CREACIÓN DE UN SISTEMA WEB PARA LA INDEXACIÓN, BÚSQUEDA Y ADQUISICIÓN DE VIDEOS MEDIANTE DESCRIPTORES MPEG-7

Proyecto Final de Carrera

Alumna: Mireia Luna Baena

Tutor: Javier Ruiz Hidalgo

Ingeniería Técnica de Telecomunicación,
especialidad Sonido e Imagen
EUETIT (UPC), Junio de 2006

AGRADECIMIENTOS

Como proyectista, me gustaría agradecer ante todo a mi tutor Javier Ruiz Hidalgo su guía y ayuda a lo largo de estos cuatro meses, su paciencia en mis momentos de desesperación y sus rápidas soluciones en los momentos más difíciles.

A mi familia especialmente, por aguantar mi mal humor, por ayudarme y aconsejarme cuando los nervios me ganaban la batalla.

Como no, a mi novio, quien siempre ha estado ahí para animarme y apoyarme cuando más lo necesitaba.

Un agradecimiento especial a mis compañeras, con quienes he compartido momentos de risa, llanto y desesperación.

Sobretodo a Vero, por sus maravillosos consejos, su eficiente descriptor Edge Histogram, y por todas esas tardes encerradas en el laboratorio 101 del TR2.

A Nuria, por enseñarme el “método de la burbuja”.

A Aída y Sandra por su apoyo.

A profesores del departamento de TSC que en momento de duda me han ayudado.

Y a Albert Márquez, el responsable de laboratorio, por instalarme todo lo que necesitaba.

A todos ellos, que a pesar del duro camino, me han ayudado a que mi proyecto final de carrera haya llegado a su fin.

Simplemente gracias.

ÍNDICE

- **INTRODUCCIÓN** 3
- **ESTADO DEL ARTE** 4
- **HERRAMIENTAS** 11
  - Herramientas relacionadas con el estándar MPEG-7 11
  - Herramientas relacionadas con el software 15
- **ARQUITECTURA DEL SISTEMA** 20
  - Arquitectura de la base de datos 20
  - Arquitectura del sistema web a implementar 23
- **ESTRUCTURA DE DIRECTORIOS** 25
- **REALIZACIÓN** 28
  - Diseño de la base de datos 28
  - Diseño del portal web para videos 31
  - Diseño del portal web para descriptores 35
  - Diseño del portal web para formatos 36
  - Diseño del portal web del buscador 37
  - Diseño de la interfaz del usuario 40
- **INTERACTIVIDAD DEL USUARIO** 42
  - Funcionamiento del sistema web 42
    - Inserción y borrado de videos 43
    - Inserción y borrado de descriptores 46
    - Inserción y borrado de formatos 49
    - Acceso a la búsqueda 51
- **REQUISITOS DEL SISTEMA** 55
- **PRUEBAS** 58
- **APLICACIONES** 61
- **CONCLUSIONES** 62
- **REFERENCIAS BIBLIOGRÁFICAS** 63

INTRODUCCIÓN

El proyecto final de carrera (PFC) que se expone a continuación, tiene como objetivo crear un portal o página web que permita al usuario el acceso y la búsqueda de secuencias de video mediante descriptores MPEG-7.

La idea de realizar un proyecto como éste surge del hecho que en la sociedad actual, cada vez hay más cantidad de información y de contenido audiovisual. La razón de todo esto está en el avance desmesurado de las nuevas tecnologías de la comunicación, a la misma vez que se hace un uso masivo de Internet.

Para valorar la importancia de un proyecto como este, un ejemplo claro es que, hoy en día, la faena de un documentalista es recuperar toda la información guardada en cada una de las cintas de video y adquirir de entre todas ellas únicamente unos segundos que hagan referencia a lo que necesita. Para recuperar la información necesaria, debe seguir el mecanismo de introducir cada una de las cintas dentro de un magnetoscopio y mediante las teclas de avance y rebobinado, ir encontrando la escena deseada. Es por ello que cada vez se necesitan más tecnologías que faciliten todo el tratamiento del contenido, por lo tanto, es necesario disponer de un sistema que sea capaz de almacenar toda la información, de tratarla y de permitir un acceso y una búsqueda automática a ésta, a partir de una serie de características.

Es por ello, que el objetivo principal del proyecto, recae en dejar atrás la indexación manual para dar paso a la innovación y eficiencia de la indexación automática, aplicando todos los medios posibles que las nuevas tecnologías proporcionan.

ESTADO DEL ARTE

Antes de la realización de este proyecto ya ha habido otras personas que debido a la problemática descrita en la introducción, han visto la necesidad de crear un sistema web capaz de indexar contenido audiovisual y que al mismo tiempo permita acceder y buscar el contenido.

A continuación, se muestran algunas aplicaciones similares al sistema web creado para el proyecto, bien sea por su indexación del contenido multimedia o por el tratamiento que hacen de éste.

Videoma

Entre estas aplicaciones se encuentra el ejemplo de Videoma, que consiste en una herramienta de ingesta, administración y recuperación de activos digitales, principalmente el video. [1]

Su objetivo es implementar un sistema que facilite a los usuarios los procesos implicados en la gestión de contenido audiovisual, desde la entrada del video, hasta su recuperación habiendo pasado primeramente por el procesado de éste.

La siguiente figura (Figura 1) muestra un esquema del sistema completo:

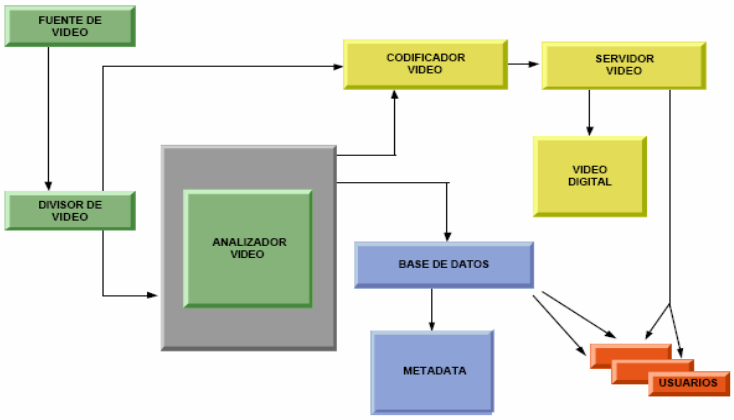

Figura 1. Esquema dels sistema completo de Videoma

Como se puede observar en la Figura 1, el sistema está formado por dos grandes grupos:

El primero, es el bloque llamado núcleo, que toma una de las salidas procedentes del divisor y en tiempo real detecta automáticamente los cambios en la escena del video, los segmenta obteniendo imágenes fijas, creando de esta manera un índice visual del video. Posteriormente, almacena los keyframes (cuadros que han sido tomados como referencia para analizar los cambios posteriores que se proceden en una secuencia de imágenes) o imágenes fijas en la base de datos, al mismo tiempo que pone en funcionamiento los codificadores de video para poder obtener una referencia temporal única que permita acceder y recuperar la información almacenada. Lo que hacen estos codificadores es almacenar los datos en los servidores de video para que el usuario pueda acceder al contenido.

El segundo bloque se encarga de catalogar todos los clips de video generados por el núcleo. Con el fin de permitir la búsqueda y la recuperación del video almacenado en el archivo digital. Tiene como misión acceder a la base de datos donde se encuentran los keyframes junto con los códigos de tiempo. De esta manera, permite al catalogador realizar la tarea de descripción de secuencias mediante la inserción de metadatos (datos que describen otros datos). La catalogación de video, es un proceso que consiste en asociar objetos de información descriptiva (descriptores) llamados metadatos, a cada uno de los keyframes obtenidos en el proceso de segmentación. La catalogación es esencial para permitir posteriormente la recuperación de los contenidos del archivo. Para que esto sea posible, el usuario dispone de una interfaz que le va guiando en el proceso a través de un menú. [1]

Para que el usuario pueda recuperar las secuencias de video, debe de introducir una serie de parámetros que describan la imagen deseada y la aplicación le mostrará aquellos keyframes que correspondan a la búsqueda. En este momento, el usuario puede ver en forma de stream de video la secuencia y decidir cuál es la que más le interesa. En este caso, el usuario también dispone de una interfaz que le va guiando durante la búsqueda, consiguiendo así que sea un sistema todavía más eficiente.

Videoma utiliza las siguientes tecnologías para la realización del sistema [1]:

- Para la representación interna de datos e intercambio de información y contenido, incorpora estándares como **MPEG-7**. Utilizado para la anotación, descripción y categorización de los contenidos audiovisuales.

  El **W3C XML** (eXtensible Mark-up Language), que es un lenguaje que utiliza para la formalización de esquemas de descripción y descriptores.

  El **W3C SMIL** (Synchronized Multimedia Integration Language), que es un lenguaje basado en XML, que se utiliza para la manipulación y posicionamiento de medios audiovisuales en pantalla, así como para la presentación de medios conforme las preferencias del usuario.
- Para la codificación utiliza los estándares Windows Media, RealVideo y QuickTime, también estándares de la ISO como MPEG-1, MPEG-2 y MPEG-4.
- El sistema gestor de bases de datos está basado en la tecnología de base de datos objeto-relacional con un soporte de herencia y poliformismo. Facilita la manipulación de datos, objetos y flujos de información.
- Se desarrolla la aplicación en un entorno web, mediante herramientas de interrogación a base de datos y páginas web.

Videoma utiliza aplicaciones estándar para la programación con PHP y C++, base de datos Oracle 9i y MySQL y servidor web IIS y Apache.

MISE

El laboratorio Murray Hill del grupo de búsqueda de Mitsubishi Electric, ha desarrollado un sistema capaz de indexar y de hacer un sumario de videos basándose en el movimiento de cada escena de la secuencia. A parte de basarse en descriptores de actividad de movimiento, también combina descriptores de color y esquemas de descripción, todos aceptados por el estándar MPEG-7. [2]

El objetivo del sistema es ayudar al usuario a acceder y hacer un sumario de forma rápida y eficaz de todo tipo de contenido de video ya grabado.

A continuación (Figura 2) se muestra la interfaz de este sistema llamado MISE:

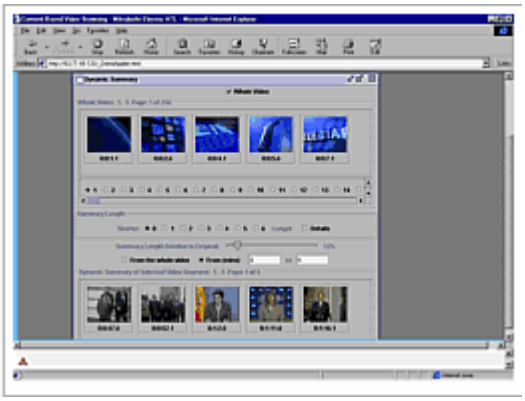

Figura 2. Interfaz del sistema MISE

Este proyecto realizado por Mitsubishi Electric acentúa el uso personal del video como en el caso del PVR (Personal Video Recorder). Permitiendo de esta manera que el usuario pueda almacenar el contenido que él desee y cuando lo desee y proporcionar al mismo tiempo y de manera eficiente un acceso a los datos almacenados utilizando la indexación y la técnica de resumen (summarization).

El sistema está formado por descriptores concretos en el dominio de la compresión, los cuáles se encargan de que tanto la preparación como el acceso al contenido sea lo más rápido posible. Principalmente esta función recae sobre el descriptor de actividad de movimiento del estándar MPEG-7 y también hace uso de los histogramas simples de color.

Actualmente el grupo de laboratorio de Mitsubishi Electric continúa trabajando para mejorar e incrementar el uso del sistema.

MPEG-7 BADED ARCHIVAL OF DIGITAL VIDEO

Entre Noviembre del 2002 y Marzo del 2005, BMWA (German Federal Ministry of Economics and Labour) estuvo trabajando con un proyecto llamado **MPEG-7 based Archival of Digital Video**. El objetivo de este proyecto era la realización de una plataforma que permitiera una gestión completa de una

base de datos audiovisual. Permitiendo segmentar, indexar y recuperar el contenido audiovisual, basándose en los descriptores del estándar MPEG-7. [3]

El proyecto estaba dividido en dos grupos, el primer grupo se encargaba de la parte de imagen y el segundo de la parte de audio.

De la parte de imagen se encargaba el Heinrich-Hert-Institut (HHI). Su faena consistía en dirigir el análisis de la información visual. El encargado de dirigir la estructura general de los métodos de indexación para metadatos del sistema era Canto Software.

En la parte de audio se incluía la segmentación basándose en descriptores de bajo nivel del estándar MPEG-7. También había un reconocimiento y clasificación del sonido ya que el estándar dispone de herramientas de reconocimiento del sonido. Por último, se encontraba la parte de indexación y recuperación de contenido del habla. MPEG-7 también dispone de herramientas de descripción de contenido que permiten una descripción detallada de palabras incluido el habla por teléfono a partir de un stream de audio.

El proyecto dispone de una interfaz web para la búsqueda de contenido audiovisual a través de una indexación automática.

CONVERA/EXCALIBUR RETRIEVAL WARE

Convera/Excalibur RetrievalWare, es otra de las primeras y más avanzadas herramientas de recuperación, indexación y búsqueda de gran cantidad de activos de conocimiento.

Este sistema solamente trabaja con servidores de archivos que guardan 200 tipos de documentos, con sistemas groupware, con bases de datos relacionales y con sistemas de gestión de documentos.

