TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

CURSO 2015-2016

HERRAMIENTA PARA LA EXTRACCIÓN AUTOMÁTICA DE METADATOS EN VÍDEOS DE DISPOSITIVOS MÓVILES

MIGUEL ESTEBAN COBO

Directores:

Luis Javier García Villalba

Ana Lucila Sandoval Orozco

FACULTAD DE INFORMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID

El abajo firmante autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Grado: "Herramienta para la Extracción Automática de Metadatos en Vídeos de Dispositivos Móviles", realizado durante el curso académico 2015-2016 bajo la dirección de Luis Javier García Villalba y Ana Lucila Sandoval Orozco en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

_____

Miguel Esteban Cobo

Agradecimientos

A mis directores Luis Javier García Villalba y Ana Lucila Sandoval Orozco por haberme dado la oportunidad de trabajar con ellos, por haber hecho posible la realización de este trabajo fin de grado y por toda su dedicación durante estos meses de duro trabajo.

A todos mis compañeros de la Facultad por vuestra amistad y vuestros ánimos.

Mi más especial agradecimiento está dedicado a mi familia por su apoyo, cariño y esfuerzo que me han permitido realizar todos mis estudios.

Resumen

Actualmente el número de cámaras fotográficas en dispositivos móviles crece a un ritmo imparable. Asimismo la calidad y prestaciones de las mismas hacen que sean de uso común, desbancando poco a poco a las Cámaras fotográficas digitales. Este escenario produce que el análisis forense de este tipo de vídeos cobre especial importancia y sea necesario y útil en multitud de situaciones (pruebas en casos judiciales, espionaje industrial, privación de la libertad de prensa, pederastia, etc).

En este trabajo se ha desarrollado una herramienta de ayuda al analista forense en el proceso de análisis de los metadatos de vídeos en formato MP4 con compresión H.264 y ACC. La herramienta permite diversas funciones complejas como son los distintos tipos de consultas avanzadas sobre la información de los metadatos de grandes conjuntos de vídeos o funciones de geoposicionamiento.

Palabras clave

Adquisición de vídeo, cámara de teléfonos móviles, metadatos, análisis forense, métodos forenses.

Abstract

Currently the number of cameras in mobile devices is growing at an unstoppable rate. Also the quality and performance of the same make are in common use, edging slowly to Digital Cameras. This scenario causes the forensic analysis of such videos is particularly important and necessary and useful in many situations (evidence in court cases, industrial espionage, deprivation of freedom of the press, child abuse, etc.).

In this work has been developed a tool to help the forensic analyst in the analysis process metadata of videos in MP4 format with H.264 compression and ACC The tool allows various complex functions such as different types of advanced queries on Exif metadata information of large sets of images or functions of geopositioning.

Keywords

Video acquisition, camera mobile phones, metadata, forensics analysis, forensic methods.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Lista de Acrónimos

AAC	Advanced Audio Coding
ASCII	American Standard Code for Information Interchange
AVC	Advanced Vídeo Coding
AVI	Audio Vídeo Interleave
BCD	Binary Code Decimal
B-Frames	Imágenes de codifcicación mediante predicción bidireccional
CCD	Charge Coupled Device
CFA	Color Filter Array
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
CYGM	Cyan-Yellow-Green-Magenta
CYYM	Cyan-Yellow-Yellow-Magenta
DC	Discrete Cosine
DCT	Discrete Cosine Transform
DQ	Double Quantification
DSC	Digital Still Camera
EXIF	Exchangeable Image File Format
GOP	Group of Pictures
GPS	Global Position System
GRGB	Green-Red-Green-Blue

HD	High Definition
IFD	Image File Directory
I-Frames	Imágenes intra codificadas
IIM	Information Interchange Model
I-MB	Macrobloques intra codificados
IPTC	International Press Telecommunications Council
ITU-T	International Telecommunication Union - Telecommunication
JPEG	Join Photograph Expert Group
KML	Keyhole Markup Languaje
MKV	Matroska Vídeo
MOS	Metal Oxide Semiconductor
MOV	Quick Time Movie Format
MP4	Media Player 4
MPEG	Moving Picture Expert Group
PC	Personal Computer
P-Frames	Imágenes de codifcicación mediante predicción
P-MB	Macrobloques inter codificados
PSD	Photoshop Document
RAM	Random Access Memory
RGB	Red-Green-Blue
RGBE	Red-Green-Blue-Emerald

RGBW	Red-Green-Blue-White
S-MB	Macrobloques omitidos
SO	Sistema Operativo
TIFF	Tagged Image File Format
UTC	Coordinated Universal Time
VPF	Variation Prediction Footprint
XMP	Extensible Metadata Platform

1. INTRODUCCIÓN

1.1. Motivación

En los últimos años se ha venido presentando un aumento vertiginoso en la demanda de dispositivos móviles. Este aumento se debe en gran medida al desarrollo de la tecnología necesaria para abaratar costes y hacerlos más accesibles al público en general. Actualmente, existe una gran competencia entre los fabricantes por integrar una videocámara de alta definición al alcance del usuario en todo momento.

La mayoría de los tipos de dispositivos móviles poseen una cámara digital integrada. De hecho, el 97% de teléfonos móviles tienen una cámara digital integrada pudiendo así, ser llevadas por sus dueños gran parte del tiempo a la mayoría de los lugares a los que asisten [1]. Existen predicciones que indican que las cámaras digitales tradicionales (del inglés *Digital Still Camera* (DSC)) desaparecerán en pro de las nuevas cámaras digitales integradas en dispositivos móviles [2].

Según un estudio realizado por IC Insights Inc. [3] sobre la diversidad del mercado de los sistemas con cámaras, la venta de DSCs descenderá de un 47% obtenido en 2012 a un 27% en 2016. Asimismo, las ventas de cámaras digitales integradas en teléfonos móviles, ordenadores y tabletas aumentarán de un 31% en 2012 a un 42% en 2016.

Como consecuencia de lo anterior y dada la gran cantidad de tiempo que una persona pasa junto a los dispositivos móviles, éstos se han convertido para muchas personas en el primer dispositivo de captura de fotografías y grabación de vídeos. Diariamente pueden verse imágenes generadas por dispositivos móviles en telenoticias, distintas aplicaciones, correo electrónico o en redes sociales. Por ejemplo, en las redes sociales más utilizadas (Facebook, YouTube,

Flikr, Twitter, etc.), una parte considerable de su contenido es capturado con cámaras digitales de dispositivos móviles [4]. Consecuentemente, las imágenes y vídeos digitales generados con dispositivos móviles son más utilizadas como testigos silenciosos en procesos judiciales (pornografía infantil, violencia callejera, redes sociales,...), siendo piezas cruciales de la evidencia de un delito [5, 6]. Todo esto hace que en ciertos casos existan restricciones legales o limitaciones tanto para su utilización en distintos lugares (colegios, universidades, oficinas de gobierno, empresas, etc) así como para el contenido capturado, como por ejemplo: capturas en las que aparezcan menores sin el consentimiento de su tutor legal o situaciones íntimas de las personas que vulneran su derecho a la privacidad, etc.

Por todas estas razones el análisis forense de imágenes y vídeos digitales de dispositivos móviles cobra especial fuerza en la actualidad. El estudio debe ser concreto para este tipo de dispositivos, ya que poseen características específicas que permiten obtener mejores resultados, no siendo válidas las técnicas forenses para imágenes digitales generadas por otros tipos de dispositivos. En [7] se describe de forma clara y razonada la necesidad de técnicas de análisis forense específicas para dispositivos móviles

1.2. Objetivos

El presente Trabajo Fin de Grado (TFG) tiene los siguientes objetivos:

- Realizar un estudio de la estructura de un vídeo digital, los átomos de los que están compuestos y las técnicas de extracción de metadatos de vídeos digitales con objeto de analizar y comprender las técnicas más relevantes.
- Presentar las principales técnicas de análisis forense que determinan la geolocalización de una imagen digital.

- Analizar detalladamente la especificación de almacenamiento de información de vídeos digitales con formato MP4 (Media Player 4) con compresión H.264.

- Diseñar e implementar en el lenguaje de programación Python un algoritmo que permita la extracción automática de metadatos almacenados en vídeos de dispositivos móviles.

1.3. Plan de Trabajo

El proyecto se ha desarrollado en 3 fases: Definición, Ejecución y Documentación del Proyecto. Las actividades realizadas en cada una de estas fases se presentan en la Tabla 1.1.

Nombre de tarea	Duración (días)	Inicio	Fin
• **Definición del proyecto**	**40**	**12/10/15**	**04/12/15**
- Reuniones semanales de seguimiento con los tutores	40	12/10/15	04/12/15
- Estudio de las técnicas de análisis forense en dispositivos móviles	15	12/10/15	30/10/15
- Estudio de las técnicas de análisis forense aplicadas a vídeos de dispositivos móviles	15	02/11/15	20/11/15
- Definición del proyecto	10	23/11/15	04/12/15
• **Ejecución del Proyecto**	**126**	**11/12/15**	**03/06/16**
- Especificación de requisitos	**45**	**14/12/15**	**12/02/16**
- Diseño	**40**	**15/02/16**	**08/04/16**
- Implementación	**35**	**11/04/16**	**27/05/16**
- Pruebas	**20**	**09/05/16**	**03/06/16**
- Control	**126**	**11/12/15**	**03/06/16**
• **Documentación**	**167**	**19/10/15**	**07/06/16**
- Generación de documentación del proyecto	165	19/10/15	03/06/16
- Preparación de la memoria	27	02/05/16	07/06/16

Tabla 1.1. Fases del proyecto

1. **Definición del proyecto:** En esta fase se realizaron varias reuniones con los tutores para definir y delimitar el proyecto de fin de grado, así como establecer las pautas del trabajo y definir horarios de tutorías y seguimiento del mismo.

2. **Ejecución del proyecto:** En esta fase tiene como finalidad el desarrollo del proyecto definido en la fase anterior. En ella se realizaron las siguientes actividades generales: Especificación de requisitos, diseño, implementación y pruebas. Asimismo, se realizaron actividades de seguimiento y control de todo el avance del proyecto, con el objetivo de agilizar los ajustes necesarios en cada una de las actividades. La Tabla 1.2 presenta las actividades realizadas en esta fase así como la duración de las mismas.

3. **Fase de documentación:** Esta fase se realizó en paralelo a las fases anteriormente mencionadas. Su principal objetivo fue generar toda la documentación del proyecto.

La Figura 1.1 muestra el diagrama de Gantt del proyecto.

Nombre de tarea	Duración (días)	Inicio	Fin
Especificación de requisitos	**45**	**14/12/15**	**12/02/16**
• Estudio de las técnicas de extracción de metadatos de imágenes de dispositivos móviles	10	14/12/15	25/12/15
• Estudio de contenedores multimedia para vídeos digitales	8	04/01/16	13/01/16
• Análisis detallado de la especificación de almacenamiento de metadatos en videos con formato MP4	12	14/01/16	29/01/16
Diseño	**40**	**15/02/16**	**08/04/16**
• Diseño de la base de datos que almacenará la información de los metadatos	5	15/02/16	19/02/16
• Diseño del algoritmo de extracción de átomos de un contenedor multimedia MP4	18	22/02/16	16/03/16
• Diseño del módulo de tratamiento individual de vídeos digitales	7	17/03/16	25/03/16
• Diseño del módulo de gestión masiva de metadatos de vídeos digitales	10	28/03/16	08/04/16
Implementación	**35**	**11/04/16**	**27/05/16**
• Algoritmo de extracción de átomos de un contenedor multimedia	20	11/04/16	06/05/16
• Módulo de tratamiento individual de vídeos digitales	5	09/05/16	13/05/16
• Módulo de gestión masiva de metadatos de vídeos digitales	10	16/05/16	27/05/16
Pruebas	**20**	**09/05/16**	**03/06/16**
• Algoritmo de extracción de átomos de un contenedor multimedia	7	09/05/16	17/05/16
• Módulo de tratamiento individual de vídeos digitales	5	19/05/16	25/05/16
• Módulo de gestión masiva de metadatos de vídeos digitales	5	30/05/16	03/06/16
Control	**126**	**11/12/15**	**03/06/16**

Tabla 1.2. Actividades de la fase de ejecución del proyecto

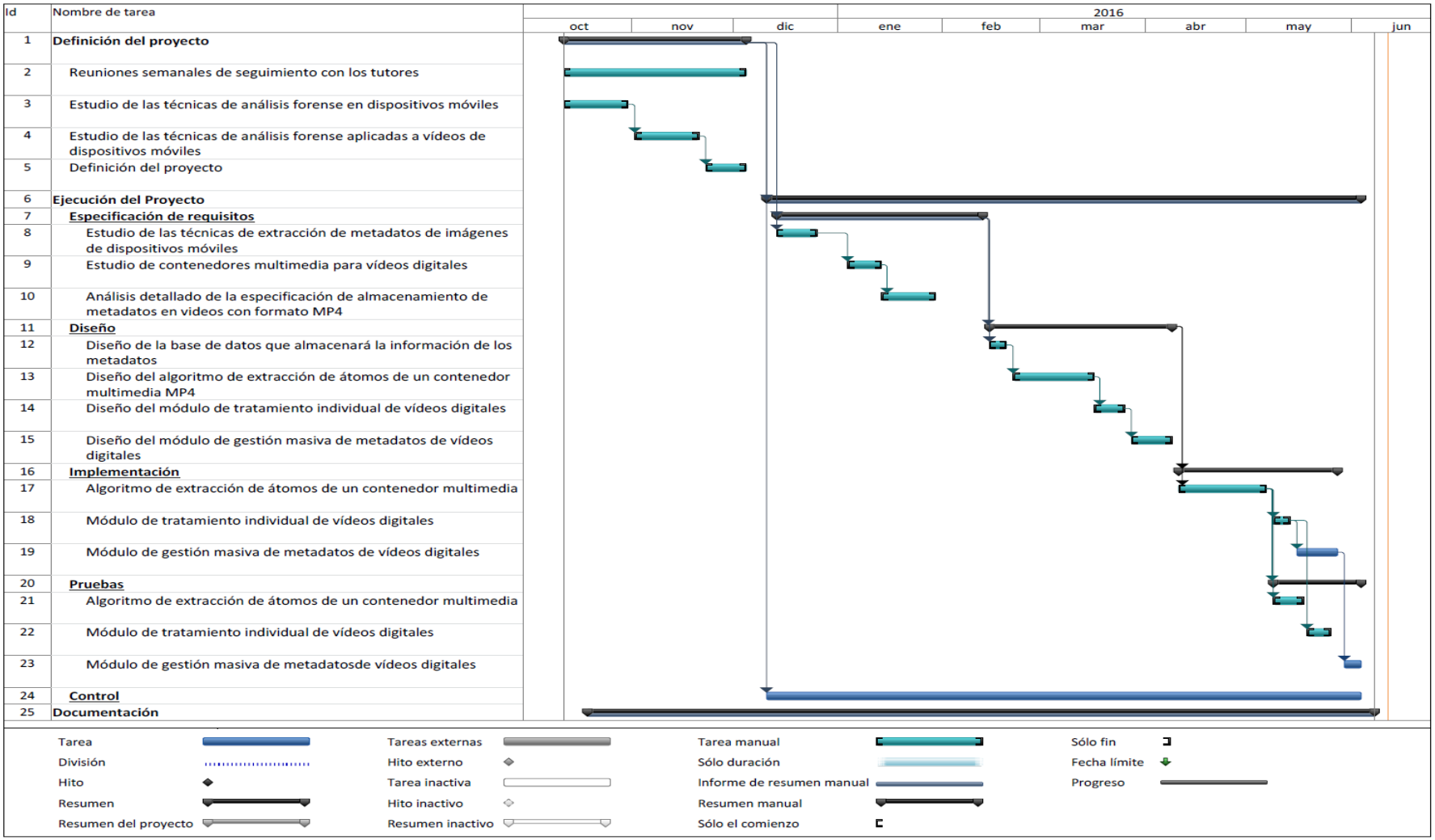


Figura 1.1: Diagrama de Gantt de la planificación del proyecto

1.4. Estructura de la memoria

El resto del trabajo está organizado en 7 capítulos con la estructura que se comenta a continuación.

En el capítulo 2 se presenta el proceso de generación de un vídeo, la metodología de compresión MPEG (*Moving Picture Expert Group*) y se realiza un estado del arte del análisis forense para vídeos generados por dispositivos móviles.

En el capítulo 3 se realiza una descripción de los principales sistemas de metadatos en vídeos digitales de dispositivos móviles, dando una especial importancia a la especificación del contenedor multimedia MP4 con compresión H.264 por su alto grado de utilización en los vídeos generados por dispositivos móviles.

En el capítulo 4 presenta la herramienta desarrollada para la extracción de metadatos de vídeos. Seguidamente se realiza una comparación de la herramienta con otras del mismo segmento.

En el capítulo 5 se realizan diversos análisis sobre un conjunto propio de vídeos de dispositivos móviles. Para éstos se utiliza la herramienta desarrollada.

En el capítulo 6 se presentan las principales conclusiones extraídas de este trabajo y las líneas de trabajo futuro.

En los capítulos 7 y 8 se realiza un resumen en inglés de la introducción y las conclusiones del trabajo.

Finalmente, en el anexo A se realiza una descripción en profundidad de los distintos módulos de la herramienta.

2. ANÁLISIS FORENSE EN VÍDEOS DE DISPOSITIVOS MÓVILES

2.1. Proceso de Generación de un Vídeo

Antes de mencionar alguna de las técnicas existentes para el análisis forense de vídeos de dispositivos móviles, es importante comprender cuál es el procedimiento realizado para generar un vídeo. Este proceso es similar en la generación de una imagen y se muestra esquemáticamente en la Figura 2.1. Adicionalmente, en un vídeo existe una fase en la que se codifica una secuencia de imágenes a lo largo del tiempo.

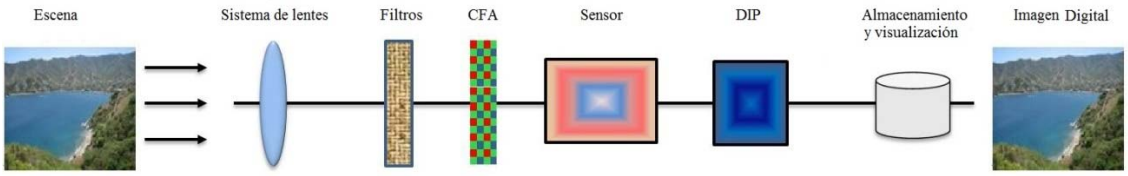


Figura 2.1 Proceso de adquisición de imágenes en cámaras digitales

Primeramente, el sistema de lentes captura la luz de la escena controlando la exposición, el foco y la estabilización de la imagen. Después, la luz pasa por un grupo de filtros que mejoran la calidad visual de la imagen, para a continuación, dejar pasar la luz al sensor de la imagen llamado: matriz de filtros de color (*Color Filter Array*, CFA), esta es una matriz de elementos sensibles a la luz denominados pixeles. La elección de la matriz CFA, puede influir en la nitidez y la apariencia final de la imagen ya que existen distintos patrones CFA el Green-Red-Green-Blue (GRGB) conocido como patrón de Bayer, Red-Green-Blue-Emerald (RGBE), Cyan-Yellow-Yellow-Magenta (CYYM), Cyan-Yellow-Green-Magenta (CYGM) o el Red-Green-Blue-White (RGBW). El patrón de Bayer es el modelo más comúnmente utilizado por las cámaras digitales.

La luz incidente sobre los filtros de color pasa a un sensor, que se encarga de generar una señal analógica proporcional a la intensidad de la luz recibida, guardando estos valores en una matriz interna. Las señales almacenadas por el

sensor CCD/CMOS posteriormente son convertidas en una señal digital y se transmiten al procesador de imagen, una vez el procesador de imagen recibe la señal digital elimina el ruido y otras anomalías introducidas. Algunos otros procesos que se aplican sobre la señal son la interpolación cromática, corrección gamma y corrección de color, entre otros.

2.1.1. Tipos de Sensores

Actualmente existen dos tipos de tecnologías de sensores utilizadas en las cámaras digitales: CCD (*Charge Coupled Device*) y CMOS (*Complementary Metal Oxide Semiconductor*). Ambos tipos de sensores están formados en esencia por semiconductores de metal-oxido MOS (*Metal Oxide Semiconductor*), estos sensores funcionan de forma similar, aunque la diferencia clave está en la forma en la que se digitalizan los pixeles y la forma en la que se lleva a cabo la lectura de las cargas. Los sensores CCD necesitan contar con un chip adicional para tratar la información de salida del sensor, haciendo esto que la fabricación de dispositivos sea más costosa y que los sensores sean más grandes. En contraste, los sensores CMOS cuentan con pixeles activos independientes, ya que ellos mismos realizan la digitalización ofreciendo velocidad, reduciendo el tamaño y el coste de los sistemas que integran una cámara digital. Otra diferencia entre estos dos tipos de sensores, es que los pixeles de una matriz CCD captan la luz simultáneamente, lo cual propicia una salida más uniforme. Los sensores CMOS realizan la lectura generalmente como barrido progresivo (evitando el efecto blooming). Los sensores CCD son muy superiores a los CMOS en el rango dinámico y términos de ruido, en contrapartida los sensores CMOS son más sensibles a la luz y en condiciones de poca iluminación se comportan mejor. En sus inicios los sensores CMOS eran algo peor que los CCD, pero hoy día es un mal que prácticamente esta subsanado. La tecnología CCD ha llegado a su límite y es ahora cuando se está desarrollando la CMOS y superando sus deficiencias, siendo así que la mayoría de los teléfonos inteligentes contienen sensores de tipo CMOS.

2.1.2. Metodología de compresión MPEG

Tras realizar el proceso anterior y exclusivamente para el caso de la generación de un vídeo existe un último paso que consiste en codificar los fotogramas resultantes para la creación de un archivo único final de vídeo. Esta codificación tiene como objetivo transformar todos los fotogramas capturados en una secuencia de ellos a lo largo de un tiempo. También se busca el conseguir un tamaño lo más óptimo posible del archivo final, ya que en un vídeo existen fotogramas capturados que son redundantes entre sí. Es decir, en ocasiones entre un fotograma y otro, se puede compartir características de la escena que facilitan el poder optimizar el tamaño del vídeo final sin perder contenido visual. Por ejemplo, para la codificación MPEG existe una estructura llamada GOP (*Group of Pictures*), mostrada en la Figura 2.2, que especifica el orden en el que las imágenes son ordenadas y soluciona el problema de redundancia en la codificación.

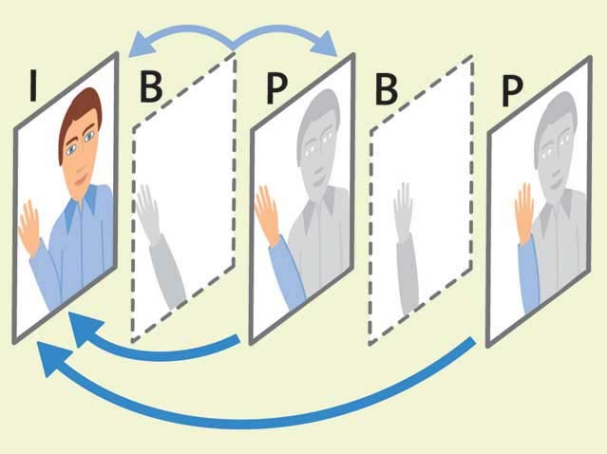


Figura 2.2 Grupo de imágenes (GOP)

Un GOP puede contener distintos tipos de imágenes:

- **Imágenes de codificación intra (I-Frames)**: Son imágenes de referencia que representan una imagen fija que son independientes de los otros tipos de imágenes.

- **Imágenes de codificación mediante predicción (P-Frames)**: Contienen información de la compensación de movimiento de la imagen precedente, ya sea de tipo P-Frame o I-Frame.

- **Imágenes codificación mediante predicción bidireccional (B-Frames)**: contienen diferente información de la imagen precedente y la siguiente.

