilizar el campo de size del átomo para omitir este átomo y todo su contenido. Esto permite compatibilidad hacia adelante con las extensiones del formato de archivo QuickTime.

3. **Tamaño extendido:** Si el campo de size de un átomo se establece en 1, el campo de type es seguido por un campo de extended size extendido de 64 bits, que contiene el tamaño real del átomo como un entero sin signo de 64 bits. Se utiliza cuando el tamaño de un átomo de datos de medios supera 2^32 bytes.

   Cuando el campo de size contiene el tamaño real del átomo, el campo de size extendido no está presente. Esto significa que cuando se modifica un átomo de QuickTime mediante la adición de datos y su tamaño cruza el límite de 2^32 bytes, no hay size extendido en el que se registre el nuevo tamaño de átomo. En consecuencia, no siempre es posible agrandar un átomo más allá de 2^32 bytes sin copiar su contenido a un nuevo átomo.

   Para evitar este inconveniente, los átomos de datos de medios se crean típicamente con un átomo de espacio reservado de 64 bits que los precede inmediatamente en el archivo de película. El átomo de espacio reservado tiene un tipo de kWideAtomPlaceholderType (wide).

   Al igual que un átomo 'free' o 'skip', el átomo 'wide' es espacio reservado, pero en este caso el espacio está reservado para un propósito específico. Si un átomo 'wide' precede inmediatamente a un segundo átomo, el segundo átomo puede extenderse desde un tamaño de 32 bits a un tamaño de 64 bits simplemente iniciando el encabezado de átomo 8 bytes antes (sobrescribiendo el átomo 'wide'), size campo a 1 y agregar un campo de size extendido. De esta manera no es necesario volver a calcular los desplazamientos para los datos de muestra.

   El átomo 'wide' tiene exactamente 8 bytes de tamaño y consiste únicamente en su size y campos de type. No contiene ningún otro dato. Como se ve en la Figura 2.6.

La Tabla 2.6 muestra los tipos de átomos esenciales en un archivo de película QuickTime dentro del cual se almacenan otros átomos. Además, el archivo puede contener átomos de espacio libre, átomos de vista previa y otros átomos no enumerados en esta especificación de formato de archivo. En la Figura 2.7 se ilustra la estructura de los átomos en el contenedor multimedia MOV.

Asimismo, para un mejor detalle, en el Capítulo A se presenta la descripción de los principales átomos del contenedor multimedia MOV.

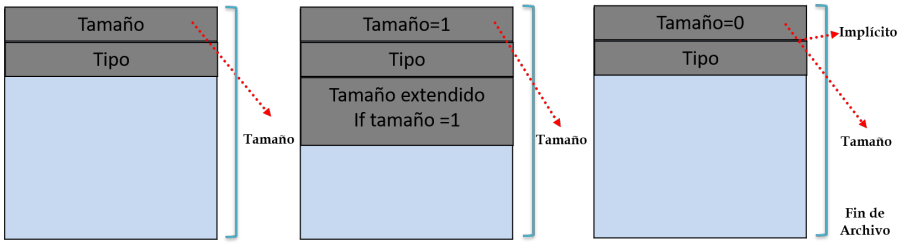


Figura 2.6: Átomo wide

Tabla 2.6: Tipos de átomos básicos de un archivo QuickTime

Tipo de átomo	Utiliza
'ftyp'	Compatibilidad de tipo de archivo identifica el tipo de archivo y lo diferencia de tipos de archivos similares, como archivos MPEG-4 y archivos JPEG-2000.
'moov'	Metadatos del recurso de película sobre la película (número y tipo de pistas, ubicación de los datos de muestra, etc.). Describe dónde se pueden encontrar los datos de la película y cómo interpretarlos.
'mdat'	Ejemplos de películas de muestras de medios de datos, como fotogramas de vídeo y grupos de muestras de audio. Normalmente, estos datos sólo se pueden interpretar utilizando el recurso de película.
'free'	Espacio no utilizado disponible en el archivo.
'skyp'	Espacio no utilizado disponible en el archivo.
'wide'	El espacio reservado puede ser sobrescrito por un campo de tamaño extendido si el siguiente átomo excede 2^32 bytes, sin desplazar el contenido del siguiente átomo.
'pnot'	Referencia a los datos de vista previa de la película.

En la Figura 2.8 se muestra gráficamente el contenido del átomo MOOV.

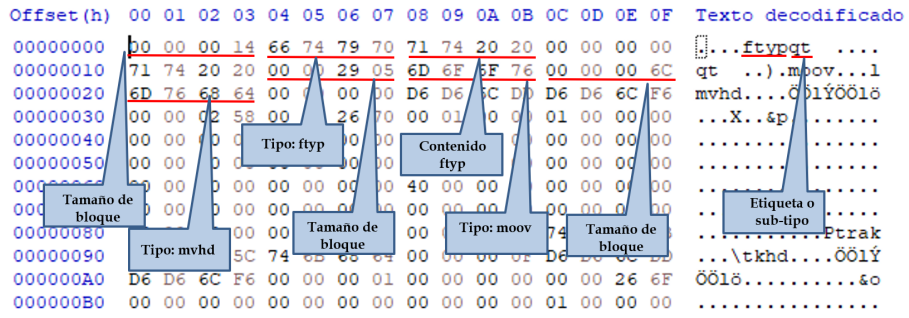


Figura 2.7: Estructura de los átomos en el contenedor multimedia MOV

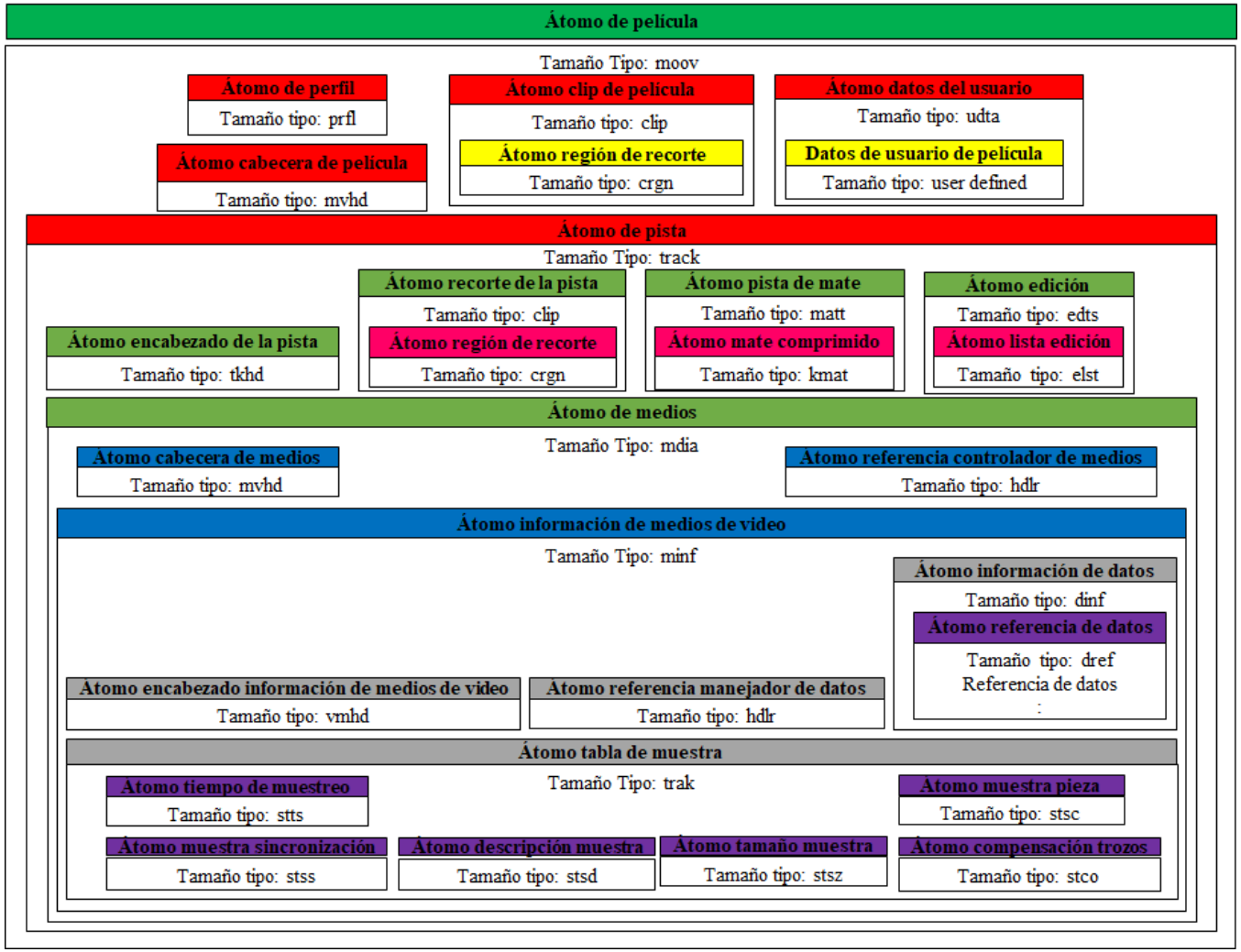


Figura 2.8: Contenido de átomo MOOV

2.5.3. Átomos QT y Contenedores de Átomos

Los átomos QT son una estructura de datos mejorada que proporciona un formato de almacenamiento de propósito más general y elimina algunas de las ambigüedades que surgen al usar átomos simples. Un átomo QT tiene un encabezado expandido; Los campos tamaño y tipo son seguidos por campos para un ID de átomo y un recuento de átomos hijos. Esto permite especificar múltiples átomos hijos del mismo tipo mediante números de identificación. También hace posible analizar el contenido de un átomo QT de tipo desconocido, caminando el árbol de sus átomos hijos.

Los átomos QT se envuelven normalmente en un contenedor de átomos, una estructura de datos con una cabecera que contiene un recuento de bloqueo. Cada contenedor de átomos contiene exactamente un átomo de raíz, que es el átomo QT. Los átomos contenedores no son átomos y no se encuentran en la jerarquía de átomos que forman un archivo de película QuickTime. Sin embargo, se pueden encontrar contenedores de átomos como estructuras de datos dentro de algunos átomos. Los ejemplos incluyen mapas de entrada de medios y átomos de propiedad de medios.

La Figura 2.9 representa la disposición de un átomo QT. Cada átomo QT comienza con un encabezado de contenedor de átomo QT, seguido por el átomo de raíz. El tipo de átomo de la raíz es el tipo del átomo QT. El átomo de la raíz contiene cualquier otro átomo que forme parte de la estructura. Cada átomo de contenedor comienza con un encabezado de átomo QT seguido por el contenido del átomo. Los contenidos son átomos hijos o datos, pero nunca ambos. Si un átomo contiene hijos, también contiene todos los datos y descendientes de sus hijos. El átomo de la raíz está siempre presente y nunca tiene hermanos.

Un encabezado de contenedor de átomo QT contiene dos campos: a) “reservado” que es un elemento de 10 bytes que se debe establecer en 0; b) “número de bloqueos” que es un entero de 16 bits que debe establecerse en 0. Cada encabezado de átomo QT contiene los siguientes datos:

1. **Tamaño:** Un entero de 32 bits que indica el tamaño del átomo en bytes, incluye el encabezado del átomo QT y el contenido del átomo. Si el átomo es un átomo de hoja, entonces este campo contiene el tamaño del átomo individual. El tamaño de los átomos contenedores incluye todos los átomos contenidos. Puede recorrer el árbol de átomos utilizando los campos de tamaño y sus subátomos.

2. **Tipo:** Un entero de 32 bits que contiene el tipo del átomo. Si se trata del átomo raíz, el valor de tipo se establece en ‘sean’.

3. **ID de Átomo:** Un entero de 32 bits que contiene el valor de ID del átomo. Este valor debe ser único entre sus hermanos. El átomo de la raíz siempre tiene un valor ID de átomo de 1.

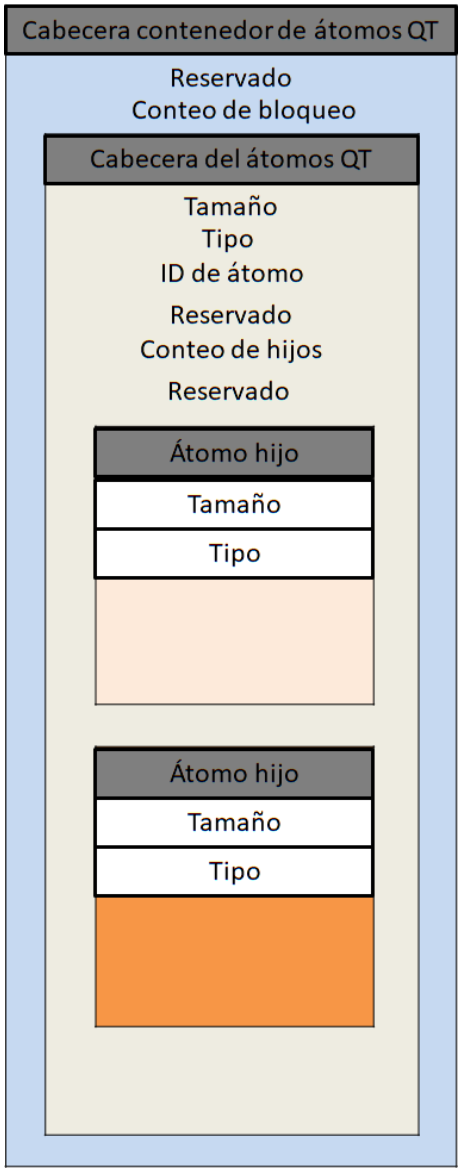


Figura 2.9: Contenedor de átomos

4. **Reservado:** Un entero de 16 bits que debe establecerse en 0.

5. **Número de subátomos:** Un entero de 16 bits que especifica el número de átomos internos que contiene un átomo. Este conteo incluye sólo subátomos inmediatos. Si este campo se establece en 0, el átomo es un átomo de hoja y sólo contiene datos.

6. **Reservado:** Un entero de 32 bits que se debe establecer en 0.

Por otro lado, un contenedor de átomos no es lo mismo que un átomo contenedor. Un contenedor de átomos es un contenedor, no un átomo. Tiene una estructura básica para almacenar información en QuickTime y posee una jerarquía estructurada en árbol de átomos QT.

Capítulo 3

Análisis Forense de Vídeos

En este capítulo se muestran las principales técnicas forenses para vídeos digitales. A pesar de que este campo es bastante investigado existen pocas investigaciones que utilicen el contenedor multimedia como elemento para el análisis forense de vídeos digitales. En la Sección 3.1 se presentan las técnicas de autenticación del contenido. En la Sección 3.2 se estudia las técnicas de detección de manipulaciones de vídeos digitales. En la Sección 3.3 se presentan las técnicas que utilizan información del contenedor como parte de la investigación forense de vídeos digitales. Finalmente, en la Sección 3.4 se presentan las herramientas de extracción de información del contenedor multimedia más completas existentes en la literatura.

3.1. Técnicas de Autenticación del Contenido de un Vídeo

En [MD04] la información del formato de archivo y los metadatos son otra fuente de evidencia forense, pero han recibido menos atención. Las obras existentes se centran en imágenes digitales. En [10994] se realiza el análisis de las propiedades de los metadatos JPEG y en [EA02] se realiza el estudio de los archivos de imagen intercambiable (del inglés Exchangeable Image File Format (EXIF)) que han ganado interés. Asimismo, en [Far08] se estudian las tablas de cuantificación y los algoritmos de procesamiento de la imagen que sirven también como características para la identificación del artefacto fuente de una imagen cuestionada. Las imágenes en miniatura es otro grupo de características forenses relevantes.

En [PM09] [Lew12] se basas en una descripción de configuración de prueba y observaciones generales en el formato del contenedor del vídeo. Se demuestra que las peculiaridades del formato del contenedor MOV y del contenedor relacionado con formatos MP4, pueden arrojar información importante sobre la procedencia y el historial de procesamiento de vídeos digitales.

En [KJF11] se combinan diferentes propiedades de la imagen y su imagen resumen (parámetros de compresión, sus dimensiones, el número de entradas EXIF, firmas del modelo de

la cámara y configuraciones de software de procesamiento) para identificar las configuraciones de origen de una imagen digital. Las imágenes se marcan como sospechosas de haber sido manipuladas si no se encuentra una coincidencia.

En [Glo12] se describen las peculiaridades en el orden de las estructuras JPEG y EXIF es particularmente valioso y brinda información distintiva para la autenticación de imágenes digitales. Tales características son de bajo nivel, por lo tanto, ofrecen mayor fiabilidad, que algunos degradados comunes que indican el formato de archivo y la información de metadatos en sí mismo.

En [FCK13] se explotan las características de ruido para determinar si el contenido de la imagen y los datos EXIF son consistentes. En [SM08] las imágenes se marcan como sospechosas si no se encuentra una coincidencia. Sin embargo, para alterar los parámetros de compresión o las entradas EXIF solo se necesita usar las herramientas de software adecuadas.

En [GFK14] mientras la mayoría de los consumidores en sus dispositivos carecen de soporte de autenticación práctica en absoluto, el aumento de ataques contra sistemas de autenticación de cámaras profesionales han demostrado debilidades de las soluciones existentes en el dispositivo. Técnicas forenses para inferir la procedencia y el historial de procesamiento de los archivos de medios ex-post ha ganado más interés entre investigadores y profesionales. En [SM14] el análisis forenses de imágenes ha sido el principal impulsor del campo, pero también en [MFB+12] el análisis de los archivos de vídeo recientemente se han puesto a la vanguardia.

3.2. Técnicas de Detección de Manipulaciones en un Vídeo Digital

En [SA17] existen diferentes técnicas de clasificación de detección de manipulaciones. Las clasificaciones más utilizadas en la literatura son las siguientes:

- Según el objetivo de la manipulación se divide en:
  - **Técnicas intra-fotogramas**: El análisis se realiza considerando un fotograma a la vez.
  - **Técnicas inter-fotogramas**: Se analizan las relaciones entre los fotogramas adyacentes para detectar la falsificación.
- Según el nivel de análisis realizado se divide en:
  - **Técnicas de detección de manipulaciones**: Se detecta la presencia de manipulación en el vídeo, pero no su ubicación exacta, ni su localización espacial o su localización temporal.
  - **Técnicas de localización de manipulaciones**: Además de detectar que hay manipulación en un vídeo, también la localiza.

- Según el nivel de intrusión se divide en:
  - **Técnicas activas**: Utilizan un rastro de identificación conocido, como una firma o marca de agua que está incrustado en el contenido en el tiempo de grabación o se envía con el receptor.
  - **Técnicas pasivas**: Solo utilizan el contenido recibido para determinar su autenticidad sin la ayuda de ningún otro tipo de información secundaria. La Figura 3.1 muestra las principales técnicas pasivas.

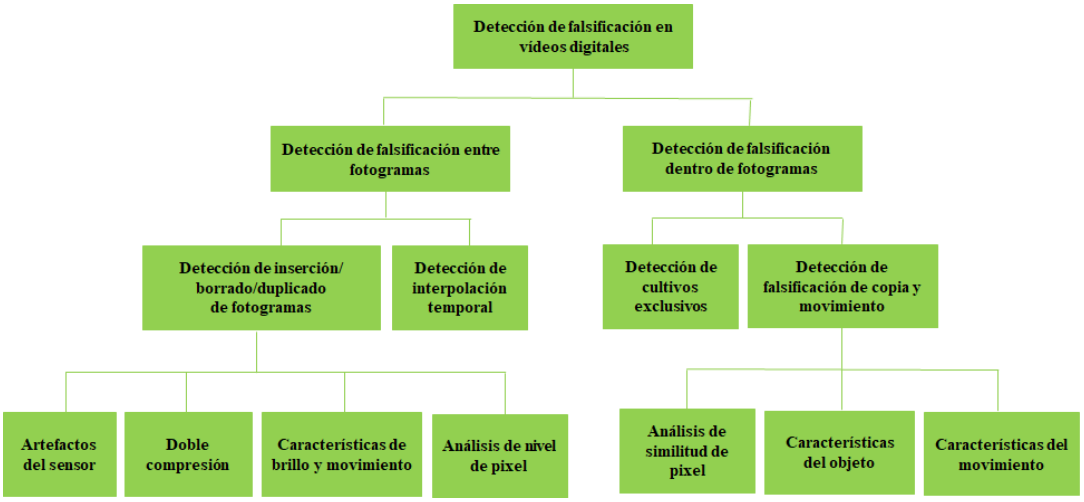


Figura 3.1: Técnicas de detección de manipulaciones en vídeos digitales

Las técnicas de detección de manipulación inter-fotogramas más comunes se basan en:

- **Detección de interpolación temporal**: En [Yan15] [YXL+16] falsifican vídeos mediante el corte temporal, es decir, interpolando dos o más vídeos diferentes para generar un nuevo vídeo. Si los vídeos originales no tienen la misma tasa de fotogramas, tienen que ser interpolados temporalmente antes de que pueda realizarse el empalme. Esto se hace con la conversión ascendente de frecuencia de fotogramas (del inglés Frame Rate Up Conversion (FRUC)), en donde se generan nuevos fotogramas con la ayuda de la salida de otro y se insertan en el vídeo original, aumentando la velocidad de los fotogramas.
- **Detección de inserción, borrado o duplicado de fotogramas**: Estas técnicas se dividen a su vez en:

  1. *Artefactos del sensor*: En [KOS10] [GCCF16] se indican las técnicas que se basan en la premisa de que los dispositivos de grabación dejan rastros específicos, es decir una huella, que sirven para determinar si todo el vídeo fue realizado usando la misma cámara .
  2. *Doble compresión*: En [SX10] [SNZ11] se establecen las técnicas que se basan en que un vídeo comprimido dos veces puede resultar de una manipulación, puesto que, para

editarlo, primero hay que descomprimir, ir a la secuencia de fotogramas, para después volver a comprimir .

3. *Características del brillo y del movimiento*: En [SA15] [AB16] se logra con la ayuda de los artefactos de borde compensados por movimiento (del inglés Motion Compensated Edge Artifacts (MCEA)), el cálculo no requiere información adicional con respecto al vídeo original .
4. *Análisis a nivel de píxel*: En [LCC11] [WLZM14] se analiza de manera fina la similitud, entre el clip de consulta dado y los candidatos a clip de fecha y así se detectan los fotogramas duplicados y se localizan sus posiciones.

Las técnicas de detección de manipulación intra-fotogramas se subdividen en:

- **Detección de cultivos exclusivos**: En [HRLL13] se indica que consiste en agrandar los fotogramas de un vídeo y luego recortarlos para eliminar la evidencia de algún evento incriminatorio en la parte más externa de los fotogramas. La presencia de tales falsificaciones se puede detectar buscando rastros de remuestreo, ya que cada vez que se recortan y agrandan los fotogramas de un vídeo, se someten a un proceso llamado muestreo para mantener una resolución consistente en la totalidad de fotogramas de dicho vídeo.
- **Detección de la clonación de regiones de un vídeo**: Estas técnicas buscan regiones duplicadas en un fotograma. La mayoría de estas técnicas se basan en:

  1. *Análisis de similitud de píxel*: En [BRR11] [NAF+13] se buscan similitudes o correlaciones entre regiones de fotogramas de vídeo sucesivos o regiones del mismo fotogramas, que en teoría no deberían tener nada en común, principalmente porque tienen orígenes diferentes o tienen diferentes tiempos/lugares asociados con su linaje.
  2. *Características del objeto*: En [COF12] [RGN14] se centran en detectar los artefactos que surgen después de que un objeto es eliminado de un cuadro de vídeo.
  3. *Características del movimiento*: En [BG15] [CTLH16] se utiliza extractores de características, originalmente utilizados en esteganálisis en imágenes, para extraer las características forenses de información de residuos de movimiento.

Existen otros tipos de manipulaciones orientadas a la alteración de las características o atributos del vídeo.

- **Manipulación de la tasa de bits de vídeos**, en [BLH14a] [BLH14b] se refiere al proceso de aumentar fraudulentamente las tasas de bits de vídeos. La tasa de bits es un criterio importante para juzgar la calidad de un vídeo digital; altas tasas de bits suelen ser similares con una mejor calidad audiovisual. Al aumentar deliberadamente la velocidad de bits de un vídeo originalmente de baja tasa de bits, se puede hacer que los vídeos de baja calidad aparezcan como de alta calidad y, por lo tanto, pueden ser utilizados por individuos

explotadores para obtener mayores ganancias comerciales. Otra razón para aumentar la velocidad de bits de un vídeo podría ser para ocultar la evidencia de una falsificación.

- **Detección de vídeo filogenia**, en [DRUC+10] [DRG11] se indica que si dos vídeos tienen el mismo contenido, pero son diferentes en términos de atributos, como tamaño, resolución y color, se denominan 'duplicados cercanos' entre sí. Dado un conjunto de vídeos casi duplicados, uno podría estar interesado en descubrir la razón detrás de generar un vídeo de otro y comprender la asociación causal entre estos vídeos.
- **Detección de recaptura de vídeo**, en [VSD13] [BVT+13] se refiere a la readquisición o recaptura se refiere a la actividad de capturar vídeos que se reproducen en monitores de pantalla o se proyectan en pantallas, el desafío de la detección de recaptura de vídeo es necesario en el campo del análisis forense digital, ya que la recaptura a menudo indica la existencia de alguna actividad previa de manipulación.

3.3. Uso del Contenedor Multimedia en el Análisis Forense de Vídeos Digitales

Los estándares de los formatos de archivo de vídeo definen solo un número limitado de características obligatorias y dejan espacio a la interpretación. Las decisiones de diseño de los fabricantes de dispositivos y vendedores de software son un recurso fructífero para la autenticación de vídeo forense. El análisis forense de vídeos sigue desarrollando nuevas tecnologías para verificar la autenticidad y la integridad de los vídeos digitales. La mayoría de las técnicas presentes se fundamentan en el análisis del flujo de datos de vídeo.

Recientemente se ha introducido una nueva línea de investigación para investigar el período de vida del vídeo fundamentado en el estudio del contenedor de vídeo. Sin embargo, las búsquedas en este terreno se fundamentan en la comparación manual de la estructura del contenedor de vídeo y de su contenido. Este análisis manual requiere tiempo y es propenso a errores. Los trabajos más relevantes en esta línea son:

En [GK14] se analiza el formato AVI y las reproducciones de vídeo similares a MP4 de teléfonos móviles y cámaras digitales. En el trabajo se extraen todas las estructuras del formato de los vídeos de 19 modelos de cámaras digitales, 14 modelos de teléfonos móviles y 6 herramientas de edición de vídeo. Para preparar la prueba se utilizó un software de edición para cortar secuencias cortas de las transmisiones de vídeo grabadas con una duración 10 segundos. Se comparan las estructuras de los contenedores tipo AVI y MP4 observando diferencias tanto en el orden como en presencia de datos individuales. Los archivos AVI a menudo contienen listas INFO específicas. Los archivos tipo MP4 pueden emplear tipos de átomos no estándar y diferentes parámetros de entradas atómicas. Se detectan un conjunto de peculiaridades específicas por

modelo y software, además proporcionan pistas valiosas para verificar la autenticidad de las transmisiones de vídeo digital. En las herramientas de edición de vídeo, se observa que cada software tiene su propia firma única, que enfatiza la utilidad de tales características en escenarios de autenticación. Los principales hallazgos en vídeos de cámaras digitales y teléfonos móviles a menudo emplean diferentes formatos de contenedor y compresión códecs. Se reportan diferencias considerables en la elección del formato del contenedor, los algoritmos de compresión de audio y vídeo, los parámetros de adquisición y la configuración interior de archivos. La combinación de estas características puede ayudar a la autenticación de archivos de vídeo digital en entornos forenses. Después de analizar los vídeos originales se determinó que las estructuras de cada tipo de contenedor no están estrictamente definida como se especifica en los estándares. Se encontraron diferencias considerables entre los vídeos generados por dichos dispositivos. Asimismo, los vídeos AVI después de ser manipulados con los programas de edición, cambiaron la estructura interna incluyendo los valores de los metadatos, entre otras características esenciales para determinar el origen de los vídeos.