Normalmente todos sus documentos son de texto, a diferencia de los sistemas anteriores o del propio proyecto. Sin embargo, desde Marzo del 2002, Excalibur incluye la posibilidad de realizar una búsqueda de imágenes de videos. Pero todo y ser un sistema que permite la búsqueda de imágenes, únicamente permite realizar la búsqueda utilizando el lenguaje natural, es decir, mediante texto que define un tipo de búsqueda conceptual para permitir al usuario encontrar con más facilidad el contenido que busca. [4]

GOOGLE VIDEO & YAHOO! VIDEO SEARCH

Actualmente en la web, hay disponibles otros buscadores, como son Google y Yahoo, que han desarrollado un sistema para la búsqueda de contendido visual. Lo que diferencia estos sistemas de los sistemas mencionados anteriormente y del proyecto, es que tanto Google como Yahoo, se basan en palabras claves para realizar la búsqueda.

Google anunció a mediados de Enero del 2005 el lanzamiento y de forma experimental en inglés, de una nueva herramienta que permitía la búsqueda de imágenes y videos televisivos a partir de una palabra clave que el usuario tendría que introducir. [5]

Figura 3. Interfaz del buscador Google Video

Yahoo, por su parte, también sacó un nuevo buscador que permitía localizar todos los videos que se pudieran reproducir a partir de programas como son QuickTime, Real Media y Windows Media.

El buscador Yahoo!Search utiliza un etiquetado textual o un sistema de catalogación también de referencia textual para analizar imágenes. Pero de hecho, la iniciativa más interesante de Yahoo es establecer un enlace entre el sistema de catalogación de videos con la sindicación de contenidos RRS (Really Simple Sindication). [6]

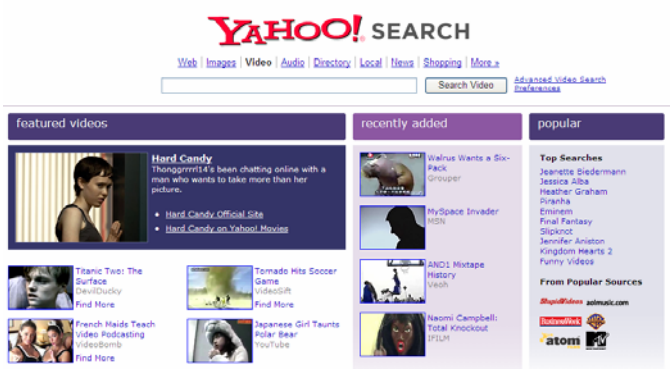

Figura 4. Interfaz del buscador Yahoo! Search

HERRAMIENTAS

Para poder describir todas las herramientas utilizadas a lo largo del proyecto es necesario separarlas en dos bloques.

En un primer bloque se describe todas la parte relacionada con el grupo MPEG y en especial el estándar MPEG-7, el cual es la base de proyecto. En un segundo bloque se hace referencia a la parte de software, como es el servidor local Xampp, el sistema operativo Linux, el servidor web APACHE, los lenguajes de programación PHP y HTML y la base de datos MySQL.

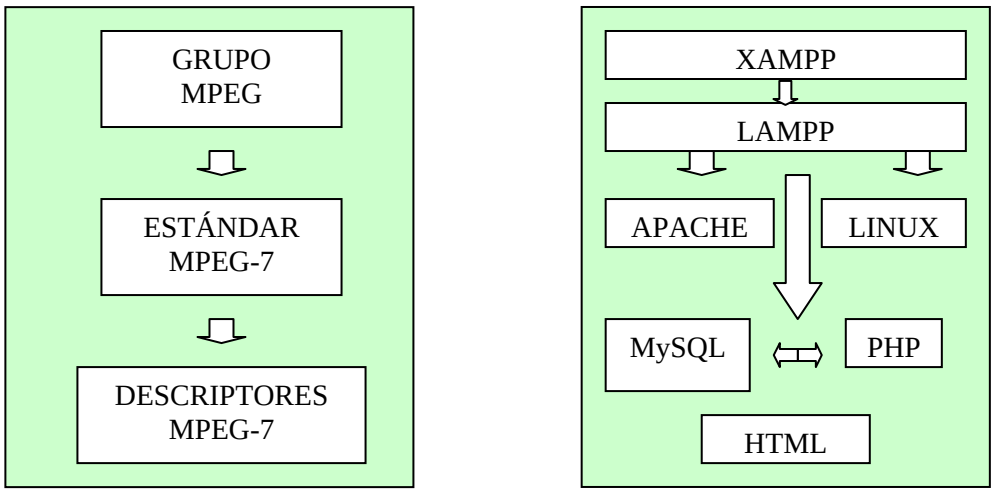

Figura 5. Estructura de las herramientas utilizadas en el proyecto.

Herramientas relacionadas con el estándar MPEG-7

Debido a la necesidad de disponer de un sistema de almacenamiento, de acceso y de búsqueda de información audiovisual, acabó siendo imprescindible la creación de un estándar que permitiese cubrir esta necesidad, al mismo tiempo que permitiese reducir la cantidad de espacio de almacenamiento requerido por el contenido, es por ello, que en 1988 se creó el grupo MPEG (Moving Picture Experts Grup).

MPEG es un grupo de la organización **ISO/IEC** (ISO, International Organization for Standardization / IEC, International Electrotechnical Commission) que está a cargo del desarrollo de estándares internacionales para la compresión, descompresión, procesamiento y representación codificada de

imágenes en movimiento (video), audio, así como la combinación de ambas. [7] Este grupo ha creado otros estándares a parte de MPEG-7:

MPEG-1: es el primer estándar del grupo MPEG y consiste en un formato de compresión de audio y video, su método de compresión se basa en la reutilización de los fragmentos existentes en un cuadro (frame), aprovechando las limitaciones psicológicas y físicas de los seres humanos. [8] Este estándar da lugar a productos como MP3 y VideoCD.

MPEG-2: este estándar dispone de las mismas características que MPEG-1, diferenciándose únicamente en el ancho de banda que utilizan, MPEG-1 se utiliza para anchos de banda medios, en cambio, MPEG-2 se utiliza para anchos de banda altos. MPEG-2 se aplica en TV digital, DVD-Video y SVCD.

MPEG-4: es un estándar diseñado para la codificación de audio y de video en anchos de banda bajos. Resulta una expansión de MPEG-1 para poder soportar "objetos" de audio y video, contenido 3D y soporte para la gestión de derechos digitales. [7]

MPEG-21: tiene como objetivo definir un marco abierto para aplicaciones multimedia. [9]

El **estándar MPEG-7** por su parte, es un estándar que intenta tratar aspectos diferentes a la compresión de la información. Está involucrado en la descripción de características de contenidos multimedia. Su objetivo es describir la manera de ofrecer, filtrar, buscar y tratar información multimedia digitalizada.

MPEG-7 consiste en una representación estándar de la información audiovisual que permite la descripción de contenido multimedia por palabras clave y por significado semántico (quien, que, cuando, donde) y estructural (formas, colores, texturas, movimientos, sonidos). [7] El estándar indica como indexar el contenido audiovisual, como acceder a él, y también identifica derechos de los contenidos y al mismo tiempo permite filtrar la información que le interesa al usuario prescindiendo del resto.

MPEG-7 proporciona por lo tanto toda una serie de herramientas para poder describir contenido multimedia, estas herramientas de descripción (metadatas, descriptores, esquemas de descriptores), sirven para crear

descripciones que serán la base para aplicaciones que permitan el acceso a contenido de este tipo como en el caso del proyecto.

Las partes más importantes del estándar son las siguientes: [10]

- **MPEG-7 Systems**: hace referencia a la arquitectura del estándar y a las herramientas necesarias para preparar las descripciones de MPEG-7 para el transporte y almacenamiento eficiente.
- **MPEG-7 Description Definition Language**: lenguaje para definir nuevos esquemas de descripciones.
- **MPEG-7 Audio**: esta parte proporciona estructuras para describir material sonoro.
- **MPEG-7 Visual**: estructuras básicas y descriptores que definen características de color, textura, forma y movimiento de material audiovisual.
- **MPEG-7 Multimedia Description Schemes**: describen información genérica.

Para entender mejor los objetivos así como la funcionalidad del estándar se observa la siguiente figura (Figura 6):

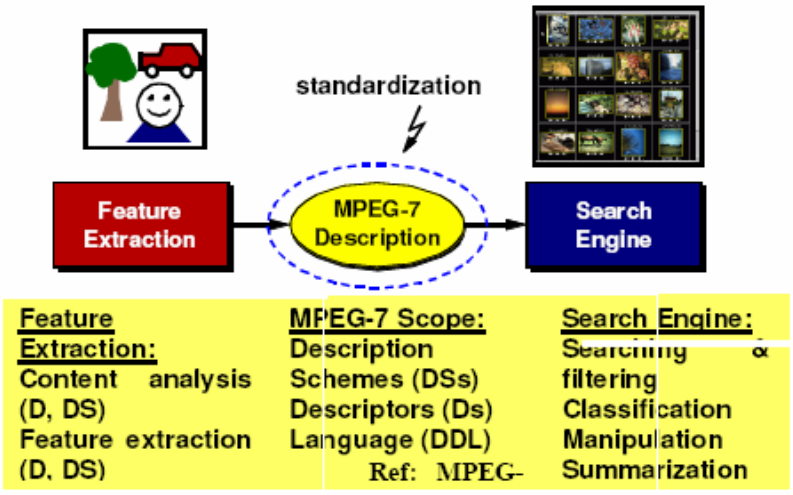

Figura 6. Síntesis de los objetivos del estándar MPEG-7

En un principio se requiere un análisis del documento multimedia para poder obtener sus características, así como la relación entre los elementos. Para que este análisis sea posible, el estándar MPEG-7 dispone de una serie de

descriptores definidos previamente. Una vez ya se ha tratado el contenido y se almacena dentro de la base de datos, el usuario puede o bien buscar y filtrar la información (searching & filtering), es decir, puede seleccionar una imagen o bien un video, como en el caso del proyecto, y proceder al filtrado del contenido. O bien, lo que también puede hacer es, a partir de los descriptores que están clasificados por características, extraer los que se crea adecuados para lo que se quiere hacer. De este modo, es posible obtener la información que más le convenga, al mismo tiempo que puede manipularla (manipulation) si le hace falta o sino hacer un resumen (summarization).

De esta manera, el estándar es capaz de desarrollar potentes buscadores o clasificadores de contenido multimedia.

MPEG-7 utiliza el **esquema XML** (Extensible Markup Language) basado en metadatos como lenguaje para la representación textual del contenido, el cual permite al estándar ser flexible y aumentar las herramientas de descripciones existentes.

Otra herramienta fundamental y que de hecho es la base del proyecto final de carrera, son los descriptores MPEG-7.

Los **descriptores MPEG-7** son una representación de una característica definida de manera sintáctica y semántica.

MPEG-7, dispone de un conjunto de descriptores que son útiles para estandarizar la forma como describir el contenido audiovisual. De descriptores MPEG-7 se pueden encontrar de tres tipos diferentes, los basados en catálogos, donde se puede indicar el título de una obra, el escritor, los derechos de la obra, etc. También hay descriptores semánticos, donde se describe toda información referente a los objetos y acontecimientos que pueden aparecer en un documento, en un vídeo, etc. Y por último, los descriptores estructurales, por ejemplo el histograma de color de un video, de una imagen, etc.

Herramientas relacionadas con el software

Para trabajar con PHP, sin utilizar el servidor de un proveedor externo, lo mejor es instalar un servidor local en el PC.

Un **servidor local** es simplemente un programa, instalado en el ordenador, que permite mostrar las páginas HTML, PHP, etc mediante un navegador. Para la realización del proyecto se usa como servidor local el programa **XAMPP** que al instalarlo dentro del ordenador crea una carpeta llamada **LAMPP** donde están instalados todos los módulos necesarios (Apache, PHP, MySQL, etc).

Lampp es una plataforma que se emplea para definir el trabajo conjunto con Linux, Apache, MySQL y lenguajes como Perl, Python o PHP. Con esta combinación, se pueden crear páginas web dinámicas y obtener información de la base de datos para presentarla vía web. Lampp esta considerada como una de las mejores herramientas disponibles para que cualquier organización o individuo pueda implementar un servidor web versátil y potente. [11]

Algunas de las ventajas que se obtienen de utilizar Lampp son:

- Soporte a gran cantidad de arquitecturas, como Intel, SPARC, Mips y PPC (Macintosh).
- Código sencillo y con pocos cambios de una plataforma a otra.
- Parches generados en poco tiempo después de encontrarse un agujero de seguridad.
- Actualizaciones del software vía Internet.
- Posibilidad de incrementar los servicios y funciones desde el código fuente.

Pero a pesar de las ventajas que presenta Lampp, también tiene desventajas que deben considerarse:

- Es muy distinto a Windows, por lo que dificulta el trabajo a aquellos que estén acostumbrados a este sistema operativo.
- Las actualizaciones requieren en ocasiones tener conocimientos profundos del sistema.
- Configurar algunos servicios de red requiere de más tiempo que en Windows.

- Mayor coste del personal.

Los cuatro elementos que forman Lampp son al igual que la plataforma, de software libre por lo que disfrutan de ventajas como son:

- Libertad de copia y distribución.
- Libertad de modificación.

Lampp trabaja con el sistema operativo Linux.

LINUX es un software libre de código, es decir, está disponible públicamente y cualquier persona con un mínimo de conocimientos informáticos adecuados, puede libremente utilizarlo, modificarlo y redistribuirlo. [12] El software está basado en los estándares Unix, y surgió a principios de los 90 gracias a Linus Torvalds para mejorar y ampliar Minix (otra implementación gratuita de Unix por Andy Tanenbaum, dirigida al ámbito educativo). Linux viene arropado por librerías y utilidades distribuidas bajo la licencia libre GPL o similares.