Un GOP siempre empieza con una imagen tipo I-Frame, siguiéndole cualquier número de I-Frames y P-Frames, que se consideran como marcos de anclaje. Entre cada par de fotogramas consecutivos de anclaje pueden aparecer varias B-Frames como se observa en la Figura 2.3.

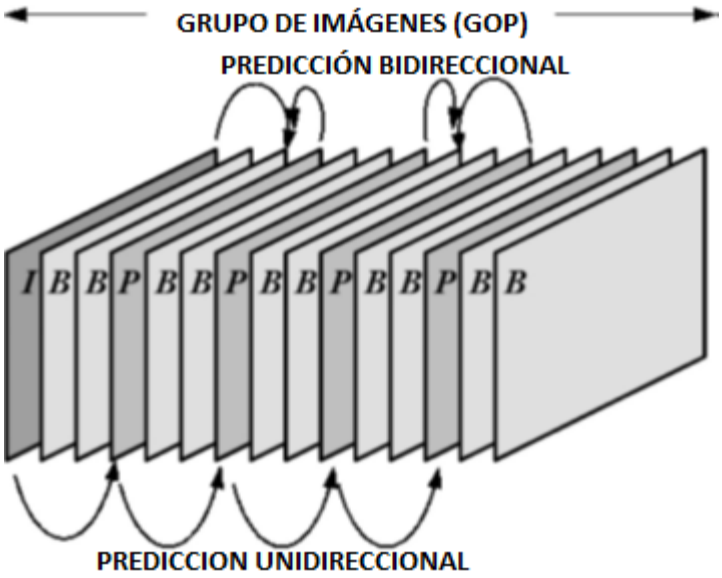


Figura 2.3 Estructura del grupo de imágenes

Una I-Frame no se refiere a ningún otro cuadro de vídeo y, por lo tanto, puede ser decodificado de forma independiente, proporcionando un punto de entrada para un rápido acceso aleatorio al vídeo comprimido. Por otro lado, la codificación de un cuadro P-Frame se basa en un marco de anclaje anterior, mientras que la codificación de un cuadro B-Frame se puede basar en dos marcos de anclaje, una anterior, así como un marco de anclaje posterior.

Cada imagen I-Frame se divide en una secuencia de macrobloques que no se

superponen. Cada macrobloque está completamente intracodificado, cada bloque se transforma a la frecuencia de dominio mediante la transformación discreta del coseno (DCT, del inglés *Discrete Cosine Transform*). Los coeficientes del DCT se cuantifican a continuación (con pérdidas), la entropía (longitud) y ejecutar Huffman (sin pérdida) se codifica para lograr la compresión.

El marco tipo de codificación (I-Frame, P-Frame o B-Frame), el número de bloques intra-codificados, y los coeficientes DC (Discrete Cosine) de cada DCT codificados pueden obtenerse a través del análisis sintáctico y la entropía (Huffman) de la decodificación MPEG. Esas operaciones tienen menos del 20% de la carga computacional involucrado en la decodificación completa de un archivo MPEG vídeo. Todo este proceso se detalla en [8].

2.2. Uso de la Formación de un Vídeo en el Análisis Forense

En general, las herramientas de edición de vídeo existentes no funcionan directamente en el dominio comprimido. Por lo tanto, el proceso de edición de una secuencia de vídeo se compone de tres pasos principales:

- La decodificación de la secuencia de entrada
- La edición del vídeo real
- La re-codificación del vídeo editado.

Conocer y entender este proceso es muy útil para el análisis forense ya que permite identificar posibles manipulaciones. Algunos investigadores ya están utilizando este proceso para tal fin.

En [9] se propone un método para identificar las regiones manipuladas en MPEG- 2 con sólo I-Frames. Posteriormente, los mismos autores proponen tener en cuenta la información sobre el error de movimiento al utilizar P-Frames, a fin de detectar eliminación o adición de marcos.

En [11], estiman el parámetro de cuantificación y los vectores de movimiento de marcos decodificados.

En [12] se propone un nuevo enfoque para la identificación de una secuencia de vídeo que se ha codificado doblemente. Este método funciona al recomprimir el vídeo bajo análisis con los tres posibles códecs y al calcular una medida de similitud entre las dos secuencias.

En [13] se propone un método para detectar si un vídeo se ha codificado dos veces, si este es el caso, estima el tamaño del grupo de imágenes (GOP) empleado durante la primera codificación. En la propuesta se utiliza una huella robusta y muy distintiva basada en la variación de los tipos de predicción de macrobloques en los P-frames recodificados. Una ventaja de esta variación de predicción de huella (VPF) es su presencia en el vídeo codificado dos veces sin la necesidad de re-compresión. Por otra parte, teniendo en cuenta que la VPF se hace evidente sólo en P-Frames que se intracodifican en la primera codificación no consideran el uso de B-Frames. Para tal fin, se limita la compresión que se realiza de acuerdo al perfil de base para H.264 y al perfil equivalente para MPEG-2 y MPEG-4. Estos perfiles de soporte sólo están en I-Frames y P-Frames, junto con los tres principales tipos de macrobloques: macrobloques intra-codificado (I-MB), macrobloques inter-codificados (P-MB) y macrobloques omitidos (S-MB). Un I-Frame codificado sólo puede contener macrobloques I-MB, mientras que P-Frames codificados pueden contener cualquiera de los macrobloques mencionados, es decir, I-MB, P-MB o S-MB. En general, la matriz de cuantificación o el factor de calidad para codificar un I-Frame difieren de la considerada para un cuadro P-Frame debido a que los I-Frames se utilizan de forma directa o indirectamente como referencia para codificar varias tramas futuras. Se concluye que si se puede detectar esas variaciones en el número de tipos de predicción I-MB y S-MB, entonces se puede de detectar si una doble codificación de la misma secuencia ha sido llevada a cabo y, si este es el caso, se puede estimar el tamaño de la primera GOP de esas variaciones. La VPF se

puede utilizar para detectar doble codificación y para estimar el tamaño de GOP de la primera compresión

En [14] se aborda la localización de falsificación en vídeos comprimidos en MPEG-2. El método propuesto se basa en el análisis de Doble de Cuantificación (DQ) que se traza en los I-Frames. Estos marcos se encuentran en el vídeo bajo análisis pudiendo estimar con ellos el tamaño del grupo de imágenes (GOP) que era utilizado en la primera compresión. Posteriormente, el análisis DQ se ideó para el esquema de codificación MPEG-2 y se aplica a los marcos que fueron intra-codificados tanto en la primera y segunda compresión. De tal manera, que las regiones que fueron manipuladas entre las dos decodificaciones se detecta. En comparación con los métodos existentes basados en el doble análisis de cuantificación, el esquema propuesto hace posible la localización de falsificaciones en una amplia gama de configuraciones. Este método permite determinar qué partes de un marco han sido alteradas. El método funciona básicamente mediante la búsqueda de rastros de doble cuantificación a nivel espacial, lo que permite la construcción de un mapa de grano fino de probabilidad de manipulación para cada marco analizado. Se centran en el escenario de falsificación con marcos intra-codificados, y suponen que, a partir de una secuencia de vídeo MPEG-2, el atacante decodifica el vídeo, altera la contenido de un grupo de tramas, y finalmente se codifica nuevamente la secuencia resultante usando MPEG-2 con un tamaño diferente.

2.3. Trabajos Relacionados

La mayor parte de las investigaciones realizadas en el campo de la identificación de la fuente se han realizado para imágenes fotográficas estáticas. Sin embargo, la investigación científica requiere soluciones a los temas forenses relacionados con las señales de vídeo debido a sus peculiaridades y la amplia gama de posibles alteraciones que se pueden aplicar a ellos. La mayoría de las técnicas de análisis forense en este campo, que se pueden aplicar a una imagen

pueden ser aplicadas a los diferentes fotogramas de un vídeo [30].

El análisis forense de imágenes digitales se puede dividir en dos grandes ramas [31]: autenticidad de imágenes digitales e identificación de la fuente de adquisición de una imagen.

La primera de las ramas trata de discernir si una imagen ha sufrido algún procesamiento posterior al de su creación, es decir, que no haya sido manipulada. La segunda de las ramas pretende identificar el tipo o clase de fuente que generó la imagen digital. Dentro de esta segunda rama puede realizarse una subdivisión en dos grupos: identificación del tipo de dispositivo fuente (cámara, escáner, generadas por computador,...) o identificación de la marca y modelo del dispositivo.

Para el diseño de técnicas y algoritmos en cualquiera de estas ramas se aprovechan algunas características especiales de las imágenes o vídeos que sirven como herramienta para el análisis forense. En [7, 32] se realiza un estudio de las características que pueden ser objeto de análisis forense en dispositivos móviles. El mayor problema con este enfoque es que los diferentes modelos de las cámaras digitales usan componentes de un número reducido de fabricantes y que los algoritmos que usan para la generación de las imágenes y vídeos también son muy similares entre modelos de la misma marca.

En [33] se realiza una comparación detallada de los principales grupos de técnicas de identificación de fuente de adquisición. Estas se dividen en cinco grupos y están basadas en: metadatos, características de la imagen, defectos de la matriz CFA e interpolación cromática, imperfecciones del sensor y las transformadas wavelet.

Aun teniendo en cuenta estas dos grandes ramas no se puede dejar pasar por alto la información de los metadatos que los dispositivos introducen en el proceso de adquisición de la fotografía. Suponiendo la veracidad de los datos

contenidos en la imagen, es decir, que no se hayan dado manipulaciones mal intencionadas a posteriori, dependiendo de cada fabricante y dispositivo se arroja en una diversidad de formatos, una información útil para el analista forense (localización GPS, fuente de la foto, características técnicas de la imagen, etc.).

Los archivos de imágenes digitales pueden ser modificados además de una forma más o menos elaborada por cualquier usuario. Existen una gran cantidad de programas de edición accesibles a cualquier tipo de usuario que permiten modificar este tipo de contenido digital siendo muchas veces estos cambios imperceptibles para el ojo humano. Igualmente al caso anterior, estos cambios pueden ser intencionados o malintencionados, pero independientemente de la fe con la que se realizó el cambio, la imagen pierde su originalidad con respecto a la generación por parte de la fuente de adquisición.

Estas situaciones pueden generar problemas o indefiniciones cuando las imágenes son utilizadas como evidencias en algún proceso, ya sea judicial o no, dado que no se puede garantizar la identificación de la fuente de adquisición del contenido o la no manipulación del mismo sin realizar un análisis forense previo.

A pesar de las debilidades de este tipo de técnicas, si existen en el archivo los metadatos y de alguna manera se logra comprobar que no han sufrido modificaciones externas, su uso es de gran utilidad para los analistas forenses. Existe información difícilmente inferible del propio contenido de la imagen como por ejemplo la información GPS o la fecha y hora de la toma de la imagen, entre muchas otras. Sin embargo, estas técnicas dependen en gran medida de los metadatos que los fabricantes deciden insertar cuando la imagen es generada y la corrección en seguimiento de la especificación o estándar de metadatos que utilice. [34, 35] realizan un estudio a fondo, donde se demuestra que los fabricantes no siguen fielmente la especificación Exif. Esto puede

conllevar la extracción de información errónea o inválida para fines forenses. Asimismo, este método es el más vulnerable a modificaciones malintencionadas, e incluso se puede dar el caso de la eliminación total de los metadatos, ya sea intencionadamente o de manera inconsciente.

Ejemplos de ello son algunos programas de edición fotográfica, que al editar o comprimir una imagen, actualizan incorrectamente los metadatos o provocan la pérdida de los mismos.

En el caso del desarrollo de técnicas de análisis forense en vídeo, existen pocas referencias al respecto. Algunas se basan directamente en la secuencia de codificación y otras en la extracción de fotogramas aplicando a algún método de clasificación para imágenes fijas [36, 37].

3. METADATOS EN VÍDEOS DIGITALES DE DISPOSITIVOS MÓVILES

3.1. Contenedores Multimedia

Los metadatos o "datos sobre datos" registran información relacionada con las condiciones de captura de la imagen o vídeo, como fecha y hora de generación, presencia o ausencia de flash, distancia de los objetos, tiempo de exposición, apertura del obturador e información GPS (*Global Position System*), entre otras. En otras palabras, información de interés que complementa el contenido principal de un documento digital. Los metadatos, entre otros usos, pueden llegar a ser una gran ayuda para la organización y búsqueda en librerías de imágenes y vídeos.

Los formatos utilizados para almacenar las imágenes digitales son: TIFF (*Tagged Image File Format*), JPEG (*Join Photograph Expert Group*) [15] o PSD (*Photoshop Document*). Asimismo, cada formato puede tener diferentes contenedores de metadatos, entre los que se destacan los siguientes: IFD (*Image File Directory*) Exif (*Exchangeable Image File Format*), TIFF, Adobe XMP (*Extensible Metadata Platform*), e IPTC-IIM (*International Press Telecommunications Council- Information Interchange Model*). El contenedor más utilizado para metadatos de imágenes de cámaras digitales es Exif [3].

La especificación Exif [16] es la más utilizada para identificación de la fuente. Entre los cientos de etiquetas que incluye la especificación se encuentran la marca y modelo de la cámara. Sin embargo, cabe destacar que la propia especificación no hace obligatoria su existencia en los archivos.

Por su parte, los vídeos digitales son almacenados en una amplia variedad de formatos, denominados "contendores multimedia", que almacenan información de vídeo, audio, metadatos e información de sincronización y corrección de

errores siguiendo un formato preestablecido en su especificación técnica. Como su propio nombre indica, contendor multimedia es un archivo que contiene en su interior varios elementos (como mínimo las pistas de vídeo y audio) [17][18]. Algunos contenedores también permiten incluir otros elementos como imágenes o subtítulos integrados, sin necesidad de archivos externos.

Las pistas de vídeo y audio normalmente están comprimidas a través de los diferentes códecs de cada uno de los contenedores multimedia. Estos códecs son los encargados de descomprimir la información para su posterior reproducción. Los códecs son la base para que todos los dispositivos actuales sean capaces de capturar o reproducir un archivo que contenga imágenes y sonido en su interior. Dependiendo del códec elegido se obtiene una mejor o peor calidad, así como, un mayor o menor tamaño. Asimismo, como ocurre con el vídeo, hay canales de audio incluidos en el archivo. También suelen estar comprimidos con un códec determinado para ahorrar espacio.

La Tabla 3.1 presenta los contenedores multimedia más conocidos con los códecs de vídeo y audio que utilizan y especificando si contienen metadatos almacenados. En resumen, no solo es necesario conocer el formato del contenedor para poder separar las pistas de vídeo y de audio, sino que también es necesario poder decodificarlas. Los contenedores más populares hoy en días son: MP4 (MPEG-4) [19], MOV (archivo QuickTime de Apple) [20], AVI (*Audio Vídeo Interleave de Microsoft*) [21] y MKV (Matroska).

Todos los vídeos se componen de pequeñas estructuras llamadas átomos, para almacenar toda la información comentada anteriormente. A continuación se detalla la estructura de un átomo y los principales átomos presentes en un vídeo.

Contenedor	Propietario	Formato de codificación de video	Formato de codificación de audio	metadatos
3GP	3GPP	H.263,MPEG-4 Part 2, H.264/MPEG-4 AVC	AMR-NB, AMR-WB,AMR-WB+, AAC, HE-AAC and HE-AAC v2	?
3G2	3GPP2	H.263,MPEG-4 Part 2, H.264/MPEG-4 AVC	AMR-NB, AMR-WB,AAC, HE-AAC, EVRC,EVRC-B, EVRC-WB, 13K (QCELP), SMV orVMR-WB	?
ASF	Microsoft	VFWor DMO	todos a través de ACM o DMO	Sí
AVI	Microsoft	VFW	todos a través de ACM	Sí
divx	DivX, Inc.	MPEG-4 Part 2	MP3, PCM, AC-3	?
EVO	MPEG	MPEG-2 Part 2,H.264/MPEG-4 AVC, VC-1	AC-3, E-AC-3, Dolby TrueHD, Linear PCM, DTS, DTS-HD, MPEG-2 Part 3	?
F4V	Adobe Systems	H.264/MPEG-4 AVC	MP3, AAC, HE-AAC[7]	Sí
FLV	Adobe Systems	Sorenson,VP6, H.264/MPEG-4 AVC	MP3, Nellymoser, ADPCM, Linear PCM,AAC, Speex	Sí
Matroska	CoreCodec	Virtually anything	Cualquiera virtualmente	Sí
MP4	MPEG	MPEG-2 Part 2, MPEG-4 ASP, H.264/MPEG-4 AVC, H.263, VC-1,Dirac	MPEG-2/4 (HE)-AAC, MPEG-1/2 Layers I, II, III (MP3), AC-3, Apple Lossless, ALS, SLS	Sí
MPG/MPEG	MPEG	MPEG-1,MPEG-2	MPEG-1 Layers I, II, III (mp3)	No
MXF	SMPTE	Virtually anything	Cualquiera virtualmente	Sí
Mov / QT	Apple	MPEG-2, MPEG-4 Part 2, H.264, H.263, H.261, Apple ProRes, Apple Pixlet, Cinepak, , DV, DVC Pro 50, Graphics, Motion JPEG, Photo JPEG, QuickTime Animation, Sorenson Video 2, Sorenson Video 3	AAC, HE-AAC, Apple Lossless, MP3, AMR Narrowband, MS ADPCM, QDesign Music 2, QCELP, IMA 4:1, MACE 3:1	Sí
RMVB	RealNetworks	RealVideo 8, 9, 10	(HE)-AAC, Cook Codec, Vorbis	?
VOB+IFO	DVD Forum	MPEG-2 Part 2, MPEG-1 Part 2	AC-3, Linear PCM,DTS, MPEG-2 Part 3,MPEG-1 Layer II	No

Tabla 3.1. Contenedores multimedia de vídeos

3.2. Estructura de un Átomo

La estructura elemental de un vídeo es el átomo. Los metadatos, el vídeo y el sonido de un vídeo se encuentran dentro de ellos. Los átomos son de naturaleza jerárquica. Es decir, un átomo puede contener otros átomos, que pueden contener aún otros, y así sucesivamente.

El tipo de átomo viene especificado por un entero sin signo de 32 bits, típicamente interpretado como un código ASCII (*American Standard Code for Information Interchange*) de cuatro caracteres normalmente en letras minúsculas

El formato de los datos almacenados dentro de un átomo no siempre puede ser determinado sólo por el campo "type" del átomo. El tipo del átomo padre también puede ser importante. En otras palabras, un tipo de átomo puede contener diferentes tipos de información en función de su átomo raíz.

Los átomos dentro de los átomos de contenedores no tienen que estar en ningún orden en particular, aunque como se observará más adelante suelen seguir el mismo esquema.

La cabecera del átomo tiene los campos tamaño ("size") y tipo ("type"), que indican el tamaño del átomo en bytes y su tipo. Adicionalmente, el átomo puede contener un campo de tamaño ampliado ("extended size"), indicando que el tamaño de un átomo grande como un entero de 64 bits. Si un campo de tamaño ampliado está presente, el campo de tamaño ("size") se establece en 1. El tamaño real de un átomo no puede ser menor de 8 bytes, siendo éste el tamaño mínimo de los campos tipo y tamaño.

Algunos átomo también contienen los campos versión ("versión") y bandera ("flags"). Los campos "flag" y "versión" no son tratados como parte de la cabecera del átomo en este documento, sino como campos de datos específicos para cada tipo de átomo que los contiene. Tales campos siempre deben ser

puestos a cero, a menos que se especifique lo contrario.

3.2.1. Cabecera de un Átomo

- **Tamaño ("Size")**: Un número entero de 32 bits que indica el tamaño del átomo, que incluye tanto la cabecera del átomo como su contenido, incluyendo todos los átomos contenidos. Normalmente, el campo tamaño ("size") contiene el tamaño real del átomo, en bytes, expresado como un número entero sin signo de 32 bits. Sin embargo, el campo tamaño ("size") puede contener valores especiales que indican un método alternativo para determinar el tamaño del átomo. Estos valores especiales se utilizan normalmente sólo para los átomos media data ("mdat"). Hay dos valores especiales válidos para el tamaño del campo:

  - **0**: que permite, sólo por un átomo de nivel superior, designar el último átomo en el 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 tamaño extendido ("extended size"), un campo de 64 bits opcional que sigue al campo tipo ("type"). Esto es utilizado por átomos con datos que contienen más de 2^{32} bytes.

  En la Figura 3.1 se puede observar de manera visual los posibles valores del campo "size" y el significado de cada caso.

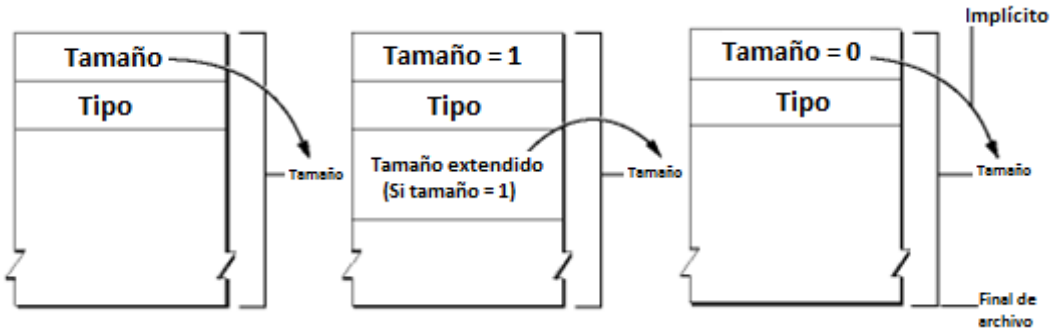


Figura 3.1: Estructura de los átomos de un vídeo MP4

- **Tipo (“Type”)**: Un número entero de 32 bits que contiene el tipo de átomo. Esto a menudo puede ser tratado de manera útil como un campo de cuatro caracteres con un valor mnemotécnico, tales como 'moov' (0x6D6F6F76) por un átomo de película, o 'trak' (0x7472616B) para un átomo de pista, pero los valores no ASCII (por ejemplo, 0x00000001) también se utilizan.

Conocer el tipo de un átomo permite interpretar sus datos. Éstos datos se pueden organizar como cualquier colección arbitraria de campos, tablas u otros átomos. La estructura de datos es específica para el tipo de átomo. Un átomo de un tipo dado tiene una estructura de datos definida. Si se encuentra un átomo de un tipo desconocido, no se debe intentar interpretar los datos del átomo. Se utiliza el campo tamaño del átomo para omitir este átomo y todo su contenido. Esto permite un grado de compatibilidad hacia adelante y poder seguir analizando el resto de átomos del vídeo.

La estructura interna de un determinado tipo de átomo puede cambiar cuando se introduce una nueva versión. Hay que comprobar siempre el campo de versión, si es que existe. Nunca se debe interpretar los datos que se hallen fuera del átomo, tal como se define por los campos tamaño (“size”) o tamaño ampliado (“extended size”).

En la Tabla 3.2 se muestran los principales átomos que se pueden encontrar en un vídeo y el uso de cada uno de ellos. Estos átomos son considerados átomos raíz de un vídeo ya que no tienen ningún átomo padre.

Tipo de átomo	Uso
ftyp	Tipo de compatibilidad de archivo, identifica el tipo de archivo y lo diferencia de los tipos de archivos similares, tales como archivos MPEG-4 y JPEG-2000.
moov	Película de metadatos de recursos sobre la película (número y tipo de pistas, localización de datos de la muestra, y así sucesivamente). Describe donde se pueden encontrar y cómo se interpretan los datos de la película.
mdat	Muestras de datos media de la muestra de películas tales como marcos y grupos de muestras de audio de vídeo. Por lo general, estos datos se pueden interpretar sólo mediante el uso del recurso de película.
free	El espacio no utilizado disponible en el archivo.

Tabla 3.2: Átomos principales y su uso

3.3. Especificación del Contenedor Multimedia MP4 con Compresión H.264 y ACC.

El formato MP4 [28] forma parte del estándar MPEG-4 parte 14 y se utiliza para distribuir vídeo y audio usando una compresión H.264 AVC (*Advanced Vídeo Coding*) [19] para vídeo y AAC (*Advanced Audio Coding*) para audio, pero también puede almacenar otro tipo de datos (subtítulos, información de capítulos e imágenes fijas, entre otros). La extensión asociada a este contenedor es .MP4, pero no es poco frecuente encontrar archivos de audio que lleven la extensión .M4A, que es la extensión adoptada por Apple para distribuir vídeos en iTunes y su reproductor iPod.

