



SISTEMAS INFORMÁTICOS

CURSO 2010-2011

**ImgTool: Herramienta para el Análisis
Forense de Imágenes tomadas desde
Dispositivos Móviles**

David de la Fuente González

José Ignacio Pérez Solís

Juana María Vázquez Sagrado

Director:

Luis Javier García Villalba

Departamento de Ingeniería del Software e Inteligencia Artificial

FACULTAD DE INFORMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID

Abstract

Nowadays, the use of mobile devices has increased significantly, mainly due to its easy way of use, usefulness and the increasing necessity among users of being connected. From this, there has been a significant change in the way people communicate, but also these devices have been increasingly used in criminal activities and attacks.

The need for mobility and availability of information in organizations and companies, which has led to the increasing number of these mobile devices, rocketed the number of security incidents (theft of proprietary information and customer data loss).

Computer forensics is the application of scientific and analytical techniques to acquire, preserve, collect and present data(that have been processed and stored or transmitted electronically through a computer environment), which are valid in a legal proceeding. Forensic analysis carried out on a mobile device, may be admitted in court, by a judge.

In this application pertaining to computer forensics We pretend to show the fraudulent actions that can be carried out in photographs taken from mobile devices. That's possible thanks to meta-information available in the photo so We can check if the photo was taken from a device or not.

Keywords

Camera mobile phones, forensics, image classification, support vector machines, fraudulent photographs.

Resumen

En la actualidad, el uso de dispositivos móviles ha aumentado considerablemente, debido principalmente a su facilidad de uso, utilidad y la necesidad cada vez mayor entre los usuarios de estar conectados. Debido a esto, no solo ha habido un cambio significativo en la forma de comunicarse, sino que también estos dispositivos han sido cada vez más utilizados en actividades criminales y ataques.

La necesidad de la movilidad y la disponibilidad de la información en las organizaciones y empresas, ha llevado al aumento del número de estos dispositivos móviles, y al aumento de número de incidentes de seguridad (robo de información propietaria y la pérdida de datos de clientes).

La informática forense es la aplicación de técnicas científicas y analíticas para adquirir, conservar, recopilar y presentar datos (que han sido procesados y almacenados o transmitidos electrónicamente por medio de un entorno informático), que son válidos en un proceso legal. El análisis forense realizado en un dispositivo móvil, podrá ser admitido en la corte, por un juez.

En esta aplicación perteneciente a la informática forense, se pretende mostrar las acciones fraudulentas que pueden llevarse a cabo en las fotografías tomadas desde dispositivos móviles. Esto es posible gracias a la meta-información disponible en la foto que permite que se pueda comprobar si la foto fue tomada desde un dispositivo o no.

Palabras clave

Teléfonos móviles con cámara, informática forense, clasificación de imágenes, maquinas de soporte vectorial, fraudulent photographs.

Autorizamos a la Universidad Complutense de Madrid difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

David de la Fuente González

José Ignacio Pérez Solís

Juana María Vázquez Sagrado

Lista de acrónimos

APP	Application Marker Segment
CCD	Charge-Coupled Device
CFA	Color Filter Array
CMOS	Complementary Metal-Oxide-Semiconductor
CMYK	Cyan, Magenta, Yellow, Key
DCT	Discrete Cosine Transform
EXIF	Exchangeable Image File
GPS	Global Positioning System
IFD	Image File Directory
IPTC	International Press Telecommunications Council
JFIF	JPEG File Interchange Fomat
JPEG	Joint Photographic Experts Group
PCA	Principal Component Analysis
RBF	Radial Basis Function
RGB	Red Green Blue
SVM	Support Vector Machine
SVM- DAG	SVM Directed Acyclic Graph
TIFF	Tagged Image File Format

ÍNDICE

1	INTRODUCCIÓN	1
1.1	OBJETO DE LA INVESTIGACIÓN	2
1.2	ESTRUCTURA DEL TRABAJO	2
2	APLICACIONES DEL ANÁLISIS FORENSE EN IMÁGENES DIGITALES	5
2.1	DISPOSITIVOS MÓVILES CON CÁMARA FOTOGRÁFICA	5
2.2	PROCESO DE ADQUISICIÓN DE UNA IMAGEN	6
2.2.1	Sensores	6
2.2.2	Proceso de Adquisición de Imágenes	8
2.3	CARACTERÍSTICAS DE EXTRACCIÓN DE UNA IMAGEN	10
2.3.1	Coeficiente de interpolación del color como característica	10
2.3.2	Ruido estadístico como característica de extracción	11
2.4	ESTADÍSTICAS SOBRE TELEFONÍA MÓVIL	11
2.4.1	Estadísticas sobre ventas y usuarios	11
2.4.2	Estadísticas sobre uso de dispositivos móviles	15
3	FORMATOS DE ARCHIVOS INTERCAMBIABLES	17
3.1	TAGGED IMAGE FILE SPECIFICATION 6.0	17
3.1.1	Ámbito de TIFF	17
3.1.2	Características TIFF	18
3.2	JPEG FILE INTERCHANGE FOMAT	18
3.3	INTERNATIONAL PRESS TELECOMMUNICATIONS COUNCIL	18
3.4	EXCHANGEABLE IMAGE FILE	19
3.4.1	Introducción a la especificación del archivo de imagen EXIF	20
3.4.2	Características EXIF	20
3.4.3	Especificación del archivo de imagen EXIF	20
3.4.3.1	Especificación de los datos de la imagen	21
3.4.3.2	Estructura básica de los datos de la imagen	22
3.4.3.3	Etiquetas	27
3.4.4	Estructura de IFD	27
3.4.4.1	Etiqueta	28
3.4.4.2	Tipo	28
3.4.4.3	Contador	29
3.4.4.4	Valor de desplazamiento	29
3.4.5	Exif especifica IFD (<i>Image File Directory</i>)	29
3.4.5.1	Exif IFD	29
3.4.5.2	GPS IFD	30
3.4.5.3	Interoperabilidad IFD	30
4	GESTIÓN DEL PROYECTO	31
4.1	PLANIFICACIÓN	31
4.2	HERRAMIENTAS UTILIZADAS	35
4.2.1	Python v. 2.6	35
4.2.2	Glade v. 3.0	36
4.2.3	MySql v. 5.0	36
4.2.4	Asamblea Workspace	36
4.3	ANÁLISIS DE APLICACIONES EXISTENTES	36
4.3.1	PhotoInfoEx	36
4.3.2	Jhead	37
4.3.3	Exiftool	37

4.3.4. Exif viewer (Complemento Firefox).....	37
5. VISIÓN GENERAL DEL SISTEMA	39
5.1. DESCRIPCIÓN DE LA APLICACIÓN.....	39
5.2. MODELO DE DATOS.....	40
5.3. ESPECIFICACIÓN DE TABLAS	41
5.3.1. Tablas de Configuración.....	41
5.3.1.1. Data_Set.....	42
5.3.1.2. Data_Dictionary	42
5.3.1.3. Table_Dictionary.....	43
5.3.2. Tablas de Generación de Consultas	43
5.3.2.1. Query_Set.....	43
5.3.2.2. Adv_Query	43
5.3.2.3. AQ_Select.....	44
5.3.2.4. AQ_Where	44
5.3.2.5. AQ_Val_Where.....	44
5.3.3. Tablas de Información EXIF	45
5.3.3.1. Main_Info.....	45
5.3.3.2. Image_Info	45
5.3.3.3. Thumbnail_Info.....	47
5.3.3.4. Exif_Info	48
5.3.3.5. Interoperability_Info.....	51
6. IMPLEMENTACIÓN.....	52
6.1. ASPECTOS RELEVANTES DE LA IMPLEMENTACIÓN.....	52
6.2. FUNCIONALIDAD DE LA APLICACIÓN	53
6.2.1. Exif Information.....	54
6.2.1.1. Menú Image.....	55
6.2.1.2. Menú GPS	56
6.2.2. DDBB Projects	57
6.2.2.1. Menú Project.....	58
6.2.2.2. Menú DDBB Operations.....	60
6.2.2.3. Menú GPS Operations.....	62
6.2.2.4. Project Images.....	64
6.2.2.5. Query Set.....	64
6.2.2.6. Advanced Query	66
7. CONCLUSIONES Y TRABAJO FUTURO.....	69
7.1. CONCLUSIONES	69
7.1.1. Conocimientos Adquiridos:	69
7.2. TRABAJO FUTURO.....	70
REFERENCIAS	73
ANEXOS	77
A. CLASIFICACIÓN.....	79
A.1. CLASIFICADORES	80
A.1.1. Aprendizaje	80
A.1.1.1. Aprendizaje supervisado	81
A.1.1.2. Aprendizaje no supervisado.....	81
A.1.1.3. Aprendizaje semi-supervisado.....	81
A.1.1.4. Aprendizaje por refuerzo.....	81
A.2. TIPOLOGÍA DE CLASIFICADORES.....	82
A.2.1. Clasificador Bayesiano	82
A.2.2. Clasificador Parzen.....	82
A.2.3. Clasificador Backpropagation.....	82

A.2.4. Clasificador con PCA	83
A.2.5. Máquinas de vectores de soporte (Support vector machine).....	83
A.3.SUPPORT VECTOR MACHINE (SVM)	83
A.3.1 Clasificación SVM Multiclase (k clases).....	88
A.3.1.1.SVM uno vs resto (1 vs R)	88
A.3.1.2.SVM- DAG (Directed Acyclic Graph).....	88
B. CODIFICACIÓN PARA LA CLASIFICACIÓN DE IMÁGENES	90
B.1. LIBRERÍA LIBSVM.....	90
B.1.1. Introducción	90
B.1.2. Preprocesamiento de la información.....	91
B.1.2.1. Escalado.....	91
B.1.2.2. Kernel RBF	91
B.1.2.3. Validación cruzada y búsqueda en rejilla	91
B.2. FILTROS SOBRE IMÁGENES	91
B.2.1. Filtro de la media	91
B.2.2. Filtro de la mediana.....	92
B.2.3. Filtro Gaussiano	92
C. APLICACIONES.....	93

ÍNDICE DE TABLAS

TABLA. 2.1.	DISTRIBUCIÓN DE DISPOSITIVOS MÓVILES POR PAÍSES.	11
TABLA. 2.2.	DISTRIBUCIÓN DE DISPOSITIVOS MÓVILES POR MARCAS (2009-2010).	14
TABLA. 2.3.	DISTRIBUCIÓN DE DISPOSITIVOS MÓVILES POR SISTEMA OPERATIVO (2009-2010).....	14
TABLA. 4.1.	TABLA COMPARATIVA ENTRE APLICACIONES EXISTENTES.....	38
TABLA. B.2.	VENTAJAS E INCONVENIENTES DE APLICAR FILTRO DE LA MEDIANA.	92

ÍNDICE DE FIGURAS

FIG. 2.1. ESTRUCTURA INTERNA DEL SENSOR CCD	7
FIG. 2.2. ESTRUCTURA INTERNA DEL SENSOR CMOS	8
FIG. 2.3. PATRONES CFA	9
FIG. 2.4. CRECIMIENTO DE USUARIOS DE DISPOSITIVOS MÓVILES 2000-2010. (FUENTE: ITU)	13
FIG. 2.5. FRECUENCIA DE USO DE CÁMARAS EN DISPOSITIVOS MÓVILES	15
FIG. 3.1. COMPOSICIÓN Y MUESTREO PARA UNA IMAGEN DE ANCHO W Y ALTURA H.....	22
FIG. 3.2. ESTRUCTURA BÁSICA DE LOS ARCHIVOS DE DATOS SIN COMPRIMIR	24
FIG. 3.3. ESTRUCTURA BÁSICA DE ARCHIVOS DE DATOS COMPRIMIDOS	26
FIG. 3.4. ESTRUCTURA DEL IMAGE FILE DIRECTORY IFD.....	28
FIG. 3.5. ESTRUCTURA DEL CAMPO I-ÉSIMO DEL ARRAY DE INTEROPERABILIDAD	29
FIG. 4.1. DIAGRAMA DE GANTT.....	34
FIG. 5.1. DIAGRAMA DE ENTIDAD - RELACIÓN.....	41
FIG. 6.1. EXIF INFORMATION.....	54
FIG. 6.2. FILE TYPE FILTER	55
FIG. 6.3. IMAGE SIZE.....	56
FIG. 6.4. INFORMACIÓN GPS DE UNA FOTOGRAFÍA ANALIZADA.....	56
FIG. 6.5. INFORMACIÓN GPS EN FORMATO KML.....	57
FIG. 6.6. PESTAÑA DDBB PROJECT.....	58
FIG. 6.7. NEW PROJECT.....	59
FIG. 6.8. EDIT PROJECT	59
FIG. 6.9. DELETE PROJECT	60
FIG. 6.10. ADD FILES.....	61
FIG. 6.11. REMOVE FILES.....	61
FIG. 6.12. EXPORT FILES.....	62
FIG. 6.13. GPS INFORMATION	63
FIG. 6.14. GPS INFORMATION	63
FIG. 6.15. PROJECT IMAGES.....	64
FIG. 6.16. QUERY SET	65
FIG. 6.17. QUERY SET	65
FIG. 6.18. ADVANCED QUERY	66
FIG. 6.19. ADVANCED QUERY -SELECT COLUMNS	67
FIG. 6.20. ADVANCED QUERY - FILTER COLUMNS	67
FIG. 6.21. ADVANCED QUERY -RUN QUERY	68
FIG. A.1. REPRESENTACIÓN GRÁFICA DE MÁQUINAS DE SOPORTE VECTORIAL	84

1 INTRODUCCIÓN

En la actualidad, el empleo de dispositivos móviles se ha incrementado notablemente, principalmente, por su facilidad de uso y la utilidad de mantener en contacto permanente a sus usuarios. A partir de esto, se ha generado un cambio significativo en la forma en que las personas se comunican, pero también se ha incrementado su uso en actividades delictivas y los ataques a las vulnerabilidades que presentan.

Por la necesidad de movilidad y disponibilidad de la información en las organizaciones y por consiguiente el aumento del uso de estos dispositivos móviles, se han incrementado los incidentes de seguridad (robo de información de propietarios y pérdida de datos de clientes) que ocurren sobre ellos [1].

La informática forense es la aplicación de técnicas científicas y analíticas para adquirir, preservar, obtener y presentar datos, que han sido procesados electrónicamente y almacenados o transmitidos a través de un medio informático, que sean válidos en un proceso legal.

Dichas técnicas incluyen reconstruir el bien informático, examinar datos residuales, autenticar datos y explicar las características técnicas del uso aplicado a los datos y bienes informáticos.

Para la protección y el análisis de datos hay que tener presente la información disponible de los datos (visibles y metadatos) [2].

Un análisis forense que se lleve a cabo sobre un dispositivo móvil, en el juzgado, puede ser admitido o no dependiendo de lo que considere el juez que lleve dicho caso y la formalidad con que se desarrolle el procedimiento de recolección, control, análisis y presentación de las evidencias.

En el presente trabajo, perteneciente a la informática forense, se trata del diseño e implementación de una herramienta de apoyo que permita resolver

las acciones fraudulentas que se puedan llevar a cabo en las fotografías tomadas desde dispositivos móviles. Este proceso se realiza a partir de la metainformación de la que dispone el dispositivo móvil, para poder llegar a verificar si dicha fotografía ha sido tomada desde cierto dispositivo o no.

1.1 Objeto de la investigación

En este documento, introducimos un nuevo enfoque en el análisis forense sobre imágenes con el objetivo de determinar el tipo concreto de dispositivo móvil que se utilizó para adquirir una imagen. Presentamos características básicas de clasificación enfocadas a facilitar la adquisición forense, y mostramos que el método propuesto proporciona unos muy altos niveles de identificación del origen de la imagen digital, diferenciándolo de los diferentes dispositivos móviles desde dónde fueron tomadas.

Básicamente, clasificamos imágenes teniendo en cuenta características distintas a las apuntadas en la metainformación existente en la imagen tomada (EXIF).

1.2 Estructura del trabajo

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

En el capítulo 1, se hace una introducción sobre la informática forense y su relación con este trabajo, así como el objeto de la investigación.

En el capítulo 2, se realiza un estado del arte sobre la adquisición de imágenes, tanto en general, como de dispositivos móviles, adquisición y extracción de características de imágenes y algunas estadísticas de dispositivos móviles.

En el capítulo 3, repasamos los distintos tipos de formato de archivo intercambiables, ya que es necesario este conocimiento para el desarrollo y

entendimiento de éste trabajo.

En el capítulo 4, se explica el proceso de gestión del proyecto, tanto la planificación como las herramientas utilizadas. También se citan diversas herramientas existentes.

En el capítulo 5, se da una visión general del sistema, incluyendo el modelo de datos resultante y la especificación de tablas.

En el capítulo 6, se repasa las partes más importantes en la implementación del proyecto.

En el capítulo 7, se muestra una visión sobre trabajo futuro, en cuanto a la mejora del proyecto por clasificadores estadísticos y su implementación.

Como Anexo A, se profundiza sobre la teoría de los clasificadores estadísticos, y en especial, SVM (máquina de soporte vectorial).

Como Anexo B, se detalla la implementación del clasificador SVM en una librería LIBSVM.

Como Anexo C, se citan las aplicaciones que podrían tener estos clasificadores.

2. APLICACIONES DEL ANÁLISIS FORENSE EN IMÁGENES DIGITALES

2.1. Dispositivos Móviles con Cámara Fotográfica

Un teléfono con cámara fotográfica es un teléfono móvil que tiene una cámara fotográfica incorporada y que junto a una infraestructura basada en servidor permite al usuario compartir fotos y vídeos con cualquier persona de forma inmediata.

El teléfono con cámara fotográfica, como muchos sistemas complejos, es el resultado de la convergencia de tecnologías.

El primer teléfono comercial con cámara fotográfica fue el J-SH04, hecho por Sharp Corporation, que tenía un sensor integrado CCD, con la infraestructura del Sha-Cor.

El primer despliegue de los celulares con cámaras en Norteamérica fue en 2002; con el primer teléfono con cámara fotográfica fue fabricado por Sanyo y distribuido por Sprint con la infraestructura de PictureMail se convirtió y manejó por LightSurf. Como la mayoría de los sistemas basados en tecnología compleja, hay varias patentes y las tecnologías relevantes a los aspectos de la cámara fotográfica telefónica. El advenimiento del sensor CMOS es una tecnología que permite para la producción en masa.

Las imágenes se guardan generalmente en formato JPEG. La resolución está, usualmente en la gama del megapixel.

Sabiendo de antemano las características de la interpolación del color o el balanceo de blancos, se puede utilizar para diferenciar el tipo de dispositivo que realizó la imagen. El ruido obtenido en el estudio de las imperfecciones del sensor, también se puede utilizar para la identificación del modelo.

2.2. Proceso de Adquisición de una Imagen

2.2.1. Sensores

El sensor, es un elemento hardware que se encuentra dentro de la cámara, tanto en las cámaras digitales tradicionales como en las cámaras de los teléfonos móviles. Está formado por diminutas células fotoeléctricas que registran la luz que reciben. Desde allí, la imagen es procesada por la cámara y registrada en la tarjeta de memoria [3].

Los sensores son, en realidad, un circuito integrado, compuesto por millones de pequeños semiconductores de silicio, los cuales captan los fotones (elementos que componen la luz, la electricidad). A mayor intensidad de luz, más carga eléctrica existirá. Estos fotones desprenden electrones dentro del sensor de imagen, los cuales se transformarán en una serie de valores (datos digitales) creando un píxel. Por lo tanto cada célula que desprenda el sensor de imagen se corresponde a un píxel, el cual, formará cada punto de la imagen.

La capacidad de resolución o detalle de la imagen depende del número de células fotoeléctricas del CCD. Este número se expresa en píxeles. A mayor número de píxeles, mayor resolución.

Principalmente, existen dos tipos de sensores, los CCD (Charged Couple Device) y los CMOS (Complementary Metal Oxide Semiconductor).

En el caso del CCD, convierte las cargas de las celdas de la matriz en voltajes y entrega una señal analógica en la salida, que posteriormente será digitalizada por la cámara. Se hace una lectura de cada uno de los valores correspondientes a cada una de las celdas. Entonces, es esta información la que un convertidor analógico-digital traduce en forma de datos.

La estructura interna del sensor es muy simple, como se muestra en la figura 2.1, pero tenemos como inconveniente la necesidad de un chip adicional que se encargue del tratamiento de la información proporcionada por el sensor, lo que

se traduce en un gasto mayor y equipos más grandes.

El ruido que provocan los sensores CCD, es menor al que producen los sensores CMOS, ya que el procesado de la señal se lleva a cabo en un chip externo, el cual puede optimizarse mejor para realizar esta función.

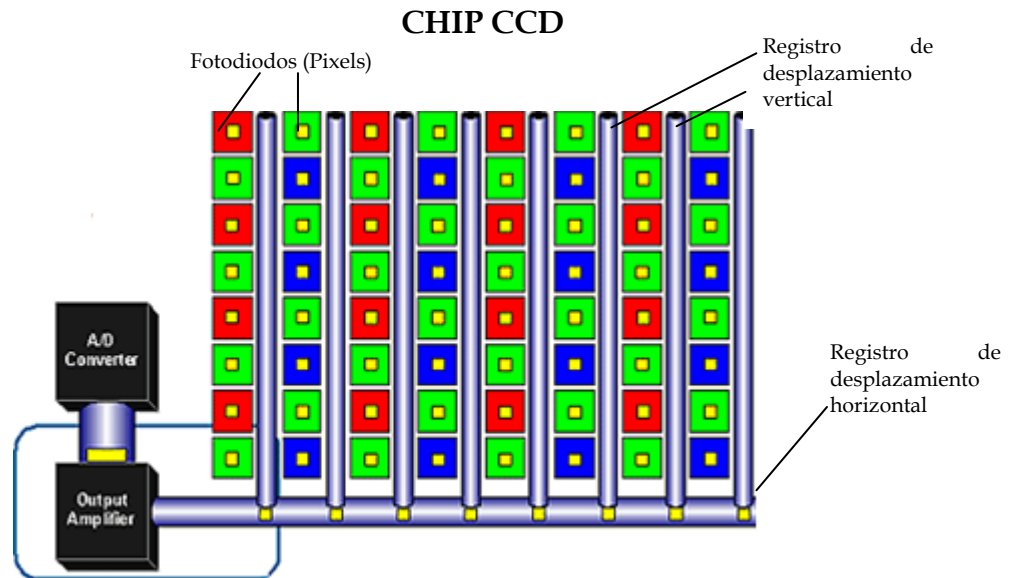


Fig. 2.1. Estructura interna del sensor CCD

En el caso del CMOS, aquí cada celda es independiente. La diferencia principal es que aquí la digitalización de los píxeles se realiza internamente en unos transistores que lleva cada celda, por lo que todo el trabajo se lleva a cabo dentro del sensor y no se hace necesario un chip externo encargado de esta función. Con esto conseguimos reducir costes y equipos más pequeños.