Este sistema operativo presenta infinidad de ventajas respecto a otros sistemas:

Dispone de una excelente relación calidad-precio y en el ámbito de los servidores web es calificado de excelente. Linux, es multitarea, multiusuario, multiplataforma, multiprocesador, tiene protección de la memoria entre procesos, soporta muchos tipos de sistemas de archivos, dispone de una amplia variedad de protocolos de red soportados en el núcleo y, finalmente permite compartir por red ficheros e impresoras, incluso con otros sistemas operativos. [12] Su potencia, estabilidad, gratuidad, modificalidad y portabilidad hacen que Linux acabe siendo unos de los mejores sistemas operativos.

Como servidor web se utiliza **Apache**. Un **servidor web** es un programa que permite que otros ordenadores vean una web creada mediante un navegador. Sabiendo que el proyecto consiste en la creación de un sistema web, se entiende el porqué del requerimiento del servidor Apache. El **servidor APACHE**, es el servicio que se encarga de resolver las peticiones de páginas web de los clientes utilizando el protocolo de Internet HTTP. El servidor se basa en NCSA server, y surgió a partir de ampliaciones y parches.

Un grupo de administradores web pusieron en marcha una lista de correo y fundaron el Apache Group, que acabó convirtiendo a Apache en el número 1 en la lista de Netcraft. [11]

Apache, es el servidor web por excelencia, con más de un 60% de los servidores de Internet que confían en él. Entre sus características cabe destacar las siguientes:

- Fiabilidad
- Gratuidad: es totalmente gratuito y se distribuye bajo la licencia Apache Software License, que permite la modificación del código.
- Extensibilidad: permite añadir módulos para ampliar sus capacidades, que pueden ser creados por cualquier persona con conocimientos de programación.

La base de datos que se utiliza en el proyecto es **MySQL** (My Structured Query Language). Esta base de datos procede del lenguaje de consulta estructurado SQL (Structured Query Language) que es un lenguaje declarativo de acceso a base de datos relacionales que permiten especificar diferentes tipos de operaciones sobre ellas. [13]

MySQL es un servidor de base de datos relacionales muy rápido y robusto. Es software libre, publicado bajo la licencia GLP (GNU Public License) y mantenido por la compañía sueca MySQL AB.

A pesar de que MySQL no dispone de muchas de las características de los gestores comerciales más importantes como son Oracle, Sybase o SQL server. MySQL es el más indicado para aplicaciones que requieren mucha lectura y poca escritura y que no necesiten de características muy avanzadas, como es el caso de las aplicaciones web. Siendo así, la mejor base de datos para cumplir con los objetivos del proyecto.

Como lenguaje de alto nivel se utiliza **PHP** (Hypertext Preprocessor inicialmente PHP Tools, o, Personal Home Page Tools). **PHP** es un lenguaje interpretado utilizado por la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Últimamente también se utiliza para la creación de programas incluyendo aplicaciones con interfaz gráfica. [14] PHP es

un lenguaje orientado a conexiones entre páginas web y servidores donde se almacenan toda clase de base de datos. [15]

En el ámbito del proyecto, las capacidades de HTML, Javascript y demás tecnologías de cliente son bastante reducidas. Una página realmente profesional no puede limitarse a mostrar información y disponer de formularios para conectarse con los usuarios. También requiere que los navegadores sean capaces de comunicarse con programas alojados en el servidor como es MySQL.

PHP dispone de entre las siguientes características la capacidad de conectarse con otros programas respecto a otros lenguajes:

- Es un software libre, lo que implica menor coste y servidores más baratos. A la vez que el tiempo entre el hallazgo de un fallo y la resolución es más corto.
- Gran volumen de código PHP libre, que permite construir sitios web más interesantes.
- Es muy rápido, incluso en su integración con la base de datos MySQL.
- Soporta gran cantidad de base de datos InterBase, mSQL, MySQL, Oracle, Informix, PosgreSQL, entre otras.
- Sintaxis inspirada en C, que permite a las personas que conocen el lenguaje una fácil familiarización con PHP.
- Dispone de una librería estándar amplia.
- Es una multiplataforma, es decir, funciona en cualquier máquina que sea capaz de compilar su código. El código se ejecuta siempre en el servidor.
- Es muy versátil y potente por lo que sirve para aplicaciones grandes como para pequeños scripts.

En cuanto a desventajas hay que destacar que en España hay pocos especialistas en PHP y por otra banda, como lenguaje, PHP no soporta polimorfismo ni tiene excepciones u otro sistema de errores aceptable.

Otro lenguaje de programación que se utiliza en el proyecto es **HTML** (**H**yper**t**ext **M**arkup **L**anguage). Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los

navegadores del tipo Internet Explorer, Opera, Firefox o Netscape, HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos. HTML es una aplicación de SGML conforme al estándar internacional ISO 8879. [16]

Como que el software que se usó para la realización del proyecto es libre, todo el código referente a las páginas PHP que se implementaron son del mismo modo libre. De esta manera, todo aquel que quiera consultar o acceder a él, puede hacerlo gratuitamente y públicamente.

ARQUITECTURA DEL SISTEMA

Antes de conocer como se realizó el proyecto, es necesario conocer su arquitectura para poder entender el procedimiento. Primeramente se verá la base de datos que forma el proyecto y por último se describirá el sistema web que se implementó.

Arquitectura de la base de datos del sistema

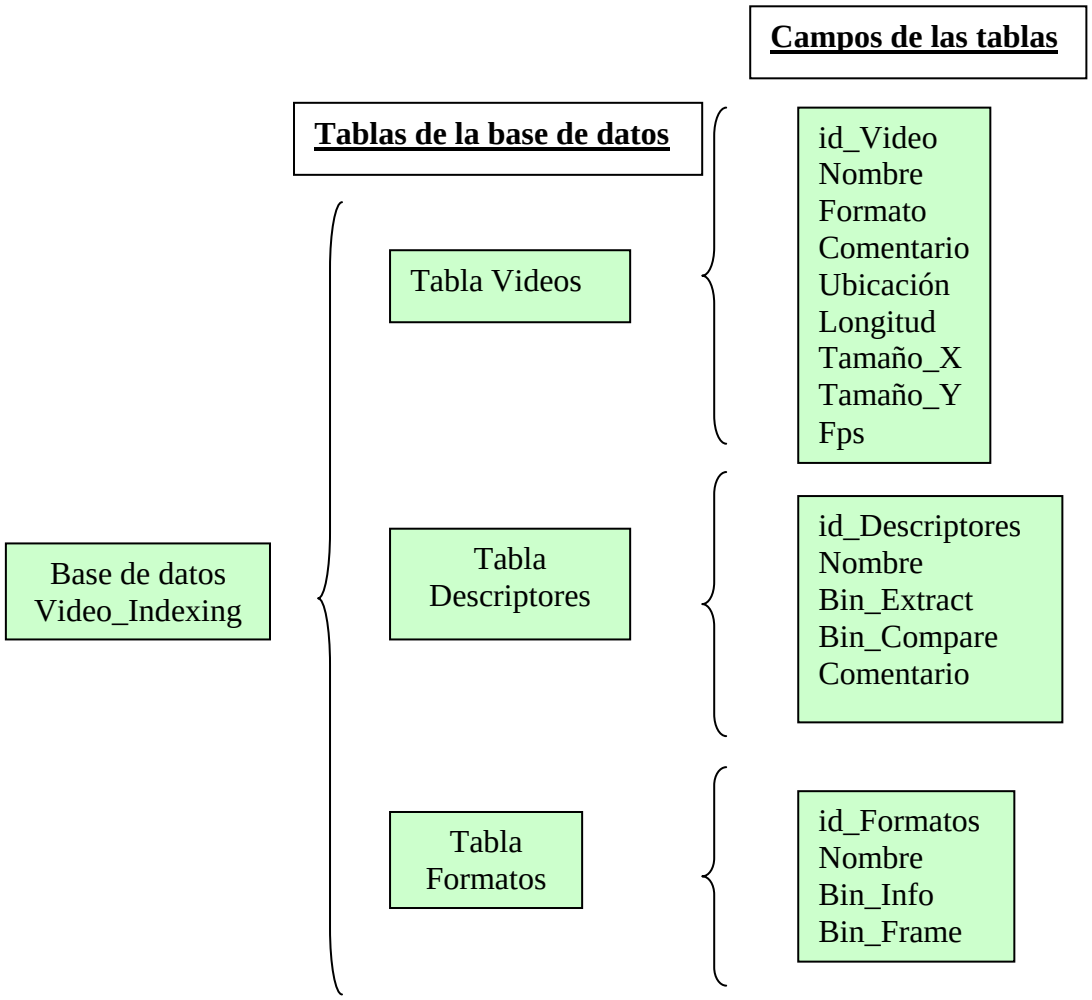

Figura 7. Arquitectura de la base de datos del sistema.

El proyecto lo forma una base de datos creada mediante MySQL que recibe el nombre de *Video_Indexing*. La base de datos se caracteriza por disponer de tres tablas diferentes, la primera tabla es la tabla de **Videos**, que está formada por los campos siguientes:

- **Nombre**: consiste en una cadena de caracteres del tipo blob, que es un tipo destinado a almacenar texto más largo de 255 caracteres, donde se diferencia entre mayúsculas y minúsculas.
  Dentro de este campo se indica el nombre de cada video que se introduce en la base de datos.
- **Formato**: el campo formato es también una cadena de caracteres, pero esta vez el tipo es varchar, que permite almacenar la cadena con la longitud que ocupa ésta. La longitud máxima de este tipo es de 255 caracteres.
  Este campo se caracteriza por estar relacionado con otra tabla que se verá a continuación (tabla Formatos), donde el usuario puede seleccionar mediante un menú desplegable el formato del video que quiere introducir a partir de los formatos disponibles en la base de datos.
- **Comentario**: este campo se utiliza para que el usuario pueda introducir cualquier aspecto a destacar o cualquier comentario que crea conveniente referente al video.
  El tipo de este campo es de nuevo blob, se creyó conveniente utilizar este tipo porque se desconocía qué longitud aproximada tendría el campo, por lo que se requería un espacio bastante largo para evitar problemas.
- **Ubicación**: el campo ubicación se necesita también dentro de la tabla Videos porque indica la ruta completa donde está ubicado el video dentro del ordenador, es imprescindible conocer este campo para el correcto funcionamiento del sistema.
  Su tipo es también del tipo blob porque al igual que en el campo ubicación se desconocía la longitud que podría tener el campo.
- **Longitud, Tamaño_X, Tamaño_Y y Fps**: estos cuatro campos son todos del tipo entero menos el campo referente a los frames por segundo (Fps) del video, que es del tipo float.

El campo Longitud indica la longitud total del video y Tamaño_X y Tamaño_Y indican la longitud X y Y del video respectivamente.

No es necesario que el usuario conozca estos campos puesto que el sistema dispone de un programa específico para cada formato que se encarga automáticamente de hacer los cálculos.

Una segunda tabla es la tabla **Descriptores**, que como se ha comentado con anterioridad, los descriptores MPEG-7 son la base del proyecto y permiten caracterizar a cada video según una característica concreta.

Antes de describir los campos de la tabla de descriptores, es necesario conocer que cada descriptor requiere de dos binarios, el binario de extracción y el binario de comparación. El binario de extracción se encarga a partir de un video seleccionado por el usuario crear un fichero con el valor del descriptor para ese video. El binario de comparación, por su parte, se encarga a partir de dos ficheros de extracción creados por el binario anterior, compararlos y hacer el cálculo del valor de comparación entre ambos, que tendrá un valor entre 0 y 1, donde el valor 0 indica que los videos no se parecen en nada y el valor 1 indica que los videos son iguales.

La tabla Descriptores está formada por cuatro campos:

- **Nombre**: este campo es del tipo blob y hace referencia al nombre del descriptor.
- **Bin_Extract**: el campo Bin_Extract consiste en una cadena de caracteres del tipo blob que indica la ruta completa donde está ubicado el binario de extracción del descriptor junto con el nombre del propio binario.
- **Bin_Compare**: consiste al igual que Bin_Extract, en una cadena de caracteres del tipo blob que indica esta vez, la ruta completa del binario de comparación.

Conocer las rutas de los dos binarios permite poder ejecutarlos cuando sea necesario.

- **Comentario**: al igual que en la tabla Videos, este campo, del tipo blob, hace referencia a cualquier comentario que el usuario desee hacer sobre el descriptor.

Y por último, la tercera tabla es la tabla **Formatos**, donde se recogen todos los formatos disponibles para cada video de la base de datos.

Esta tabla únicamente dispone de tres campos:

- **Nombre**: referente a un string del tipo blob que recoge el nombre del formato.
- **Bin_Info**: que hace referencia a la ruta completa donde se sitúa el binario de información de cada formato. El binario de información es aquel que se encarga de calcular de forma automática los valores de Longitud, Tamaño_X, Tamaño_Y y Fps del video que se introduce.

  El campo una cadena de caracteres del tipo blob.
- **Bin_Frame**: este campo pertenece a la ubicación del binario de extracción de frame . El binario de extracción de frame se utiliza para poder extraer un cuadro de una secuencia de video en función del formato que utilice el video, es decir, para cada formato hay un Bin_Frame diferente, lo mismo pasa para el binario de información visto anteriormente.

  Al igual que Bin_Info, este campo también es del tipo blob.

A parte de todos los campos de los que disponen cada tabla, cada una de ellas tiene al mismo tiempo, un campo llamado *id (id_Video, id_Descriptores, id_Formatos*). Este campo se caracteriza por ser un entero que se autoincrementa cada vez que se añade un nuevo registro dentro de una tabla. Cada *id* de cada tabla es independiente a las demás. Y cada *id* de cada registro, por ejemplo la i*d* de los videos, es única y exclusiva para cada video. Esto permite identificar cada registro de forma única.