La gran mayoría de los archivos de audio de MP4 están comprimidos con el formato AAC, aunque también admite compresión en MP3. En condiciones normales no supone un problema cambiar la extensión de un archivo .M4A a .MP4 manualmente si ello ayuda a trabajar más cómodamente con él. También se puede encontrar archivos de vídeo con las extensiones .M4V o .MP4V. Asimismo, H.264 / MPEG-4 AVC [29] es el códec más utilizado en las videocámaras modernas, cámaras digitales de dispositivos móviles y tabletas que almacenan los vídeos capturados en discos duros, tarjetas de memorias, etc. Este formato permite codificar en casi cualquier resolución, desde la más pequeña que permite un iPod, HD (*High Definition*), 2K, hasta la reciente 4K con una excelente calidad y archivos de pequeño tamaño relativo.

H.264 se ha convertido en el códec de vídeo más popular en los últimos años gracias a su relación calidad/tamaño, ideal para el uso de contenidos en calidad HD. Se usa igualmente en televisores, Smartphone, lectores Blu-ray o vídeos de YouTube [29]. H.264/MPEG-4 AVC es un proyecto del ITU-T (*International Telecommunication Union - Telecommunication*) y el MPEG con el objetivo de lograr buena calidad de imagen que admita resoluciones altas con poca complejidad y tasa binarias inferiores que las normas anteriores, aunque sin compatibilidad con ellas.

Debido al gran número de vídeos que se realizan hoy día y a la necesidad de saber su estructura para el análisis forense de los mismos se ha investigado, analizado y posteriormente creado una especificación para la estructura del contenedor multimedia MP4 con códec de compresión H.264 de vídeos capturados con dispositivos móviles Android.

Este análisis es de suma importancia ya que conociendo la estructura del vídeo se puede obtener información que sirva de evidencia de manipulación del mismo y adquirir los conocimientos necesarios para crear una herramienta de extracción automática de la información contenida en un vídeo.

Antes de comenzar con la especificación cabe destacar que los átomos que se detallan a continuación, tanto para éste como para el resto de formatos, pueden o no aparecer en un vídeo. Asimismo, estos átomos pueden seguir o no el mismo orden dado que algunos de ellos son átomos opcionales y dependen de si el fabricante del dispositivo los ha introducido o no, siendo igual al orden en que se encuentran los átomos.

A continuación se especifica la estructura y los átomos más comunes del formato MP4, pudiendo éstos aparecer o no, cambiar su orden o incluso aparecer otros átomos no especificados.

La figura 3.2 muestra la estructura de los átomos de un vídeo MP4.

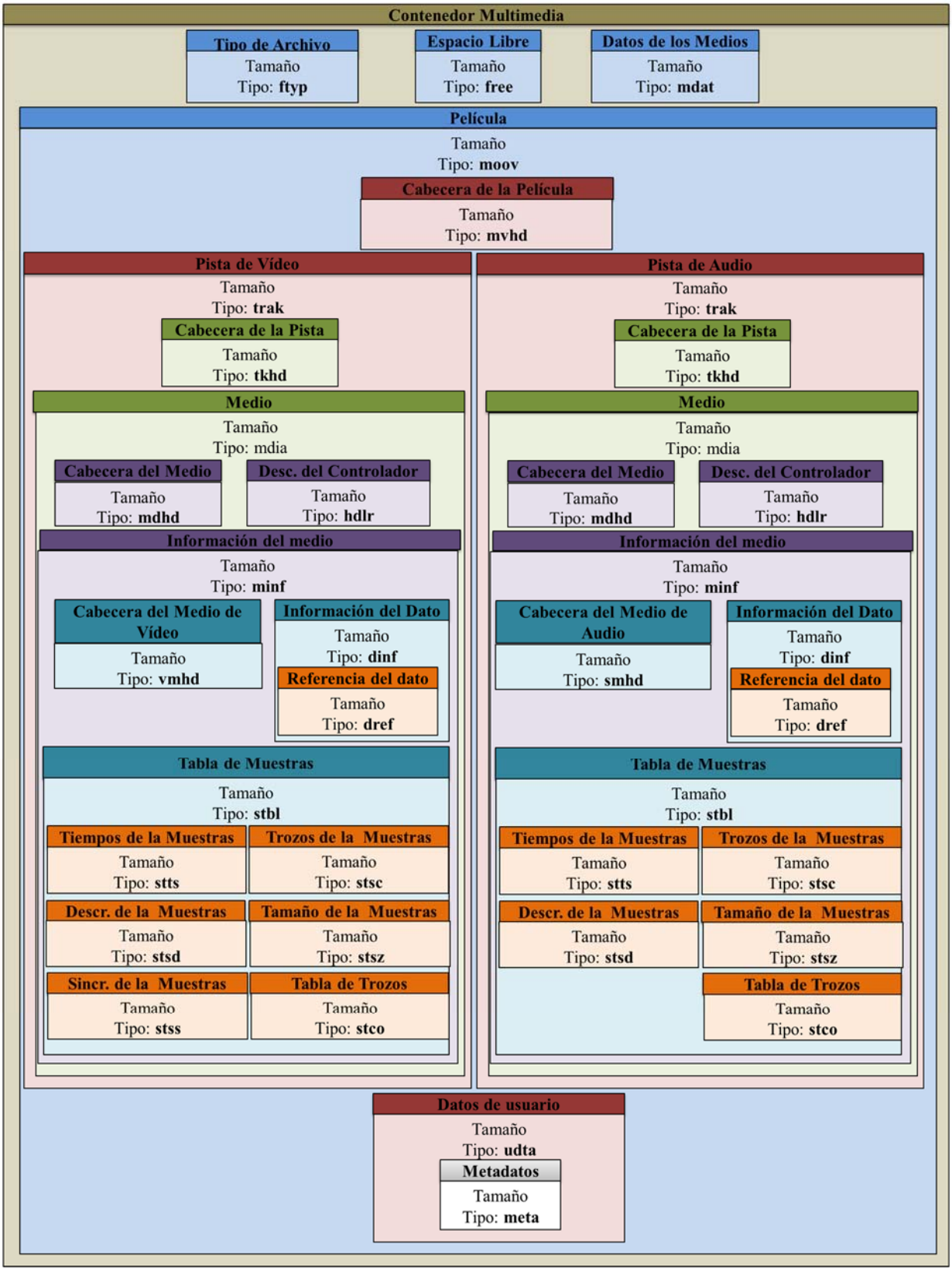


Figura 3.2: Estructura de los átomos de un vídeo MP4

3.3.1. Átomo File Type – "ftyp"

Este átomo muestra el formato compatible del archivo, pudiendo haber más de uno en cuyo caso indica cuál es su tipo preferido. Es un átomo opcional pero muy recomendable y si aparece debe ser el primer átomo significativo del archivo. Contiene los campos que se indican en la Tabla 3.3.

Size	Type	Major Brand	Minor Version	Compatible Brands
	66747970 (ftyp)			
32 bits	32 bits	32 bits	32 bits	Lista de 32 bits

Tabla 3.3: Estructura del átomo: ftyp

- **Size**: Un entero sin signo de 32 bits que especifica el número de bytes en este átomo "File Type".
- **Type**: Un entero sin signo de 32 bits que identifica el tipo de átomo, representado como un código de cuatro caracteres; este campo debe establecerse a "ftyp" (0x66747970).
- **Major Brand**: Un entero sin signo de 32 bits que identifica el tipo de archivo de película, representado como un código de cuatro caracteres; Si un archivo es compatible con múltiples marcas, todas estas marcas se encuentran en los campos "Compatible_Brands", y la "Major_Brand" identifica la marca preferida o de mejor uso.
- **Minor Version**: Un entero de 32 bits sin signo que identifica el tipo de archivo de película "Minor Version", representado como un número de cuatro bytes representado en forma decimal codificado en binario (BCD); lo que indica año y mes, seguido de un decimal codificado en binario cero. Por ejemplo, para el "Minor_Versión" de junio de 2004, este campo se establece en los valores de la BCD 20 04 06 00.
- **Compatible Brands**: Una serie de enteros sin signo de 32 bits, representados como un código de cuatro caracteres cada uno, que anuncia los formatos de archivo compatibles.

3.3.2. Átomo Free - “free”

Este átomo muestra el espacio no utilizado en el archivo, pudiéndose éste sobrescribirse si fuera necesario. Contiene los campos que se indican en la Tabla 3.4.

Size	Type	Free Space
	66726565 (free)	
32 bits	32 bits	

Tabla 3.4: Estructura del átomo: free

- **Size**: Un entero sin signo de 32 bits que especifica el número de bytes en este átomo “Free”.
- **Type**: Un entero sin signo de 32 bits que identifica el tipo de átomo, representado como un código de cuatro caracteres; este campo debe establecerse a "free" (0x66726565).
- **Free Space**: Contiene los bytes de espacio libre; Estos bytes valen todos 0.

3.3.3. Átomo Media Data - “mdat”

Este átomo contiene los medios tales como marcos y grupos de muestras de audio y video, por lo general, estos datos solo se pueden interpretar mediante el uso del recurso de película (átomo moov). Contiene los campos que se indican en la Tabla 3.5.

Size	Type	Data
	6D646174 (mdat)	
32 bits	32 bits	

Tabla 3.5: Estructura del átomo: mdat

- **Size**: Un entero sin signo de 32 bits que especifica el número de bytes en esta porción de datos media del archivo de película MP4.
- **Type**: Un entero sin signo de 32 bits que identifica el tipo, representado como un código de cuatro caracteres; este campo debe estar a “mdat” (0x6D646174).
- **Data**: Contiene los datos de audio y vídeo de la película.

La Figura 3.3 muestra la estructura de los átomos “ftyp”, “mdat” y “free” de un vídeo MP4

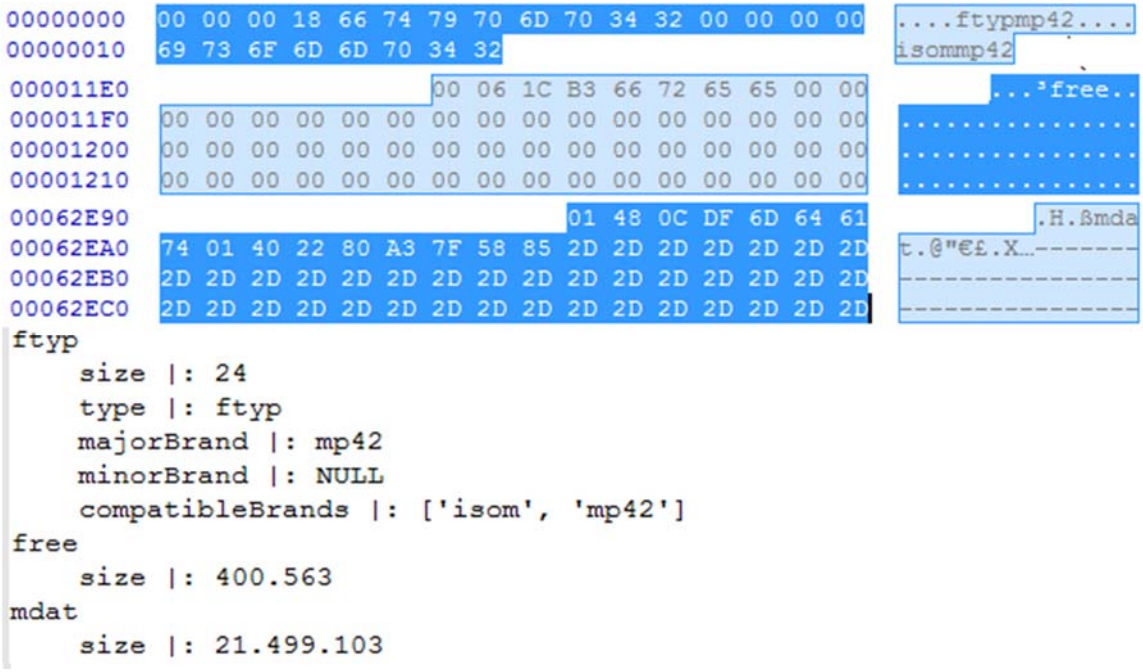


Figura 3.3: Estructura de los átomos “ftyp”, “mdat”, y “free”

3.3.4. Átomo Movie - “moov”

Este átomo describe dónde encontrar los datos de la película y cómo interpretarlos. Contiene los campos que se indican en la Tabla 3.6.

Size	Type	mvhd	trak	trak	udta
	6D6F6F76 (moov)				
32 bits	32 bits	átomo	átomo	átomo	átomo

Tabla 3.6: Estructura del átomo: moov

- **Size**: Un entero sin signo de 32 bits que especifica el número de bytes en este átomo de película.
- **Type**: Un entero sin signo de 32 bits que identifica el tipo, representada como un código de cuatro caracteres; este campo debe estar a “moov” (0x6D6F6F76).

La figura 3.4 muestra los átomos que contiene el átomo “moov”.

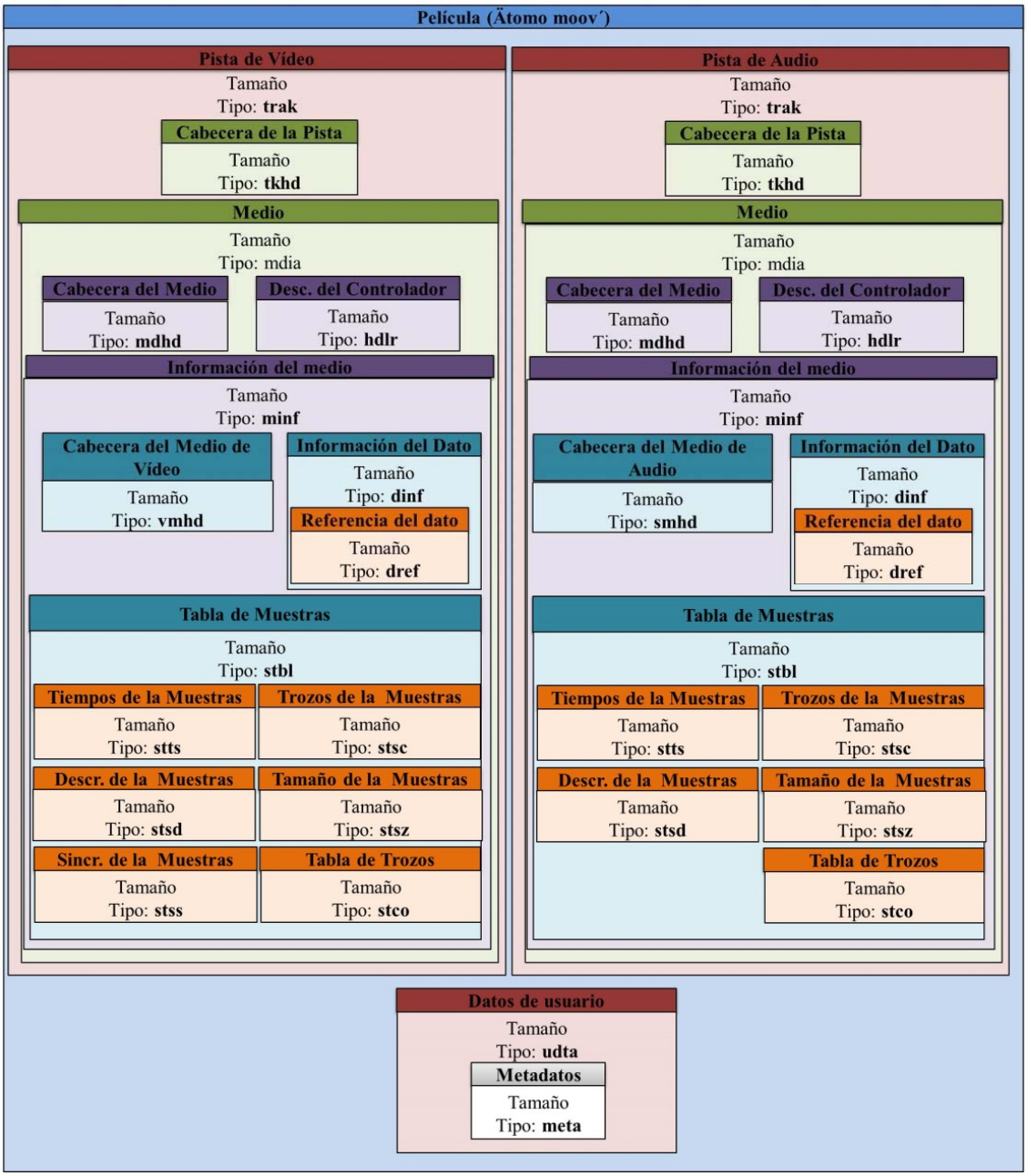


Figura 3.4: Estructura de los átomos de un vídeo MP4

4. HERRAMIENTA PARA LA EXTRACCIÓN AUTOMÁTICA DE METADATOS EN VÍDEOS DE DISPOSITIVOS MÓVILES

4.1. Generalidades de la Herramienta

Dado que el proceso necesario para el análisis binario manual de los metadatos en vídeos es lento y tedioso, es necesaria la existencia de herramientas para la extracción automática y su visualización de forma gráfica. Este tipo de herramientas sirven de apoyo al analista forense en las tareas de análisis de los metadatos. Sin el uso de este tipo de aplicaciones sería complejo realizar el procesamiento para un gran número de vídeos.

En este trabajo se ha desarrollado una herramienta que facilita la extracción y el tratamiento de metadatos en vídeos con formato MP4. La herramienta desarrollada detecta inconsistencias en la información de los átomos con respecto a la especificación. Además, muestra la localización GPS en Google Maps si el vídeo dispone de tal información.

A grandes rasgos la herramienta se divide en dos grandes módulos:

- **Tratamiento de vídeos a nivel individual.** Este módulo permite obtener la información detallada de todos los átomos existentes en un vídeo de forma individual desde un fichero con formato MP4. Adicionalmente, se apoya de Google Maps y Google Earth para situarla geográficamente en el mapa, si el vídeo posee información de geoposicionamiento. A la hora de mostrar la información de los átomos se creado una estructura de árbol organizado por los átomos de la especificación.

- **Tratamiento de vídeos a nivel grupal.** En este módulo se almacena toda la

información almacenada en los vídeos en una base datos para posteriormente realizar consultas diversas consultas sobre la información de distintos proyectos o conjuntos de vídeos. Asimismo, como en el caso del módulo anterior, se puede situar gráficamente en Google Maps los vídeos que posean información de geoposicionamiento.

En el anexo B se muestra el funcionamiento de la herramienta desarrollada.

4.2. Diseño e implementación de la herramienta

Para el diseño de la base de datos se ha usado MySQL y para la implementación se ha utilizado Python. Finalmente, el sistema operativo (SO) utilizado para el desarrollo del proyecto es Linux con distribución Debian Jessie.

La base de datos se ha diseñado utilizando el sistema de gestión de bases de datos relacional MySQL, así como MySQL Workbench para administrar la base de datos de forma gráfica. La Tabla 4.1 presenta las tablas que conforman la base de datos.

Se ha utilizado el lenguaje de programación Python en su versión 2.7.6 ya que tiene una licencia de código abierto (*Python Software Foundation License).* Esta licencia es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1

Para la implementación de la misma se han utilizado las siguientes librerías:

- PyGTK, para la parte visual de la aplicación, la cual se ha diseñado previamente con Glade 3.6.1 y convertido a Python con Tepache.
- Scipy, numpy, pyWavelets y Python Imaging Library PIL 1.1.7.
- PyQt4 para el reproductor de vídeo y OpenCV para la extracción y reproducción de fotogramas del vídeo.

Tabla	Descripción
Diccionario de Tablas	Contiene información de todas las tablas utilizadas, su nombre, una descripción breve y una más completa.
Diccionario de campos	Almacena la información de todos los campos de todas las tablas, como: tabla a la que pertenecen, nombre del campo, tipo, longitud, descripción corta, una descripción extensa, título del campo, etc.
Datos de Configuración	Almacena toda la información necesaria para la ejecución de la herramienta, como colores o el valor máximo del buffer entre otros
Proyecto	Guarda toda la información necesaria de un proyecto.
Información general del vídeo	Guarda toda la información del vídeo y todos los átomos que estén fuera de alguna de las pistas, tanto de vídeo como de audio. Almacena el identificador del proyecto al que pertenece, el nombre del archivo, su path original, un identificador del vídeo, el tipo de archivo, el thumbnail del vídeo y la información los átomos pertenecientes a esta tabla, es decir, todos los que quedan fuera de alguna de las pistas.
Pista de vídeo	Guarda toda la información relativa a la pista de vídeo. Almacena la información de todos los átomos de esta pista
Pista de audio	Guarda toda la información relativa a la pista de sonido. Almacena la información de todos los átomos de esta pista
Átomos desconocidos	Tabla para los átomos desconocidos que encuentre dentro de cada vídeo, que guarda el identificador del vídeo al que pertenece, el path del átomo, el tipo del átomo y sus datos.
Inconsitencias	Tabla para las incongruencias con la especificación que encuentra en cada vídeo, donde almacena el identificador del vídeo donde se ha encontrado, el path del átomo donde se ha identificado y el mensaje generado, donde se especifica cuál ha sido la incongruencia encontrada.

Tabla 4.1: Base de datos de la herramienta

4.3. Comparativa con otras Herramientas

Para realizar una comparativa de la herramienta desarrollada con otras con fines similares, se han buscado principalmente herramientas de extracción y tratamiento de metadatos en vídeos para archivos mp4, aunque no ha sido un criterio que excluya a otro tipo de herramientas relacionadas.

Gspot 2.7 [23], MediaInfo 0.7.81 [24], ExifTools 10.16 [25]. Se han seleccionado estas herramientas entre otras porque son las más completas y porque todas ellas admiten vídeos con formato MP4, ya que hay algunas que o bien los metadatos extraídos son insuficientes o porque no admiten este formato como ocurre con VideoInspector [26] o SUMo [27].

Se ha procedido a analizar el mismo vídeo con las tres aplicaciones indicadas obteniendo los siguientes resultados.

4.3.1. GSpot

Cuenta con interfaz gráfica donde se selecciona el archivo de vídeo y extrae una serie de datos divididos en 5 bloques:

En el primer bloque "File" se encuentra el Path y tamaño del archivo. En el segundo bloque "Container", se encuentra la información del formato utilizado: formato, tipo de compatibilidad, versión, fecha de creación y modificación. En el tercer bloque los datos de usuario, que este vídeo en concreto no cuenta con ninguno. En el cuarto bloque se encuentra los datos referentes a la pista de audio: códec, velocidad, hercios y tipo de audio. Y en el quinto bloque analiza el códec de vídeo, duración, números de fotogramas, velocidad, Qf, número de fotogramas por segundo, resolución y relación de aspecto.

Esta herramienta no ofrece la posibilidad de análisis grupal de vídeos para un posterior tratamiento. Tampoco ofrece la posibilidad de reproducir dentro de la misma aplicación el vídeo que estás analizando, ni ofrece el thumbnail al menos

del vídeo. Tampoco cuenta con la posibilidad de poder localizar el vídeo en un mapa y respecto a los datos extraídos observamos que sólo extrae unos pocos datos de todos los que contiene un vídeo.

Cabe destacar que esta herramienta extrae los datos más interesantes posiblemente para un usuario y cuenta con un interfaz bastante intuitiva y muy bien organizada, pero para el análisis forense esta herramienta puede que no sea la más adecuada debido a la limitación de metadatos extraídos. Su aspecto y los datos extraídos se pueden observar en la Figura 4.1.

