

PROPUESTA DE UN MODELO BASE PARA UN SISTEMA DE BÚSQUEDA DE VIDEOS DIGITALES A TRAVÉS DE METADATOS

Trabajo de diploma para optar por el título
de Ingeniero en Ciencias Informáticas.

Autores:

Ivis Cabrera Guerra
Roexcy Vega Prieto

Tutor:

Ing. Guillermo Báez Ramos



SEARCH



Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio, conscientes de recibir el premio de la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.

Ernesto Che Guevara

A mis padres:

A mi mamá que abnegadamente me ha apoyado y guiado en todo momento, siendo madre, amiga, hermana, por siempre estar ahí cuando la necesito.

A mi papá por la educación que supo darme, por confiar siempre en mí, por ser mi apoyo y mi ejemplo en la vida. No hay espacio ni tiempo capaz de describir mi amor y mi agradecimiento hacia ellos.

A mis hermanos por convertirse en mis segundos padres, renunciando muchas veces a sus gustos por complacer los míos, por estar ahí y saber que siempre puedo contar con ellos, los adoro.

A todos en mi familia por el cariño que me han dado, por preocuparse siempre por mí, por guiarme en cada etapa de mi vida.

A Yerandi porque me guió en el momento preciso.

A Alleyne por ser ese novio comprensivo, cariñoso y atento, por estar ahí cuando lo necesito, siempre dispuesto a ayudarme, a quererme.

A mis amigas por comprenderme, soportarme, y significar tanto para mí; Mayrolis y Yussel saben que las quiero mucho.

A mis otras amigas, Maggie, Roxana, Yaneisis, Lisbet, siempre han estado presentes, las quiero.

A mis amigos que cuando más los necesitas aparecen con maravillosos consejos, por estar ahí para apoyarme, ayudarme y comprenderme.

A Roexcy por acompañarme en la realización de este trabajo, porque unidos llegamos al final.

A nuestro tutor por todo el apoyo y la ayuda que nos ha brindado, por su paciencia y profesionalismo, muchas gracias.

A todos mis compañeros durante estos 5 años, los que estuvimos juntos desde primer año y los demás.

A mis profesores que tanto han significado en mi formación personal y profesional.

A la revolución por haberme dado esta maravillosa oportunidad.

A todos los que de una forma u otra han contribuido a que este sueño se haga realidad.

A mis padres:

Por guiarme y darme apoyo en todo momento, por la educación incondicional de ambos, por siempre estar ahí cuando los necesité. No hay espacio ni tiempo capaz de describir mi amor y mi agradecimiento hacia ellos.

A mis hermanos por darme tanto apoyo, por confiar en que llegaría al final y por renunciar muchas veces a sus gustos por complacer los míos, por estar ahí y saber que siempre puedo contar con ellos.

A mi abuela Ramona y Maye que no pudieron ver como su nieto y amigo llegaba hasta el final.

A todos en mi familia por el cariño que me han dado, por preocuparse siempre por mí, por guiarme en cada etapa de mi vida.

A mis amigos por comprenderme, soportarme, y significar tanto para mí; Aylin, Zaylí, Yuna, Yadira, Yorjenys, Mercedes, Yanio saben que los quiero mucho.

A mis dos madres UCI Ivón y Marzo y mi padre Ortiz por tantos consejos y apoyo en estos cinco años.

A mis amigos que cuando más los necesitas aparecen con maravillosos consejos, por estar ahí para apoyarme, ayudarme y comprenderme.

A Ivis por acompañarme en la realización de este trabajo, porque unidos llegamos al final.

A nuestro tutor por todo el apoyo y la ayuda que nos ha brindado, por su paciencia y profesionalismo, muchas gracias.

A todos mis compañeros durante estos 5 años, los que estuvimos juntos desde primer año y los demás.

A mis profesores que tanto han significado en mi formación personal y profesional.

A la revolución por haberme dado esta maravillosa oportunidad.

A todos los que de una forma u otra han contribuido a que este sueño se haga realidad.

Roexcy

A mi familia que siguió de cerca mis estudios en la universidad.

A mi abuela que no pudo ver como terminaba mis estudios.

A los amigos que no dejaron de influir en cada momento.

A Fidel y la Revolución cubana.

Roexcy Vega Prieto

A mis padres que son mi razón de ser.

A mis hermanos porque han sido mis segundos padres.

A mi sobrinito, la personita dueña de mi corazón.

A mi tío y mi abuelito que aunque ya no están siempre estarán en mi corazón.

Ivis Cabrera Guerra

Resumen

Actualmente, la investigación y la práctica han demostrado que las búsquedas de archivos audiovisuales son más eficientes si están basadas en su contenido, lo que lo hace una necesidad para este tipo de sistemas. De manera general las búsquedas se hacen tradicionalmente a partir de descripciones de las medias, para almacenar la duración, el título, las extensiones, entre otras características que no logran describir el contenido. Es por ello que la eficiencia de las búsquedas no es la deseada en estos sistemas.

El Polo de Video y Sonido Digital de la Facultad 9 de la Universidad de las Ciencias Informáticas no cuenta con un sistema de búsqueda de archivos audiovisuales por contenido, lo que implica que este proceso resulte complejo. Por esta razón el objetivo fundamental del presente trabajo está centrado en crear una propuesta de un modelo para el procesamiento y la búsqueda de medias siguiendo este criterio.

Se propone el modelo de un sistema capaz de analizar los archivos audiovisuales y extraer las características que describen su contenido, para usarlas posteriormente en el proceso de búsqueda especificada. La propuesta cuenta con los elementos requeridos en materia de procesamiento de medias y para ello utiliza el estándar de descripción de archivos MPEG-7.

Palabras claves:

Archivos audiovisuales, videos digitales, metadatos, estándares.

Abstract

Nowadays, it has been demonstrated by both research and practice that the search for audiovisual files is more efficient if it is based on their content, which makes it a need for this kind of system. In general, these searches are traditionally made from media descriptions, in order to store the length, title, extensions and other features that do not correctly describe the content. This is the reason why those searches do not have the expected efficiency.

The Digital Sound and Video Pole of faculty 9, in the University of Informatics Sciences, does not have a system of searching for audiovisual files based on their content, making it a complex process to carry out. Due to this, the main objective of this research work is to propose a model for processing and searching for audiovisual files following this criterion.

A model of a system is proposed. The system is capable of analyzing the audiovisual files, and extracting the features that describe their content, to use them later in the search process. The proposal contains all the elements required to handle audiovisual files. The standard for file description proposed is MPEG-7.

Keywords: Audiovisual files, digital video, metadata and standards.

Índice

Introducción	1
Capítulo I: Búsqueda de videos digitales.....	6
1.1. Introducción.....	6
1.2. Principales conceptos asociados al dominio del problema.....	6
1.3. Búsqueda de contenidos audiovisuales.	7
1.4. Proceso de búsqueda tradicional.	17
1.4.1. Modelos de recuperación.	21
1.5. Proceso de búsqueda en contenidos audiovisuales.	22
1.6. Resumen del capítulo.	25
Capítulo II: Metadatos en la búsqueda de videos digitales.....	26
2.1. Metadatos.....	26
2.2. Tipos de metadatos.....	28
2.3. Estándares de metadatos.....	28
2.3.1. Estándar MPEG-7.....	29
2.3.2. Estándar Dublin Core.....	36
2.4. Herramientas internacionalmente utilizadas para la elaboración y procesamiento de metadatos de materiales audiovisuales.....	43
2.5. Utilización y procesamiento de metadatos en el polo de Video y Sonido Digital.	44
2.6. Resumen del capítulo.	45
Capítulo III: Propuesta del modelo conceptual del sistema de búsqueda.	46
3.1. Propuesta del modelo conceptual del sistema de búsqueda.....	46
3.2. Propuesta de herramientas a utilizar.	56

3.2.1.	Metodologías de desarrollo de software.	56
3.2.2.	Herramienta de modelado.	59
3.2.3.	Arquitectura de Software	61
3.2.4.	Estilos arquitectónicos.....	61
3.2.5.	Lenguaje de programación para una aplicación Web.....	69
3.2.6.	Lenguaje de programación para una aplicación de escritorio.	74
3.2.7.	Sistema Gestor de Base Datos (SGBD).	77
3.2.8.	Sistema Operativo.	80
3.3.	Resumen del capítulo.	83
	Conclusiones.....	84
	Recomendaciones.....	85
	Referencias Bibliográficas.....	86
	Anexos	91
	Glosario de Términos.....	93

Índice de Tablas

Tabla 1 Atributos con valor fijo del estándar Dublin Core	38
Tabla 2 Atributo título.	38
Tabla 3 Atributo creador.	39
Tabla 4 Atributo materia.	39
Tabla 5 Atributo descripción.....	39
Tabla 6 Atributo editor.	39
Tabla 7 Atributo colaborador.	40
Tabla 8 Atributo fecha.	40
Tabla 9 Atributo tipo.	40
Tabla 10 Atributo formato.....	41
Tabla 11 Atributo identificador.	41
Tabla 12 Atributo fuente.	41
Tabla 13 Atributo lenguaje.	42
Tabla 14 Atributo relación.	42
Tabla 15 Atributo cobertura.....	42
Tabla 16 Atributo derecho.	42

Índice de Figuras

Figura 1 Interfaz de Google Video	13
Figura 2 Interfaz ShadowTV	14
Figura 3 Interfaz de Videoma.....	14
Figura 4 Interfaz MISE	16
Figura 5 Modelo conceptual del sistema de búsqueda propuesto	47
Figura 6 Ejemplo de regiones	49
Figura 7 Ejemplo de distribución espacial	51
Figura 8 Espacio frecuencial.....	52
Figura 9 Ejemplo de bordes	53
Figura 10 Proceso de indexación	54

Introducción

Con el vertiginoso desarrollo de las TIC (Tecnologías de la Información y las Comunicaciones) el mundo comenzó una nueva era, la Era de la Información. Grandes volúmenes de datos recorren todo el planeta y se hace indispensable su manipulación. El uso generalizado de Internet ha provocado que el flujo de información que transita por la red aumente cada día, y en especial aquella información relacionada con archivos multimedia, entre los que se encuentran los videos digitales.

Debido a sus características, los videos digitales han adquirido un lugar importante en la preferencia de los usuarios de Internet y en general de las personas que tienen acceso a él. Su utilización brinda la posibilidad de ir más allá de las capacidades de la televisión tradicional, pueden adquirirlo, manipularlo y almacenarlo. Además, les brinda no sólo imágenes en movimiento, sino también la palabra hablada, el sonido, las texturas de luz y oscuridad, el color y el tiempo.

Sin embargo la búsqueda de videos digitales no está siendo para los usuarios una tarea fácil, pues aunque se hace sencillo realizarla a través de buscadores disponibles en la red, por palabras claves que describen solo el título, el autor, y otras características del video digital que el usuario domina; se presentan problemas a la hora de responder a las necesidades y preferencias específicas de los clientes, ya que los resultados son ineficaces, debido a que la mayoría de los buscadores se limitan a las características antes citadas y no al contenido del video digital (Amengual Galdón, 2006).