En [Vel15] se describe InformaCam una aplicación desarrollada por la empresa de software de código abierto The Guardian Project, que ayuda a los usuarios de dispositivos móviles a proteger su comunicación de intrusiones y monitoreo, es una aplicación de teléfono móvil para administrar metadatos integrados en fotos y vídeos. Su objetivo es recopilar documentación de actos violentos y la vigilancia en línea. Asimismo, estructura y almacena los metadatos de tal manera que el material documentado se acomode mejor a entornos probatorios. Los metadatos de la aplicación, son datos que están incrustados en el archivo, cuando se mueve o se copia. InformaCam incorpora en los dispositivos móviles, un testigo, una funcionalidad adicional que utiliza los sensores del dispositivo móvil para realizar el registro de metadatos contextuales disponibles y manejables para el usuario. InformaCam reconoce explícitamente los teléfonos móviles para facilitar el análisis de datos después de capturar imágenes. A través de estas modificaciones crea una forma de análisis de datos sensoriales forense. Al tratar los datos de esta manera particular, la vigilancia se hace más fácil. En lugar de preocuparse por lo que se puede ver en una imagen, el proyecto muestra cómo la atención se desplaza hacia el arte de mirar, que se refiere a cómo organizar los datos, ubicación, etc.

En [ISF+19] se presenta una técnica novedosa para análisis forense de contenedores de archivos de vídeo no supervisado y presentan dos aplicaciones forenses principales, el primero se ocupa de la verificación de integridad de vídeo, basado en la diferencia entre una referencia y una consulta del contenedor de archivos; el segundo se concentra en la identificación y clasificación de la marca del dispositivo origen, basada en el análisis de la estructura y contenido del contenedor. Notablemente, la última aplicación se basa en el marco de la razón de probabilidad, que es cada vez más es aprobado por la comunidad forense pata exhibir los resultados en la corte. se probó la efectividad de ambas aplicaciones en un conjunto de datos compuesto por 578 vídeos (con

formato MP4 y MOV) tomados con 31 modelos de teléfonos inteligentes de 8 marcas distintas. Los archivos fueron compartidos por YouTube y whatsapp. Se observa una fuerte modificación tanto en la secuencia de datos como en la estructura del contenedor de archivos en los vídeos de youtube. En los vídeos compartidos por WhatsApp se realiza una fuerte modificación tanto del flujo de datos como en la estructura del contenedor de archivos. El método que se propone es capaz de cuantificar objetivamente el resultado de un análisis de la integridad de vídeo basado en contenedores, que puede ser de gran interés en el campo forense.

El avance de la tecnología ha llevado a mejorar el servicio de redes de comunicación en dispositivos móviles. Los teléfonos inteligentes están siendo utilizado para acceso a las redes sociales, transacciones comerciales, así como cometer delitos, comprometiendo la disponibilidad de la evidencia para el proceso forense que causa problemas al analista. Un significante número de delitos cibernéticos se refieren a la posesión ilícita, modificación, y distribución de archivos multimedia. El uso de teléfonos inteligentes para esta finalidad constituyen una motivo de evidencia. Es crucial para los examinadores forenses tener la capacidad de recuperar, analizar y autenticar la fuente de contenidos multimedia almacenados en estos dispositivos. Las investigaciones existentes en este campo se relacionan a continuación.

En [MN15] se propone una solución a la técnica anti-forense de esteganografía mediante diseño e implementación de una aplicación que analiza cualquier información oculta en una imagen, vídeo o archivo de audio en un dispositivo Android y recopila datos para perfiles o investigaciones digitales. El análisis de los datos los hace en png, mp3 y formatos de archivo MP4, La aplicación propuesta preserva los datos usando MD5 como función hash. El escaneado es el proceso de descubrimiento de datos ocultos dentro archivos en un dispositivo Android. El módulo de extracción es el responsable de la adquisición de datos. La aplicación, denominada *Android Mobile Forensic Analyzer*, genera un informe detallado de los resultados del análisis.

El estudio realizado en [HIBS16] se centra en el análisis de archivos multimedia creados en la mayoría teléfonos inteligentes populares con el fin de determinar la fuente y examinar si los archivos son originales o editados a través de estos dispositivos. El análisis se divide en dos partes, se concentra en analizar los archivos multimedia para autenticar la fuente y determinar si los archivos son originales o modificados. Las populares marcas de teléfonos inteligentes analizadas en este documento incluyen iPhone 5, iPhone 6, Blackberry Z10, Samsung Galaxy Note 3, Nokia Lumia 930 y Lenovo A536, utilizando sus utilidades oficiales, como la suite para PC Nokia, Samsung Kies 3, iTunes y BlackBerry Link. Se utiliza las aplicaciones WhatsApp y Viber. El procedimiento se divide en adquisición de firma de archivos multimedia e inspección del contenido. Para inspeccionar el contenido se utiliza el software ExifToolGui. Las contribuciones que se presentan son que la mayoría de los análisis realizados en el zona de estudio de firma de archivos se centran en archivos de computadora, mientras que el estudio se hace en teléfonos inteligentes, además, el análisis de firma de archivos se ha utilizado de manera efectiva para

determinar la fuente y la integridad de los archivos requeridos para el análisis forense. Este estudio proporciona detalles completos de los cambios realizados en el archivo multimedia firmas y metadatos en los teléfonos inteligentes para detectar modificaciones y para autenticar la fuente de archivos multimedia. El enfoque presentado se puede aplicar en otros teléfonos inteligentes como Sony Xperia.

En [MTW16] se propone un sistema de autenticación donde los valores de autenticación del vídeo y el audio se tejen explotando las técnicas de integración de datos en la capa de compresión/medios y capa del sistema. El valor de autenticación del vídeo está incrustado en el audio y viceversa. Los valores de autenticación para audio y vídeo son generados mediante el uso de la clave integrada en la capa del sistema y función hash criptográfica. Hay dos partes en el proceso de verificación; cálculo de los valores de autenticación y se extrae los valores de autenticación incorporados. Para las pruebas se utiliza el software MP4Box. Se descargaron 6 clips de película de YouTube con una resolución de 1280 x 720 píxeles, CNN News, Digital 3D Prueba, Prueba Dolby Digital, Prueba de video de YouTube, Music Video y 3D Transform, de cada película, cinco películas cortas, es decir, 10 segundos de duración. Como prueba de concepto, las ideas se implementan en el contenedor de formato MP4 y se evalúan usando varios vídeos cortos de películas descargados de YouTube. Resultados sugieren que la degradación de calidad imperceptible e insignificante y un incremento de tamaño de archivo debido a la incrustación de datos técnicos.

En [CB16] se estudian los códecs de vídeo de nueva generación están diseñados para mejorar eficiencia de codificación con respecto a normas anteriores y para incorporar el hardware y las aplicaciones más recientes. La detección de doble compresión es una etapa crucial en el estudio forense de imágenes y vídeos digitales. La detección de vídeos doblemente comprimidos es un desafío cuando el falsificador usa la misma matriz de cuantificación y la estructura sincronizada del grupo de imágenes GOP durante el historial de recompresión para ocultar la manipulación efectos: Codificación de vídeo de alta eficiencia (del inglés High Efficiency Video Coding (HEVC)), es el sucesor de Advanced Video Coding (AVC), que es el estándar más adoptado en todo el mundo. El HEVC es el nuevo estándar de generación compresión de vídeo desarrollado conjuntamente por el grupo de expertos en imágenes en movimiento de ISO/IEC y el ITU-T en codificación visual. Al igual que la mayoría de los tipos de codificación de vídeos, HEVC procesa todas las imágenes de la secuencia de entrada en unidades de bloque, cada una de las cuales se compara con un bloque de referencia que se calcula ya sea de imágenes previamente decodificadas (inter-predicción) o muestras previamente descodificadas de la misma imagen (intrapredicción). Una técnica forense para detectar si una secuencia de HEVC se obtuvo de una forma no comprimida secuencia o recodificando una secuencia de AVC existente, el proceso se realiza en cuatro pasos, se vuelve a codificar en el formato AVC, se extrae la huella, se obtienen las distancias de cada elemento y por último utilizando las distancias se detecta vídeo recodificado en

formato AVC o HEVC. Para las pruebas se utilizan 20 secuencias de prueba (del inglés Luminance Bandwidth Chrominance (YUV)) con resolución 532 x 288, se comprimen los vídeos por medio de códecs de código abierto libx264 AVC y libx265 HEVC incluidos en el software multimedia FFmpeg framework5, la discriminación se basa en la presencia de alteraciones introducido por la compresión AVC inicial, de la posterior codificación HEVC con respecto al movimiento.

En [SLLL16] se utiliza el contenido del vídeo almacenado en el contenedor (del inglés Video Event Data Recorders (VEDR)) como evidencia de eventos, como colisiones de vehículos. La mayoría de los formatos multimedia utilizados en VEDR son archivos AVI, MP4 y WMA. El análisis se centra en los vídeos de tipo AVI y en menor proporción los de tipo MP4. Con un sofisticado software de edición de vídeo, los agresores pueden manipular los registros de vídeo en su beneficio sin dejar pistas visibles. La integridad del contenido de vídeo grabado a través de VEDR no se puede garantizar y la cantidad de problemas forenses relacionados aumenta. Específicamente, se propone un mecanismo verificación de integridad, que usa los campos de la estructura en un archivo de vídeo. El método propuesto implica verificar la información del encabezado del contenido de vídeo solo una vez, se requiere mucho menos tiempo de detección en comparación con los métodos existentes que examinan fotogramas enteros. Se almacena una estructura de archivo ordenada del contenido del vídeo como firma de cada herramienta de edición de vídeos. El contenido sospechoso de vídeo se compara con ese conjunto de firmas. Si la estructura del archivo coincide con una firma, se identifica como un vídeo manipulado por esa herramienta de edición. Se prueban 296 vídeos adquiridos con 43 VEDR y manipulados con 5 diferentes herramientas de edición de vídeos (Sony VEGAS, Adobe Premiere, Edius, Avid MC y Avid Studio) que cubren el 99 % de la porción del mercado del software de edición de vídeo. Se concluye que las diferentes herramientas de edición de vídeos tienen un campo único de estructuras, asimismo, el formato AVI no tiene un estándar estricto, a su vez, la manipulación del contenido de vídeo cambia la estructura de datos del vídeo de acuerdo con el software de edición de vídeo y la estructura del contenido de vídeo editado es consistente e independientemente de la estructura de campo original y dependiendo del códec, si un usuario selecciona opciones idénticas al guardar un vídeo editado, la estructura de datos es la misma.

En [AB18] se propone un enfoque pasivo para detectar vídeos MPEG con doble compresión con la misma matriz de cuantificación y estructura GOP sincronizada. Para diseñar el algoritmo propuesto, se estudian matemáticamente los efectos de la recompresión en marcos P. Con base en las pautas obtenidas, se propone un vector de características para detectar fotogramas doblemente comprimidos en el nivel GOP. Representaciones dispersas de los vectores de características se utilizan para reducir la dimensionalidad y enriquecer los rastros de la recompresión. El Support Vector Machine (SVM) se emplea para detectar y localizar la compresión doble en el dominio temporal. El algoritmo propuesto comprende cuatro etapas, recompresión de vídeo y extracción de fotogramas P, función extracción y representación

dispersa de las características extraídas y por último la clasificación. En las pruebas la solución muestra que el algoritmo logra la precisión en más del 95 %, usando siete fotogramas P de los GOP. También se comprueba que el clasificador SVM tiene superior rendimiento en comparación con el clasificador K-Nearest Neighbors (KNN). Las comparaciones de los resultados del método con los de otros métodos revelan la eficiencia del algoritmo. Se concluye que la diferencia de los fotogramas P durante la compresión sucesiva tiene una tendencia a disminuir, la matriz de cuantificación y la estructura GOP sincronizada permiten identificar la simple y doble compresión.

En [HFC18] centra su estudio en las formas estenográficas de ocultar información utilizando técnicas de bits menos significativas. Se demuestra que un certificado digital relacionado con un archivo de imagen se puede insertar dentro de ese archivo junto con los metadatos que contienen referencias del emisor. A pesar de las variaciones entre dispositivos, sistemas operativos y aplicaciones, un archivo JPEG mantiene su estructura cuando se producen cambios, se encuentra en el área de metadatos y no afecta los datos de la imagen codificada, que debe ser verificable. Las referencias al autor se pueden insertar en los metadatos del archivo. Hay una ventaja de tener el certificado digital como parte integral del archivo al que se aplica, se demuestra que los metadatos de un archivo pueden incluir datos que se puede usar para demostrar la integridad, autenticidad y procedencia del contenido digital. Utilizando metadatos para autenticidad, la aplicación más flexible disponible en código abierto es ExifTool, cubre una amplia gama de tipos de archivos, ejecutables y archivos de imagen, además ofrece una amplia funcionalidad tanto para ver y editar metadatos. Los dos primeros bytes en un archivo JPEG son la firma del archivo con un valor de 0xFFD8. JPEG es una compresión con pérdida técnica que utiliza la cuantificación y codificación de Huffman. Se asignan dieciséis segmentos de aplicación para metadatos, ofrece flexibilidad y es un lugar ideal para guardar un certificado digital. Los metadatos EXIF siempre están contenidos en un segmento, las etiquetas de metadatos hay 231 y solo unas pocas se usan para JPEG. Los cambios en los metadatos no necesariamente afectan la integridad de un archivo, si los datos centrales del archivo no cambian. El proceso correspondiente para verificar un archivo es sencillo, primero se extrae el certificado digital del archivo, segundo se lee valores hash del certificado digital, tercero se calcula valores hash para cada sección del archivo: segmentos de metadatos, segmentos de compresión y datos de imágenes codificadas, cuarto se compara los valores hash calculados con los valores extraídos del certificado digital según corresponda:

- 0 (El archivo no ha sido modificado desde que el certificado digital fue emitido)
- 1 (Los metadatos o la compresión han sido alterados pero el núcleo de datos de imagen no se modifica, por lo que el certificado digital sigue siendo válido)
- 2 (El certificado digital ya no es válido como imagen principal los datos han sido alterados)

Al realizar las pruebas se llega a la decisión, que a pesar de variaciones entre dispositivos, sistemas operativos y aplicaciones, un archivo JPEG mantiene su estructura. Se realizan cambios en el área de metadatos y esto no afecta los datos de la imagen codificada. Referente a la emisión se puede insertar en los metadatos del archivo, el certificado digital como parte integral del archivo. Se demuestra que los metadatos dentro de un archivo ofrecen la posibilidad de incluir datos que puedan usarse para demostrar integridad, autenticidad y procedencia del contenido digital. La investigación también descubre cómo los metadatos son manejados por varios sistemas operativos, dispositivos y aplicaciones de gestión de imágenes. Debido a su gran popularidad, los archivos de imagen, especialmente JPEG, ofrecen un alto potencial como portadores de otra información.

En [KYK19] se propone un eficiente sistema forense de archivos multimedia basado en la similitud de archivos. El sistema planteado se sostiene en la detección de información con reconocimiento de medios y un esquema de coincidencia basado en huella digital. El sistema divide un archivo multimedia en muchos planos a través del esquema de detección de límites de disparo. Extrae fotogramas clave de cada disparo, y calcula los valores hash de cada uno de los fotogramas clave mediante el uso de la función hash. La huella dactilar se genera a partir de cada fotograma clave utilizando la función de hash perceptual basada en DCT. Cada huella dactilar de vídeo sirve como el identificador de cada uno de los fotogramas clave, y se usa como la unidad estándar para una comparación de disparo a disparo. El sistema usa la coincidencia basada en huella digital esquema para la búsqueda de similitud de archivos. Para las pruebas se utilizan cinco vídeos en formato MOV y MP4. Los resultados del experimento se obtuvieron comparando el contenido original y el contenido modificado por cambios de resolución. Se muestra que la precisión promedio de las huellas dactilares de vídeo es del 97.86 % y el promedio del vídeo en relación a su similitud es 91.8 % como resultado de la correspondencia entre el contenido original y el contenido modificado, el sistema propuesto puede usarse para determinar si la distribución no autorizada contra el contenido original mediante el uso de la búsqueda de similitud entre archivos multimedia.

En este Trabajo Fin de Máster se propone la idea de procesar la información almacenada en el contenedor de datos del vídeo digital como un datagrama para el análisis forense. Hasta el momento no existe en la literatura forense una investigación que reporte detalles específicos sobre el contenedor multimedia, ni sobre la exploración sistemática del formato del contenedor y/o metadatos. Las diferencias en la estructura de archivos JPEG con la estructura de los fotogramas del vídeo, es que ésta últimas identifican características del formato del contenedor del vídeo, del fabricante, el modelo y los rastros que puede dejar el software de procesamiento. Tales rastros se pueden usar para autenticar secuencias del vídeo digital y para atribuir grabaciones desconocidas o de cuestionable procedencia a modelos de cámaras de vídeo.

3.4. Herramientas de Extracción de Información de Vídeos Digitales

Al realizar una revisión en todo el espectro de mercado no existen muchas herramientas de extracción de átomos y metadatos de vídeos, por lo cual se han seleccionado dos herramientas que han permitido realizar una comparación con la herramienta desarrollada:

- **ExifToolGUI**: En [Wikb] se indica que esta disponible como una biblioteca Perl (Image::ExifTool), asimismo como una aplicación de línea de comandos. Es una aplicación de software gratuito y de código abierto para leer, escribir y manipular metadatos de imagen, audio, vídeo y PDF. ExifToolGUI admite muchos tipos de metadatos, incluidos EXIF, IPTC, XMP, JFIF, GeoTIFF, ICC Profile, Photoshop IRB, FlashPix, AFCP e ID3, así como los formatos de metadatos específicos del fabricante de muchas cámaras digitales y se incorpora comúnmente en diferentes tipos de flujos de trabajo digitales.
- **Mp4 Parser**: En [Inc] se describe la herramienta Mp4 Parser tiene una versión web para realizar la lectura de la estructura del archivo MP4 y una aplicación (del inglés Application Programming Interface (API)) que además permite editar la estructura. Es una herramienta de bajo nivel que analiza los fotogramas, y las estructuras como pistas y películas. Iso Viewer ayuda a visualizar la estructura del formato MP4. Las tareas típicas del analizador MP4 son la mezcla de audio o vídeo en un archivo MP4, agregar grabaciones que usan la misma configuración de codificación, y también se pueden agregar o cambiar metadatos y reducir las grabaciones omitiendo fotogramas.

El resultado de esta comparación se puede observar también de forma gráfica en el Capítulo A, dónde se presenta las capturas de pantalla de las aplicaciones ExifToolGUI y Mp4 Parser.

Capítulo 4

Método Propuesto

En este capítulo se presentan las contribuciones del presente trabajo que consiste en un método forense que permite realizar un análisis exhaustivo de la estructura del contenedor multimedia de vídeos con formato MOV y generados por dispositivos móviles. El capítulo se divide en tres secciones. En la Sección 4.1 se detalla las consideraciones generales de la propuesta, explicando las características utilizadas y algunas limitaciones de trabajos similares. En la Sección 4.2 se presenta el método propuesto que consiste en un conjunto de operaciones agrupadas en tres fases. Por último, en la Sección 4.3 se da a conocer la tecnología utilizada para el desarrollo del método propuesto.

4.1. Consideraciones Generales

Como se mencionó en el Capítulo 2, el sistema de los depósitos multimedia de vídeos están constituidos por elementos o características intrínsecas denominados átomos. En consecuencia, en este trabajo se propone un método forense que permite analizar exhaustivamente el contenedor de vídeos con formato MOV, concretamente vídeos generados por dispositivos móviles. El método está basado a la extracción del contenido almacenado en el contenedor multimedia y operaciones de análisis de datos.

El resultado del método propuesto hace posible realizar comparaciones entre estructuras de vídeos originales con estructuras de vídeos procesados por redes sociales y editores de vídeos. Para ello, se aprovecha el hecho que tanto los fabricantes de dispositivos móviles, las redes sociales y editores de vídeos insertan a los contenedores un conjunto de características capaces de distinguir la procedencia de un vídeo dado.

Los últimos años han surgido trabajos con objetivos similares pero la mayoría se ha enfocado sobre vídeos con formato MP4 y AVI. Asimismo, otras investigaciones están dirigidas a vídeos capturados por cámaras digitales tradicionales.

4.2. Descripción del Método Propuesto

El método propuesto tiene por objetivo analizar detalladamente la estructura del contenedor multimedia MOV. Para conseguir este objetivo se realiza una serie de operaciones agrupados en tres fases:

1. **Extracción de átomos:** Teniendo como entrada un conjunto de vídeos con formato MOV, se realiza la extracción de la estructura de contenedor con un algoritmo propuesto.
2. **Pre-procesamiento:** Se realiza una limpieza de datos de la información extraída en la fase anterior para asegurar la calidad de los datos.
3. **Análisis:** Se efectúa un análisis del rendimiento del algoritmo de extracción de átomos propuesto y un análisis exhaustivo de la estructura del contenedor siguiendo una configuración (vídeos originales y vídeos post-procesados).

En la Figura 4.1 se presenta las 3 fases del método propuesto.

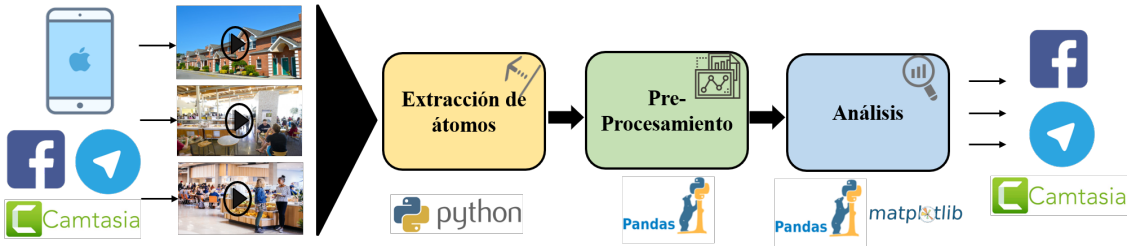


Figura 4.1: Fases del método de análisis de la estructura del contenedor MOV

4.2.1. Extracción de los Átomos

Previamente al análisis de estructura del contenedor MOV, es necesario extraer la información almacenada en cada uno de los vídeos del grupo de datos de entrada. Como se mencionó en el Capítulo 2, los átomos son elementos organizados de forma jerárquica en el contenedor multimedia y tienen la capacidad de diferenciar la marca de dispositivo, modelo, red social y programa de edición. Dicha jerarquía se define como sigue:

- **/moov/**: Es el átomo que contiene una serie de átomos hijos.
- **/moov/trak/tkhd/:** Es una secuencia de átomos. El átomo tkhd es hijo del átomo trak y este último es hijo del átomo moov.
- **moov/trak/tkhd/trackID:** El elemento trackID es una etiqueta del átomo tkhd.
- **/moov/trak/tkhd/trackID: 2** : Generalmente las etiquetas tienen un valor trackID : “2”.

Considerando lo descrito en los párrafos precedentes, el proceso de extracción de la estructura del contenedor multimedia MOV se realiza con el Algoritmo 1. Dicho proceso tiene como entrada un vídeo con formato MOV y como salida un conjunto características que son almacenados en una hoja de datos o *DataFrame*. La descripción del algoritmo propuesto se realiza en 8 pasos:

1. Se obtiene la longitud total de un átomo $\text{Tamaño}_{\text{atomoleido}}$, que ayuda a determinar el número de bytes que se puede de leer, esto permitirá identificar los sub átomos que puede contener un átomo padre.

2. Sí $\text{Tamaño}_{\text{atomoleido}}$ es mayor que 0, se obtiene el primer byte de cada átomo que se desea leer $\text{Byte}_{\text{inicialatomo}}$.

3. Se obtiene la el tamaño total en bytes del átomo $\text{Posicion}_{\text{lectura}}$, este valor asegura una lectura de bytes de forma correcta mediante la comprobación de la posición.

4. Se obtiene el nombre o etiqueta del átomo $\text{Tipo}_{\text{atomoleido}}$. Normalmente es representado en un tipo de dato string de 4 bytes, posteriormente con este dato se realizan comprobaciones de duplicidad de etiquetas.

5. Mientras $\text{Byte}_{\text{inicialatomo}}$ más $\text{Tamaño}_{\text{atomoleido}}$ mayor que $\text{Posicion}_{\text{lectura}}$ esto significa que el átomo inicialmente leído contiene uno o más átomos hijos, con esto se tiene un mejor control, representación de los átomos o etiquetas y evitar duplicidad de nombres, luego se ejecuta el procedimiento $\text{Leer}_{\text{atomos}}$. Este último procedimiento es el más importante debido a que contiene el diccionario de todos los átomos referenciados en el estándar; su ejecución es recursiva mientras exista átomos por leer, va realizando un control para evitar duplicidad, esto afecta a todos los átomos contenidos en un vídeo.

6. Si cumple con la especificación del fabricante, se extrae el nombre del átomo y su valor correspondiente $\text{Etiquetayvalor}_{\text{atomoleido}}$. Por ejemplo, la etiqueta *creationTime* tiene como valor la fecha de creación del vídeo en el formato establecido por la especificación.

7. En el caso de producirse una excepción, se muestra un error de lectura.

8. Finalmente, si el átomo extraído no cumple con la especificación del estándar se guarda como átomo desconocido.

Algoritmo 1: Extracción de Átomos de Vídeos con formato MOV

Input Video_{mov}: Vídeo con formato multimedia MOV
Output $\text{Átomos}_{extraídos}$: Átomos extraídos del vídeo MOV

```
EXTRAERATOMOS(Video_mov)
Obtener Tamaño_atomoleido;
if Tamaño_atomoleido > 0 then
    Obtener Byte_inicialatomo;
    Obtener Posicion_lectura;
    Obtener Tipo_atomoleido;
    While Byte_inicialatomo + Tamaño_atomoleido > posicion_lectura do
        Leer_atomos;
        Obtener Etiquetayvalor_atomoleido;
    except
        Error de lectura
else
    Guardar átomo desconocido
end procedure
```

El DataFrame tiene un total de 7 campos que se detallan en la Tabla 4.1.

Tabla 4.1: Campos del DataFrame

Ruta-origen	Nombre-fichero	Marca	Modelo	Ruta-etiqueta	Valor	Orden
/home/user/tfm/Ipad2	VIDEO_5254.MOV	Apple	iPad2	/moov/trak3/edts/elst/version	0	111

4.2.2. Pre-procesamiento de los Datos Extraídos

En esta fase se hace una depuración de los datos para eliminar posibles errores con el fin de obtener un conjunto de datos perfectamente codificados y sobre todo consistentes. Para ello se ejecuta un proceso de limpieza de datos que no intervienen en el análisis y son los siguientes:

1. Eliminación de las columnas Ruta-origen y Nombre-fichero que no se hacen uso en el presente análisis. Además, se elimina las columnas Marca y Modelo porque dicha información se encuentra en el campo Valor.

2. Búsqueda y corrección de valores especiales. Por ejemplo, lo valores de las etiquetas "entries", que por su gran tamaño dichos valores son ignorados.

4.2.3. Análisis

El análisis propuesto depende de cinco procedimientos previos que permitirán obtener resultados claros y precisos. Estos procedimientos se describen a continuación:

1. **Uso del algoritmo de extracción de átomos de vídeos MP4:** Se utiliza un algoritmo robusto capaz de extraer toda la información del contenedor multimedia de vídeos con formato MP4. Esta técnica se realiza con la finalidad de hacer comparaciones de las

estructuras de vídeos que inicialmente tenían el formato MOV, pero al ser manipulados cambian obligatoriamente el tipo de contenedor a MP4. Este algoritmo fue desarrollado por el Grupo de Análisis, Seguridad y Sistemas (GASS) [Gro17].