Otra de las grandes ventajas es que los sensores CMOS son más sensibles a la luz, por lo que en condiciones pobres de iluminación se comportan mucho mejor. Esto se debe principalmente a que los amplificadores de señal se encuentran en la propia celda, por lo que hay un menor consumo a igualdad de alimentación.

Los sensores CMOS son más rápidos a la hora de procesar la señal, debido a que todo el procesado se realiza dentro del propio sensor, ofreciendo mayor velocidad. La figura 2.2 muestra la estructura interna del sensor CMOS.

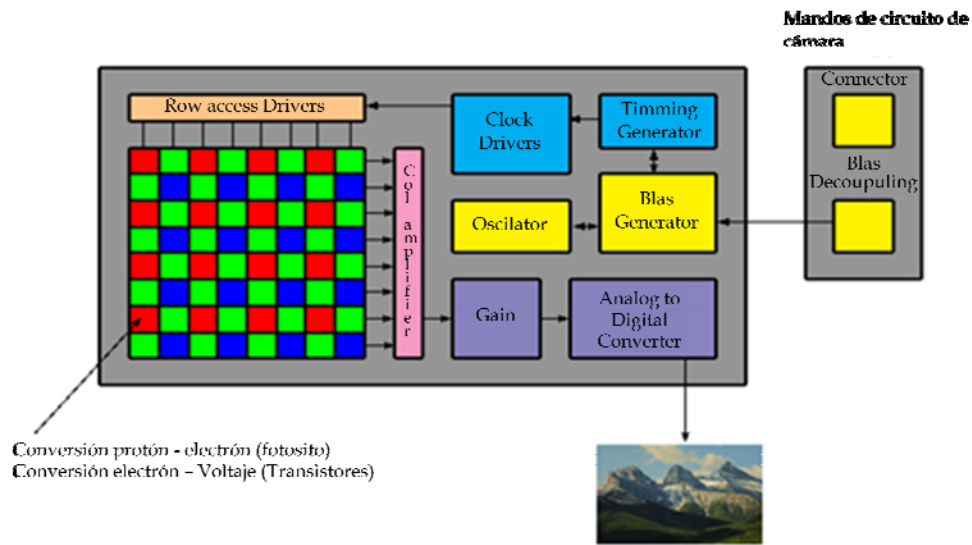


Fig. 2.2. Estructura interna del sensor CMOS

2.2.2. Proceso de Adquisición de Imágenes

La estructura básica y la secuencia para el procesamiento de imágenes en cámaras digitales es muy similar en todos los casos. En una cámara digital, la luz entra a través de la lente y va a los elementos del CCD, cada uno de ellos llamado pixel.

El CCD es el principal y más caro elemento en las cámaras digitales, cada elemento del array del CCD recoge la luz que incide sobre él, del espectro de luz que pasa por el filtro de color, que está delante del sensor y de esta manera se obtiene la señal eléctrica que representa la escena. Cada elemento del CCD es esencialmente monocromático, y debido a esto, para capturar el color de la imagen se requiere separar cada componente del color, normalmente rojo, verde y azul. Sin embargo, para abaratar el precio de los sensores, en la mayoría de las cámaras digitales un solo CCD es organizado para tener diferentes espectros de filtro en cada pixel, de esta manera el CCD es capaz de recoger todo el espectro de luz. La máscara de delante del sensor se llama matriz de filtro del color (CFA). Como el ojo humano es más sensible a la luz verde, los patrones del CFA tienen normalmente más patrones de verde que de rojo y azul.

Como resultado de la utilización del patrón del color CFA, cada pixel tiene solo el color de la imagen asociado a él. Los valores perdidos en el RGB son calculados basándose en los valores de los pixeles vecinos, esta operación se llama demosaicing. La figura 2.3 muestra los patrones comunes del CFA el RGB y el CMYK.

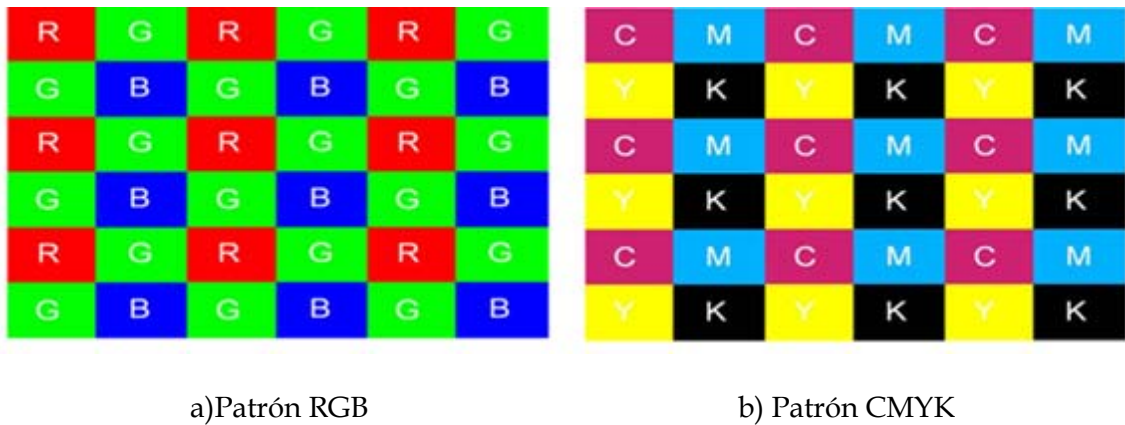


Fig. 2.3. Patrones CFA

Después de demosaicing, la siguiente operación que se realiza es el balanceo de blancos, básicamente es un control de la cámara que sirve para ajustar el brillo de los colores básicos rojo, verde y azul (RGB) con el objeto de que la parte más brillante de la imagen aparezca como color blanco, y la menos brillante como negro.

A continuación, le sigue la etapa de interpolación del color, y corrección gamma. Esta etapa es importante para hacer una redistribución tonal de la información y acercarla lo más posible a la manera en que el ojo humano recibe los brillos. Después de esto, se realiza una reducción del ruido y suavizado de la imagen.

Por último, la imagen es comprimida y se guarda en memoria [4].

2.3. Características de Extracción de una Imagen

Para crear un esquema de identificación fiable, la primera tarea que debemos hacer es encontrar en el código alguna diferencia entre los tipos de dispositivos y los diferentes modelos de este. Luego, estas diferencias pueden ser extraídas y representadas como características únicas de cada dispositivo que pueden ser usadas para la identificación de una imagen desconocida.

2.3.1. Coeficiente de interpolación del color como característica.

La mayoría de las cámaras digitales y las cámaras de móviles utilizan diferentes métodos para la interpolación del color, y estas diferencias pueden ser identificadas forensemente para la identificación de los diferentes tipos de modelos. Para extraer el coeficiente de interpolación del color, empezaremos asumiendo la patente de Bayer para CFA, la cual da la localización de los píxeles que son obtenidos directamente por el sensor y son interpolados.

Nosotros dividimos esta imagen de píxeles en tres diferentes regiones basadas en la inclinación local de valores como (1) región con inclinación horizontal significativa (2) región con inclinación vertical significativa (3) regiones suaves. En cada una de estas tres regiones, aproximamos la interpolación del color para ser línea, y representamos la interpolación de píxeles como una suma ponderada de los píxeles vecinos que son obtenidos directamente por el sensor. El conjunto de ecuaciones obtenidas por esta vía, son resueltas mediante la computación del coeficiente de interpolación.

2.3.2. Ruido estadístico como característica de extracción

El ruido ocurre cuando los fotoelectrones son creados en el dispositivo de imágenes. Un ejemplo de medición del ruido es la señal oscura no uniforme, o la variación entre el voltaje del pixel bajo condiciones de poca luz. La respuesta de la foto no-uniforme puede ser medida con la variación entre el voltaje del pixel con poca luz con intensidad fija. Cuando los dispositivos de imagen aplican el post-proceso para compensar y reducir el ruido de la imagen, algunas estadísticas propiedad del ruido que aún permanecen inalteradas dependen de la naturaleza específica del sensor y filtros usados.

2.4. Estadísticas sobre telefonía móvil

2.4.1. Estadísticas sobre ventas y usuarios

La tabla 2.1 presenta una estadística sobre la distribución del uso de dispositivos móviles por países.

Puesto	País	Disp. Móviles	Habitantes	%	Fecha
1	China	878.830.000	1.341.000.000	64,4%	Feb 2011
2	India	791.381.574	1.210.193.422	66,36%	Feb 2011
3	Estados Unidos	302.947.098	310.866.000	96%	Dic 2010
4	Rusia	213.900.000	142.905.200	147,3%	Jun 2010
5	Brasil	207.500.000	190.732.694	106,91%	Feb 2011
6	Indonesia	168.264.000	237.556.363	73,1%	May 2009
7	Pakistán	101.641.122	171.901.000	60,32%	Nov 2010
8	Japón	107.490.000	127.370.000	84,1%	Mar 2009
9	Alemania	107.000.000	81.882.342	130,1%	2009
10	México	88.797.186	112.322.757	79,8%	Sept 2010

Tabla. 2.1. Distribución de dispositivos móviles por países.

De este estudio realizado se pueden sacar las siguientes conclusiones:

- Para el año 2010, la competencia estaba disponible en más del 90% de los países en los servicios móviles e Internet. Como se refleja en la tabla 2.1.
- En la actualidad hay 157 autoridades reguladoras independientes en todo el mundo - hasta de 106 a principios de la década.
- El número de suscripciones móviles en todo el mundo celular creció por debajo de *mil millones* a más de *5000 millones* a lo largo de la década.
- El número de usuarios de Internet en todo el mundo creció por debajo de *400 millones* a más de *dos mil millones* desde 2000 hasta 2010.
- Los siguientes servicios se han visto el mayor crecimiento en la competencia en los últimos 10 años (ver figura 2.4):
 - Pasarelas internacionales: La competencia ya está disponible en el 78% de los países, frente a sólo el 38% en el año 2000.
 - Servicios básicos de voz: la competencia ya está disponible en dos tercios de los países en todo el mundo, frente a menos del 40% en el año 2000.
 - Líneas arrendadas: la competencia ya está disponible en tres cuartas partes de los países del mundo, frente a menos del 50% en el año 2000.
 - Bucle local inalámbrico: La competencia ya está disponible en el 82% de los países en todo el mundo, por encima del 62% en el año 2000.

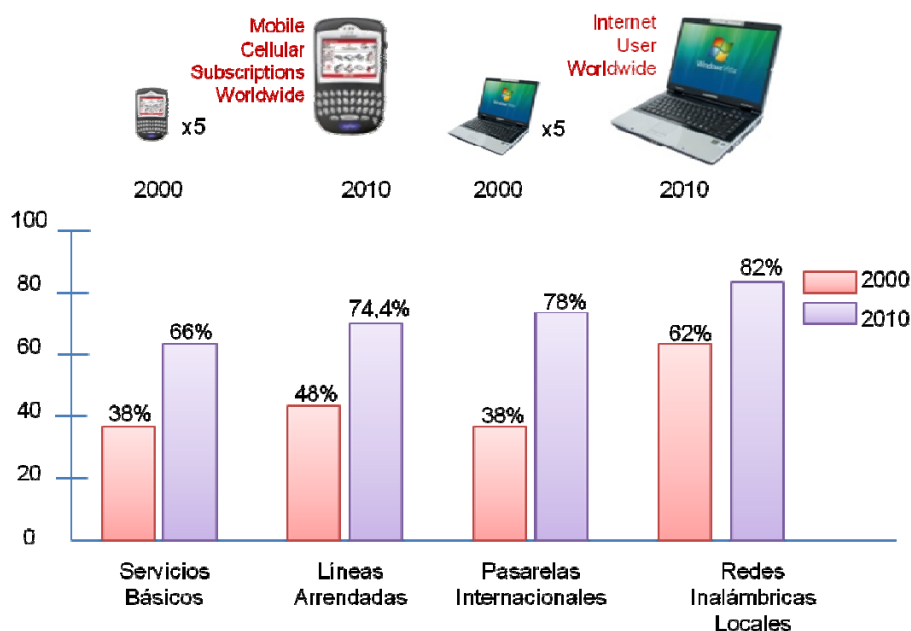


Fig. 2.4. Crecimiento de usuarios de dispositivos móviles 2000-2010. (Fuente: ITU)

Los **crecimientos** más destacados han venido de la mano de HTC (128%) y Apple (87,2%), y las **pérdidas** en unidades vendidas se han asentado en Motorola (-33,55%) y Sony Ericsson (23,90%). El dato más relevante lo tenemos en **Nokia**, que solo ha podido crecer en un 4,6%, como se muestra en la tabla 2.1. Veremos cómo varían los datos en 2011, año en el que veremos los primeros Smartphones Nokia Windows Phone.

Empresa	Unds. 2010	%Mercado 2010	Unds, 2009	% Mercado 2009
Nokia	461.318,2	28,9	440.881,6	36,4
Samsung	281.065,8	17,6	235.772,0	19,5
LG Electronics	114.154,6	7,1	121.972,1	10,1
Research In Motion	47.451,6	3,0	34.346,6	2,8
Apple	46.598,3	2,9	24.889,7	2,1
Sony Ericson	41.819,2	2,6	54.956,6	4,5
Motorola	38.553,7	2,4	58.475,2	4,8
ZTE	28.768,7	1,8	16.026,1	1,3
HTC	24.688,4	1,5	10.811,9	0,9
Huawei	23.814,7	1,5	13.490,6	1,1
Otros	488.569,3	30,6	199.617,2	16,5
Total	1.596.802,4	100,0	1.211.239,6	100,0

Tabla. 2.2. Distribución de dispositivos móviles por marcas (2009-2010). [5]

El incremento de presencia de Apple se ha debido, sobre todo, a la finalización de los contratos de exclusividad con los operadores de telefonía móvil, aparte de la expansión a nuevos mercados.

Es importante fijarse en como acaparan cuota de mercado los dispositivos englobados en la categoría "OTROS", que han afectado directamente a la cuota de mercado global de los principales **jugadores**.

Android ha sido el Sistema Operativo para Smartphone que ha explotado de 2009 a 2010, multiplicando por 10 las ventas al usuario final. **Symbian** ha aumentado en un 37,95% y **BlackBerry** en un 38,15%.

Por otro lado, Windows Phone, que de 2009 a 2010 ha reducido sus ventas en un 17,65%, se presume que en este año verá una recuperación, aunque muy probablemente no logre los números de récord de Android.

Los terminales con Google Android y los Apple iPhone son los únicos que han ganado **market share** de un año a otro. La tabla 2.3 muestra el detalle de esta información.

Empresa	Unidades 2010	% Mercado 2010	Unidades 2009	% Mercado 2009
Symbian	111.576,7	37,6	80.878,3	46,9
Android	67.224,5	22,7		6,798,4
Research in Motion	47.451,6	16,0	34.346,6	19,9
iOS	46.598,3	15,7	24.889,7	14,4
Microsoft	12.378,2	4,2	15.031,0	8,7
Otros Oss	11.417,4	3,8	104.32,1	6,1
Total	296.646,6	100,0	172.376,1	100,0

Tabla. 2.3. Distribución de dispositivos móviles por Sistema Operativo (2009-2010). [5]

2.4.2. Estadísticas sobre uso de dispositivos móviles

Según un estudio practicado por IAB Spain Research [6] mostrado en la figura 2.5, sobre 1265 usuarios de telefonía móvil, arrojaron los siguientes resultados, acerca del uso de las cámaras fotográficas de dispositivos móviles:

- Un 94% de los dispositivos móviles poseían cámara fotográfica
- Frecuencia de uso:
 - El 6% de los usuarios la utilizaban varias veces al día
 - El 5% de los usuarios la utilizaban por lo menos una vez al día
 - El 34% de los usuarios la utilizaban una o varias veces a la semana
 - El 34% de los usuarios la utilizaban una o varias veces al mes
 - El 16% de los usuarios la utilizaban con menos frecuencia que las indicadas

- El 5% de los usuarios no la utilizaban o sólo la habían utilizado una vez.

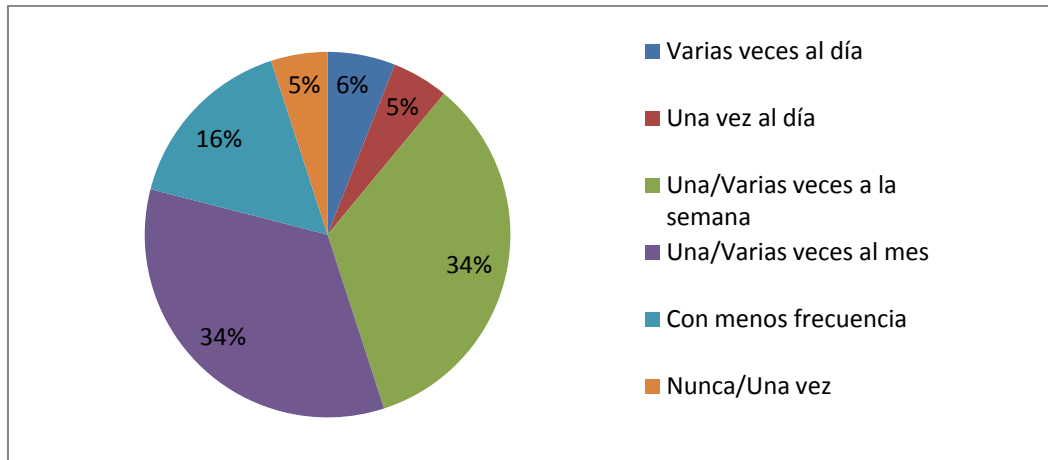


Fig. 2.5. Frecuencia de uso de cámaras en dispositivos móviles

3. FORMATOS DE ARCHIVOS INTERCAMBIABLES

En este capítulo se presentan los diferentes de formatos de archivo, que permiten el almacenamiento de meta información sobre las imágenes. Estos formatos evitan la pérdida de información en el intercambio de ficheros. Seguidamente se describirán los más importantes relacionados con los archivos de imagen.

3.1. Tagged Image File Specification 6.0

TIFF es un formato de archivo basado en etiquetas para el almacenamiento e intercambio de imágenes [7].

La primera versión de la especificación TIFF fue publicada por la Corporación Aldus en otoño 1986, tras una serie de encuentros con una serie de desarrolladores software y fabricantes de escáneres. La última versión data de Junio de 1992.

3.1.1. Ámbito de TIFF

El propósito de TIFF es describir y almacenar datos de la imagen.

Un objetivo de TIFF es proporcionar un entorno rico, con el cual las aplicaciones puedan intercambiar datos de la imagen.

Aunque TIFF es un formato rico en información, puede usarse fácilmente, ya que el número de campos requeridos es pequeño

TIFF puede ser mejorado tanto como las necesidades de las nuevas imagen aumenten. Una prioridad ha sido dada a la construcción de TIFF para que las futuras mejoras puedan ser añadidas sin causar un gran esfuerzo a los desarrolladores.

3.1.2. Características TIFF

Entre las características más relevantes de TIFF se encuentran las siguientes:

- TIFF incluye varios esquemas de compresión, que permiten a los desarrolladores elegir el más apropiado para sus aplicaciones.
- No está unido a dispositivos electrónicos específicos.
- Es portable, no favorece a un sistema operativo particular ni a un sistema de archivos, compiladores o procesadores concreto.
- Está diseñado para ser ampliable para evolucionar según las nuevas necesidades lo requieran.
- Permite la inclusión de una ilimitada cantidad de información privada para propósitos particulares.

3.2. JPEG File Interchange Fomat

JPEG File Interchange Fomat (JFIF) es un formato de archivos de imagen estándar, que contienen las imágenes guardadas en compresión JPEG [8]. Permite el intercambio de metadatos entre una gran variedad de plataformas y aplicaciones. Es un formato simple cuyo único objetivo es el intercambio de imágenes comprimidas JPEG. No incluye algunas de las características avanzadas de otros formatos de archivo de intercambio de imágenes

Formalmente los estándares EXIF y JFIF son incompatibles ya que ambos especifican que sus segmentos de aplicación deben de ir los primeros en el archivo de imagen. En la práctica muchas aplicaciones producen archivos con ambos segmentos, pero esto puede crear problemas en algunas aplicaciones.

3.3. International Press Telecommunications Council

El modelo de intercambio de información (IPTC) es una estructura y un conjunto de atributos de metadatos que pueden ser aplicados tanto a archivos de texto como a otros tipos de archivos multimedia [9].

Fue desarrollado al comienzo de los años 90 por el *International Press Telecommunications Council* para mejorar los intercambios internacionales de información. Los metadatos comúnmente utilizados eran el nombre del autor o fotógrafo e informaciones sobre descripciones del archivo o copyright. Este modelo ha sido utilizado por el mundo de los periódicos, las agencias de prensa y los fotógrafos.

3.4. Exchangeable Image File

EXIF son las siglas de *EXchangeable Image File* (archivo de imagen intercambiable) [10]. La información EXIF es una especificación para formatos de archivos de imagen, un conjunto de metadatos encapsulados en el fichero de imagen que se almacenan cuando se toma la foto. En esta información se recogen, entre otros, los parámetros de la exposición y los ajustes de la cámara en el momento de hacer la fotografía.

El formato *EXIF* fue creado en octubre de 1995 por JEIDA (*Japan Electronic Industry Development Association*). En noviembre de 1997 se lanzó la versión 2.0, la revisión 2.1 apareció en junio de 1998. EXIF versión 2.2 apareció en Abril de 2002. La versión 2.3 es de Abril de 2010.