Este problema se acentúa más en el caso de las empresas, entidades y grupos de desarrollo dedicados a la producción y/o distribución de materiales audiovisuales, pues a menudo cuentan con gigantescas bibliotecas de videos digitales que muy pocas veces tienen descripciones profundas, lo que impide realizar búsquedas tanto por características de alto nivel (como las antes mencionadas); como por la información de bajo nivel de abstracción (color, texturas, formas,), que solo se pueden obtener a través del procesamiento del video.

Contar con un método eficaz y sencillo de buscar, gestionar, adquirir y recuperar el video digital que se necesite, es el mayor reto que enfrentan las tecnologías y estándares que dan soporte a la creación, desarrollo, codificación, decodificación y descripción de video independientemente de la plataforma.

La manera común de buscar un video digital actualmente es a través de buscadores disponibles en la red, o sistemas que realizan la búsqueda creados por diferentes compañías que se dedican al manejo de la

información, o creados por organizaciones y empresas para las cuales se hace indispensable la gestión de videos digitales.

Existen dos generaciones de buscadores de video según la tecnología que utilizan (Casares, 2003). La primera generación son aquellos en los que la forma de organizar los resultados se basa en los metadatos del video, es decir, los datos que puede incorporar el propio video junto al contenido que hay en la página donde se encuentra (título, descripción, etiquetas, resumen, transcripción, resolución).

La segunda generación aunque también incorpora los elementos anteriores, llevan la tecnología un paso más allá, siendo ellos mismos los que utilizan reconocimiento de voz, análisis visual, reconocimiento del texto que puede aparecer por pantalla.

La mayoría de los buscadores existentes actualmente son del primer tipo, realizan las búsquedas basados solamente en las características que vienen acompañando al video como se describió en la definición de buscadores de primera generación; esto provoca que los resultados arrojados no sean los esperados por los usuarios.

Los del segundo tipo están menos generalizados debido a que su implementación es bastante difícil porque se requiere de robustos métodos de búsquedas, que realicen el proceso en el contenido del video, además se debe contar con un estándar de metadatos que los represente de una forma normalizada. Permiten recuperar un video a través de una imagen, forma, textura o color, así como otras características; por ejemplo, un video que contenga una secuencia del Che realizando un trabajo voluntario, solo especificando una foto de él sin la necesidad de especificar el nombre del fichero.

Un punto de partida esencial para lograr la eficiencia en el funcionamiento de sistemas de búsquedas de videos digitales, son los datos asociados al mismo material, más conocidos como metadatos. En ellos se resume la información del contenido de los videos, lo que permite el desarrollo de sistemas de búsquedas robustos que brinden respuestas rápidas y precisas, sin la necesidad de reprocesar nuevamente el material audiovisual.

Los metadatos de materiales audiovisuales provienen de diferentes fuentes: desarrolladores de contenido, editores, terceras partes o usuarios. Algunos de los metadatos se crean automáticamente y otros se añaden de forma manual; son parte del contenido o externos al mismo.

En la Facultad 9 de la Universidad de las Ciencias Informáticas se encuentra el Polo de Video y Sonido Digital el cual se dedica al desarrollo de aplicaciones para el tratamiento y la gestión de medias. En el mismo la búsqueda de video se realiza mediante del software “*Where is it*”¹, que es un sistema sencillo para la catalogación² y recuperación de ficheros digitales, basándose solamente en características elementales como nombre, extensión, tamaño y fecha de creación de los ficheros, lo cual lo hace completamente inefectivo a la hora de recuperar materiales digitales atendiendo a su contenido.

Esto provoca que se pueda afirmar que el PVSD³ carece de un sistema capaz de recuperar materiales audiovisuales atendiendo a su contenido de una forma rápida y precisa. Adicionalmente el polo no cuenta con un estándar de metadatos definido que permitiría desarrollar un sistema de este tipo.

A partir de la situación planteada se ha identificado el siguiente **problema a resolver**:

¿Cómo diseñar un modelo de un sistema que permita realizar búsquedas de videos digitales mediante la utilización de metadatos?

Para darle solución al problema planteado se definió como **objeto de estudio**: el proceso de desarrollo de Sistemas de Recuperación de Información en archivos de video digital en la Universidad de las Ciencias Informáticas.

El campo de acción de la investigación lo constituyen: los modelos para sistemas de búsquedas de archivos de video digital.

Se plantea como **objetivo general** de la investigación: definir un modelo para desarrollar un sistema de búsquedas de archivos de video digital, basado en la utilización de metadatos.

Entonces se infiere la **idea a defender** siguiente: Con la definición de un modelo, se podrá desarrollar un sistema que permita la recuperación de archivos de videos digitales, partiendo de los metadatos asociados a los mismos.

¹ Donde está.

² Relación ordenada de elementos pertenecientes al mismo conjunto, que por su número precisan de esa catalogación para facilitar su localización.

³ Polo de Video y Sonido Digital.

Para darle cumplimiento al objetivo general se definieron diferentes **tareas de investigación** las cuales se relacionan a continuación.

- Identificar y caracterizar de los principales métodos de búsquedas que se pudieran emplear para el desarrollo de un sistema de búsqueda de archivos de videos digital.
- Caracterizar de los estándares de elaboración de metadatos para la descripción de archivos de video digital.
- Caracterizar las herramientas internacionalmente utilizadas para la elaboración y procesamiento de metadatos de materiales audiovisuales.
- Identificar y describir de los descriptores y etiquetas de los estándares de metadatos a emplearse en el proceso de búsquedas.
- Definir y describir una propuesta de modelo para el desarrollo de un sistema de búsqueda basado en la utilización de metadatos en archivos de video digital.
- Definir y caracterizar una propuesta de herramientas para el desarrollo del modelo conceptual propuesto. .

Para darle cumplimiento a las tareas especificadas se emplearán los siguientes **métodos científicos**:

Métodos teóricos.

- **Analítico – sintético:** Cuando se busca lo principal de los procesos que conforman un buscador y para determinar las características de los metadatos, a partir de las diferentes teorías que numerosas personalidades han planteado.
- **Análisis histórico – lógico:** Durante el estudio de la evolución que han tenido los diferentes sistemas relacionados con la búsqueda de videos a través de metadatos.

Métodos empíricos.

- **Entrevista:** Se emplea con el objetivo de conocer como se realiza la búsqueda de video en el polo de Video y Sonido Digital. La población estará constituida por 3 profesores. La muestra escogida estará representada por el total de la población.

Para facilitar su comprensión, el documento está estructurado en tres capítulos:

Capítulo I: Búsqueda de videos digitales. Fundamentos Teóricos, se esclarecen conceptos que sustentan la investigación. Se hace un análisis sobre el proceso de búsqueda través de sus tendencias actuales, haciendo énfasis en el proceso tradicional, que está basado en texto y en la nueva forma de búsqueda, basada en el contenido. Se exponen puntos importantes, que tienen repercusión directa en la creación de la propuesta.

Capítulo II: Metadatos en la búsqueda de videos digitales, se realiza un estudio sobre los estándares de metadatos existentes para describir contenidos audiovisuales, destacando el estándar MPEG-7 y Dublin Core, indagando además sobre la existencia en el Polo de Video y Sonido Digital de la Facultad 9 de un sistema para la búsqueda de videos digitales basado en metadatos. Los resultados del estudio muestran un mejor dominio de la situación problemática Se efectúa un análisis sobre los estándares antes mencionados que conlleva a proponer el estándar MPEG-7 para el desarrollo de un sistema de búsqueda de videos digitales a través de sus metadatos en el Polo de Video y Sonido Digital.

Capítulo III: Propuesta del modelo conceptual del sistema de búsqueda, se muestra la propuesta efectuada teniendo en cuenta los resultados obtenidos en la investigación, proponiendo así un modelo de un sistema de búsqueda de archivos audiovisuales a través del estándar para la descripción de metadatos MPEG-7, el cual posibilitará el desarrollo de un sistema de búsqueda de videos digitales a través de los metadatos de los mismos en el Polo de Video y Sonido Digital.

Capítulo I: Búsqueda de videos digitales.

1.1. Introducción.

En el capítulo se esclarecen conceptos importantes que sustentan la investigación. Se realiza un análisis sobre el sistema de búsqueda de videos digitales, a través de sus tendencias actuales. Se exponen puntos importantes sobre las principales directrices en la realización de sistemas de búsqueda de videos digitales, que tienen repercusión directa en la creación de la propuesta del modelo.

1.2. Principales conceptos asociados al dominio del problema.

Contenidos audiovisuales: lo constituyen cualquier producción que combine imágenes y sonidos, como por ejemplo: películas, grabaciones de video.

Video digital: es una secuencia de imágenes y audio que son almacenadas y reproducidas en forma digital (López Guzmán, 2000). El video hace referencia a la captación, procesamiento, transmisión y reconstrucción de una secuencia de imágenes y sonidos que representan escenas en movimiento. Está compuesto por fotogramas, que son cada una de las imágenes individuales de una secuencia o animación; por tomas, que son las secuencias de imágenes realizadas en un mismo plano y por escenas, que representan un conjunto de tomas relacionadas entre sí.

Recuperación de información: se define a través de la representación, almacenamiento, organización y su posterior acceso y recuperación para responder a las necesidades de un usuario. Es la disciplina encargada de la búsqueda de información en documentos, búsqueda de metadatos que describen un documento, o la búsqueda en base de datos de textos, sonidos, imágenes, videos.

Metadatos: son objetos que describen o proporcionan información sobre otros datos (Paulus Velázquez, 2005), se pueden definir además, como “datos sobre datos” (INFLANET, 1998)

Buscador o motor de búsqueda (también llamado “*Searchbot*”): Es una herramienta usada para localizar los documentos en un repositorio por medio de palabras claves o una clasificación temática. Algunos requieren el uso de los operadores booleanos (*or, and, not*), los más avanzados permiten la interrogación en lenguaje natural. Mayormente son utilizados en sitios Web de Internet para ser empleados por los usuarios conectados a la red, con la necesidad de adquirir algún servicio o información; aunque también existen los buscadores realizados para trabajos específicos, en los que los trabajadores necesitan de una

herramienta que sea capaz de brindarle el acceso a la información con la que cuentan, de una manera más rápida y eficiente.

Metabuscador: Sistemas que realizan el proceso de búsqueda en diferentes buscadores.

Indexación: Operación de registrar de forma ordenada la información para confeccionar un índice.

1.3. Búsqueda de contenidos audiovisuales.

El concepto de búsqueda ha estado presente desde los comienzos de la informática y ha ido evolucionando hasta los días actuales. Los primeros algoritmos de búsqueda que fueron creados, trataban de localizar de la forma más rápida posible los datos almacenados dentro de una base de datos local.