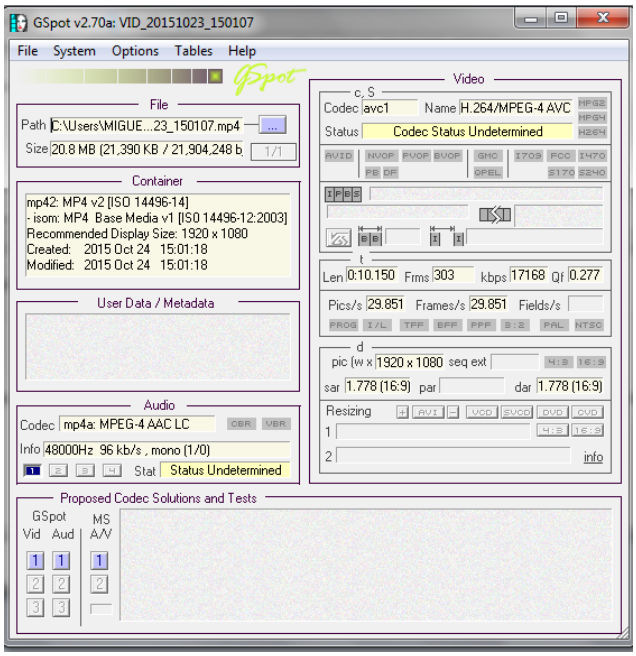


Figura 4.1: Aspecto de la herramienta GSpot

4.3.2. MediaInfo

Esta herramienta cuenta con una interfaz gráfica, estructura en 3 bloques:

En el primer bloque se encuentra la información general del vídeo como: formato, tamaño, duración, identifica las pistas, si son de audio o vídeo y su códec, la tasa de bits, fecha de modificación y creación y en este vídeo concreto extrae información de la versión del SO con el que ha sido capturado, pues este vídeo contiene dicha información. En el segundo bloque se ve la información de

la pista de vídeo: idioma, velocidad, resolución, fotogramas por segundo y el códec utilizado. En el Tercer bloque se ve la información de la pista de audio: idioma, velocidad, número de canales y códec de audio.

Esta herramienta sí posee la opción de realizar análisis de un directorio completo. Al igual que con la herramienta anterior el número de datos extraídos es muy limitado, tampoco incluye la opción de incorporarlos a una base de datos, ni un reproductor propio donde poder reproducir el vídeo a analizar o visualizar su thumbnail. Tampoco se observa la posibilidad de obtener un mapa con la localización de los vídeos. Cabe destacar de esta herramienta su posibilidad de analizar grupos de vídeos y su amplia gama de opciones de visualización de los mimos, como se puede ver en las imágenes.

Su aspecto y los datos extraídos se pueden observar en la Figura 4.2.

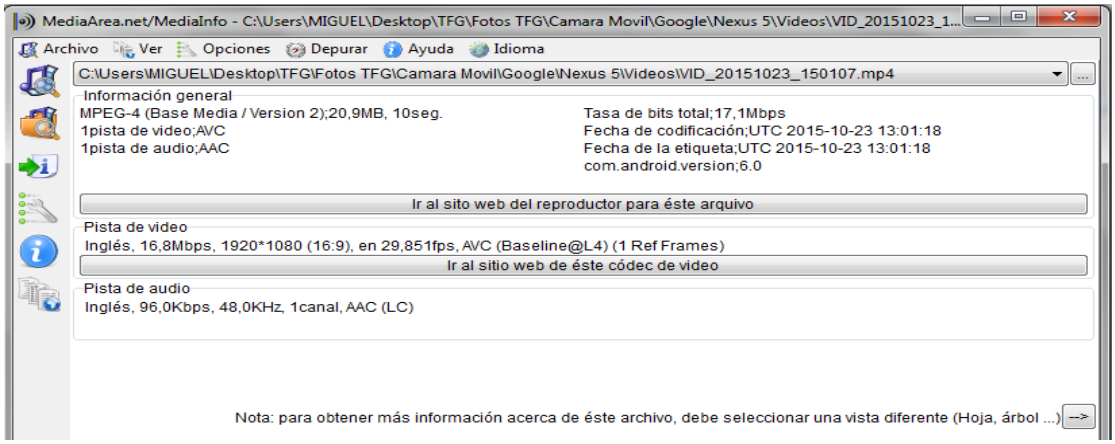


Figura 4.2: Aspecto "Básico" de la herramienta MediaInfo

La herramienta cuenta con diferentes formas de visualización. A pesar de ello la herramienta sigue siendo inadecuada para el análisis forense. En la "Visualización Básica", se encuentra el mismo vídeo analizado con la herramienta GSpot y podemos ver algunas incongruencias como la fecha de modificación y creación.

En GSpot dichas fechas son 24/10/2015 15:01:18 para ambos campos, pero en

MediaInfo encontramos que el valor para estos campos es: 23/10/2015 13:01:18.

Esta falta de coherencia en un dato tan importante para el análisis forense como es la fecha de creación y modificación hace pensar que una o las dos herramientas no siguen el estándar para la obtención de dicho campo y por lo tanto ser inadmisibles para el análisis forense.

El resto de datos si concuerdan, pero como se puede observar una herramienta extrae unos datos y la otra herramienta extrae otros, como el dato de la versión del SO que extrae MediaInfo y no podemos ver en GSpot.

Ahora se procede a analizar una tercera herramienta, donde se aprecia la extracción de una gran cantidad de datos en comparación con las anteriores, pero cuenta sin interfaz gráfica lo que dificulta su usabilidad.

4.3.3. Exiftools

Esta herramienta se ejecuta por consola, por lo que se limita a usuarios que tengan los conocimientos suficientes para saber utilizarla. El resultado de la ejecución es una lista con todos los datos extraídos. Su aspecto y los datos extraídos se pueden observar en la Figura 4.3.

Se observa que se extraen muchos más datos que en las herramientas anteriores aunque sigue sin mostrar todos los datos que podemos obtener de un vídeo.

Extrae por ejemplo el campo de la versión del SO con el que ha sido capturado como hace MediaInfo, pero no hacía GSpot. Extrae todos los datos que se obtenían en las herramientas anteriores aunque se visualizan de una manera menos eficiente y al no contar con interfaz gráfica tampoco cuenta con distintas formas de visualización como MediaInfo, un reproductor de vídeo, una visualización del thumbnail del vídeo, análisis de vídeos de forma grupal ni opción de contar con una base de datos.

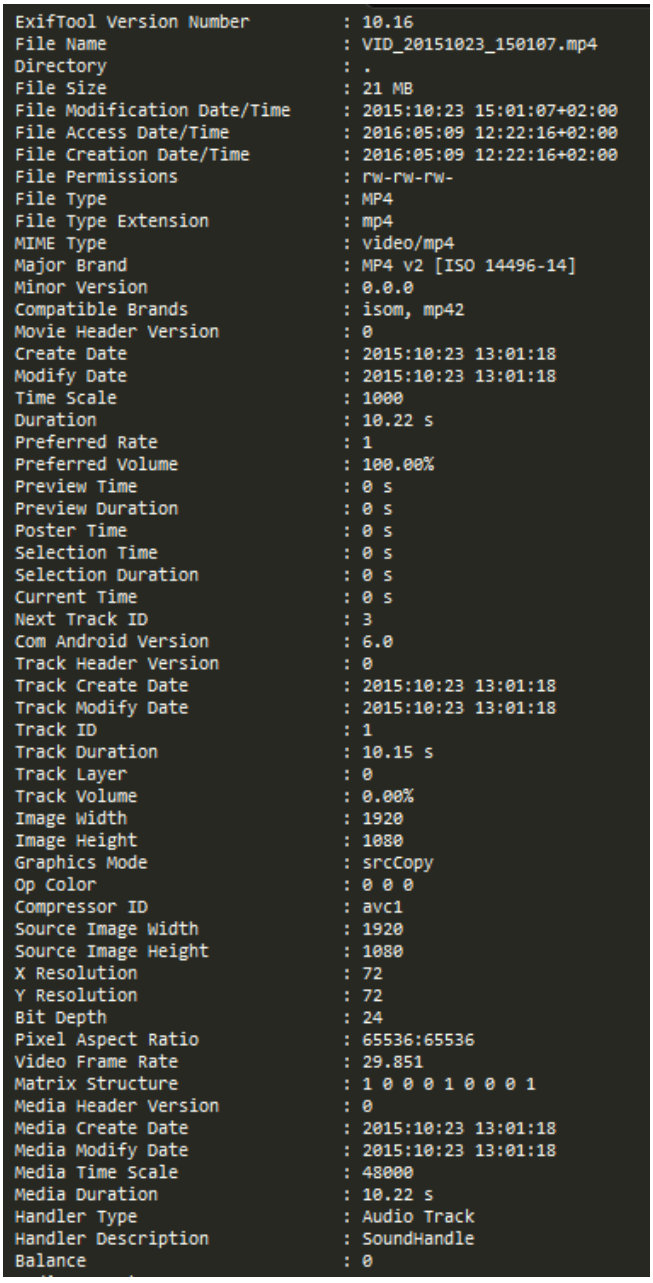


Figura 4.3: Aspecto de los resultados de la herramienta Exiftools

Podemos observar que la fecha indicada es el 23/10/2015 15:01:07+02:00, debido a que la herramienta está diseñada para un sistema horario +02:00, pero en la fecha de creación del track se puede ver la fecha obtenida: 23/10/2015 13:01:18.

La fecha real del vídeo analizado por las tres herramientas es el 23/10/2015 15:01:18, por lo tanto GSpot, obtiene bien la hora pero mal el día, MediaInfo

obtiene bien el día pero mal la hora, posiblemente debido a un problema de zona horaria y Exiftools, que obtiene bien la fecha pero mal la hora, aunque en el campo “File Modification Time”, si le añade esas dos horas para obtener la hora correcta, problema posiblemente causado por la zona horaria de los diseñadores de las herramientas, pero esta información es fundamental para el análisis forense y se debería obtener siempre la misma fecha independientemente de la zona horaria en la que se encuentren, y una vez obtenida la hora según la especificación, añadir la posibilidad de introducir la zona horaria que se desee, indicando siempre ese desfase horario para evitar problemas de interpretación.

Como se observa, en este análisis, las herramientas existentes para la extracción correcta de metadatos en vídeos digitales de dispositivos móviles tienen un alcance limitado pues no muestran todos los metadatos al usuario, ni cuentan con algunas características fundamentales como la visualización del propio vídeo o la posibilidad de guardar los proyectos para futuras consultas. Estas herramientas extraen la información de una serie de átomos concretos dentro de los metadatos de un vídeo. Estas condiciones hacen necesaria la creación de una nueva herramienta adecuada para el análisis forense.

Donde como se muestra en las Figura 4.4 se extraen todos los átomos del vídeo y se estructura por la jerarquía de átomos. Así podemos analizar tantos el contenido de los átomos de un vídeo como su disposición. Todos estos detalles no los podemos encontrar en las anteriores herramientas analizadas, donde se muestra al usuario sólo una parte de estos datos. También posee un reproductor multimedia propio y muestra un thumbnail, de forma que el usuario pueda visualizar en todo momento el vídeo que se está analizando como se muestra en la Figura 4.5, estas características no las podemos encontrar en ninguna de las aplicaciones analizadas y puede ser de gran utilidad para el usuario.

Figura 4.4: Átomos encontrados con la herramienta

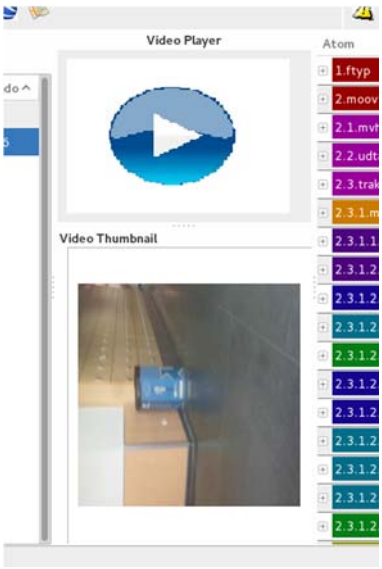


Figura 4.5: Reproductor y Thumbnail

La otra gran diferencia con las herramientas analizadas es la de poder crear proyectos donde almacenar en una base de datos todos los vídeos que se quieran analizar, pudiendo analizarlos de forma individual (Figura 4.6), como grupal (Figura 4.7). Tiene la capacidad de poder analizar los vídeos sin necesidad de tenerlos físicamente, algo muy útil si se trabaja desde varios terminales o se cuenta con un espacio limitado, permitiendo en cualquier

momento exportar estos vídeos de la base de datos a un disco físico.

Figura 4.6: Átomos mostrados de forma individual usando la base de datos

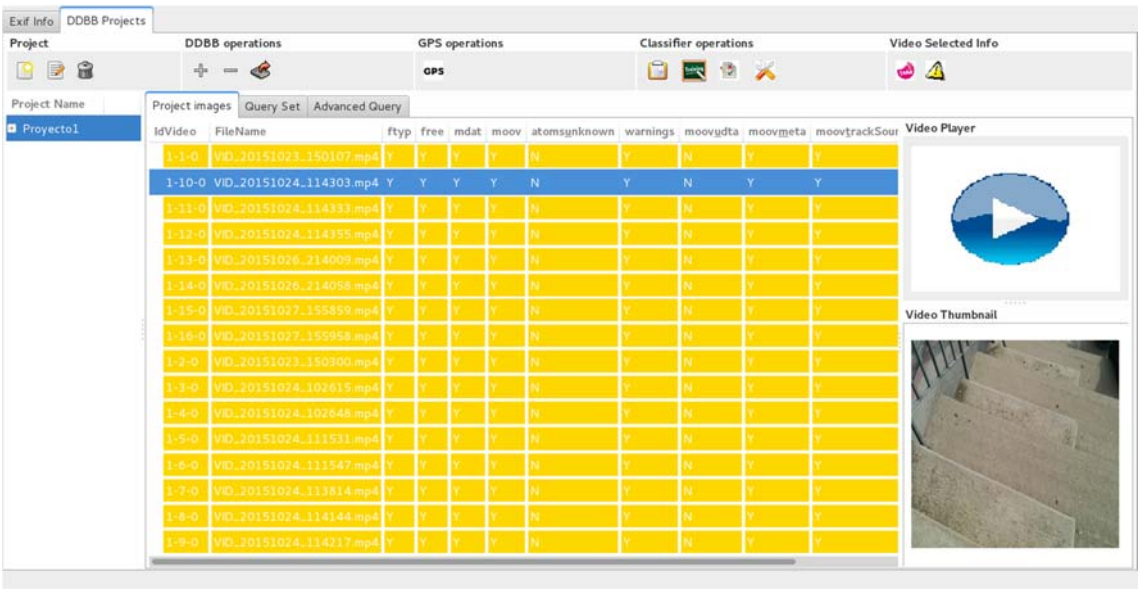


Figura 4.7: Información mostrada de forma grupal usando la base de datos

4.3.4. Conclusiones de la comparativa

Una vez comparada la herramienta una a una con otras con propósitos comunes, se puede concluir que, no habiendo ninguna que ofrezca todas las mejores posibilidades, sin duda la herramienta presentada en este trabajo es la que ofrece una mayor funcionalidad y versatilidad en el tratamiento de metadatos de vídeos. En la Tabla 4.2 se presenta una comparación de todas las herramientas evaluadas.

Ninguna de las aplicaciones comparadas posee un tratamiento de vídeos en grupo, así como una extracción de metadatos de vídeos más completa y organizada. Esta aplicación no tiene como objetivo primordial la visualización de galerías de vídeos, sino favorecer y automatizar, en la medida de lo posible, la tarea del análisis forense de vídeos de dispositivos móviles con respecto a los metadatos. Este objetivo se consigue con mayor éxito que con cualquiera de las herramientas presentadas en la comparación.

	Gspot	**MediaInfo**	**ExifTools**	**Herramienta desarrollada**
Sistema operativo	Windows	Windows	Windows Mac	Linux
Entorno Gráfico	✓	✓	✗	✓
Procesamiento masivo	✗	✗	✗	✓
Reproductor interno	✗	✗	✗	✓
Thumbnail	✗	✗	✗	✓
Mapa GPS	✗	✗	✗	✓
Metadatos	Algunos	Algunos	Muchos	Todos

Tabla 4.2. Tabla comparativa entre aplicaciones existente

5. ANÁLISIS DE UN CONJUNTO DE VÍDEOS MEDIANTE LA HERRAMIENTA

Una vez presentada la herramienta, se ha estimado oportuno realizar un análisis de vídeos reales capturados con dispositivos móviles utilizando las distintas funcionalidades de la herramienta desarrollada.

Este análisis tiene como objetivo profundizar en el conocimiento de la propia especificación y comprobar si ésta es seguida por los fabricantes. Obviamente dado el alto número de átomos que posee la especificación y que cada imagen solo posee un subconjunto de ellos, se han elegido algunas estructuras para el análisis. El análisis ha seguido un orden lógico de estructuras de mayor a menor nivel.

Los vídeos han sido obtenidos de dispositivos móviles de personas conocidas, que además de aportar los archivos, han aportado la información sobre la marca y modelo del dispositivo. Se ha intentado buscar la máxima heterogeneidad posible con respecto a las marcas y los modelos de los dispositivos.

El conjunto de vídeos está formado por 70 vídeos de 5 marcas y 7 modelos. En la Tabla 5.1 se muestran los modelos agrupados por marca.

Marca	Modelo
Google	Nexus 5
Samsung	Galaxy Nexus
	Galaxy S4 Mini
	Galaxy S3 Neo
One Plus One	One Plus One
Sony Ericsson	Xperia M2
Xiomi	Mi3

Tabla 5.1: Teléfonos móviles clasificados por marca y modelo

El experimento se ha realizado con la herramienta desarrollada en un ordenador portátil con sistema operativo Debian 3.16.7, disco duro: 103.7 GB, CPU Intel Core i7 2.20GHz y memoria RAM de 4GB. Los 70 videos fueron sometidos a la aplicación, obteniendo el siguiente rendimiento:

Nº de vídeos: 70
Tamaño total: 1,6 GB
Tiempo de procesamiento: 5'43 min
Tiempo promedio: 297.89 MB/min = 4.965 MB/seg

5.1. Análisis de la Estructuras de los Átomos

Tras analizar los 70 videos se ha podido concluir que un mismo teléfono móvil siempre tiene los mismos átomos de video y que estos mantienen el mismo orden. Se observa que los átomos encontrados concuerdan con lo descrito en la especificación. En la Figura 5.1 se presenta la marca, el modelo del dispositivo y los átomos encontrados en cada uno de los vídeos capturados por ellos.

El primer átomo encontrado es el "fytp" cómo indica la especificación. Posteriormente hay dos opciones: Si el video tiene el átomo "free", el siguiente átomo que se encuentra es el "moov" y todos sus ´átomos hijos, terminando los datos del video con los átomos "free" y "mdat". Pero si el video no contiene el átomo "free", el siguiente átomo que se encuentra es el "mdat", seguido del átomo "moov" y todos sus ´átomos hijos.

Este orden no está establecido en la especificación y podría cambiar para un determinado modelo o fabricante, al igual que los átomos, ya que son opcionales y su orden también podría variar, aunque tras un análisis previo de los videos se observa que generalmente siguen el mismo patrón.

Se puede observar que el átomo "moov" siempre mantiene la misma estructura con los mismos átomos hijos. Se observa también que el primer átomo "trak"

encontrado siempre se relaciona con la pista de video y el segundo con la pista de audio, puesto que dentro del átomo “hdlr” del primer “track”, se encuentra los campos “Component Type” y “Subtype” con valor “vide” y “VideoHandle”, al igual ocurre con el átomo “hdlr” del segundo “track”, donde se encuentra los campos “Component Type” y “Subtype” con valor “soun” y “SoundHandle”.

A continuación se analiza el primer átomo “trak”, track de vídeo, en concreto su átomo hijo “minf”, ya que el resto es igual en todos los videos. El átomo “minf” tiene los átomos “vmhd”, “dinf”, que siempre tiene el átomo hijo “dref” y éste a su vez el átomo “url”, como se indica en la especificación. Por último encontramos el átomo “stbl”, que siempre posee los átomos “stsd”, “stts”, “stss”, “stsz”, “stsc” y por último el átomo “stco”, pero a veces este átomo no está presente, pero en su lugar se encuentra el átomo “co64”, que tras un análisis más profundo de los datos se concluye que el átomo “co64” y el “stco” son el mismo pero que reciben distinto nombre, dependiendo de la marca del dispositivo que grabó el video o del tamaño del video, ya que “co64” se suele utilizar cuando el tamaño del video es muy grande. Se puede concluir que el átomo “co64” se utiliza siempre, independientemente del tamaño del vídeo, en los móviles de marcha china, como “OnePlus” o “Xiomi”, notando que si el átomo es “co64” el tamaño de sus entradas es de 64 bits y si es “stco” el tamaño es de 32 bits. En el átomo “stsd” se encuentra el tipo de códec “avc1” y el átomo de extensión “avcC” y en algunos casos un átomo más, el átomo “pasp”, ya que este es opcional como se indica en la especificación.

En el segundo átomo “trak”, track de audio, en su átomo hijo “minf” encontramos que tiene a su vez los átomos hijos “smhd”, en sustitución del “vmhd”, “dinf”, que es igual al del primer átomo “trak” descrito anteriormente, y el átomo “stbl”, que también es igual al del primer átomo “trak”, con la excepción del átomo “stsd”, que en este caso el tipo de códec es el “mp4a” y el átomo de extensión el “esds” y de que el átomo “stss” no está presente.

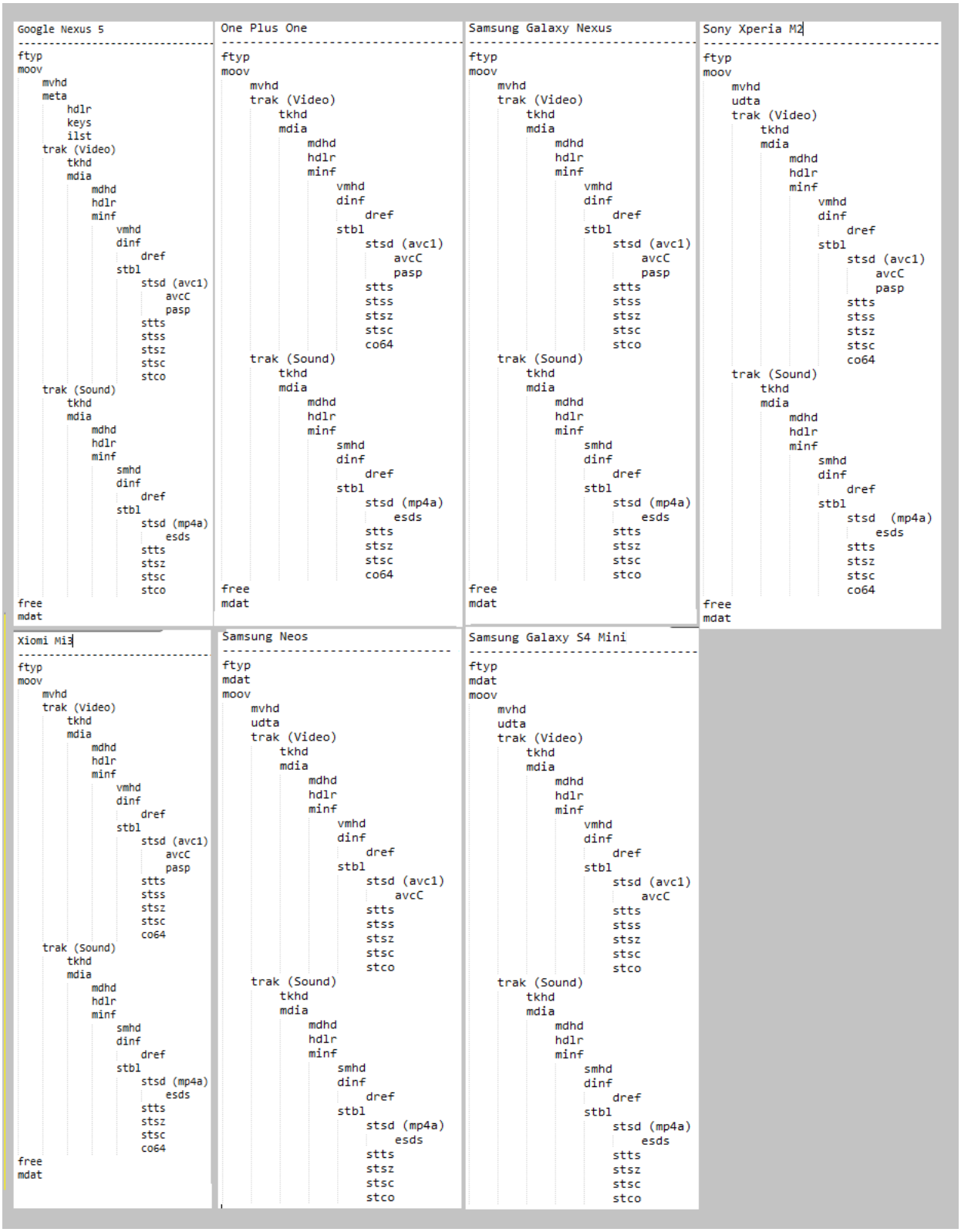


Figura 5.1: Estructuras de los átomo de vídeo por marca y modelo

Finalmente, se observa que la mayoría de los modelos analizados tienen los mismos átomos y su estructura es similar. Sin embargo, los dispositivos móviles Nexus 5 del fabricante Google se encuentra el átomo "meta", junto con sus hijos "hdlr", "keys" e "ilst" que no están presentes en los demás modelos.