Algunos de los parámetros que se recogen normalmente en la información EXIF de una fotografía son:

- Marca y modelo de la cámara digital.
- Dimensiones de la imagen.
- Fecha y hora de la exposición.
- Información técnica de la exposición (velocidad de obturación, longitud de enfoque, distancia, balance de blancos, configuración del flash, etc.).
- Una imagen en miniatura para previsualizar la foto.
- Descripción e información de copyright.

La información EXIF no siempre está presente en la imagen, ya que puede

ser borrada mediante determinados programas de edición de imágenes. También existen programas que permiten modificar los valores de algunos de estos parámetros.

3.4.1. Introducción a la especificación del archivo de imagen EXIF

La especificación del archivo de imagen EXIF estipula el método de grabación de datos de imagen en archivos, especificando los siguientes elementos:

- Estructura de archivos de datos de imagen
- Las etiquetas usadas por este estándar

3.4.2. Características EXIF

La forma de guardar los datos en el archivo esta basada en otros formatos existentes. Los archivos comprimidos están guardados como JPEG con la inserción de datos en los segmentos de marcado APP1 Y APP2. Los archivos sin comprimir están grabados en formato TIFF Rev. 6.0 [1]. El uso de formatos existentes hace que pueda ser leído por aplicaciones comerciales, y hace posible el uso de funciones para la vista y manipulación de las imágenes.

La información relativa para ambos, archivos comprimidos y no comprimidos, es almacenada en el formato de etiqueta de información definido en TIFF Rev. 6.0[1]. La información específica de la cámara y no definida en TIFF es guardada en etiquetas para datos privados definidos para EXIF.

También especifica el formato de guardado de las miniaturas. La razón de usar el formato de etiqueta TIFF en el formato de archivo comprimido APP1 es facilitar el intercambio de atributos de datos entre archivos EXIF comprimidos y no comprimidos.

Los archivos comprimidos pueden guardar datos que excedan los 64 Kbyte, dividiéndolos en varios segmentos APP2.

3.4.3. Especificación del archivo de imagen EXIF

Nos vamos a centrar en la especificación de:

- Los datos de la imagen
- La estructura básica de los datos de imagen
- Etiquetas

3.4.3.1. Especificación de los datos de la imagen

- Numero de pixeles: No hay límite del número de pixeles de la imagen principal, este está comprimida en JPEG o no.
- Aspecto del pixel: Los datos de imagen están grabados como pixeles cuadrados con un ratio 1:1, Igualmente aplicado a las miniaturas
- Composición de Pixel y Muestreo: La combinación de composición de pixel y muestreo de pixel debería ser RGB 4:4:4 y, o bien Y:Cb:Cr = 4:2:2 o Y:Cb:Cr = 4:2:0 La composición de los datos de imagen deben de ser de 8 bits cada uno. Igualmente aplicado a las miniaturas.

La figura 3.1 muestra la composición y el muestreo para una imagen de ancho w y altura h .

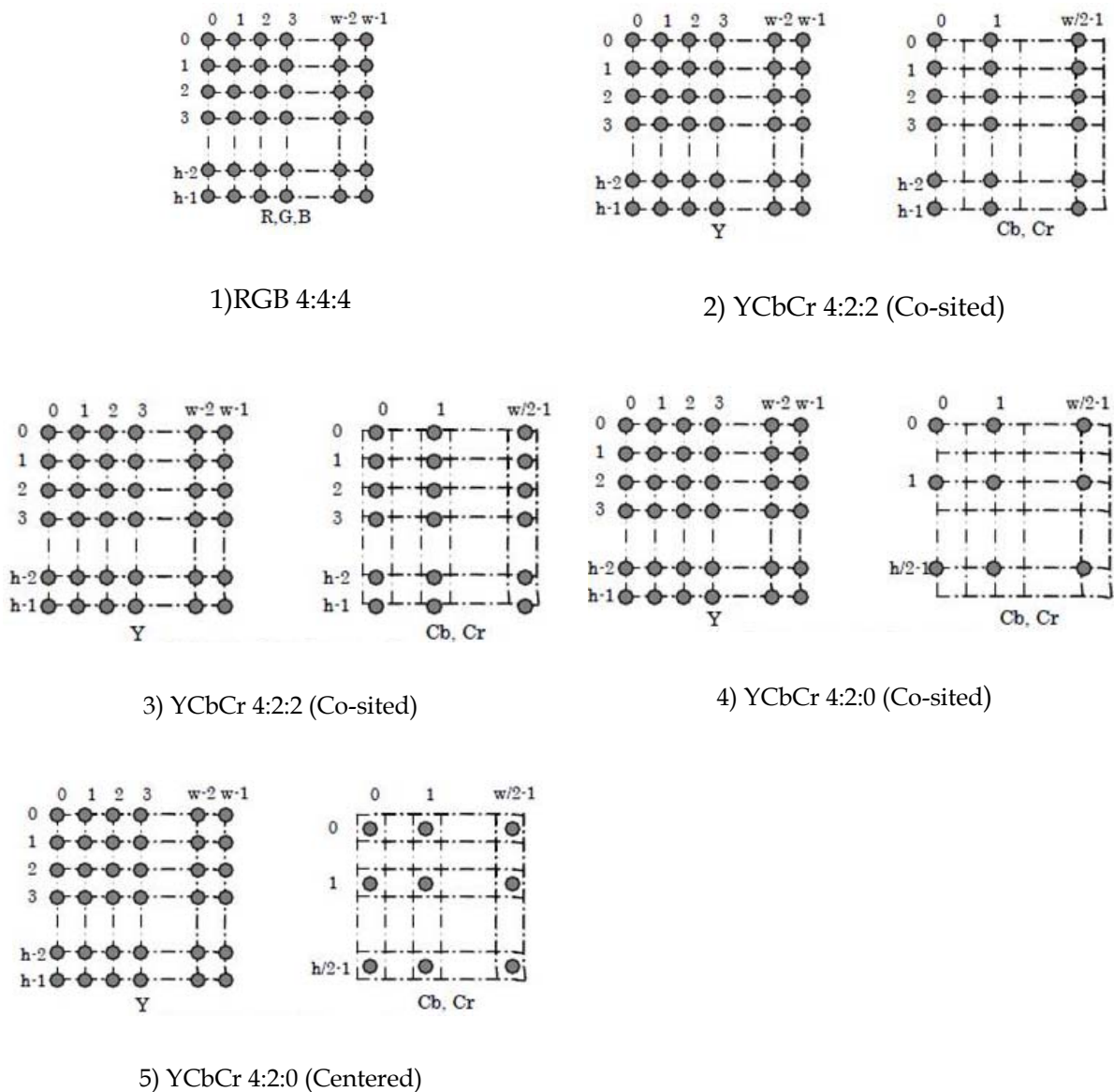


Fig. 3.1. Composición y Muestreo para una Imagen de Ancho W y Altura H

3.4.3.2. Estructura básica de los datos de la imagen

Estructura básica de los datos de la imagen principal: En función del tipo de imagen podemos encontrarnos con los siguientes formatos:

- RGB sin compresión
- YCBr sin comprimir
- Datos comprimidos JPEG

Para los datos comprimidos la información de los atributos se guardara en

APP1. Los datos escritos en APP1 serán compatibles con TIFF. El uso de un método de información común para los atributos para almacenar los datos comprimidos y sin comprimir, hace que el formato de archivo sea más sencillo. A la vez la forma de añadir etiquetas privadas, se aprovecha de la extensibilidad del mecanismo de etiquetas.

Estructura básica de los datos RGB sin comprimir: Los datos sin comprimir RGB se guardan según el estándar TIFF Rev. 6.0 Imágenes a Color [7]. La información de los atributos que se registran en las etiquetas sigue el formato TIFF Rev. 6.0. La información de los atributos específicos de Exif se registra en las etiquetas privadas reservadas en el formato TIFF. Las etiquetas privadas apuntan a estos conjuntos de información de atributos (*EXIF IFD Image File Directory*).

Al igual que en el estándar TIFF la posición de grabado de cada valor IFD no está especificado. Los archivos se estructuran de la siguiente manera (ver figura 3.2):

- Archivo de encabezado
- IFD 0
- Valor 0 de IFD
- IFD 1
- 1^{er} Valor de IFD
- 1^a (Miniatura) Datos de la imagen
- 0 (Principal) Datos de la imagen

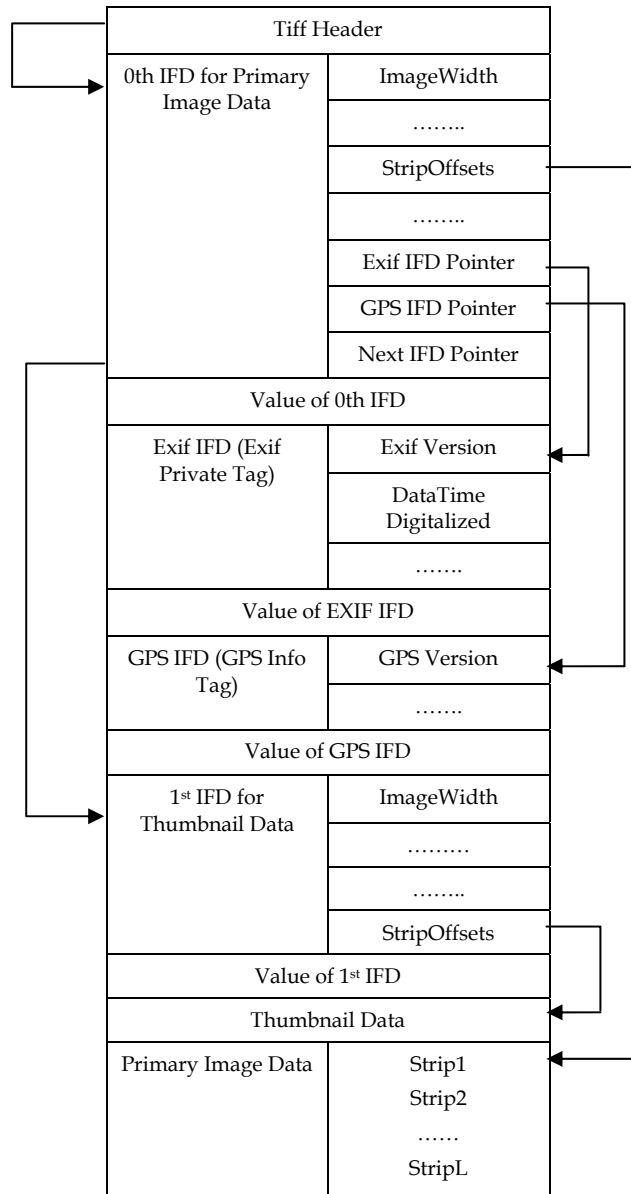


Fig. 3.2. Estructura básica de los archivos de datos sin comprimir

Las cabeceras son de 8 bytes como se define en el estándar TIFF, con el desplazamiento de los IFD indicando el inicio de la IFD 0

El campo de IFD 0, Siguiente puntero IFD, indica la dirección de comienzo del IFD 1 (Imágenes Miniatura) Cuando la primera IFD no está grabada termina con 00000000.H

Estructura básica de los datos sin comprimir YCbCr: Los datos sin comprimir YCbCr se guardan según el estándar TIFF Rev. 6.0 Extensión de las imágenes YCbCr[1].

- Coeficientes de la matriz de transformación de color RGB, YCbCr (*YCbCrCoefficients*).
- Submuestreo de crominancia de la información (*YCbCrSubSampling*).
- Información sobre la coincidencia/no coincidencias de las muestras de crominancia y luminancia (*YCbCrPositioning*).

El método de guardado de datos de los atributos es igual que para los datos sin comprimir RGB

Estructura básica de los datos comprimidos JPEG: Los archivos de datos comprimidos se guardan en conformidad con el formato JPEG *Discrete Cosine Transform* DCT especificados en la norma ISO / IEC 10918-1.

El segmento APP1 se ubica inmediatamente después del marcador SOI (*Start of Image* -Principio de imagen).

Se pueden registrar varios segmentos de aplicación APP2 si es necesario comenzando inmediatamente después del segmento APP1.

Los segmentos de aplicación APPn distintos de APP1 y APP2 no son utilizados por EXIF. Sin embargo los lectores Exif deben de estar diseñados para saltar estos segmentos desconocidos APPn.

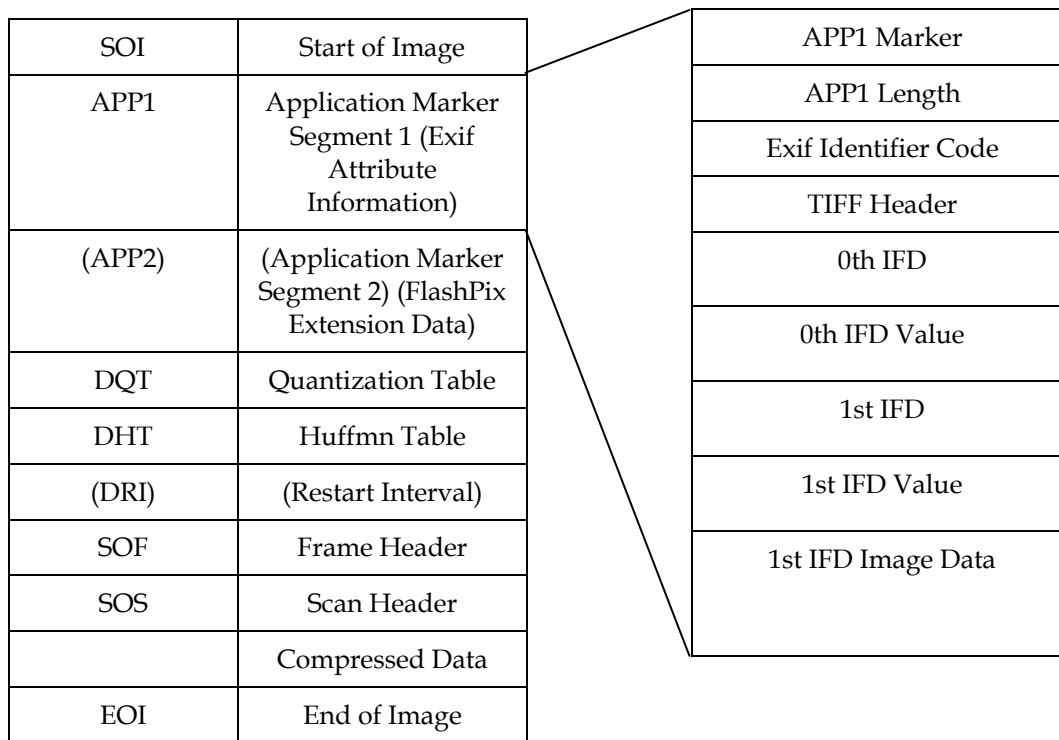


Fig. 3.3. Estructura básica de archivos de datos comprimidos

El segmento de aplicación APP1, consta del Marcador APP1, identificador de código EXIF y la información del atributo. El tamaño del segmento APP1 incluyendo todos estos elementos, no debe exceder los 64 Kbyte como esta especificado en el estándar JPEG.

La información del atributo se almacena en la estructura TIFF incluyendo un archivo de encabezado con un máximo de dos IFD (IFD 0 e IFD 1)

IFD 0 guarda información relativa a la imagen principal.

IFD 1 guarda información relativa a la miniatura.

APP2 consta del Marcador APP2, el código de identificación y la lista de contenidos para la grabación de la extensión FlashPix o para un flujo de datos. Se pueden utilizar varios segmentos APP2 para guardar datos superiores a 64 Kbyte.

3.4.3.3. Etiquetas

Características de la información de atributos: Al igual que los datos RGB e YCbCr que respetan el estándar TIFF, las partes que siguen en la estructura de TIFF se registran conforme a el estándar TIFF.

Además de la información de los atributos indicados como obligatorios en la norma TIFF, esta norma EXIF, añade las etiquetas opcionales como por ejemplo las etiquetas del GPS para registrar la información de posición. También hay especificaciones originales de EXIF, que no se encuentran en el estándar TIFF, para la grabación comprimida de imágenes en miniatura. La grabación de datos comprimidos difiere de los datos sin comprimir en los siguientes aspectos:

- Cuando los datos de la imagen principal se registran en forma comprimida, no hay ninguna etiqueta que indique la imagen principal en sí misma o su dirección (puntero).
- Cuando los datos de la miniatura se registran en forma comprimida, la dirección y el tamaño son designados con las etiquetas específicas EXIF.
- La información relativa a la compresión se puede grabar utilizando las etiquetas para este propósito.

3.4.4. Estructura de IFD

Los IFD utilizados en esta norma consisten en un contador de 2 bytes (número de campos), Arrays de 12 bytes, y 4- bytes desplazamiento al IFD siguiente, en conformidad con TIFF Rev. 6.0.

Cada uno de los *i* campos del array de interoperabilidad de 12 bytes se compone de los siguientes cuatro elementos, respectivamente.

- Bytes 0-1 Etiqueta
- Bytes 2-3 Tipo
- Bytes 4-7 Contador
- Bytes 8-11 Valor de compensación

Cada elemento se explica brevemente a continuación.

Tamaño	Nombre del registro	Descripción
2 bytes	Number of n of Interoperability arrays	
12n bytes	The n arrays	12 bytes cada uno
4 bytes	Link to next IFD	Puede ser 0
.....	Area de datos adicionales

Fig. 3.4. Estructura del Image File Directory IFD [10]

3.4.4.1. Etiqueta

A cada etiqueta se le asigna un número único de 2 bytes para identificar el campo. Los números de etiqueta en el IFD 0 de Exif y en el IFD 1 son los mismos que los números de las etiquetas TIFF.

3.4.4.2. Tipo

Los siguientes tipos se utilizan en Exif:

- 1= BYTE Un entero sin signo de 8 bits.
- 2 = ASCII 8 bits que contienen un código ASCII de 7 bits. El byte final se termina con NULL.
- 3 = SHORT Un entero de 16 bits (2 bytes) sin signo
- 4 = LONG Entero de 32 bits (4 bytes) sin signo
- 5 = RATIONAL Dos enteros largos. El primer entero largo es el numerador y el segundo de duración, expresa el denominador
- 7 = UNDEFINED 8 bits que pueden tomar cualquier valor en función de la definición del campo
- 9 = SLONG Un entero de 32 bits (4 bytes) con signo (complemento a 2 notación)
- 10 = SRATIONAL Dos SLONGs. El SLONG primero es el numerador y es el segundo SLONG denominador.

3.4.4.3. Contador

- El número de valores.
- Debe tenerse en cuenta cuidadosamente que la cuenta no es la suma de los bytes.
- En el caso de un valor de corto (16 bits), por ejemplo, la cuenta es "1" a pesar de que es de 2 bytes.

3.4.4.4. Valor de desplazamiento

Esta etiqueta registra el desplazamiento desde el inicio de la cabecera TIFF a la posición donde se registra el valor en sí. En los casos en que el valor se ajusta en 4 bytes, el valor en sí se registra. Si el valor es menor de 4 bytes, el valor es almacenado en el área de 4 bytes a partir de la izquierda.

Tamaño	Nombre del registro	Descripción
2 bytes	Tag	Un número único de 2 bytes
2 bytes	Type	Uno de los 12 tipos
4 bytes	Count	Número de valores
4 bytes	Value Offset	Valor del desplazamiento

Fig. 3.5. Estructura del campo i-ésimo del array de interoperabilidad [10]

3.4.5. Exif específica IFD (*Image File Directory*)

3.4.5.1. Exif IFD

EXIF IFD es un conjunto de etiquetas para el guardado de información de atributos específicos de EXIF. Es apuntado por el desplazamiento desde la cabecera TIFF (Valor desplazamiento), indicado por un valor de etiqueta privada EXIF.

EXIF Puntero IFD:

- Tag = 34665 (8769.H)
- Type =LONG
- Count = 1
- Default =none
- Un puntero al EXIF IFD

3.4.5.2. GPS IFD

GPS IFD es un conjunto de etiquetas para el guardado de los datos GPS. Este conjunto es señalado por la distancia a la cabecera TIFF (Valor Desplazamiento) indicado por un valor de etiqueta privada GPS.

Puntero IFD GPS Info:

- Tag =34853(8825.H)
- Type =LONG
- Count =1
- Default =none
- Un puntero al IFD de información GPS

3.4.5.3. Interoperabilidad IFD

Interoperabilidad IFD está compuesto por etiquetas que almacenan la información para garantizar la interoperabilidad y son señaladas por la etiqueta siguiente situada en el IFD EXIF.

Puntero Interoperabilidad IFD :

- Tag =40965(A005.H)
- Type =LONG
- Count =1
- Default =none

4. GESTIÓN DEL PROYECTO

En este capítulo se presentan los elementos relacionados con el proyecto a un alto nivel. Entre estos aspectos se debe resaltar la planificación y el seguimiento de la realización del mismo.

También se comentan aspectos relacionados con el ciclo de vida del software y la planificación temporal en el desarrollo del proyecto. De igual forma, se realiza una clasificación de los medios y recursos que han sido necesarios en el transcurso del proyecto.

4.1. Planificación

El proyecto se ha desarrollado en 5 fases: Definición del Proyecto, Ejecución, Control, Documentación y Finalización del Proyecto. Esta planificación se ha realizado mediante diagramas de Gantt como muestra la figura 4.1.

1. **Fase de Definición del Proyecto:** [04/10/2010 - 22/11/2010] El objetivo de esta fase es la puesta en marcha del proyecto.
 - a. *Reuniones con el tutor:* [04/10/2010 - 13/10/2010] En ésta etapa se realizaron varias reuniones con el tutor del proyecto. Con el objetivo de exponernos el propósito en el cual se enmarca nuestro proyecto fin de carrera. Y establecernos las pautas de nuestro trabajo, horarios de reuniones y seguimiento del mismo.
 - b. *Estudio del estado del arte:* [14/10/2010 - 05/11/2010] En esta etapa se hace un estudio del estado del arte a nivel general del proyecto. Y se realiza una investigación bibliográfica enfocada a nuestro objetivo en este proyecto.
 - c. *Definición del proyecto:* [08/11/2010 - 22/11/2010] En esta etapa se barajaron varias ideas para el desarrollo de nuestro trabajo, hasta concretar en una sola y posteriormente exponerla a nuestro tutor para definir y delimitar el proyecto.