Con los avances de la informática se descubrió rápidamente que mediante la indexación de estos datos el proceso de búsqueda podía ser acelerado considerablemente. De esta forma, surgieron los primeros motores de búsqueda, cuyo proceso consistía en facilitar la localización de contenidos dentro de bases de datos. Estos buscadores procesaban únicamente datos textuales y no eran capaces de realizar una adquisición activa de contenidos.

La aparición de las bases de datos distribuidas, unido a la aparición de Internet, propiciaron la creación de un nuevo concepto de buscador: sistema que de forma activa es capaz de explorar e indexar los contenidos independientemente de que se haya solicitado su búsqueda.

Así surge el concepto de buscador existente en la actualidad: un sistema informático que indexa archivos almacenados en servidores, como, por ejemplo, los buscadores de Internet (Siri, 2000).

Actualmente, estos buscadores tienen el desafío de enfrentarse a los contenidos audiovisuales. La cantidad de contenidos de video y audio se incrementa cada día, y éstos no pueden ser tratados del mismo modo que la información textual; los contenidos audiovisuales son difíciles de analizar y de comprender por un programa informático, mucho más que el texto. Estas nuevas situaciones necesitan esfuerzos de adaptación por parte de los motores de búsqueda.

Los motores de búsqueda se dividen en tres categorías: (García de León, et al., 2001)

- Directorios de materias.
- Metabuscadores.
- Motores de búsqueda individuales.

Directorio

Es una herramienta que proporciona una organización temática jerárquica⁴ con la cual se permite recorrer los recursos de información de Internet; es decir, serían catálogos con una ordenación temática. Bajo cada categoría o sub-categoría, se listan los enlaces de los documentos que corresponden a un tema. No admiten la polijerarquía⁵, por lo tanto, cada página Web se clasifica en un único epígrafe, trasladando de esa forma, el problema de la “ubicación física” de los documentos a la red.

Cuando se clasifica un documento para establecer su lugar en una colección, aunque el sistema utilizado acepte la relación temática, deberá tener una ubicación única. Sin embargo, cuando se habla de la red se entiende como una respuesta simplificadora a una tarea bien compleja. Para colaborar aún más con esa simplificación se trabaja sólo con las páginas de inicio. La clasificación es efectuada por el autor o los autores del documento, o por los administradores del directorio, por lo tanto, se trata de una clasificación manual.

A pesar de esta debilidad, pueden ser muy útiles en ocasiones. Lo importante es reconocer sus características para saber cuándo puede servirle al usuario. Los directorios son adecuados cuando no se tiene una idea precisa de lo que se busca, o cuando interesa buscar por grandes temas, para conocer el estado de una disciplina, o el tratamiento que un tema recibe en Internet.

Algunos directorios poseen también mecanismos propios de búsqueda por palabra clave, otros remiten la búsqueda que no pueden resolver por su condición de directorios, a un buscador que la resuelve sin que el usuario cambie de interfaz.

Algunos ejemplos de directorios:

- Yahoo <<http://www.yahoo.com>>.
- Looksmart <<http://www.looksmart.com/>>.
- Ozú <<http://www.ozu.es/>>.

⁴ Orden de los elementos de una serie según su valor.

⁵ Relación que se establece entre un concepto específico y dos o más que le son genéricos.

Metabuscadores o agentes multibuscadores

Los metabuscadores son estructuras que permiten remitir una búsqueda hacia varios buscadores en forma simultánea. Los metabuscadores no disponen de una base de datos propia, sino que utilizan la información almacenada en las bases de datos de otros buscadores y directorios. En la gran mayoría de los metabuscadores, los usuarios pueden seleccionar los buscadores en los que quieren que se efectúe la búsqueda.

Algunos ejemplos de metabuscadores:

- Meta Crawler <<http://www.metacrawler.com/>>.
- Dogpile <<http://www.dogpile.com>>.
- Mamma <<http://www.mamma.com/>>.

Buscadores

Los buscadores son programas computacionales que recorren Internet examinando la información de acceso público en la red para su indización⁶ y almacenamiento; con este material se generan bases de datos en constante actualización, que permiten su interrogación por palabra clave para la recuperación de la información. En los buscadores la indización es realizada por los robots que son programas que rastrean la red sin pausa para identificar los recursos de dominio público e indizarlos. Esta información pasa a formar parte de la base de datos del buscador y una vez interrogados presentan una lista de enlaces (*links*), muchas veces anotada, que llegan a incluir la extensión del documento y grado de pertinencia. La estructura de los buscadores varía de acuerdo con el tipo de indización, el tamaño de su índice, la frecuencia de actualización, las opciones de búsqueda, el tiempo de demora para su respuesta, la presentación de los resultados y las facilidades de uso.

Es por este motivo que una misma búsqueda podrá arrojar resultados diferentes en los distintos buscadores. Si bien todos comparten los rasgos necesarios para entrar en esta categoría de herramientas, las diferencias entre ellos pueden ser muy grandes en algunos casos.

⁶ Ordenación de una serie de datos o informaciones de acuerdo con un criterio común a todos ellos, para facilitar su consulta y análisis.

Generalmente, ofertan la información recogida en sus propios sitios Web, donde pueden ser interrogados por los usuarios, aunque también es posible llegar a sus servicios a través de aquellos portales que los adquieren. Las capacidades de recolección de información, así como de interrogación y suministro de respuestas han variado con el tiempo y varían de acuerdo con las potencialidades de los diferentes buscadores. Inicialmente, recogían pocos datos de las páginas que visitaban, como título y primeras palabras del texto, más tarde comenzaron a leer los metadatos y considerarlos para la recuperación de la información y su descripción, actualmente algunos llegan a indizar el texto completo.

Si bien existe un amplio abanico a la hora de indizar, no ocurre lo mismo en cuanto al criterio para declarar la pertinencia de una referencia, pues para la mayoría de los motores de búsqueda este es similar. Tomada del título y las primeras palabras, de los metadatos o del texto completo, básicamente los buscadores indizan por el número de ocurrencias de un término y devuelven las referencias a los documentos por un sistema de *ranking* (originalmente se presentaban sin ningún tipo de orden).

Los motores de búsqueda que recorren la red están compuestos principalmente por tres partes (Montaño, et al., 1999):

- Los robots que recorren la red escrutándola.
- El motor de búsqueda que facilita la consulta a la base.
- La base de datos o índice.

Los robots son programas encargados de visitar servidores de la red Internet y, fundamentalmente en los servidores Web, extraer de los recursos que contienen aquella información necesaria para la formación o actualización de sus índices. Se ocupan de actualizar la información de la base de datos, de forma constante. Algunos motores de búsqueda permiten que se les envíe una dirección URL para posteriormente ellos visitarla e incorporarla a su base de datos.

La interfaz permite consultar la base de datos y representar en pantalla los datos y resultados de la búsqueda. La base de datos cuenta con sus propios programas de gestión que almacenan toda la información indizada, lista las direcciones Web, títulos, cabeceras, palabras, primeras líneas, *abstracts*, partes del texto o el texto completo.

Según su contenido, los motores de búsqueda pueden ser distinguidos entre motores de carácter general, que indizan recursos de toda la Web, y motores especializados, que se ocupan de formar sus bases de datos con recursos que tratan sobre un tema determinado.

Los motores de búsqueda estadísticos se caracterizan por la utilización de algoritmos⁷ que dan peso o importancia a las palabras en función de la rareza en el índice, la frecuencia del término en el documento, la posición de la palabra y la proximidad de los términos de búsqueda en el texto. El resultado se añade al cálculo de relevancia de una búsqueda y de este modo se establece el orden de aparición de los registros.

Los motores de búsqueda construyen índices con los recursos de la red. El software que contienen les permite visitar un gran número de servidores, extraer información para registrar en sus bases de datos y actualizarla si ha sufrido variaciones desde la anterior visita. Su análisis se centra tanto en la información del recurso, como de los enlaces de hipertexto que pueda contener dicho recurso.

Determinados motores resumen el contenido informativo de los recursos y extraen palabras claves, mientras que otros recogen la información de etiquetas Meta que contienen los términos de carácter descriptivo más interesantes para la localización del recurso. (Vidal Bordés, 2001)

Por lo general, los motores de búsqueda se basan en texto; es decir, para poder rastrearse e indexarse, su contenido debe estar en formato de texto, por lo que existen pocos motores de búsqueda para otros archivos como videos digitales, los cuales necesitan una información más precisa para responder a la petición de búsqueda realizada por el usuario de una manera más eficiente.

Para lograr esta eficiencia en la respuesta al usuario, los motores de búsqueda deben estar centrados en realizar la búsqueda por contenido, que aunque hasta el momento no tiene una gran utilización debido a constituir un campo de exploración relativamente joven, es el que mejores resultados aporta teniendo en cuenta su capacidad de respuesta.

Para los motores de búsqueda que realizan la exploración en la red a través de páginas Web de solo texto, la búsqueda por contenido le resulta, aunque un poco difícil para los realizadores del motor, bastante fácil para el usuario que haga la petición de búsqueda, la respuesta será con más certeza que la que podría dar un motor que realice la búsqueda normalmente por el nombre de la página o por palabras

⁷ Es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema.

claves, debido a que la mayor parte de los buscadores sólo reconocen texto, por lo que se hace más difícil generar una identificación automática de las características de recursos tales como las imágenes digitales, o los diferentes contenidos audiovisuales, entre ellos, los videos.

Para los materiales audiovisuales, la búsqueda por contenido no está totalmente desarrollada, debido a que no es una tarea fácil para los desarrolladores de buscadores. La realización de sistemas que analicen el contenido de estos archivos, sería la manera más segura de responder apropiadamente a la petición del usuario, este definiría la búsqueda, con las características específicas del archivo, las cuales el sistema concebiría como metadatos; estos contienen los términos descriptivos relevantes para la localización del fichero, con lo cual se respondería a la solicitud de acuerdo a lo especificado.

Los sistemas que realizan las búsquedas de archivos audiovisuales como videos, lo hacen basados en el contenido, en varios casos en el histograma, y en otros en el texto que acompaña a las imágenes, en ocasiones se asocia el texto del guión de una película o serie de televisión con las imágenes u otro archivo de video digital, típicamente a nivel de plano o escena, según se pretendan implementar posteriormente las búsquedas. Tener una buena descripción del video hace posible desarrollar sistemas dotados de una mayor inteligencia que los clásicos mecanismos de búsqueda.

Un factor importante en la aparición de nuevas aplicaciones de este tipo será la adopción de estándares para el desarrollo del contenido audiovisual. (Navarrete, et al., 2003)

Pero estos estándares no contemplan ni el análisis de la imagen, ni el sistema de conocimiento para la búsqueda. Se requiere, por tanto, abordar la tarea de segmentación del material audiovisual, la obtención de *key-frames*⁸ y la extracción de contenido semántico. El primer paso en el análisis del video digital consiste en su segmentación temporal (inicio y final de cada plano) para poder construir índices que permitan localizar las imágenes del plano de un video. (Vilches, 2001)

Existen varios sistemas que dentro de sus servicios utilizan motores de indexación que aunque no utilizan sistemas que realicen las búsquedas por contenidos específicos como los señalados anteriormente si realizan una indización de videos digitales.