5.2. Análisis de la Información Almacenada en los Átomos

En este apartado se analiza la información almacenada en los átomos. En la Tabla 5.2 se puede ver que casi todos los fabricantes siguen la misma especificación con algunas excepciones, como el formato preferido o la lista de formatos compatibles, si contienen datos de usuario o metadatos o el átomo de extensión de códec "pasp", entre otros.

En la Tabla 5.2 se puede encontrar los 7 modelos analizados de las 5 marcas de dispositivos móviles. Tras el estudio se ha determinado que cada modelo siempre sigue una misma estructura, a excepción de que se modifique alguna opción del dispositivo como por ejemplo la ubicación GPS en cuyo caso aparecerá el átomo "udta" con la entrada "@xyz". Como se puede ver en la imagen. También se puede observar lo descrito anteriormente referente a la posición de los átomos y de cómo dependiendo de los átomos que encontramos la posición es una u otra.

Marca				Google	One Plus One	Samsung			Sony	Xiomi
Modelo				Nexus 5	One Plus One	Galaxy Nexus	Galaxy S4 Mini	Galaxy S3 Neo	Xperia M2	Mi3
Formato	Preferido			mp42	mp42	isom	isom	isom	mp42	isom
	Compatibles			['isom', 'mp42']	['isom', 'mp42']	['isom', '3gp4']	['isom', '3gp4']	['isom', '3gp4']	['isom', 'mp42']	['isom', '3gp4']
Película	Escala de tiempo			1000	1000	1000	1000	1000	1000	1000
	Volumen			1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Velocidad			1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Datos de Usuario			-	-	-	SDLN:SEQ_PLAY smrd:TRUEBLUE smta:saut	SDLN:SEQ_PLAY smrd:TRUEBLUE smta:saut	-	-
	Metadatos	Campo		com.android.version	-	-	-	-	-	-
		Valor		6.0	-	-	-	-	-	-
Pista de video	Escala de tiempo			90000	90000	90000	90000	90000	90000	90000
	Color			[0,0,0]	[0,0,0]	[0,0,0]	[0,0,0]	[0,0,0]	[0,0,0]	[0,0,0]
	Modo gráfico			0	0	0	0	0	0	0
	Manipulador			VideoHandle	VideoHandle	VideoHandle	VideoHandle	VideoHandle	VideoHandle	VideoHandle
	Subtipo del manipulador			vide	vide	vide	vide	vide	vide	vide
	Ancho			1920	1920	1920	1920	1920	1920	1280
	Alto			1080	1080	1080	1080	1080	1080	720
	ID			1	1	1	1	1	1	1
	Volumen			0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Códec	Tipo		avc1	avc1	avc1	avc1	avc1	avc1	avc1
		Resolución	Horizontal	72.0	72.0	72.0	72.0	72.0	72.0	72.0
			Vertical	72.0	72.0	72.0	72.0	72.0	72.0	72.0
		Profundidad		24	24	24	24	24	24	24
		Tabla de color		65535	65535	65535	65535	65535	65535	65535
		Ancho		1920	1920	1920	1920	1920	1920	1280
		Alto		1080	1080	1080	1080	1080	1080	720
		Extensiones		[avcC,pasp]	[avcC,pasp]	[avcC,pasp]	[avcC]	[avcC]	[avcC,pasp]	[avcC,pasp]
		Espaciado de píxeles	Horizontal	65536	65536	65536	-	-	65536	65536
			Vertical	65536	65536	65536	-	-	65536	65536
Pista de audio	Escala de tiempo			48000	48000	48000	48000	48000	48000	48000
	Manipulador			SoundHandle	SoundHandle	SoundHandle	SoundHandle	SoundHandle	SoundHandle	SoundHandle
	Subtipo del manipulador			soun	soun	soun	soun	soun	soun	soun
	Ancho			0	0	0	0	0	0	0
	Alto			0	0	0	0	0	0	0
	ID			2	2	2	2	2	2	2
	Volumen			1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Códec	Tipo		mp4a	mp4a	mp4a	mp4a	mp4a	mp4a	mp4a
		Extensiones		[esds]	[esds]	[esds]	[esds]	[esds]	[esds]	[esds]

Tabla 5.2: Análisis de la información de vídeos almacenados

6. CONCLUSIONES Y TRABAJO FUTURO

6.1. Conclusiones

Las conclusiones que se obtener de este trabajo son las siguientes:

Los vídeos se estructuran en átomos, éstos a su vez contienen una cabecera donde se indica el tamaño y el tipo del átomo. El conjunto de todos estos átomos conforman el vídeo y sus metadatos., no hay ninguna norma respecto a los átomos que deben aparecer y el orden de los mismos. Sin embargo, la mayoría sigue una estructura similar. Este es el caso de los vídeos con formato MP4 y compresión de vídeo H.264, que siguen la especificación redactada en el punto 4 de este trabajo.

Al igual que ocurre con las imágenes hay dos campos de interés para el análisis forense de vídeo, la identificación de la fuente del vídeo y la detección de manipulaciones sobre el vídeo. En este último campo se pueden dar dos vertientes: manipulación en los metadatos del vídeo y manipulación en los fotogramas del vídeo, que éste a su vez puede ser resultado de una modificación en el contenido de los fotogramas o una modificación en la estructura de los fotogramas. Es decir, añadiendo o eliminando fotogramas.

El análisis de los metadatos de un vídeo es una tarea demasiado tediosa y lenta si se realiza manualmente por ellos es necesario crear una herramienta de análisis automático de estos datos.

Muchas de las herramientas existentes que obtienen y analizan los metadatos de los vídeos no consiguen ser lo suficientemente completas para utilizarlas en el análisis forense, por ello es preciso crea una herramienta como la desarrollada en este proyecto, que aunque todavía le quede mucho por hacer, pero ya se ha marcado un comienzo.

6.2. Trabajo Futuro

Como se ha podido observar a lo largo de todo este trabajo, el campo del análisis forense de vídeo es un tema que está en sus inicios y hay mucho que investigar y desarrollar:

- Crear métodos que identifiquen de forma certera si un vídeo ha sido manipulado o no, tanto en sus metadatos como en el contenido de sus fotogramas y desarrollar herramientas que integren estos métodos y permitan al usuario gestionar y visualizar los resultados.

- La herramienta desarrollada en este proyecto, está diseñadas para el análisis automático de vídeos pero todavía no cumple con todo lo necesario para ser un herramienta útil para el análisis forense, ya que no incluye los métodos de detección de manipulaciones, aunque si incluye ya métodos de identificación de la fuente. Por lo tanto como trabajo futuro queda seguir completando las funcionalidades de la herramienta hasta que ésta pueda ser usada para el análisis forense de vídeos.

- Investigar cuáles son los átomos útiles para el análisis forense y cuáles no, para poder obviar estos últimos y buscar la manera más óptima para mostrárselos al usuario, de forma que pueda identificar rápida y cómodamente la información que esté buscando.

- Ampliar el análisis de metadatos realizado en este trabajo a otros formatos de vídeo: MOV, AVI, MKV, entre otros. Así como ir manteniendo al día los átomos existentes, su estructura y sus versiones, ya que estos irán cambiando con el paso de los años y será imprescindible ir actualizando la herramienta. Y así tener una herramienta bastante completa y fiable desarrollada para el análisis forense de vídeos.

RESUMEN EN INGLÉS

7. INTRODUCTION

7.1. Motivation

Over the last decade there has been a rapid increase in the demand for mobile devices. This rise is mostly due to the development of technology necessary to cut costs and make them more accessible to the general population. Currently, there is fierce competition among manufacturers to integrate a high definition camcorder available to the user at all times.

Most types of mobile devices have an integrated digital camera. In fact, 97% of mobile phones have an integrated digital camera therefor can be carried by their owners most of the time everywhere [1]. There are predictions that digital still cameras (DSC) will disappear in favor of the new digital cameras integrated into mobile devices [2].

According to a study IC Insights Inc. [3] about the market diversity of systems with cameras, DSCs sales will go down from 47% obtained in 2012 to 27% in 2016. In addition, pursache of integrates digital cameras in mobile phones, computers and tablets will increase from 31% in 2012 to 42% in 2016.

As a result of the foregoing and given the amount of time a person spends with mobile devices, they have become for many people the first device photo capture and video recording device. Daily can be seen images generates by mobile devices in television news, several applications, email or social networks. For example, in the most used social networks (Facebook, YouTube, Flikr, Twitter, etc.), a significant portion of its content is captured with digital cameras of mobile devices [4]. Consequently, images and digital videos generates with mobile devices are also used as silent witnesses in judicial proceedings (child pornography, street violence, social networks ...), being crucial pieces of evidence of a crime [5, 6]. All this means that in certain cases

there are legal restrictions or limitations both for use in different places (schools, universities, government offices, businesses, etc.) as well as for content captured, such as: catch in which minors appear without the consent of their legal guardian or intimate situations of people who violate their right to privacy, etc.

For all these reasons the forensic analysis of digital images and videos from mobile devices has special force today. The study should be concreted for such devices because they have specific features that allow better results, no being valid the forensic techniques for digital images generated by other types of devices. In [7] is described clearly and reasoned the need for specific forensic analysis techniques for mobile devices

7.2. Objectives

This Final Degree Work (TFG) has the following objectives:

- Conduct a study about the structure of digital videos, the atoms of which are composed and metadata extraction techniques of digital videos in order to analyze and understand the most relevant techniques.
- Present the main forensic analysis techniques that determine the geolocation of a digital image.
- Analyze detailed specification information storage MP4 digital video (Media Player 4) H.264 compression format.
- Design and implement in Python programming language an algorithm that allows automatically extract metadata videos stored on mobile devices.

7.3. Work Schedule

The project has been developed in 3 phases: Definition, Execution and Documentation Project. Activities in these phases are presented in Table 8.1.

Task name	Duration (days)	Start	End
• **Project definition**	**40**	**12/10/15**	**04/12/15**
- Weekly monitoring meetings with tutors	40	12/10/15	04/12/15
- Study of techniques of forensic analysis on mobile devices	15	12/10/15	30/10/15
- Study of forensic analysis techniques applied to mobile devices video	15	02/11/15	20/11/15
- Project definition	10	23/11/15	04/12/15
• **Project execution**	**126**	**11/12/15**	**03/06/16**
- Requirements specification	**45**	**14/12/15**	**12/02/16**
- Design	**40**	**15/02/16**	**08/04/16**
- Implementation	**35**	**11/04/16**	**27/05/16**
- Testing	**20**	**09/05/16**	**03/06/16**
- Control	**126**	**11/12/15**	**03/06/16**
• **Documentation**	**167**	**19/10/15**	**07/06/16**
- Generation project documentation	165	19/10/15	03/06/16
- Memory preparation	27	02/05/16	07/06/16

Table 8.1: Work Schedule

1. **Definition of the project:** At this stage several meetings with tutors were conducted to define and delimit the final degree project and establish guidelines and set work schedules mentoring and monitoring.

2. **Project execution:** This phase has the aim to develop the project defined in the previous phase. The following general activities were carried out: requirements specification, design, implementation and testing. In addition, monitoring and control of the entire project progress were made, in order to expedite the necessary adjustments in each of the activities. Table 8.2

presents the activities in this phase as well as the duration thereof.

3. **Documentation Phase:** This phase is performed in parallel to the above stages. Its main objective was to gather all project documentation.

Task name	Duration (days)	Start	End
Requirements specification	**45**	**14/12/15**	**12/02/16**
• Study of techniques for extracting image metadata from mobile devices	10	14/12/15	25/12/15
• Study of multimedia container for digital videos	8	04/01/16	13/01/16
• Detailed analysis on the specification metadata storage in MP4 video format	12	14/01/16	29/01/16
Design	**40**	**15/02/16**	**08/04/16**
• Design of the database that will store the metadata information	5	15/02/16	19/02/16
• Design of the atoms extraction algorithm for multimedia MP4 container	18	22/02/16	16/03/16
• Design of the individual treatment module for digital videos	7	17/03/16	25/03/16
• Design of metadata massive management module for digital videos	10	28/03/16	08/04/16
Implementation	**35**	**11/04/16**	**27/05/16**
• Atoms extraction algorithm for multimedia container	20	11/04/16	06/05/16
• Individual treatment module for digital videos	5	09/05/16	13/05/16
• Metadata massive management module for digital videos	10	16/05/16	27/05/16
Testing	**20**	**09/05/16**	**03/06/16**
• Atoms extraction algorithm for multimedia container	7	09/05/16	17/05/16
• Individual treatment module for digital videos	5	19/05/16	25/05/16
• Metadata massive management module for digital videos	5	30/05/16	03/06/16
Control	**126**	**11/12/15**	**03/06/16**

Tabla 8.2: Activities of execution phase

8. CONCLUSIONS AND FUTURE WORK

8.1. Conclusions

The conclusions obtained from this work are as follows:

The videos are divided into atoms, they in turn contain a header where the size and type of atom are indicated. The set of all these atoms make up the video and its metadata. There are no rules regarding the atoms must appear and their order. However, most follow a similar structure. This is the case of videos with MP4 format and H.264 video compression, which follow the specification written in chapter 4 of this work.

As is the case with the images there are two areas of interest for forensic video analysis, identification of the source video and tamper detection of the video. In the latter field are two possible areas: handling video metadata and handling in the frames of the video, that this in turn can result from a change in the content of frames or a change in the structure of the frames. That is, by adding or removing frames.

The analysis of a video metadata is a task too tedious and time consuming if done manually therefor is necessary to create a tool for automatic analysis of these data.

Many existing tools to obtain and analyze metadata videos do not get to be complete enough to use in forensic analysis, so it is necessary to create a tool like the one developed in this project, although remains much to be done, but there has already been marked a beginning.

8.2. Future work

As has been observed throughout all this work, the field of forensic video analysis is a subject still too virgin, where there is much to research and develop:

- Create methods to accurate identify if a video has been manipulated or not, in both metadata and the content of their frames and develop tools that integrate these methods to allow the user to manage and visualize the results.

- The tool developed in this project is designed for automatic video analysis but still does not meet all the necessary to be a useful tool for forensic analysis tool because it does not include the methods of detection of tampering, although it includes identification methods of the source. Therefore as future work is to complete the development capabilities of the tool until it can be used for forensic video analysis.

- Investigate atoms which are useful for forensic analysis and which not, to ignore the latter and find the most optimal way to showing them to the people, so that you can quickly and easily identify the information you are seeking.

- Expanding the metadata analysis made in this work to other video formats: MOV, AVI, MKV, among others. AS well as maintaining the existing atoms, their structure and their versions, because these will change over the years and will be the tool must be updated. And this way have a quite complete and reliable tool developed for forensic video analysis.

ANEXOS

A. ESPECIFICACIÓN DE LA HERRAMIENTA

En la Figura A.1, se puede ver el aspecto básico de la aplicación para el tratamiento de vídeos de manera individual y en la Figura A.2, el aspecto básico de la aplicación para tratamiento de vídeos usando una base de datos.

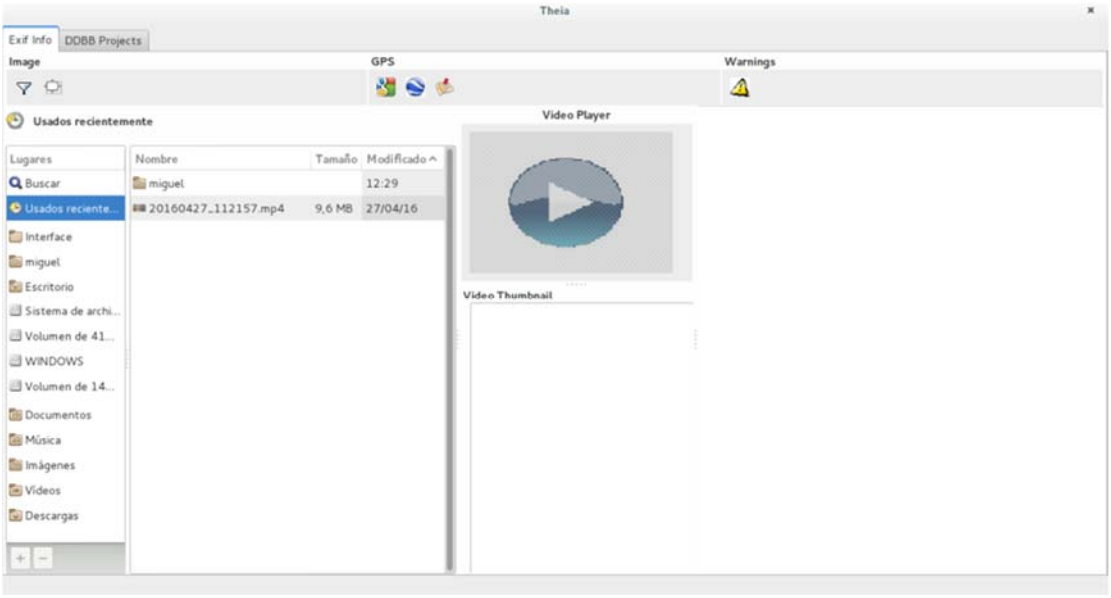


Figura A.1: Aspecto básico de la aplicación 1

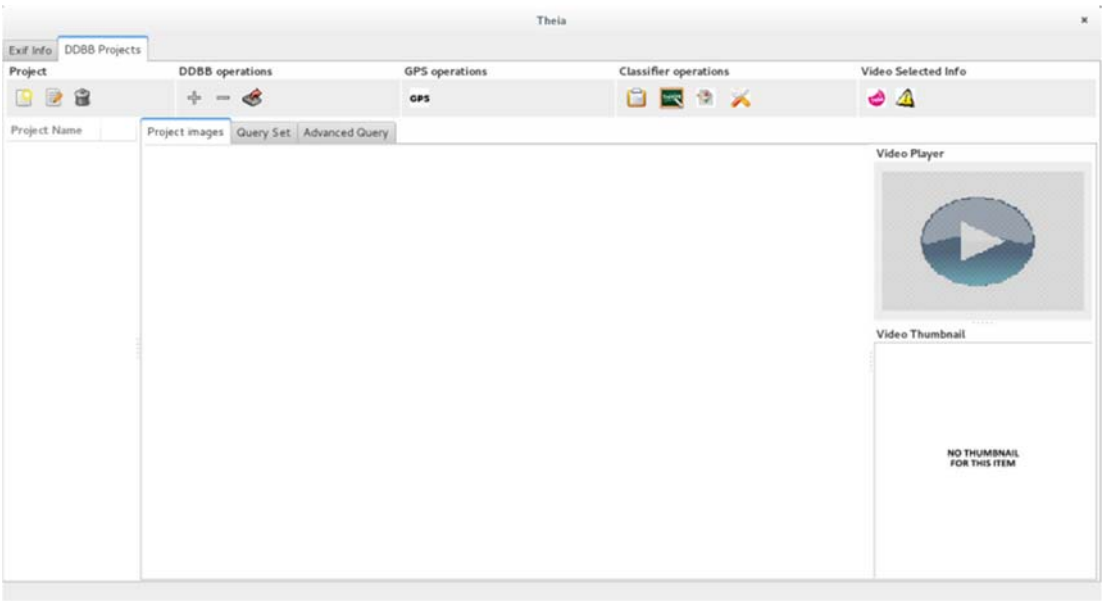


Figura A.2: Aspecto básico de la aplicación 2

Primeramente se mostrará las funcionalidades para el tratamiento individual de vídeo.

A.1. Tratamiento Individual de Vídeos

Como se muestra en la Figura A.1, en la parte derecha se encuentra un explorador donde el usuario podrá buscar el vídeo que desea analizar. A esta búsqueda se le puede aplicar un filtro para acotar la búsqueda al formato de archivo deseado, como se muestra en la Figura A.3.

Una vez seleccionado el archivo a analizar, si es un vídeo, se habilitará el reproductor (Vídeo Player, representado por el botón play azul), se mostrará al usuario el thumbnail del vídeo, que siempre será el primer fotograma a reproducir y finalmente en la parte derecha se mostrará una lista con todos los átomos del vídeo encontrados. Estos átomos tienen una numeración y un color que representan su jerarquía, cada color representa un nivel, de tal forma que los átomos que tiene el mismo color quiere decir que están al mismo nivel de profundidad, así como la numeración, ya que es importante saber quién es el padre del átomo o qué hijos tiene. Todo esto se puede ver en la Figura A.4.

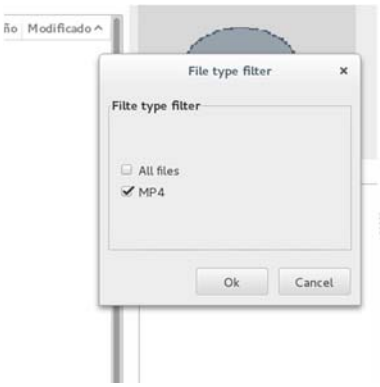


Figura A.3: Filtro de extensión de archivos a buscar.

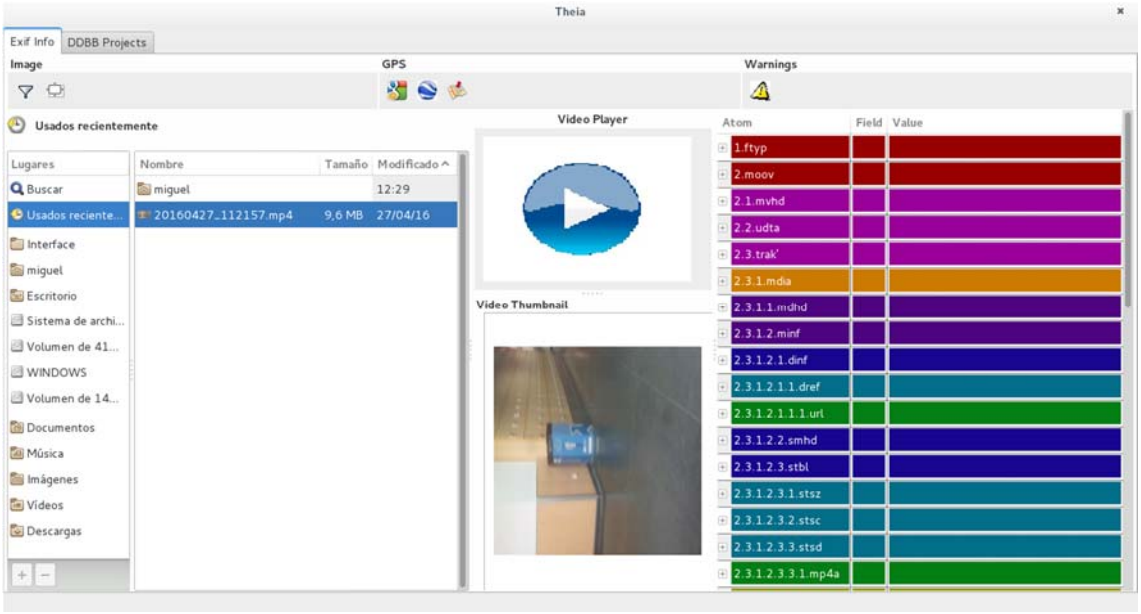


Figura A.4: Selección de vídeo

Una vez llegados a este punto ya se puede ir abriendo los diferentes átomos para ver su contenido. En la segunda columna se puede encontrar el nombre del campo que se está mostrando y en la tercera su valor, como se muestra en la Figura A.5.