2. **Consolidación de la estructura**: Se procede a consolidar una estructura única por cada dispositivo, marca, modelo, red social y programa de edición. Para lograr este cometido se recurre al uso de variables binarias como se explica a continuación:
   * 1: Cuando la Ruta-etiqueta tiene la situación de presencia el vídeo.
   * 0: Cuando el Ruta-etiqueta tiene una situación de ausencia en el vídeo.

3. **Implementación de una función de relación lineal**: Se procede a implementar una función que permite medir la correspondencia o relación lineal entre dos variables, por ejemplo (entre marcas). En este sentido, se utilizó el coeficiente de correlación de pearson que posee propiedades para variables binarias y toma un valor en el rango [-1,1]. En la Tabla 4.2 se muestra el el resumen de los criterios de interpretación:

Tabla 4.2: Criterios de interpretación de la correlación

Criterio de Interpretación	Valor
Correlación negativa perfecta	-1,00
Correlación negativa muy fuerte	-0.90
Correlación negativa media	-0.50
Correlación negativa débil	-0.10
No existe correlación lineal alguna entre las variables	0.00
Correlación positiva débil	0.10
Correlación positiva media	0.50
Correlación positiva muy fuerte	0.90
Correlación positiva perfecta	1.00

4. **Configuración del análisis propuesto**:

   Esta configuración consiste en planificar los diferentes tipos de análisis que se ejecutarán en la siguiente fase de la propuesta. Se tiene en cuenta factores como la diversidad de origen de los vídeos del conjunto de datos y las características obtenidas con el algoritmo de extracción de átomos.

   En la Figura 4.2 se muestra la configuración del análisis de la estructura del contenedor de vídeos. Esta configuración está dividida en 4 partes y son las siguientes:

   - **Análisis del algoritmo de extracción:** Se realiza un análisis de la capacidad de lectura y extracción del algoritmo propuesto y se compara con otras herramientas existentes en el mercado. Esto permite identificar las ventajas y desventajas de cada herramienta y proponer mejoras en la propuesta.

- **Análisis de la estructura del contenedor:** En primer lugar, se realiza el análisis de la estructura del contenedor de vídeos originales a nivel de dispositivos a fin de detectar si dispositivos de la misma marca y modelo generan vídeos con la misma estructura. En segundo lugar, se realiza el mismo análisis anterior pero a nivel modelos. En tercer lugar, se analiza la estructura a nivel de fabricantes. Finamente, se realiza un análisis de los vídeos procesados por redes sociales y editores de vídeos ya que ambas aplicaciones realizan cambios al contenedor a la hora de pasar por sus plataformas e inclusive la mayoría cambia el tipo de contenedor MOV a MP4.
- **Análisis del orden de almacenamiento de los átomos:** Este análisis se basa en el orden de aparición de cada átomo en el contenedor multimedia. Primero, se analiza el orden de aparición de los vídeos originales. Después, se realiza el mismo análisis a los vídeos de redes sociales y vídeos manipulados por editores a fin de observar las variaciones que sufre el vídeo luego de ser manipulado por ambas herramientas.
- **Análisis a nivel de contenido de los átomos:** Este análisis está orientado a identificar lo valores específicos que tiene cada etiqueta en el contenedor multimedia. Para ello, se analizan los vídeos originales y posteriormente los vídeos de redes sociales y de editores. Al igual que la estructura y el orden de almacenamiento de los átomos, el contenido de un vídeo original también cambia luego de ser manipulado.

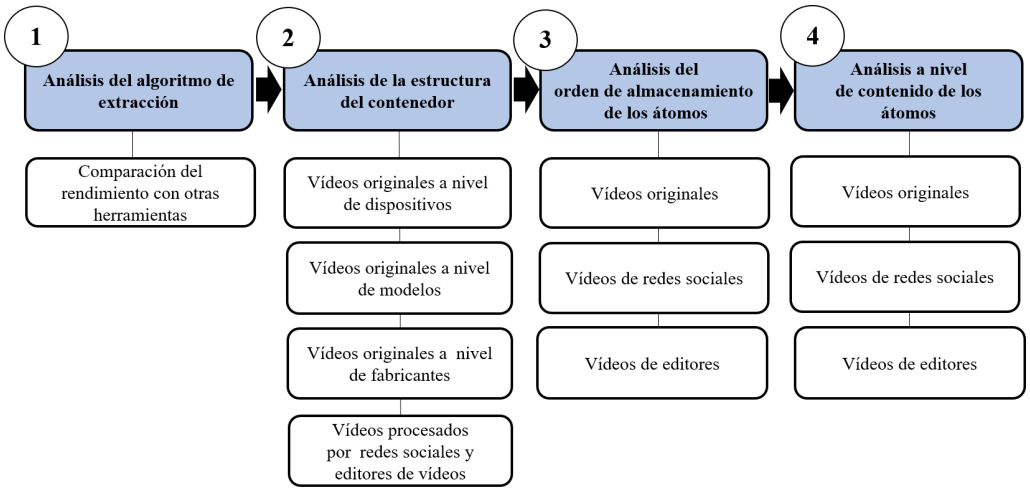


Figura 4.2: Configuración del análisis propuesto

5. **Análisis Exploratorio**: Este análisis preliminar se realiza para entender como están organizadas las características extraídas y demostrar cual es el comportamiento de los vídeos originales después de ser manipulados por una red social o un editor de vídeo.

   En la Figura 4.3, se observa el DataFrame de un vídeo original generado por un dispositivo móvil de marca apple y modelo Ipad 2 (X). En la Figura 4.4, se muestra otro DataFrame del mismo vídeo (X), después de ser compartido por WhatsApp (X_1). En la Figura 4.5, se

muestra otro DataFrame del mismo vídeo (X), después de ser manipulado por el programa Adobe Premier (X_2).

	Ruta-origen	Nombre-fichero	Marca	Modelo	Ruta-etiqueta	Valor	Orden
0	/home/user/tfm/Ipad2_D01	video 14-4-18 17 34 00.mov	Apple	iPad 2	/ftyp/	ftyp	1
59	/home/user/tfm/Ipad2_D01	video 14-4-18 17 34 00.mov	Apple	iPad 2	/ftyp/compatibleBrands	qt	1
13	/home/user/tfm/Ipad2_D01	video 14-4-18 17 34 00.mov	Apple	iPad 2	/ftyp/majorBrand	qt	1
35	/home/user/tfm/Ipad2_D01	video 14-4-18 17 34 00.mov	Apple	iPad 2	/ftyp/minorVersion	0	1
3	/home/user/tfm/Ipad2_D01	video 14-4-18 17 34 00.mov	Apple	iPad 2	/wide/	wide	2
7	/home/user/tfm/Ipad2_D01	video 14-4-18 17 34 00.mov	Apple	iPad 2	/moov/trak/	trak	6
10	/home/user/tfm/Ipad2_D01	video 14-4-18 17 34 00.mov	Apple	iPad 2	/moov/trak1/	trak	34
11	/home/user/tfm/Ipad2_D01	video 14-4-18 17 34 00.mov	Apple	iPad 2	/moov/trak2/	trak	60
12	/home/user/tfm/Ipad2_D01	video 14-4-18 17 34 00.mov	Apple	iPad 2	/moov/trak3/	trak	108
1	/home/user/tfm/Ipad2_D01	video 14-4-18 17 34 00.mov	Apple	iPad 2	/mdat/	mdat	3
650	/home/user/tfm/Ipad2_D01	video 14-4-18 17 34 00.mov	Apple	iPad 2	/moov/trak2/mdia/minf/stbl/stsd/mebx/keys/loca...	0	88

Figura 4.3: DataFrame de vídeo original (X)

	Ruta-origen	Nombre-fichero	Marca	Modelo	Ruta-etiqueta	Valor	Orden
1	/home/user/tfm/Wattsapp /Ipad2_D02	video 14-4-18 17 34 00.mp4	NaN	NaN	/ftyp/	ftyp	1
16	/home/user/tfm/Wattsapp /Ipad2_D02	video 14-4-18 17 34 00.mp4	NaN	NaN	/ftyp/minorVersion	1	1
28	/home/user/tfm/Wattsapp /Ipad2_D02	video 14-4-18 17 34 00.mp4	NaN	NaN	/ftyp/compatibleBrands	mp41mp42isom	1
8	/home/user/tfm/Wattsapp /Ipad2_D02	video 14-4-18 17 34 00.mp4	NaN	NaN	/ftyp/majorBrand	mp42	1
0	/home/user/tfm/Wattsapp /Ipad2_D02	video 14-4-18 17 34 00.mp4	NaN	NaN	/beam/	beam	2
6	/home/user/tfm/Wattsapp /Ipad2_D02	video 14-4-18 17 34 00.mp4	NaN	NaN	/moov/trak/	trak	5
7	/home/user/tfm/Wattsapp /Ipad2_D02	video 14-4-18 17 34 00.mp4	NaN	NaN	/moov/trak1/	trak	27
2	/home/user/tfm/Wattsapp /Ipad2_D02	video 14-4-18 17 34 00.mp4	NaN	NaN	/mdat/	mdat	46
236	/home/user/tfm/Wattsapp /Ipad2_D02	video 14-4-18 17 34 00.mp4	NaN	NaN	/moov/trak/mdia/minf/stbl/stsd/avc1/avcC /avcDe...	3,39106093385327E+067	19

Figura 4.4: DataFrame del vídeo compartido por WhastApp (X_1)

Al compararse los registros de X con X_1 y X_2 con operaciones básicas se identificaron notables diferencias entre las estructuras. Los resultados de esta comparación se detallan en la Tabla 4.3.

Se observa que un vídeo original con formato MOV tiene una estructura definida y generalmente cumple con las especificaciones del estándar. Sin embargo, este vídeo cambia drásticamente su estructura cuando sufre algún proceso de manipulación. Entre los principales cambios están los efectuados a los valores de cada Ruta-etiqueta, modificación del orden de aparición de los átomos y eliminación átomos.

Por lo tanto, este análisis previo demuestra claramente que existen elementos diferenciadores entre un vídeo original y otros que hayan sido manipulados por algún actor secundario (red social o programa de edición).

	Ruta-origen	Nombre-fichero	Marca	Modelo	Ruta-etiqueta	Valor	Orden
1	/home/user/tfm/AdobePremiere /Ipad2_D02	video 14-4-18 17 34 00.mov	NaN	NaN	/ftyp/	ftyp	1
21	/home/user/tfm/AdobePremiere /Ipad2_D02	video 14-4-18 17 34 00.mov	NaN	NaN	/ftyp/compatibleBrands	qt	1
3	/home/user/tfm/AdobePremiere /Ipad2_D02	video 14-4-18 17 34 00.mov	NaN	NaN	/moov/	moov	2
54	/home/user/tfm/AdobePremiere /Ipad2_D02	video 14-4-18 17 34 00.mov	NaN	NaN	/moov/mvhd/matrixStructure	[[1, 0, 0], [0, 1, 0], [0, 0, 1]]	3
55	/home/user/tfm/AdobePremiere /Ipad2_D02	video 14-4-18 17 34 00.mov	NaN	NaN	/moov/mvhd/preferredVolume	1	3
6	/home/user/tfm/AdobePremiere /Ipad2_D02	video 14-4-18 17 34 00.mov	NaN	NaN	/moov/trak/	trak	4
8	/home/user/tfm/AdobePremiere /Ipad2_D02	video 14-4-18 17 34 00.mov	NaN	NaN	/moov/trak1/	trak	29
9	/home/user/tfm/AdobePremiere /Ipad2_D02	video 14-4-18 17 34 00.mov	NaN	NaN	/moov/trak2/	trak	51
2	/home/user/tfm/AdobePremiere /Ipad2_D02	video 14-4-18 17 34 00.mov	NaN	NaN	/mdat/	mdat	77
380	/home/user/tfm/AdobePremiere /Ipad2_D02	video 14-4-18 17 34 00.mov	NaN	NaN	/moov/trak/mdia/minf/stbl/stsd/dvc /colr/trans...	1	22

Figura 4.5: DataFrame del vídeo manipulado por Adobe Premiere (X_2)

Tabla 4.3: Comparación de estructuras entre X, X_1 y X_2

Característica	Vídeo X	Vídeo X_1	Vídeo X_2	Observaciones
Valor de la etiqueta compatibleBrands	qt	mp41, mp42, isom	qt	Según la especificación del estándar el vídeo original cumple con poseer el valor de "qt". Sin embargo, el vídeo de WhatsApp cambia el valor a "mp41, mp42, isom". Por otro lado, Adobe Premiere mantiene el valor "qt".
Átomo en la posición 2	wide	beam	moov	El vídeo original contiene el átomo "wide", WhastsApp cambia a "beam" y Adobe Premiere también modifica a "moov".
Cantidad de átomos trak	4	2	3	El vídeo original tiene 4 átomos trak,de los cuales 1 tiene información de vídeo, 1 información de audio y las dos restantes tienen información de metadatos. Cuando el vídeo es transferido por WhatsApp, el número de átomos trak se reduce a 2. El vídeo manipulado por Adobe Premiere reduce de 4 a 3 átomos trak con respecto al original.
Orden de lectura del átomo mdat	3	46	77	En el vídeo original aparece en el orden de lectura 3; mientras que en WhatsApp aparece en el orden 46 y en Adobe Premiere en el orden 77.
Cantidad de etiquetas y átomos (campo Ruta-etiqueta)	651	237	381	La cantidad de características o también llamado Ruta-etiqueta del vídeo original es de 651, mientras se reduce a 237 en el caso de WhatsApp y 381 para Adobe Premiere.

4.3. Tecnología Utilizada

El algoritmo desarrollado se puede puede encontrar en el siguiente enlace: http://gitlab.fdi.ucm.es/tfm-2020/tfm-roberto.git.

Para el desarrollo del método se propuesto se ha utilizado las siguientes herramientas:

- **Lenguaje de programación:** Se ha utilizado *Python* en su versión 2.7 bajo el sistema operativo Linux con distribución Ubuntu versión 18.04. Python Software Foundation posee una licencia de código abierto y es compatible con la licencia pública general de GNU a partir de la versión 2.1.1 [Fou].
- **Entorno de desarrollo:** Se ha aprovechado la asistencia de *PyCharm Professional* versión 2017.3.3. La elección de este entorno desarrollo se ha debido a la gran cantidad de funcionalidades que ofrece como la asistencia inteligente a la codificación, es decir, proporciona una finalización del código inteligente, inspecciones del código, indicación de errores sobre la marcha y arreglos rápidos, así como refactorización de código automática y completas funcionalidades de navegación [S.R].
- **Librerías:** Para las fases de pre-procesamiento y análisis propuesto se ha utilizado la librería *Pandas* que fue construido con el paquete Numpy y su estructura de datos clave es llamada el DataFrame [Num]. Asimismo, para la visualización de datos y generación de gráficos a partir de los datos extraídos de los contenedores multimedia se ha recurrido a la librería *Matplotlib* para Python [Tea].

Capítulo 5

Resultado del Análisis

En este capítulo se describen los resultados conseguidos en el análisis realizado en el Capítulo 4. En la Sección 5.1 se explica como se configura el dataset utilizado sobre un grupo de vídeos de dispositivos móviles y las características de como se procesaron por las redes sociales y aplicaciones de edición de vídeos, en la Sección 5.2 se hace un estudio del algoritmo de extracción de átomos proporcionado por nuestra herramienta frente al ofrecido por otras herramientas. En la sección 5.3 se hace un análisis de la estructura del contenedor para vídeos originales a nivel de dispositivos, modelos y fabricantes, asimismo, para los vídeos procesados por redes sociales y aplicaciones de edición de vídeos. En la sección 5.4 se hace un análisis del orden de almacenamiento de los átomos tanto para los vídeos originales como para los procesados por redes sociales y aplicaciones de edición de vídeos y por último en la Sección 5.5 se hace un análisis a nivel de contenido de átomos tanto para los vídeos originales como para los procesados por redes sociales y aplicaciones de edición de vídeos.

5.1. Configuración del dataset

Para la evaluación del método propuesto se ha construido un conjunto de vídeos de dispositivos móviles de diferentes marcas y modelos recolectados directamente de los propios dispositivos. De algunos de los modelos se recolectaron vídeos de dos usuarios diferentes con el mismo modelo. Posteriormente, se seleccionó un dispositivo por cada modelo del conjunto de datos para compartir sus vídeos a través de varias redes sociales. Esos mismos vídeos han sido editados por por varias herramientas de edición. En la Tabla 5.1 se muestran las características del conjunto de 1630 vídeos clasificados por dispositivo, modelo y marca.

Tabla 5.1: Dataset

Vídeos de Dispositivos Móviles					Redes Sociales							Editores de Vídeos			Total
Marca	**Modelo**	**Id_Dispositivo**	**Versión**	**Nº de vídeos**	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**E1**	**E2**	**E3**	**vídeos**
Apple (M_01)	Ipad2 (MO_01)	D01	7.1.1	10											10
		D02	9.3.5	10	10	10	10	10	10	10	10	10	10	10	110
	IpadAir (MO_02)	D03	11.3	10	10	10	10	10	10	10	10	10	10	10	110
	Iphone5 (MO_03)	D04	7.0.4	10											10
		D05	8.4	10	10	10	10	10	10	10	10	10	10	10	110
		D06	9.3.3	10											10
	Iphone5S (MO_04)	D07	9.2	10	10	10	10	10	10	10	10	10	10	10	110
	Iphone6 (MO_05)	D08	8.4	10											10
		D09	9.2	10	10	10	10	10	10	10	10	10	10	10	110
		D10	10.1.1	10											10
		D11	11.2	10											10
	Iphone7 (MO_06)	D12	11.0.3	10											10
		D13	11.2.6	10	10	10	10	10	10	10	10	10	10	10	110
		D14	11.2.6	10											10
	Iphone8+ (MO_07)	D15	11.2.5	10	10	10	10	10	10	10	10	10	10	10	110
		D16	11.2.6	10											10
Samsung (M_02)	A6 (MO_08)	D17	8.0.0	10	10	10	10	10	10	10	10	10	10	10	110
	S9+ (MO_09)	D18	8.0.0	10	10	10	10	10	10	10	10	10	10	10	110
	J5 (MO_10)	D19	6.0.1	10	10	10	10	10	10	10	10	10	10	10	110
	TabA (MO_11)	D20	-	10	10	10	10	10	10	10	10	10	10	10	110
Huawei (M_03)	Ascend (MO_12)	D21	-	10	10	10	10	10	10	10	10	10	10	10	110
	P9 (MO_13)	D22	6.0	10	10	10	10	10	10	10	10	10	10	10	110
Xiaomi (M_04)	Redmi Note 5 (MO_14)	D23	8.1.0	10	10	10	10	10	10	10	10	10	10	10	110
				230	140	140	140	140	140	140	140	140	140	140	1630

Como se observa en la Tabla 5.1 el conjunto de datos está compuesto por dos tipos de vídeos:

- **Vídeos de dispositivos móviles**: Este conjunto de datos está compuesto por vídeos sin post-procesamiento extraídos de dispositivos móviles. Los vídeos fueron recopilados directamente de dispositivos móviles de personas conocidas lo que confirma que los vídeos no han sido objeto de ningún tipo de manipulación después de haberse grabado. Se recopilaron 230 vídeos grabados por 23 dispositivos de 14 modelos provenientes de 4 marcas (Apple, Samsung, Huawei y Xiaomi). Estos vídeos los denominaremos en adelante "Vídeos originales".
- **Vídeos generados a partir de los vídeos originales**: Se eligió de forma aleatoria 1 dispositivo para cada uno de los 14 modelos del conjunto de vídeos para ser transferidos a través de 7 redes sociales y manipulados por 3 programas de edición. Cada red social o programa de edición realiza algunas operaciones o configuraciones al momento de compartir el vídeo o lo edita. Se generaron 980 vídeos compartidos por redes sociales y 420 vídeos manipulados desde 3 programas de edición.

En la Tabla 5.2 se muestran las características y operaciones realizadas sobre los vídeos compartidos por las diferentes redes sociales, aplicaciones de mensajería instantánea o programa de edición de vídeos.

Tabla 5.2: Características de las redes sociales y los programas de edición utilizados

Tipo	Id	Herramienta Utilizada	Versión	Características	Contenedor Resultante
Red Social	R1	Facebook SD	Sitio web	Firefox	MP4
	R2	Facebook HD	Sitio web	Firefox	MP4
	R3	Flickr	Sitio web	Firefox	MOV
	R4	LinkedIn	Sitio web	Firefox	MP4
	R5	YouTube	Sitio web	YouTube studio beta	MP4
Mensajería instantánea	R6	Telegram	5.0	App. Móvil	MP4
	R7	WhatsApp	2.19.20	App. Móvil	MP4
Editor de vídeos	E1	Adobe Premiere	2018-12.0	Abrir/Guardar	MOV
	E2	Camtasia	2018.0.1	Abrir/Guardar	MP4
	E3	Lightworks	14.5	Importar/Exportar	MP4

Para el caso de la red social Facebook se usaron dos tipos de calidades para compartir los vídeos: la definición estándar (del inglés Standard Definition (SD)) o alta definición High Definition (HD). El procesamiento se realizó directamente de su sitio web subiendo con el navegador Firefox. Los vídeos compartidos por la mayoría de las redes sociales cambiaron de formato MOV a MP4. La única red social que mantiene el formato original de los vídeos compartidos es Flickr. Para compartir los vídeos por las aplicaciones de mensajería instantánea Telegram y WhatsApp se utilizaron las aplicaciones de los dispositivos móviles. Tanto Telegram

como WhatsApp cambian el formato del contenedor multimeda del vídeo compartido a MP4. Con las herramientas de edición no se pretendía manipular el contenido en sí, sino ver las marcas que dejan dichos programas en el contenedor multimedia. En Adobe Premiere y Camtasia solo se abrieron los vídeos y se guardaron como copia para observar las operaciones que realizaban sobre el vídeo sin realizar ninguna manipulación sobre el mismo. A pesar que Adobe Premiere mantiene el formato original del vídeo sí realiza cambios sobre el contenedor multimedia. El programa Lightworks no permite abrir el vídeo sino que realiza una importación del vídeo que exporta en un formato MP4.

5.2. Análisis del algoritmo de extracción de átomos

Para evaluar la cantidad de información que extrae el algoritmo propuesto en la Sección 4.2.1 se ha extraído la información contenida en un vídeo original y las versiones de éste descargados de las 7 redes sociales y 3 programas de edición utilizados. Se utilizaron las dos herramientas más populares para la extracción de información de los contenedores multimedia (*MP4 File Parser* y *ExifToolGUI*). Esta última es además, la herramienta más utilizada por los analistas forenses.

La Figura 5.1 muestra un ejemplo de la información que extrae el algoritmo propuesto del contenedor multimedia de vídeos. Como se ve en la salida de nuestra aplicación se van mostrando los átomos y sub átomos, con sus correspondientes etiquetas y valores como si fueran un árbol, tenemos un diccionario de átomos que realiza una comprobación de los átomos y sub átomos para que no se repitan dentro del mismo árbol y les va asignando una numeración aumentando su valor en 1 cada vez que se repite, según lo que estipula el estándar del contenedor MOV.

```
trak
    edts
        elst    numberOfEntries:    1
                ediListTable:       [{mediaRate 65536, trackDuration 10468, mediaTime 0}]
                version:            0
                flags:              0
                type:               elst
    mdia
        mdhd    language:           21956
                quality:            0
                creationTime:       14 abr 2018 15:37:16 UTC
                modificationTime:   14 abr 2018 15:37:34 UTC
                duration:           17.4467
                version:            0
                flags:              0
                type:               mdhd
        minf
                dinf
                    dref    numberOfEntries:    1
                            version:            0
                            flags:              0
                            alis    type:       alis
                                    value:      1
                            type:   dref
                    type:   dinf
⋮
trak1
```

Figura 5.1: Valores etiquetas extraídas

En la Tabla 5.3 se muestra la salida ya procesada en formato .csv, esta salida tiene una serie de columnas que corresponden a los datos que se obtienen del vídeo estudiado con los cuales vamos a realizar el análisis, la columna *Ruta-origen* representa la ruta desde donde se lee el vídeo, la columna *Nombre-fichero* representa el nombre del archivo de vídeo, la columna *Marca* representa el nombre del fabricante del vídeo, la columna *Modelo* representa el nombre del modelo, la columna *Ruta-etiqueta* muestra la ruta completa donde se encuentra ubicado el átomo en la estructura del contenedor, las columna *Valor* muestran todas la información almacenada y por último la columna *Orden* representa el orden de aparición del átomo en la estructura del contenedor.

Tabla 5.3: Ejemplo de los resultados de la extracción del algoritmo propuesto

Ruta-origen	Nombre-fichero	Marca	Modelo	Ruta-etiqueta	Valor	Orden
/home/video/	Video14.mov	Apple	iPad2	/moov/trak3/edts/type	edts	110
/home/video/	Video14.mov	Apple	iPad2	/moov/trak3/edts/elst/type	elst	111
/home/video/	Video14.mov	Apple	iPad2	/moov/trak3/edts/elst/version	0	111
/home/video/	Video14.mov	Apple	iPad2	/moov/trak3/edts/elst/flags	0	111
/home/video/	Video14.mov	Apple	iPad2	/moov/trak3/edts/elst/numberOfEntries	0	111
/home/video/	Video14.mov	Apple	iPad2	/moov/trak3/edts/elst/editListTable	[{mediaRate 65536, trackDuration 10468, mediaTime 0}]	111

En la Tabla 5.4 se muestran las etiquetas que se extraen en un vídeo obtenido de nuestro Dataset procesado por las tres herramientas comparadas (el algoritmo propuesto, *MP4 file parser* y *ExifToolGUI*). Como el algoritmo propuesto extrae todas las etiquetas existentes en el átomo extraído, en la columna *Algoritmo propuesto* se coloca solo el número de etiquetas que extrae de cada átomo. En las columnas *MP4 file parser* y *ExifToolGUI* se muestran las etiquetas que no se leen de cada herramienta del átomo extraído, salvo para el caso de los átomos mp4a y avc1 que muestra las etiquetas que si se leen. En la Tabla 5.5 se establecen las equivalencias de símbolos indicados en la Tabla 5.4.