Arquitectura del sistema web a implementar

El sistema web está formado por una página principal que es la página de acceso al portal del proyecto y desde la cual se puede acceder a las diferentes páginas implementadas, entre las que se encuentra la página del buscador, donde se muestran las tablas Videos y Descriptores y por la cual se puede acceder a la búsqueda de los videos a partir de la selección de un video y de un

descriptor. Desde la página principal, también se puede acceder a la página de videos, donde se muestra la tabla de videos disponible en la base de datos, y desde la cual se puede insertar y borrar el video o los videos deseados. Las páginas de descriptores y formatos muestran las respectivas tablas, y al igual que en la página de videos, el usuario puede insertar y borrar descriptores y formatos. Otro enlace existente en la página principal, es el enlace “ACERCA DE...” donde se muestra un escrito referente al proyecto con el objetivo de informar al usuario.

ESTRUCTURA DE DIRECTORIOS

A continuación se puede observar un esquema que indica la estructura del directorio usado para el proyecto:

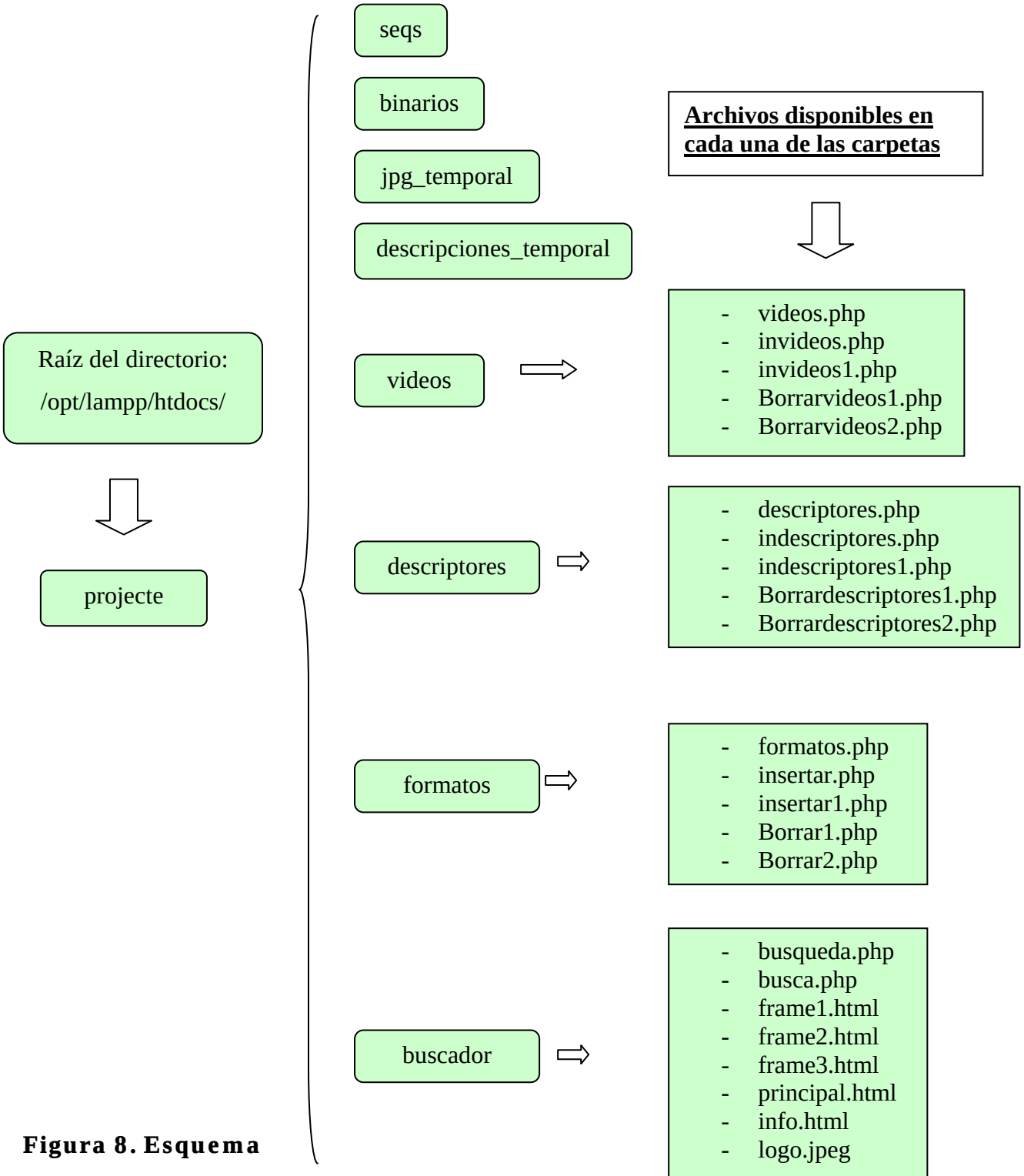

Figura 8. Esquema de la estructura del directorio.

Dentro del directorio */opt/lampp/htdocs/* (como información, saber que Xampp utiliza como directorio raíz el directorio */opt/lampp/htdocs/*), hay una carpeta llamada "projecte" donde se encuentran todos los archivos relacionados con el proyecto. La carpeta "**projecte**" está formada por distintas subcarpetas, donde cabe destacar tres de principales que son las llamadas **videos**, **descriptores** y **formatos**, donde se encuentran almacenados todos los scripts PHP pertenecientes a las páginas web de videos, de descriptores y de formatos respectivamente.

Aparte de estas tres, se encuentran otras:

- Subcarpeta "**seqs**": dentro de *seqs* se encuentran almacenadas todas las secuencias de videos que están disponibles en la base de datos. El usuario cada vez que desee introducir una nueva secuencia, debe de dirigirse al directorio */opt/lampp/htdocs/projecte/seqs* y copiar ahí el video, de este modo, siempre que se necesite acceder a él, el sistema web sabrá donde encontrarlo.
- Subcarpeta "**binarios**": dentro de *binarios* están disponibles los binarios referentes a los descriptores, binarios de extracción y de comparación de cada descriptor. También se encuentran los binarios de información de cada formato así como los binarios que extraen una imagen de la secuencia, que también son únicos para cada formato. Por lo tanto antes de introducir un nuevo descriptor, el usuario debe de copiar en esta subcarpeta los binarios correspondientes, lo mismo pasa al introducir un nuevo formato dentro de la base de datos.
- Subcarpeta "**descripciones_temporal**": en esta subcarpeta se almacena de manera automática todos aquellos ficheros creados por el sistema web. Estos ficheros se crean en el momento en que se inserta o bien un video o bien un descriptor dentro de la base de datos. Hacen referencia a los ficheros que almacenan el valor de extracción de un video, así como el valor de comparación entre dos videos y que indica la similitud entre ambos.

Esta subcarpeta no debe de ser modificada por el usuario, ya que una modificación de ésta, podría provocar que alguna de las búsquedas no diera el resultado correcto.

- Subcarpeta "**jpg_temporal**": al igual que la subcarpeta "*descriptores_temporal*", "*jpg_temporal*" se creó para almacenar de forma automática por el sistema, todas la imágenes que se extraen al ejecutar el binario de extracción de frame de cada formato, por lo tanto, también se requiere que no haya ninguna modificación de la misma por parte del usuario.
- Subcarpeta "**buscador**": dentro de *buscador* se pueden encontrar las páginas relacionadas con la búsqueda (busqueda.php y busca.php) así como ficheros HTML, donde se incluye el código de la página principal del buscador (principal.html). También se encuentran todos los archivos que definen las ventanas (frames, en lenguaje HTML) en las que se estructura principal.html. La página principal del buscador, es la página de acceso inicial al portal web implementado en el proyecto. Por lo tanto, al encontrarse aquí la página principal, cada vez que se quiera acceder al sistema web del proyecto, se debe escribir en la barra del navegador la dirección:
  http://localhost/projecte/buscador/principal.html
  En el caso que se acceda desde una máquina externa al servidor local, se debe de cambiar localhost por la IP del ordenador donde está ubicado el sistema web.
  Dentro de esta misma carpeta, se encuentra también una página HTML (info.html), que hace referencia al apartado de "ACERCA DE...".

REALIZACIÓN

En este apartado se pretende hacer una descripción exacta de cada uno de los pasos necesarios que se fueron siguiendo para conseguir implementar un sistema web como el creado para el proyecto.

Diseño de la base de datos

El primer paso que se debió hacer fue diseñar la base de datos. En este caso la base de datos Video_Indexing.

Antes de crear la base de datos fue necesario crear un usuario. El nuevo usuario se creó para que pueda acceder a la base de datos y de esta manera poder manejarla. Para ello, se tuvo que dar permisos totales al usuario sobre la base de datos, por lo que se debió conocer su nombre de usuario y contraseña. A continuación se puede ver como se creó el usuario que permite conectarse con la base de datos del proyecto. Estos pasos son necesarios seguirlos estrictamente para crear la base de datos de Video_Indexing en cualquier máquina que no esté conectada en red con la máquina donde esté implementado el proyecto.

Antes de nada se debe acceder al servidor MySQL desde la línea de comandos de Linux (Terminal). Una vez se accede al terminal, se debe conocer la ubicación del servidor dentro del disco duro. En el caso de Xampp, el servidor MySQL se encuentra dentro del directorio */opt/lampp/bin* por lo que para acceder a él se necesita escribir el siguiente comando:

```
cd /opt/lampp/bin
./mysql
```

Una vez ejecutadas las instrucciones anteriores se accede al prompt de MySQL "mysql>". A partir de ahí ya se puede trabajar con el servidor MySQL.

Para crear un nuevo usuario se accede a la base de datos de mysql, esto se consigue con la siguiente sentencia:

```
mysql> show databases;
```

A continuación, se indica la base de datos con la que se va a trabajar, que en este caso será la base de datos llamada mysql, para indicarlo, se ejecuta *use mysql*:

```
mysql> use mysql;
```

Se muestran las tablas que hay dentro de la base de datos mysql:

mysql> show tables;

Al ejecutar el comando anterior se observa una tabla llamada *user*, al hacer "*select * from user;"* se observa toda la tabla de usuarios aceptados en el sistema.

El campo password, puede estar cifrado usando la función password() o bien, en texto plano.

En este momento, se introduce el nuevo usuario, en el caso del proyecto, el nombre de usuario es "*Mire*" y de password "*PFC*".

Para que en la tabla, la contraseña salga cifrada y de esta manera asegurar que nadie pueda modificar la base de datos sin saber el password, se hace con password ().

mysql> insert into user (Host, User, Password) values ('IP del ordenador', 'Mire', password ('PFC'));

Si se observa la tabla de nuevo, se puede observar el nuevo usuario, pero no se podrá usar hasta que se reinicie el servicio.

Seguidamente se procede a crear la base de datos *Video_Indexing*:

mysql>create database Video_Indexing;

En este momento es necesario dar permisos totales al usuario sobre las tablas de la base de datos que se crearán a continuación.

mysql> GRANT ALL ON Video_Indexing.* TO Mire@localhost IDENTIFIED BY 'PFC';

mysql>flush privileges;

mysql>quit;

Ya se tiene el usuario *Mire* creado para poder manejar la base de datos *Video_Indexing*. Recordar de reiniciar el servicio con MySQL.

Ahora ya se puede empezar a crear las tablas que forman parte de la base de datos. Primeramente se crea la tabla Formatos:

mysql>use Video_Indexing;

mysql> create table Formatos(

➔ id_Formatos int (6) NOT NULL AUTO_INCREMENT,

- Nombre BLOB NOT NULL,
- Bin_Info BLOB NOT NULL,
- Bin_Frame BLOB NOT NULL,
- PRIMARY KEY (id_Formatos));

A continuación la tabla Descriptores:

mysql> create table Descriptores(

- id_Descriptores int (6) NOT NULL AUTO_INCREMENT,
- Nombre BLOB NOT NULL,
- Bin_Extract BLOB NOT NULL,
- Bin_Compare BLOB NOT NULL,
- Comentario BLOB NOT NULL,
- PRIMARY KEY (id_Descriptores));

Y por último se crea la tabla Videos:

mysql> create table Videos(

- id_Video int (6) NOT NULL AUTO_INCREMENT,
- Nombre BLOB NOT NULL,
- Ubicacion BLOB NOT NULL,
- Formato VARCHAR (100) NOT NULL,
- Comentario BLOB NOT NULL,
- Longitud int (10) NOT NULL,
- Tamano_X int (10) NOT NULL,
- Tamano_Y int (10) NOT NULL,
- Fps float NOT NULL,
- PRIMARY KEY (id_Video));

Si ahora se ejecuta el comando " *show tables;* " se pueden visualizar las tablas que forman la base de datos de Video_Indexing;

Cada vez que se conecte al servidor MySQL desde la línea de comandos de Linux y se quiera trabajar con la base de datos del proyecto, es necesario escribir la sentencia "*use Video_Indexing;*"

Una vez creada la base de datos, ya se puede empezar a trabajar con ella.

Mediante PHP se accede de forma fácil a *Video_Indexing* para lanzar consultas del tipo insertar datos, borrar, listar, etc. que serán muy útiles para el correcto funcionamiento del sistema web a implementar.

Un dato importante a destacar es saber que cuando se cambie de máquina, siempre que ésta no esté conectada en red con el ordenador donde está implementado el sistema, el usuario deberá crear de nuevo la base de datos de Video_Indexing, así como el usuario "Mire" con password "PFC" porque si se crea un usuario diferente será necesario modificar el código de las páginas en PHP donde las variables "Mire" y "PFC" son variables fijas.