Figura A.5: Información de los átomos.

También podemos ver si durante el análisis algún átomo del vídeo se ha analizado de forma irregular, con el botón de arriba a la derecha, donde pone "Warnings", donde se obtiene la información del path del átomo y la irregularidad sufrida, como se muestra en la Figura A.6.

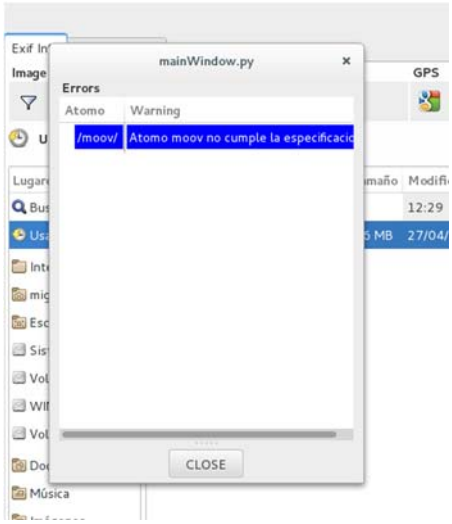


Figura A.6: Irregularidad de los átomos de un vídeo.

También podemos ver su localización en Google Maps, con el primer botón, de la parte central de arriba, de la sección GPS. Esta localización sólo es posible si el vídeo que se está analizando dispone de los datos GPS. En la Figura A.7 se puede ver qué ocurre al pulsar encima de este botón. Abre el navegador predeterminado del usuario y muestra en la página www.google.com/maps la localización del vídeo.

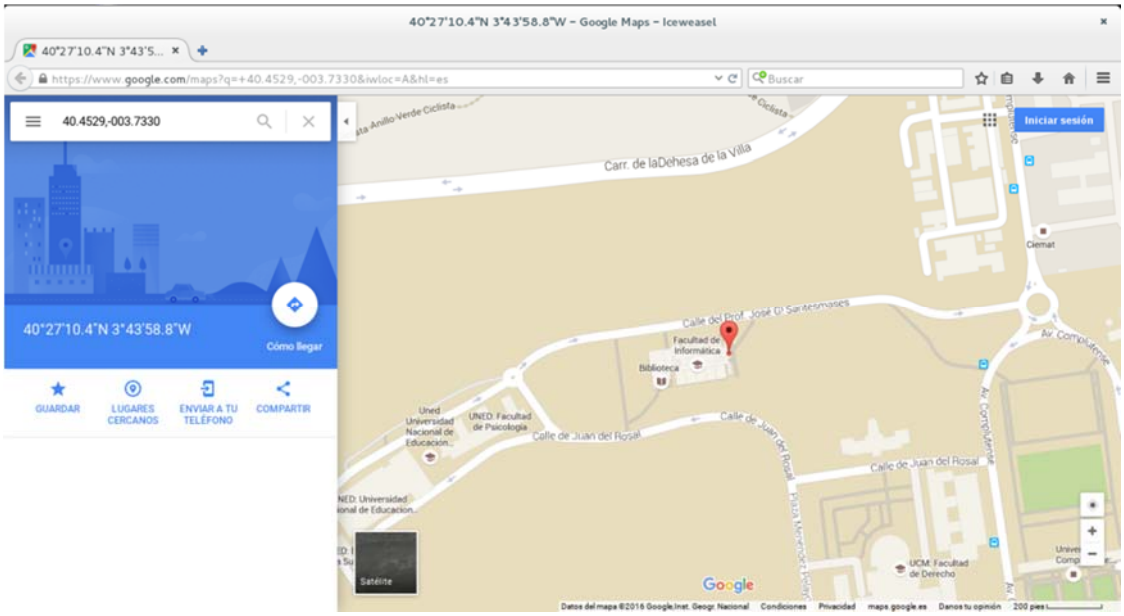


Figura A.7: Localización GPS.

La aplicación también permite guardar dicha localización en un documento KML (Keyhole Markup Languaje), para poder abrirlo desde la herramienta Google Earth [38], tal y como se muestra en la Figura A.8, donde pide que introduzcas un nombre para el archivo y selecciones el directorio donde se desea guardarlo.

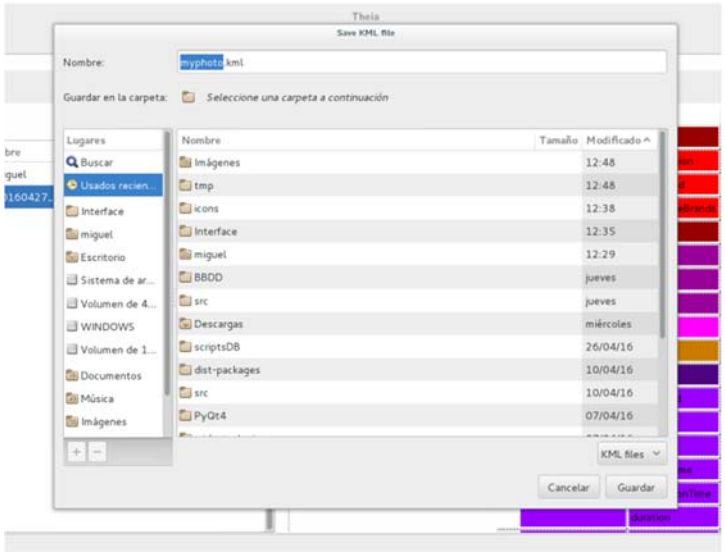


Figura A.8: Guardar localización GPS para Google Earth.

Por último, se ha desarrollado también un reproductor multimedia propio para la herramienta, como se muestra en la Figura A.9, con el que poder reproducir el vídeo que se está analizando, por si fuera necesario consultar su contenido sin necesidad de ir al directorio que lo contiene, ahorrando así al usuario un tiempo que podría aprovechar en analizar más en profundidad los metadatos obtenidos del vídeo.

Este reproductor cuenta con botón de pausa, play, stop, con un controlador de volumen y de tiempo, de forma que se puede pasar a un momento concreto del vídeo.

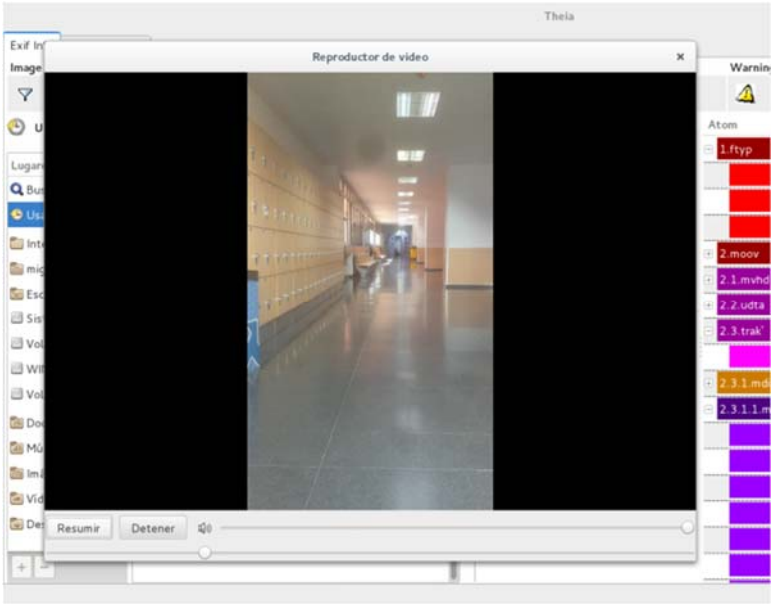


Figura A.9: Reproductor multimedia integrado en la aplicación.

A.2. Tratamiento Grupal de Vídeos

Ahora se procederá a explicar la segunda parte de la aplicación, donde usa una base de datos para gestionar los vídeos. Con esto se quiere conseguir poder analizar los vídeos sin necesidad de tenerlos físicamente en nuestro ordenador, y así conseguir un ahorro en espacio bastante importante. También se busca conseguir con este desarrollo el poder analizar los vídeos de forma grupal gracias a la funcionalidad de realizar consultas de las bases datos.

Una vez accedido a ésta parte de la aplicación, si ya hay algún proyecto guardado en la base de datos, se cargará y aparecerá automáticamente en la aplicación. Pero para hacernos una idea del comportamiento original, se va a proceder sin ningún proyecto guardado previamente.

Lo primero que se puede hacer es crear un nuevo proyecto, donde se pide un nombre para el proyecto y que se seleccione un directorio a analizar, como que se muestra en la Figura A.10. Sólo se incluirán dentro del proyecto aquellos archivos que correspondan con el formato MP4, para ayudarte a visualizarlo mejor, al igual que ocurre en la sección anterior se puede aplicar un filtro para

que sólo aparezcan archivos con extensión MP4, como se muestra en la Figura A.11.

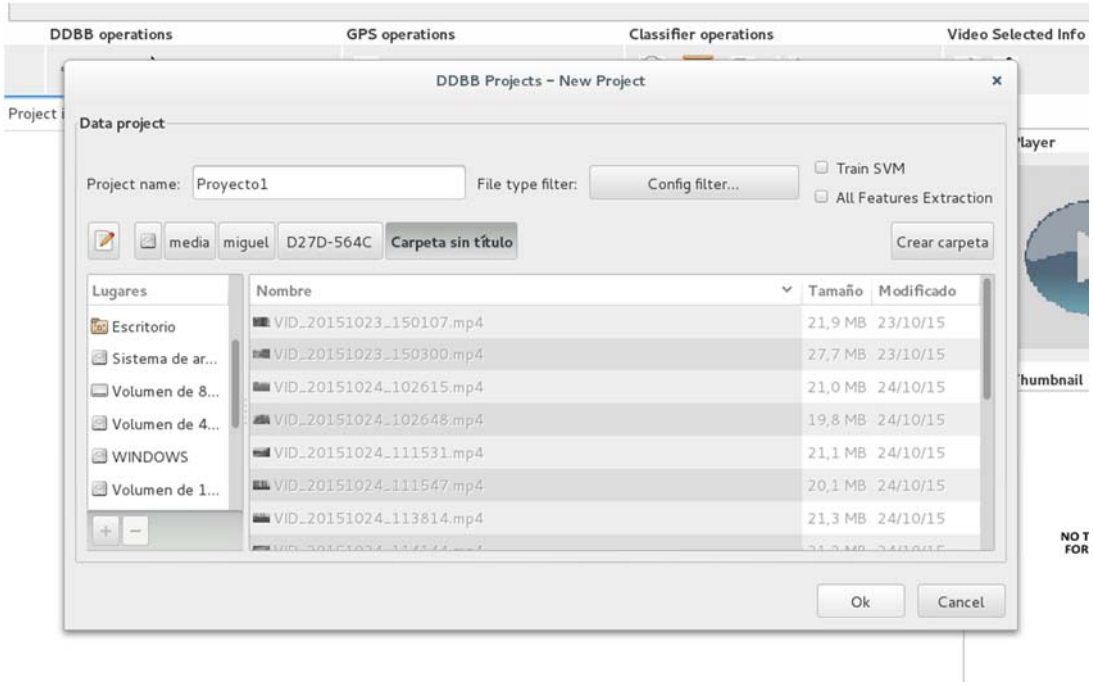


Figura A.10: Crear un nuevo proyecto

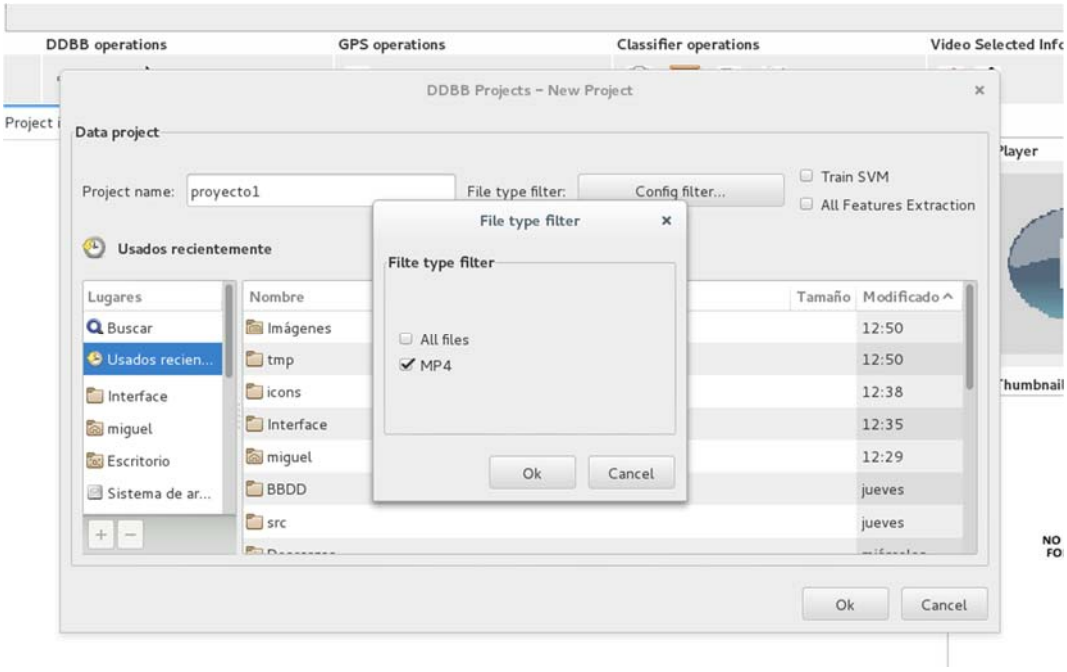


Figura A.11: Filtro para la extensión de los archivos

Si se olvida de dar un nombre al proyecto, al hacer click en el botón OK, se le recordará como en la Figura A.12.

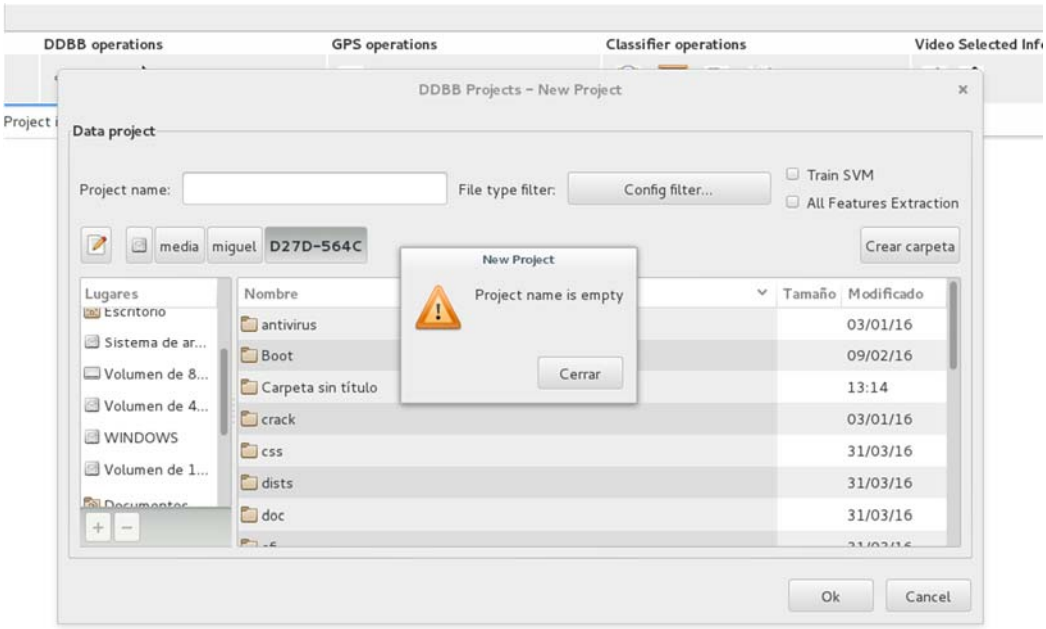


Figura A.12: Fallo al crear un proyecto, "nombre del proyecto vacío".

Una vez dado click en Ok, se empezarán a cargar los vídeos del directorio seleccionado en la base de datos. Para hacerse una idea de la velocidad del programa en gestionar los vídeos e introducirlos en la base de datos, se ha procedido a realizar una prueba con los siguientes resultados:

Número de vídeos insertados: 16 vídeos
Tamaño total: 497,676 MB
Tamaño medio por vídeo: 31,10475 MB
Tiempo total: 1,26 minutos
Tiempo medio por MB: 394.98 MB/min => 6.583 MB/seg

La configuración y características de la máquina donde se ha ejecutado la prueba se pueden encontrar en el capítulo 8, en el punto 8.1, configuración de análisis.

Si toda la inserción de los vídeos se ha realizado correctamente se obtiene el aspecto de la Figura A.13, donde se puede consultar en la parte izquierda, el nombre del/los proyecto/s, en la parte central todas los vídeos pertenecientes al proyecto y alguna información general sobre cada uno, y en la parte derecha

se encuentra el acceso al reproductor multimedia y al thumbnail del vídeo, que se corresponde con el primer fotograma del mismo, pero estas últimas opciones sólo son accesibles cuando se pulsa encima de algún vídeo, como en la Figura A.14.

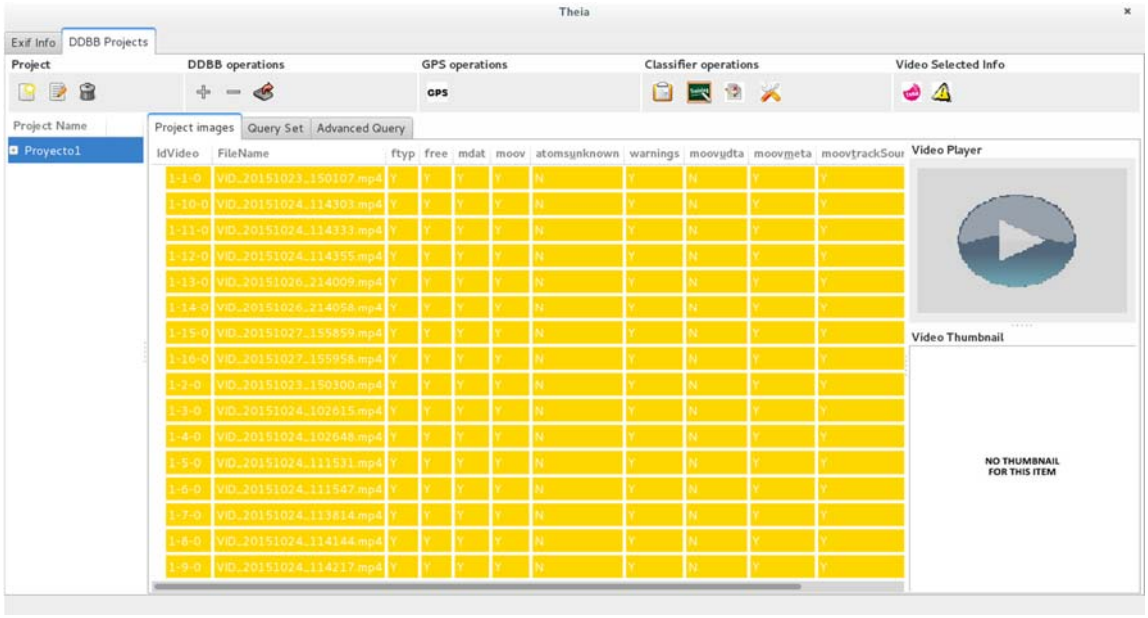


Figura A.13: Resultado de crear un proyecto

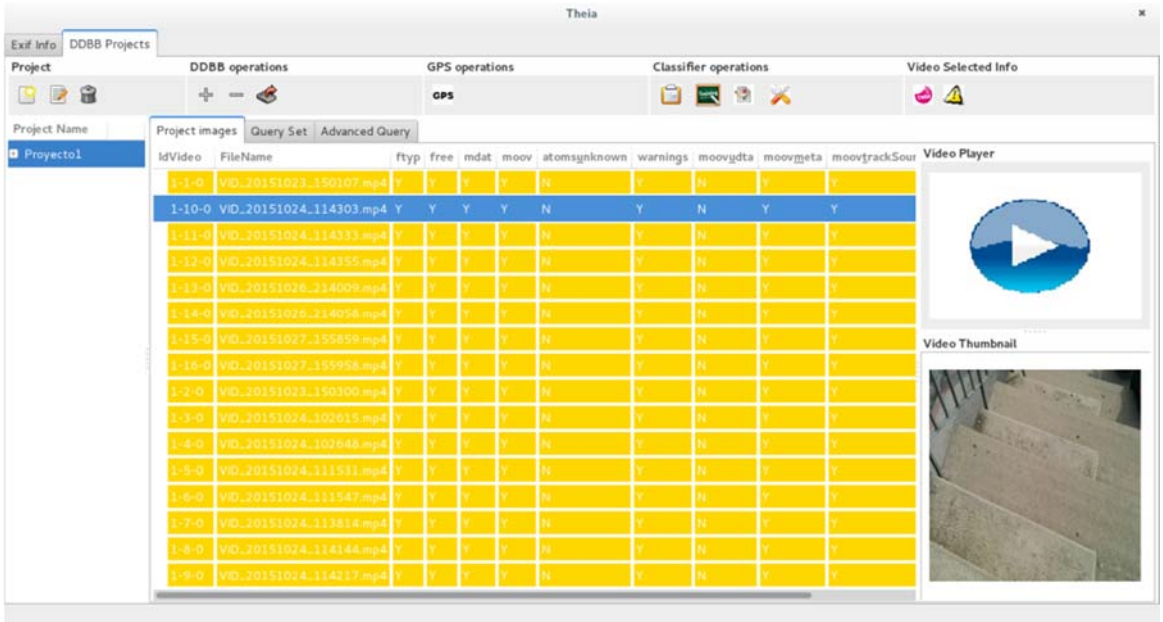


Figura A.14: Resultado de seleccionar un vídeo del proyecto

Una vez seleccionado un archivo de vídeo podemos reproducirlo con el reproductor integrado en la aplicación. Es el mismo reproductor que se usa en la primera parte de este capítulo. Se puede ver un ejemplo en la Figura A.15.