Tabla 5.4: Comparación etiquetas de átomos por herramientas

Átomo	Algoritmo propuesto	MP4 File Parser	ExifToolGUI
/ftyp/	5	=	=
/mvhd/	19	reserved	flags reserved
/trak*/tkhd/	17	reserved_1 reserved_2 reserved_3 trackWidth trackHeight	flags reserved_1 reserved_2 reserved_3 alternativeGroup trackWidth trackHeight
/trak*/edts/elst/	6	=	-
/trak*/mdia/mdhd/	10	=	version flags quality
/trak*/mdia/hdlr/	11	componentType componentManufacturer componentFlags componentFlagsMask componentNameCount	version flags componentFlags componentFlagsMask componentNameCount
/trak/mdia/minf/smhd/	6	=	version flags reserved
/trak*/mdia/minf/hdlr/	11	-	version flags

Continua en la siguiente página

Tabla 5.4 – continuación de la página anterior

Átomo	Aplicación	MP4fileparser	ExifToolGUI
			componentFlags componentFlagsMask componentNameCount
/trak*/mdia/minf/dinf/dref/	5	flags numberOfEntries	-
/trak*/mdia/minf/dinf/dref/alis/	5	-	-
/trak*/mdia/minf/stbl/stsd/	5	=	-
/trak/mdia/minf/stbl/stsd/mp4a/	17	reserved dataReferenceIndex numOfChannels sampleSize sampleRate	numOfChannels sampleSize sampleRate
/trak/mdia/minf/stbl/stsd/mp4a/chan/	5	-	-
/trak/mdia/minf/stbl/stsd/mp4a/wave/frma	3	-	-
/trak/mdia/minf/stbl/stsd/mp4a/wave/mp4a	3	-	-
/trak/mdia/minf/stbl/stsd/mp4a/wave/esds	4	-	-
/trak/mdia/minf/stbl/stsd/mp4a/wave/zero	3	-	-
/trak*/mdia/minf/stbl/stts	6	=	-
/trak*/mdia/minf/stbl/stsc	6	entries	-
/trak*/mdia/minf/stbl/stsz	7	=	-
/trak*/mdia/minf/stbl/stco	6	entries	-
/trak1/tapt/clef	6	=	version flags
/trak1/tapt/prof	6	=	version flags
/trak1/tapt/enof	6	=	version flags
/trak1/mdia/minf/vmhd/	6	=	version flags
/trak1/mdia/minf/stbl/stsd/avc1/	20	stsd_tmcd version revisionLevel dataSize	mediaWidth mediaHeight horizontalResolution verticalResolution

Continua en la siguiente página

Tabla 5.4 – continuación de la página anterior

Átomo	Aplicación	MP4fileparser	ExifToolGUI
		sizeCompressorName colorTableID	CompressorName depth
/trak1/mdia/minf/stbl/stsd/avc1/avcC/	3	-	-
/trak1/mdia/minf/stbl/stsd/avc1/colr/	6	-	-
/trak1/mdia/minf/stbl/ctts	6	entries	-
/trak1/mdia/minf/stbl/cslg	8	=	-
/trak1/mdia/minf/stbl/stss	6	entries	-
/trak1/mdia/minf/stbl/sdtp	5	-	-
/trak2 y 3/tref/	3	-	-
/trak2 y 3/mdia/minf/gmhd/gmin/	9	stsd_tmcd	stsd_tmcd flags reserved
/trak2 y 3/mdia/minf/stbl/stsd/mebx/.../	**	-	Solo lo menciona
/meta/hdlr/	8	=	-
/meta/keys/mdta.../	***	***	***
/meta/ilst/data.../	***	***	***
/wide/	3	=	-
/mdat/	3	=	-

Tabla 5.5: Equivalencias

Símbolo	Significado
=	Extrae las mismas etiquetas que el algoritmo propuesto
-	La aplicación no lee ninguna etiqueta en comparación con la nuestra
*	Corresponde a los 4 átomos trak del vídeo
**	En el caso de nuestra aplicación para el caso del átomo trak 2 se encuentra los átomos keys, lidp y lidl, asimismo dentro de keys los átomos local_key_id1 al local_key_id5 y de cada uno de estos, los átomos keyd, dtyp, sdpd y dentro de éste a su vez el átomo sdpi, también el átomo ctps y dentro de éste el átomo dtyp, para el caso del trak 3 se encuentran el átomo keys y dentro el átomo local_key_id1 y dentro los átomos keyd, dtyp, sdpd y dentro de éste el átomo sdpi.
***	Donde ambos átomos toman valores de 0 al 5, los cuales se corresponden uno con otro, para el caso de la aplicación ExifToolGUI se muestra como información antes de leer los trak y en el caso de la aplicación MP4 file parser los lee en el user data.

Como se observa, la herramienta *ExifToolGUI* muestra algunas etiquetas que para ellos son relevantes pero sin distinguir a que átomo corresponden, esto se debe a que no muestra la etiqueta type de ningún átomo, que es la que determina el nombre del átomo, mientras que la aplicación

MP4 file parser muestra las etiquetas en un árbol con su correspondiente átomo, pero no llega a la precisión de nuestra aplicación, además la aplicación ExifToolGUI tampoco muestra la etiqueta size que determinan el tamaño del átomo.

En la Tabla 5.6 se muestran las etiquetas que se extraen en un vídeo obtenido de una cámara de seguridad procesado por las tres herramientas comparadas (el algoritmo propuesto, *MP4 file parser* y *ExifToolGUI*). Como el algoritmo propuesto extrae todas las etiquetas existentes en el átomo extraído, en la columna *Algoritmo propuesto* se coloca solo el número de etiquetas que extrae de cada átomo. En las columnas *MP4 file parser* y *ExifToolGUI* se muestran las etiquetas que no se leen de cada herramienta del átomo extraído, salvo para el caso del átomo pads que muestra las etiquetas que si se leen.

Como se observa, la herramienta *ExifToolGUI* muestra algunas etiquetas que para ellos son relevantes pero sin distinguir a que átomo corresponden, esto se debe a que no muestra la etiqueta type de ningún átomo, que es la que determina el nombre del átomo, mientras que la aplicación MP4 file parser muestra las etiquetas en un árbol con su correspondiente átomo, pero no llega a la precisión de nuestra aplicación, además la aplicación ExifToolGUI tampoco muestra la etiqueta size que determinan el tamaño del átomo.

Como se ha observado en las Tablas 5.4 y 5.6 el algoritmo propuesto extrae todas las etiquetas presentes en los átomos del contenedor, tanto las etiquetas de control como las de datos. Cosa que no ocurre con las herramientas *MP4 file parser* y *ExifToolGUI*. La importancia del análisis de estas etiquetas radica en que pueden verse modificadas por herramientas de índole maliciosa. Especialmente etiquetas como reserved y flags de vídeos MP4 son manipuladas mediante herramientas de esteganografía en vídeos como OpenPuff que utiliza átomos de vídeo y de audio para ocultar información de forma segura. Por consiguiente, mediante el análisis de los átomos de control se puede detectar el uso de esteganografía en un vídeo determinado. Además, se puede extraer el contenido del átomo padre de esta etiqueta anómala para su posterior análisis. Estas sutiles características pueden vincular la herramienta con la que se produjo dicha manipulación. Los resultados obtenidos en [SHC15] demuestran que al explotar estas modificaciones a los metadatos de un archivo, se identifica con éxito la existencia de contenido oculto.

Tabla 5.6: Comparación etiquetas de átomos por herramientas

Átomo	Aplicación	MP4fileparser	ExifToolGUI
/ftyp/	5	=	=
/mvhd/	19	reserved	flags reserved
/trak/tkhd/	17	reserved_1 reserved_2 reserved_3 trackWidth trackHeight	flags reserved_1 reserved_2 reserved_3 alternativeGroup trackWidth trackHeight
/trak/mdia/mdhd/	10	=	version flags quality
/trak/mdia/hdlr/	10	componentType componentManufacturer, componentFlags componentFlagsMask componentNameCount	version flags componentFlags componentFlagsMask componentNameCount
/trak1/mdia/minf/vmhd/	6	=	version flags
/trak/mdia/minf/dinf/dref/	5	flags numberOfEntries	-
/trak/mdia/minf/dinf/dref/url/	5	-	-
/trak/mdia/minf/stbl/co64	6	=	-
/trak/mdia/minf/stbl/pads	8	type syze	-
/trak/mdia/minf/stbl/stsc	6	=	-
/trak/mdia/minf/stbl/stsd	5	=	-
/trak/mdia/minf/stbl/stsz	7	entries	-
/trak/mdia/minf/stbl/stts	6	entries	-
/trak/mdia/minf/stbl/stsd/jpeg/	20	-	Solo lo menciona
/mdat/	2	=	-

5.3. Análisis de la Estructura del Contenedor

El primer análisis tiene como objetivo evaluar si los diferentes fabricantes implementan correctamente la especificación. Se analizan tanto la estructura del contenedor como el orden en el que se almacenan los átomos en tres niveles: Por Marca, por modelo de cada fabricantes y por cada uno de los dispositivos recolectados por modelo.

5.3.1. Análisis de la Estructura del Contenedor de los Vídeos Originales a Nivel de Dispositivos

En este análisis se compararon los modelos que tienen vídeos recolectados de mínimo dos dispositivos diferentes. Esto con el fin de evaluar cómo influyen los diferentes componentes del dispositivo en la estructura del contenedor. Como se observa en la Tabla 5.1, los modelos que cumplen con esta condición son: MO_01, MO_03, MO_05, MO_06 y MO_07. Por tanto, se toman como referencia los vídeos del formato MOV.

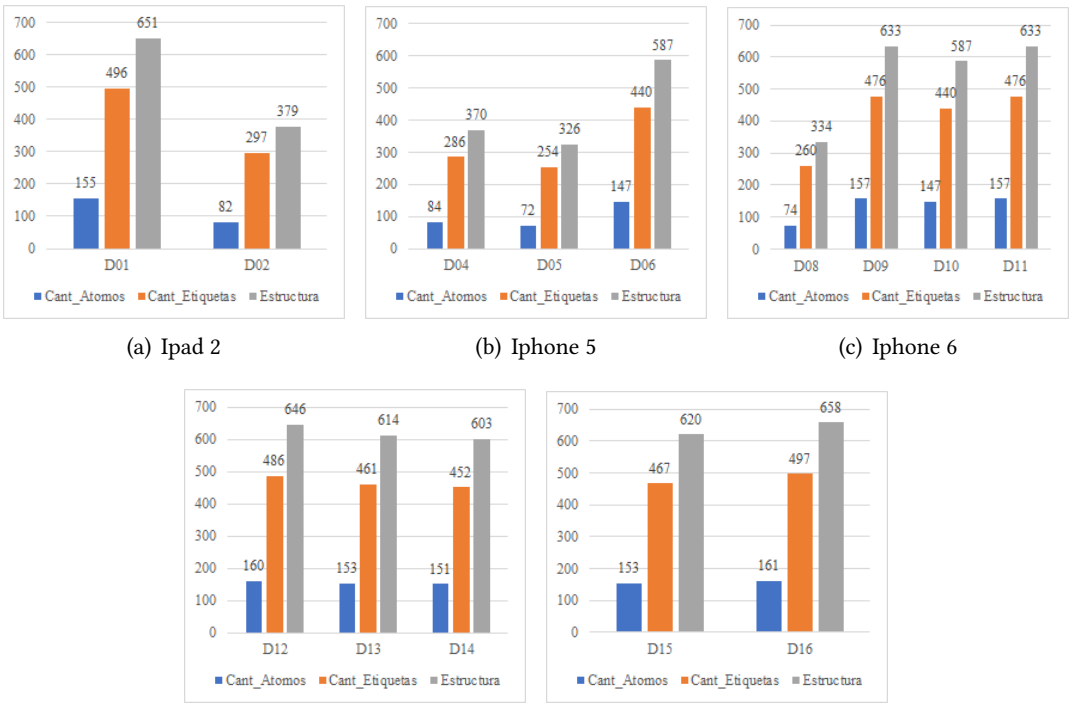
En la Figura 5.2 se muestran el número de átomos y etiquetas presentes en el contenedor de los dispositivos de cada modelo. Estos datos se consolidan en una estructura que representa el dispositivo. La estructura es definida como la suma de los diferentes átomos y etiquetas presentes en el contenedor multimedia de un vídeo.

Como se observa en la Figura 5.2(a) los vídeos de los dispositivos D01 y D02 tienen 651 y 379 elementos en la estructura del contenedor. Unificando los elementos de la estructura de cada dispositivo se obtiene una estructura consolidada de 684 elementos para el modelo IPAD 2. Se observa que 33 elementos de la estructura consolidada no los contiene D01 y 305 no se encuentran en D02. Estos elementos corresponden a los átomos *meta* y *udta* con sus respectivos subátomos en el caso de D01, y a los átomos *free1*, *trak2* y *trak3* con sus respectivos subátomos en el caso de D02. Análogamente, los resultados observados en los modelos restantes (Iphone 5, Iphone 6, Iphone 7 y Iphone 8) indican que:

Los dispositivos del modelo Iphone 5 (Figura 5.2(b)) se diferencian entre sí por la presencia de un átomo *free* adicional dentro del átomo *meta* y un átomo adicional *data4* y *mdta4* en D05 y la presencia del átomo *trak2* y *trak3* con sus subátomos en D06.

En el caso del Iphone 6 (Figura 5.2(c)), además de presentarse diferencias similares a las del Iphone 5 se presentan unos átomos adicionales en D10 (*data4*, *mdta4* y *udta*).

En la estructura en el Iphone 7 (Figura 5.2(d)) se observan diferencias por la presencia de los siguientes átomos: *cslg*, *ctts*, *sdtp*, *avc1*, *stss*, *vmhd*, *tapt*, *smhd*, *mp4a* y *udta*. En este modelo se detectaron inconsistencias en los átomos *free* presentes en D12: Aparece el átomo *free1* dentro del átomo *moov* y otro átomo *free2* dentro del átomo *meta*. Finalmente, la estructura consolidada del Iphone 8 (Figura 5.2(e)). Las diferencias radican como en los modelos anteriores por la presencia de los átomos *free1* y *free2* dentro de los átomos *meta* y *udta*.



(a) Ipad 2 (b) Iphone 5 (c) Iphone 6

(d) Iphone 7 (e) Iphone 8

Figura 5.2: Evaluación de la estructura del contenedor en dispositivos MOV

La Figura 5.3 muestra la estructura resultante de todos los dispositivos originales. En ella se observa una diferencia sustancial entre los modelos de marca Apple y los dispositivos de los demás fabricantes Samsung, Huawei y Xiaiomi (D17 a D23). Estas diferencias demuestran que la estructura del contenedor facilita las tareas de reconocimiento de la fuente de adquisición. Asimismo, se observa que Apple almacena el doble de información en el contenedor multimedia de sus dispositivos que el resto de fabricantes.

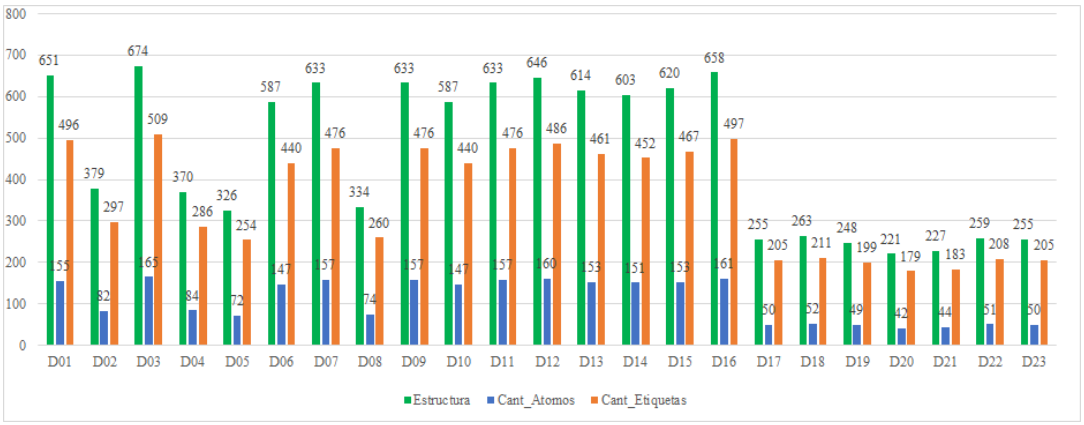
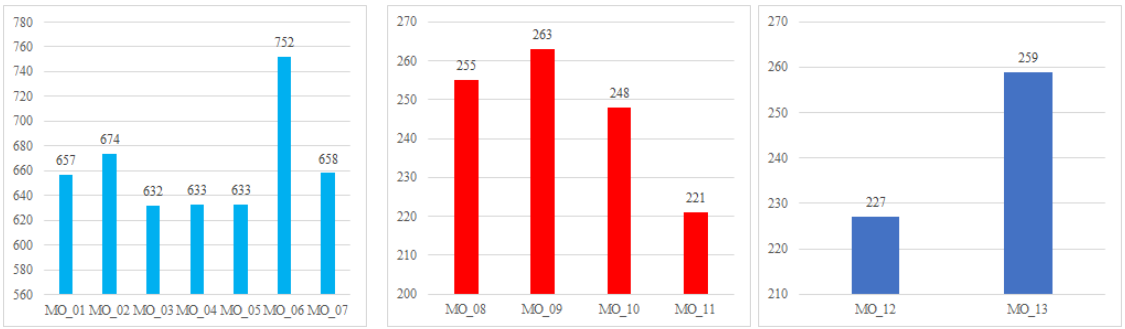


Figura 5.3: Resumen de la estructura del contenedor de los dispositivos de los vídeos originales

5.3.2. Análisis de la Estructura del Contenedor de los Vídeos Originales a Nivel de Modelos

Para realizar este análisis se consolidó la estructura de todos los dispositivos de cada modelo dejando aquellos átomos y etiquetas comunes en todos los dispositivos. Los resultados del análisis por modelo de los fabricantes Apple, Samsung y Huawei se muestran en la Figura 5.4.



(a) Apple (b) Samsung (c) Huawei

Figura 5.4: Evaluación de la estructura del contenedor por modelos

Las principales diferencia entre las estructuras consolidadas de los modelos del fabricante Apple (Figura 5.4(a)) se debe al no seguimiento estricto de la especificación en los siguientes átomos: El átomo *meta* y *udta* que almacenan metadatos e información del usuario contiene elementos *key*, *data* y *mdta* adicionales. Átomos *free* adicionales ubicados dentro de otros átomos (*meta* o *moov*) no coincidiendo con lo indicado en la especificación. Átomos *smhd*, *cslg*, *ctts*, *mp4a*, *sdtp*, *avc1*, *stss*, *vmhd* y *tapt* relacionados con el *trak* de audio.

A diferencia del fabricante Apple, los fabricantes Samsung y Huawei contienen un 34 % menos de elementos en la estructura consolidada de sus modelos. En el caso de Samsung (Figura 5.4(b)) la diferencia entre las estructuras consolidadas radica principalmente en los átomos *colr*, *pasp* y *meta*, en el que hay poca información de metadatos a lo sumo tienen un átomo *key1*, *data1* y *mdta1*. Por su parte, las estructuras consolidadas de los dos modelos analizados del fabricante Huawei, mostrados en la Figura 5.2(a), siguen el mismo patrón del fabricante Samsung solo contienen dos átomos *trak* uno para audio y otro para vídeo. Las estructuras no poseen información de metadatos en los átomos *meta*, sino que están en el átomo *udta*. Adicionalmente, se observa la existencia del átomo *stco* dentro de los átomos *trak* y *trak1*.

Para medir cuán asociadas se encuentran las variables entre sí y tomando en consideración los procedimientos descritos en la Sección 4.2.3. En la Tabla 5.7 se muestra la matriz de correlación por modelos (solo modelos de dispositivos que generan vídeos con formato MOV).

Tabla 5.7: Matriz de correlación por modelos (MOV)

	Ipad 2	Ipad Air	Iphone 5	Iphone 5S	Iphone 6	Iphone 7	Iphone 8 Plus
Ipad2	1.000	0.390	0.862	0.862	0.865	0.264	0.566
IpadAir		1.000	0.520	0.520	0.515	0.655	0.186
Iphone5			1.000	0.782	0.785	0.231	0.504
Iphone5S				1.000	0.997	0.558	0.733
Iphone6					1.000	0.550	0.729
Iphone7						1.000	0.193
Iphone8Plus							1.000

Se aprecia claramente que existe una correlación positiva perfecta solo entre los mismos modelos. Esto significa que ninguno de los 7 modelos posee características idénticas. No obstante, los modelos Iphone 5S y Iphone 6 alcanzaron una correlación positiva muy fuerte (0.997), lo que denota que ambos modelos generan vídeos con una estructura muy similar, pero mantienen una mínima diferencia que permite identificar el modelo de un vídeo dado. Además, de los 7 modelos, Ipad 2 alcanza una correlación positiva media con tres modelos (Iphone 5, Iphone 5S y Iphone 6). En cambio, los modelos que logran una correlación positiva débil son: Ipad Air con Iphone 8 Plus, Iphone 7 con Iphone 8 Plus, esto quiere decir que tienen una mayor diferencia entre ellos. En general, ningún modelo analizado se parece a otro en términos de la estructura del contenedor. Las diferencias sean mínimas o considerables son suficientes para determinar el modelo del dispositivo.

Asimismo, en la Tabla 5.8 se presenta la matriz de correlación por modelos (solo modelos que generan vídeos con formato MP4). Al igual que el análisis anterior, se observa que hay una relación lineal positiva muy fuerte solo entre los mismos modelos. No obstante, la relación entre la gran mayoría de modelos es relativamente más baja, lo que significa que la identificación exacta del modelo en vídeos con fomato MP4 es más previsible ya que existe mayor diferencia de características entre ellas. Un claro ejemplo es la correlación negativa débil que alcanzan los modelos AscendGU10-Galaxy S9 Plus (-0.103) y Galaxy S9 Plus-P9 (-0.057).

Tabla 5.8: Matriz de correlación de dispositivos MP4

	Ascend GU10	Galaxy A6	GalaxyJ5 2016	GalaxyS9 Plus	Galaxy TabA	P9	Redmi Note5
AscendGU10	1.000	0.153	0.315	-0.103	0.694	0.473	0.153
GalaxyA6		1.000	0.854	0.775	0.575	0.381	0.897
GalaxyJ52016			1.000	0.662	0.674	0.663	0.764
GalaxyS9Plus				1.000	0.446	-0.057	0.646
GalaxyTabA					1.000	0.438	0.507
P9						1.000	0.495
RedmiNote5							1.000

5.3.3. Análisis de la Estructura del Contenedor de los Vídeos Originales a Nivel de Fabricantes

En el análisis a nivel de fabricantes, se observa en la Figura 5.5 que el fabricante Apple (M_01) contiene una estructura con un 68 % de elementos adicionales que los 3 fabricantes restantes. Los átomos más relevantes y diferenciadores entre fabricantes, son los siguientes: En Samsung, los átomos *pasp*, *esds*, *colr* y *url* dentro de los átomos *trak* y *trak1*. Huawei además de los átomos anteriores se encuentra el átomo *co64* dentro del átomo *trak* y el átomo *trak1*. Análogamente, el fabricante Xiaomi incluye el átomo *free* además de los átomos presentes en Huawei.

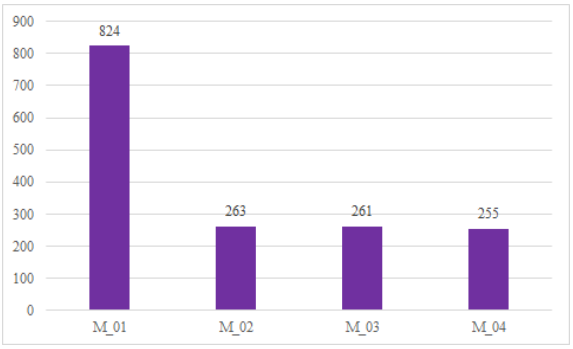
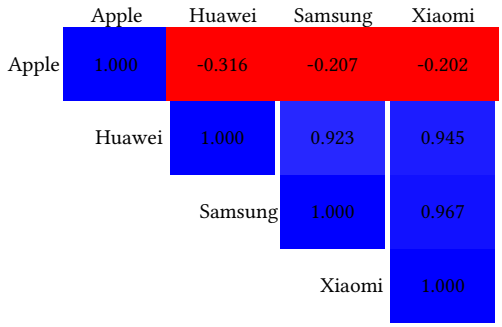


Figura 5.5: Evaluación de la estructura del contenedor por fabricantes

En la Tabla 5.9 se muestra la matriz de correlación por marcas, donde se aprecia que existe una correlación positiva perfecta entre las mismas marcas. No obstante, la marca Huawei tiene un relación positiva muy fuerte tanto con Samsung (0.923) y Xiaomi (0.945). De igual forma Samsung y Xiaomi alcanzan una correlación muy alta (0.967), lo que quiere decir que estos modelos tienen pocas diferencias entre sí, pero son lo suficientemente necesarios para detectar la marca del

dispositivo. Por otro lado, la marca Apple alcanza una correlación negativa débil con Huawei (-0.316), Samsung (-0.207) y Xiaomi (-0.202). Esto significa que Apple tiene una estructura del contenedor muy diferente a las tres marcas y eso permite distinguirlo mucho más fácil.

Tabla 5.9: Matriz de correlación por Marcas

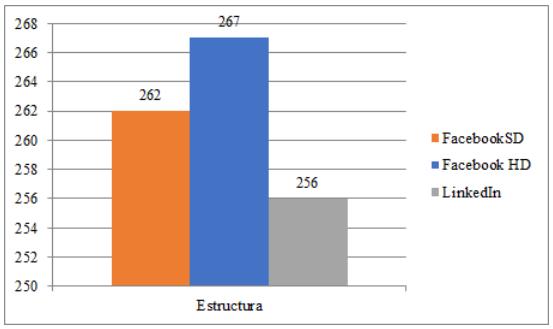


5.3.4. Análisis de la Estructura del Contenedor de los Vídeos Procesados por Redes Sociales y Editores de Vídeos

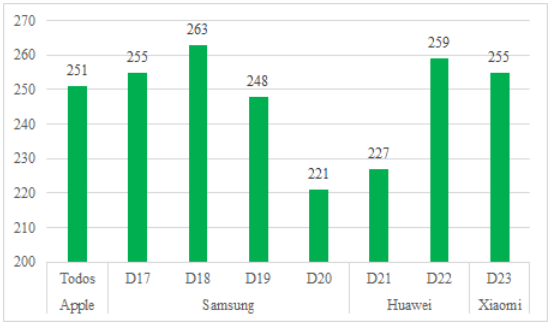
Una vez analizada la estructura de los vídeos originales, se procedió a estudiar los cambios que realizaban las diferentes redes sociales y herramientas de edición de vídeos. Para ello, se utilizaron en el análisis los vídeos procesados por redes sociales y herramientas de edición de vídeos mostradas en la Tabla 5.1 con las características de descarga de la Tabla 5.2. Los resultados obtenidos se muestran en la Figura 5.6.

La Figura 5.6(a) muestra que las redes sociales Facebook en sus formatos SD y HD y LinkedIn modifican uniformemente la estructura del contenedor independientemente de la marca o modelo del mismo, aunque los átomos que componen la estructura varíe. Como se observa, FacebookHD es el que mantiene el mayor número de átomos de la estructura original, esto se debe a la presencia del átomo day y sus subátomos. Asimismo, FacebookSD tiene dos átomos que no están presentes en los vídeos de LinkedIn y sus respectivos subátomos.

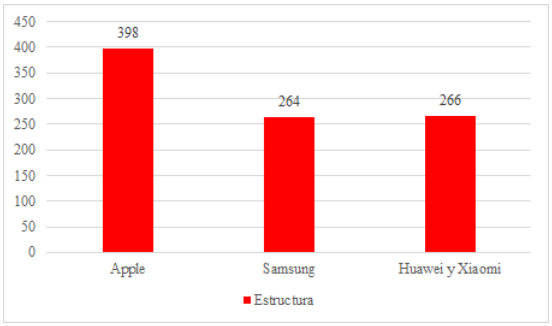
Flikr no hace ningún cambio sobre el contenedor de los vídeos que se comparte en esta red social. Algo muy parecido realizan Telegram y Adobe Premiere sobre los vídeos que manipulan. Por un lado Telegram solo modifica sustancialmente los vídeos provenientes de dispositivos del fabricante Apple a los que les cambia el formato de MOV a MP4. Sin embargo cuando los vídeos tienen formato MP4 no les realiza ninguna modificación (Ver Figura 5.6(b)). Por su parte, Adobe Premiere unifica el contenedor multimedia por Fabricante como se puede ver en la Figura 5.6(c). Finalmente, los resultados presentados en la Figura 5.6(d) muestra que los cambios realizados por la redes sociales Youtube y WhatsApp y las herramientas de edición Camtasia y Lightworks en los vídeos manipulados a nivel de estructura dependen de si el formato del vídeo es MOV o MP4.