Google Video(Cataldi, 2000)

⁸ Son los fotogramas claves dentro del video.

Esta es la propia solución de Google (todavía Beta) para indexar y brindar acceso a la búsqueda de contenido de video. Para realizar la búsqueda se basa en palabras claves. Google indexa un número limitado de canales de transmisión comerciales y también videos clips. Google Video busca la toma que se aproximan más y las descripciones de texto de todos los videos en archivo para resultados relevantes. Disponible en <http://video.google.com/>.

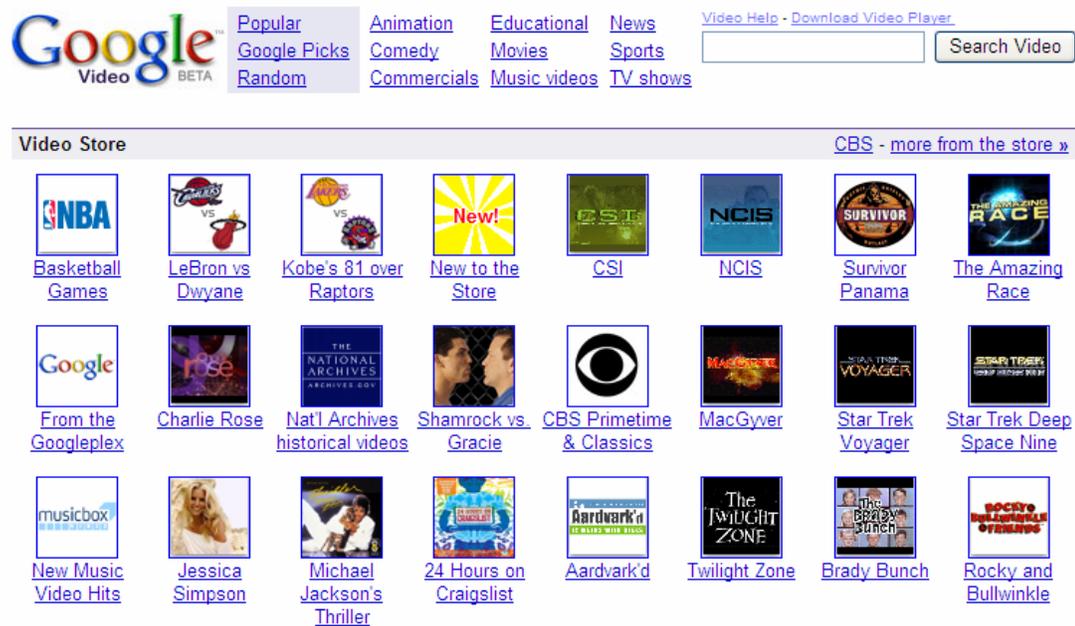


Figura 1 Interfaz de Google Video

ShadowTV (Cataldi, 2000)

ShadowTV Webcasting permite a los emisores instantáneamente convertir material del video archivado y en vivo en contenido digital de video para su sitio Web que permite ser buscado. Los emisores pueden explotar su propio contenido de video o sindicarlo para otros sitios. Consultar en <http://www.shadowtv.com/>.

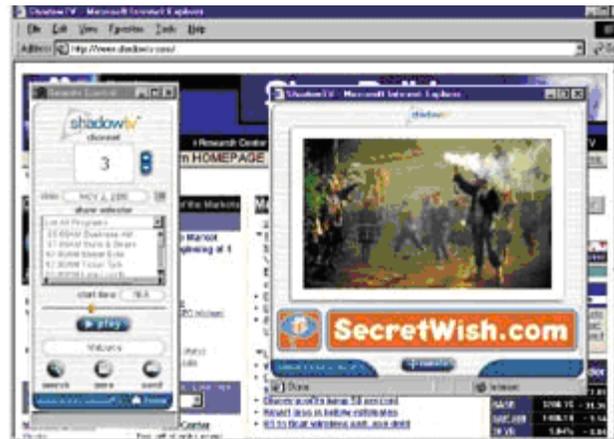


Figura 2 Interfaz ShadowTV

BlinkxTV (Cataldi, 2000)

A diferencia de otros proveedores de búsqueda, blinkx.tv no solamente le permite buscar utilizando palabras clave y consulta Booleana sino que también puede utilizar búsqueda conceptual. Este tipo de búsqueda solamente es brindada por blinkx.tv y le permite ingresar texto normal para el cual blinkx.tv le va a devolver resultados cuyo contenido sea conceptualmente similar a su texto de búsqueda.

Videoma

Consiste en una herramienta de ingesta, administración y recuperación de activos digitales, principalmente el video. (Echemendía Tourt, et al., 2008)



Figura 3 Interfaz de Videoma

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.

Videoma utiliza las siguientes tecnologías para la realización del sistema (Echemendía Tourt, et al., 2008):

- 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 XML⁹ (*Extensible Mark-up Language*), que es un lenguaje que utiliza para la formalización de esquemas de descripción y descriptores.
- El W3C SMIL Integración multimedia sincronizada de lenguaje (*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 polimorfismo. 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.

⁹ Metalenguaje de definición de documentos estructurados mediante marcas o etiquetas.

MISE

El laboratorio *Murray Hill*¹⁰ del grupo de búsqueda de *Mitsubishi Electric*¹¹, 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. (García Albacete, 2002)



Figura 4 Interfaz MISE

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.

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.

¹⁰ <http://www.murrayhillinn.com/>.

¹¹ <http://www.mitsubishielectric.es/>

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.

Basado en archivo de video digital MPEG-7 (*MPEG-7 Based Archival of Digital Video*)

Entre Noviembre del 2002 y Marzo del 2005, BMWA (*German Federal Ministry of Economics and Labour*¹²) 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. (Vetro, 2008)

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.

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 pues 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*¹³ de audio.

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

1.4. Proceso de búsqueda tradicional.

Un motor de búsqueda que realice su proceso en archivos audiovisuales, basado en el contenido de los elementos en los cuales realizará su búsqueda, requiere de un proceso mediante el cual se logren satisfacer las necesidades y la petición expuesta por el usuario.

¹² <http://www.bmwa.gv.at/>

¹³ Es un protocolo de comunicación para audio, video y datos especificado en los estándares de MPEG

El desarrollo de un sistema de búsqueda comienza por medio del procesado de los contenidos a indexar. Para ello, la primera etapa consiste, por un lado, en un análisis y extracción de características y, por otro, en una etapa de anotación semiautomática, siendo el objetivo de ambos bloques la realización de un análisis profundo, también a nivel semántico, de los contenidos. Consecutivamente, la etapa de gestión del conocimiento genera las descripciones de los contenidos a partir de los resultados previos, y los concentra por medio de ontologías y otras estructuras semánticas. A continuación, cuando se tienen los contenidos y sus descripciones, el sistema puede realizar su procesado para ofrecer nuevas funcionalidades y servicios. Las siguientes etapas, forman parte de lo que son los buscadores clásicos. El motor de búsqueda realiza el recorrido entre los contenidos indexados para ofrecer aquellos resultados que concuerdan con las peticiones del usuario; los bloques de la parte de la consulta, son los encargados de traducir y procesar la orden de búsqueda del usuario; las medidas de relevancia evalúan el nivel de validez de los aciertos ofrecidos por el motor de búsqueda; y, por último, el bloque de distribución y presentación, constituye la interfaz con la cual el usuario interactúa con el motor y donde se le presentan los resultados.

La indexación, como se indicó anteriormente es el primer paso de un sistema de búsqueda, una vez procesada la base de datos o el servidor donde se encuentra el contenido dejándola en un formato cuya manipulación por parte del sistema sea fácil y rápida. A partir de un archivo procesado, se generará una representación del mismo, formada por una secuencia de términos de indexación, los cuales mantendrán lo más fielmente posible el contenido original del archivo.

Para realizar la indexación se necesitan términos específicos entre los que se encuentran, la frecuencia de aparición de una palabra, eliminación de las palabras vacías. Siendo estos los de mayor importancia, se plantean a continuación. (Fernández Lunas, 2001)

La frecuencia de aparición de una palabra en un texto determina su importancia en él, sugiriendo que dichas frecuencias pueden ser utilizadas para extraer palabras con objeto de resumir el contenido de un documento. Si se obtiene la frecuencia de aparición f , de cada palabra de un texto y se ordena decrecientemente, siendo p la posición que ocupa en dicha ordenación, se cumple que $f \cdot p = c$, donde c es una constante.

Si se representa gráficamente esta curva (p en el eje X, y f en él Y), se obtiene una hipérbola, en la cual se pueden establecer dos límites en cuanto a p se refiere, todas las palabras que excedan el superior, se

considerarán muy comunes (haciendo búsquedas por ellas se podrían recuperar casi todos los documentos), y todas las que estén por debajo del inferior, muy raras. Las que queden dentro de ambos límites serán las que tengan una mayor capacidad para discriminar el contenido de un texto y, por tanto, las que deban ser usadas. El problema radica en establecer los dos límites anteriores, porque, la eliminación de palabras con frecuencias muy altas puede provocar una reducción de la exhaustividad, el uso de conceptos generales es útil a la hora de recuperar muchos documentos relevantes. Por el contrario, el descartar términos con una frecuencia baja, produce pérdidas en la precisión.

Otro aspecto a tener en cuenta a la hora de seleccionar los términos consiste en eliminar las palabras vacías de significado, como pueden ser artículos, preposiciones, conjunciones, incluso en algunos casos, se pueden calificar así algunos verbos, adverbios y adjetivos. Por tanto, estas palabras no sirven como términos de indexación, por un lado son muy frecuentes, y por otro no representan correctamente el contenido del documento. La acción normal que se lleva a cabo con ellas es su eliminación del texto, proceso que se conoce como eliminación de palabras vacías, y se pone en práctica mediante la comparación de cada palabra del texto con un diccionario que contiene la lista de palabras no aptas para la indexación.

Este proceso de selección pasa por determinar la importancia de un término en el documento, de tal forma que, si es lo suficientemente importante, se escogerá para ser incluido en el conjunto de términos final. El cálculo de la importancia de cada término se conoce como ponderación del término.

Este se basa en contar las ocurrencias de cada término en un documento, medida que se denomina frecuencia del término i -ésimo en el documento j -ésimo, y se nota como tf_{ij} . El problema que puede aparecer es que, independientemente del número de veces que aparezca el término “computadora” en los documentos de una colección sobre Informática, no sería un buen término para asignárselo a ellos.