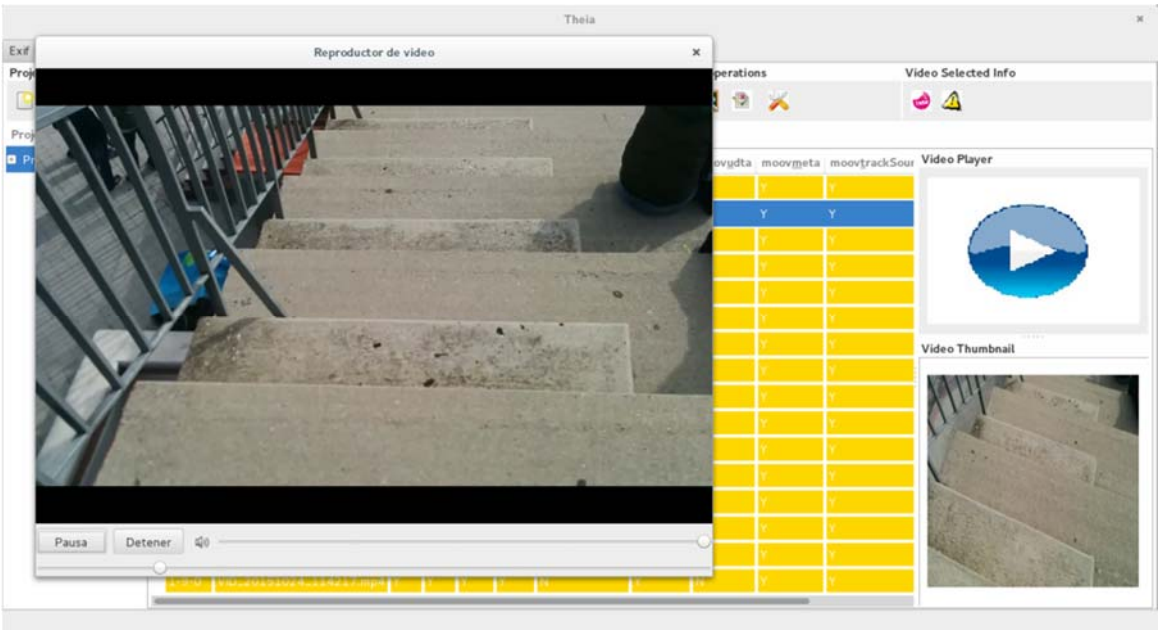


Figura A.15: Reproductor multimedia integrado en la herramienta

También podemos consultar los átomos del vídeo seleccionado, a través del botón de arriba a la derecha que dice "tags", en la sección de "Vídeo Selected Info", esta parte sigue la misma estructura que en la primera parte de este capítulo, muestra la jerarquía de los átomos utilizando colores y numeración, como en la Figura A.16, estos al igual que antes se pueden desplegar para ver los campos y los valores que tiene, como en la Figura A.17:

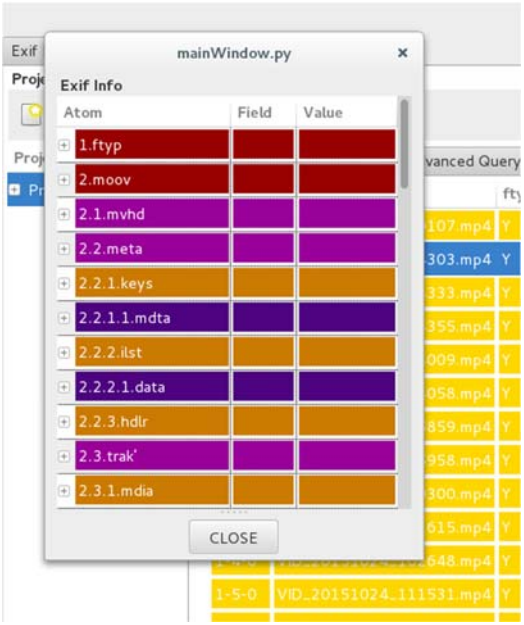


Figura A.16: Consulta de los átomos de un vídeo

Figura A.17: Campos y valores de los átomo

También al igual que en la primera parte, se encuentra el botón "warnings", para consultar las irregularidades encontradas al analizar el vídeo. Este botón se encuentra en la sección "Vídeo Selected Info", con el mismo icono que antes, justo al lado del botón para consultar los átomos del vídeo. Su estructura es la misma que en la primera parte del capítulo, como se observa en la Figura A.18.

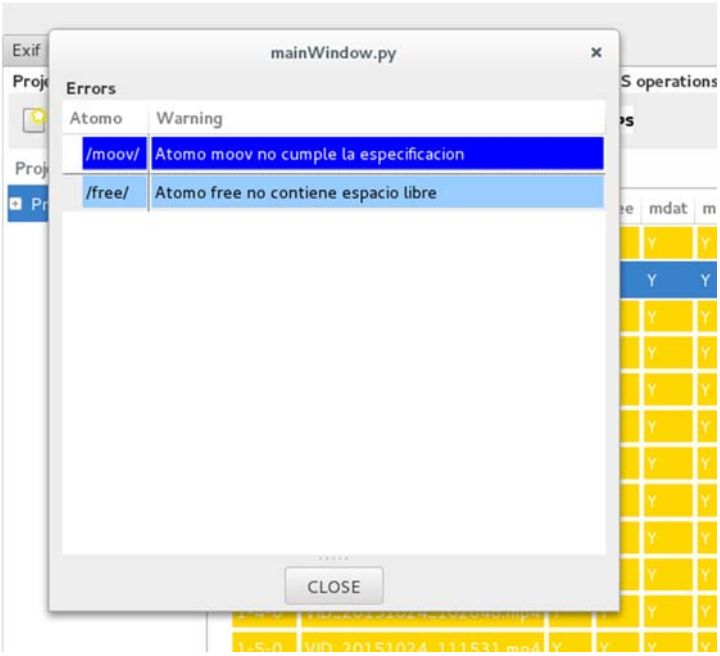


Figura A.18: Irregularidades de los átomos de un vídeo.

En resumen, una vez creado el proyecto y seleccionado un vídeo del proyecto creado, podemos consultar su thumbnail, reproducirlo, consultar sus átomos y sus correspondientes campos así como sus valores y las irregularidades encontradas.

Una vez se seleccione un proyecto se puede editar el nombre del mismo como en la Figura A.19, lo que generará un cambio en el orden en el que aparecen los proyectos en la aplicación pero no generará ningún cambio en su contenido (los vídeos) o eliminar dicho proyecto como se muestra en la Figura A.20, en esta parte, por motivos de seguridad se pide al usuario una segunda confirmación de la orden como se observa en la Figura A.21.

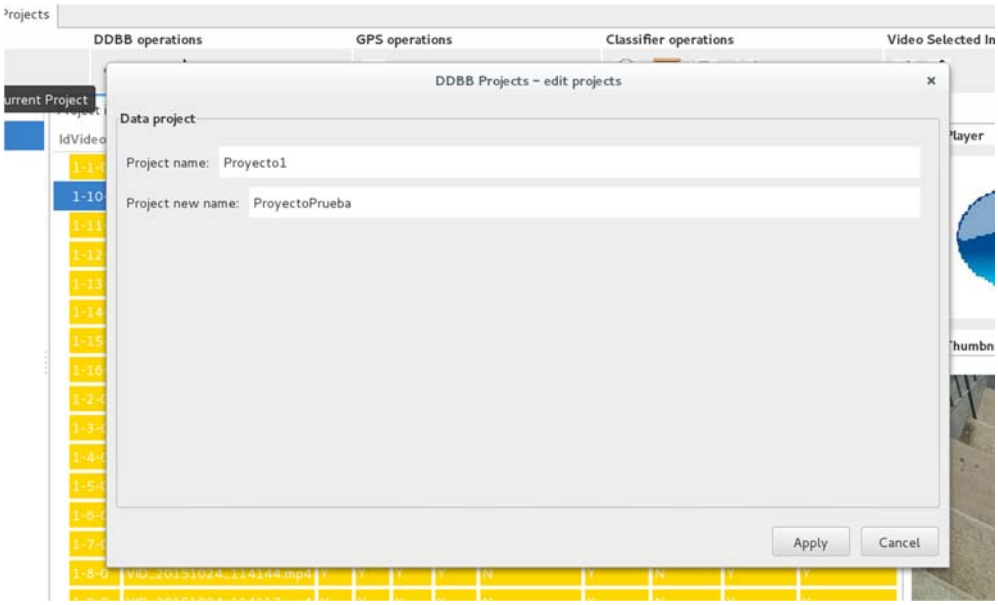


Figura A.19: Editar el nombre de un proyecto

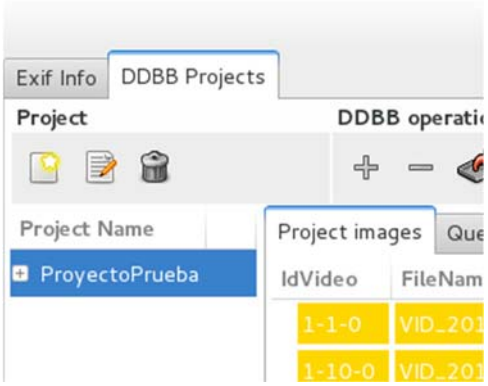


Figura A.20: Proyecto editado

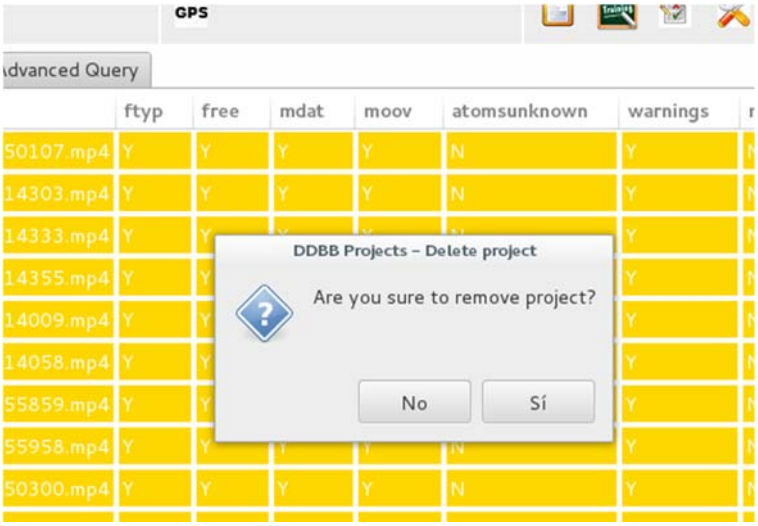


Figura A.21: Eliminar un proyecto

A parte de editar o eliminar un proyecto, también podemos añadir archivos al mismo, Figura A.22

También cuenta con el filtro para la extensión de archivos como en el caso de crear un nuevo proyecto, Figura A.23, y un control por si no se selecciona ningún archivo a añadir, Figura A.24.

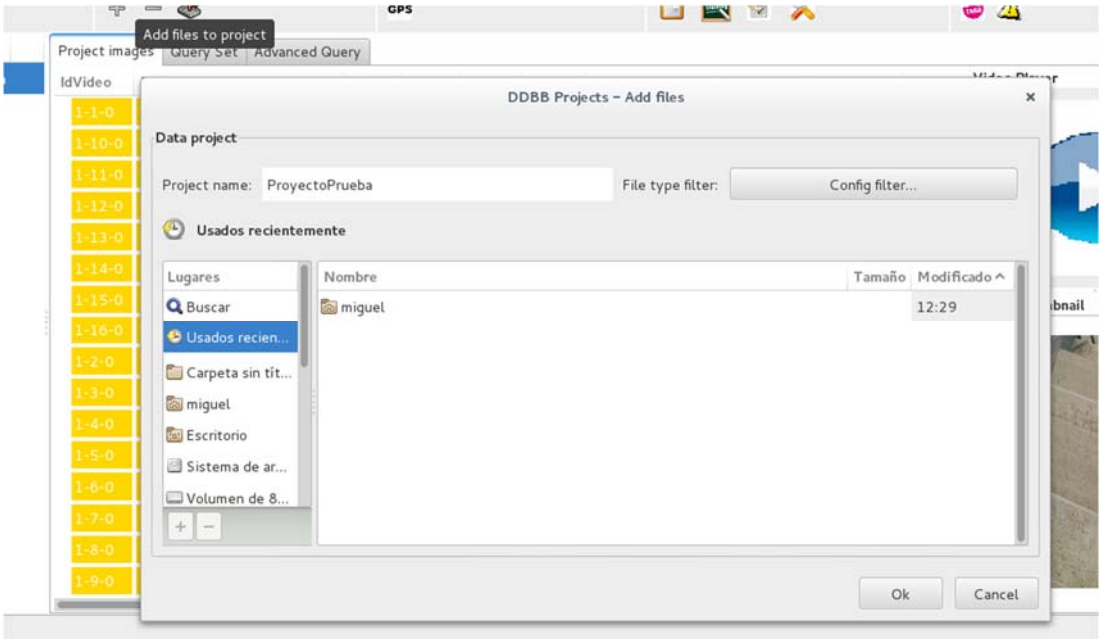


Figura A.22: Añadir archivos al proyecto

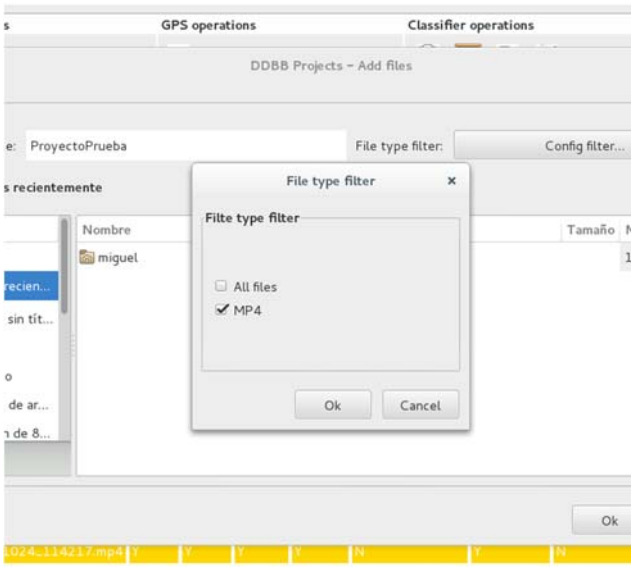


Figura A.23: Añadir archivos al proyecto. Aplicar filtro de extensión de los archivos.

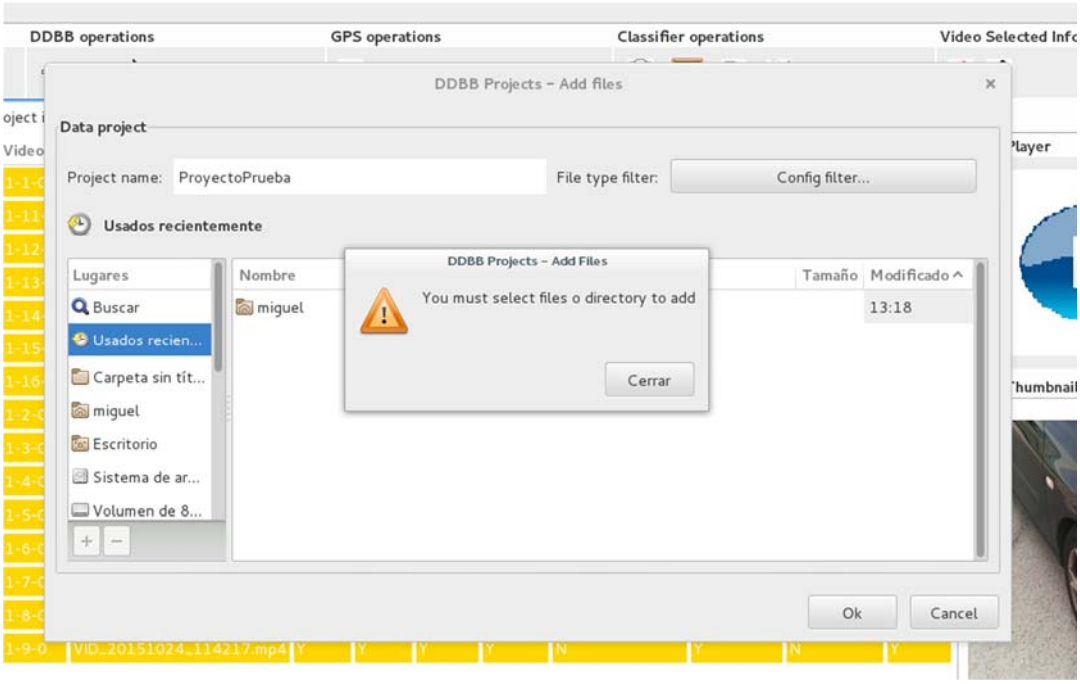


Figura A.24: Añadir archivos al proyecto. Error, archivos no seleccionados.

Al igual que podemos añadir archivos, también podemos eliminarlos. En esta parte, como se observa en la Figura A.25, se puede seleccionar uno o más archivos a eliminar.

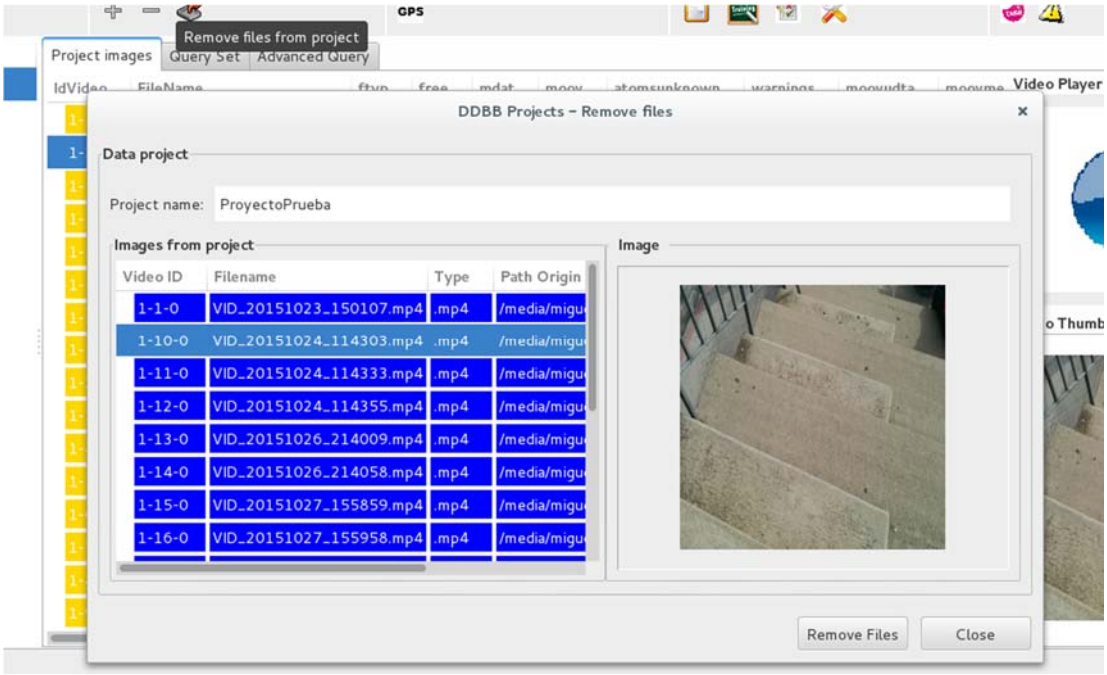


Figura A.25: Eliminar archivos del proyecto

Dado que se está trabajando desde una base de datos, se ha incorporado la funcionalidad de exportar uno o más vídeos de la base de datos a un medio físico de almacenamiento, para que el usuario pueda obtenerlos y disponer de ellos de manera física y de forma independiente a la base de datos.

Cuenta con un control e informa al usuario por si no ha seleccionado un directorio, Figura A.26, o al menos un vídeo que exportar, Figura A.27. Una vez seleccionado el/los vídeo/s a exportar y un directorio donde realizar la operación, informa al usuario de los vídeos exportados correctamente, Figura A.28. De no producirse ningún error se puede ir a la carpeta seleccionada y comprobar que efectivamente los vídeos se han exportado correctamente, Figura A.29.

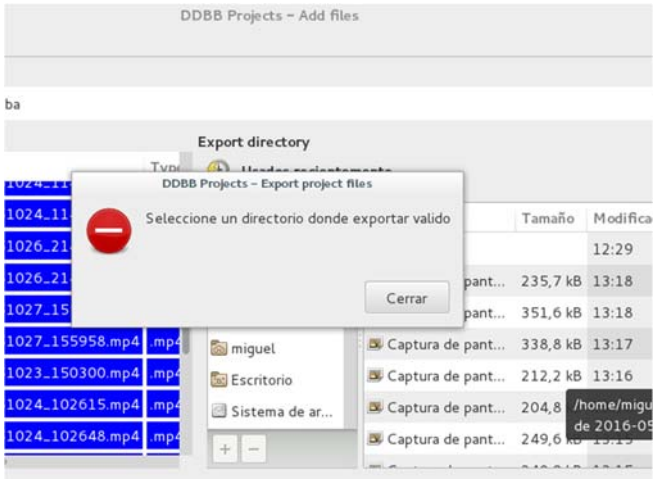


Figura A.26: Exportar vídeos. Error, directorio no seleccionado.

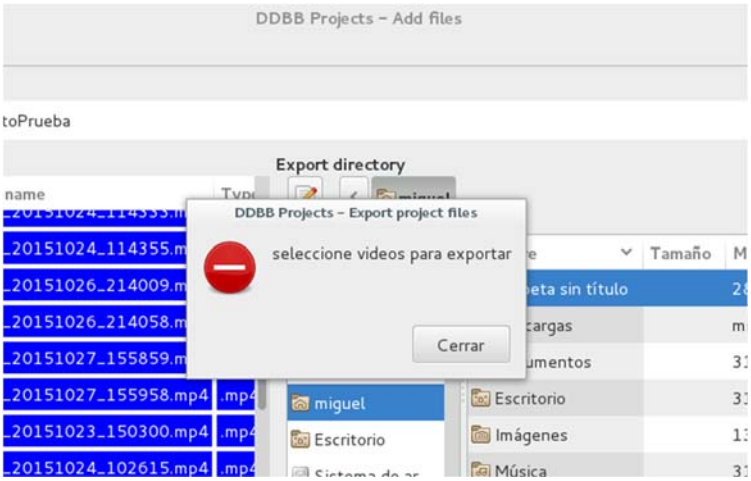


Figura A.27: Exportar vídeos. Error, vídeos/s no seleccionado.

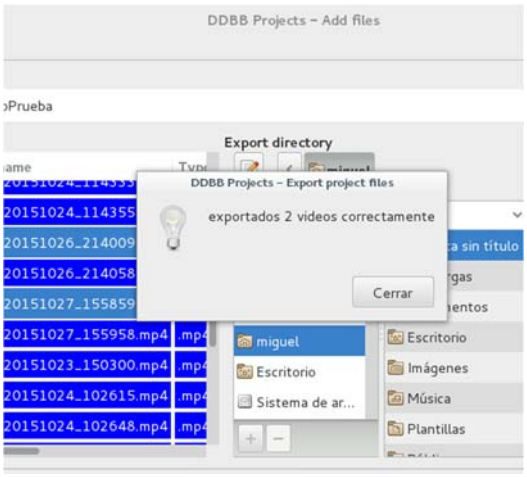


Figura A.28: Exportar vídeos. Informa de vídeos exportados.

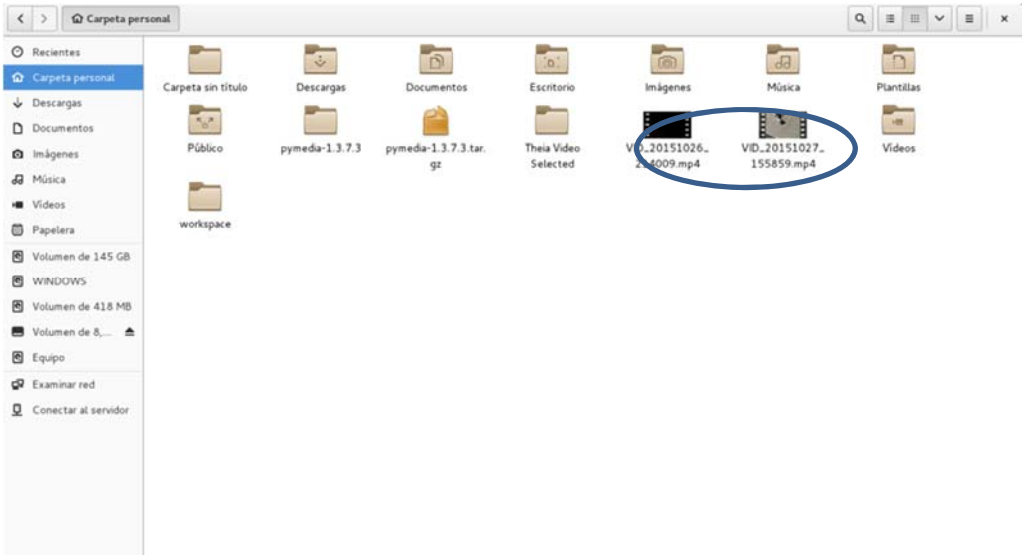


Figura A.29: Exportar vídeos. Comprobación de exportación exitosa.