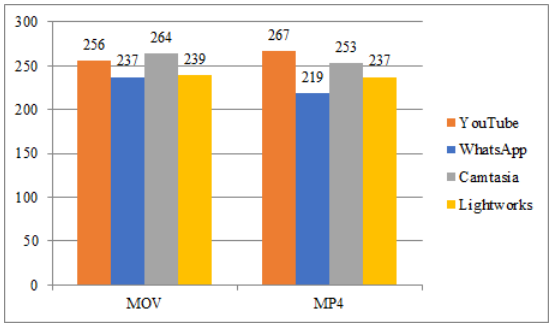
(a) FacebookSD, FacebookHD y LinkedIn



(b) Telegram



(c) Adobe Premiere



(d) YouTube, WhatsApp, Camtasia y Lightworks

Figura 5.6: Evaluación de la estructura del contenedor en redes sociales y programas de edición de vídeos

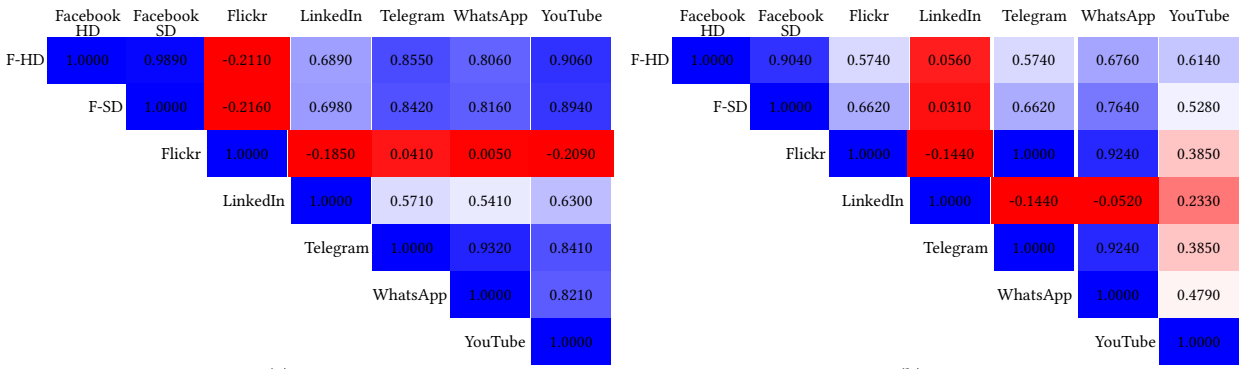
Más detalladamente se encontró lo siguiente:

- Tanto la aplicación Facebook HD, Facebook SD y LinkedIn, añade el átomo nam y el átomo too, además de tratar el campo fecha con el siguiente formato 01/01/1904 0:00:00.
- La aplicación Telegram en el átomo colr cambia el valor nclc por nclx en el caso de MOV, como fecha utiliza la de envío del mensaje para el caso de MOV y la fecha del vídeo en sí para el caso de MP4.
- La aplicación WhatsApp añade el átomo beam, en el átomo colr cambia el valor nclc por nclx en el caso de MOV, como fecha utiliza la de envío del mensaje para el caso de MOV y la fecha del vídeo en sí para el caso de MP4.
- La aplicación YouTube añade el átomo gsst, el átomo gstd y el átomo too, en el átomo data dentro del átomo too genera el valor Google, para el caso de las fechas utiliza la fecha de subida a la plataforma.
- La aplicación Adobe específicamente para Apple genera el átomo load, el átomo dvc, el átomo fiel y el átomo sowt. Para MP4 genera el átomo uuid.
- Como valores propios Adobe específicamente para Apple genera en el átomo ftyp en la etiqueta minorVersion el valor 537199360, en el átomo hdlr dentro del átomo minf dentro del átomo trak y trak1 tanto para Apple como para MP4, en el trak2 solo lo genera para

Apple, en la etiqueta componentName el valor Apple Alias Data Handler, en el átomo dvc en la etiqueta CompressorName genera el valor DV25 NTSC para Apple y VC Coding para MP4, en el átomo hdlr dentro del átomo trak2 en la etiqueta componentName el valor Time Code Media Handler solo para Apple y en el átomo udta guarda valores propios del editor tanto para Apple como para MP4.

- La aplicación Camtasia específicamente para Apple genera el átomo iods, el átomo ART, el átomo desc y el átomo nam. Tanto para Apple como para MP4 genera el átomo Xtra y el átomo uuid.
- La aplicación Lightworks específicamente para Apple genera el átomo iods.
- Como valores propios Camtasia y Lightworks específicamente para Apple genera en el átomo ftyp en la etiqueta majorBrand el valor mp42 y en la etiqueta compatibleBrands el valor isommp42, en el átomo avc1 en la etiqueta CompressorName el valor VC Coding tanto para Apple como para MP4.

A continuación, se calculó el coeficiente de correlación de la estructura resultante de los vídeos con formato MOV y MP4 compartidos por las 7 redes en las redes sociales. Las matrices de correlación de la Figura 5.7 muestran los resultados obtenidos.



(a) MOV (b) MP4

Figura 5.7: Matriz de correlación de vídeos con formato MOV y MP4 compartidos en redes sociales

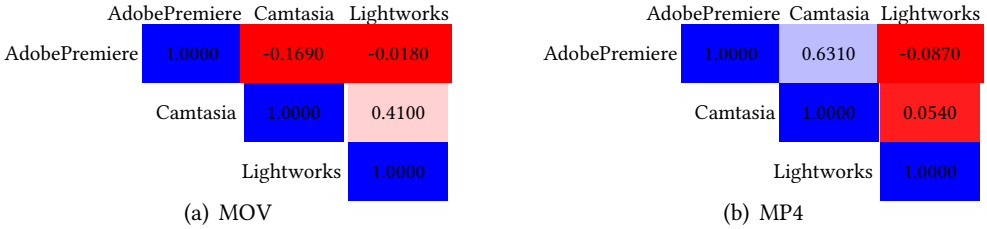
Como se aprecia en la Figura 5.7(a) la red social Flickr alcanza una correlación negativa débil con Facebook HD (-0.2110), Facebook SD (-0.2160), LinkedIn (-0.1850) y YouTube (-0.2090). Además, logra una correlación positiva débil con Telegram (0.0410) y WhatsApp (0.0050). De este resultado se puede concluir que si se analizan las 7 redes sociales, es más fácil identificar a Flickr debido a la gran diferencia de sus características frente a las demás redes sociales. Sin embargo, hay redes sociales que alcanzan una correlación positiva muy fuerte y su identificación se ve reducida por la similitud de sus características en el contenedor (Facebook HD y Facebook SD), (Facebook HD y YouTube), (Telegram y WhatsApp).

Análogamente, en la matriz de correlación de vídeos MP4 presentada en la Figura 5.7(b), se observa que a diferencia de los vídeos compartidos con formato MOV, la red social LinkedIn es la

que tiene mayor diferencia entre las demás, concretamente logra una correlación negativa débil tanto con Flickr (-0.1440), Telegram (-0.1440) y WhatsApp (-0.0520). Asimismo, de las 7 redes sociales, Youtube es la red social que logra una correlación positiva débil con Flickr, LinkedIn, Telegram y WhatsApp. En general, todas las redes sociales tienen los elementos necesarios que permiten distinguirlos mediante este tipo de análisis.

La Figura 5.8 muestra las matrices de correlación de las aplicaciones de edición de vídeos para vídeos con formato MOV y MP4.

En la Figura 5.8(a), se observa que el editor de vídeo Adobe Premiere alcanza una correlación negativa débil con Camtasia (-0.1690) y Lightworks (-0.0180). Asimismo, Camtasia tiene una relación positiva débil con Lightworks. Estos resultados confirman que los tres editores se diferencian notablemente entre sí, lo que facilita su identificación. De manera similar, los resultados de la Figura 5.8(b) muestran que los tres editores se diferencian considerablemente. En resumen, todo vídeo manipulado con estas tres editores conservan características unicas que ayudan al analista forense determinar su origen.



(a) MOV (b) MP4

Figura 5.8: Matriz de correlación de vídeos con formato MOV y MP4 manipulados con editores

5.4. Análisis del Orden de Almacenamiento de los Átomos

El orden en que se almacenan los átomos es otro factor que puede ser una característica importante en el análisis forense de vídeos. Con este fin, se analizó el orden de aparición de los átomos en los vídeos originales con formato MOV y MP4 y, posteriormente, se analizaron los vídeos resultantes de los procesos aplicados por las redes sociales y programas de edición (ver Tablas 5.10 y 5.11). En las tablas solo se muestran los átomos que están almacenados en un orden diferente en alguno de los modelos analizados.

Como se observa en la Tabla 5.10, el orden de los átomos se mantiene constante en la mayoría de los modelos de los vídeo originales salvo en los modelos Iphone7 y Iphone8Plus donde varía el orden de aparición de los átomos /wide/ y /mdat/, asimismo se observa que, a excepción de los modelo IpadAir y Iphone7 que no tienen átomo free, los modelos de Apple tienen más de un átomo free contenido en otros átomos (/moov/free/, /moov/free1 y /moov/met/free/). Al analizar los vídeos compartidos por redes sociales y manipulados con herramientas especializadas en

edición se observa los siguientes cambios relevantes:

- Flikr mantiene el contenedor intacto al no realizar ningún cambio en los vídeos.
- Las redes sociales y programas de edición que cambian el formato del vídeo a MP4, por el lado de las redes sociales Facebook en su versión SD y HD, LinkedIn, YouTube, Telegram y WhatsApp, asimismo por el lado de los editores de vídeo Camtasia y Lightworks eliminan los átomos wide y meta.
- Los vídeos compartidos por las redes sociales Facebook y LinkedIn y editados por Adobe Premier agrupan todos los átomos free en el átomo /free/.
- Las aplicaciones de mensajería Telegram y Whatsapp y el editor de vídeos Lighyworks solo mantienen los átomos que permiten la reproducción del vídeo.

Análogamente, en la Tabla 5.11, se observa que los vídeos originales con formato MP4 sufren menos cambios al ser compartidos que los vídeo con formato MOV. Al igual que Flikr, Telegram tampoco hace ninguna modificación en los vídeos MP4 compartidos.

Tabla 5.10: Posición originales MOV

	Vídeos Originales							Redes Sociales							Editores		
Átomo	**Ipad2**	**IpadAir**	**Iphone5**	**Iphone5S**	**Iphone6**	**Iphone7**	**Iphone8Plus**	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**E1**	**E2**	**E3**
/wide/	2	2	2	2	2	152	2, 152	-	-	2, 152	-	-	-	-	76	-	-
/mdat/	3	3	3	3	3	153	3, 153	54	55	3, 153	53	47, 49, 50	2	46	77	54	47
/moov/udta/	136	-	60	136	136	153	140	45	46	60, 136, 140, 153	46	41	-	-	74	42	-
/free/	-	-	-	-	-	-	-	53	54	-	52	-	-	-	75	-	-
/moov/free/	137	-	61	137	137	-	141	-	-	61, 137, 141	-	-	-	-	-	-	-
/moov/free1/	151	-	75	153	153	-	157	-	-	75, 151, 153, 157	-	-	-	-	-	-	-
/moov/meta/free/	150	-	74	152	152	-	156	-	-	74, 150, 152, 156	-	-	-	-	-	-	-
/moov/meta/	138	136	62	138	138	136	136, 142	-	-	62, 136, 138, 142	-	-	-	-	-	-	-

Tabla 5.11: Posición originales Mp4

Átomo	**Vídeos Originales**							**Redes Sociales**							**Editores**		
	Samsung				**Huawei**		**Xiaomi**										
	A6	**S9Plus**	**J5**	**TabA**	**Ascend**	**P9**	**RedmiNote5**	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**E1**	**E2**	**E3**
/free/	-	-	-	-	43	48	49	53	54, 55	43, 48, 49	52	-	-	43, 48, 49	51	51	-
/mdat/	2	2	2	2	44	49	50	54	2, 55, 56	2, 44, 49, 50	53	52, 54	2, 42	2, 44, 49, 50	52	53	47
/moov/meta/	6	6	6	-	-	4	4	6	6	4, 6	-	-	6	4, 6	-	-	-
/moov/udta/	5	5	5	5	4	-	-	45	46	4, 5	46	44	5	4, 5	-	-	-

5.5. Análisis a Nivel de Contenido de Átomos

La información almacenada en los átomos es otro factor que puede ser una característica importante en el análisis forense de vídeos. Con este fin, se analizó el contenido de los átomos en los vídeos originales con formato MOV y MP4 y, posteriormente, se analizaron esos mismo átomos en los vídeos resultantes de los procesos aplicados por las redes sociales y programas de edición.

5.5.1. Contenido de los Átomos del Contenedor de los Vídeos Originales

En la Tabla 5.12 se muestra el valor de lectura de los átomos de vídeos originales para el caso de MOV y MP4, para el caso de la etiqueta majorBrand con valor qt para el caso de los dispositivos MOV, con valor mp42 para el caso de los dispositivos MP4 menos el dispositivo Ascend con valor isom, para el caso de la etiqueta compatibleBrands con valor qt para el caso de los dispositivos MOV, con valor isommp42 para el caso de los dispositivos MP4 menos el dispositivo Ascend con valor isom3gp4, para el caso de la etiqueta timeScale dentro del átomo mvhd con valor 600 para el caso de los dispositivos MOV, con valor 1000 para el caso de los dispositivos MP4, para el caso de la etiqueta timeScale dentro del átomo mdhd con valor 44100 para el caso de los dispositivos MOV, con valor 90000 para el caso de los dispositivos MP4, para el caso de la etiqueta opColor con valor [32768,32768,32768] para el caso de los dispositivos MOV, con valor [0,0,0] para el caso de los dispositivos MP4, para el caso de la etiqueta graphicsMode con valor 64 para el caso de los dispositivos MOV, con valor 0 para el caso de los dispositivos MP4, para el caso de la etiqueta componentName con valor Core Media Video para el caso de los dispositivos MOV, con valor VideoHandle para el caso de los dispositivos MP4, para el caso de la etiqueta trackWidth con valor 568 en particular el Ipad2, con valor 1280 en particular el Ipad2, el TabA y el Ascend, con valor 1920 en particular el IpadAir, el Iphone5, el Iphone5S, el Iphone6, el Iphone7, el Iphone8Plus, el A6, el J5 y el P9, con valor 3840 en particular el Iphone7 y el S9Plus, para el caso de la etiqueta trackHeight con valor 120 en particular el TabA, con valor 320 en particular el Ipad2, con valor 720 en particular el Ipad2 y el Ascend, con valor 1080 en particular el IpadAir, el Iphone5, el Iphone5S, el Iphone6, el Iphone7, el Iphone8Plus, el A6, el J5, el P9 y el RedmiNote5, con valor 2160 en particular el Iphone7 y el S9Plus, para el caso de la etiqueta componentName con valor Core Media Audio para el caso de los dispositivos MOV, con valor SoundHandle para eongl caso de los dispositivos MP4.

Tabla 5.12: Contenido originales MOV y MP4

Átomo	Valor	Dispositivo
/ftyp/majorBrand	qt	D02, D03, D05, D07, D09, D13 y D15
	mp42	D17, D18, D19, D20, D22 y D23
	isom	D21
/ftyp/compatibleBrands	qt	D02, D03, D05, D07, D09, D13 y D15
	isommp42	D17, D18, D19, D20, D22 y D23
	isom3gp4	D21
/mvhd/timeScale	600	D02, D03, D05, D07, D09, D13 y D15
	1000	D17, D18, D19, D20, D21, D22 y D23
/mdia/mdhd/timeScale	44100	D02, D03, D05, D07, D09, D13 y D15
	90000	D17, D18, D19, D20, D21, D22 y D23
/vmhd/opColor	[32768,32768,32768]	D02, D03, D05, D07, D09, D13 y D15
	[0,0,0]	D17, D18, D19, D20, D21, D22 y D23
/vmhd/graphicsMode	64	D02, D03, D05, D07, D09, D13 y D15
	0	D17, D18, D19, D20, D21, D22 y D23
/hdlr/componentName	Core Media Video	D02, D03, D05, D07, D09, D13 y D15
	VideoHandle	D17, D18, D19, D20, D21, D22 y D23
/tkhd/trackWidth	568	D02
	1280	D02, D20 y D21
	1920	D03, D05, D07, D09, D13, D15, D17, D19, D22
	3840	D13 y D18
/tkhd/trackHeight	120	D20
	320	D02
	720	D02 y D21
	1080	D03, D05, D07, D09, D13, D15, D17, D19, D22 y D23
	2160	D13 y D18
/mdia/hdlr/componentName	Core Media Audio	D02, D03, D05, D07, D09, D13 y D15
	SoundHandle	D17, D18, D19, D20, D21, D22 y D23

5.5.2. Contenido de los Átomos del Contenedor de los Vídeos compartidos por Redes sociales

En la Tabla 5.13 se muestra el valor de lectura de los átomos de vídeos de redes sociales de MOV y MP4, para el caso de la etiqueta majorBrand con valor qt para el caso de la red social Flickr, con valor isom para el caso de las redes sociales Facebook en sus versiones SD y HD, LinkedIn, Flickr, Telegram, WhatsApp y YouTube, con valor mp42 para el caso de las redes sociales Facebook en sus versiones SD y HD, LinkedIn, Flickr, Telegram, WhatsApp y YouTube, para el caso de la etiqueta compatibleBrands con valor qt para el caso de la red social Flickr, con valor isomiso2avc1mp41 para el caso de las redes sociales Facebook en sus versiones SD y HD y LinkedIn, con valor isommp42 para el caso de las redes sociales Flickr, Telegram, WhatsApp

y YouTube, con valor isom3gp4 para el caso de las redes sociales HDkr, Telegram y WhatsApp, para el caso de la etiqueta timeScale dentro del átomo mvhd con valor 600 para el caso de la red social Flickr, con valor 1000 para el caso de las redes sociales Facebook en sus versiones SD y HD, LinkedIn, Flickr, Telegram y YouTube, con valor 44100 para el caso de las redes sociales Telegram y WhatsApp, con valor 48000 para el caso de la red social WhatsApp, con valor 30000 para el caso de la red social YouTube y con valor Varios para el caso de las redes sociales WhatsApp y YouTube, para el caso de la etiqueta timeScale dentro del átomo mdhd con valor 600 para el caso de las redes sociales Telegram y WhatsApp, con valor 15360 para el caso de las redes sociales Facebook en su versión SD, LinkedIn y YouTube, con valor 44100 para el caso de las redes sociales Facebook en su versión HD, Flickr y YouTube, con valor 48000 para el caso de la red social LinkedIn, con valor 90000 para el caso de las redes sociales Facebook en su versiones SD y HD, LinkedIn, Flickr, Telegram y WhatsApp, con valor Varios para el caso de las redes sociales Facebook en su versión HD, WhatsApp y YouTube, para el caso de la etiqueta opColor con valor [32768, 32768, 32768] para el caso de la red social Flickr y con valor [0, 0, 0] para el caso de las redes sociales Facebook en sus versiones SD y HD, LinkedIn, Flickr, Telegram, WhatsApp y YouTube, para el caso de la etiqueta componentName con valor ISO Media file produced by Google Inc. Created on: 10/30/2018. para el caso de la red social YouTube, con valor VideoHandler para el caso de las redes sociales Facebook en sus versiones SD y HD y LinkedIn, con valor VideoHandle para el caso de las redes sociales Facebook en su versión HD, Flickr, Telegram y WhatsApp, con valor Vacío para el caso de la red social WhatsApp, para el caso de la etiqueta trackWidth con valor 320, 360, 568, 640, 720, 1280 y 1920 para el caso de la red social Facebook en su versión HD, con valor 224, 400 y 3840 para el caso de la red social Facebook en su versión SD, con valor 568, 1280, 1920 y 3840 para el caso de la red social Flickr, con valor 320, 404, 560, 848 y 1280 para el caso de la red social LinkedIn, con valor 320, 464, 560, 848, 1280, 1920 y 3840 para el caso de la red social Telegram, con valor 480, 576, 640, 848 y 3840 para el caso de la red social WhatsApp y con valor 202, 406, 568, 640, 1280 para el caso de la red social YouTube. Como se observa la etiqueta trackHeight con valor 320, 360, 568, 720, 1080, 1280 y 2160 para el caso de la red social Facebook en su versión HD, con valor 224, 400 y 2160 para el caso de la red social Facebook en su versión SD, con valor 320, 720, 1080, 1280 y 2160 para el caso de la red social Flickr, con valor 320, 568, 720 para el caso de la red social LinkedIn, con valor 320, 464, 560, 720, 848, 1080 y 2160 para el caso de la red social Telegram, con valor 320, 352, 480 y 2160 para el caso de la red social WhatsApp y con valor 320, 360, 720 para el caso de la red social YouTube, para el caso de la etiqueta componentName con valor SoundHandler para el caso de las redes sociales Facebook en sus versiones SD y HD y LinkedIn, con valor SoundHandle para el caso de las redes sociales Facebook en su versión HD, Flickr, Telegram y WhatsApp, con valor Core Media Audio para el caso de la redes sociales Flickr, Telegram y WhatsApp, con valor ISO Media file produced by Google Inc. Created on: 10/30/2018. para el caso de la red social YouTube y con valor Vacío para el caso de la red social WhatsApp.

Tabla 5.13: Contenido redes sociales MOV y MP4

Átomo	Valor	Red Social
/ftyp/majorBrand	qt	Flickr
	isom	FacebookSD, FacebookHD, Flickr, Linkedin, YouTube, Telegram y WhatsApp
	mp42	FacebookSD, FacebookHD, Flickr, Linkedin, YouTube, Telegram y WhatsApp
/ftyp/compatibleBrands	qt	Flickr
	isomiso2avc1mp41	FacebookSD, FacebookHD y Linkedin
	isommp42	Flickr, YouTube, Telegram y WhatsApp
	isom3gp4	Flickr, Telegram y WhatsApp
	mp41mp42isom	Telegram y WhatsApp
	mp42isom	Telegram y WhatsApp
/mvhd/timeScale	600	Flickr
	1000	FacebookSD, FacebookHD, Flickr, Linkedin, YouTube y Telegram
	44100	Telegram y WhatsApp
	48000	WhatsApp
	30000	YouTube
	Varios	YouTube y WhatsApp
/mdia/mdhd/timeScale	600	Telegram y WhatsApp
	15360	FacebookSD, LinkedIn y YouTube
	44100	FacebookHD, Flickr y YouTube
	48000	Linkedin
	90000	FacebookSD, FacebookHD, Flickr, Linkedin, Telegram y WhatsApp
	Varios	FacebookHD, WhatsApp y YouTube
/vmhd/opColor	[32768, 32768, 32768]	Flickr
	[0,0,0]	FacebookSD, FacebookHD, Flickr, Linkedin, YouTube, Telegram y WhatsApp
/hdlr/componentName	ISO Media file produced by Google Inc. Created on: 10/30/2018.	YouTube
	VideoHandler	FacebookSD, FacebookHD y Linkedin
	VideoHandle	FacebookHD, Flickr, Telegram y WhatsApp

Continua en la siguiente página

Tabla 5.13 – continuación de la página anterior

Átomo	Valor	Red Social
	Core Media Video	Flickr, Telegram y WhatsApp
	Vacío	WhatsApp
/tkhd/trackWidth	320, 360, 568, 640, 720, 1280 y 1920	FacebookHD
	224, 400 y 3840	FacebookSD
	568, 1280, 1920 y 3840	Flickr
	320, 404, 560, 848 y 1280	LinkedIn
	320, 464, 560, 848, 1280, 1920 y 3840	Telegram
	480, 576, 640, 848 y 3840	WhatsApp
	202, 406, 568, 640, 1280	YouTube
/tkhd/trackHeight	320, 360, 568, 720, 1080, 1280 y 2160	FacebookHD
	224, 400 y 2160	FacebookSD
	320, 720, 1080, 1280 y 2160	Flickr
	320, 568, 720	LinkedIn
	320, 464, 560, 720, 848, 1080 y 2160	Telegram
	320, 352, 480 y 2160	WhatsApp
	320, 360, 720	YouTube
/hdlr/componentName	SoundHandler	FacebookSD, FacebookHD y LinkedIn
	SoundHandle	FacebookHD, Flickr, Telegram y WhatsApp
	Core Media Audio	Flickr, Telegram y WhatsApp
	ISO Media file produced by Google	YouTube
	Inc. Created on: 10/30/2018.	
	Vacío	WhatsApp

5.5.3. Contenido de los Átomos del Contenedor de los Vídeos modificados por Editores de vídeo

En la Tabla 5.14 se muestra el valor de lectura de los átomos de vídeos de editores para MOV y MP4.

Tabla 5.14: Contenido editores MOV y MP4

Átomo	Valor	Editor
/ftyp/majorBrand	qt	Adobe
	mp42	Adobe, Camtasia y Lightworks
/ftyp/compatibleBrands	qt y mp42mp41	Adobe
	isommp42	Camtasia y Lightworks
/mvhd/timeScale	30000	Adobe
	90000	Adobe, Camtasia y Ligthworks
/mdia/mdhd/timeScale	[0,0,0]	Adobe, Camtasia y Lightworks
	[32768,32768,32768]	Adobe
/vmhd/graphicsMode	0	Adobe, Camtasia y Lightworks
	64	Adobe
/hdlr/componentName	Apple Video Media Handler y #Mainconcept	Adobe
	Video Media Handler	
	Mainconcept MP4 Video Media Handler	Camtasia
	Mainconcept MP4 Sound Media Handler	Lightworks
/tkhd/trackWidth	720, 1080, 1280, 1920	Adobe
	568, 608, 1920	Camtasia
	1280	Lightworks
/tkhd/trackHeight	480, 720, 1080, 1280, 1920	Adobe
	320, 1080	Camtasia
	720	Lightworks
/hdlr/componentName	Apple Sound Media Handler y	Adobe
	#Mainconcept MP4 Sound Media Handler	
	Mainconcept MP4 Sound Media Handler	Camtasia
	Mainconcept MP4 Video Media Handler	Lightworks

Para el caso de la etiqueta majorBrand con valor qt para el caso del editor Adobe, con valor mp42 para el caso de los editores Adobe, Camtasia y Lightworks, para el caso de la etiqueta compatibleBrands con valor qt y mp42mp41 para el caso del editor Adobe, con valor isommp42 para el caso de los editores Camtasia y Lightworks, para el caso de la etiqueta timeScale dentro del átomo mvhd con valor 30000 para el caso del editor Adobe, con valor 90000 para el caso de los editores Adobe, Camtasia y Lightworks, para el caso de la etiqueta timeScale dentro del átomo mdhd con valor [0,0,0] para el caso de los editores Adobe, Camtasia y Lightworks, con valor [32768,32768,32768] para el caso del editor Adobe, para el caso de la etiqueta graphicsMode con valor 0 para el caso de los editores Adobe, Camtasia y Lightworks, con valor 64 para el caso del editor Adobe, para el caso de la etiqueta componentName con valor Apple Video Media Handler para el caso del editor Adobe, con valor #Mainconcept Video Media Handler para el caso del editor Adobe, con valor Mainconcept MP4 Video Media Handler para el caso del editor Camtasia y con valor Mainconcept MP4 Sound Media Handler para el caso del editor Lightworks, para el caso de la etiqueta trackWidth con valor 720, 1080, 1280, 1920 para el caso del editor Adobe,

con valor 568, 608, 1920 para el caso del editor Camtasia y con valor 1280 para el caso del editor Lightworks, para el caso de la etiqueta trackHeight con valor 480, 720, 1080, 1280, 1920 para el caso del editor Adobe, con valor 320, 1080 para el caso del editor Camtasia y con valor 720 para el caso del editor Lightworks, para el caso de la etiqueta componentName con valor Apple Sound Media Handler y #Mainconcept MP4 Sound Media Handler para el caso del editor Adobe, con valor Mainconcept MP4 Sound Media Handler para el caso del editor Camtasia y con valor Mainconcept MP4 Video Media Handler para el caso del editor Lightworks.

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

En este trabajo se ha diseñado una herramienta para la extracción de átomos, etiquetas y metadatos de un vídeo digital con formato MOV, con sus correspondientes valores y orden de lectura, para dicho fin se han obtenido vídeos descargados de dispositivos en su versión original, luego estos mismos vídeos se han distribuido por las redes sociales y los editores de vídeo.