Una segunda medida de la importancia del término es la conocida como frecuencia documental inversa de un término en la colección, conocida normalmente por sus siglas en inglés *idf* (*inverse document frequency*), y que responde a la siguiente expresión:

$$idf_i = \log \frac{N}{n_i} + 1$$

Donde N es el número de documentos de la colección, y n_i el número de documentos donde se menciona al término i -ésimo. Como se puede observar, el valor obtenido por la expresión decrece conforme n_i crece, variando desde $\log N + 1$ cuando n_i es 1, a 1 cuando n_i toma valor N . Por tanto, cuantas menos veces aparezca un término en la colección, más alto será su *idf*, dando así una forma de medir la calidad global del término en toda la colección. El hecho de introducir un logaritmo se justifica para suavizar el crecimiento del tamaño de la colección. Lo ideal sería combinar ambas medidas anteriores utilizando un esquema de ponderación que permita identificar a los términos que aparecen bastante en varios documentos individuales, y a la vez, que se hayan observado en contadas ocasiones en la colección completa. Estos son los términos que tendrán una capacidad de discriminación mayor con respecto a los documentos en los que aparecen. O lo que es lo mismo, calcular un peso que fuera proporcional a la frecuencia del término i -ésimo en el documento j -ésimo, e inversamente proporcional al número de documentos de la colección completa en los que aparece ese término. Así, el peso final asignado al término i -ésimo en el documento j -ésimo, que se denotará como *tfidf*, corresponde al producto: $t f_i * i d f_i$

En este caso, la importancia crece con respecto a la frecuencia del término en el documento y disminuye con respecto al número de documentos que lo contienen. Cuanto más alto sea el valor, mejor será el término desde el punto de vista de la indexación.

El último paso es extraer la raíz morfológica de cada palabra, eliminando sufijos y prefijos, originando así que el sistema de búsqueda pueda recuperar archivos incluyendo variantes morfológicas de los términos contenidos en la consulta, mejorando la recuperación, a la vez que ahorra espacio al almacenar sólo las raíces.

Por último, se debe poner en práctica un proceso de reconocimiento de raíces equivalentes, con objeto de evitar confusiones con palabras que poseen la misma raíz, pero no están relacionadas en su significado. A partir de este punto, al hablar de términos de indexación, se hará referencia a las raíces morfológicas, en lugar de a las palabras completas.

Una vez que ha finalizado el análisis automático de la base de datos, un aspecto importante es su organización para conseguir un acceso eficiente y rápido en las operaciones que se realizarán posteriormente en el proceso de recuperación. Así, se conoce como *fichero invertido* a una estructura de datos que almacena de manera ordenada todos y cada uno de los términos del glosario y, para cada uno de ellos, guarda la lista de documentos donde aparece, junto con su peso asociado.

Cuando se efectúa una consulta al sistema, ésta es pasada también por el módulo de indexación para conseguir su correspondiente representación. Dependiendo del modelo de recuperación utilizado, la consulta podrá ser una expresión booleana, formada por los términos y conectivos lógicos, o una lista de palabras, con sus correspondientes pesos.

1.4.1. Modelos de recuperación.

Existe una gran cantidad de modelos de recuperación basados en tecnologías muy diferentes. Seguidamente se hace referencia a los diferentes modelos existentes actualmente. Específicamente se tratará los tres clásicos: el booleano, el del espacio vectorial y el probabilístico.

El modelo booleano.

El modelo booleano está basado en la teoría de conjuntos y en el álgebra booleana. Su marco está compuesto por los documentos representados como conjuntos, las consultas, como expresiones booleanas (términos conectados por los conectivos booleanos *Y*, *O*, y *NO*), y las operaciones existentes para tratar conjuntos: unión, intersección y complemento.

Los pesos de los términos en los documentos son binarios: 0 indica ausencia y 1 presencia. Así, en este modelo, dada una consulta al sistema, se va evaluando la expresión booleana mediante la realización de las operaciones anteriores con los conjuntos formados por los documentos donde aparece cada término de la consulta. El conjunto de documentos resultante está compuesto por todos aquéllos que hacen verdad la consulta booleana. Es por esto que su clasificación está enmarcada dentro de los modelos de emparejamiento exacto.

El modelo del espacio vectorial.

En el modelo del espacio vectorial, el marco está compuesto por el espacio vectorial de dimensión M (cada dimensión equivale a un término distinto del glosario), representando en él los documentos, las consultas y las operaciones algebraicas sobre los vectores de dicho espacio. Concretamente, la función que obtiene la similitud de un documento con respecto a una consulta se basa en la medida del coseno, la cual devuelve el coseno del ángulo que forman ambos vectores en el espacio vectorial.

El modelo probabilístico.

El marco del modelo probabilístico está compuesto por conjuntos de variables, operaciones con probabilidades y el teorema de Bayes (Benavides, et al., 2001). Todos los modelos de recuperación probabilísticos están basados en el que se traduce como el Principio de la Ordenación por Probabilidad. Este principio asegura que el rendimiento óptimo de la recuperación se consigue ordenando los documentos según sus probabilidades de ser juzgados relevantes con respecto a una consulta, siendo estas probabilidades calculadas de la forma más precisa posible a partir de la información disponible.

1.5. Proceso de búsqueda en contenidos audiovisuales.

Para desarrollar satisfactoriamente el proceso de búsqueda en archivos audiovisuales el motor requiere utilizar métodos de búsqueda basados en el contenido, que garanticen realizar las funcionalidades especificadas a continuación. Entre los principales métodos de búsqueda se encuentran (Fernández Lunas, 2001):

- Comparación por el color.
- Comparación por Textura.
- Comparación por Formas.
- Extracción del texto presente en las imágenes.
- Proceso de descripción de las escenas automáticamente detectadas.

Se relacionan los métodos teniendo en cuenta la definición de video digital que se especificó anteriormente, donde se destaca que un video está formado por una secuencia de imágenes, siendo posible así realizar búsquedas en videos digitales a través del contenido de las imágenes y de las operaciones de búsquedas que se emplean en estos archivos.

Comparación por el color: esta técnica no depende del tamaño de la imagen o de su orientación. La búsqueda basada en el color se ejecuta comúnmente comparando los histogramas de color de las imágenes.

Comparación por Textura: la medida de las texturas busca patrones visuales en las imágenes y trata de encontrar la forma en que se define el espacio. La textura se representa mediante texels¹⁴ que luego se

¹⁴ Unidad mínima de una textura aplicada a una superficie.

colocan en una serie de conjuntos dependiendo de cuantas texturas se han detectado en la imagen. Estos conjuntos definen además de las texturas, en qué punto de la imagen se encuentra esa textura.

Comparación por Formas: se refiere a la forma de una región en particular. Las formas generalmente vienen determinadas aplicando primero el método de segmentación o la definición de los bordes de una imagen. En determinados casos una cuidada detección de las formas requiere intervención humana porque los métodos como la segmentación son muy difíciles para automatizarlos.

Extracción del texto presente en las imágenes (Echemendía Tourt, et al., 2008): en una secuencia de video se pueden encontrar dos tipos de texto embebido: *captions*¹⁵(*subtítulos*), y los textos propios de la escena, que aunque pueden proporcionar información semántica de la secuencia, la proporcionada por los *captions* es considerablemente mayor y, por lo tanto, más útil a la hora de indexar el video. Algoritmos desarrollados permiten extraer el texto embebido en las imágenes (Echemendía Tourt, et al., 2008), lo cual puede utilizarse para realizar una búsqueda que tendría importantes resultados. Se pueden identificar dos tendencias fundamentales en las técnicas o métodos vinculados a la detección de texto artificial contenido en fotogramas de videos en el dominio no comprimido. Estos son los métodos basados en detección de bordes y los basados en análisis de textura.

Los algoritmos basados en detección de bordes explotan características bien definidas en las líneas de texto artificial. Los *captions* presentan generalmente alineación horizontal en los fotogramas, con una alta densidad de trazos verticales y los caracteres de una misma línea de texto o de una palabra son del mismo tamaño y se encuentran homogéneamente espaciados; además de poseer un alto contraste en relación al fondo del fotograma.

Los métodos de detección basados en análisis de textura manipulan el texto artificial como un tipo de textura bien definida y diferenciable del resto de la imagen. Los *captions* poseen cierta información de frecuencia, orientación y cohesión espacial que los hace identificables mediante la segmentación por textura.

¹⁵ Son un conjunto de caracteres alfanuméricos que se encuentran superpuestos o incrustados en los fotogramas de videos digitales. Representan una versión textual de diálogos o narraciones que se producen en las escenas o simplemente brindan información.

Proceso de descripción de las escenas automáticamente detectadas: una escena se refiere a la grabación continua de uno o más *frames*¹⁶ (cuadros) que representan una acción continua en el tiempo y el espacio y así constituye una unidad válida para organizar la información acerca de un video. Un video editado está compuesto de dos tipos de límites entre escenas, también conocidos como cortes (*cuts*). Estos son: cortes bien definidos (*straight cuts*) y cortes graduales. Los cortes bien definidos se caracterizan por un cambio de escena repentino o de cámara. La separación de tomas con cortes bien definidos es relativamente fácil. Los cortes graduales, que hacen transiciones graduales entre dos escenas, son introducidos en un video a través de equipo especial de edición y abarca varios *frames*. Las tomas con cortes graduales son, por lo general, difíciles de detectar por la naturaleza gradual de la transición entre una toma y la siguiente. Normalmente la estrategia utilizada para la detección de escenas es el procesamiento de los datos puros (sin comprimir o realizando descompresión), es decir, cuadro a cuadro y píxel¹⁷ a píxel, comparando pares de imágenes consecutivas y verificando con algún método que dichas imágenes son diferentes de manera significativa, para determinar si existe o no cambio de escena.

Existen también métodos de búsquedas clásicos, los cuales se basan en el análisis semiautomático realizado a los videos para extraer las características básicas de los mismos. Estos métodos son los más utilizados en el proceso de búsqueda de videos digitales, aunque no arrojan resultados precisos, por su simplicidad no permiten realizar una búsqueda eficiente. Es por este motivo que surgieron los métodos de búsqueda por contenido, aunque se siguen utilizando unido a ellos los métodos clásicos: Búsqueda por nombre, descripción, autor, y formato.

Búsqueda por nombre: se implementa un método de búsqueda por el nombre, o título del video.

Búsqueda por descripción: este método se implementa semejante al utilizado para la búsqueda de documentos textuales, pues la descripción del video no es más que un fragmento de texto relacionado con el contenido del video.

Búsqueda por autor: se implementaría la búsqueda por el autor o los autores del video.

¹⁶ Es un fotograma o cuadro, una imagen particular dentro de una sucesión de imágenes que componen una animación. La continua sucesión de estos fotogramas producen a la vista la sensación de movimiento, fenómeno dado por las pequeñas diferencias que hay entre cada uno de ellos.

¹⁷ Cada uno de los puntos que componen una imagen digital.

Búsqueda por formato (avi, mpg, wmv, rmw): también es un método de búsqueda semejante al utilizado en documentos textuales, se realiza a través del formato del video, el cual el usuario puede especificar en su petición.