Diseño del portal web para videos

Para crear un script PHP se puede hacer desde cualquier editor de texto, simplemente se requiere que el script esté guardado dentro de un directorio concreto que en el caso del proyecto es */opt/lampp/htdocs/*, y a la misma vez el fichero debe disponer de una extensión del tipo nombre_fichero.php. De esta manera únicamente ejecutando la sentencia "http://localhost/nombre_fichero.php" desde la URL de cualquier navegador se puede acceder a visualizar el script creado en formato de página web. Escribiendo la sentencia anterior se llega a acceder al servidor APACHE de la máquina.

Antes de crear el portal web para videos fue necesario diseñar la estructura que se quería implementar. En la página de videos, simplemente se muestra la tabla de videos y se tiene la opción de insertar y borrar videos de la tabla.

Siempre que se requiere que un script PHP acceda a la base de datos MySQL es necesario conectarse a ella, para poder hacerlo, MySQL dispone de una función propia que lo permite, donde únicamente hay que indicarle la máquina desde la que se quiere establecer la conexión, el nombre de usuario y la contraseña, y por supuesto indicarle también el nombre de la base de datos a la que se quiere acceder. En el caso del proyecto, la base de datos es *Video_Indexing*.

Para poder obtener los datos de la tabla de videos y mostrarlos por pantalla, se construyó en el script una tabla en HTML que sirve para estructurar los datos en forma de tabla, ya que los datos que PHP recoge de la base de datos se obtienen en forma de vector. A medida que se va llenando la tabla creada con los datos de los videos disponibles, y cada vez que acaba de completarse una fila, se ejecuta el binario de extracción de un frame o cuadro de la secuencia que se acabe de añadir.

La manera de ejecutar este binario, es mediante una sentencia PHP *shell_exec()* que se encarga de ejecutar una instrucción como si se hiciera desde la propia consola de comandos de Linux. Es por lo tanto una función diseñada para ejecutar comandos del sistema Unix. Para ejecutar el binario simplemente se necesita conocer la ubicación completa del binario, que se obtiene a partir de una consulta a la base de datos, conociendo previamente el formato que utiliza el video que se introduce, ya que el binario de extracción de una imagen es único para cada formato.

Otro parámetro necesario en la ejecución del binario, es la ubicación de la secuencia de video, para poder acceder a ella y hacer la extracción de la imagen. Como último parámetro es necesario la ubicación donde se va a guardar la imagen. Como ya se ha comentado en el apartado de "Estructura de directorios" todos los cuadros extraídos de la secuencia, se almacenan dentro de una carpeta llamada "*jpg_temporal*". En la ubicación también se indica el nombre que se le va a dar a la imagen. En todos los casos, es: id_Video.jpg, de esta manera se sabe qué frame se corresponde con cada video.

Para poder entender claramente esta ejecución véase un ejemplo:

shell_exec = (Ubicación del binario de extracción de frame . ' ' . Ubicación de la secuencia de video . ' ' . Ubicación del frame extraído);

Una vez se consigue extraer todos los datos referentes a la tabla de videos de la base de datos y colocarlos dentro de la tabla HTML se implementan las funciones de inserción y borrado de videos. Para que el usuario pueda insertar un video desde el sistema web se creó un enlace desde la página principal de videos a la página donde se inserta el video, para crear esta nueva página se requiso otro script PHP *(invideos1.php)*. Este nuevo script tiene dos partes bien diferenciadas, una parte dedicada a HTML que se usa para crear las ranuras

necesarias para que el usuario pueda introducir la información necesaria para la inserción del video. Y una parte dedicada a PHP, ya que para introducir el formato que utiliza el video se selecciona de un menú desplegable donde se exponen todos los formatos disponibles en la base de datos, por lo tanto, se utiliza PHP para poder acceder a la base de datos y efectuar la consulta apropiada para conocer el listado de formatos existentes.

Una vez el usuario ha llenado toda la información necesaria sobre el video, estos datos son recogidos por el sistema y los almacena en la base de datos mediante otro script en PHP llamado *invideos.php*.

No se debe olvidar que la tabla Videos dispone de otros campos (*Longitud, Tamaño_X, Tamaño_Y, Fps*) que son calculados e insertados en la base de datos automáticamente por el sistema. Para que esto sea posible se requiere ejecutar el binario de información asociado al formato del video. Al igual que se utiliza la función PHP *shell_exec()* para ejecutar el binario de extracción de frame, también se utiliza para ejecutar el binario de información. Para su correcta ejecución, únicamente se precisan los parámetros que hacen referencia a la ubicación del binario de información y a la ubicación de la secuencia de video.

La ubicación del binario se obtiene haciendo una consulta SQL desde PHP a la tabla Formatos, y la ubicación del video, se extrae del campo ubicación que el usuario introduce en la inserción del video. Al ejecutar el binario de información, la función devuelve una cadena de caracteres con los valores correspondientes a la *Longitud*, al *Tamaño_X*, *Tamaño_Y* y *Fps* del video, de modo que para tratarlos de forma independiente se necesita separarlos.

Para poder hacer la separación, PHP dispone de una función llamada *split()* que se encarga de dividir la cadena en elementos de un array, y que juntamente con la función *list()* obtiene los valores de forma independiente. Una vez se extraen los valores, ya se puede insertar todos los datos del video dentro de la tabla *Videos*, esto es posible mediante la ejecución de la sentencia SQL *mysql_query()*.

Dentro de este mismo script se consideró la necesidad de obtener el fichero de extracción de descriptor cada vez que se introduce un vídeo. Para obtener el fichero de extracción se necesita ejecutar el binario de extracción, de la misma manera que se hace con los demás binarios, se utiliza *shell_exec()*.

Esta vez los parámetros de entrada de la función son la ruta completa del binario, que es diferente dependiendo del descriptor, la ubicación del video y la ubicación donde se guardará el fichero junto con el nombre que se le desea dar.

El fichero se guarda como se ha comentado, en el directorio */opt/lampp/htdocs/projecte/descripciones_temporal*. Cada uno de los ficheros que se crean tiene como nombre la *id* del video y la *id* del descriptor. Este fichero se crea dentro de este script cada vez que se introduce un video, el binario de extracción se ejecuta tantas veces como descriptores haya en la base de datos, ya que cada video tendrá tantos ficheros como descriptores haya.

Un aspecto a tener en cuenta es que, para evitar problemas de funcionamiento, se consideró la opción de mostrar un mensaje de error cuando el usuario intente introducir un video con el mismo nombre que otro video que exista en la base de datos. Para que el sistema pueda detectar el error, debe comparar el nombre que le introducen con todos los otros nombres de los videos existentes en la tabla de videos. Si al hacer el recorrido detecta que el nombre introducido es igual a otro nombre de un video de la base de datos, se para el procedimiento habitual de insertado y se muestra el mensaje de error por pantalla. Junto al mensaje de error se muestra un enlace que lleva al usuario de nuevo a la página de insertado. Pero si por lo contrario, al hacer el recorrido, no detecta ningún nombre igual al introducido, se sigue con el insertado normal y se muestra por pantalla que el video se ha insertado correctamente.

Al igual que el usuario tiene la posibilidad de insertar un video, también tiene la opción de borrarlo. Para ello se necesita de un script que permita borrar registros. Para poder borrar, el usuario debe de seleccionar de un menú desplegable el nombre del video a borrar, por lo que el script necesita acceder a la base de datos para saber el nombre de todos los videos disponibles y de esta manera construir el menú. También dispone de un botón que al hacer clic sobre él, borra el video, este botón está direccionado a otro script que se encarga de realizar el borrado mediante una sentencia SQL.

Diseño del portal web para descriptores

En la página principal de los descriptores únicamente se muestra la tabla *Descriptores* y dos enlaces necesarios para la inserción y el borrado de los registros.

Para poder visualizar la tabla de descriptores al igual que se ha hecho con la tabla de videos en el portal de videos, se necesitó crear una tabla en HTML para introducir los valores que se obtienen de la consulta a la tabla *Descriptores*. Para ejecutar la consulta, primeramente el script debe de conectarse a la base de datos y seleccionar mediante la sentencia SQL *mysql_query()* la tabla *Descriptores*. Una vez se obtienen los registros de la consulta se van introduciendo dentro de la tabla creada.

Los enlaces a las páginas de inserción y de borrado se efectúan mediante código HTML.

Para que el usuario pueda insertar un nuevo descriptor se precisan dos scripts, el primer script *(indescriptores1.php)* se encarga de crear las cajas de introducción de texto para que el usuario pueda llenar todos los campos de la tabla con la información necesaria. Y el segundo script inserta dentro de la base de datos los valores introducidos por el usuario. Para hacerlo ejecuta la sentencia SQL que permite insertar datos en una tabla. Al igual que pasa con los videos, en este script se requiso también la necesidad de crear el binario de extracción para cada video. Por lo tanto al introducir un nuevo descriptor, el sistema se encarga a partir del binario de extracción, crear para cada video el fichero de descripción.

Para poder crear el fichero de descripción se necesita conocer la ubicación del binario de extracción del descriptor insertado, la ubicación de todos los videos de la base de datos y la ubicación que tendrán todos los ficheros que se creen. Al igual que en videos, el nombre del fichero es la *id* del video junto con la *id* del descriptor y se guardará una vez creado, dentro del directorio */opt/lampp/htdocs/projecte/descripciones_temporal/*. Una vez obtenidos todos estos parámetros, mediante consultas SQL a la base de datos, se ejecuta el binario de extracción del nuevo descriptor para cada uno de los videos.

En el caso que el usuario quiera introducir un descriptor con un nombre que ya existe, este script también se encarga de mostrar el mensaje de error adecuado, sino, seguirá el proceso habitual. Para detectar el error sigue el mismo procedimiento que en el caso de los videos, recorre toda la tabla de descriptores comparando cada nombre de *Descriptores* con el introducido por el usuario, y en el momento que detecte que uno coincide ya muestra el mensaje por pantalla, sino recorre la tabla hasta el final e inserta el descriptor.

El borrado del descriptor se hace de la misma manera que para borrar un video, es decir, se crea un menú desplegable con todos los nombres de los descriptores que estén en ese momento disponibles dentro de la base de datos. Por lo tanto, se requiere una conexión con ésta para poder acceder a la consulta. Para realizar el borrado se precisa un nuevo script que se encarga a partir del nombre del descriptor seleccionado por el usuario y mediante una secuencia SQL que permite borrar datos de una tabla, borrar el descriptor. Dentro de este script se crearon tres enlaces HTML, uno de ellos direccionado a la página principal de descriptores para que el usuario pueda visualizar el contenido de la tabla una vez haya efectuado algún cambio. Los demás enlaces se dirigen hacia la página de insertado y hacia la página de borrado, para que el usuario vuelva a tener la opción de insertar y de borrar un descriptor.

Diseño del portal web para formatos

Este nuevo portal no tiene mucha diferencia a los creados para videos y para descriptores. La página principal de los formatos también se encarga de mostrar la tabla *Formatos* por lo que en el script que forma la página se requiere una conexión con la base de datos.

Una vez se efectúa la consulta a la tabla, los datos se recogen dentro de una tabla HTML.

Desde la página principal de formatos también se puede enlazar a las páginas de inserción y borrado, por lo que fue necesario crear estos dos enlaces mediante código HTML.

Dentro del script para insertar un formato se crearon en lenguaje HTML las ranuras que permiten al usuario introducir los datos necesarios para llenar la tabla *Formatos*. Una vez se han rellenado los campos, otro script se encarga de introducir los datos dentro de la tabla *Formatos* utilizando la misma sentencia SQL que permite insertar datos en una tabla.

Desde este script también se crearon los enlaces necesarios para acceder a la página principal de formatos, para insertar un nuevo formato y para borrar. Para crear los enlaces se volvió a utilizar código HTML.

Tal y como ocurre en los portales anteriores (Videos y Descriptores), al insertar un formato con nombre ya existente, el sistema se queja impidiendo la inserción y dando la posibilidad al usuario de introducir de nuevo el formato con otro nombre.

Para poder borrar un formato, sólo se necesita que el script acceda a la tabla creada en la base de datos y extraiga de ella todos los nombres de los formatos disponibles. Con esos nombres se crea un menú desplegable desde el cual el usuario seleccionará el nombre del formato que desea borrar. Mediante esta selección, otro script se encarga de conectar con la base de datos y ejecutar la sentencia que permite borrar un registro de la tabla indicando el nombre seleccionado, por lo que solamente se borra el formato cuyo nombre coincida con el marcado por el usuario. Como en los otros scripts de borrado, éste, también dispone de tres enlaces creados en HTML que permiten o bien visualizar la tabla, o bien volver a insertar o bien borrar un formato, por lo tanto, estos enlaces están direccionados a las páginas webs respectivas.

Diseño del portal web del buscador

Para crear el portar web del buscador se necesitaron dos scripts diferentes. El primer script muestra la tabla referente a los videos y la tabla de descriptores, por lo tanto para visualizar estas tablas es necesaria una conexión con la base de datos mediante código PHP y SQL. Esta parte se hizo exactamente de la misma manera que en los scripts principales de videos y de descriptores, es decir, se crearon primeramente dos tablas HTML para ir rellenándolas con los valores que se obtengan de las consultas a las tablas.

Se debe de recordar que en el caso de la tabla de videos se necesita mostrar un campo más que no está incluido dentro de la tabla *Videos* creada en la base de datos. Este campo hace referencia al campo *Frame*, que consiste en un campo donde se muestra una imagen de cada secuencia de video, por lo tanto, se necesita crear este frame, para visualizarlo.

Como se ha comentado con anterioridad, se debe ejecutar el binario de extracción de frame mediante la función PHP *shell_exec()*, dando como parámetros de entrada, la ubicación del binario de extracción de frame, que se obtiene mediante una consulta SQL a la tabla de formatos. Como segundo parámetro de entrada se necesita la ubicación del video que se obtiene mediante una consulta a la tabla de videos. Y por último, la ubicación donde se guardará el frame o imagen. Una vez acabada la consulta a la tabla de videos, se hace lo mismo para la tabla de descriptores.