2. **Fase de Ejecución del Proyecto:** [23/11/2010 - 31/05/2011] El objetivo de la fase es el desarrollo del proyecto.
 - a. *Análisis y especificación de requisitos:* [23/11/2010 - 21/12/2010] En esta etapa, y con ayuda de toda la información obtenida tanto en las reuniones con el tutor, como en el estudio del estado del arte, obtenemos los requisitos de nuestra aplicación. Una vez realizado este análisis, se indica el comportamiento que tendrá la aplicación una vez desarrollada.
 - b. *Diseño:* [22/12/2010 - 25/01/2011] Es en esta etapa cuando se realiza el diseño de la aplicación. Se realiza el modelo de la base de datos, el esquema de la interfaz y funcionalidades de la aplicación. Este diseño es expuesto y aprobado por nuestro tutor, y para ello se debe ir realizando las modificaciones oportunas hasta ajustarlo a lo deseado.
 - c. *Implementación:* [26/01/2010 - 17/05/2011] En esta etapa se traduce en código el diseño realizado en la etapa anterior. Para esto se realiza un análisis de las diferentes herramientas existentes para elegir el lenguaje de programación que vamos a utilizar, el gestor de la base de datos y el IDE sobre el que vamos a trabajar. En definitiva, todos los elementos necesarios para la codificación de la aplicación.
 - d. *Pruebas:* [18/05/2010 - 31/05/2011] Esta es la última etapa de fase de ejecución. Aquí se comprueba la funcionalidad de la aplicación y si cumple con todo lo recogido en la etapa de implementación. Además, será de necesaria la ayuda de varios compañeros para que prueben el software generado.
3. **Fase de control:** [23/11/2010 - 31/05/2011] Esta fase se realiza en paralelo con la Fase de Ejecución, ya que aquí se realiza un control sobre todo el avance de la realización del proyecto y se realizan los ajustes necesarios que se requieran. Esta fase será muy supervisada por el tutor y los autores para llevar un control y seguimiento del Proyecto.
4. **Fase de documentación:** [23/11/2010 - 31/05/2011] En esta fase se genera

toda la documentación correspondiente de cada uno de los módulos de la aplicación. Dependiendo de la tarea que se esté desarrollando en la Fase de ejecución, se debe generar la documentación y los diagramas o gráficos determinados. Esta fase también se realiza en paralelo con la fase de Ejecución.

5. **Fase de Finalización del Proyecto:** [01/06/2010 - 10/06/2011] En esta fase se realiza la creación de los paquetes de instalación de la aplicación y la redacción de la memoria.

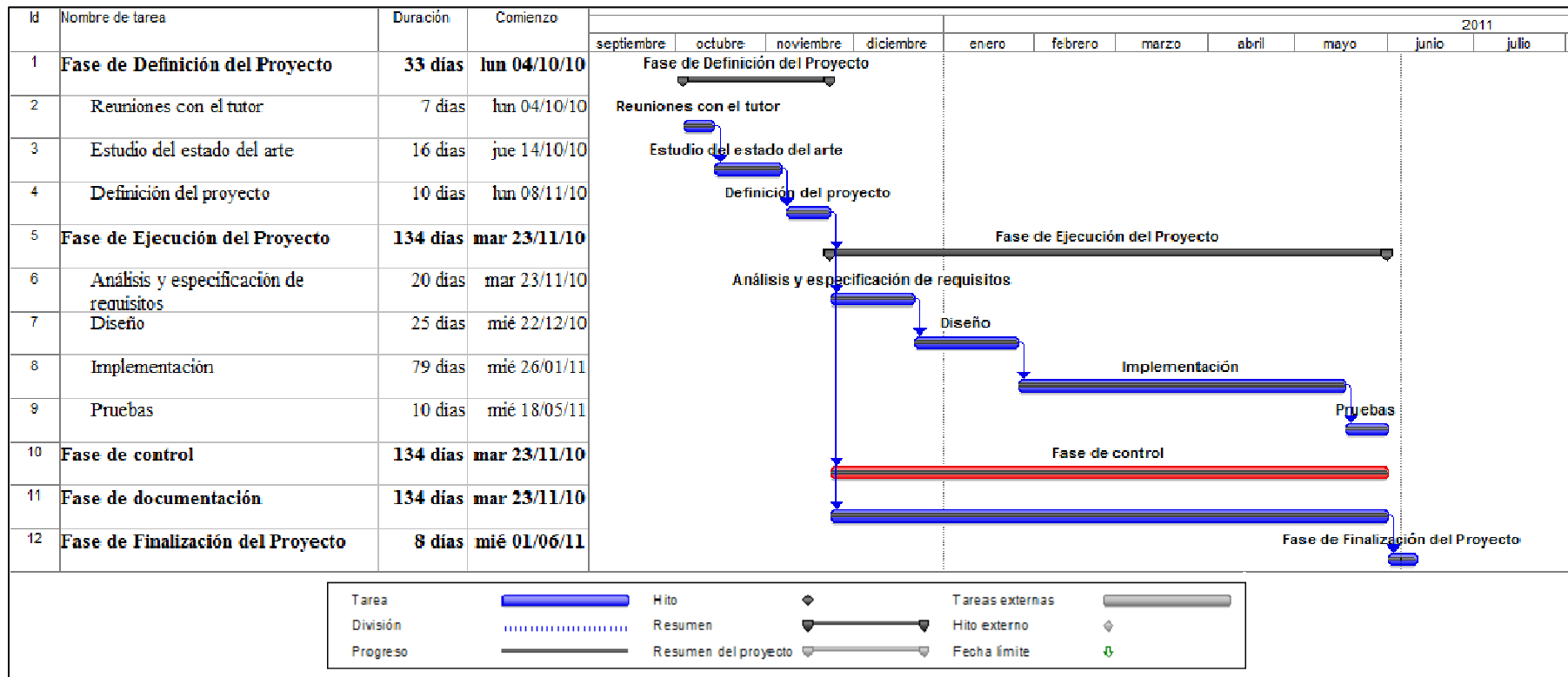


Fig. 4.1. Diagrama de Gantt.

4.2. Herramientas Utilizadas

A continuación, se detallan las herramientas software que se han utilizado para el desarrollo del proyecto:

- Para el desarrollo del código se seleccionó el lenguaje de programación Python.
- Para desarrollar la interface gráfica se utilizó Glade.
- Para la base de datos se utilizó MySQL.
- Para compartir código y trabajar con diferentes versiones se utilizó el repositorio de subversión Assembla.

4.2.1. Python v. 2.6

Python es un lenguaje de programación de alto nivel, creado a finales de los años ochenta por Guido Van Rossum [11]. Es un lenguaje, que permite una sintaxis muy limpia y que el código sea fácilmente legible.

Posee una licencia de código abierto, denominada *Python Software Foundation License*, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

Para ayudar a la hora de desarrollar la aplicación y, sobre todo en el momento de depurar errores, se utilizó el *plugin PyDev* que tiene Eclipse para Python.

Eclipse es un entorno de desarrollo integrado de código abierto. Inicialmente estaba pensado para programar en JAVA, pero gracias a su gran popularidad se han desarrollado muchos *plugin* para soportar diferentes lenguajes de programación.

4.2.2. Glade v. 3.0

Es una herramienta de desarrollo visual de interfaces gráficas mediante GTK/GNOME [12]. Es independiente del lenguaje de programación y predeterminadamente no genera código fuente sino un archivo XML, que posteriormente es leído por Python.

4.2.3. MySql v. 5.0

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario, de *Oracle corporation* [13].

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, aunque las empresas deben comprar una licencia específica que les permita el uso.

4.2.4. Assembla Workspace

Assembla [14] es un repositorio web que permite trabajar con diferentes versiones de archivos. Es muy cómodo ya que todos los usuarios pueden tener acceso a la última versión del documento, así como a las versiones anteriores, por si es necesario deshacer algún cambio.

4.3. Análisis de Aplicaciones Existentes

4.3.1. PhotoInfoEx

Aplicación de pago que permite ver y editar información de metadatos de una imagen [15].

Características:

- Soporta el estándar de las etiquetas Exif 2.21, de las etiquetas GPS e información adicional en diferentes tipos de imágenes.
- Posibilidad de añadir información del estándar EXIF o modificarla en imágenes TIFF o JPEG

- Posibilidad de ver información Exif en imágenes RAW.
- Impresión de fotos y metadatos.
- Restauración y back-up de datos Exif.

4.3.2. Jhead

La información que Jhead puede extraer de Exif es la que se puede extraer de las etiquetas definidas por el estándar [16].

Cosas que Jhead puede modificar de una imagen JPEG EXIF:

- Cambiar fechas de las fotos
- Permite modificar las cabeceras EXIF.
- Intercambiar cabeceras entre imágenes
- Modificar miniaturas dentro de las cabeceras EXIF
- Borrar o editar comentarios JPEG pero no comentarios EXIF
- Crear una cabecera mínima que contenga fecha y miniatura

4.3.3. Exiftool

Exiftool es una aplicación para leer, editar o escribir meta información de archivos [17]. Soporta formatos de metadatos tales como EXIF, GPS, IPTC, XMP, JFIF, GeoTIFF, ICC Profile, Photoshop IRB, FlashPix, AFCP and ID3, como a su vez información de marcado de muchas de las principales marcas de cámaras de fotos.

4.3.4. Exif viewer (Complemento Firefox)

Extrae y muestra metadatos Exif además de otros metadatos, en imágenes JPEG tanto locales como remotas [18].

	Plataforma	Interfaz	Versión EXIF	Visualización de los Datos EXIF	Edición Información EXIF	Observaciones
ImgTool	Linux/Windows	Interfaz grafica	2.3	Organizada por IFD	No	Interfaz intuitiva y manejable, Software Libre
PhotoInfoEx	Windows	Interfaz grafica	2.21	Organizada por IFD	Si	Con licencia de pago.
Jhead	Windows Os-X Linux	Comandos	No Especifica	No organizada	Cambios de cabecera entre imágenes JPEG	<ul style="list-style-type: none"> • Difícil uso. • Software Libre
ExifTool	Windows, Mac OS X and Unix systems	Comandos	No Especifica	No organizada	Si	Difícil uso. Software Libre
Exif viewer	Windows, Mac OS X and Unix systems	Interfaz Grafica	No Especifica	No organizada	No	Complemento de navegador Firefox 1.5 - 6.*

Tabla. 4.1. Tabla comparativa entre aplicaciones existentes

5. VISIÓN GENERAL DEL SISTEMA

En este capítulo se hace una descripción general de la herramienta propuesta en este trabajo haciendo énfasis en el diseño de la misma.

5.1. Descripción de la Aplicación

El análisis forense de imágenes fotográficas es una disciplina que se suele emplear frecuentemente en procesos judiciales. En determinados eventos criminales, una cámara fotográfica y sus fotografías pueden ser evidencias que incriminen o eximan a un acusado en la comisión de un delito, y por tanto, es importante analizar las imágenes que contenga el dispositivo para poder obtener de ellas evidencias suficientes que sustenten una acusación o una defensa ante un juez.

La información básica de una imagen suele ser la fecha de toma de la imagen, el fabricante de la cámara y el modelo de cámara utilizado. Esta información se puede deducir mediante la invocación de las propiedades de una imagen. Cualquier sistema operativo lo permite (en Windows, por ejemplo, haciendo botón derecho → propiedades sobre una imagen).

La aplicación que se ha desarrollado tiene como objetivo realizar análisis forense de imágenes fotográficas tomadas desde dispositivos móviles, y tomar decisiones a partir del análisis de la información almacenada. El sistema está compuesto de dos procesos automatizables para evitar tener que extraer los datos imagen a imagen, y además, permitir extraer mucha más información. De igual forma muestra toda la información contenida en el estándar EXIF grabada en cada fotografía existente en un directorio específico.

Con el uso de esta Aplicación se puede hacer un análisis de metadatos que puede proporcionarnos información para deducir, por ejemplo, si dos imágenes han sido tomadas con la misma cámara o no. Esto es importante, ya que en

ocasiones, la fecha, el fabricante y el modelo no bastan para demostrar la culpabilidad o inocencia de un acusado.

5.2. Modelo de Datos

La base de datos se ha diseñado utilizando MySQL como gestor de base de datos. En ella se almacenará toda la información necesaria para el funcionamiento de la aplicación.

En la figura 5.1 se presenta el modelo entidad - relación resultante del diseño de la base de datos.

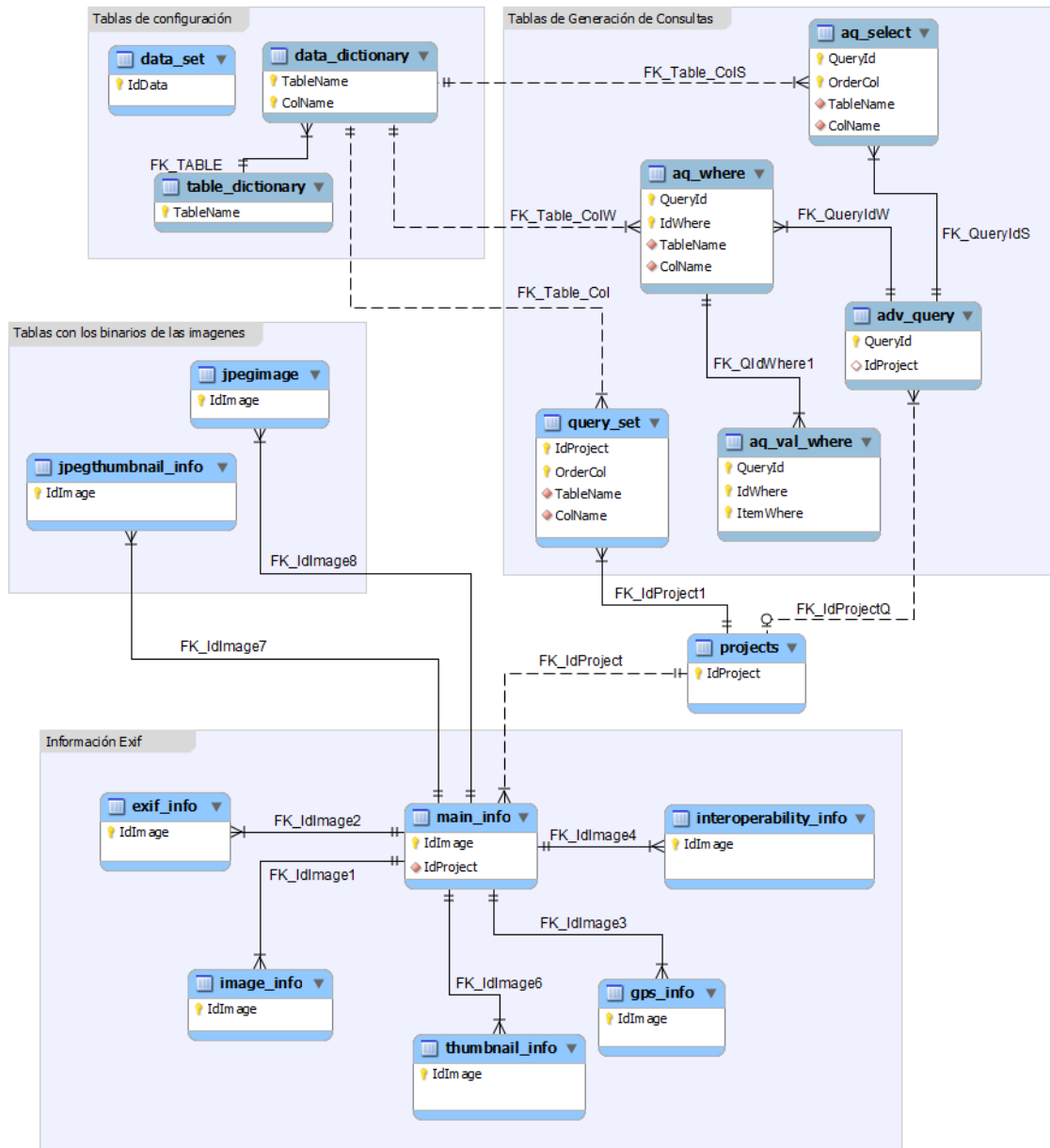


Fig. 5.1. Diagrama de entidad - relación.

5.3. Especificación de Tablas

Las tablas que conforman la base de datos se han organizado en 4 grupos: Tablas de configuración, Tablas de generación de Consultas, Tablas de binarios de las fotografías y tablas de información Exif.

5.3.1. Tablas de Configuración

Este grupo de tablas se utiliza para permitir el correcto funcionamiento de la

aplicación.

5.3.1.1. Data_Set

En esta tabla se almacena los parámetros de configuración de la aplicación.

	Columna	Tipo de dato	Tamaño
PK	IdData	Varchar	10
	DescData	Varchar	100
	CalData	Varchar	50
	DateCreate	Date	8

5.3.1.2. Data_Dictionary

En esta tabla se almacenan cada una de las columnas de las tablas que conforman la base de datos.

	Columna	Tipo de dato	Tamaño
PK	TableName	varchar	50
PK	ColName	varchar	50
	DataType	varchar	50
	Length	integer	5
	dataNull	varchar	1
	PK	varchar	1
	FK	varchar	1
	ColDesc	varchar	200
	ColLargeDesc	varchar	1000
	query_set_col	Char	1
	ColTitle	varchar	50

5.3.1.3. Table_Dictionary

En esta tabla se almacena las tablas que conforman la base de datos.

	Columna	Tipo de dato	Tamaño
PK	TableName	varchar	50
	TabDesc	varchar	200
	TabLargeDesc	varchar	1000
	query_set_tab	char	1

5.3.2. Tablas de Generación de Consultas

Este grupo de tablas se utiliza para realizar la configuración de las consultas que se pueden realizar sobre las fotografías almacenadas en la base de datos.

5.3.2.1. Query_Set

En esta tabla se almacena las columnas en las que se agruparán las fotografías para mostrar una información resumen.

	Columna	Tipo de dato	Tamaño
PK	IdProject	integer	11
PK	OrderCol	integer	2
FK	TableName	varchar	50
FK	ColName	varchar	50

5.3.2.2. Adv_Query

En esta tabla se almacena la información general de las consultas avanzadas que se pueden realizar sobre las fotografías almacenadas en la base de datos.

	Columna	Tipo de dato	Tamaño
PK	QueryId	integer	11
	QueryName	varchar	200
FK	IdProject	integer	11
	DateCreate	date	

5.3.2.3. AQ_Select

En esta tabla se almacena las columnas que van a ser parte del SELECT de las consultas avanzadas.

	Columna	Tipo de dato	Tamaño
PK	QueryId	integer	11
PK	OrderCol	integer	10
FK	TableName	varchar	50
FK	ColName	varchar	50

5.3.2.4. AQ_Where

En esta tabla se almacena las columnas que van a ser parte del WHERE de las consultas avanzadas.

	Columna	Tipo de dato	Tamaño
PK	QueryId	integer	11
PK	IdWhere	integer	10
FK	TableName	varchar	50
FK	ColName	varchar	50

5.3.2.5. AQ_Val_Where

En esta tabla se almacenan los valores de las columnas que van a ser parte del WHERE de las consultas avanzadas.

	Columna	Tipo de dato	Tamaño
PK	QueryId	integer	11
PK	IdWhere	integer	10
PK	ItemWhere	integer	10
	valCol	varchar	500

5.3.3. Tablas de Información EXIF

Este grupo de tablas se utiliza para almacenar toda la información EXIF existente en la fotografía.

5.3.3.1. Main_Info

En esta tabla se almacena información general sobre la fotografía, incluyendo información general del fichero como tal, y un resumen de la información EXIF presente en la misma.

	Columna	Tipo de dato	Tamaño
PK	IdImage	integer	11
	Filename	varchar	100
	PathOrigen	varchar	250
FK	IdProject	integer	11
	typeFile	varchar	5
	dateCreate	timestamp	
	exif_info	varchar	1
	image_info	varchar	1
	gps_info	varchar	1
	interop_info	varchar	1
	thumb_info	varchar	1
	maker_note	varchar	1

5.3.3.2. Image_Info

En esta tabla se almacena información de los datos del IFD de imagen.

	Columna	Tipo de dato	Tamaño
PK	IdImage	integer	11
	ImageWidth	integer	11
	ImageLength	integer	11
	BitsPerSample	integer	11
	Compression	integer	11
	PhotometricInterpretation	integer	11

	Orientation	integer	11
	SamplesPerPixel	integer	11
	PlanarConfiguration	integer	11
	YCbCrSubSampling	integer	11
	YCbCrPositioning	integer	11
	XResolution	varchar	150
	YResolution	varchar	150
	ResolutionUnit	integer	11
	StripOffsets	integer	11
	RowsPerStrip	integer	11
	StripByteCounts	integer	11
	JPEGInterchangeFormat	integer	11
	JPEGInterchangeFormatLength	integer	11
	TransferFunction	integer	11
	WhitePoint	varchar	150
	PrimaryChromaticities	varchar	150
	YCbCrCoefficients	varchar	150
	ReferenceBlackWhite	varchar	150
	DateTime	datetime	
	ImageDescription	varchar	150
	Make	varchar	150
	Model	varchar	150
	Software	varchar	150
	Artist	varchar	150
	Copyright	varchar	150

5.3.3.3. Thumbnail_Info

En esta tabla se almacena información de los datos del IFD de thumbnail.

	Columna	Tipo de dato	Tamaño
PK	IdImage	integer	11
	ImageWidth	integer	11
	ImageLength	integer	11
	BitsPerSample	integer	11
	Compression	integer	11
	PhotometricInterpretation	integer	11
	Orientation	integer	11
	SamplesPerPixel	integer	11
	PlanarConfiguration	integer	11
	YCbCrSubSampling	integer	11
	YCbCrPositioning	integer	11
	XResolution	varchar	150
	YResolution	varchar	150
	ResolutionUnit	integer	11
	StripOffsets	integer	11
	RowsPerStrip	integer	11
	StripByteCounts	integer	11
	JPEGInterchangeFormat	integer	11
	JPEGInterchangeFormatLength	integer	11
	TransferFunction	integer	11
	WhitePoint	varchar	150
	PrimaryChromaticities	varchar	150
	YCbCrCoefficients	varchar	150
	ReferenceBlackWhite	varchar	150
	DateTime	datetime	
	ImageDescription	varchar	150
	Make	varchar	150
	Model	varchar	150
	Software	varchar	150
	Artist	varchar	150
	Copyright	varchar	150

5.3.3.4. Exif_Info

En esta tabla se almacena información de los datos del IFD de EXIF.