1.6. Resumen del capítulo.

En este capítulo se realizó una conceptualización de los principales temas asociados al dominio del problema (videos, buscador, metabuscador); los cuales se emplearan en la propuesta del modelo a presentar. Se hace un análisis del desarrollo del proceso de búsqueda de contenidos audiovisuales, aportando a la propuesta elementos importantes para desarrollar un sistema de búsqueda eficiente y preciso en la capacidad de respuesta. Se definen los principales métodos de búsqueda, entre los que se encuentran el reconocimiento facial, búsqueda por color, búsqueda por textura, por forma, además de los métodos tradicionales como el probabilístico, el booleano y el espacio vectorial. Se analizan los principales sistemas de búsqueda de videos digitales existentes que de una forma u otra realizan el proceso de búsqueda a través de los metadatos.

Capítulo II: Metadatos en la búsqueda de videos digitales.

En el capítulo se esclarecen conceptos que sustentan la investigación. Se realiza un análisis sobre el empleo de metadatos en la búsqueda de videos digitales, a través de sus tendencias actuales. Se exponen puntos importantes sobre la selección de los estándares de metadatos, que tienen repercusión directa en la creación de la propuesta del modelo conceptual del sistema de búsqueda. Además, en este capítulo se plantea un estudio sobre las acciones que se acometen para la utilización de metadatos en la búsqueda de videos digitales en el polo, indagando sobre la existencia y/o empleo de un sistema de búsqueda con estas características. Dicha investigación se basa en la utilización de entrevistas a los líderes del polo. Los resultados obtenidos permiten mostrar un mejor dominio de la situación problemática.

2.1. Metadatos.

Aunque el término metadatos es relativamente nuevo, lo que representa como tal no lo es, desde los años 60 con la intención de facilitar el trabajo a los investigadores y usuarios de las bibliotecas internacionales se empezaron a compartir metadatos descriptivos usando sistemas automatizados como catálogos de acceso público y reglas de catalogación internacionalmente aceptadas para con ello lograr integrar, facilitar y divulgar información.

Un archivo de imágenes en movimiento, es un sistema de información que se caracteriza por estar estructurado mediante una serie de áreas de trabajo esenciales para su administración, gestión y desarrollo. Entre ellas, la de Análisis de documentos (catalogación, descripción de contenidos e indización), que interaccionan entre sí mediante un conjunto de operaciones destinadas a identificar y describir los documentos de imágenes en movimiento.

Ese conjunto de operaciones permiten representar la forma y el contenido de estos documentos y por lo tanto, desentrañar la información en ellos contenida, lo que facilita al usuario la identificación, recuperación y difusión de los mismos. Y es precisamente, en esa dinámica que genera el archivo, donde se resalta la importancia de la construcción de metadatos, como fuentes electrónicas disponibles en red, con los que se pueden realizar consultas a los diferentes sistemas de base datos existentes en el mundo de la informática sin tener que reprocesar nuevamente el material.

Sin embargo, antes de llegar a ese objetivo, es imprescindible definir el concepto. Para comprender el mismo es preciso conocer que fue definido como “datos sobre datos”, por su creador, Jacques Myers, en

el decenio de los años sesenta para describir conjuntos de datos. Es este significado el que más se utiliza en la actualidad.

Otros autores como Cristian Velásquez Paulus plantean que los metadatos son en general objetos que describen o proporcionan información sobre otros datos. (Paulus Velásquez, 2005)

En los materiales audiovisuales el término de metadatos cobra otro significado: “información que describe datos que incluyen el contenido, la forma y las características técnicas y editoriales de la información electrónica, los cuales son generados, consultados, manipulados y distribuidos en red”. (Sánchez Calas, 2002)

El siguiente trabajo tomará como punto de partida el último concepto citado de metadatos, es el que más se acerca al tema de la investigación en curso. También, puede resaltarse que el metadato es una herramienta para describir, identificar, definir, organizar y localizar distintos tipos de recursos de información. Razón por la cual, conocer las características de los sistemas de metadatos es imprescindible para los analistas de información bibliográfica y documental.

Los metadatos pueden ser generados manualmente, automáticamente o semi-automáticamente. La mayor parte de la creación actual está basada en la anotación humana, generalmente realizada de forma manual, lo cual es difícil y sumamente costoso, no logra extraer toda la información importante y está sujeto a ambigüedades y errores. Esta forma de elaboración exige que se realice por personas con conocimientos del recurso audiovisual que se analiza. En ella se extraen los metadatos clásicos para la búsqueda referencial de los videos como título, autor, categorías, duración, formato, información descriptiva (como personas u objetos que aparecen), situaciones, entre otros.

En la producción automática se adquieren las informaciones que se necesitan sin utilizar herramientas externas, aunque es difícil que se logre extraer todos los metadatos, por lo que se considera la producción semiautomática más exacta. En esta última un anotador humano realiza esta operación de forma totalmente manual, o auxiliándose en los metadatos extraídos de forma automática. (Amengual Galdón, 2006)

Los metadatos tienen tres funciones básicas:

- Proporcionar una descripción de un objeto o entidad de información junto con otra información necesaria para su manejo y preservación.

- Suministrar los puntos de acceso a esa descripción por medio de los cuales se generará un índice.
- Codificar la descripción para facilitar su manejo por medios automatizados.

Los metadatos presuponen procedimientos, normas y especificaciones transparentes y consecuentes; los metadatos y la formulación de estándares dependen del área o disciplina en la que son utilizados (De Jong, 2003). Los metadatos deben formularse únicamente tras un cuidadoso análisis de las necesidades de los usuarios.

2.2. Tipos de metadatos.

En el Entorno Audiovisual se encuentran los siguientes tipos de metadatos (Orozco García, 2008):

- **Metadatos específicos de los medios** (frecuencia de muestreo, en audio; textura, en imágenes fijas; movimiento en video o el tipo de carácter en los textos escritos).
- **Metadatos específicos del proceso** o gestión de medios como los de “composición”, que se generan durante la creación de materiales (listas de edición, códigos de tiempo).
- **Metadatos específicos de contenido** que sirven para aclarar el significado de un objeto media en un contexto determinado y se genera de forma manual e intelectual.
- **Metadatos clásicos** como los metadatos descriptivos objetivos (autor, título, duración del programa, datos de producción y de catalogación), los metadatos tópicos (descripción del contenido, el tema y el significado), los metadatos adicionales generados a partir de apreciaciones subjetivas del contenido, las palabras claves o descriptores¹⁸.

Los anteriores tipos de metadatos, usados en el entorno de los archivos de televisión, según (De Jong, 2003), son generados dentro de los estándares o esquemas de representación comunes, básicos para el control y la explotación de los materiales digitales y los metadatos.

2.3. Estándares de metadatos.

En la actualidad existen varios estándares de metadatos. Para la descripción de los archivos audiovisuales se utilizan básicamente dos estándares, el MPEG-7 y el Dublin Core. En la presente

¹⁸ Término normalizado que define el contenido de un documento y se utiliza como elemento de entrada para la búsqueda sistemática de información.

investigación se hace una comparación sobre estos descriptores y como ambos realizan la descripción de los materiales.

2.3.1. Estándar MPEG-7.

Consiste en una representación estándar de la información audiovisual que permite la descripción de contenidos. Con MPEG-7 ¹⁹ se busca la forma de enlazar los elementos del contenido audiovisual, encontrar y seleccionar la información que el usuario necesita e identificar y proteger los derechos del contenido.

Este estándar ofrece un mecanismo para describir información audiovisual, de manera que sea posible desarrollar sistemas capaces de indexar grandes bases de material multimedia. Las descripciones del material multimedia pueden ser de dos tipos: datos sobre el contenido e información existente en el contenido.

Datos sobre el contenido, como puede ser el autor, el género, el título o el formato; o la información existente en el contenido, la cual permite describir el elemento a través de significado semántico (descripción de alto nivel), relacionado con la interpretación del contenido o significado estructural (descripción de bajo nivel) el cual permite la extracción automática de color, forma, texturas. Los descriptores de alto nivel se caracterizan por ser eficientes y directos pero poco flexibles mientras que los de bajo nivel son genéricos, flexibles y permiten búsquedas “inteligentes”.

Los objetivos del estándar son realizar un método rápido y eficiente de búsqueda, filtrando e identificando contenido. Tiene la posibilidad de informar cómo los objetos están combinados dentro de una escena. Posee independencia entre la descripción y el soporte donde se encuentra la información.

Este estándar se le puede aplicar a una serie de elementos como audio, voz, video, imágenes, gráficos y modelos 3D. La arquitectura MPEG-7 se basa en que la descripción debe estar separada del contenido audiovisual. Por otro lado, tiene que haber una relación entre contenido y descripción.

¹⁹ <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>

Herramientas de MPEG-7 (Herranz Arribas, 2003)

MPEG-7 proporciona una serie de herramientas para describir contenido multimedia, estas herramientas de descripción (metadatos, descriptores, esquemas de descriptores), sirven para crear descripciones que serán la base para aplicaciones que permitan el acceso a contenido de este tipo.

- **Descriptores (D):** un descriptor es una representación de una característica definida sintáctica y semánticamente.
- **Descripción de sistemas multimedia (Multimedia Description Schemes) (DS):** especifica la estructura y semántica de las relaciones entre sus componentes, que pueden ser descriptores (D) o esquemas de descripción (DS).
- **Lenguaje de descripción y definición (Description Definition Language) (DDL):** es un lenguaje basado en XML que se utiliza para definir las relaciones estructurales entre los descriptores y permite la creación o modificación de esquemas de descripción (DS) y la creación de nuevos descriptores (D).
- **Herramientas del sistema:** son herramientas que hacen referencia a la binarización, sincronización y almacenamiento de descriptores. También se encargan de la protección de la propiedad intelectual.

Las partes más importantes del estándar son las siguientes: (Herranz Arribas, 2003)

- **Sistema MPEG-7 (MPEG-7 System):** 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.
- **Definición de lenguaje MPEG-7 (MPEG-7 Description Definition Language):** lenguaje para definir nuevos esquemas de descripciones.
- **Audio MPEG-7 (MPEG-7 Audio):** esta parte proporciona estructuras para describir material sonoro.
- **Visual MPEG-7 (MPEG-7 Visual):** estructuras básicas y descriptores que definen características de color, textura, forma y movimiento de material audiovisual.
- **Descripción de sistemas multimedia MPEG-7 (MPEG-7 Multimedia Description Schemes):** describen información genérica.

Descriptores MPEG-7

El estándar MPEG-7 define una serie de descriptores que permiten analizar y caracterizar el contenido de fuentes audiovisuales para su posterior indexación, búsqueda o comparación (Martínez, 2004). Los descriptores cubren distintas características visuales básicas y elementales como son: el color, textura, forma, movimiento y localización, entre otras. La descripción es generada automáticamente mediante procesamiento digital.

Descriptores de color (Delcor Ballesteros, et al., 2006.)

Representan la cualidad más básica del contenido visual. Se definen cinco descriptores para describir el color. Los cinco primeros representan la distribución del color. Los dos restantes, describen la distribución espacial del color y la relación del este entre una secuencia o conjunto de imágenes.