El algoritmo desarrollado extrae toda la información almacenada en el contenedor descrita en su estándar QTFF. Para tal fin vamos leyendo cada átomo y verificamos su tipo y tamaño, de manera recursiva vamos leyendo por si los átomos tienen sub átomos, se cuenta con un diccionario de átomos que va comprobando que no existe duplicidad y que los átomos cumplen la especificación del estándar, si no es así se guardan como átomos desconocidos para luego verificarlo, nuestra salida es un dataframe con todas las características del vídeo para luego realizar el estudio de la información.

Se ha realizado un estudio de la información sobre la identificación de la fuente para poder tratar y reflejar dicha información en nuestra herramienta, cabe destacar que existe muy poca literatura que se base en el estudio del contenedor MOV para tareas de análisis forense de vídeos, la mayoría de la literatura esta orientada a imágenes.

En el caso de la experimentación, se consolida la información obtenida por nuestra herramienta, esta consolidación se basa en la estructura, inicialmente se han analizado los vídeos en formato original de los dispositivos, realizando un análisis de los aspectos principales de los vídeos digitales, hallando diferencias entre dispositivos del mismo modelo, pero con versiones de software distintas.

Seguidamente se han analizado todos los dispositivos comparando con los vídeos con formato MP4 al ser el formato más usado en el campo de los dispositivos móviles. Posteriormente, se han

analizado por modelos y hemos finalizado haciendo un estudio por marcas como son Apple, Samsung, Huawei y Xiaomi.

A continuación, se han analizado los vídeos generados por las redes sociales como Facebook en sus dos versiones SD y HD, LinkedIn, Flickr y YouTube, aplicaciones de mensajería como Telegram y WhatsApp, además de los vídeos generados por editores de vídeo como Adobe Premiere, Camtasia y Lightworks. Finalmente, todos estos resultados obtenidos se han consolidado para determinar la diferencia entre los vídeos originales con los vídeos generados por redes sociales y editores de vídeo, se han encontrado características propias que definen a cada uno de ellos, analizando el orden de lectura en el que aparecen y los valores propios generados por cada una de las aplicaciones.

6.2. Trabajo Futuro

Se han establecido las siguientes columnas como investigaciones a realizar después de la culminación del presente trabajo:

- El análisis forense de vídeos es un campo en el que queda todavía mucho por investigar.
- Determinar de manera fidedigna que un vídeo puede haber sido manipulado, revisando dentro de sus características y logrando identificarlo de manera óptima.
- La herramienta de este proyecto, se creo con el fin de analizar vídeos pero aún le falta para llegar a ser una buena herramienta para el análisis forense, queda pendiente ampliar sus funciones hasta lograr dicho objetivo.
- Determinar que información es relevante y descartar la que no sea necesaria, para que de este modo podamos optimizar el reconocimiento para el análisis forense de vídeos.
- Obtener nueva información con respecto al presente trabajo que sirva para la investigación de nuevos formatos de vídeo, así como para ir actualizando los que ya tenemos, para no quedar desfasados y poder cumplir de manera excelente con nuestro análisis.

Capítulo 7

Introduction

In [Zif18] it is indicated that it is no secret that the most popular camera in the world today is the telephone. In parallel, independent camera sales have plummeted. Apple's emphasis on impressive new camera features on its latest iPhones has underscored the question of whether it's only a matter of time before the phone really owns the general-purpose camera market. Some things to improve on mobile device cameras would be lenses, sensors, high dynamic range HDR functionality, and zoom or telephoto. Given the massive investment in phones, it's only a matter of time before they replace every segment of the camera market they are physically capable of.

In [Inc19] it is established that the images and videos used in different applications, social networks such as YouTube, Facebook, LinkedIn, WhatsApp and Telegram, in addition to video editors such as Adobe, are generated by cameras on mobile devices.

In [Ass19] it is described that at the end of 2018, as seen in Figure 7.1 5.1 billion people worldwide subscribed to mobile services, which represents 67 % of The world population. A total of one billion new subscribers have been added in the four years since 2013, representing an average annual growth rate of 5 %, but the rate of growth is slowing. An average annual growth rate of 1.9 % between 2018 and 2025 will bring the total number of mobile subscribers to 5.8 billion, 71 % of the population.

Of the 710 million people expected to subscribe to mobile services for the first time in the next seven years, half will come from the Asia Pacific region and just under a quarter will come from sub-Saharan Africa.

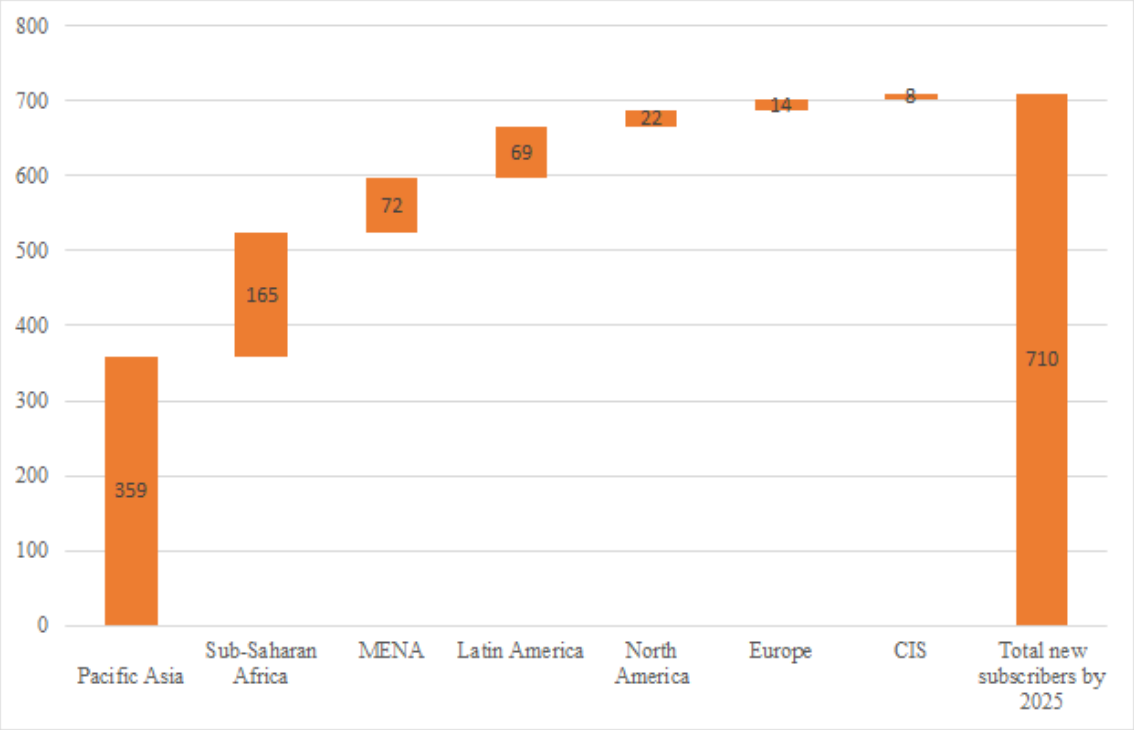


Figura 7.1: Subscribers by 2025

Meanwhile, mobile devices continue to make a significant contribution to socio-economic development worldwide. As seen in Figure 7.2 in 2018, mobile technologies and services generated $ 3.9 trillion of economic value 4.6 % of PIB worldwide, a contribution that will reach $4.8 trillion, 4.8 % of PIB by 2023, as countries increasingly benefit from improvements in productivity and efficiency brought about by increased use of mobile services. Going forward, 5G technologies are expected to contribute $ 2.2 trillion to the global economy in the next 15 years.

The connectivity gap also continues to close: Over the next seven years, 1.4 billion people will start using mobile internet for the first time, bringing the total number of mobile internet subscribers globally to 5 billion by 2025, more than 60 % of the population. This growth in connectivity is helping the mobile industry to increase its impact on all the United Nations Sustainable Development Goals and is driving the adoption of mobile-based tools and solutions, for example in agriculture, education and healthcare that They aim to improve livelihoods in middle-income countries, according to the report by the Global Communications System Association GSM.

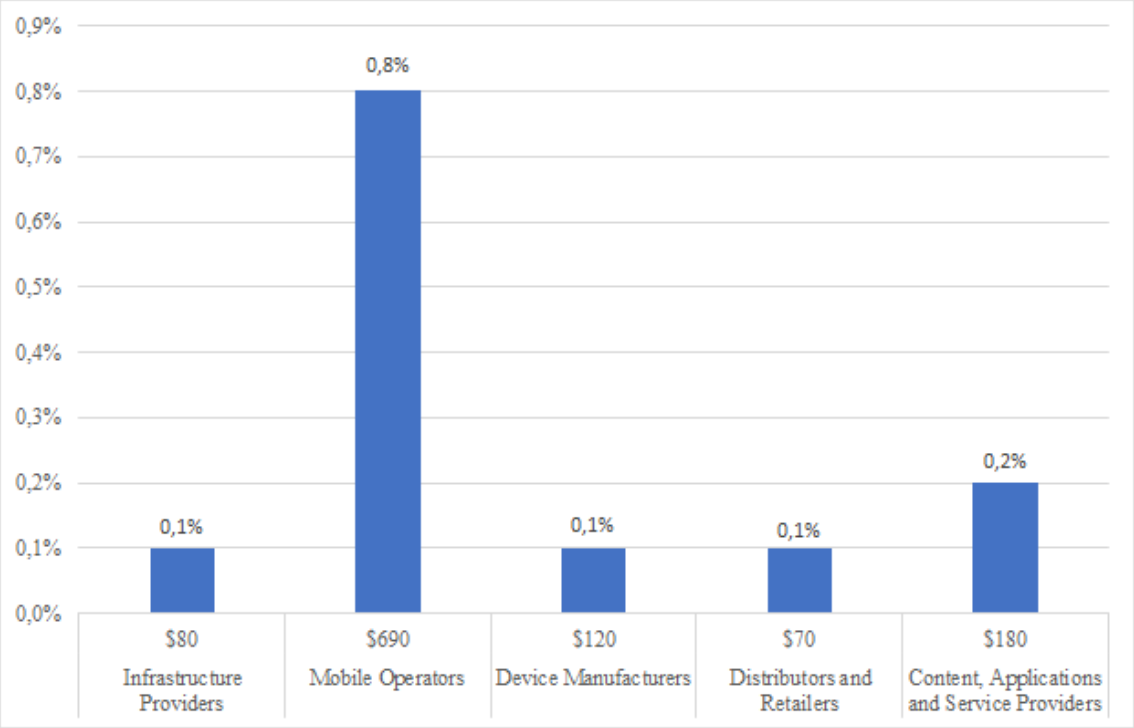


Figura 7.2: Economical mobile ecosystem

The Section 1.2 describes the object of the investigation of the present work. The Section 1.1 indicates the context of the investigation. Finally, in the Section 1.3 the rest of this work is presented as structured.

7.1. Research Purpose

Prepare a study of the structure of a digital video in general, the atoms that compose it and the extraction of metadata from digital videos in order to examine and understand the most notable ones. The following objectives are required for this:

- Analyze in detail the information storage specification of digital videos in MOV format.
- Design and implement an algorithm that allows automatic extraction of metadata and data stored in videos from mobile devices.
- Analyze the structure of the videos for later identification and differentiation with respect to the MP4 format.

7.2. Context

The present Final Master's Project is part of a research project entitled RAMSES approved by the European Commission within the Framework Program for Research and Innovation Horizon 2020 (Call H2020-FCT-2015, Innovation Action, Proposal Number: 700326) and in which the GASS Group of the Department of Software Engineering and Artificial Intelligence of the Faculty of Computing of the Complutense University of Madrid participates (Group of Analysis, Security and Systems, http://gass.ucm.es, group 910623 of the catalog of research groups recognized by the UCM).

In addition to the Complutense University of Madrid, the following entities participate:

- Treelogic Telemática y Lógica Racional para la Empresa Europea SL (España)
- Ministério da Justiça (Portugal)
- University of Kent (Reino Unido)
- Centro Ricerche e Studi su Sicurezza e Criminalità (Italia)
- Fachhochschule fur Offentliche Verwaltung und Rechtspflege in Bayern (Alemania)
- Trilateral Research & Consulting LLP (Reino Unido)
- Politecnico di Milano (Italia)
- Service Public Federal Interieur (Bélgica)
- Universitaet des Saarlandes (Alemania)
- Dirección General de Policía - Ministerio del Interior (Spain)

7.3. Structure of the Work

The present memory is divided into 7 chapters, this one as a whole is part of the first.

The Chapter 2 introduces essential explanations about videos, video compression systems, audio compression systems and the different multimedia containers.

The Chapter 3 presents the state of the art of forensic and antiphorean techniques related to the authentication of video content.

The chapter 4 presents the contributions of this work.The structure of the metadata, the algorithm of extraction of atoms and metadata and the technology used for this purpose.

The Chapter 5 explains the experimentation carried out in this work, obtaining the sample and presenting the results obtained for the videos without processing and with processing.

The Chapter 6 the product of the analysis and the lines of future research from this work are exhibited.

The Appendix A shows an annex with a description of the modules of the Multimedia Container MOV and screenshots of the ExifToolGUI and Mp4 Parser applications to some analyzed videos.

Capítulo 8

Conclusions and Future Work

8.1. Conclusions

In this work, a tool has been designed for the extraction of atoms, labels and metadata from a digital video in MOV format, with their corresponding values and reading order. For this purpose, videos downloaded from devices in their original version, then these same videos have been distributed by social media and video editors.

The developed algorithm extracts all the information stored in the container described in its standard QTFF. For this purpose we are reading each atom and verify its type and size, recursively we are reading in case the atoms have sub atoms, there is a dictionary of atoms that checks that there is no duplication and that the atoms meet the specification of the standard If this is not the case, they are stored as unknown atoms and then verified, our output is a dataframe with all the characteristics of the video, to then study the information.

A study of the information on the identification of the source has been carried out to be able to treat and reflect this information in our tool, it should be noted that there is very little literature that is based on the study of the container MOV for forensic analysis tasks of videos, most of the literature is image oriented.

An analysis of the information on the identification of the source has been carried out to be able to treat and reflect said information in our tool, it should be noted that there is very little literature that is based on the study of the container MOV for forensic analysis tasks. of videos, most of the literature is image oriented.

In the case of experimentation, the information obtained by our tool is consolidated, this consolidation is based on the structure, initially the videos in the original format of the devices were analyzed, conducting a study of the most important aspects of digital videos, finding differences between devices of the same model, but with different software versions.

Subsequently, all the devices have been analyzed, comparing with the videos in MP4 format, as this is the format most used by most mobile devices. Subsequently, they have been analyzed by model and we have finished doing a study by brands such as Apple, Samsung, Huawei and Xiaomi.

Next, the videos generated by social networks such as Facebook have been analyzed in their two versions SD and HD, LinkedIn, Flickr and YouTube, messaging applications such as Telegram and WhatsApp, in addition to the videos generated by video editors like Adobe Premiere, Camtasia and Lightworks. Finally, all these results obtained have been consolidated to determine the difference between the original videos and the videos generated by social networks and video editors. Their own characteristics have been found that define each of them, analyzing the reading order in which appear and the eigenvalues generated by each of the applications.

8.2. Future Work

The following columns have been established as investigations to be carried out after the completion of this work:

- Video forensics is a field in which there is still a lot to investigate.
- Accurately determine that a video may have been manipulated, reviewing its characteristics and optimally identifying it.
- The tool of this project, I think that in order to analyze videos but it still lacks to become a good tool for forensic analysis, it remains to expand its functions to achieve this objective.
- Determine what information is relevant and discard that which is not necessary, so that this mode requires recognition for the forensic analysis of videos.
- Describe new information regarding this work that serves to research new video formats, as well as to update those that we already have, so as not to be out of date and be able to comply excellently with our analysis.

Bibliografía

[10994] ISO/IEC 10918-1. Information Technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines. https://www.iso.org/standard/18902.html, Febraury 1994.

[11193] ISO/IEC 11172-2. Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s – Part 2: Video. https://www.iso.org/standard/22411.html, August 1993.

[AB16] J. Aghamaleki and A. Behrad. Malicious Inter-Frame Video Tampering Detection in MPEG Videos Using Time and Spatial Domain Analysis of Quantization Effects. *Multimedia Tools and Applications*, 76, October 2016.

[AB18] J. Aghamaleki and A. Behrad. Detecting Double Compressed MPEG Videos with the Same Quantization Matrix and Synchronized Group of Pictures Structure. *Journal of Electronic Imaging*, 27:1, February 2018.

[Ass19] GSM Association. GSMA. https://www.gsma.com/r/mobileeconomy/, 2019.

[BG15] A. Bidokhti and S. Ghaemmaghami. Detection of Regional Copy/Move Forgery in MPEG Videos Using Optical Flow. In *2015 The International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pages 13–17, 2015.

[BLH14a] S. Bian, W. Luo, and J. Huang. Detecting video frame-rate up-conversion based on periodic properties of inter-frame similarity. *Multimedia Tools and Applications*, 72, September 2014.

[BLH14b] S. Bian, W. Luo, and J. Huang. Exposing Fake Bit Rate Videos and Estimating Original Bit Rates. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(12):2144–2154, 2014.

[BRR11] Michael Bleyer, Christoph Rhemann, and Carsten Rother. PatchMatch Stereo - Stereo Matching with Slanted Support Windows. In *BMVC*, pages 14.1–14.11, January 2011.

[BVT+13] P. Bestagini, M. Visentini, M. Tagliasacchi, P. Dragotti, and S. Tubaro. Video Recapture Detection Based on Ghosting Artifact Analysis. In *2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings*, pages 4457–4461, September 2013.

[CB16] A. Costanzo and M. Barni. Detection of Double AVC/HEVC Encoding. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 2245–2249, August 2016.

[COF12] V. Conotter, J. O'Brien, and H. Farid. Exposing Digital Forgeries in Ballistic Motion. *IEEE Transactions on Information Forensics and Security*, 7(1):283–296, 2012.

[CTLH16] S. Chen, S. Tan, B. Li, and J. Huang. Automatic Detection of Object-Based Forgery in Advanced Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(11):2138–2151, 2016.

[DRG11] Z. Dias, A Rocha, and S. Goldenstein. Video phylogeny: Recovering near-duplicate video relationships. In *2011 IEEE International Workshop on Information Forensics and Security*, pages 1–6, 2011.

[DRUC+10] A. De Rosa, F. Uccheddu, A. Costanzo, A. Piva, and M. Barni. Exploring Image Dependencies: a New Challenge in Image Forensics. In *Media Forensics and Security II*, page 75410, February 2010.

[EA02] Japan Electronics and Information Technology Industries Association. Exchangeable Image File Format for Digital Still Cameras: EXIF Version 2.2. https://www.bibsonomy.org/bibtex/2ebd00c16dc2f30f091e4439ffb15ad54/dret, April 2002.

[Far08] H. Farid. Digital Image Ballistics from JPEG Quantization: A Followup Study. https://www.cs.dartmouth.edu/farid/downloads/publications/tr08.pdf, 2008.

[FCK13] J. Fan, H. Cao, and A. Kot. Estimating EXIF Parameters Based on Noise Features for Image Manipulation Detection. *IEEE Transactions on Information Forensics and Security*, 8(4):608–618, April 2013.

[Fou] Python Software Foundation. Python. https://www.python.org/.

[GCCF16] M. Goljan, M. Chen, P. Comesaña, and J. Fridrich. Effect of Compression on Sensor-Fingerprint Based Camera Identification. *Electronic Imaging*, 2016:1–10, February 2016.

[GFK14] T. Gloe, A. Fischer, and M. Kirchner. Forensic Analysis of Video File Formats. https://www.sciencedirect.com/science/article/pii/S1742287614000140, May 2014.

[GK14] A. Gloe, T.and Fischer and M. Kirchner. Forensic Analysis of Video File Formats. *Digital Investigation*, 11:S68 – S76, 2014.

[Glo12] T. Gloe. Forensic Analysis of Ordered Data Structures on the Example of JPEG Files. In *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 139–144, December 2012.

[Gro17] Group of Analysis, Security and Systems. Metada Video Extraction Tool. Technical Report, Faculty of Computer Science and Engineering, Universidad Complutense de Madrid, Spain, August 2017.

[HFC18] M. Harran, W. Farrelly, and K. Curran. A Method for Verifying Integrity & Authenticating Digital Media. *Applied Computing and Informatics*, 14(2):145 – 158, 2018.

[HIBS16] D. Hamdi, F. Iqbal, T. Baker, and B. Shah. Multimedia File Signature Analysis for Smartphone Forensics. In *2016 9th International Conference on Developments in eSystems Engineering (DeSE)*, pages 130–137, August 2016.

[HRLL13] D. Hyun, S. Ryu, H. Lee, and H. Lee. Detection of Upscale-Crop and Partial Manipulation in Surveillance Video Based on Sensor Pattern Noise. *Sensors (Basel) 2013*, 13(9):12605-12631, September 2013.

[Inc] Stack Exchange Inc. MP4Parser. https://stackoverflow.com/tags/mp4parser/info.

[Inc16] Apple Inc. QuickTime File Format Specification. https://developer.apple.com/library/archive/documentation/QuickTime/QTFF/QTFFChap1/qtff1.html, 2016.

[Inc19] Moz Inc. The Moz Top. https://moz.com/top500, 2019.

[ISF+19] M. Iuliani, D. Shullani, M. Fontani, S. Meucci, and A. Piva. A Video Forensic Framework for the Unsupervised Analysis of MP4-Like File Container. *IEEE Transactions on Information Forensics and Security*, 14(3):635–645, March 2019.

[JP09] J. A. Jódar and D. Polo. Distribución de Contenidos Audiovisuales en Internet. La Expansión de la Alta Definición y la Migración del FLV al MPEG-4. http://www.razonypalabra.org.mx/N/N70/JODAR%20Y%20POLO_REVISADO.pdf, 2009.

[KJF11] E. Kee, M. Johnson, and H. Farid. Digital Image Authentication From JPEG Headers. *IEEE Transactions on Information Forensics and Security*, 6(3):1066–1075, September 2011.

[KOS10] M. Kobayashi, T. Okabe, and Y. Sato. Detecting Forgery From Static-Scene Video Based on Inconsistency in Noise Level Functions. *IEEE Transactions on Information Forensics and Security*, 5(4):883–892, 2010.

[KYK19] M. Kim, C. Yoo, and Y. Ko. Multimedia File Forensics System Exploiting File Similarity Search. *Multimedia Tools and Applications*, 78(5):5233–5254, March 2019.

[LCC11] G. Lin, J. Chang, and C. Chuang. Detecting Frame Duplication Based on Spatial and Temporal Analyses. In *2011 6th International Conference on Computer Science Education (ICCSE)*, pages 1396–1399, 2011.

[LDL14] Z. Li, M. S. Drew, and J. Liu. *Fundamentals of Multimedia*. Springer Publishing Company, Incorporated, 2014.

[Lew12] A. Lewis. Reconstructing Compressed Photo and Video Data. Technical report, University of Cambridge, Computer Laboratory, February 2012.

[Lim19] Vcodex Limited. VCODEX. https://www.vcodex.com/historical-timeline-of-video-coding-standards-and-formats/, 2019.

[Ló17] B. López. Detección de Manipulación en Vídeos con Formato MPEG. FMP Thesis, Facultad de Estudios Estadísticos, Universidad Complutense de Madrid, Spain, 2017.

[MD04] S. J. Murdoch and M. Dornseif. Hidden Data in Internet Published Documents. http://cdn.media.ccc.de/congress/21C3/papers/271%20Hidden%20Data%20in%20Internet%20Published%20Documents.pdf, 2004.

[MFB+12] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro. An Overview on Video Forensics. *APSIPA Transactions on Signal and Information Processing*, 1:1229–1233, August 2012.

[MN15] W. Mambodza and A. NagoorMeeran. Android Mobile Forensic Analyzer for Stegno Data. In *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, pages 1–8, March 2015.

[MTW16] I. MaungMaung, Y. Tew, and K. Wong. Authentication of MP4 File by Joint Data Embedding in Audio and Video Tracks. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–6, December 2016.

[NAF+13] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez. Towards Fast Generic Video Inpainting. In *ACM International Conference Proceeding Series*, June 2013.

[Num] NumFOCUS. Pandas. https://pandas.pydata.org/.

[PM09] A. Pal and N. Memon. The Evolution of File Carving. *IEEE Signal Processing Magazine*, 26(2):59–71, March 2009.

[QEAL16] C. Quinto, Armas E., Sandoval A., and García L. Análisis de Metadatos en Vídeos Digitales de Dispositivos Móviles. *VIII Congreso Internacional de Computación y Telecomunicaciones 2016*, 09 2016.

[Qui12] L. Quintero. Formatos de Vídeo para Multimedia y Páginas Web. http://revista.ismm.edu.cu/index.php/revista_estudiantil/article/view/731/349, 2012.

[RDS10] C. Rodrigo, J. L. Delgado, and T. Sastre. Accesibilidad a los Contenidos Educativos Audiovisuales: Nuevas Tecnologías con Formatos Contenedores. http://revistas.uned.es/index.php/ried/article/view/820/729, 2010.

[RGN14] C. Richao, Y. Gaobo, and Z. Ningbo. Detection of Object-Based Manipulation by the Statistical Features of Object Contour. *Forensic Science International*, 236:164 – 169, 2014.

[Ric10] I. E. Richardson. *The H.264 Advanced Video Compression Standard.* Wiley Publishing, 2010.

[Rom13] P. Romero. Cine Digital y sus Contenedores. Herramientas de Software libre basadas en las Especificaciones DCI. FDP Thesis, Facultad de Informática, Euskal Herriko Unibertsitatea, País Vasco, June 2013.

[SA15] R. Singh and N. Aggarwal. Detection of re-compression, transcoding and frame-deletion for digital video authentication. In *2015 2nd International Conference on Recent Advances in Engineering Computational Sciences (RAECS)*, pages 1–6, 2015.

[SA17] Raahat Singh and Naveen Aggarwal. Video Content Authentication Techniques: A Comprehensive Survey. *Multimedia Systems*, February 2017.

[SHC15] T. Sloan and J. Hernandez-Castro. Steganalysis of OpenPuff Through Atomic Concatenation of MP4 Flags. *Digital Investigation*, 13, June 2015.

[SLLL16] J. Song, K. Lee, W. Lee, and H. Lee. Integrity Verification of the Ordered Data Structures in Manipulated Video Content. *Digital Investigation*, 18:1 – 7, 2016.

[SM08] H. Sencar and N. Memon. Overview of State-of-the-Art in Digital Image Forensics. *World Scientific*, 3, November 2008.

[SM14] H. Sencar and N. Memon. *Digital Image Forensics: There is More to a Picture Than Meets the Eye*. Springer Publishing Company, Incorporated, 2014.

[SNZ11] Y. Su, W. Nie, and C. Zhang. A Frame Tampering Detection Algorithm for MPEG Videos. In *2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference*, pages 461–464, 2011.

[S.R] JetBrains S.R.O. PyCharm. https://www.jetbrains.com/es-es/pycharm/.

[SX10] Y. Su and J. Xu. Detection of Double-Compression in MPEG-2 Videos. In *2010 2nd International Workshop on Intelligent Systems and Applications*, pages 1–4, 2010.

[Tea] Matplotlib Development Team. Matplotlib. https://matplotlib.org/.

[Tek15] A. M. Tekalp. *Digital Video Processing*. Prentice Hall Press, June 2015.

[Vel15] L. Velden. Forensic Devices for Activism: Metadata Tracking and Public Proof. *Big Data & Society*, 2(2), November 2015.

[VSD13] M. Visentini-Scarzanella and P. Dragotti. Modelling Radial Distortion Chains for Video Recapture Detection. In *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, pages 412–417, 2013.