Cuando ya se consigue visualizar las dos tablas, se crean dos cajas de texto para que el usuario pueda seleccionar un video y un descriptor de la base de datos. Para crear las ranuras, se utiliza código PHP para conectar con *Video_Indexing* y consultar los nombres tanto de videos como de descriptores. En este caso también se necesita código HTML para crear las ranuras y crear los menús desplegables.

Por último, dentro del fichero se crea un botón en HTML que permita realizar la búsqueda. El botón recibe el nombre “Buscar”. Al hacer clic sobre “Buscar” el sistema web muestra otra página con el resultado de la búsqueda.

Para que la búsqueda sea correcta se necesita otro script en PHP *(busca.php)*. Este nuevo script se encarga primeramente de conectar con la base de datos para poder realizar las consultas que hagan falta. Al saber el nombre del video y del descriptor seleccionados, se encarga de acceder a las respectivas tablas y consultar las *id* de cada uno. Conocer estos dos valores ayuda a encontrar el fichero de descripción creado bien cuando se haya introducido el video o bien cuando se haya introducido el descriptor. Se debe recordar que los ficheros de descripción reciben como nombre la *id* del video más la *id* del descriptor.

Saber el nombre que tiene el fichero es necesario para poder ejecutar el binario de comparación del descriptor, porque en este momento lo que debe hacer el sistema es a partir de este fichero de descripción y de los demás ficheros

correspondientes a todos los videos de la base de datos, ejecutar el binario de comparación para determinar la similitud entre videos. Para conseguir los demás ficheros, se necesita conocer las *id* de cada uno de los videos de la base de datos. Esto se consigue mediante una consulta a la tabla *Videos*, por lo tanto, una vez se obtienen los valores *id* de los videos, al conocer ya la *id* del descriptor, es fácil llamar a los ficheros.

Otro parámetro necesario para la ejecución del binario de comparación es conocer su ubicación, por lo tanto, se debe hacer otra consulta a la tabla *Descriptores*, esta vez indicando que devuelva la ubicación del binario de comparación del descriptor seleccionado. Cuando ya se tienen todos los parámetros necesarios, se ejecuta el binario de comparación, recordando que se ejecutará tantas veces como videos haya en la base de datos. Esto se debe a que la búsqueda se hace para todos los videos para poder extraer un listado de los que más se le parecen al que menos al video seleccionado.

El resultado que se obtiene al ejecutar el binario de comparación, que en este caso es un número entre 0 y 1, se va guardando dentro de un vector.

Al finalizar la ejecución del binario para cada uno de los videos, se necesita ordenar el vector con los valores de comparación de mayor a menor, para poder determinar qué video es el que más se le parece al que menos. Al ordenar el vector hay que tener en cuenta que, posteriormente, se debe de relacionar cada valor de comparación con el nombre del video. Para evitar errores, se colocan dentro de otro vector todos la *id* de los videos de la base de datos ordenadas según el orden del valor de comparación al que hacen referencia, de manera que al ordenar un vector se ordene el otro simultáneamente.

La ordenación de los vectores en orden descendiente se hace utilizando el llamado “método de la burbuja”, que consiste en ir comparando cada valor del vector con el de la posición de al lado, de manera que si el valor de al lado es mayor, se invierten las posiciones. El procedimiento se hace tantas veces como sea necesario hasta que no se produzca ningún cambio de posición en el vector, en ese momento el vector ya está ordenado.

Para ordenar el vector con los identificadores (*id*) se sigue el mismo procedimiento. Cuando un valor de comparación cambie de posición, la *id* correspondiente también cambia tomando la misma posición.

Una vez están los dos vectores ordenados, se relaciona cada *id* con el nombre del video correspondiente, esto se hace mediante una consulta a la tabla *Videos* indicando que devuelva un vector de nombres a partir del vector con las *id*, así los nombres que devuelva también estarán ordenados.

Como último paso sólo queda visualizar el frame correspondiente a cada video. Como todos los frames se han guardado como id_Video.jpg, introduciendo el vector con las *id* ordenadas en la instrucción que permite visualizar imágenes por pantalla, se muestran también todos los frames ordenados, por lo que el resultado de la búsqueda será el esperado.

Diseño de la interfaz del usuario

La interfaz del usuario se creó mediante código HTML. Dispone de una página principal dividida en cuatro frames o ventanas que hacen referencia a cada una de las particiones de la página. Los frames se crearon mediante código HTML indicando el tamaño que se deseaba para cada frame. Cada uno de estos frames se enlazó con un script que contiene el código necesario para mostrar lo que se desea.

El frame situado en la parte superior izquierda de la pantalla corresponde con el logotipo de la universidad donde se ha realizado el proyecto (UPC). Para mostrar el logotipo se creó un fichero *(frame3.html)* donde únicamente se refleja el color de fondo que tiene el frame, que es el mismo utilizado para toda la interfaz. Al mismo tiempo se utilizó la instrucción HTML necesaria para mostrar imágenes por pantalla.

El frame donde se refleja el título del sistema web "BUSCADOR DE VIDEOS", se creó con el script *(frame1.html)*, donde únicamente se describe el color de fondo y el tamaño y el título de lo que se quería escribir. Otro de los frames muestra el listado de las principales páginas disponibles en todo el portal web. Cada uno de los puntos de la lista es un link hacia la página que hace referencia. Para crear estos enlaces, se utilizó HTML que permite direccionar cada uno de los enlaces con las páginas correspondientes.

En el mismo fichero y junto a cada enlace se indica el frame donde se quiere que se abra la nueva página una vez se seleccione el nombre de la web. Todas las páginas se abren sobre el último frame que queda por describir y que

es el frame principal. El frame principal no hizo falta describirlo con ningún frame, ya que únicamente se utiliza para mostrar todas las páginas que el usuario desee visitar. Por lo tanto, para su correcto funcionamiento, sólo se necesita que los enlaces del frame anterior estén redireccionados a este. Cuando se accede por primera vez al buscador, en este frame se muestra la página principal de la búsqueda.

Para seleccionar el color de fondo del portal web así como la letra, simplemente fue necesario introducir al principio de cada script unas líneas de código HTML indicando el color deseado para el fondo, para el texto, para los enlaces, etc. El color de fondo escogido es un color cielo similar al color de la UPC. Todo el texto y los enlaces se describieron en color blanco puro para ser más legible. Para poder observar estas características véase la figura 9, que viene a continuación

INTERACTIVIDAD DEL USUARIO

En este apartado se pretende dar una serie de instrucciones sobre el funcionamiento del sistema web.

Funcionamiento del sistema web

Antes de abrir el navegador para acceder al portal web del proyecto, hace falta arrancar el servidor local Xampp. Para ello, simplemente se debe introducir en el terminal de Linux la siguiente instrucción:

```
cd /opt/lampp/
./lampp start
```

En el momento que la línea de comandos nos indica que el servidor está funcionando sólo queda acceder al navegador e introducir en la barra URL la dirección de la página principal del buscador: *http://localhost/projecte/bucador/principal.html*.

Es necesario conocer que para conectarse al sistema web desde el ordenador local, se debe de introducir la dirección anterior, pero en el caso que se desee acceder desde fuera, se necesita cambiar "localhost" por la dirección IP o el nombre del ordenador donde esté ubicado el sistema.

Se debe de tener en cuenta que para parar el servidor el comando ha escribir es:

```
cd /opt/lampp/
./lampp stop
```

Una vez el navegador haya encontrado la página principal se observa en la pantalla la siguiente interfaz (Figura 9). Esto es posible siempre que existan videos dentro de la base de datos. En el caso que, *Video_Indexing* estuviera vacía para poder visualizar una página similar a la siguiente se necesita introducir videos dentro de la base de datos.

Figura 9. Página principal del buscador

Antes de realizar cualquier búsqueda, se necesita llenar la base de datos con los videos, los descriptores y los formatos necesarios.

A continuación se muestran los pasos para realizar estas funciones:

Inserción y borrado de videos

Una vez se accede a la página principal del buscador, se puede observar en la parte izquierda de la imagen, un enlace a la página de videos. Si se hace un clic en el enlace, el navegador abre el portal web referente a los videos.

Figura 10. Página web de videos

Como se puede observar en la figura anterior (Figura 10), la web de videos, muestra la tabla de videos disponible en la base de datos, indicando todos los aspectos o campos a destacar de la tabla.

Para poder insertar un nuevo video dentro de la tabla *Videos*, es necesario hacer clic sobre el enlace situado en la parte inferior de la página "Insertar un video". El enlace conduce de nuevo a otra página web, donde es necesario introducir los campos requeridos para la correcta inserción de un nuevo registro.

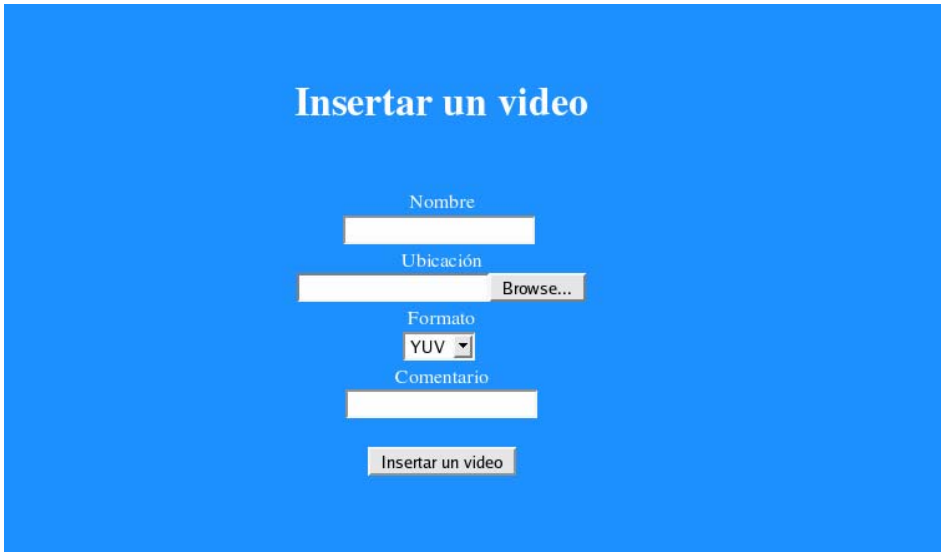

Figura 11. Página de inserción de un video

Como se puede observar en la figura (Figura 11), el usuario que desee introducir un video, únicamente tiene que introducir cuatro de los campos que pertenecen a la tabla *Videos*. El sistema se encarga de forma automática de introducir los demás.

En la casilla referente al nombre, el usuario debe de introducir el nombre que le quiera dar al video. Es indiferente que este nombre coincida o no con el nombre dado a la secuencia almacenada dentro del directorio */opt/lampp/htdocs/projecte/seqs*. Sólo se requiere que el nombre a insertar no exista.

En el apartado de "*Ubicación*", se debe de introducir el path (ubicación o ruta) completo dónde se encuentre ubicado el video dentro del disco duro. Para poder seleccionar el path con más comodidad hay disponible junto a la casilla un botón llamado "Browse" que permite al hacer clic sobre él, navegar por todos

los directorios existentes en la máquina. Es recomendable como se ha dicho con anterioridad que el video esté ubicado dentro del directorio */opt/lampp/htdocs/projecte/seqs*, para evitar posibles confusiones en el momento que el sistema tenga que acceder al video.

El tercer campo que se debe de introducir por parte del usuario, es el campo referente al formato. Para saber cuáles son los formatos disponibles dentro de la base de datos únicamente se debe de desplegar el menú del campo formato y seleccionar el deseado. En el caso que el usuario desee un formato que no se encuentra disponible, se requiere acceder a la página principal de formatos y desde ahí insertar un nuevo formato. Una vez insertado el nuevo formato, ya se tendrá disponibilidad de él en la página de inserción de videos.

Como último, comentar que en el momento de insertar un video también se puede insertar un comentario referente a este. Una vez estén todos los campos correctamente rellenados, simplemente hay que hacer clic sobre el botón “Insertar un video” y el video, quedará almacenado dentro de la base de datos.

Como se ha comentado en el apartado de realización, se puede dar el caso que al introducir un video, el usuario esté introduciendo un nombre ya existente. En el momento que se vaya a insertar el video, el sistema muestra un mensaje de error. Al mostrar el mensaje se da la posibilidad de que el usuario vuelva a introducir el video, esta vez con un nombre diferente, esto es posible mediante el enlace “Insertar de nuevo el video”.

Figura 12. Mensaje de error

Si no hay ningún problema, en el momento en que se introduce el nuevo video, se abre otra página indicando que el video ha sido insertado. Dentro de esa misma página existen tres enlaces que permiten al usuario o bien visualizar el contenido de la tabla de videos, o insertar un nuevo video o bien borrar alguno de los ya existentes en la tabla. Si se decide visualizar el contenido de la tabla, se accede de nuevo a la página principal de la web de videos, donde se puede observar la correcta inserción. Si por lo contrario se desea introducir un nuevo video, el enlace conduce a la página anterior de inserción. Pero si lo que el usuario desea hacer es borrar un video de la base de datos, el enlace "borrar un video" accede a una nueva web donde se observa una caja que dispone de un menú desplegable donde se indican todos los videos existentes en la tabla.

Figura 13. Borrar un video.

Si se selecciona el nombre del video que se desea borrar y se hace clic sobre el botón "Borrar un video", el video queda automáticamente borrado y se accede a otra página donde al igual que en el insertado se tiene la opción de visualizar el contenido de la tabla, de insertar un nuevo video o bien de borrar un video ya existente.