	Columna	Tipo de dato	Tamaño
PK	IdImage	integer	11
	ExifVersion	varchar	150
	FlashpixVersion	varchar	150
	ColorSpace	integer	11
	Gamma	varchar	150
	ComponentsConfiguration	varchar	150
	CompressedBitsPerPixel	varchar	150
	PixelXDimension	integer	11
	PixelYDimension	integer	11
	MakerNote	longblob	
	UserComment	varchar	150
	RelatedSoundFile	varchar	150
	DateTimeOriginal	datetime	
	DateTimeDigitized	datetime	
	SubSecTime	varchar	150
	SubSecTimeOriginal	varchar	150
	SubSecTimeDigitized	varchar	150
	ExposureTime	varchar	150
	FNumber	varchar	150
	ExposureProgram	integer	11
	SpectralSensitivity	varchar	150
	PhotographicSensitivity	integer	11
	OECF	varchar	150
	SensitivityType	integer	11
	StandardOutputSensitivity	integer	11
	RecommendedExposureIndex	integer	11
	ISOspeed	integer	11
	ISOspeedLatitudeyyy	integer	11
	ISOspeedLatitudezzz	integer	11
	ShutterSpeedValue	varchar	150

	ApertureValue	varchar	150
	BrightnessValue	varchar	150
	ExposureBiasValue	varchar	150
	MaxApertureValue	varchar	150
	SubjectDistance	varchar	150
	MeteringMode	integer	11
	LightSource	integer	11
	Flash	integer	11
	FocalLength	varchar	150
	SubjectArea	integer	11
	FlashEnergy	varchar	150
	SpatialFrequencyResponse	varchar	150
	FocalPlaneXResolution	varchar	150
	FocalPlaneYResolution	varchar	150
	FocalPlaneResolutionUnit	integer	11
	SubjectLocation	integer	11
	ExposureIndex	varchar	150
	SensingMethod	integer	11
1	FileSource	varchar	150
	SceneType	varchar	150
	CFAPattern	varchar	150
	CustomRendered	integer	11
	ExposureMode	integer	11
	WhiteBalance	integer	11
	DigitalZoomRatio	varchar	150
	FocalLengthIn35mmFilm	integer	11
	SceneCaptureType	integer	11
	GainControl	varchar	150
	Contrast	integer	11
	Saturation	integer	11
	Sharpness	integer	11
	DeviceSettingDescription	varchar	150
	SubjectDistanceRange	integer	11
	ImageUniqueID	varchar	150

	CameraOwnerName	varchar	150
	BodySerialNumber	varchar	150
	LensSpecification	varchar	150
	LensMake	varchar	150
	LensModel	varchar	150
	LensSerialNumber	varchar	150
	Padding	varchar	150
	BatteryLevel	varchar	150
	CFARRepeatPatternDim	varchar	150
	CVAPattern	varchar	150
	DocumentName	varchar	150
	FillOrder	varchar	150
	InterColorProfile	varchar	150
	IPTC/NAA	varchar	150
	JPEGProc	varchar	150
	PageName	varchar	150
	PrintIM	varchar	150
	Rating	varchar	150
	Thresholding	varchar	150
	TransferRange	varchar	150
	XPAuthor	varchar	150
	XPCComment	varchar	150
	XPKeywords	varchar	150
	XPSubject	varchar	150
	XPTitle	varchar	150
	OffsetSchema	varchar	150

5.3.3.5. Interoperability_Info

En esta tabla se almacena información de los datos del IFD de Interoperabilidad.

	Columna	Tipo de dato	Tamaño
PK	IdImage	integer	11
	InteroperabilityIndex	varchar	150
	InteroperabilityVersion	varchar	150
	RelatedImageFileFormat	varchar	150
	RelatedImageLength	varchar	150
	RelatedImageWidth	varchar	150

6. IMPLEMENTACIÓN

El objetivo de este capítulo es plasmar el diseño realizado en código abarcando todas las funcionalidades identificadas, generando el programa ejecutable. En etapas anteriores se ha establecido el diseño enfocado a Python como lenguaje de programación, registrando así las bases que definirán la implementación de la aplicación.

Se hace énfasis en obtener una aplicación implementada con calidad y que sea multiplataforma, que obtenga resultados fiables, y que funcione eficientemente facilitando su mantenimiento en el futuro.

6.1. Aspectos Relevantes de la Implementación

Como aspectos relevantes de la implementación cabe señalar las diversas herramientas utilizadas para llevar a cabo la codificación de los diferentes ficheros que forman la aplicación:

- Se ha utilizado como plataforma de programación ECLIPSE, ya que permite crear entornos integrados de desarrollo multilenguaje y adaptables. Adicionalmente, ofrece una extensa flexibilidad de configuración con los *plugins* necesarios para adaptar los requerimientos de desarrollo que no estén contemplados dentro de las configuraciones básicas.
- El lenguaje de programación utilizado para la codificación de la aplicación es Python, como se ha comentado en capítulos anteriores, por lo que se requirió el uso del *plugin Pydev* de Eclipse para permitir el desarrollo en Python. Este *plugin* viene con muchas características que facilitarán la codificación, como por ejemplo la estructura de código, el resaltado de sintaxis, analizador de sintaxis, o el seguimiento paso a paso entre otras.
- Para el diseño de cada unos de los formularios de las ventanas de la

aplicación se ha utilizado Galde 3. Esta herramienta de desarrollo visual de interfaces gráficas mediante GTK/GNOME permite generar ficheros con extensión .glade con la interfaz gráfica diseñada.

- Para traducir la interfaz gráfica generada en formato .glade a código Python se ha utilizado el paquete *Tepache.py*, que es un realizador de bocetos de código de Python que usa la librería *pygtk*.
- Para organizar todas las tareas relacionadas con la gestión del proyecto se ha utilizado el servicio *ASSEMBLA*, que es un portal que permite gestionar un proyecto y hacer un seguimiento de lo realizado por los miembros del equipo. Además provee de un servidor privado de SVN para subir todo el código fuente utilizando el *plugin Subversive* para integrar Subversion con Eclipse.
- Se han utilizado la librería *PIL (Python Imaging Library)* que agrega capacidades de procesamiento de imágenes al intérprete Python.
- Para realizar la documentación del proyecto se ha utilizado un generador de documentación para código fuente llamado *Doxygen*.

6.2. Funcionalidad de la Aplicación

La aplicación está formada por un conjunto de pantallas que permiten manipular la información contenida en la base de datos, de forma que la interfaz sea de uso intuitivo y amigable.

En la pantalla principal de la aplicación se encuentra el menú con las opciones de la aplicación. Por regla general el menú se encuentra en la parte superior de la pantalla. Desde aquí se puede tener acceso a dos grandes procesos *Exif Information*, *DDBB Projects*.

- **Exif Information**: es la pestaña en donde el usuario puede seleccionar una carpeta para visualizar la información *exif* que contienen las fotografías que se encuentran en ella.

- **DDBB Project:** En esta pestaña es donde el usuario puede convertir una carpeta de trabajo en proyecto y almacenar todas las fotos en la base de datos, con el fin de poder realizar análisis de la información contenida en todas las fotografías.

A continuación se presentan todas las pantallas que visualizarán los usuarios y se detallan los procesos que se llevan a cabo en las mismas.

6.2.1. Exif Information

En esta pestaña se pueden identificar 3 zonas principales. La zona superior contiene una barra de menús. La zona de la izquierda, es la zona dedicada para la navegación de carpetas y ficheros almacenados en el Disco duro del equipo donde está la aplicación. La zona de la derecha muestra la información almacenada en la fotografía a analizar. La figura 6.1 muestra la Pestaña *Exif Information* con las 3 zonas delimitadas.

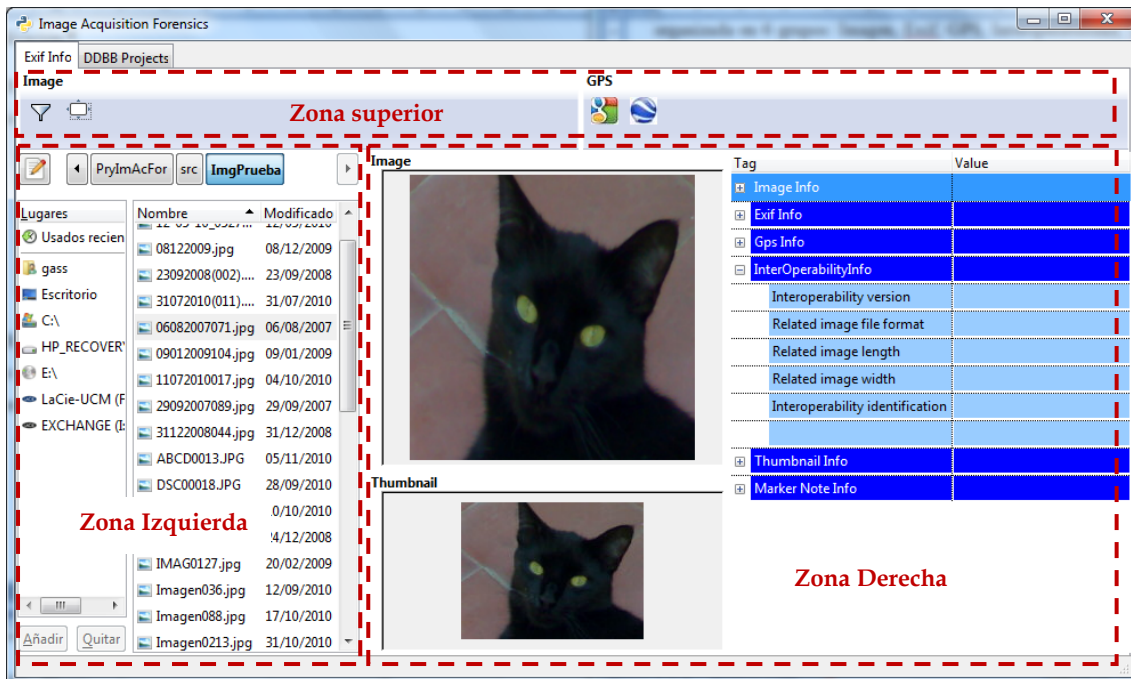


Fig. 6.1. Exif Information

En esta pantalla el usuario puede seleccionar una carpeta para ver el listado de fotografías que se encuentran almacenadas en ella. Una vez seleccionada la fotografía que se quiere analizar, se muestran los binarios de la imagen

almacenada, el *thumbnail* de la imagen (si lo posee); y la información *exif* contenida en el fichero organizada en 6 grupos: Imagen, Exif, GPS, Interoperabilidad, Thumbnail y Notas del Fabricante.

En la barra de menú se pueden observar dos grupos de menú Image y GPS

6.2.1.1. Menú Image

En las opciones de imagen, se puede modificar el tipo de filtro de imagen, así como los tamaños de las imágenes y de *thumbnail* que se muestran en la aplicación. La figura 6.2 muestra la pantalla para modificar tipos de ficheros que se pueden analizar.

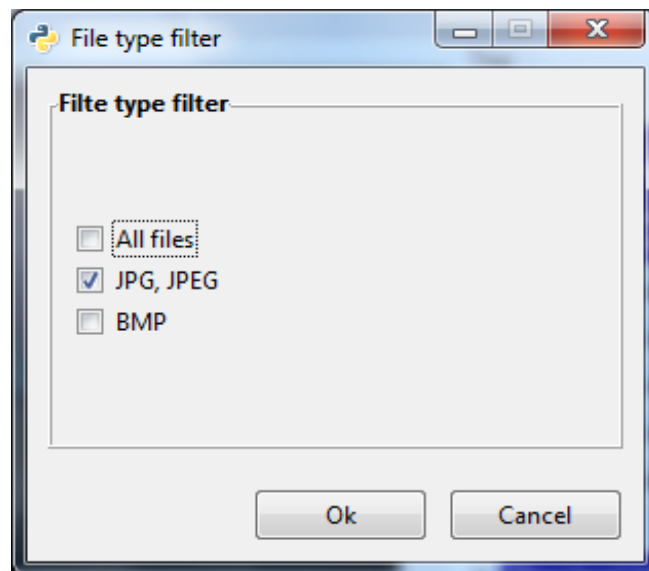


Fig. 6.2. File type filter

La figura 6.3 muestra la pantalla para modificar el tamaño de la imagen y del *thumbnail*.

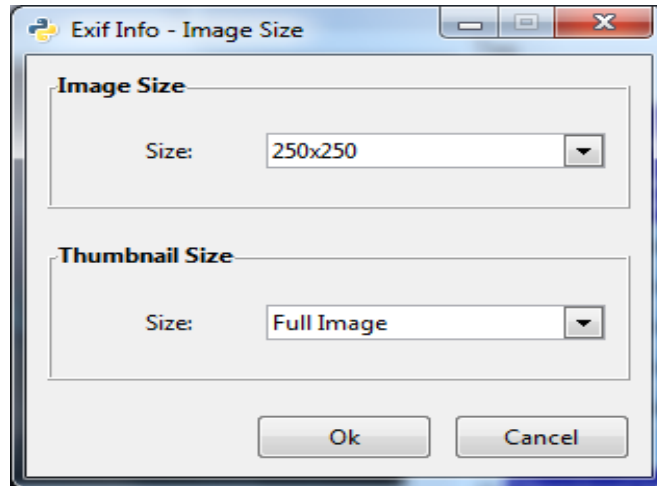


Fig. 6.3. Image Size

6.2.1.2. Menú GPS

En las opciones del menú GPS, se pueden visualizar la posición geográfica de la imagen (si ésta contenía información GPS).

Esta información se puede mostrar de dos formas:

La primera es utilizando los servicios de Google Maps. Para esto se debe contar con conexión a internet. La figura 6.4 muestra un ejemplo de la información GPS en una fotografía.

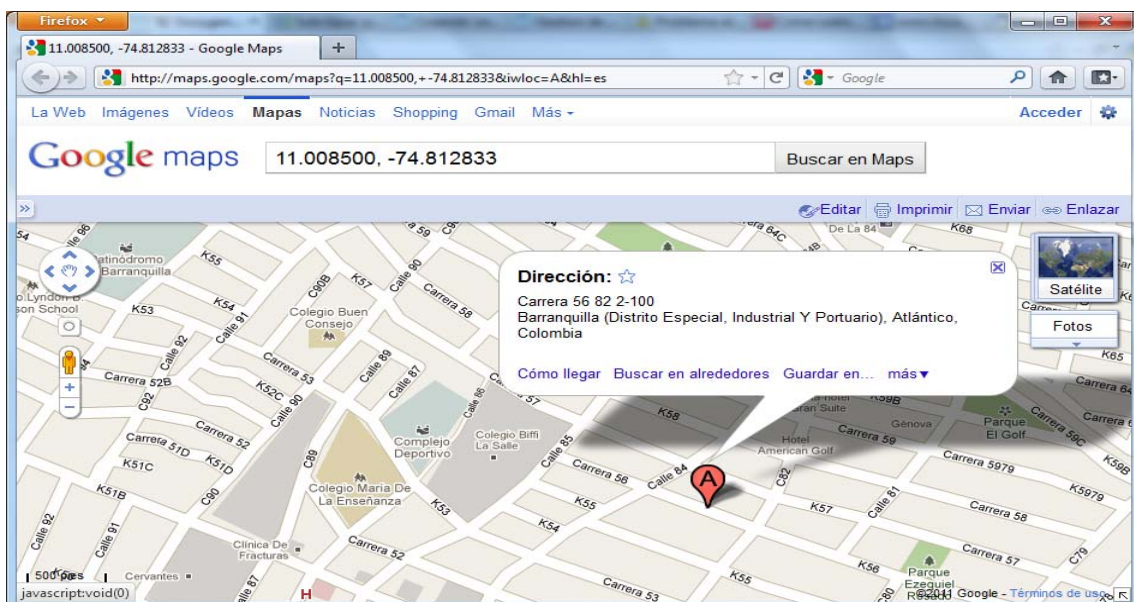


Fig. 6.4. Información GPS de una fotografía analizada.

Otra forma de mostrar esta información es generar un fichero .mkl que

puede exportarse a Google Earth y visual. La figura 6.5 muestra la pantalla para guardar el mapa de localización de la imagen para visualizarlo desde Google Earth.

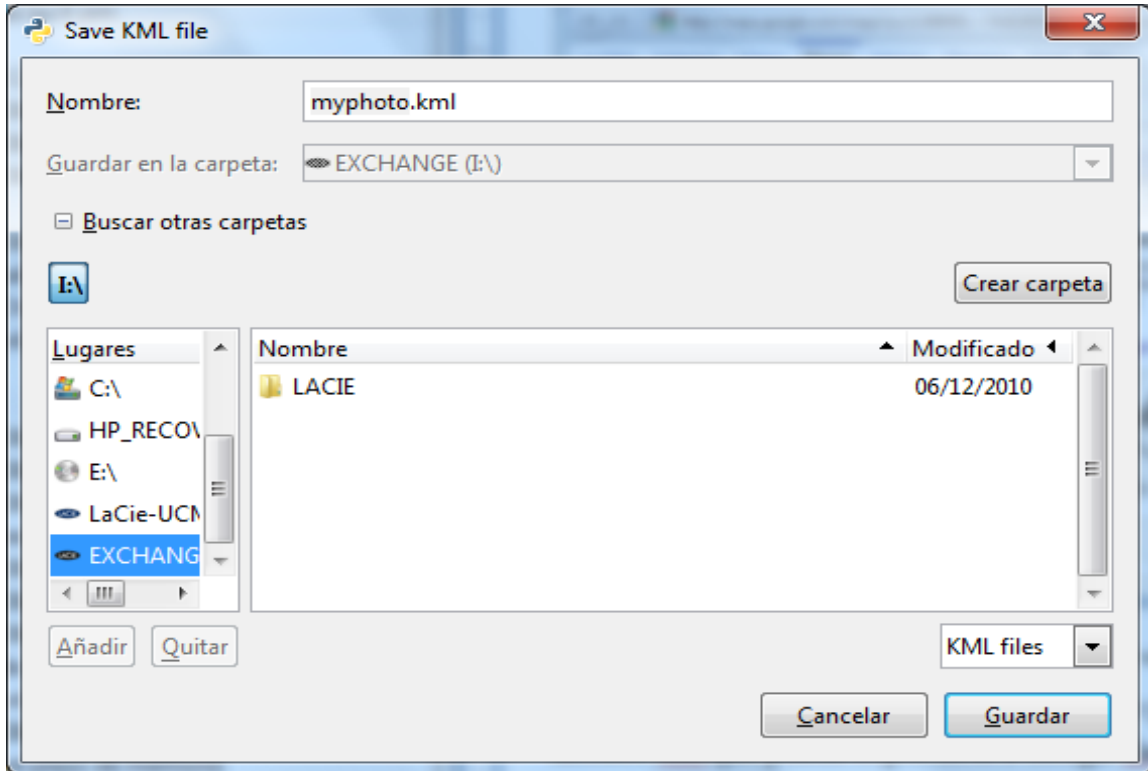


Fig. 6.5. Información GPS en formato KML

6.2.2. DDBB Projects

En esta pestaña se pueden almacenar todas las fotografías de una carpeta en la base de datos agrupadas en un proyecto para poder realizar los diferentes análisis con la información incluida en cada fotografía. De la misma forma que en la pestaña *Exif Information*, en esta pestaña se pueden identificar 3 zonas principales. La zona superior contiene una barra de menús; la zona de la izquierda en la que se muestra la información general de los proyectos existentes en la base de datos y la zona de la derecha se muestra una tabla con un resumen de la información Exif de cada una de las fotos del proyecto.

La figura 6.6 muestra la pestaña DDBB Project con las 3 zonas delimitadas.

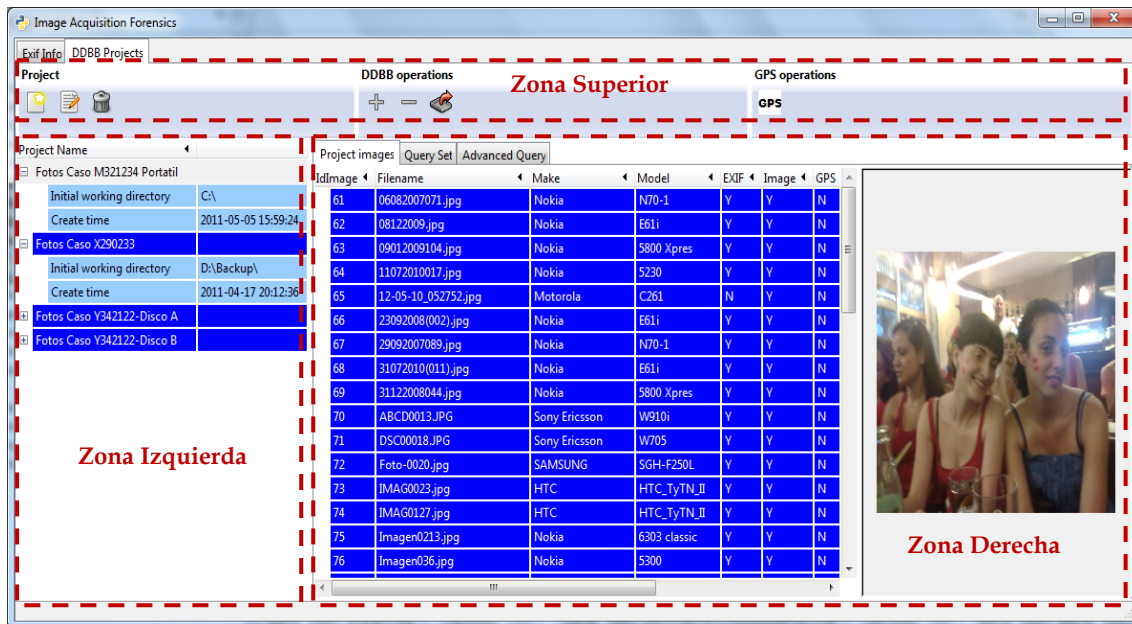


Fig. 6.6. Pestaña DDBB Project

En la barra de menús se pueden observar tres grupos de opciones: Project, DDBB Operations y GPS Operations.