Estructura del descriptor de color (*Color Structure Descriptor*): caracteriza la distribución de los colores en una imagen. Construye una especie de histograma de color, en el cual tendrán mayor importancia los colores que más se reparten por la imagen. El descriptor divide la imagen en bloques de 8x8 píxeles y analiza dentro de estos bloques los distintos colores que aparecen, incrementándolos así en el histograma. A diferencia de un histograma de color, permite distinguir entre dos imágenes que tengan la misma cantidad de píxeles de un color pero con distinta distribución de estos píxeles. Este descriptor es útil para comparaciones imagen-imagen y añade funcionalidades distintas a las del histograma de color que permiten mejorar la búsqueda de similitud en determinados tipos de imágenes, como por ejemplo las imágenes de naturaleza.

Color Escalable (*Scalable Color*): consiste en un histograma²⁰ de color, codificado. Su representación se puede escalar de manera que se adecue lo máximo al tamaño de datos con el que se quiere trabajar. Este descriptor es útil en comparaciones imagen-imagen o en búsquedas basadas en características de color. La fiabilidad de la búsqueda aumenta proporcionalmente al número de colores distintos que se tengan en cuenta en el histograma.

Color Dominante (*Dominant Color*): es el más conveniente para ser utilizado en imágenes o zonas de ellas, en las cuales un pequeño número de colores es suficiente para caracterizar la información

²⁰ Representa la cantidad de píxeles de la imagen para cada posible valor de color.

cromática²¹ de la región determinada. Sería aplicable por ejemplo en imágenes de banderas o marcas determinadas. En este caso la cuantificación se usa para extraer un reducido número de colores que sean suficientes como para caracterizar la imagen o región. También se calcula una coherencia espacial entre estos colores y dónde están situados lo cual se utilizará en algoritmos de similitud.

Espacio de Color (*Color Space*): Consiste en un tipo de datos que especifica el espacio de color en el cual se expresan o trabajan los otros descriptores de color.

Cuantización de Color (*Color Quantization*): define una cuantificación²² uniforme de un espacio de color determinado. El número de valores que el cuantificador produce es configurable de manera que posee una flexibilidad elevada que le da una amplia gama de usos. Dentro de MPEG-7 este descriptor se combina con descriptores del color dominante, para hacer comparables por ejemplo dos resultados de un determinado descriptor.

Diseño de Color (*Color Layout*): permite representar la distribución espacial del color dentro de las imágenes de una manera muy compacta, con lo cual representa una herramienta de gran utilidad a la hora de buscar imágenes a partir de modelos determinados, y lo hace con gran eficiencia y velocidad. Su fácil cálculo permite también usarlo en la comparación de secuencias de imágenes, en las cuales se precisa un análisis de similitud entre cada una de sus componentes. Las grandes ventajas de este descriptor son:

- No depende ni del formato, ni de la resolución, ni del margen dinámico de las imágenes o videos en que se use. Por este motivo, puede usarse para comparar imágenes o videos con distintas resoluciones o para comparar imágenes enteras con partes de imágenes.
- El software-hardware que requiere es relativamente mínimo (usa solamente 8 bytes por imagen cuando trabaja por defecto). Esto lo convierte en un descriptor adecuado para ser utilizado en dispositivos móviles en los que los posibles recursos se ven limitados por la capacidad del hardware.
- Permite trabajar con distintas precisiones de descripción de manera que se agudizan las comparaciones cuando sea necesario.

²¹ Este término se refiere al matiz relativo de un color.

²² Expresa numéricamente una magnitud.

Grupo de marco o Grupo de imágenes (*Group of Frames - Group of Pictures*): es una extensión del *Scalable Color*, que a diferencia de éste, que está definido para imágenes inmóviles, se aplica a secuencias de video o secuencias de imágenes fijas. Brinda la posibilidad de calcular de tres formas distintas el histograma de color:

- Histograma promedio: toma de cada imagen de la secuencia el promedio de los valores del histograma.
- Histograma de mediana: toma de cada imagen de la secuencia el valor central del conjunto de valores del histograma. Es más fiable ante errores o picos de intensidad de la imagen.
- Histograma de intersección: toma de cada imagen de la secuencia el mínimo del conjunto de valores del histograma, para así ver cuál es el color “menos común” en el conjunto de imágenes.

Descriptores de textura (Delcor Ballesteros, et al., 2006.)

La textura es una cualidad muy importante para describir una imagen. Se diseña para caracterizar las texturas o regiones. Observan la homogeneidad de las regiones y los histogramas de los bordes de dichas regiones.

Textura Homogénea (*Homogeneous Textura*): es una importante herramienta a la hora de buscar y escoger dentro de grandes colecciones de imágenes de gran similitud visual. Este descriptor utiliza un banco de 30 filtros que permite obtener una afinada descripción de las distintas texturas de la imagen para poder comparar de esta manera con las de otras. Es una herramienta muy útil por ejemplo para distinguir determinadas zonas en imágenes aéreas, por ejemplo, cultivos.

Textura de navegación (*Texture Browsing*): especifica la caracterización perceptiva de una textura, la cual es similar a la caracterización de ella que hace un ojo humano, en cuanto a términos de regularidad, tosquedad y direccionalidad. Es útil para búsquedas y clasificaciones a “grosso modo” de texturas. Su implementación es parecida a la del anterior.

Borde de histograma (*Edge Histogram*): facilita información sobre el tipo de contornos o bordes que aparecen en la imagen. Trabaja dividiendo la imagen en 16 sub-imágenes y es capaz de analizar en ellas el tipo de bordes existentes con el uso de distintos filtros que le permiten diferenciar si son bordes horizontales, verticales, oblicuos o aleatorios. Su utilización principal es la comparación imagen-imagen,

especialmente en imágenes de naturaleza con una gran no uniformidad de contornos. Su uso es muy útil también en combinación con el de otros descriptores como por ejemplo el histograma de color.

Descriptores de forma (Delcor Ballesteros, et al., 2006.)

La forma posee una información semántica muy importante, debido a que los humanos son capaces de reconocer los objetos sólo viendo su forma. Esta información sólo puede ser extraída mediante una segmentación similar a la que realiza el sistema visual humano. Estos descriptores describen las regiones, contornos y formas para imágenes 2D y volúmenes 3D.

Forma de la región (*Region Shape*): Este descriptor permite clasificarlas según esta característica, de esta manera se puede comparar las formas de distintas imágenes y ver por ejemplo si se trata del mismo objeto u objetos similares. Las grandes ventajas de este descriptor son su reducido tamaño y su velocidad, hay que tener en cuenta que el tamaño de los datos necesarios para su representación está fijado en 17,5 bytes.

Forma del contorno (*Contour Shape*): A diferencia del anterior, este descriptor en lugar de analizar el conjunto de regiones que dan lugar a una forma, relaciona esta última con su contorno. Se caracteriza por representar muy bien las características de contorno con lo que facilita posteriores búsquedas y recuperaciones, es robusto ante movimientos, ante oclusiones en las formas y ante distintas perspectivas, y es sumamente compacto.

Forma en 3D (*Shape 3D*): La forma 3D permite describir con detalle la forma de mallas en 3D. Herramienta que hoy en día debido al continuo desarrollo de las tecnologías multimedia es de gran utilidad.

Descriptores de movimiento (Delcor Ballesteros, et al., 2006.)

El movimiento se define mediante cuatro descriptores que lo describen en las secuencias de video. Este va asociado a los objetos de la secuencia o al movimiento propio de la cámara.

Movimiento de cámara (*Camera Motion*): Es un descriptor que da información sobre los movimientos que efectúa la hipotética cámara que toma la secuencia de imágenes.

Trayectoria del movimiento (*Motion Trajectory*): Este descriptor permite analizar la trayectoria de un objeto en una secuencia de imágenes, la cual se consigue con la localización en tiempo y espacio de un punto representativo del objeto determinado.

Movimiento de parámetros (*Parametric Motion*): Consiste en describir el movimiento de ciertos objetos en una cadena de imágenes. Estos objetos se definen como regiones en la imagen, y su movimiento se registra de una manera compacta como un conjunto de parámetros. Este descriptor permite diferenciar numerosos tipos de movimiento elementales como translaciones, rotaciones, *zooms*²³, de manera que cualquier otro movimiento se puede especificar como una combinación de estos.

Actividad del movimiento (*Motion Activity*): Se puede decir que se trata de un descriptor que intenta evaluar la “intensidad de la acción” en una secuencia de imágenes, de manera parecida a como lo perciben los humanos.

Descriptores de localización (Delcor Ballesteros, et al., 2006.)

La localización de los elementos dentro de una imagen se utiliza para describir elementos en el dominio espacial o temporal dentro de la secuencia de video.

Locutor de región (*Region Locutor*): Este descriptor permite la localización de determinadas regiones en una imagen.

Locutor de espacio temporal (*Spatio Temporal Locutor*): Sería como el anterior pero aplicado a secuencias de imágenes de manera que localiza determinadas regiones analizando tiempo y espacio.

Descriptor de reconocimiento facial (Delcor Ballesteros, et al., 2006.)

Como su nombre indica, este descriptor permite comparar caras para analizar su parecido o buscar caras con alto parecido a una en concreto. Esto se consigue relacionando distintas posiciones de la cara con las intensidades de la imagen en esa posición, de manera que se crean datos de referencia que luego pueden compararse con otros para analizar la similitud.

²³ Tiene como objetivo la distancia focal variable, mediante él se puede variar a voluntad la distancia focal y en consecuencia el ángulo de visión.

2.3.2. Estándar Dublin Core.

Dublin Core ²⁴ es un modelo de metadatos elaborado y auspiciado por la DCMI (*Dublin Core Metadata Initiative*), una organización dedicada a fomentar la adopción extensa de los estándares interoperables de los metadatos y a promover el desarrollo de los vocabularios especializados de metadatos para permitir sistemas más inteligentes en el descubrimiento de recursos.

Las implementaciones de Dublin Core utilizan generalmente XML y se basan en el *framework* (*Resource Description Framework*). Es un sistema de 15 definiciones semánticas descriptivas que pretenden transmitir un significado semántico a las mismas.

Este sistema de definiciones fue diseñado específicamente para proporcionar un vocabulario de características "base", capaces de proporcionar la información descriptiva básica sobre cualquier recurso, sin que importe el formato de origen, el área de especialización o el origen cultural.

Clasificación y Elementos

En general, se puede clasificar estos elementos en tres grupos que indican la clase o el ámbito de la información que se guarda en ellos:

- Elementos relacionados principalmente con el contenido del recurso.
- Elementos relacionados principalmente con el recurso cuando es visto como una propiedad intelectual.
- Elementos relacionados principalmente con la aplicación del recurso.

Dentro de cada clasificación se encuentran los siguientes elementos (Morales Sánchez, 2005):

En Contenido:

1. Título (*Title*). Se refiere al título que lleva por nombre el documento.
2. Materia (*Subject*). En este campo se hace referencia a los diversos temas que puede contener el material.