[Wika] Wikipedia. Compresión de Audio. https://es.wikipedia.org/wiki/Compresi%C3%B3n_de_audio.

[Wikb] Wikipedia. EXIF. https://en.wikipedia.org/wiki/ExifTool.

[Wik19] Wikipedia. Códec de Audio. https://es.wikipedia.org/wiki/C%C3%B3dec_de_audio, 2019.

[WLZM14] Q. Wang, Z. Li, Z. Zhang, and Q. Ma. Video Inter-Frame Forgery Identification Based on Consistency of Correlation Coefficients of Gray Values. *Journal of Computer and Communications*, 02:51–57, January 2014.

[Yan15] G. Yang. Detecting Video Frame-Rate Up-Conversion Based on Periodic Properties of Edge-Intensity. *Journal of Information Security and Applications*, December 2015.

[YXL+16] G. Yang, M. Xia, L. Li, R. Li, and S. Xingming. Detecting Video Frame Rate Up-Conversion Based on Frame-Level Analysis of Average Texture Variation. *Multimedia Tools and Applications*, 76:8399–8421, February 2016.

[Zif18] D. Ziff. ExtremeTech. https://www.extremetech.com/electronics/277457-will-phones-soon-finish-off-the-camera-market, 2018.

Apéndice A

Anexos

A.1. Descripción de los átomos del Contenedor Multimedia MOV

En este capítulo se describe los átomos principales del contenedor multimedia MOV que son objeto de nuestro estudio, que se basan en el estándar QTFF, en [Inc16] se describe el estándar en su totalidad.

A.1.1. Átomo Ftyp

El átomo de compatibilidad de tipo de archivo, también llamado átomo de tipo de archivo permite al lector determinar si se trata de un tipo de archivo que el lector entiende. Específicamente, el tipo de archivo atom identifica las especificaciones de tipo de archivo con las que el archivo es compatible. Esto permite al lector distinguir entre tipos de archivos estrechamente relacionados, como archivos de película QuickTime, archivos MPEG-4 y JPEG-2000 (todos los cuales pueden contener átomos de tipo de archivo, átomos de película y átomos de datos de película).

Cuando un archivo es compatible con más de una especificación, el tipo de archivo atom lista todos los tipos compatibles e indica la marca preferida o el mejor uso entre los tipos compatibles. Por ejemplo, un reproductor de música que utilice un formato de archivo compatible con QuickTime podría identificar el mejor uso de un archivo como archivo de música para ese reproductor, pero también identificarlo como una película de QuickTime.

El átomo del tipo de archivo sirve además para distinguir entre diferentes versiones o especificaciones del mismo tipo de archivo, permitiéndole transmitir más información que la extensión de archivo o el tipo MIME. El átomo de tipo de archivo también tiene la ventaja de ser interno al archivo, donde está menos sujeto a una alteración accidental que una extensión de archivo o un tipo MIME. El tipo de archivo atom es opcional, pero se recomienda encarecidamente. Si está presente, debe ser el primer átomo significativo en el archivo, precediendo al átomo de película (y cualquier átomo de espacio libre, átomo de vista previa o átomo de datos de película).

El Átomo tipo de archivo tiene un valor de tipo de átomo de 'ftyp' y contiene los siguientes campos:

1. **Tamaño** Un entero sin signo de 32 bits que especifica el número de bytes en este átomo.

2. **Tipo** Un entero sin signo de 32 bits que identifica el tipo de átomo, típicamente representado como un código de cuatro caracteres; Este campo debe estar definido como 'ftyp'.

3. **Major_Brand** Un entero sin signo de 32 bits que debe establecerse en 'qt' (observe los dos caracteres de espacio ASCII) para archivos de película QuickTime. Si un archivo es compatible con varias marcas, todas estas marcas aparecen en los campos Compatible_Brands y la Major_Brand identifica la marca preferida o el mejor uso.

4. **Minor_Version** Un campo de 32 bits que indica la versión de especificación de formato de archivo. Para archivos de película QuickTime, esto toma la forma de cuatro valores decimales codificados en binario, que indican el siglo, el año y el mes de la especificación de formato de archivo de QuickTime, seguido de un código decimal binario codificado.

5. **Compatible_Brands** Una serie de enteros sin signo de 32 bits listando formatos de archivo compatibles. La marca principal debe aparecer en la lista de marcas compatibles. Se permiten una o más entradas de "marcador de posición" con valor cero; tales entradas deben ser ignoradas.

   Si ninguno de los campos Compatible_Brands está establecido en 'qt', el archivo no es un archivo de película QuickTime y no es compatible con la especificación.

A.1.2. Átomo Free o Skip

Tanto los átomos free como skip designan el espacio no utilizado en el archivo de datos de la película. Estos átomos constan sólo de un encabezado de átomo (campos de tamaño y tipo), seguido del número apropiado de bytes de espacio libre. Al leer una película de QuickTime, su aplicación puede saltar con seguridad estos átomos. Al escribir o actualizar una película, puede reutilizar el espacio asociado con estos tipos de átomos.

Un átomo wide suele preceder a un átomo de datos de película. El átomo wide consiste sólo en un campo de tipo y tamaño. Esto ocupa 8 bytes de espacio suficiente para agregar un campo de tamaño extendido a la cabecera del átomo que sigue, sin desplazar el contenido de ese átomo. Si un átomo crece hasta superar los 2^32 bytes de tamaño y está precedido por un átomo de wide, puede crear un nuevo encabezado de átomo que contenga un campo de tamaño extendido sobrescribiendo el encabezado de átomo existente y el átomo de wide.

A.1.3. Átomo Mdat

Al igual que con los átomos libres y omitidos, el Átomo de datos de película está estructurado de forma muy simple. Consiste en un encabezado de átomo (tamaño de átomo y campos de tipo), seguido de los datos de medios de la película. Su aplicación sólo puede comprender los datos de este átomo utilizando los metadatos almacenados en el átomo de la película. Este átomo puede ser bastante grande y puede exceder de 2ˆ32 bytes, en cuyo caso el campo de tamaño se establecerá en 1 y el encabezado contendrá un campo de tamaño extendido de 64 bits.

A.1.4. Átomo Pnot

Los Átomos de vista previa contienen información que le permite encontrar la imagen de vista previa asociada a una película de QuickTime. La imagen de vista previa, o póster, es una imagen representativa adecuada para mostrar al usuario en, por ejemplo, fotogramas de diálogo Abrir. La Figura A.1 muestra el diseño del átomo de vista previa.

Preview atom	**Bytes**
Atom size	4
Type = 'pnot'	4
Modification date	4
Version number	2
Atom type	4
Atom index	2

Figura A.1: Estructura de un grupo de imágenes

El átomo de vista previa tiene un valor de tipo de átomo de 'pnot' y, siguiendo su encabezado de átomo, contiene los siguientes campos:

1. **Tamaño** Un entero de 32 bits que especifica el número de bytes en este átomo de vista previa.

2. **Tipo** Un entero de 32 bits que identifica el tipo de átomo; este campo se establece en 'pnot'.

3. **Fecha de modificación** Un entero sin signo de 32 bits que contiene una fecha que indica cuándo se actualizó por última vez la vista previa.

4. **Número de versión** Un entero de 16 bits que debe establecerse en 0.

5. **Tipo de átomo** Un entero de 32 bits que indica el tipo de átomo que contiene los datos de vista previa. Normalmente, se establece en 'PICT' para indicar una imagen QuickDraw.

6. **Índice de átomos** Un entero de 16 bits que identifica qué átomo del tipo especificado se va a utilizar como la vista previa. Normalmente, este campo se establece en 1 para indicar que debe utilizar el primer átomo del tipo especificado en el campo de tipo de átomo.

A.1.5. Estructura de Metadatos

El contenedor de metadatos es un átomo de tipo "meta". El átomo de metadatos debe contener los siguientes subátomos: átomo controlador de metadatos ("hdlr"), claves de elementos de metadatos átomo ("claves") y átomo de lista de elementos de metadatos ("ilst"). Otros átomos opcionales que pueden estar contenidos en un átomo de metadatos incluyen el átomo de la lista de países ("ctry"), el átomo de la lista de idiomas ("lang") y el átomo de espacio libre ("libre"). Los átomos de la lista de países y la lista de idiomas se pueden usar para almacenar datos localizados de manera eficiente. El átomo de espacio libre se puede usar para reservar espacio en un átomo de metadatos para adiciones posteriores, o para poner a cero los bytes dentro de un átomo de metadatos después de editar y eliminar elementos de él. El átomo del espacio libre no puede ocurrir dentro de ningún otro subátomo contenido en el átomo de metadatos. El átomo de metadatos es el contenedor para transportar metadatos.

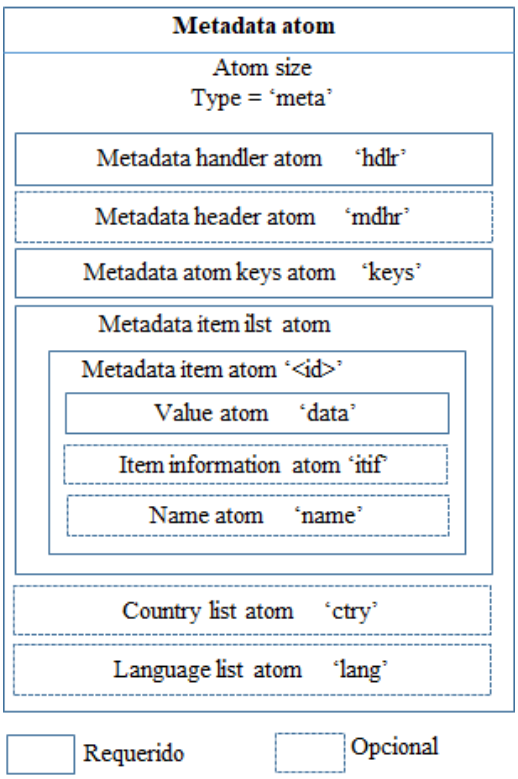


Figura A.2: Muestra de un átomo de metadatos y subátomos

A.1.6. Átomo Hdlr

El átomo del controlador de metadatos es un átomo completo con un tipo de átomo de "hdlr". Define la estructura utilizada para todos los tipos de metadatos almacenados dentro del átomo de metadatos.

El diseño del átomo del controlador de metadatos se define:

- **Tamaño** Un entero sin signo de 32 bits que indica el tamaño en bytes de la estructura del átomo.
- **Tipo** Un valor entero sin signo de 32 bits establecido en 'hdlr'
- **Versión** Un byte que se establece en 0
- **Banderas** Tres bytes que se establecen en 0
- **Predefinido** Un entero de 32 bits que se establece en 0
- **Tipo de controlador** Un entero de 32 bits que indica la estructura utilizada en el átomo de metadatos, establecida en "mdta"
- **Reservado** Una matriz de 3 enteros de 32 bits sin signo const establecidos en 0
- **Nombre** El nombre es una cadena terminada en NULL en caracteres UTF-8 que proporciona un nombre legible para un tipo de metadatos, con fines de depuración e inspección. La cadena puede estar vacía o un solo byte que contenga 0.

Nota: Un lector que analiza un átomo de metadatos debe confirmar que el tipo de controlador en el átomo de controlador de metadatos es "mdta" antes de interpretar cualquier otra estructura en el átomo de metadatos de acuerdo con la especificación presentada aquí. Si el tipo de controlador no es "mdta", la interpretación se define mediante otra especificación.

A.1.7. Átomo Mdhr

El formato de metadatos opcionalmente asigna identificadores únicos a los elementos de metadatos para propósitos tales como definir identificadores estables para referencias externas en el conjunto de elementos de metadatos. Esto se logra al incluir un átomo de información de elemento en los átomos de elemento de metadatos agregados contenidos por el átomo de la lista de elementos de metadatos. Se debe garantizar que dichos identificadores únicos sean únicos.

Para que la asignación de identificadores únicos de elementos sea más eficiente, el átomo de metadatos puede contener un átomo de encabezado de metadatos que contiene el valor entero para el siguiente identificador único de elementos que se asignará almacenado en el campo nextItemID. En general, tiene un valor uno mayor que el identificador más grande utilizado hasta ahora.

Importante: El átomo del encabezado de metadatos debe existir si hay átomos de elementos de metadatos que contienen un átomo de información del elemento que indica la ID única del elemento.

Al asignar el identificador a un elemento de metadatos, si el valor del campo nextItemID es menor que 0xFFFFFFFF, debe incrementarse al siguiente valor no utilizado. Si el valor de nextItemID es igual a 0xFFFFFFFF, no se debe cambiar: en ese caso, se necesita una búsqueda de un valor de identificador de elemento no utilizado en el rango de 0 a 0xFFFFFFFF para todas las adiciones.

El átomo del encabezado de metadatos es un átomo completo con un tipo de átomo de "mhdr". Contiene los siguientes campos:

- **tamaño** Un entero sin signo de 32 bits que indica el tamaño en bytes de la estructura del átomo.
- **Tipo** Un valor entero sin signo de 32 bits establecido en 'mdhr'
- **Versión** Un byte que se establece en 0.
- **Banderas** Tres bytes que se establecen en 0.
- **nextItemID** Un entero sin signo de 32 bits que indica el valor a utilizar para la ID del artículo del siguiente artículo creado o asignado a una ID de artículo. Si el valor es todos unos, indica que las adiciones futuras requerirán una búsqueda de un ID de elemento no utilizado.

Nota: Si se elimina el último elemento de metadatos con un átomo de información del elemento y el valor de nextItemID es 0xFFFFFFFF, una implementación puede restablecer el valor nextItemID del átomo del encabezado de metadatos a 0 para que las nuevas asignaciones vuelvan a ser eficientes (es decir, no requieren una búsqueda para identificadores no utilizados).

A.1.8. Átomo Free y UUID

Para permitir que los metadatos se reescriban fácilmente y sin la necesidad de reescribir todo el archivo de película QuickTime, pueden aparecer átomos de espacio libre en cualquier lugar de la definición del átomo de metadatos entre las posiciones de otros átomos contenidos en el átomo de metadatos. Los átomos del espacio libre no se pueden insertar entre los elementos en el átomo de la lista de elementos de metadatos o dentro de los átomos en el átomo de la lista de elementos de metadatos. Esta restricción en la definición de átomo de espacio libre evita el riesgo de confundir un átomo de espacio libre con el significado de un identificador 'free' o un átomo de valor de tipo 'free' definido en el contexto de la estructura del átomo de metadatos.

De manera similar, los átomos de UUID para extensiones específicas se pueden colocar en cualquier posición donde se permita una sucesión de átomos. Tenga en cuenta que los átomos UUID no deben crearse para átomos ya definidos utilizando códigos de cuatro caracteres.

Los átomos no reconocidos (es decir, aquellos átomos cuyos tipos no están definidos en el contexto de la estructura del átomo de metadatos y están contenidos dentro del átomo de la lista de elementos de metadatos) se ignoran.

A.1.9. Átomo Ctry y Lang

Cuando los elementos de metadatos tienen valores individuales asociados con más de un país o más de un idioma, se requiere un átomo de la lista de países y/o un átomo de la lista de idiomas. Alternativamente, si todos los valores están asociados con cero o un país, no se requiere un átomo de la lista de países, y si todos los valores están asociados con cero o un idioma, no se requiere un átomo de la lista de idiomas.

- **Átomo ctry** Cuando uno o más elementos deben identificarse como adecuados para más de un país, cada lista de países se almacena en este átomo opcional. El átomo de la lista de países es un átomo completo con un tipo de átomo de "intento".

  Cada lista comienza con un recuento de dos bytes del número de elementos en la lista, y luego cada código ISO 3166 que representa a los países en la lista.

  El átomo consiste en un recuento del número de listas, expresado como un entero de 32 bits, y luego estas listas, agregadas de extremo a extremo. El átomo de la lista de países contiene los siguientes campos:

  - **Tamaño** Un entero sin signo de 32 bits que indica el tamaño en bytes de la estructura del átomo.
  - **Tipo** Un valor entero sin signo de 32 bits establecido en 'ctry'
  - **Versión** Un byte que se establece en 0.
  - **Banderas** Tres bytes que se establecen en 0.
  - **Entry_count** Un entero de 32 bits que indica el número de matrices de país que se deben seguir en este átomo.
  - **Country_count** Un entero de 16 bits que indica el número de países en la matriz.
  - **País [Country_count]** Un conjunto de enteros de 16 bits, definidos de acuerdo con la definición ISO 3166 de códigos de país.

  Tenga en cuenta que:

  Los índices en el átomo de la lista de países están basados en 1. Cero (0) está reservado y nunca se usa como índice. Actualmente, hay un límite de 255 países que pueden registrarse en un átomo de la lista de países.

- **Átomo lang** Cuando uno o más elementos deben identificarse como adecuados para más de un idioma, cada lista de idiomas se almacena en este átomo opcional. El átomo de la lista de idiomas es un átomo completo con un tipo de átomo de "lang".

  Cada lista comienza con un recuento de 2 bytes del número de elementos en la lista, y luego cada código ISO 639-2/T, empaquetado en dos bytes, de acuerdo con la definición del Código de Idioma ISO en la especificación MP4.

El átomo consiste en un recuento del número de listas, expresado como un entero de 32 bits, y luego estas listas, agregadas de extremo a extremo. El átomo de la lista de idiomas contiene los siguientes campos:

- **Tamaño** Un entero sin signo de 32 bits que indica el tamaño en bytes de la estructura del átomo.
- **Tipo** Un valor entero sin signo de 32 bits establecido en 'lang'
- **Versión** Un byte que se establece en 0.
- **Banderas** Tres bytes que se establecen en 0.
- **Entry_count** Un número entero de 32 bits que indica el número de matrices de idiomas a seguir en este átomo.
- **Recuento de idiomas** Un entero de 16 bits que indica el número de idiomas en la matriz.
- **Idioma [Language_count]** Un conjunto de enteros de 16 bits, definidos de acuerdo con la definición ISO 639-2 / T de códigos de idioma.

Tenga en cuenta que:

Los índices en el átomo de la Lista de idiomas están basados en 1. Cero (0) está reservado y nunca se usa como índice. Actualmente, hay un límite de 255 idiomas que pueden registrarse en un átomo de la Lista de idiomas.

A.1.10. Átomo Keys

El átomo de claves de elementos de metadatos contiene una lista de las claves de metadatos que pueden estar presentes en el átomo de metadatos. Esta lista está indexada comenzando con 1; 0 es un valor de índice reservado. El átomo de teclas de elemento de metadatos es un átomo completo con un tipo de átomo de "keys".

Este átomo tiene la siguiente estructura:

- **Tamaño** Un entero sin signo de 32 bits que indica el tamaño en bytes de la estructura del átomo.
- **Tipo** Un valor entero sin signo de 32 bits establecido en 'claves'
- **Versión** Un byte que se establece en 0
- **Banderas** Tres bytes que se establecen en 0
- **Entry_count** Un entero de 32 bits que indica el número de matrices de claves a seguir en este átomo
- **Tamaño_clave** Un entero de 32 bits que indica el tamaño de toda la estructura que contiene una definición clave. Por lo tanto, key_size = sizeof (key_size) + sizeof (key_namespace) + sizeof (key_value). Dado que key_size y key_namespace son números enteros de 32 bits,

juntos tienen un tamaño de 8 bytes. Por lo tanto, la estructura key_value será igual a key_size - 8.

- **Key_namespace** Un entero de 32 bits que define un esquema de nombres utilizado para las claves de metadatos. Las teclas de metadatos de ubicación, por ejemplo, usan el espacio de nombres de teclas “mdta”.
- **Key_value [Key_size-8]** Una matriz de enteros de 8 bits, cada uno con el nombre real de la clave de metadatos. Las claves con el espacio de nombres “mdta” utilizan una convención de nomenclatura DNS inversa. Por ejemplo, las coordenadas de metadatos de ubicación utilizan un valor_clave de metadatos de ’com.apple.quicktime.location.ISO6709’.

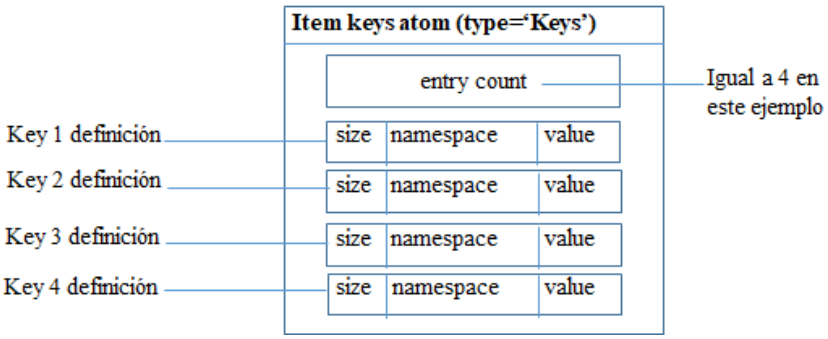


Figura A.3: Metadatos Elemento Claves átomo

Tenga en cuenta que:

- Los índices en el átomo de claves de elementos de metadatos están basados en 1 (1...entry_count).
- Cero (0) está reservado y nunca se usa como índice.
- La estructura de key_value depende del espacio de nombres clave.

A.1.11. Átomo Ilst

El átomo de la lista de elementos de metadatos contiene una lista de valores de metadatos reales que están presentes en el átomo de metadatos. Los elementos de metadatos están formateados como una lista de elementos. El átomo de la lista de elementos de metadatos es del tipo ’ilst’ y contiene varios elementos de metadatos, cada uno de los cuales es un átomo.

- **Átomo de elemento de metadatos** Cada elemento en el átomo de la lista de elementos de metadatos se identifica por su clave. El tipo de átomo para cada átomo de elemento de metadatos debe establecerse igual al índice de la clave para los metadatos dentro del átomo de elemento, tomando este índice del átomo de claves de elemento de metadatos. Además, cada átomo de elemento de metadatos contiene un átomo de valor para contener el valor del elemento de metadatos.

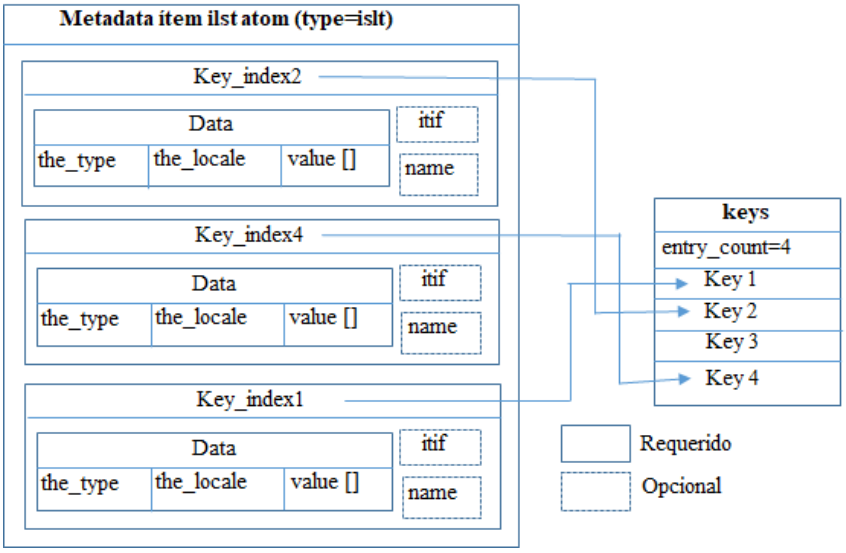


Figura A.4: El átomo ilst y la conexión de elemento/clave

El átomo del elemento de metadatos tiene la siguiente estructura:

- **Item_info** Un átomo de información de artículo opcional, ver átomo de información de artículo (ID y banderas)
- **Nombre** Un átomo de nombre opcional.
- **Valor** de datos átomo [] Una matriz de átomos de valor.

- **Átomo de Valor** El valor del elemento de metadatos se expresa como datos inmediatos en un átomo de valor. El átomo de valor comienza con dos campos: un indicador de tipo y un indicador de configuración regional. Tanto el tipo como los indicadores locales tienen cuatro bytes de longitud. Puede haber múltiples entradas de "valor", utilizando diferentes tipos, códigos de país o idioma (consulte la sección de Pedido de datos a continuación para obtener el pedido requerido).

  La estructura del átomo de valor contiene los siguientes campos:

  - **Tipo** Un indicador de tipo, definido en Indicador de tipo.
  - **Lugar** Un indicador de configuración regional, definido en Indicador de configuración regional.
  - **Indicador de tipo** El indicador de tipo está formado por cuatro bytes divididos entre dos campos. El primer byte indica el conjunto de tipos de los que se extrae el tipo. El segundo al cuarto byte forma el segundo campo y su interpretación depende del valor en el primer campo.

    El byte indicador debe tener un valor de 0, lo que significa que el tipo se extrae del conocido conjunto de tipos. Los demás valores están reservados.

Si el byte indicador de tipo es 0, los siguientes 24 bits contienen el tipo conocido. Consulte la lista de Tipos conocidos, en la sección Tipos conocidos más abajo.

- **Indicador local** El indicador de configuración regional está formateado como un valor de cuatro bytes. Está formado por dos valores de dos bytes: un indicador de país y un indicador de idioma. En cada caso, el campo de dos bytes tiene los valores posibles que se muestran en el Cuadro A.1.

Cuadro A.1: Indicadores de país e idioma

Valor	Sentido
0	Este átomo proporciona el valor predeterminado de este dato para cualquier localidad que no esté explícitamente listada.
1 a 255	El valor es un índice en la lista de países o idiomas (el byte superior es 0).
De otra manera	El valor es un código ISO 3166 (para el código de país) o un código ISO 639-2/T empaquetado (para el idioma).

Tenga en cuenta que los códigos ISO 3166 e ISO 639-2/T tienen un valor distinto de cero en su byte superior, por lo que tendrán un valor >255.

Las aplicaciones de software que leen metadatos pueden personalizarse para un conjunto específico de países o idiomas. Si un escritor de metadatos no desea limitar un elemento de metadatos a un conjunto específico de países, debe usar el valor reservado ZZ de ISO 3166 como su código de país. Del mismo modo, si el escritor de metadatos no quiere limitar el idioma del usuario (esto no se recomienda), utiliza el valor “und” (indeterminado) de la especificación ISO 639-2/T.

Una aplicación de software coincide con un código de país si (a) el valor con el que debe coincidir es 0 (predeterminado) o (b) los códigos son iguales. Una aplicación de software coincide con una lista de códigos si su valor es miembro de esa lista. Una aplicación de software coincide con una configuración regional si el país y el idioma coinciden.

Para reiterar, si el valor de country_indicator está en el rango de 1 a 255, se interpreta como el índice basado en 1 para una lista de países en el átomo del idioma del país en el átomo de metadatos. Si el valor de language_indicator está en el rango de 1 a 255, se interpreta como el índice basado en 1 para una lista de idiomas en el átomo de la Lista de idiomas en el átomo de metadatos. De lo contrario, country_indicator o language_indicator no está especificado (0) o tiene el valor inmediato para un solo país o idioma.

A.1.12. Átomo Udta

Átomos de datos de usuario le permiten definir y almacenar los datos asociados a un objeto de QuickTime, como una película ’moov’, pista ’Trak’, o ’medios mdia’. Esto incluye tanto la información que QuickTime busca, como la información de derechos de autor o si una película

Cuadro A.2: Ejemplos de etiquetas de metadatos

País	Lenguaje	Sentido
0	eng	Todos los hablantes de inglés, independientemente del país.
GB	0	Todas las personas en el Reino Unido, independientemente del idioma.
CA	fra	Francófonos en Canadá
DE,GB,FR,IT	0	Personas en Alemania, Francia, Reino Unido e Italia, independientemente del idioma
DE,GB,FR,IT	deu,fra	Personas en Alemania, Francia, Reino Unido e Italia que hablan alemán o francés.
0	0	Por defecto, todos los hablantes en todos los países.

debe repetirse, como la información arbitraria, proporcionada por y para su aplicación, que QuickTime simplemente ignora.