Inserción y borrado de descriptores

Tal y como se ha hecho en el caso de los videos, desde la página principal del buscador, se puede acceder mediante el enlace "DESCRIPTORES" a la

página web principal de los descriptores, donde se puede visualizar la tabla de descriptores que exista en el momento.

Figura 14. Página principal de los descriptores

Del mismo modo que en videos, la página de descriptores dispone de dos enlaces, que permiten al usuario o bien insertar un descriptor o bien borrarlo. Si se decide insertar un descriptor la página que se muestra es como la de la figura siguiente (Figura 15) :

Figura 15. Página de inserción de descriptores

Tal y como se puede ver en la figura (Figura 15), para insertar un descriptor se requiere diferentes campos:

- **Nombre**: dentro de este campo se debe de introducir el nombre que se le quiere dar al descriptor.
- **Binario de Extracción**: como su nombre indica, el campo hace referencia al binario de extracción perteneciente al descriptor que se va a introducir. Para completar este campo, se debe de seleccionar la ubicación exacta donde se encuentra el binario, recordando que el binario debe de almacenarse dentro del directorio */opt/lampp/htdocs/projecte/binarios/*.
  Para seleccionar la ubicación hacer clic sobre el botón "BROWSE".
- **Binario de Comparación**: al igual que Bin_Extract, este campo coincide con la ubicación exacta del binario de comparación del descriptor. La selección de la ruta se realiza también desde el botón "Browse" situado justo al lado de la casilla.
- **Comentario**: del mismo modo que en el caso de la inserción de videos, este campo permite al usuario tener la oportunidad de hacer algún comentario sobre el descriptor.

Cuando todos los campos estén completos, hacer clic sobre el botón "Insertar un descriptor". En el caso que al introducir el descriptor el sistema diera un mensaje de error, indica que el nombre que se está introduciendo ya existe. Para evitar confusiones por parte del sistema se requiere un nombre que no exista, es por ello que se le indica al usuario que inserte el descriptor de nuevo. Pero si el nombre no existe, el descriptor se inserta correctamente y aparece de nuevo una página indicando que el registro ha sido insertado. Desde esta página se puede visualizar el contenido de la tabla, insertar un descriptor o bien borrar el que se desee.

Si se decide visualizar la tabla descriptores, el sistema conduce hacia la página principal de descriptores. En el caso de que se desee insertar otro descriptor dentro de la base de datos, se dirige hacia la página de inserción. Pero si por lo contrario, se quiere borrar un video, la página que aparece es como la de la figura (Figura 16).

Figura 16. Página de borrado de descriptores

Dentro de esta página, aparece una casilla que hace referencia al nombre del descriptor y que dispone de un menú desplegable que muestra todos los descriptores existentes ordenados por nombre. El usuario únicamente debe de seleccionar el nombre del descriptor que quiere borrar y hacer clic sobre el botón "Borrar un descriptor". A la página de borrado de descriptores también se puede acceder desde la página principal de descriptores mediante el enlace "Borrar un descriptor".

Inserción y borrado de formatos

Para poder visualizar la tabla de formatos así como insertar o borrar alguno, es necesario acceder desde la página principal del buscador mediante el enlace "FORMATOS" a la página principal de formatos. Desde esta página se puede ver los formatos disponibles en la base de datos. Al igual que las páginas de videos y descriptores, desde formatos también existen dos enlaces que permiten insertar y borrar.

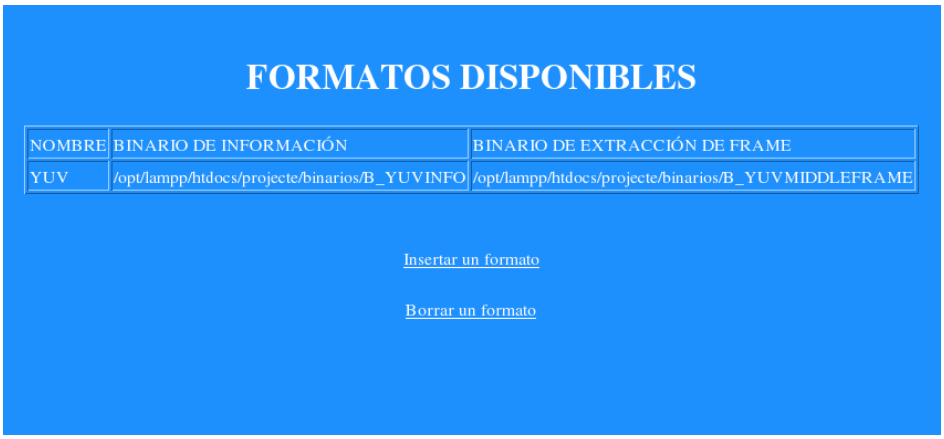

Figura 17. Página principal de formatos

Si el usuario desea insertar un formato nuevo dentro de la base de datos, debe de hacer clic sobre el enlace "Insertar un formato" para acceder a la siguiente página:

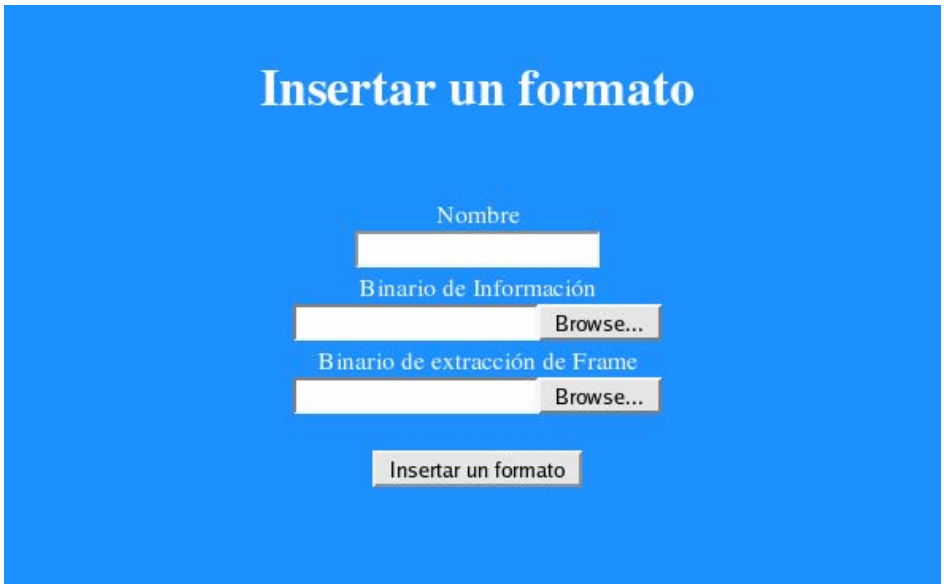

Figura 18. Página de inserción de formatos

Para insertar un formato se requiere insertar el nombre del formato, la ruta completa del binario de información y la ruta del binario de extracción de frame. Estos dos últimos campos se pueden completar navegando por el disco de la máquina a través del botón "Browse".

Al querer insertar un formato con un nombre que se corresponde a otro formato existente en la tabla, el sistema muestra un mensaje de error y se debe de introducir de nuevo el formato con un nombre diferente.

Si lo que se desea es borrar un formato, hay dos opciones de hacerlo. Una vez se ha insertado el nuevo formato, el sistema se dirige hacia una página que indica que el formato ha sido insertado, y desde la cual hay un enlace "Borrar un formato" que si se hace clic sobre él se accede a la página de borrado.

Figura 19. Página de borrado de formato.

Desde la página que se muestra en la figura (Figura 19), se puede borrar un formato únicamente seleccionando el nombre del que se quiere borrar de un menú desplegable. Otra manera de borrar un formato es desde la misma página principal de formatos mediante el enlace "Borrar un formato" que conducirá a la página referente en la figura.

Una vez ya se han insertado los videos deseados, los descriptores y los formatos que se van a utilizar, ya se puede proceder a realizar la primera búsqueda.

Acceso a la búsqueda

Para poder acceder a la búsqueda de videos en base a otro video y a un descriptor seleccionado previamente, es necesario que desde la página principal del buscador (*http://localhost/projecte/buscador/principal.html*) se acceda a la página de búsqueda.

El acceso resulta tan fácil como hacer clic sobre el enlace llamado "BUSCADOR" de la parte izquierda de la página principal.

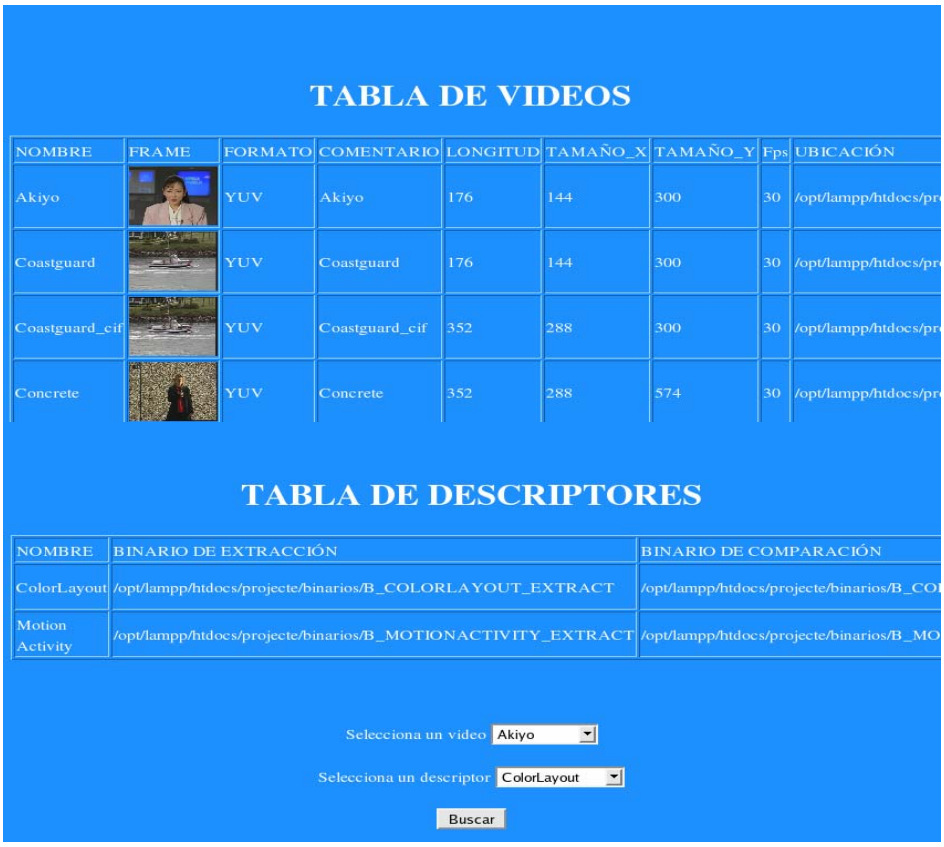

Figura 20. Página principal de búsqueda

A partir de la figura anterior (Figura 20) se puede observar que en la página principal de la búsqueda se presentan las tablas de videos y la de descriptores. La funcionalidad que tiene mostrar las tablas es que el usuario, antes de acceder a buscar los videos, puede ver cuáles son los videos actuales en la base de datos así como los descriptores. De este modo puede añadir o borrar aquellos descriptores o videos que desee o que no le interesen.

Una vez que el usuario tiene claro qué video seleccionar para hacer la búsqueda y qué descriptor utilizar dependiendo de lo que desee obtener, solamente debe de seleccionar el video y el descriptor de un menú desplegable que se presenta para ambos registros en la parte inferior de la página. Cuando la selección ya esté efectuada, queda hacer clic encima del botón “Buscar”.

El resultado de la búsqueda se expondrá en otra página, donde se puede ver un listado de videos, desde el video que más se le parece al seleccionado al que menos dentro de la base de datos.

Figura 21. Página con el resultado de la búsqueda

Como se puede ver en la figura (Figura 21), como resultado de la búsqueda se muestra una tabla con tres campos. El primer campo indica el nombre de los videos. El segundo campo muestra un frame de las secuencias de videos y el tercer campo se corresponde con el valor de comparación obtenido entre el video seleccionado y el video disponible en la base de datos, que indica el grado de similitud entre los videos. La tabla está ordenada desde el video que más se le parece al seleccionado al que menos, todo ello en base a la característica que indique el descriptor.

Como última página a destacar y para información del usuario, la web principal dispone de un enlace llamado “ACERCA DE...” donde se muestra un escrito informativo del buscador.

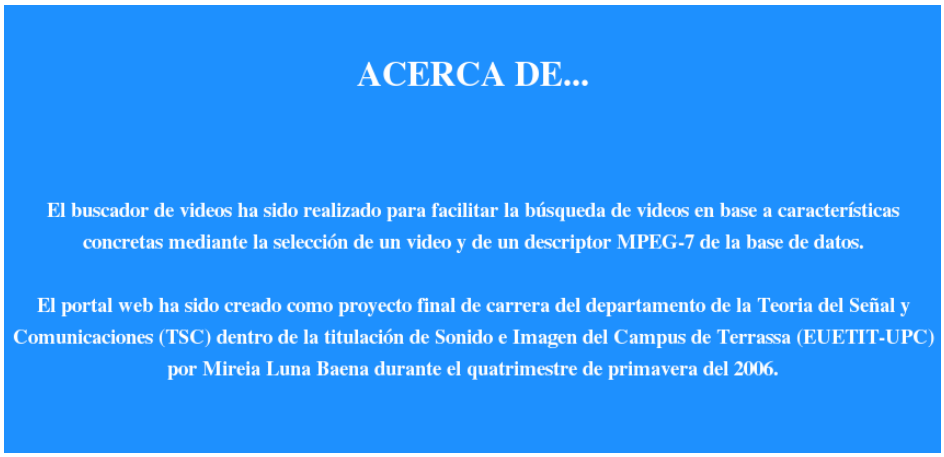

Figura 22. ACERCA DE...