6.2.2.1. Menú Project

En las opciones de Project se pueden realizar las siguientes operaciones:

Crear un Proyecto:

En esta opción se selecciona una carpeta del disco duro en la que se esté trabajando y se configuran los tipos de ficheros de imágenes que se van a procesar. Por ejemplo: jpg, bmp y gif, entre otros, o todos los ficheros existentes en la carpeta incluyendo ficheros con extensiones diferentes a imágenes. La figura 6.7 muestra la pantalla de Creación de proyectos.

Al hacer clic sobre el botón se procede a buscar dentro de la carpeta y subcarpetas todas los ficheros que sean imágenes y que coincidan con las extensiones especificadas en el botón

Si durante el proceso se presentan errores, se insertan en el proyecto creado las fotos que están correctas y se muestra un informe con los ficheros que no se pudieron agregar al proyecto y el motivo por el cual no pudieron adicionarse.

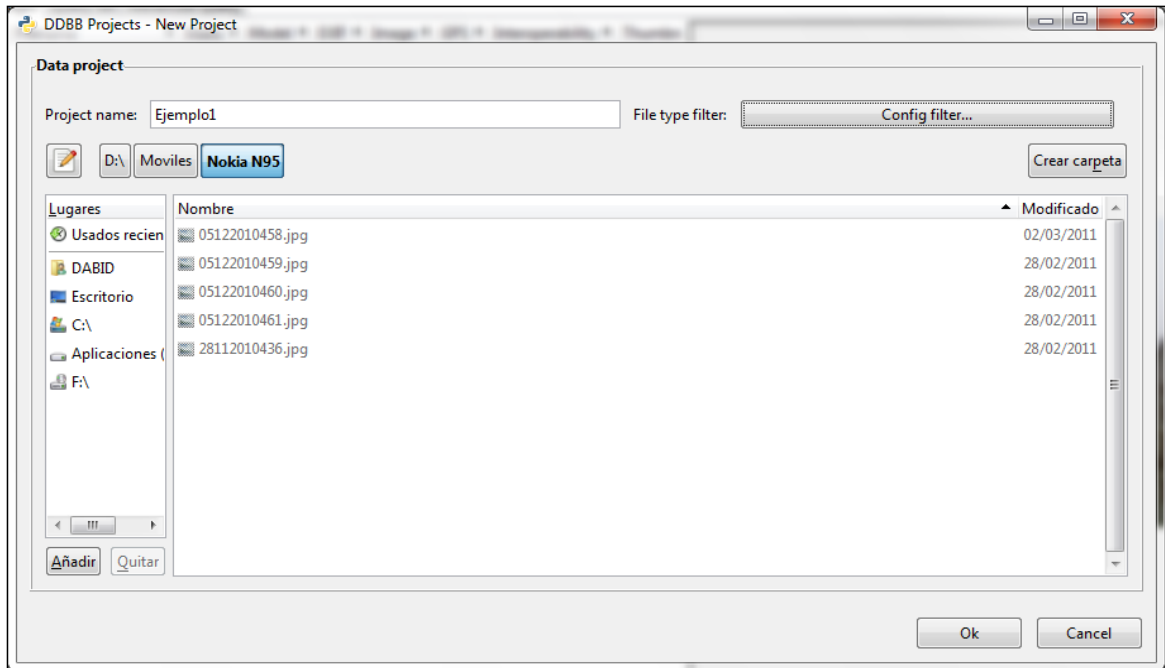


Fig. 6.7. New Project

Editar un Proyecto:

Una vez creado el proyecto aparece en la lista de proyectos existentes en la base de datos y se puede editar el nombre del proyecto. La figura 6.8 muestra la pantalla de Edición de proyectos.

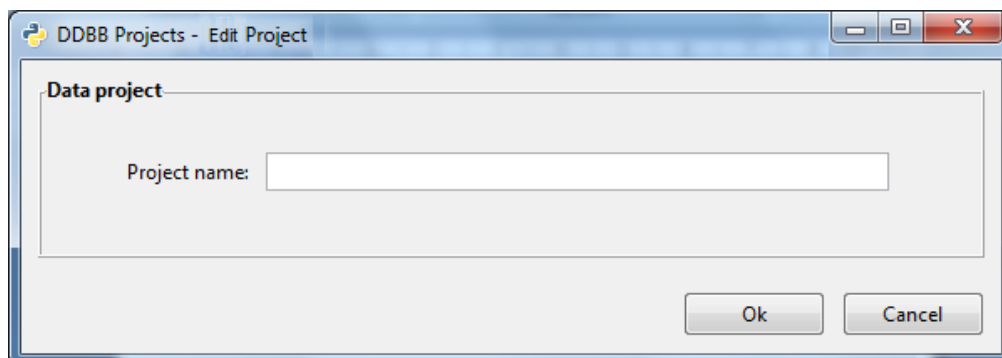


Fig. 6.8. Edit Project

Eliminar un Proyecto:

Con esta opción se selecciona el proyecto que se desea eliminar de la base de datos borrando también las fotografías de la base de datos. La figura 6.9 muestra la pantalla de confirmación para Eliminar un proyecto.

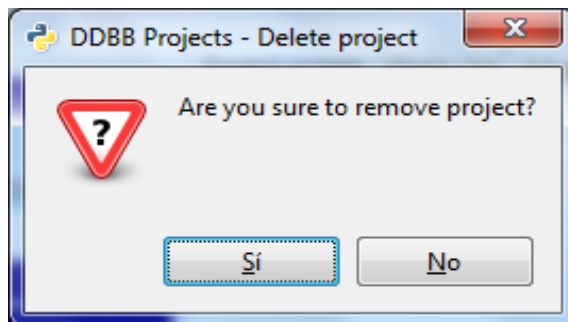


Fig. 6.9. Delete Project

6.2.2.2. Menú DDBB Operations

Las opciones que se pueden realizar en *DDBB Operations* son las siguientes:

Añadir Fotos:

Esta opción tiene como objetivo añadir fotografías a un proyecto seleccionado. Esta operación se realiza seleccionando los ficheros de imagen que se quieren añadir y/o carpetas que contienen los ficheros de fotografías. La figura 6.10 muestra la pantalla de Adición de fotografías a un proyecto.

Si durante el proceso se presentan errores se insertan en el proyecto creado las fotos que están correctas y se muestra un informe con los ficheros que no se pudieron agregar al proyecto y el motivo por el cual no pudieron adicionarse.

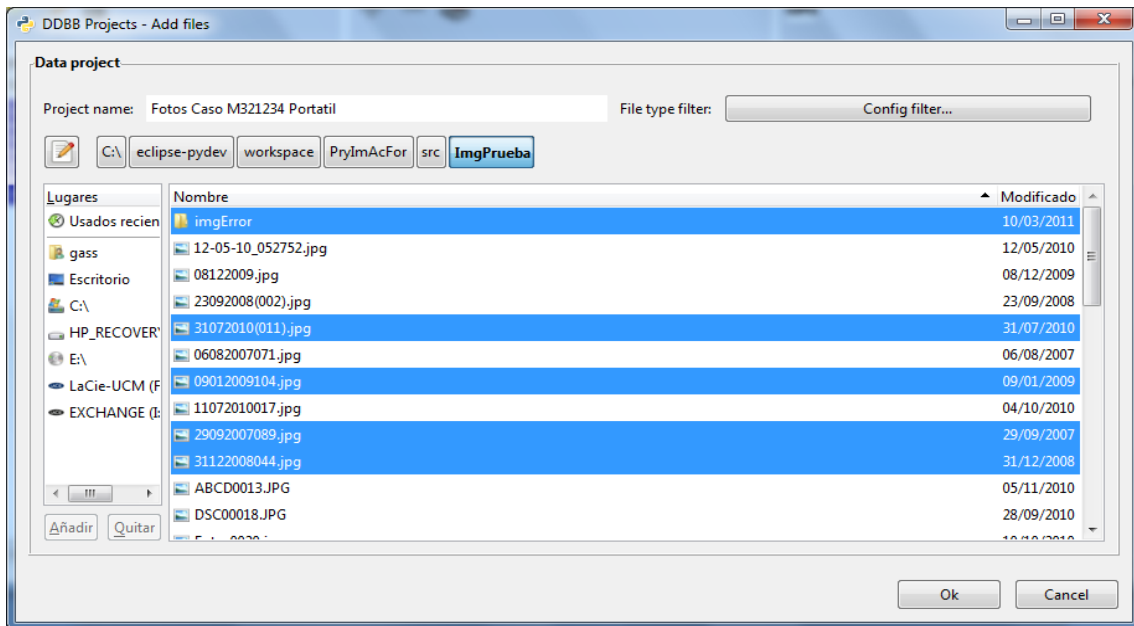


Fig. 6.10. Add Files

Eliminar Fotos:

Esta opción tiene como objetivo poder eliminar un grupo de fotografías de un proyecto seleccionado. Esta operación se realiza seleccionando las fotografías que se quieren eliminar de la base de datos. La figura 6.11 muestra la pantalla de Borrado de fotografías de un proyecto.

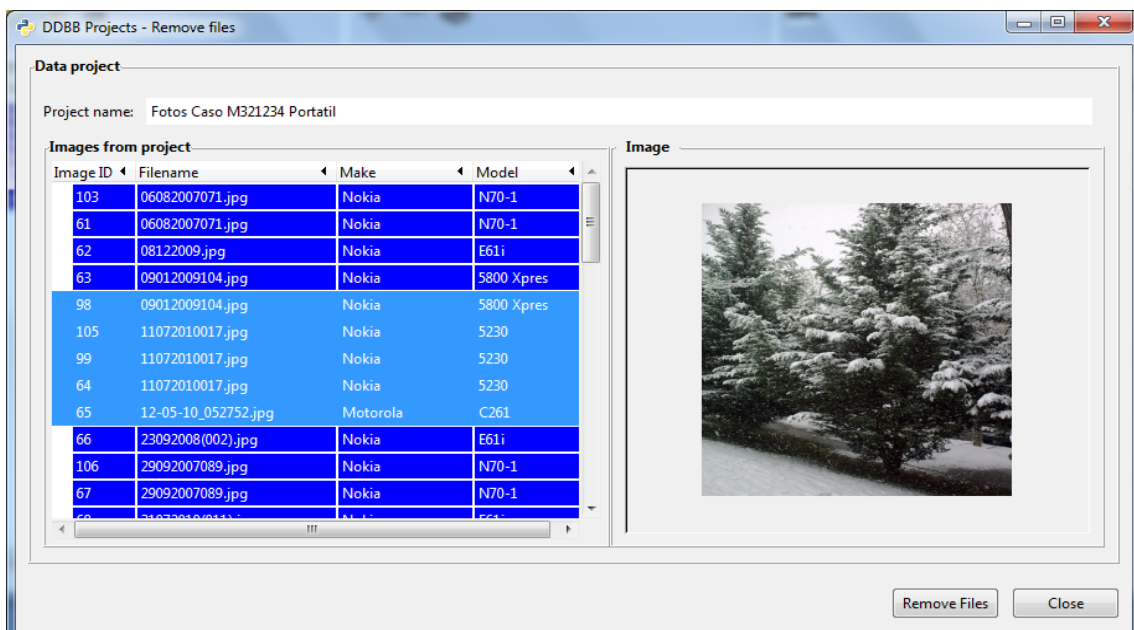


Fig. 6.11. Remove Files

Exportar Fotos:

Esta opción tiene como objetivo poder exportar un grupo de fotografías de un proyecto seleccionado a una carpeta como ficheros JPEG. La figura 6.12 muestra la pantalla de Exportar fotografías de un proyecto a Disco.

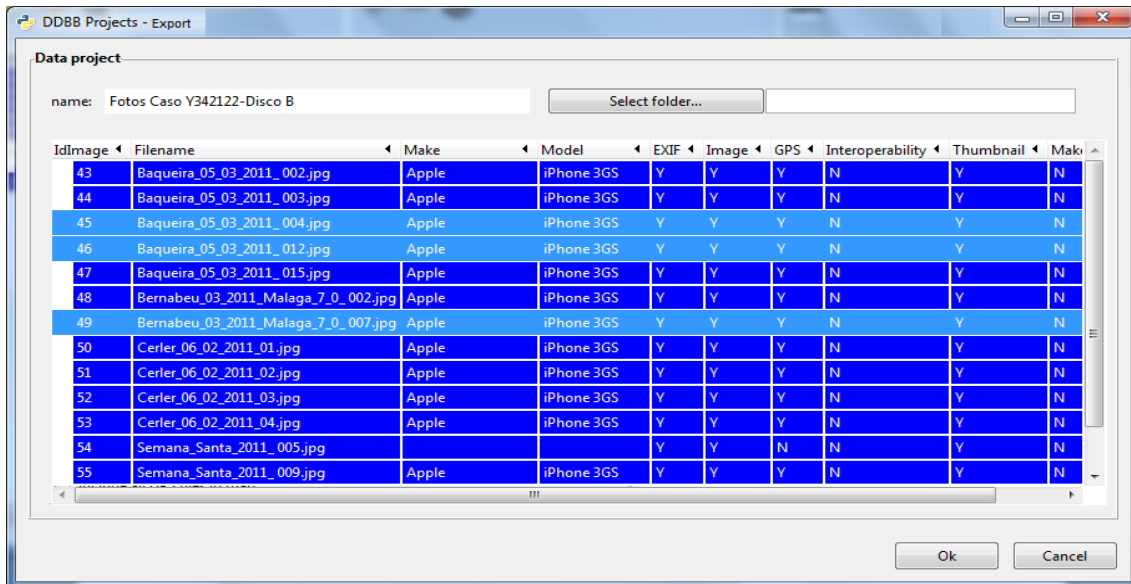


Fig. 6.12. Export Files

6.2.2.3. Menú GPS Operations

Las opciones que se pueden realizar en *GPS Operations* son las siguientes:

Visualización GPS:

Esta opción tiene como objetivo mostrar la localización de las fotografías de un proyecto seleccionado. Esta operación se realiza seleccionando los ficheros de imagen de los que se quiere ver la localización GPS y se apoya en la herramienta Google Maps para visualizar las distintas coordenadas de las fotografías. La figura 6.13 muestra la pantalla de visualización GPS. En esta pantalla se muestran las fotos del proyecto que poseen información GPS.

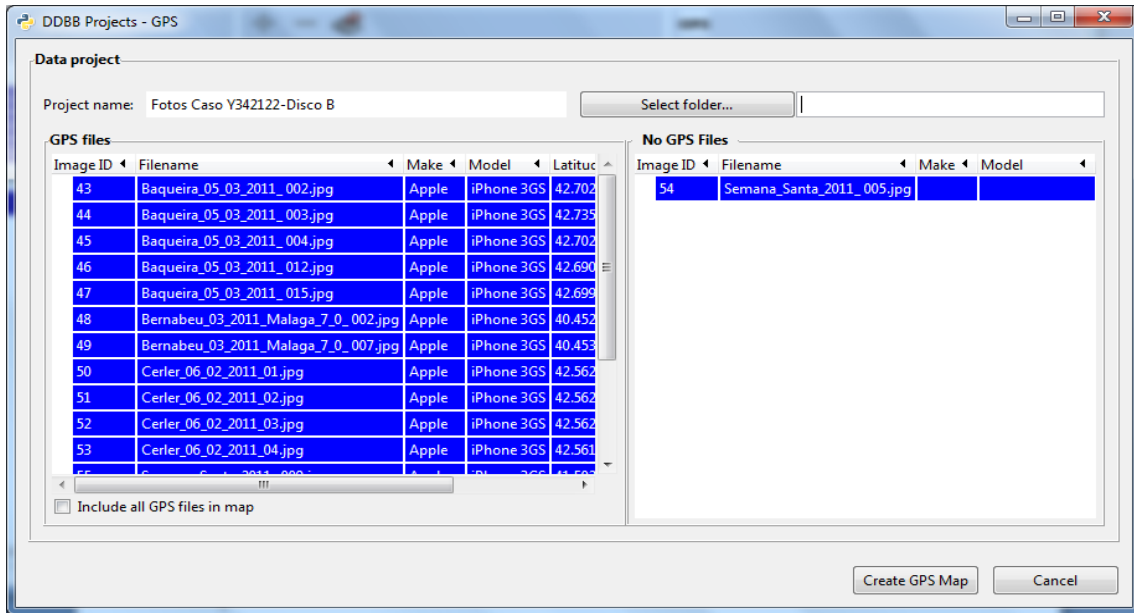


Fig. 6.13. GPS Information

Al hacer clic en el botón **Create GPS Map** se genera una URL que agrupa las fotos por zona, y a medida que se aumenta el zoom se va detallando las coordenadas. La figura 6.14 muestra un ejemplo de la URL generada aumentando el zoom en una zona.

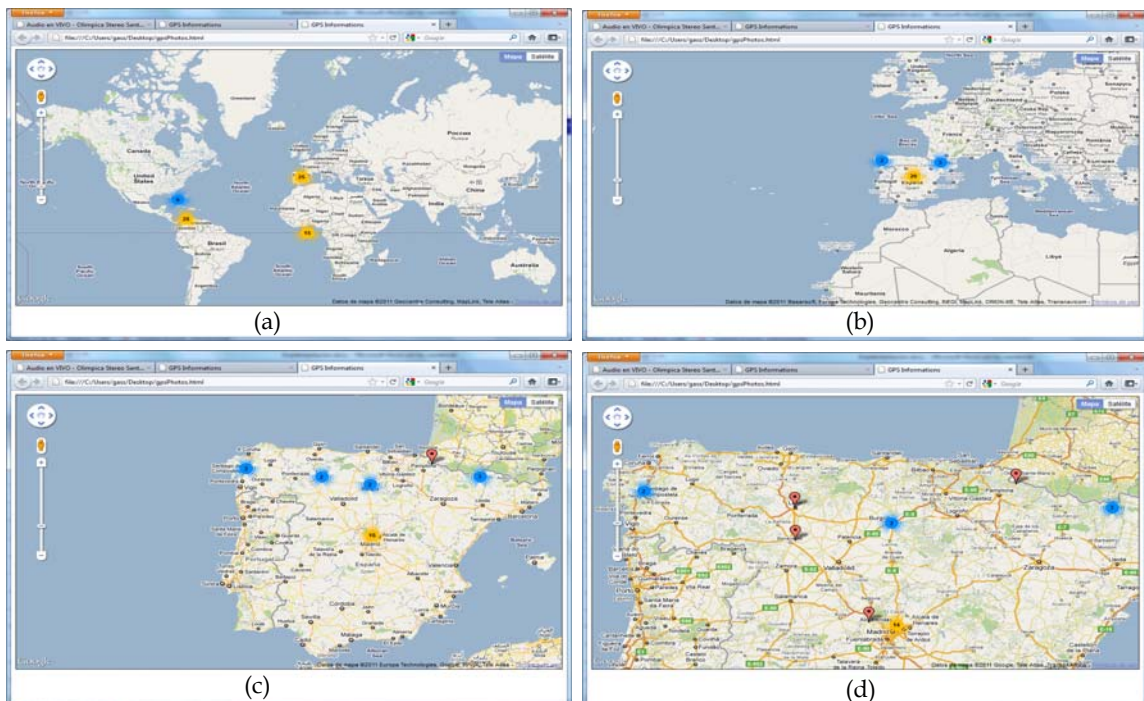


Fig. 6.14. GPS Information

6.2.2.4. Project Images

En esta pestaña se visualizan un listado de las fotografías que tiene el proyecto. Aquí se puede visualizar la imagen de la foto. La figura 6.15 muestra un ejemplo de la pantalla.

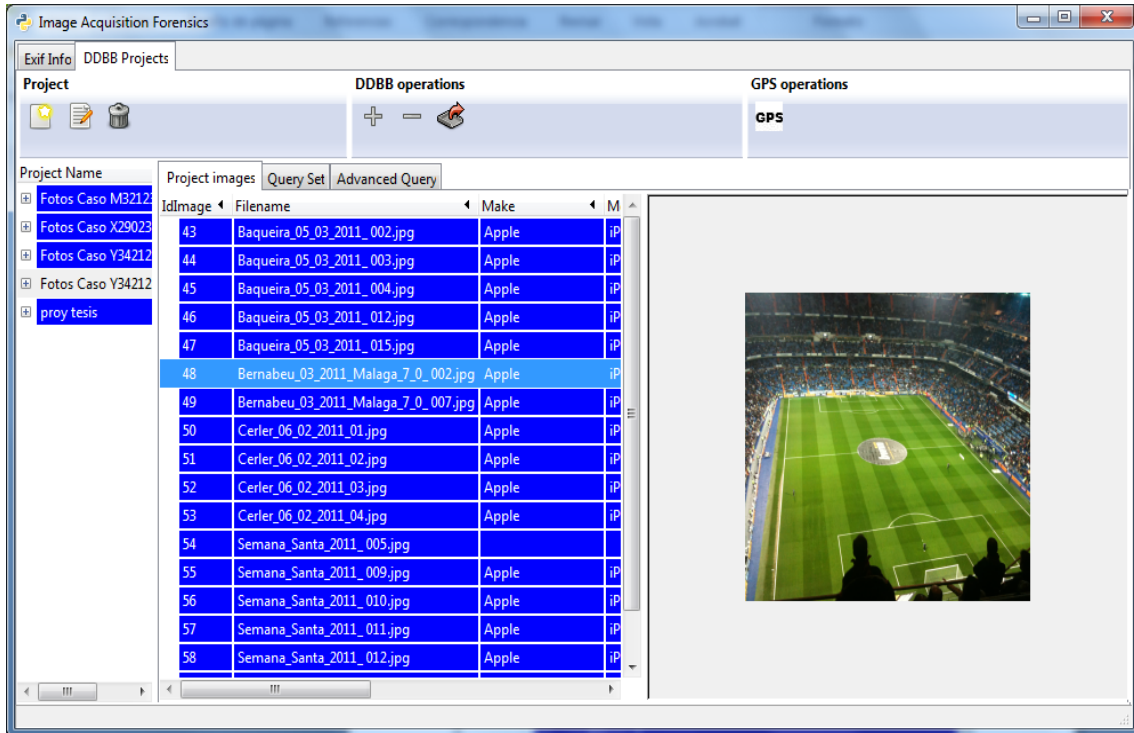


Fig. 6.15. Project Images

6.2.2.5. Query Set

En esta pestaña se pueden configurar las columnas por las que se quieren realizar estadísticas de las fotos de un proyecto. La figura 6.16 muestra un ejemplo de la pantalla.