Un átomo de datos de usuario cuyo padre inmediato es un átomo de película contiene datos relevantes para la película en su conjunto. Un átomo de datos de usuario cuyo padre es un átomo de pista contiene información relevante para esa pista específica. Un archivo de película de QuickTime puede contener muchos átomos de datos de usuario, pero sólo se permite átomo de datos de un usuario cuando el hijo inmediato de cualquier átomo de película dada o átomo de pista.

El átomo de datos de usuario tiene un tipo de átomo de 'udta'. Dentro del átomo de datos del usuario hay una lista de átomos que describe cada pieza de datos del usuario. Los datos del usuario proporcionan una manera simple de extender la información almacenada en una película QuickTime. Por ejemplo, los átomos de datos del usuario pueden almacenar la posición de la ventana de una película, las características de reproducción o la información de creación.

Esta sección describe los átomos de datos que QuickTime reconoce. Es posible crear nuevos tipos de átomos de datos que reconoce su propia aplicación. Las solicitudes deben ignorar cualquier tipo de átomos de datos que no reconocen.

El átomo de datos del usuario contiene los siguientes elementos de datos.

- **Tamaño** Un entero de 32 bits que especifica el número de bytes en este átomo de datos de usuario.
- **Tipo** Un entero de 32 bits que identifica el tipo de átomo; este campo debe establecerse en 'udta'.
- **Lista de datos del usuario** Una lista de datos de usuario que está formateada como una serie de átomos. Cada elemento de datos en la lista de datos del usuario contiene información de tamaño y tipo junto con sus datos de carga útil. Por razones históricas, la lista de datos se termina opcionalmente con un número entero de 32 bits establecido en

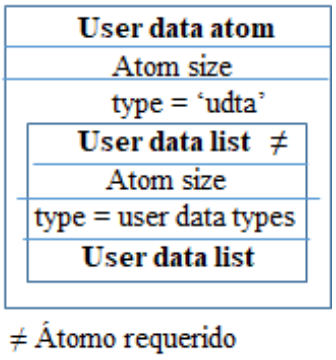


Figura A.5: El diseño de un átomo de datos de usuario

0. Si está escribiendo un programa para leer átomos de datos de usuario, debe permitir la terminación de 0. Sin embargo, si está escribiendo un programa para crear átomos de datos de usuario, puede omitir con seguridad el 0 de terminación.

A.1.13. Átomo Mebx

La descripción de la muestra de metadatos temporizados contiene información que define cómo interpretar las muestras de medios de metadatos temporizados. Esta descripción de muestra se basa en el encabezado de descripción de muestra estándar, como se describe en Átomos de descripción de muestra.

La descripción de la muestra de metadatos es un formato de descripción de muestra derivada que describe los valores de metadatos representados en los átomos. También puede incluir otros átomos que no contienen valores de metadatos.

Cero, uno o más valores pueden llevarse en una descripción de muestra de metadatos para un tiempo particular.

El campo de formato de datos contiene el formato de los medios de metadatos temporizados, que se establece en 'mebx'.

Nota: Aquí no se describen otras formas de medios de metadatos temporizados. Se indicarán mediante un código de tipo alternativo en lugar de "mebx".

La descripción de la muestra de metadatos debe contener un átomo de tabla de claves de metadatos y opcionalmente contiene un átomo de velocidad de bits que sigue al encabezado del átomo de descripción de muestra estándar, definido a continuación. Se pueden introducir otros átomos en el futuro.

Tabla de claves de metadatos Un átomo que contiene una tabla de claves y asignaciones para datos de carga útil en las muestras de medios de metadatos temporizados correspondientes.

Átomo de velocidad de bits Un átomo opcional que contiene datos que indican la velocidad de bits de un flujo de medios.

A.1.14. Átomo Keys

El átomo de la tabla de claves de metadatos contiene una tabla de claves y asignaciones para datos de carga útil en las muestras de medios de metadatos temporizados correspondientes. El tipo de átomo de la tabla de claves de metadatos se establece en “claves”.

El átomo de la tabla de claves de metadatos contiene una o más instancias de átomos de claves de metadatos, uno para cada configuración de una clave que puede ocurrir en las unidades de muestra de la pista. Por ejemplo, si hay dos claves, habrá dos átomos clave de metadatos en el átomo de la tabla de claves de metadatos, uno para cada clave.

Si el átomo de la tabla de claves de metadatos no contiene una clave para la que está buscando un cliente, ninguna muestra de medios de metadatos temporizados asociada con esta descripción de muestra contiene valores con esa clave.

Si el átomo de la tabla de claves de metadatos contiene una clave particular, esto no garantiza que se escribieron muestras de medios de metadatos temporizados que contienen un valor para la clave. Por lo tanto, los clientes que encuentren una clave en el átomo de la tabla de claves de metadatos aún pueden tener que buscar en los ejemplos de medios de metadatos cronometrados de la pista valores para determinar si la pista tiene metadatos particulares.

Nota: Tener la capacidad de que el átomo de la tabla de claves de metadatos contenga claves que no están asociadas con ninguna instancia de muestras de medios de metadatos temporizados permite que una descripción de muestra de metadatos se complete con claves que podrían descubrirse (por ejemplo, durante un proceso de captura) y luego muestras se escribirá con un enlace solo para las claves encontradas. Si nunca se usa una clave, no es necesario que la descripción de muestra de metadatos cronometrada se reescriba para excluir la clave que no es necesaria.

Si es posible eliminar entradas no utilizadas y volver a escribir la descripción de muestra de metadatos de manera eficiente, esto es preferible.

Si una pista de metadatos cronometrada incluye una clave en la descripción de la muestra de metadatos pero tiene valores que utilizan la clave en las muestras de medios asociadas, la descripción de la muestra de metadatos aún puede reescribirse para eliminar la clave del átomo de la tabla de claves de metadatos. Si bien los valores de metadatos permanecen en muestras de medios asociados, los datos ya no son accesibles porque la clave ya no está. Se debe tener cuidado si los valores se deben eliminar del archivo de película. Aunque no es un requisito, los datos restantes pero ahora inalcanzables pueden eliminarse copiando solo los valores de metadatos referenciados al copiar muestras de medios a una nueva pista.

- **Tamaño** Un entero sin signo de 32 bits que indica el tamaño en bytes de la estructura del átomo.
- **Tipo** Un valor entero sin signo de 32 bits establecido en ’claves’
- **Tabla de claves de metadatos** Una matriz de átomos clave de metadatos

Metadata Key Table Atom	Bytes
Atom size	4
Type = 'keys'	4
Metadata key table	variable

Figura A.6: El diseño de un átomo de tabla de claves de metadatos

A.1.15. Átomo Btrt

El átomo de velocidad de bits opcional puede estar presente al final de cualquier descripción de muestra de metadatos temporizados para señalar la información de velocidad de bits de un flujo. La información de velocidad de bits se puede usar para la configuración del búfer.

Bit rate atom	Bytes
Atom size	4
Type = 'btrt'	4
Buffer size	4
Max bit rate	4
Average bit rate	4

Figura A.7: El diseño de un átomo de velocidad de bits

- **Tamaño** Un entero sin signo de 32 bits que indica el tamaño en bytes de la estructura del átomo.
- **Tipo** Un valor entero sin signo de 32 bits establecido en 'btrt'
- **Tamaño del búfer** Un entero sin signo de 32 bits que indica el tamaño sugerido del búfer de datos asociado
- **Max tasa de bits** Un entero sin signo de 32 bits que indica la velocidad de bits máxima en bits / segundo del flujo de medios asociado
- **Tasa de bits promedio** Un entero sin signo de 32 bits que indica la tasa de bits promedio en bits / segundo del flujo de medios asociado

A.1.16. Átomo Local_key_id

Un átomo clave de metadatos se identifica mediante un local_key_id correspondiente a un código de tipo entero de 32 bits (o FourCC) local a la pista de metadatos temporizados que lo

contiene. Este local_key_id corresponderá a los tipos de átomos dentro de los datos de muestra de metadatos.

Por ejemplo, si el átomo de clave de metadatos tiene el tipo de átomo de 'stuf', cualquier átomo de tipo 'stuf' en muestras de metadatos temporizados que compartan esta descripción de muestra tiene el valor de esta clave. Se puede usar cualquier valor que se ajuste a un entero endian grande de 32 bits (como 'stuf' o el entero 72). Si se usa un FourCC, se recomienda que el valor sea mnemónico si es posible. Por ejemplo, un tipo de átomo clave de metadatos de "actr" podría contener información sobre actores en una película. Vea el formato de datos de muestra de metadatos a continuación.

Hay dos tipos de átomos reservados para los átomos clave de metadatos: 0 y 0xFFFFFFFF.

Un local_key_id de 0 indica que el átomo clave de metadatos no se utiliza y no debe interpretarse. Esta indicación permite que la clave se marque como no utilizada en la descripción de la muestra de metadatos temporizados sin que sea necesario reescribir o cambiar el tamaño de la descripción de la muestra y los átomos principales.

Un local_key_id de 0xFFFFFFFF no debe aparecer en un átomo clave de metadatos. Está reservado para uso futuro y puede ocurrir como un tipo de átomo en muestras de metadatos temporizados.

Todos los demás códigos de tipo están disponibles para su uso como local_key_id.

Nota: Debido a que los átomos dentro del átomo de la tabla de claves de metadatos pueden adoptar cualquier tipo de átomo, no debe haber una interpretación especial del tipo para los átomos contenidos que no sea para el valor especial 0. Por lo tanto, incluir un átomo 'libre' no tiene el significado convencional en el átomo clave de metadatos. Se recomienda que los escritores eviten el uso de códigos de tipo de átomo existentes demasiado confusos.

Cada átomo clave de metadatos contiene un número variable de átomos que definen la estructura clave, opcionalmente el tipo de datos para los valores y opcionalmente la información de localización para los valores. Los átomos pueden ser introducidos en el futuro.

Un átomo clave de metadatos debe contener un átomo de declaración de metadatos.

Metadata Key Atom	Bytes
Atom size	4
Type = 'local_key_id'	4
Variable array of atoms	variable

Figura A.8: El diseño de un átomo clave de metadatos

- **Tamaño** Un entero sin signo de 32 bits que indica el tamaño en bytes de la estructura del átomo.

- **Tipo** Un valor entero sin signo de 32 bits establecido en local_key_id
- **Matriz variable de átomos** Una matriz de átomos que llevan las definiciones de estructura clave y otra información opcional.

A.1.17. Átomo Keyd

El átomo de declaración de clave de metadatos contiene el espacio de nombres clave y el valor clave de ese espacio de nombres para los valores dados. El tipo del átomo de declaración de clave de metadatos es “keyd”.

Metadata Key Declaration Atom	Bytes
Atom size	4
Type = ‘keyd’	4
Key_namespace	4
Key_value array	variable

Figura A.9: El diseño de un átomo de declaración de clave de metadatos

- **Tamaño** Un entero sin signo de 32 bits que indica el tamaño en bytes de la estructura del átomo.
- **Tipo** Un valor entero sin signo de 32 bits establecido en ’keyd’
- **Key_namespace** Un identificador de 32 bits que describe el dominio y la estructura del valor_clave. Por ejemplo, esto podría indicar que key_value es una cadena de estilo de dirección inversa (como “com.apple.quicktime.ISO6709”), un código binario de cuatro caracteres (como una clave de datos de usuario ’cprt’), un recurso uniforme Identificador (URI) u otras estructuras (como formatos nativos de otros estándares de metadatos). Se deben registrar nuevos espacios de nombres de clave, pero debido a que a menudo se puede usar una cadena de estilo de dirección inversa, el uso del espacio de nombres de clave de dirección inversa puede ser suficiente para la mayoría de los usos.
- **Matriz de valor_clave** Una matriz de bytes de 8 bits sin signo que contiene el valor de la clave. La interpretación de esta matriz está definida por el campo key_namespace asociado.

A.1.18. Átomo Dtyp

Se puede usar un átomo de definición de tipo de datos de metadatos para especificar el tipo de datos del valor del átomo clave de metadatos. El tipo de átomo de definición de tipo de datos de metadatos es “dtyp”.

- **Tamaño** Un entero sin signo de 32 bits que indica el tamaño en bytes de la estructura del átomo.

Metadata Datatype Definition Atom	Bytes
Atom size	4
Type = 'dtyp'	4
Datatype_namespace	4
Datatype array	variable

Figura A.10: El diseño de un átomo de definición de tipo de datos de metadatos

- **Tipo** Un valor entero sin signo de 32 bits establecido en 'dtyp'
- **Espacio de nombres de tipo de datos** Un identificador de 32 bits que describe cómo interpretar el tipo de datos para el valor. Se deben registrar nuevos tipos de espacios de nombres con Apple.
- **Matriz de tipo de datos** Una matriz de bytes de 8 bits sin signo que contiene la designación de tipo de datos para valores en muestras de medios de metadatos temporizados que tienen esta clave. La interpretación de esta matriz está definida por el espacio de nombres de tipo de datos asociado.

La combinación de espacio de nombres de tipo de datos y matriz de tipos de datos indica el tipo de datos (o estructura) de un valor de elemento de metadatos. El tipo de espacio de nombres de tipo de datos indica la interpretación del valor de la matriz de tipos de datos. Esta especificación define dos tipos de espacio de nombres de tipo de datos:

Si el espacio de nombres de tipo de datos es 0, la matriz de tipos de datos contiene un entero sin signo de 32 bits big endian correspondiente a un tipo bien conocido especificado en la Tabla 3-5. Si el espacio de nombres de tipo de datos es 1, la matriz de tipos de datos contiene una cadena UTF-8 de estilo de dirección inversa que indica un tipo de datos extendido. Este tipo de espacio de nombres de tipo de datos puede usarse si el tipo de datos no tiene un tipo de datos conocido correspondiente. La matriz de tipo de datos consta de los bytes de una cadena UTF-8 que distingue entre mayúsculas y minúsculas sin un terminador nul (' 0'). Por ejemplo, una matriz de tipo de datos hipotética "com.company.my-custom-datatype" podría registrar un tipo de datos personalizado que pertenece al propietario del registro DNS "mycompany.com".

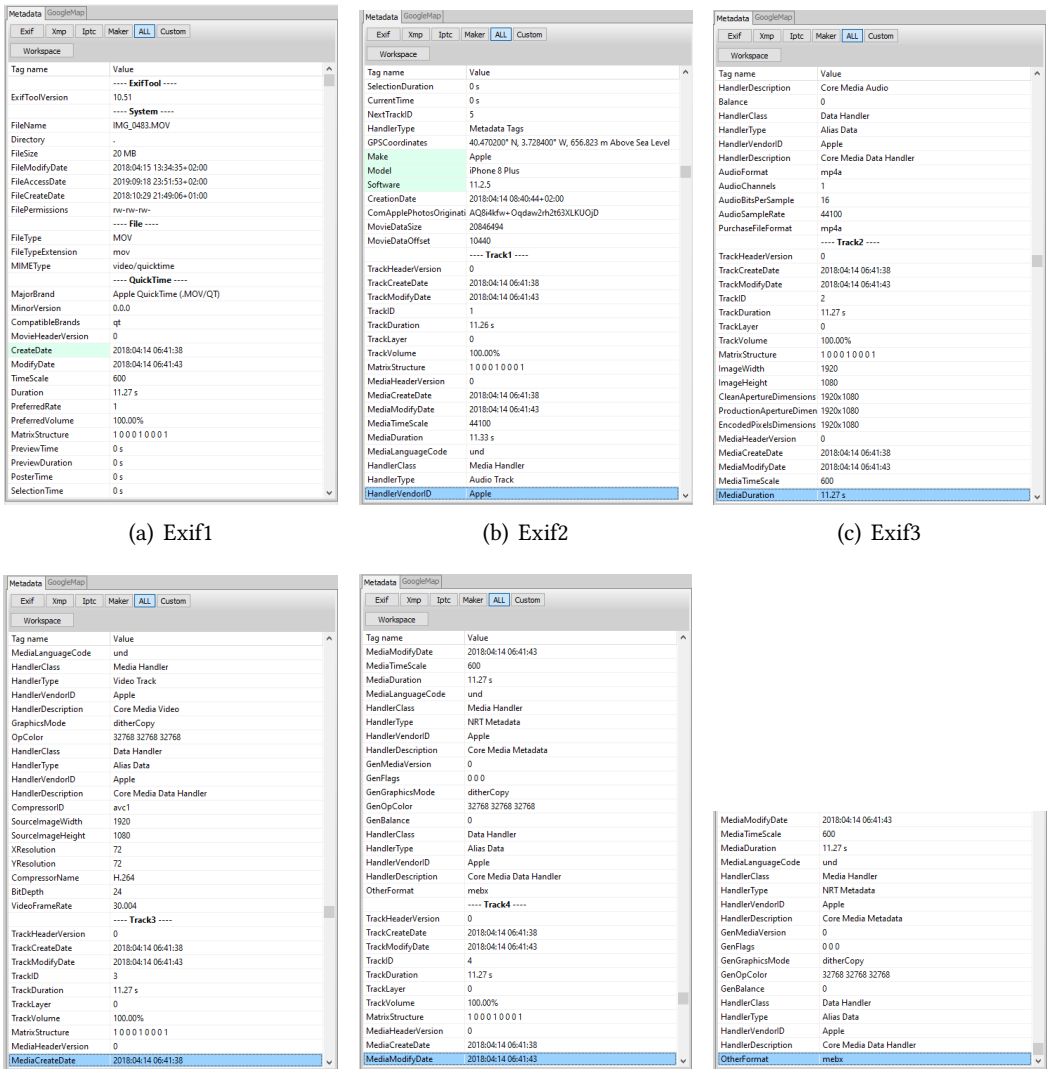
Un espacio de nombres de tipo de datos distinto de 0 o 1 puede aparecer en una pista de metadatos cronometrada, tal vez escrita de acuerdo con una versión posterior de esta especificación. Los valores de los elementos de metadatos con tipos de datos no reconocidos deben ignorarse. Aun así, aún es posible procesar algo en el elemento de metadatos con un tipo de datos no reconocido, como copiarlo entre pistas.

Nota: Los nuevos espacios de nombres de tipo de datos deben estar registrados en Apple. Muchos usos para los tipos de datos de metadatos propietarios o personalizados pueden satisfacerse utilizando el código de tipo de espacio de nombres de tipo de datos extendido 1. Esto permite que el nuevo tipo de datos se especifique sin registro. La razón para agregar un

tipo de espacio de nombres de tipo de datos personalizado es permitir que se use un esquema de numeración o nomenclatura existente de un estándar de metadatos externo con elementos de metadatos.

A.2. Capturas de pantalla de aplicaciones

En la Figura A.11 se muestra como la aplicación ExifToolGUI muestra la información acerca del vídeo estudiado obtenido de nuestro Dataset.

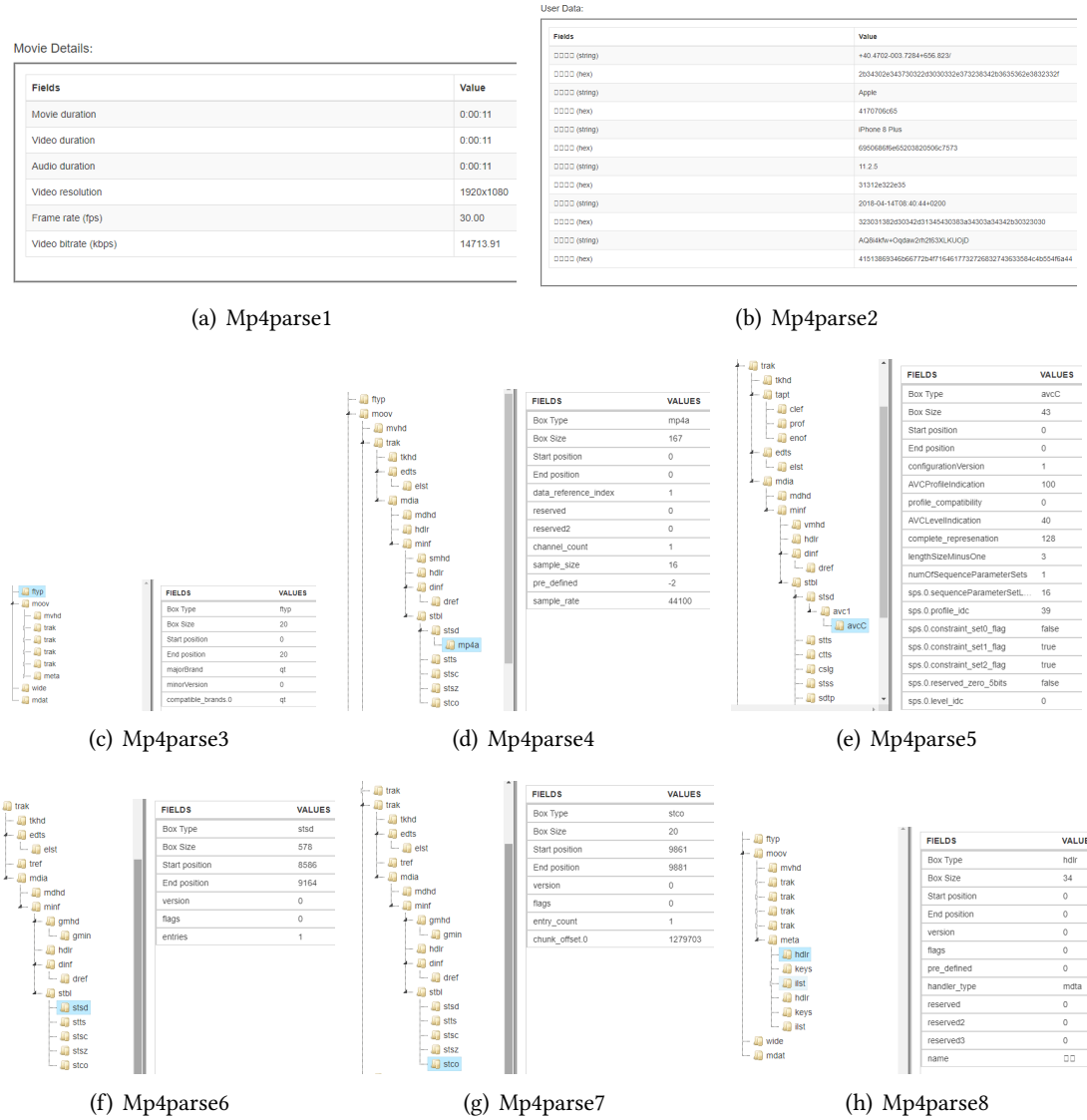


(a) Exif1 (b) Exif2 (c) Exif3

(d) Exif4 (e) Exif5 (f) Exif6

Figura A.11: Pantallazos de la herramientas ExifToolGUI

En la Figura A.12 se muestra como la aplicación Mp4parser muestra la información acerca del vídeo estudiado obtenido de nuestro Dataset.



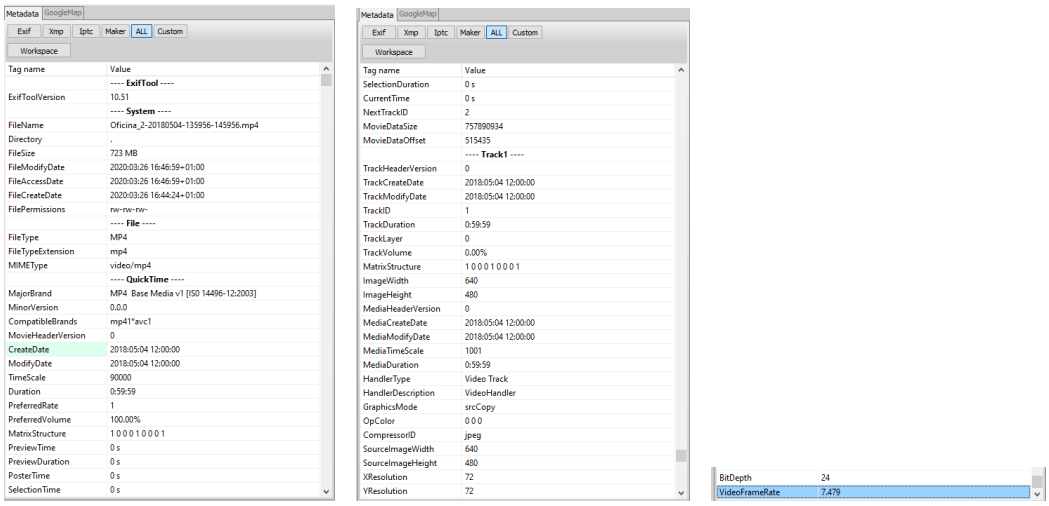
(a) Mp4parse1 (b) Mp4parse2

(c) Mp4parse3 (d) Mp4parse4 (e) Mp4parse5

(f) Mp4parse6 (g) Mp4parse7 (h) Mp4parse8

Figura A.12: Pantallazos de la herramientas Mp4parse

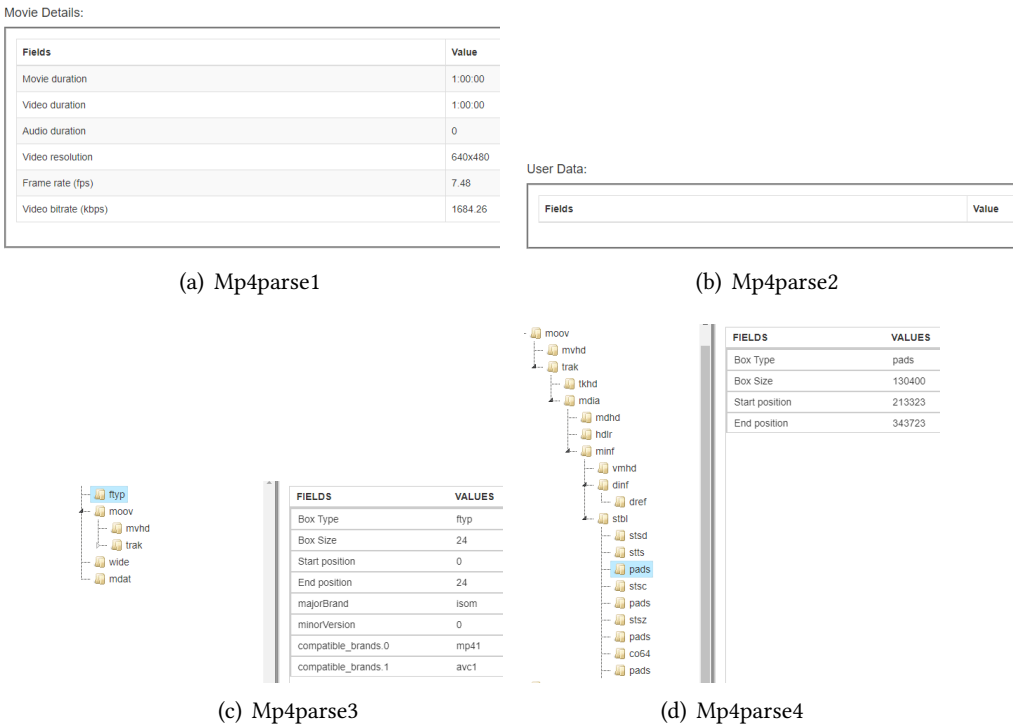
En la Figura A.13 se muestra como la aplicación ExifToolGUI muestra la información acerca del vídeo estudiado obtenido de una cámara de seguridad.



(a) Exif1 (b) Exif2 (c) Exif3

Figura A.13: Pantallazos de la herramientas ExifToolGUI

En la Figura A.14 se muestra como la aplicación Mp4parse muestra la información acerca del vídeo estudiado obtenido de una cámara de seguridad.



(c) Mp4parse3 (d) Mp4parse4

Figura A.14: Pantallazos de la herramientas Mp4parse