REQUISITOS DEL SISTEMA

Dentro de este apartado se pretende dar una serie de instrucciones prácticas para el correcto funcionamiento del sistema web.

Hay que dejar claro que en el momento que el usuario decide introducir un nuevo descriptor, debe previamente guardar los binarios dentro del directorio recomendado */opt/lampp/htdocs/projecte/binarios/* y posteriormente debe dirigirse a la línea de comandos de Linux y escribir la siguiente instrucción:

chmod +x "ubicación completa del binario"

Este paso se debe hacer cada vez que el usuario inserte un binario bien sea referente a un descriptor o referente a un formato. El comando chmod +x asegura que el binario que se introduce sea ejecutable, en el caso, que esto no se haga, el binario no se podrá ejecutar.

Otro aspecto a destacar es referente a las carpetas *descripciones_temporal* y *jpg_temporal* situadas en el directorio del proyecto */opt/lampp/htdocs/projecte/*. Como ya se sabe, estas carpetas fueron creadas para guardar archivos generados por el sistema web, por lo tanto, para poder escribir un fichero dentro de ellas es imprescindible dar permisos de escritura mediante el siguiente comando Unix:

chmod a+rwx "ubicación completa de cada una de las carpetas"

Ejemplo:

chmod a+rwx /opt/lampp/htdocs/projecte/descripciones_temporal
chmod a+rwx /opt/lampp/htdocs/projecte/jpg_temporal

Debido a problemas de seguridad de Linux se vio necesario definir una serie de variables al inicio de cada uno de los scripts. Estas variables indican la raíz de los directorios donde se almacenan las secuencias de videos así como todos los binarios. Esto se debe a que al seleccionar la ruta bien de una secuencia o bien de un binario mediante el botón "BROWSE" que aparece en las páginas de insertado, la ruta completa no queda almacenada. Solamente se almacena el nombre del binario o de la secuencia junto a su extensión, por lo

tanto, como para la ejecución de todos los binarios se requiere tener la ubicación completa, se tuvieron que definir estas variables. Para conocer la ubicación completa de un binario o de una secuencia, se requiere la suma de las dos cadenas de caracteres, la primera cadena es la variable definida con la raíz del directorio y la segunda cadena es el nombre de la secuencia o del binario junto a su extensión.

A parte de estas variables también se definieron otras que hacen referencia a la ruta donde se sitúan las carpetas temporales "*descripciones_temporal*" y "*jpg_temporal*", se requiso definir estas variables al principio de cada script o de los scripts que utilizan esta ubicación, porque si se diera el caso que el usuario deseara cambiar de ubicación las carpetas le seria más fácil cambiar estas variables que sabe que se sitúan al principio de cada fichero, que no tener que mirar todo el código e ir cambiando.

También es importante conocer que para visualizar las imágenes que se crean para cada una de las secuencias, la ruta necesaria no hace referencia a la ubicación puesto que las instrucciones que se utilizan para la visualización no es código PHP sino HTML. HTML requiere que todas las referencias a ubicaciones sean siempre con respecto al servidor web, por lo tanto la ruta que se indica es: *http://Ip_máquina/projecte/jpg_temporal*.

Indicar que en el caso que el servidor local que se utilice fuera diferente a Xampp, el directorio donde situar todos los scripts cambiaria, es decir, la raíz dejaría de ser */opt/lampp/htdocs/*. Para poder visualizar el sistema web del buscador en otro servidor, primeramente se debe averiguar en qué directorio se deben situar los ficheros del nuevo servidor para poder visualizarlos desde el navegador y luego copiar toda la carpeta "*projecte*" dentro del directorio, recordando cambiar todas las variables definidas al principio de los scripts para el correcto funcionamiento. Se debe recordar también que todos los nombres tanto de las carpetas de "*projecte*" como de los scripts que forman el proyecto deben de estar escritos de la misma manera que han sido creados. El nombre original se refleja en la figura expuesta en el apartado "ESTRUCTURA DE DIRECTORIOS"

Cabe destacar también, la necesidad de crear nuevamente la base de datos al cambiar de máquina, así como crear de nuevo el usuario "Mire" para poder manejarla. Todas las instrucciones necesarias se encuentran en el apartado de "REALIZACIÓN". Esto debe hacerse únicamente en el caso que la máquina no esté conectada en red con el ordenador donde está implementado el sistema web o en el caso que se desee implementar el sistema en otro ordenador.

Si el usuario del sistema web desea crear un usuario distinto a *Mire,* debe conocer que el sistema no funcionará con el nuevo usuario puesto que en cada script que se accede a la base de datos, tanto el nombre de usuario como el password son variables fijas, por lo tanto, al cambiar de usuario el sistema desconocerá quien tiene privilegios sobre la base de datos. Solamente funcionará, si al crear un usuario distinto, se modifica el código de los scripts PHP que conectan con *Video_Indexing*.

PRUEBAS

Dentro de este apartado cabe destacar que en un principio se consideró la opción de crear los ficheros de descripción de cada uno de los videos con los respectivos descriptores justo al hacer la búsqueda, pero se observó que el proceso retardaba mucho el acceso al resultado, ya que cada vez que se efectuaba una búsqueda el sistema creaba de nuevo todos los ficheros correspondientes, sin considerar que la mayoría de ellos ya estaban creados de la búsqueda anterior, por lo que se encontró un método poco eficaz.

La mejora que se hizo fue crear estos ficheros justo al introducir o bien un video o bien un descriptor como se ha podido observar a lo largo de la memoria, esto no acelera el tiempo que el fichero tarda en crearse, puesto que se tarda el mismo tiempo en crease en este momento, como en el momento de la búsqueda, lo único que mejora, como se ha comentado, es la rapidez con la que se accede al resultado de la búsqueda.

Durante la realización del proyecto, se hicieron infinidad de pruebas para observar tanto el funcionamiento correcto de cada script en particular como el funcionamiento de todo el sistema en conjunto. Para efectuar las pruebas necesarias se accedía desde el servidor local de la máquina (Xampp). Se pudo observar en cada una de estas pruebas que dependiendo de la función que tuviera que efectuar cada fichero, sobretodo aquellos que crean los archivos de descripción, el sistema realizaba su función en más tiempo o en menos, esto dependía y depende del descriptor que se utilice en cada uno de los casos, así como de las características específicas de las secuencias. Ya que, dependiendo de las sentencias que se utilicen dentro del algoritmo repercute en la eficiencia

del descriptor, un ejemplo claro está en que una sentencia del tipo bucle como podría ser un while o un for tarda mucho más en ejecutarse que un simple if.

Lo mismo pasa cuando un descriptor se debe de ejecutar para una secuencia de larga durada, con respecto a otra que dure menos, puesto que el análisis es más largo para la secuencia más larga que no para una secuencia de menos.

Para poder valorar si el sistema web realizaba la búsqueda de manera correcta, dentro de la base de datos se incluyeron dos videos exactamente iguales que, únicamente se diferencian en el tamaño, estos videos eran Coastguard y Coastguard_cif de manera que al realizar una búsqueda el valor de comparación entre los dos debería de ser muy próximo a 1.

Por lo tanto estos dos videos fueron la referencia más clara para determinar que el resultado era el esperado.

A continuación se exponen dos ejemplos claros donde se puede observar el correcto funcionamiento del portal web.

Ejemplo 1:

Figura 22. Ejemplo 1 de búsqueda

Tal y como se puede observar en la figura, si se selecciona como video Coastguard y como descriptor ColorLayout, el valor de comparación entre Coastguard y Coastguard_cif es muy próximo a 1, exactamente, 0.999587 de manera que nos indica que la distribución espacial de los colores entre los dos videos es exactamente la misma, esto es obvio considerando que los dos videos son iguales, siendo su única diferencia el tamaño.

Ejemplo 2:

Figura 23. Ejemplo 2 de búsqueda

Al igual que en el ejemplo anterior, se selecciona como video Coastguard y como descriptor Edge Histogram, este descriptor tiene como funcionalidad detectar los bordes de las secuencias por lo tanto considerando que el video seleccionado es igual a Coastguard_cif, es lógico que el número de bordes sea similar, por lo tanto el resultado obtenido es el adecuado.

APLICACIONES

Una vez visto el funcionamiento del sistema se valoran las aplicaciones que puede tener.

Básicamente, las aplicaciones previstas para el proyecto, primeramente es la visualización de todos los videos de la base de datos, así como su indexación mediante descriptores MPEG-7 y por último hacer posible la búsqueda de estos videos mediante los mismos descriptores, basándose sobretodo en encontrar videos que resulten similares en base a una serie de características que indican los propios descriptores.

La aplicación más segura que tendrá el proyecto será su uso dentro del departamento de Teoría del Señal y Comunicaciones (TSC) de la Universidad Politécnica de Cataluña (UPC), donde sobretodo profesores, doctorados y otros proyectistas lo usarán como método de indexación y búsqueda de secuencias de videos.

CONCLUSIONES

Para concluir se puede ver que los objetivos que se tenían en un principio para el proyecto, se cumplieron satisfactoriamente.

Aún así, pueden considerarse una serie de mejoras futuras. Alguna sugerencia podría ser que en lugar de la visualización de un único cuadro de la secuencia, se llegara a crear un enlace que al hacer clic sobre él, un reproductor permitiera mostrar el video completo. Otra opción muy eficaz, seria la construcción de un enlace extra dentro de la tabla que se visualiza en cada uno de los portales web (videos, descriptores y formatos) de manera que permitiera un borrado automático desde la propia tabla sin necesidad de implementar una página más para efectuar esta opción. Un aspecto interesante a mejorar podría ser, la creación de la imagen de un video (frame) únicamente una vez al igual que sucede con los ficheros de descripción de cada video-descriptor. Actualmente, ésta imagen se crea cada vez que se visualiza la tabla de videos, por lo que al crearse una sola vez ya seria suficiente, el momento adecuado y único para crear el cuadro sería justo al introducir un nuevo video en la base de datos. Para terminar, otra mejora que no tendría ninguna relación con los objetivos del proyecto sería crear una interfaz mucho más atractiva para el usuario.

La intención del proyecto era crear un portal web para la indexación, búsqueda y adquisición de videos mediante descriptores MPEG-7 y en definitiva es la función que tiene el sistema web implementado.

REFERENCIAS BIBLIOGRÁFICAS

Durante la realización tanto del proyecto como de la memoria se utilizaron las siguientes referencias bibliográficas:

[1] ALBACETE GARCÍA, Antonio. "Gestión Completa de Vídeo". *ISID (Ingeniería i Sistemas de Información y Documentación).* Noviembre de 2002.

[2] "MPEG-7 Video Browsing and Summarization". Developed by: The Murray Hill Laboratory (MHL) of Mitsubishi Electric Research Laboratories. < http://www.merl.com/projects/video-browsing/>

[3] "MPEG-7-based Audio Annotation for the Archival of Digital Video". Project Manager: Prof. Dr.-Ing. Thomas Sikora Founded by BMWA (German Federal Ministry of Economics and Labour) Project Period 11/2002 – 03/2005 < http://www.nue.tu-berlin.de/forschung/projekte/mpeg7/>

[4] URL: <http://www.gestiondelconocimiento.com/software_busca.htm>

[5] URL:<http://www.e-buscadores.com/noticias/2005/01/googlevideo-bsqueda-de-videos.html>

[6] URL:<http://www.e-buscadores.com/noticias/2004/12/yahoo-video-search.html>

[7] VIVANCOS VICENTE, Pedro José. "Descripción de contenidos multimedia: introducción al estándar MPEG-7". *Revista de ingeniería Informática del CIIRM* (Colegio de Ingenieros en Informática de la Región de Murcia). 15/07/2005

[8] URL: <http://www.telefonica.net/web2/blasinski/video_codecs/MPEG.html

[9] URL: <http://es.wikipedia.org/wiki/MPEG-21>

[10] HERRANZ ARRIBAS, Luis. *Accesso Multimedia Universal a través de terminales y redes fijas mediante MPEG-7 y MPEG-21*. Proyecto Final de Carrera, Universidad Politécnica de Madrid, Julio de 2003

[11] URL: < http://www.ciberaula.com/curso/lamp/que_es/>

[12] URL: <http://es.wikipedia.org/wiki/Linux>

[13] URL: <http://es.wikipedia.org/wiki/SQL>

[14] URL: <http://es.wikipedia.org/wiki/PHP>

[15] URL: <http://geneura.ugr.es/~maribel/php/index.html>

[16] URL: <http://es.wikipedia.org/wiki/HTML>

Manuales y tutoriales consultados en el proyecto

[17] GONZALEZ ESTRADA, Joel. *Desarrollo Web con PHP y MySQL*. Manual PHP y MySQL

[18] URL:
<http://www.elguruprogramador.com.ar/tutoriales/tutorial_php.asp>

[19] URL: <http://www.manualdephp.com/>

[20] URL: < http://www.programacion.com>

[21] URL:
<http://es.tldp.org/Manuales-LuCAS/manual_PHP/manual_PHP/>

[22] URL:
<http://de3.php.net/manual/es/function.mysql-connect.php>

[23] URL:
<http://www.mailxmail.com/curso/informatica/apache/capitulo5.htm>

[24] MÖHRKEN, Carsten. Uso de PHP en scripts de administración, OPERACIÓN COMANDO. (Linux Magazine, Número 03)
URL: <http://www.linux-magazine.es/issue/03/PHP.pdf>

[25] URL: <http://www.programatium.com/manuales/mysql/mysql2.htm>

[26] URL: <http://www.desarrolloweb.com/>

[27] URL: <http://club.telepolis.com/ortihuela/html.htm>

[28] URL:
<http://www.htmlweb.net/manual/formularios/formularios_7.html>

[28] URL: <http://www.webestilo.com/>