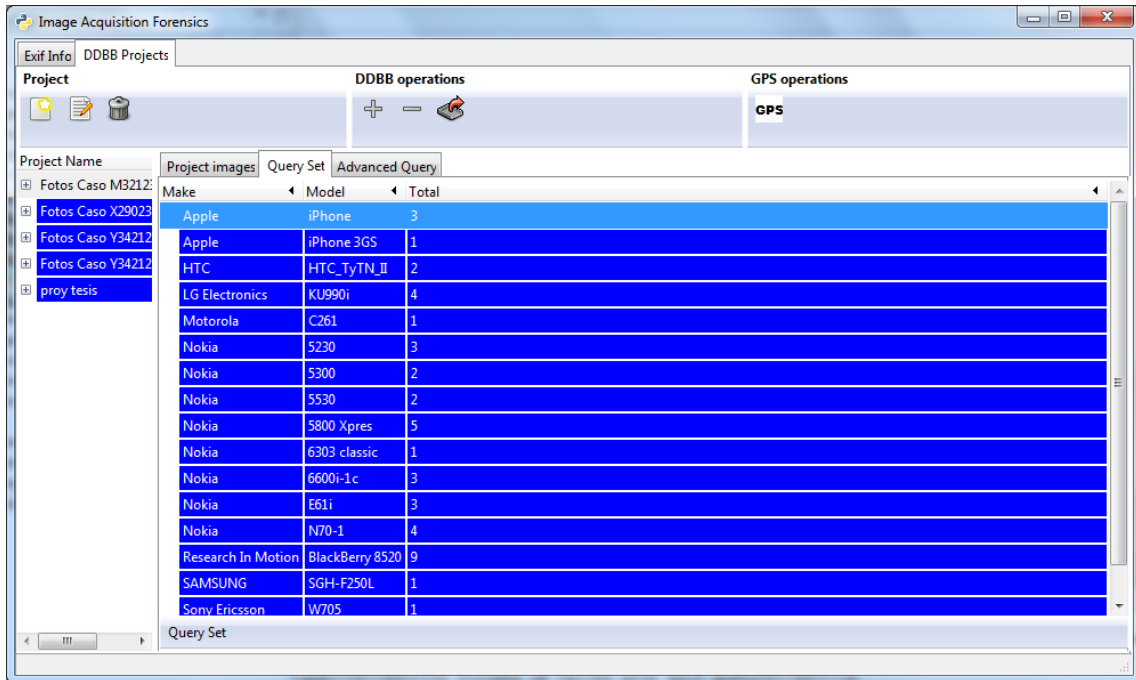


Fig. 6.16. Query Set

Para modificar las columnas que forman la parte de la estadística se hace clic en el botón **Query Set**. La figura 6.17 muestra la pantalla de configuración de Query Set donde se seleccionan las columnas que formarán parte de la estadística.

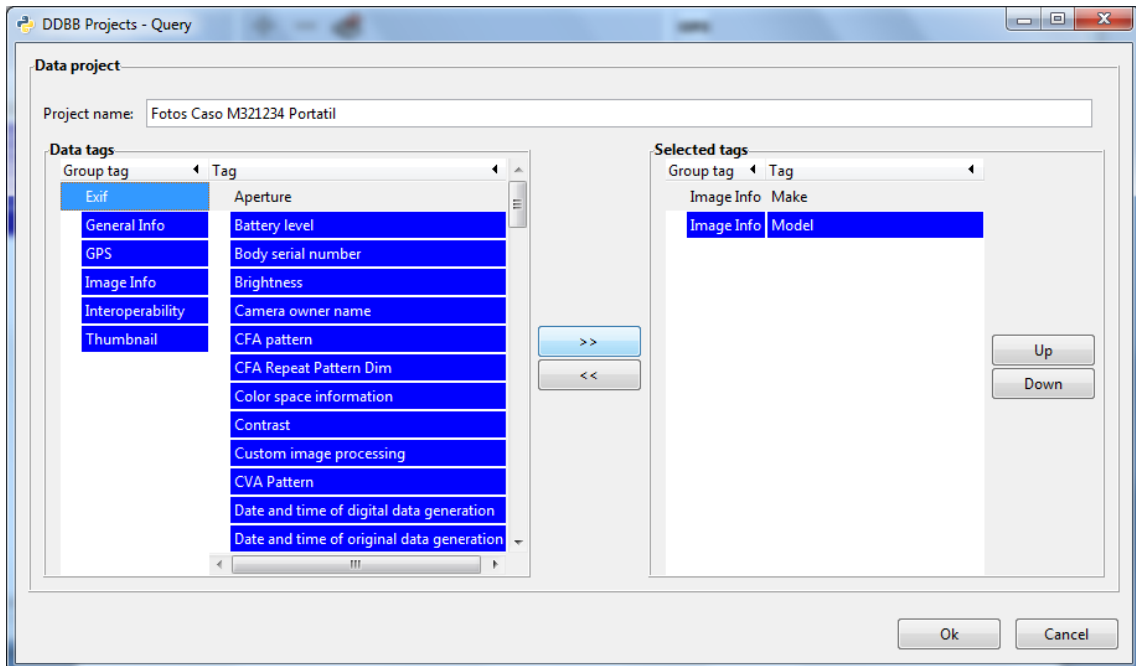


Fig. 6.17. Query Set

6.2.2.6. Advanced Query

En esta pestaña se pueden configurar diferentes consultas sobre las fotos almacenadas en la base de datos asociadas a un proyecto. La figura 6.18 muestra la pantalla de consultas avanzadas.

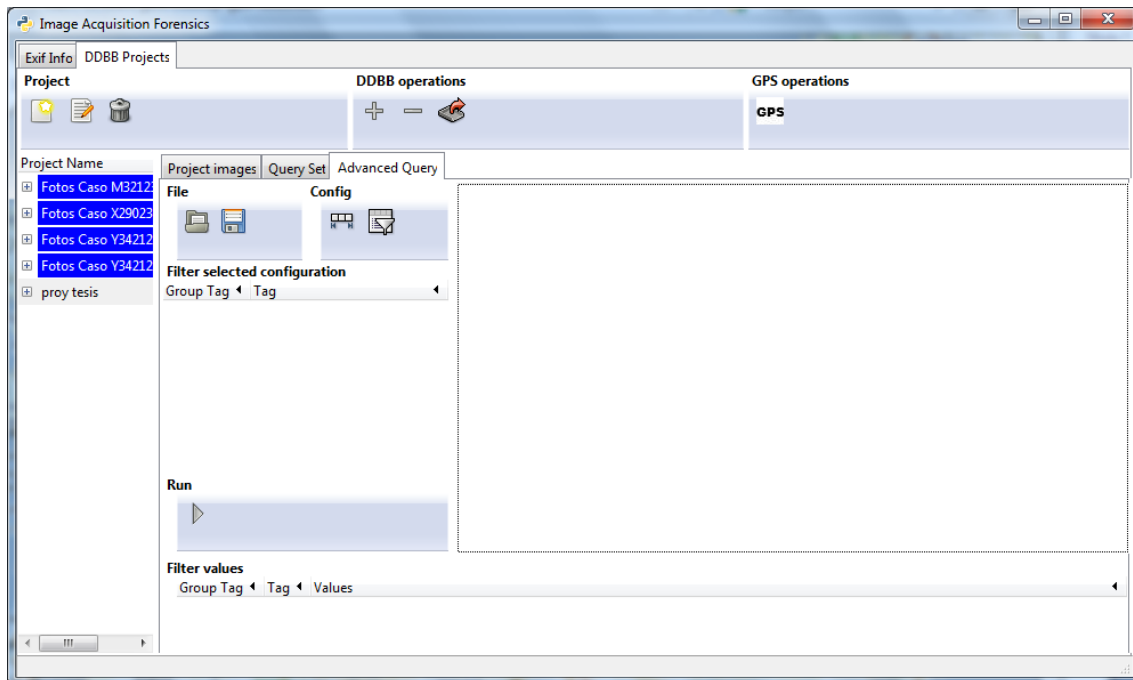


Fig. 6.18. Advanced Query

Para configurar la consulta se deben seleccionar las columnas que van a ser parte del SELECT (véase figura 6.19) y las columnas que van a conformar los filtros (véase figura 6.20)

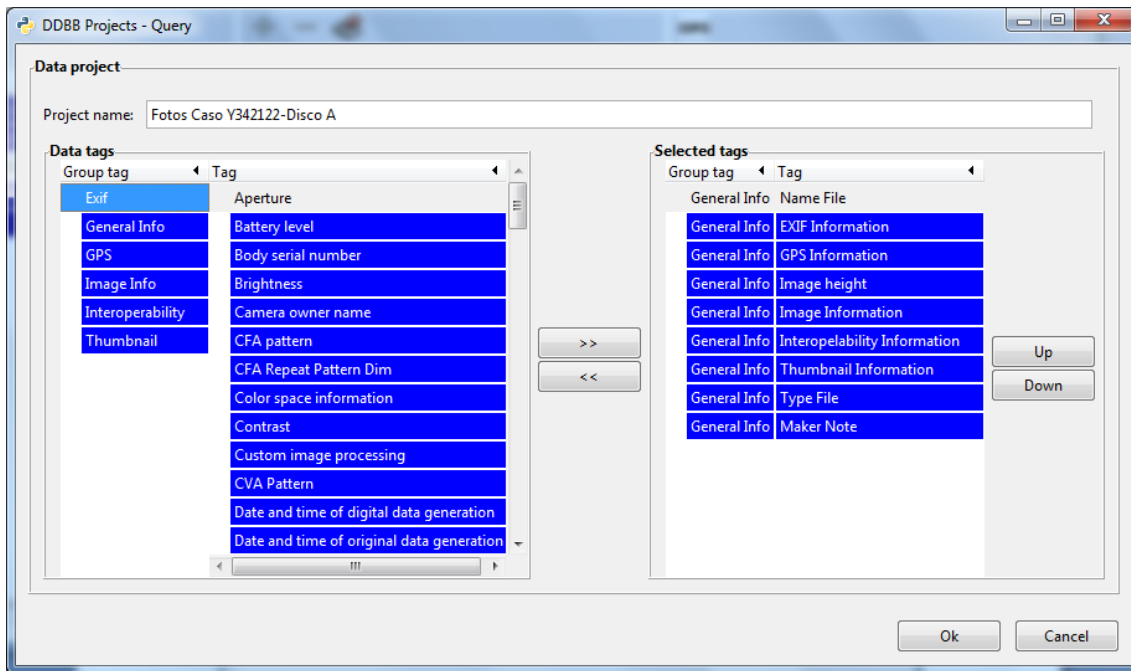


Fig. 6.19. Advanced Query -Select Columns

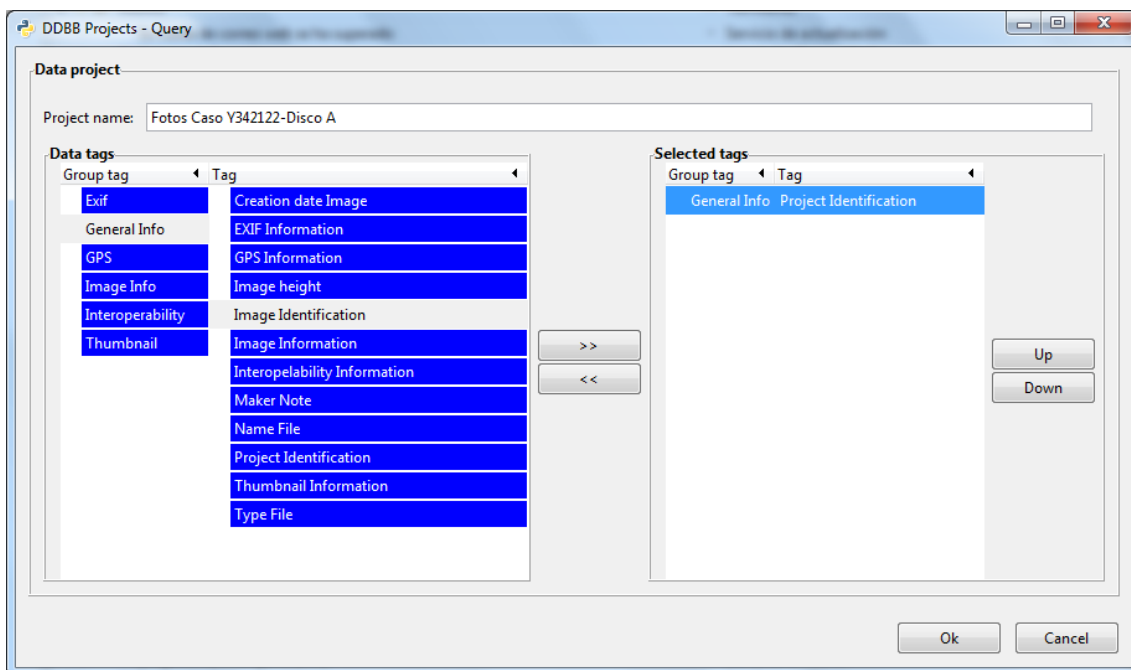



Fig. 6.20. Advanced Query - Filter Columns

Una vez se ha configurado la consulta de la base de datos ,se puede guardar para poder reutilizarla posteriormente. Para ver los resultados de la consulta configurada se hace clic en el botón . La figura 6.21 muestra los resultados de una consulta.

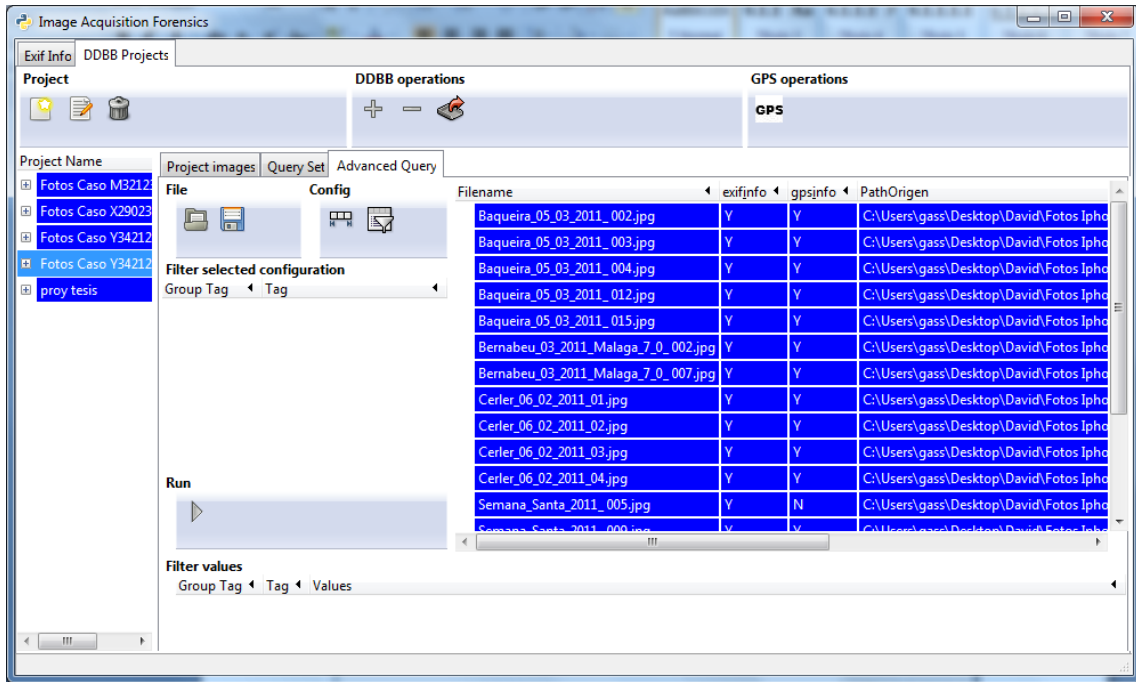


Fig. 6.21. Advanced Query –Run Query

7. CONCLUSIONES Y TRABAJO FUTURO

7.1. Conclusiones

La aplicación desarrollada en este proyecto cumple con el objetivo de ser una herramienta de apoyo para el análisis forense en dispositivos móviles. El análisis de la información que proporciona *ImgTool* resulta útil a la hora de aportar pruebas a la hora de resolver acciones fraudulentas que se pueden llevar a cabo en las fotografías tomadas desde dispositivos móviles. Este proceso se realiza a partir de la metainformación de la que dispone el dispositivo móvil, para poder llegar a verificar si dicha fotografía ha sido tomada desde cierto dispositivo o no.

El diseño e implementación de *ImgTool* cumple con el objetivo de ser una herramienta fácil de utilizar, intuitiva y amigable para el usuario. De esta manera *ImgTool* facilita el trabajo del analista forense al proporcionarles un análisis global de la información

Los aspectos que se pueden destacar hemos de citar los siguientes:

- Importancia de la metainformación en una imagen.
- Manipulación de la información añadida.
- Diversos usos de la metainformación EXIF (fabricante y modelo de la cámara, ubicación de la imagen, etc.)

7.1.1. Conocimientos Adquiridos:

Desarrollar este Proyecto ha sido toda una experiencia muy positiva por muchas razones. Aunque es cierto que ha habido momentos de desesperación muy frustrantes, sólo representa una minoría con respecto al lado positivo de esta experiencia vivida.

Se han aprendido muchas cosas nuevas, desde programar con un lenguaje de

programación nunca utilizado, como es Python, pasando por utilizar librerías como Python Imaging Library, LibSVM, desarrollar código con la API de Google Maps entre otros; conocer nuevos sistemas como han sido DoxyGen, Glade y tepache, y terminando por el aprendizaje de utilizar herramientas para conocer la información que se almacena en las imágenes digitales.

El desarrollo para esta herramienta de apoyo al análisis forense ha sido también toda una experiencia. Además de aprender el lenguaje Python, hubo que documentarse mucho para poder comprender lo mejor posible la librería PIL proporcionada para el procesamiento de imágenes digitales, para codificar algoritmos de procesamiento de imágenes.

Se podría destacar como resultado satisfactorio, es el esfuerzo personal aplicado para que, partiendo de un desconocimiento casi absoluto tanto de procesamiento digital de imágenes se haya conseguido una experiencia que, a día de hoy, se demuestra bastante soltura.

7.2. Trabajo Futuro

Cada día, se pone de manifiesto la vulnerabilidad de los dispositivos móviles, en cuanto a la seguridad que nos proporcionan. Además, observando las estadísticas proporcionadas en este documento, vemos que el uso de estos dispositivos va en aumento. Y por tanto, es predecible, que los intentos de vulnerar la seguridad como los hechos delictivos, también irán en aumento.

En el campo que ocupa este trabajo, en concreto las imágenes digitales proporcionadas en dispositivos móviles, podemos ampliar la investigación y desarrollo de herramientas, para verificar la autenticidad de dichas imágenes y proporcionar información relevante sobre ellas en un proceso judicial.

En este terreno, podemos citar como trabajo futuro:

- La implementación de diferentes algoritmos para extraer características de la imagen analizada para utilizarlas como diferenciadores en clasificadores.

- Utilizar como clasificadores Support Vector Machines, con la ya implementada librería LIBSVM. Este tipo de clasificadores nos permite, introduciendo las características adecuadas, clasificar imágenes de diferentes modelos y marcas con una probabilidad de acierto grande (sobre 80-90%). En el Anexo A, se detalla el concepto de clasificación y la utilización de librerías.

REFERENCIAS

- [1] C. Castillo, A. Romero: "Guía Metodológica Para el Análisis Forense Orientado a Incidentes en Dispositivos Móviles GSM". Pontificia Universidad Javeriana, 2008.
- [2] J. C. Ardita: "Metodología de Análisis Forense Informático". Conferencia Profesional Metodología de Análisis Forense Informático en ISACA - ADACSI, 2007.
- [3] H. Helmers, D. D. Carl, T. Sievers: "CMOS-ESPI-system with in-line digital phase stabilization using unresolved speckles". *Proceedings of Interferometry XI: Techniques and Analysis*, K. Creath and J. Schmit, eds., Proc. SPIE 4777, pp. 232-241, 2002.
- [4] S. Bayram, N. Memon: "Classification of digital camera-models based on demosaicing artifacts". *Digital Investigation*, Vol. 5, pp. 46-59, 2008.
- [5] Gartner Newsroom. 2011.
<http://www.gartner.com/it/page.jsp?id=1689814>.
- [6] Interactive Advertising Bureau. 2011.
<http://www.iabspain.net>.
- [7] Adobe Developers Association. 1993
<http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>.
- [8] C. Hamilton, C. Cube: Microsystems. JPEG File Interchange Format. Version 1.02. September 1, 1992.
<http://www.w3.org/Graphics/JPEG/jfif3.pdf>.
- [9] International Press Telecommunications Council
<http://www.iptc.org/site/Home/>.

- [10] CIPA, Standard of the Camera & Imaging Products socation, ExifVersion2.3 Exchangeable image file format for digital still cameras.
http://www.cipa.jp/english/hyoujunka/kikaku/pdf/DC-008-2010_E.pdf
- [11] Python Programming Language.
<http://www.python.org>.
- [12] Glade - A User Interface Designer.
<http://glade.gnome.org/>
- [13] MySql.
<http://www.mysql.com/>
- [14] Assembla workspaces.
<https://www.assembla.com/>
- [15] EXIF, IPTC editor & viewer.
[http://www.photoinfoex.com /](http://www.photoinfoex.com/)
- [16] Exif Jpeg header manipulation tool.
[http://www.sentex.net/~mwandel/jhead /](http://www.sentex.net/~mwandel/jhead/)
- [17] ExifTool.
<http://www.sno.phy.queensu.ca/~phil/exiftool/>
- [18] Exif Viewer 1.60.
<https://addons.mozilla.org/es-es/firefox/addon/exif-viewer/>
- [19] C.J.C. Burges: "A Tutorial on Support Vector Machines for Pattern recognition", *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, pp. 121-167, 1998.