Para finalizar con las utilidades de la herramienta, se ha desarrollado la opción de generar mapas con vídeos del proyecto. Inicialmente muestra los vídeos con información GPS y los vídeos sin dicha información, como en la Figura A.30:

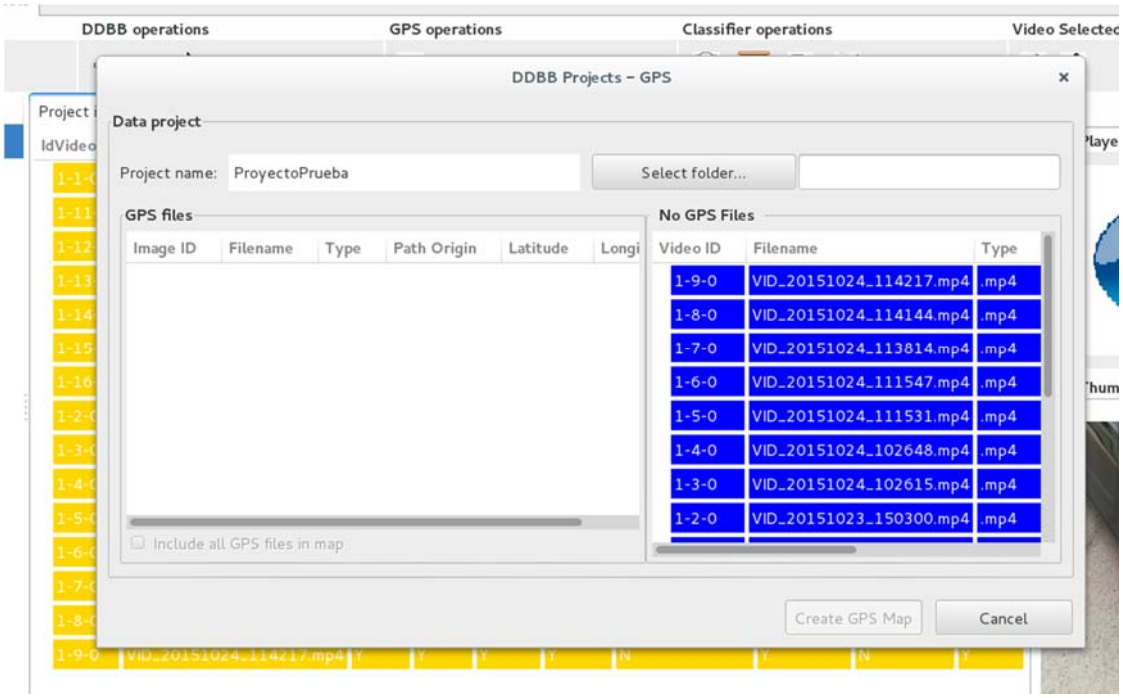


Figura A.30: GPS. Informe de vídeos con datos GPS.

Como se observa en la Figura A.30, en este proyecto no aparece ningún vídeo con datos GPS, para ello vamos a añadir tres vídeos, Figura A.31, que sí los contiene y así poder poner un ejemplo de esta utilidad.

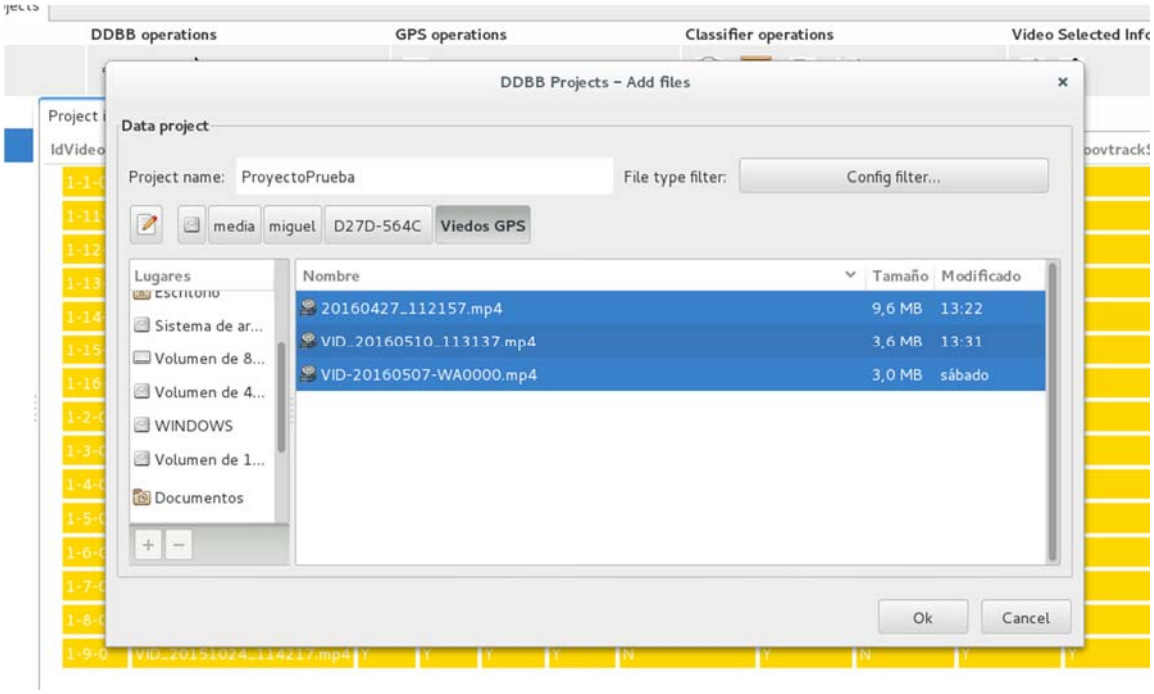


Figura A.31: Añadir archivos al proyecto

Tras añadirlos, volvemos a hacer uso del botón GPS, y ahora como se muestra en la Figura A.33, podemos ver que el proyecto ya contiene vídeos con información GPS. También debemos seleccionar un directorio para guardar el mapa GPS que vamos a utilizar, Figura A.32. En la Figura A.32 también se puede observar que arriba a la derecha ya tenemos el directorio seleccionado.

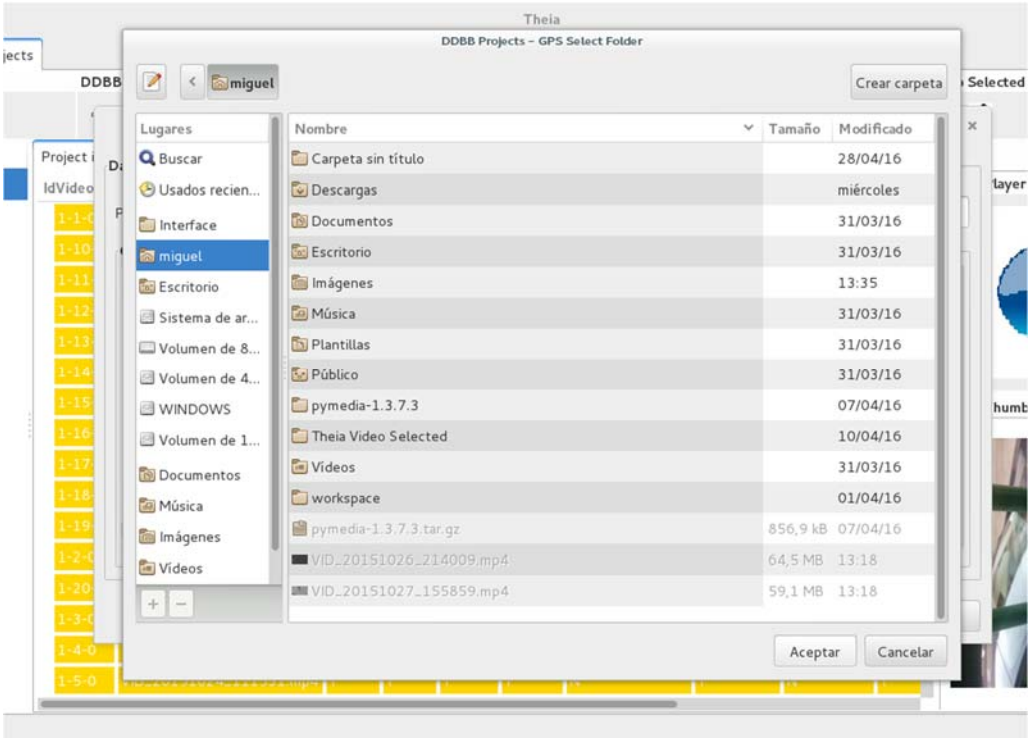


Figura A.32: GPS. Seleccionar directorio para guardar la información del mapa GPS

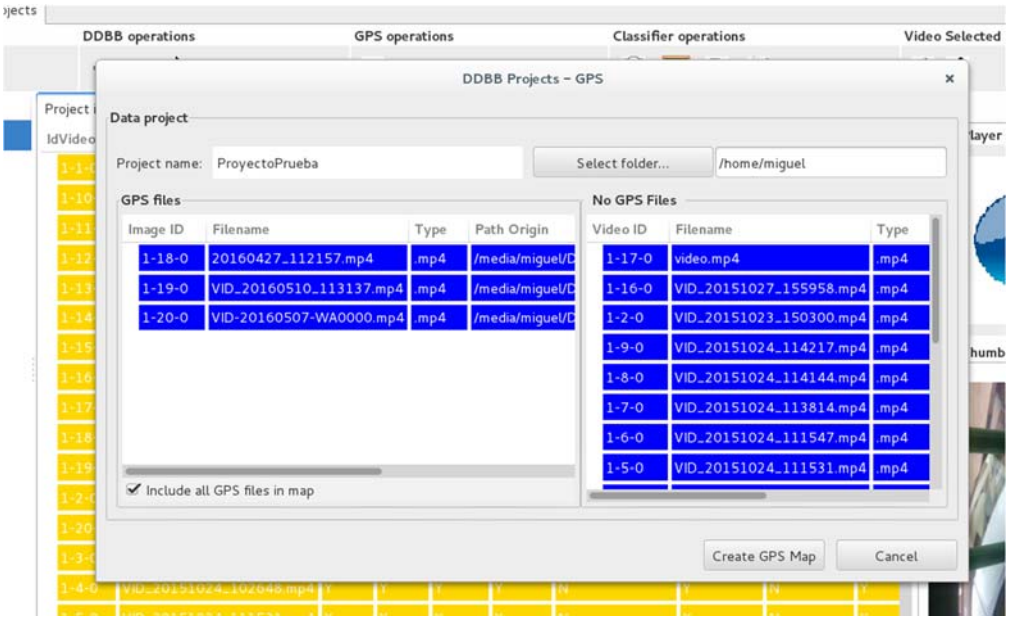
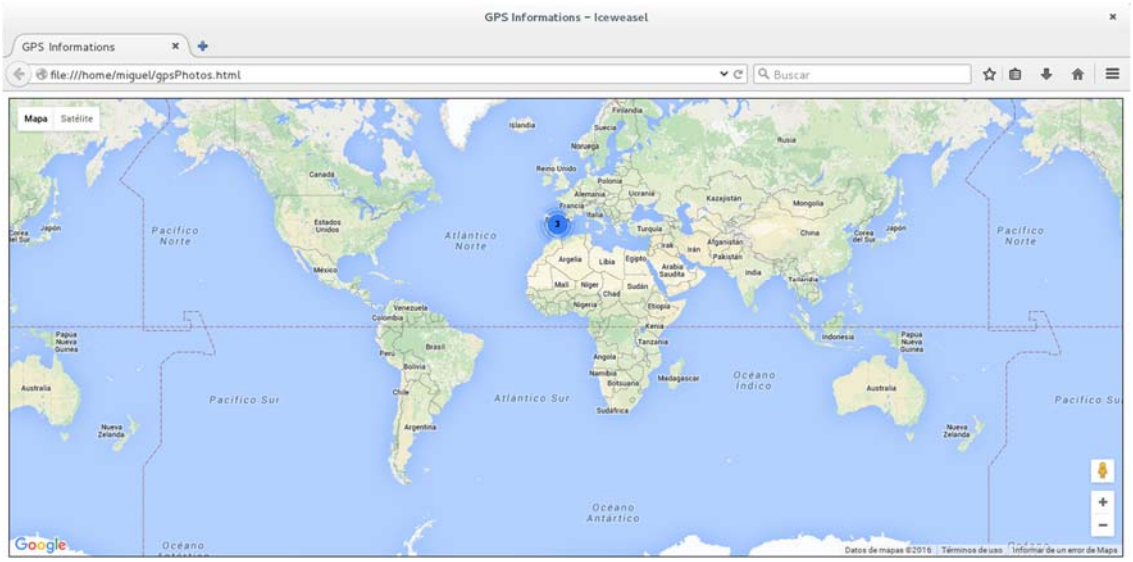


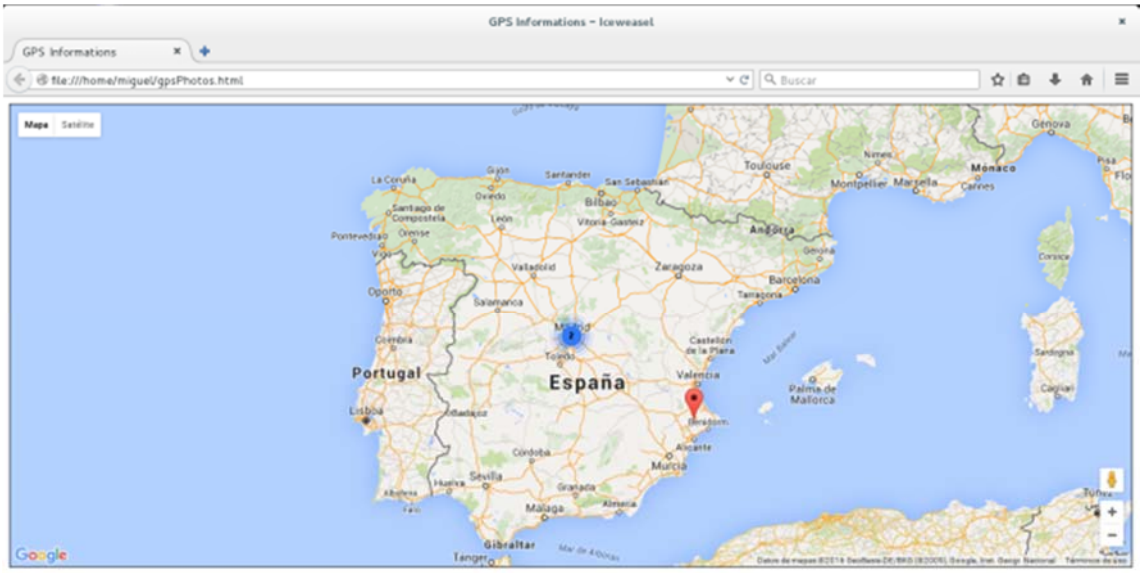
Figura A.33: GPS. Informe de vídeos con datos GPS.

En la Figura A.33 se tiene que seleccionar los archivos que queremos que aparezca en nuestro mapa, estos archivos deben pertenecer a los de la parte de la izquierda ya que son los que contienen información GPS y puesto que los de la parte derecha, la aplicación no deja deseleccionarlos. También dispone de la opción de seleccionar todos los archivos para no tener que ir seleccionando a mano todos ellos. Esta opción se puede observar en la Figura A.33, justo debajo de los vídeos con información GPS, que en este caso dicha opción ha sido marcada.

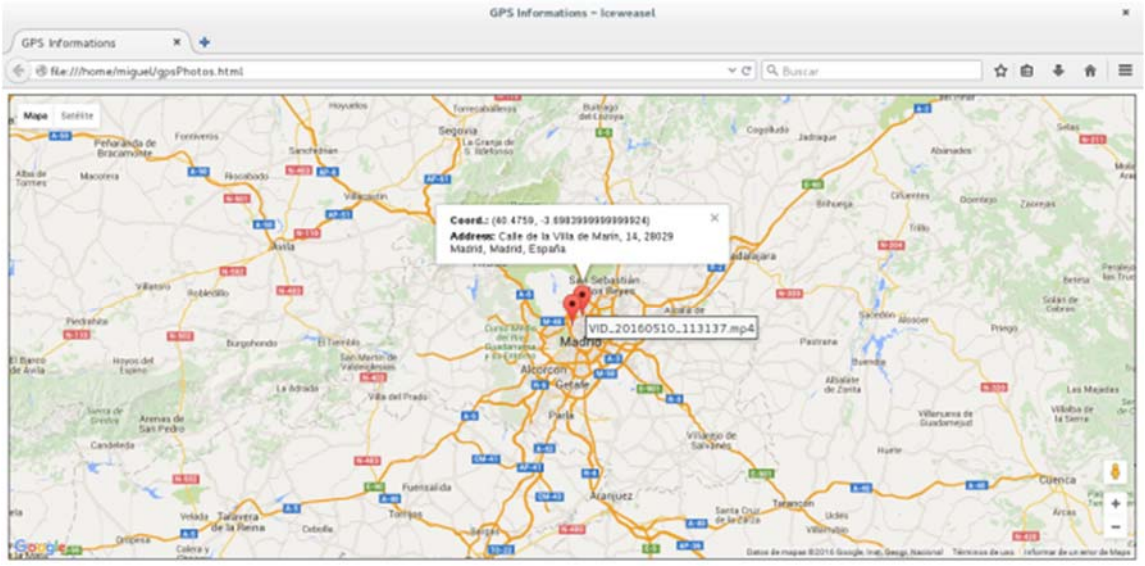
Tras seleccionar un directorio, los archivos y hacer click en el botón "Create GPS Maps", se abre el navegador predeterminado del usuario y aparece un mapa como el de la Figura A.34.a. Los vídeos si están muy lejos aparecen como grupo y a medida que nos acercamos se van haciendo más visibles, Figura A.34.b. Permite pulsar encima para conocer la localización exacta como en la Figura A.34.c.



a. Mapa GPS. Visibilidad mundial.



b. Mapa GPS. Visibilidad de España



c. Mapa GPS. Selección de vídeo

Figura A.34: Información GPS en vídeos.

REFERENCIAS

[1] T. Ahonen and A. Moore, "Tomi Ahonen Almanac 2014: Mobile Telecoms Industry Annual Review," http://goo.gl/B1eX8, 2014.

[2] R. Baer, "Resolution Limits in Digital Photography: The Looming End of the Pixel Wars," in Proceedings of the Imaging Systems Conference, Tucson, Arizona United States, June 2010.

[3] IC Insights Inc , "Embedded Imaging Takes Off as Stand-alone Digital Cameras Stall," 2013. [Online]. Available: http://www.icinsights.com/data/articles/documents/484.pdf

[4] "Alexa Top 500 Global Sites," http://www.alexa.com/topsites, 2016.

[5] M. Al-Zarouni, "Mobile Handset Forensic Evidence: a Challenge for Law Enforcement," in Proceedings of the 4th Australian Digital Forensics Conference, Perth Western, Australia, December 2006, pp. 1–10.

[6] C. Y. Wen and K. T. Yang, "Image Authentication for Digital Image Evidence," Forensic Science Journal, vol. 5, no. 1, pp. 1–11, September 2006.

[7] V. L. L. Thing, K. Y. Ng, and E. C. Chang, "Live Memory Forensics of Mobile Phones," Digital Investigation, vol. 7, pp. 74–82, August 2010.

[8] J. Almeida et al. / J. Vis. Commun. Image R. 24 (2013).

[9] 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, vol. 1, 11 2012.

[10] T. Bianchi, A. De Rosa, and A. Piva, "Improved DCT coefficient analysis for forgery localization in JPEG images," in Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on, 2011.

[11] T. Sun, W. Wang, and X. Jiang, "Exposing vídeo forgeries by detecting MPEG double compression," in Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on, 2012, pp. 1389–1392.

[12] X. Jiang, W. Wang, T. Sun, Y. Shi, and S. Wang, "Detection of double compression in MPEG-4 videos based on Markov statistics," Signal Processing Letters, IEEE, vol. 20, no. 5, pp. 447–450, 2013.

[13] Localization of Forgeries in MPEG-2 Vídeo through GOP Size and DQ Analysis D. Labartino #1, T. Bianchi #2, A. De Rosa #3, M. Fontani #4, D. Vázquez-Padín #5, A. Piva #6, M. Barni #7.

[14] Detection of vídeo double encoding with GOP size estimation, D. Vázquez-Padín#1, M. Fontani#2, T. Bianchi#3, P. Comesaña#4, A. Piva#5, M. Barni#6.

[15] E. Hamilton, "JPEG File Interchange Format. Version 1.02, September 1, 1992," http://www.w3.org/Graphics/JPEG/jfif3.pdf.

[16] S. Committee, "Exchangeable Image File for digital still cameras: Exif version 2.3, April 26, 2010," http://goo.gl/jgrCpC, 2013.

[17] M. J. Kaur and N. Sharma, "Survey on the General Concepts of MPEGMoving Picture Experts Group," PARIPEX-Indian Journal of Research, vol. 5, no. 2, 2016.

[18] B. G. Haskell, A. Puri, and A. N. Netravali, Digital vídeo: an introduction to MPEG-2. Springer Science & Business Media, 1996.

[19] I. E. Richardson, H.264 and MPEG-4 Vídeo Compression: Vídeo Coding for Next-Generation Multimedia. John Wiley & Sons, 2004.

[20] Apple Computer, Inc., "Introduction to QuickTime File Format Specification," https://goo.gl/6rs8uB, 2016.

[21] Microsoft Developer Network, "AVI RIFF File Reference," http://msdn. microsoft.com/en-us/library/ms779636(VS.85).aspx, 2016.

[22] https://mh-nexus.de/en/hxd/

[23] http://www.headbands.com/gspot/

[24] https://mediaarea.net/es/MediaInfo

[25] http://www.sno.phy.queensu.ca/~phil/exiftool/

[26] http://www.kcsoftwares.com/?vtb

[27] http://www.kcsoftwares.com/?sumo

[28] C. Lakshmanan, P. Mittal, S. Sehgal, and P. Sinha, "MP4 Container File Formats and Methods of Processing MP4 Container Files," November 2015, uS Patent 9,185,468.

[29] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Vídeo Coding Extension of the H.264/AVC Standard," Proceedings of the IEEE Transactions on Circuits and Systems for Vídeo Technology, vol. 17, no. 9, pp. 1103–1120, September 2007.

[30] P. Bestagini, M. Fontani, S. Milani, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro, "An Overview on Vídeo Forensics," in Proceedings of the 20th European Signal Processing Conference, Bucharest, Romania, August 2012, pp. 1229–1233.

[31] T. Gloe, M. Kirchner, A. Winkler, and R. Bohme, "Can We Trust Digital Image Forensics?" in Proceedings of the 15th International Conference on Multimedia, Augsburg, Germany, September 2007, pp. 78–86.

[32] T. Van Lanh, K. S. Chong, S. Emmanuel, and M. S. Kankanhalli, "A Survey on Digital Camera Image Forensic Methods," in Proceedings of the IEEE International Conference on Multimedia and Expo, Beijing, China, July 2007, pp. 16–19.

[33] A. L. Sandoval Orozco, J. Rosales Corripio, D. M. Arenas Gonz´alez, L. J. Garc´ıa Villalba, and J. C. Hern´andez Castro, "Techniques for Source Camera Identification," in

Proceedings of the 6th International Conference on Information Technology, Amman, Jordan, May 2013, pp. 1–9.

[34] A. L. Sandoval Orozco, D. M. Arenas González, L. J. García Villalba, and J. C. Hernández-Castro, "Anomalías en el Seguimiento de Exif en el Análisis Forense de Metadatos de Imágenes de Móviles," in Actas del XII Reunión Española sobre Criptología y Seguridad de la Información, Donostia-San Sebastián, España, Septiembre 2012.

[35] A. L. Sandoval Orozco, D. M. Arenas Gonz´alez, L. J. Garc´ıa Villalba, and J. C. Hern´andez Castro, "Analysis of Errors in Exif Metadata on Mobile Devices," Multimedia Tools and Applications, vol. 68, no. 1, pp. 1–29, January 2014.

[36] Y. Su, J. Xu, and B. Dong, "A Source Vídeo Identification Algorithm Based on Motion Vectors," in Proceedings of the Second International Workshop on Computer Science and Engineering, vol. 2, Qingdao, China, October 2009, pp. 312–316.

[37] S. Yahaya, A. T. S. Ho, and A. A. Wahab, "Advanced Vídeo Camera Identification Using Conditional Probability Features," in Proceedings of the IET Conference on Image Processing, London, UK, July 2012, pp. 1–5.

[38] https://www.google.es/intl/es/earth/