[20] C.-C. Chang, C.-J. Lin: "LIBSVM: A Library for Support Vector Machines", 2001.

<http://www.csie.ntu.edu.tw/~cjlin/libsvm>

[21] R. C. Gonzalez, R. E. Woods: "Tratamiento digital de imágenes", Addison-Wesley, 1996.

ANEXOS

A. CLASIFICACIÓN

La clasificación consiste en predecir el valor de un atributo (clase) basándose en los valores de otros atributos (los atributos predictores).

La clasificación puede ser formalizada como la tarea de aproximar una función objetivo desconocida:

$$\Phi = I \times C \rightarrow \{T, F\} \quad (A.1)$$

La expresión A.1 describe cómo instancias del problema deben ser clasificadas de acuerdo a un experto en el dominio) por medio de una función A.1 llamada el clasificador, donde:

$$C = \{c_1..c | c\} \quad (A.2)$$

es un conjunto de categorías predefinido, e I es un conjunto de instancias del problema.

Comúnmente cada instancia $ij \in I$ es representada como una lista :

$$A = [a_1..a | A |] \quad (A.3)$$

de valores característicos, conocidos como atributos, es decir:

$$ij = [a_1j..a | A |j] \quad (A.4)$$

Sea:

$$\Phi = ij \times ci \rightarrow T \quad (A.5)$$

Si (A.5), entonces ij es llamado un ejemplo positivo de ci , mientras

Si:

$$\Phi = ij \times ci \rightarrow F \quad (A.6)$$

éste es llamado un ejemplo negativo de ci .

Para generar automáticamente el clasificador de es necesario un proceso inductivo, llamado proceso de aprendizaje, el cual por observar los atributos de

un conjunto de instancias preclasificadas bajo ci o $\neg ci$, adquiere los atributos que una instancia no vista debe tener para pertenecer a la categoría. Por tal motivo, en la construcción del clasificador se requiere la disponibilidad inicial de una colección Ω de ejemplos tales que el valor de:

$$\Phi = (ij, ci) \tag{A.7}$$

(A.7) es conocido para cada:

$$(ij, ci) \in \Omega \times C \tag{A.8}$$

A la colección usualmente se le llama conjunto de entrenamiento Tr .

Al proceso anterior se le identifica como aprendizaje supervisado debido a la dependencia de Tr .

A.1. Clasificadores

Un clasificador es un elemento que proporciona una clase etiquetada como salida a partir de un conjunto de características tomadas como entradas. Una manera de construir un clasificador es coger un conjunto de ejemplos etiquetados y tratar de definir una regla que pueda asignar una etiqueta a cualquier otro dato de entrada.

El aprendizaje automático se considera como un subcampo de la Inteligencia Artificial y tiene que ver con el desarrollo de técnicas y métodos que permiten a la computadora aprender. En términos simples, el desarrollo de algoritmos que permiten a la máquina de aprender y realizar tareas y actividades. Máquina de aprendizaje se superpone con las estadísticas de muchas maneras. Durante el período de tiempo muchas técnicas y metodologías se han desarrollado para tareas de aprendizaje automático.

A.1.1. Aprendizaje

El Aprendizaje automático es la parte básica que tienen en común los diferentes tipos de clasificadores que existen. La idea básica del aprendizaje consiste en

utilizar las percepciones no sólo para actuar, sino también para mejorar la habilidad de un agente para actuar en el futuro. Existen diferentes tipos de técnicas de aprendizaje:

A.1.1.1 Aprendizaje supervisado

El aprendizaje supervisado consiste en aprender una función, a partir de ejemplos etiquetados anteriormente, que establezca una correspondencia entre las entradas y las salidas deseadas del sistema. No siempre es posible hacer este tipo de entrenamiento ya que tenemos que disponer de la salida esperada en la función de entrada. El sistema de aprendizaje trata de etiquetar (clasificación) una serie de vectores utilizando una entre varias categorías (clases).

A.1.1.2 Aprendizaje no supervisado

El aprendizaje no supervisado consiste en aprender a partir de patrones de entradas para los que no se especifiquen los valores de sus salidas. El principal problema de esta técnica es la toma de decisiones a la hora de escoger un patrón entre todos los proporcionados. El sistema trata los objetos de entrada como un conjunto de variables aleatorias, construyendo un modelo de densidad para el conjunto de datos.

A.1.1.3 Aprendizaje semi-supervisado

Actualmente existen técnicas que combinan las dos anteriores, ya que en algunos casos puede resultar muy costoso asignar etiquetas o clases a todos los datos. La finalidad es combinar datos etiquetados y no etiquetados para mejorar la construcción de modelos. Aunque no siempre es útil y existen varios métodos para llevarlo a cabo.

A.1.1.4. Aprendizaje por refuerzo

El aprendizaje por refuerzo consiste en aprender observando el mundo que te rodea. La idea del aprendizaje consiste en construir una función que tenga el comportamiento observado en sus datos de entrada y de salida. Los métodos de aprendizaje se pueden entender como la búsqueda de un espacio de hipótesis

para encontrar la función adecuada.

A.2. Tipología de Clasificadores

A continuación se exponen algunos de los diferentes tipos de clasificadores que existen.

A.2.1. Clasificador Bayesiano

Un Clasificador bayesiano es un clasificador de patrones basado en teorías estadísticas de aprendizaje. El aprendizaje bayesiano calcula la probabilidad de cada hipótesis de los datos y realiza predicciones sobre estas bases. Es un aprendizaje casi óptimo, pero requiere grandes cantidades de cálculo debido a que el espacio de hipótesis es normalmente muy grande, o incluso puede ser infinito.

A.2.2. Clasificador Parzen

Se trata de un clasificador basado en la estimación no paramétrica, que a diferencia de la paramétrica, donde se obtiene la función de densidad de probabilidad estimando los parámetros desconocidos de un modelo conocido, no se conoce el modelo. Esta técnica consiste básicamente en variaciones de la aproximación del histograma de una función de densidad de probabilidad desconocida. Este algoritmo se utiliza en la clasificación de imágenes o para clasificar y aprender simultáneamente. El problema de este clasificador es que tiene un tiempo de ejecución elevado.

A.2.3. Clasificador Backpropagation

En algunos problemas el uso de modelos simples de densidad paramétrica o el uso de modelos de histogramas no dan los resultados deseados. En estos casos se buscan modelos de densidad más sofisticados. Las redes neuronales son una técnica de aproximación paramétrica útil para construir modelos de densidad.

El modelo de red neuronal habitual que utiliza este algoritmo consiste en una

red con una capa de entrada con tantos nodos como entradas tengan, una capa oculta con un número de nodos variable que dependerá de las características del problema, y una capa de salida con tantos nodos como posibles salidas tenga.

A.2.4. Clasificador con PCA

En este clasificador se utiliza el método PCA (Principal Component Analysis), es una técnica que busca reducir el número de variables, transformando el conjunto original de variables correlacionadas, que tienen información en común, en un conjunto de variables no relacionadas, que no tienen repetición o redundancia entre ellas, llamado conjunto de componentes principales. Las nuevas variables son combinaciones lineales de las anteriores.

A.2.5. Máquinas de vectores de soporte (Support vector machine)

Las Máquinas de vectores de soporte (SVM) son unas técnicas sencillas y de gran aplicación cuando se pretende construir un clasificador utilizando ejemplos. A diferencia de las redes neuronales que intentan construir un modelo a posteriori, las SVM's tratan de obtener la frontera de decisión. Una de sus ventajas es su facilidad ya que únicamente hay que codificar la geometría de la frontera.

A.3. Support Vector Machine (SVM)

Una máquina de vectores de soporte es una función que asigna cada valor de entrada a una clase (positiva o negativa), es decir, es una clasificación binaria, aunque existen técnicas para extenderlas a multiclase.

Diseñada en principio para tratar problemas de clasificación binarios (en dos grupos), se trata de una máquina de aprendizaje que implementa la siguiente idea: cuando no sea posible separar los datos en el espacio de entrada con un

hiperplano lineal, trasladar, mediante una aplicación no lineal, los vectores de entrada a un nuevo espacio de dimensión más alta. En este nuevo espacio se construirá una superficie de decisión lineal. Las especiales propiedades que poseerá esta superficie garantizarán que la capacidad de generalización de la máquina de aprendizaje sea alta. Aunque esta idea se empleó en los primeros experimentos para datos que podían separarse sin errores, se puede extender para el caso no separable con notable éxito. La parte conceptual del problema la resolvió Vapnik para el caso de hiperplanos óptimos para clases separables. En este contexto, Vapnik definió un hiperplano óptimo como una función de decisión lineal con el margen de separación máximo entre los vectores de las dos clases. Se observó entonces que para construir el hiperplano, uno sólo debía tener en cuenta una cantidad pequeña de los datos de entrenamiento, los llamados vectores soporte, quienes determinaban ese margen [19].

Gráficamente: $(\mathbb{R}^n \rightarrow \{+1, -1\})$

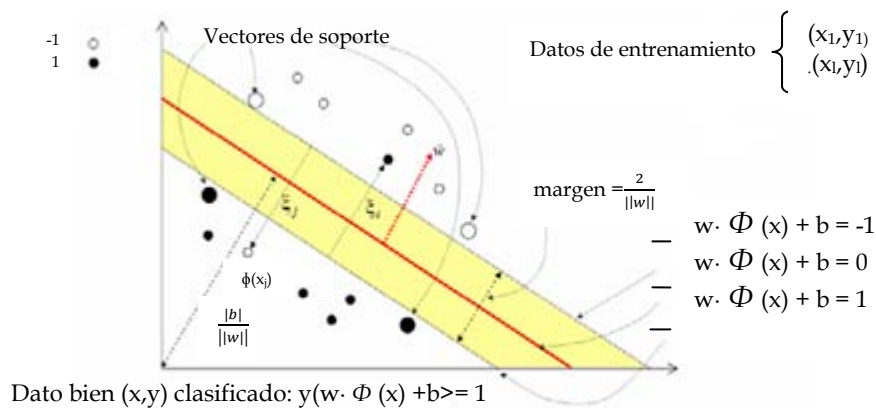


Fig. A.1. Representación gráfica de máquinas de soporte vectorial

Para tener una máquina bien adaptada, debe ser entrenada con datos que pertenezcan a clases conocidas (datos de entrenamiento).

Matemáticamente:

Sea X , tal que:

$$X = \{(x_1, y_1), \dots, (x_l, y_l) : x_i \in \mathbb{R}^n, y_i \in \{-1, +1\}\} \quad (A.9)$$

En (A.9), x_i son los puntos-dato e y_i su etiqueta o clase (puede ser -1 o 1).

La función de decisión

$$f_x : \mathbb{R}^n \rightarrow \{-1, +1\} \quad (A.10)$$

asigna los vectores de entrada x_i a la clase.

Para separar los dato-puntos en 2 clases de \mathbb{R}^2 , una forma sencilla e intuitiva es la construcción de una línea recta de separación y un plano de separación en \mathbb{R}^3 . En el espacio de dimensiones superiores hablamos de hiperplanos.

Supongamos que es posible separar este conjunto de entrenamiento mediante un hiperplano lineal. Los puntos x que pertenecen al hiperplano satisfacen la ecuación:

$$w \cdot x + b = 0 \quad (A.11)$$

donde w es un vector normal al hiperplano, y:

$$|b| / \|w\| \quad (A.12)$$

es la distancia perpendicular del hiperplano al origen (tomamos $\|\cdot\|$ como la norma euclídea). De entre todos los hiperplanos capaces de separar los datos, existe un único hiperplano óptimo, en el sentido de que es capaz de separar los puntos con el mayor margen de separación entre cada elemento del conjunto de entrenamiento y el hiperplano. En este sentido, el algoritmo de aprendizaje diseñado por Vapnik y Chervonenkis, las máquinas de soporte de vectores, resuelve el siguiente problema:

Encontrar $w \in \mathbb{R}^n$ y $b \in \mathbb{R}$ que minimicen:

$$\tau(w) = \frac{1}{2} \|w\|^2 \quad (A.13)$$

Sujeto a la condición:

$$y_i (w^T \cdot x_i + b) \geq 1 \quad \forall i = 1, \dots, l \quad (A.14)$$

Cuando se hallen w y b , la regla de clasificación será, simplemente:

$$\text{sign}(w^T \cdot x_i + b) \quad (A.15)$$

y el error de clasificación cometido vendrá dado por R :

$$ent(w,b) \tag{A.16}$$

Por otro lado, aquellos puntos que verifican la igualdad en la inecuación

$$y_i (w^T \cdot x_i + b) \geq 1 \tag{A.17}$$

La eliminación de los puntos que verifican (A.14) cambiaría la solución que encontremos, son los que llamaremos vectores soporte. Estos vectores, pertenecerán a uno de los dos posibles hiperplanos óptimos de separación de los que hablábamos antes, representados para los dos hiperplanos por las ecuaciones:

$$w^T \cdot x_i + b = 1 \tag{A.18}$$

$$w^T \cdot x_i + b = -1 \tag{A.19}$$

Si tratamos de aplicar el algoritmo anterior a datos no separables, no encontraremos ninguna solución factible, pues la función objetivo crece desmesuradamente. La formulación del problema, en este caso, sería:

Encontrar $w \in \mathbb{R}^n$ y $b \in \mathbb{R}$ y $\xi_i, i = 1, \dots, l$ que minimicen

$$\tau(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \tag{A.20}$$

Sujeto a la condición:

$$w^T \cdot x_i + b \geq -1 - \xi_i \quad y \quad \xi_i > 0 \quad \forall i = 1, \dots, l \tag{A.21}$$

donde C es un parámetro que el clasificador deberá estimar.

Por último, en el caso de la SVM no lineal, se proyectan las variables de entrada en un espacio de dimensión mayor (normalmente de dimensión infinita) que aquel al que pertenecían dichas variables, y se aplica la SVM descrita anteriormente en este nuevo espacio, conocido como espacio de características. De este modo, la SVM no lineal es capaz de clasificar con una probabilidad de error dada por $ent(w, b)$. Esa proyección se realiza utilizando las funciones núcleo (kernel).

La función kernel realiza la separación y traslado de las muestras al espacio

de características. Es como un producto escalar genérico.

Las funciones kernel extienden la clase de funciones de decisión al caso no lineal. Se mapean los datos del espacio de entrada X a un amplio espacio de características X mediante la función Φ y resolviendo el problema de aprendizaje lineal en X :

$$\Phi: X \rightarrow X \tag{A.22}$$

La función real Φ , no necesita ser conocida, es suficiente tener una función de kernel k que calcule el producto interno en el espacio de características.

$$K(x,y) = \Phi(x) \cdot \Phi(y) \tag{A.23}$$

El llamado “problema de selección de características”, esto es, la selección de los factores o rasgos que permitan desechar aquellos elementos que se revelen como irrelevantes para el estudio que se desea realizar, ha resultado ser de especial importancia en la mayoría de los problemas de aprendizaje supervisado.

Entre los distintos modos de tratar la selección de características, el más frecuente es el siguiente: dado un conjunto de datos $(x_1, y_1), \dots, (x_l, y_l)$, con $x \in n$, y $y_i \in \{-1, 1\}$ extraer un subconjunto de m variables ($m < n$) que posean el error de clasificación menor.

Matemáticamente, la función kernel se define:

$$K(x_i, x_j) \equiv [\Phi(x_i) \cdot \Phi(x_j)] \tag{A.24}$$

Las funciones implementadas son:

- Lineal :

$$K(x_i, x_j) = x_i \cdot x_j \tag{A.25}$$

- Sigmoide:

$$K(x_i, x_j) = \tanh(Yx_i \cdot x_j + coef) \tag{A.26}$$

- Polinómica:

$$K(x_i, x_j) = (Yx_i \cdot x_j + coef)^{grado} \tag{A.27}$$

- Gaussian (RBF):

$$K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2) \quad (A.28)$$

A.3.1 Clasificación SVM Multiclase (k clases).

La extensión al problema multiclase en SVM, se puede implementar de maneras distintas:

A.3.1.1. SVM uno vs resto (1 vs R)

Este sistema consiste en realizar comparaciones binarias sucesivas entre cada una de las clases y el resto de ellas. De esta forma, es de suponer que si una de las clases fuera la correcta, sólo esa comparación nos daría un resultado positivo. Esta extensión del problema binario, presenta la ventaja de permitir el resultado de “no perteneciente a ninguna de las clases”, resultado que se produciría si ninguna de las clasificaciones “uno respecto a los demás”, resultara positiva. También existe la posibilidad de que, debido a un error, varias de las comparaciones fueran positivas, y por tanto habría que realizar algún tipo de “desempate” entre las categorías en las que resultó positiva la comparación con las demás.

Con este sistema necesitaremos n máquinas y en cada una de ellas utilizaremos todas nuestras muestras.

A.3.1.2. SVM- DAG (Directed Acyclic Graph)

En este caso, para extender el problema binario, construiremos un grafo de manera que cada nodo del grafo representará una máquina SVM. Cada uno de estos nodos dispondrán de dos ramas, una por cada posible decisión tomada por el SVM. De esta manera, el gráfico completo irá tomando una estructura piramidal.

Con esta estructura, iremos descartando una de las posibles categorías en cada uno de los niveles del gráfico. En este caso utilizaremos $n(n-1)/2$ máquinas, pero en cada una sólo utilizaremos las muestras de dos clases, con lo

que cada una de las máquinas será entrenada más rápidamente que en el caso anterior. Se usa la estrategia de votación para clasificar: cada clasificador binario se considera como un voto y se toma la clase con mayor número de voto.

B. CODIFICACIÓN PARA LA CLASIFICACIÓN DE IMÁGENES

B.1. Librería LIBSVM

B.1.1. Introducción

Para realizar la clasificación de imágenes, se utilizan las librerías LIBSVM, las cuales, proporcionan funciones tanto para el entrenamiento como para la clasificación de ítems teniendo en cuenta ciertas características relevantes.

LIBSVM es una biblioteca de programación destinada a la aplicación de las support vector machines. Inicialmente desarrollada por Chih-Chung Chang y Chih-Jen Lin en 2001, se encuentra actualmente (mayo de 2011) en su versión 3.01.

LIBSVM proporciona al usuario diferentes herramientas para la clasificación y la regresión, e incluye utilidades que permiten la clasificación en múltiples clases. Una tarea de clasificación incluye las siguientes etapas. [20]

- Preparar datos característicos de los ítems a clasificar.
- Separar datos de entrenamiento y testeo.
- Transformar los datos en formato de entrada adecuado para LIBSVM.
- Escalar los datos.
- Probar diferentes kernels (el más habitual RBF).
- Usar validación cruzada para encontrar los parámetros C y γ .
- Utilizar los parámetros óptimos C y γ para entrenar SVM.
- Testear los ejemplos.

Las características que se proponen seleccionar, corresponden a la aplicación de diferentes filtros a la imagen, teniendo así unos valores característicos para los cuales, aplicaremos la clasificación.

B.1.2. Preprocesamiento de la información

Cada instancia de datos, está representada por un vector de números reales, tanto los valores característicos como el valor de la categoría en sí.

B.1.2.1. Escalado

Es importante tener en cuenta este punto. La ventaja de escalar los datos es evitar que los atributos estén en un rango numérico grande, en vez de uno menor, ya que son más complicados los cálculos. Se utiliza el mismo método de escalado para los datos de entrenamiento y los de testeo.

B.1.2.2. Kernel RBF

El kernel realiza la separación y traslado de las muestras al espacio de características \approx producto escalar genérico.

El kernel RBF hace corresponder de un modo no lineal ejemplos en un espacio de dimensión mayor, tiene menos hiperparámetros (C y γ) que el polinomial (menor complejidad de modelo) y menos problemas numéricos

$$0 < K_{ij} \leq 1 \tag{B.1}$$

Los polinomiales están entre 0 e infinito y el sigmoideo no es válido bajo algunos parámetros.

B.1.2.3. Validación cruzada y búsqueda en rejilla

Se trata de buscar los mejores C y γ para clasificar con precisión datos de testeo. Por ello, es conveniente usar validación cruzada sobre los datos de entrenamiento, y así se evita el sobreentrenamiento

B.2. Filtros sobre imágenes

B.2.1. Filtro de la media

Dada una imagen $f(i,j)$, el procedimiento consiste en generar una nueva imagen $g(i,j)$ cuya intensidad para cada píxel se obtiene promediando los valores de

intensidad de los píxeles $f(i,j)$ incluidos en un entorno de vecindad predefinido [21].

B.2.2. Filtro de la mediana

Los píxeles de la nueva imagen se generan calculando la mediana del conjunto de píxeles del entorno de vecindad, del píxel correspondiente a la imagen origen. De esta forma, se homogenizan los píxeles de intensidad muy diferente con respecto a la de los vecinos. Este tipo de filtro es bastante indicado cuando se tiene ruido aleatorio.

La tabla B.2. muestra las ventajas e inconvenientes de esta operación.

Ventajas	Inconvenientes
Atenúa el ruido impulsional (Sal y pimienta).	Pierde detalles (Puntos, líneas finas).
Elimina efectos engañosos.	Redondea las esquinas de los objetos.
Preserva bordes de la imagen.	Desplazamiento de los bordes.

Tabla. B.2. Ventajas e inconvenientes de aplicar filtro de la mediana.

B.2.3. Filtro Gaussiano

El valor de cada punto es el resultado de promediar con distintos pesos los valores vecinos a ambos lados de dicho punto. Este tipo del filtro tiene el problema del difuminado de los bordes, pero no es tan acusado como el caso de la media simple. Como ventaja, tiene que reduce especialmente el ruido tipo gaussiano.

C. APLICACIONES

Las aplicaciones de los clasificadores son muy amplios. Su uso se extiende en medicina (análisis de prueba de drogas, análisis de datos de resonancia magnética), teléfonos móviles (decodificación de la señal, corrección de errores), la visión por computador (reconocimiento facial, seguimiento de objetivos), reconocimiento de voz, minería de datos (análisis de compras en supermercados, análisis de clientes al por menor) entre otras áreas diferentes.

Un ejemplo es un clasificador que acepta datos de sueldos de una persona, edad, estado civil, dirección e historial de crédito y clasifica a la persona como aceptables o inaceptables para recibir una nueva tarjeta de crédito o préstamo.