²⁴ <http://www.dublincore.org/>

3. Descripción (*Description*). En este campo se hace un breve resumen sobre el contenido del objeto digital.
4. Fuente (*Source*). Es como una pequeña ficha bibliográfica que se elabora para asentar los datos sobre la procedencia del documento original.
5. Relación (*Relation*). Este campo tiene que ver con el material principal u objetos de su misma referencia, ya sea una colección, una serie o un documento.
6. Cobertura (*Coverage*). Este campo se refiere al proyecto o sitio donde estará resguardada la información. Aquí pueden anotarse fechas, zonas geográficas.

Dentro del apartado de Propiedad Intelectual:

7. Creador (*Creator*). Aquí se anota el autor intelectual de la obra o documento original.
8. Editor (*Publisher*). Este campo se refiere al sitio o colección responsable, a la que está adscrito el material.
9. Colaborador (*Contributor*). En este campo se anotan, si es que se da el caso, el nombre u organización que contribuyó a la creación del material, que no se especificó en la parte de Creador.
10. Derechos (*Rights*). Se anota en este campo el nombre o la institución a la cual pertenece el material y lo facilitó.

En el apartado de Aplicación:

11. Fecha (*Date*). Se anota la fecha de elaboración del registro.
12. Formato (*Format*). En este campo se registra el tipo de extensión con que se presenta el objeto digital, ya sea HTML, JPG, GIFF o PDF.
13. Identificador (*Identifier*). Se refiere a la dirección electrónica de origen a la que está el material. Para ello se utilizan las siglas URL.
14. Tipo (*Type*). Aquí se menciona la presentación que tiene el objeto digital, ya sea como texto, audio, video.
15. Lenguaje (*Language*). En este campo se establecen las siglas correspondientes al idioma en que se presenta la publicación.

Capítulo II: Metadatos en la búsqueda de videos digitales

El Dublin Core es un metadato simple, flexible, que ayuda a la hora de realizar una búsqueda ya pueda ser por contenido, por propiedad intelectual o por instanciación. Este estándar permite una interoperabilidad y un alto nivel de normalización.

Para describir los 15 elementos que el estándar Dublin Core define, se crearon 10 atributos, de los cuales se le dieron valor fijo a 6 de ellos, los cuales se observan en la **Tabla 1**.

Atributos	Valor Fijo
Versión	1.1
Autoridad de registro	Dublin Core Metadata Initiative
Idioma	En (Inglés)
Obligación	Opcional
Tipo de datos	Cadena de caracteres
Ocurrencia máxima	Ilimitada

Tabla 1 Atributos con valor fijo del estándar Dublin Core. (Olivé Peig, 2003)

Los restantes 4 atributos describen todos los elementos definidos por el estándar de la siguiente manera:

Identificador	Title
Etiqueta	Title
Definición	Nombre que recibe el recurso
Comentario	Típicamente, <i>Title</i> será el nombre con el que el recurso es formalmente conocido.

Tabla 2 Atributo título.

Identificador	Creator
Etiqueta	Creator
Definición	Una entidad responsable de la creación del recurso.
Comentario	Ejemplos de <i>Creator</i> serían una persona, una organización o un servicio. Típicamente el

Capítulo II: Metadatos en la búsqueda de videos digitales

	nombre de un <i>Creator</i> se usa para indicar cuál es la entidad.
--	---

Tabla 3 Atributo creador.

Identificador	Subjet
Etiqueta	Subjet and Keywords
Definición	El tema del contenido del recurso
Comentario	Típicamente un <i>Subjet</i> se expresará con palabras claves, frases o códigos de clasificación que describen el tema del recurso. Se recomienda para un buen uso de este elemento que se seleccione un valor de un vocabulario controlado o de algún esquema formal de clasificación.

Tabla 4 Atributo materia.

Identificador	Description
Etiqueta	Description
Definición	Un relato sobre el contenido del recurso
Comentario	Puede incluir pero no está limitado a: un resumen, una tabla de contenidos, una referencia a una representación gráfica del contenido o un relato del texto libre.

Tabla 5 Atributo descripción.

Identificador	Publisher
Etiqueta	Publisher
Definición	Una entidad responsable de hacer disponible el recurso.
Comentario	Ejemplos de <i>Publisher</i> incluyen una persona, una organización o un servicio. Típicamente, el nombre de Publisher tendría que ser usado para indicar la entidad.

Tabla 6 Atributo editor.

Identificador	Contributor
Etiqueta	Contributor

Capítulo II: Metadatos en la búsqueda de videos digitales

Definición	Una entidad responsable de hacer contribuciones al contenido del recurso.
Comentario	Ejemplos de <i>Contributor</i> incluyen una persona, una organización o un servicio. Típicamente el nombre de un <i>Contributor</i> tendría que ser usado para indicar la entidad.

Tabla 7 Atributo colaborador.

Identificador	Date
Etiqueta	Date
Definición	Una fecha asociada a un suceso en el ciclo de vida del recurso
Comentario	<i>Date</i> estará asociado a la creación o a la puesta en disponibilidad del recurso. Se recomienda para un buen uso de este elemento seguir la codificación definida en uno de los perfiles de ISO 8601 (ISO, 2004).

Tabla 8 Atributo fecha.

Identificador	Type
Etiqueta	Resource Type
Definición	La naturaleza o género del contenido del recurso.
Comentario	<i>Type</i> incluye términos que describen categorías generales, funciones, géneros o niveles de agregación. Se recomienda para un buen uso seleccionar un valor dentro de un vocabulario controlado (por ejemplo, la lista provisional que la DCMI ha confeccionado). Para describir la manifestación física o digital del recurso, se recomienda usar el elemento <i>Format</i> .

Tabla 9 Atributo tipo.

Identificador	Format
Etiqueta	Format
Definición	La manifestación física o digital del recurso.
Comentario	<i>Format</i> incluye el tipo de medio en el que se encuentra el recurso o sus dimensiones. Este

Capítulo II: Metadatos en la búsqueda de videos digitales

	<p>elemento puede ser usado para determinar el software, hardware o equipo necesario para mostrar u operar sobre el recurso. Ejemplos de dimensiones son la medida o la duración. Se recomienda para un buen uso seleccionar un valor dentro de un valor dentro de un vocabulario controlado, por ejemplo, la lista de tipos de medios de Internet.</p>
--	---

Tabla 10 Atributo formato.

Identificador	Identifier
Etiqueta	Resource Identifier
Definición	Una referencia no ambigua del recurso dentro de un contexto concreto.
Comentario	Se recomienda para un buen uso de este elemento identificar el recurso mediante una cadena de caracteres o un número que siga un sistema de identificación formal.

Tabla 11 Atributo identificador.

Identificador	Source
Etiqueta	Source
Definición	Una referencia al recurso del cual el presente recurso se deriva.
Comentario	El presente recurso puede estar derivado del <i>Source</i> . Se recomienda para un buen uso de este elemento referenciar el recurso <i>Source</i> mediante una cadena de caracteres o un número que siga un sistema de identificación formal, como el elemento <i>Identifier</i> .

Tabla 12 Atributo fuente.

Identificador	Language
Etiqueta	Language
Definición	Idioma en que está escrito el contenido del recurso.
Comentario	Se recomienda para un buen uso de este elemento usar los valores definidos en la RFC 1766, la cual incluye un código de lenguaje de dos letras (tomado del estándar ISO 639) (ISO, 2002), seguido opcionalmente por un código del país también de dos letras (tomado del estándar ISO 3166) (ISO, 1999).

Tabla 13 Atributo lenguaje.

Identificador	Relation
Etiqueta	Relation
Definición	Una referencia a un recurso relacionado.
Comentario	Se recomienda para un buen uso de este elemento referenciar el recurso mediante una cadena de caracteres o un número que siga un sistema de identificación formal.

Tabla 14 Atributo relación.

Identificador	Coverage
Etiqueta	Coverage
Definición	El alcance o ámbito donde se circunscribe el contenido del recurso.
Comentario	<i>Coverage</i> incluye típicamente una localización espacial (un nombre de un lugar o unas coordenadas geográficas), un período temporal (el nombre de una época, una fecha o un rango de fechas) o jurisdicción (como el nombre de una entidad administrativa). Se recomienda utilizar un vocabulario controlado.

Tabla 15 Atributo cobertura.

Identificador	Rights
Etiqueta	Rights Management
Definición	Información sobre los derechos que se pueden ejercer sobre el recurso.
Comentario	Este elemento contendrá una sentencia que describe la gestión de un derecho a ejercer sobre el recurso o una referencia a un servicio que proporcione una información a fin. Información sobre derechos incluye normalmente derechos de propiedad intelectual, <i>copyright</i> y parecidos. Si el atributo <i>Rights</i> no está presente, no se puede hacer ninguna aseveración sobre el estado del recurso con relación a los derechos que se pueden ejercer o no sobre él.

Tabla 16 Atributo derecho.

La definición original de los elementos de Dublin Core se hizo en inglés, pero se han hecho diferentes traducciones a otros idiomas con más o menos resultados, este esquema de metadatos pretende ser genérico pero adaptable a diferentes dominios concretos y ampliables para cubrir requerimientos específicos.

Sin embargo, a diez años de su creación y después de que ha sido aprobado como norma ISO (ISO 15836:2003), no parece que su adopción esté generalizada. Un estudio reciente sobre el uso del Dublin Core en 100 proveedores de datos registrados en la *Open Archives Initiative* muestra que sólo 82 de los proveedores tienen metadatos disponibles para su análisis, que sólo cinco de los quince elementos del Dublin Core —“*Creator*”, “*Identifier*”, “*Title*”, “*Date*” y “*Type*”— están presentes en el 71% de los casos, y que 44 de los 82 proveedores solo usan los elementos “*Creator*” y “*Identifier*” en la mitad de sus productos. A todo ello se podría añadir que el uso de los cinco elementos anteriores son tan obvios que difícilmente pueden atribuirse a la existencia del estándar Dublin Core. (Estivill, et al., 2005)

Teniendo en cuenta las características de los estándares de metadatos que se utilizan para describir archivos audiovisuales, se propone la utilización del MPEG-7. A pesar que ambos estándares permiten la descripción de los videos, el propuesto es capaz de realizar descripciones relacionadas con la interpretación de los contenidos y proporciona información descriptiva básica de los mismos mientras que el Dublin Core solo cumple con la última característica mencionada.

2.4. Herramientas internacionalmente utilizadas para la elaboración y procesamiento de metadatos de materiales audiovisuales.

Se han diseñado herramientas para la elaboración y procesamiento de los metadatos que realizan la creación automática o semi-automática de los mismos. La creación automática aunque refleje mayor facilidad no resulta la más favorable pues no tiene resultados completos, puede contener ambigüedades; se hace necesario la utilización de herramientas que realicen esta operación con la ayuda de un anotador humano que garantice la correcta extracción de los metadatos.

Actualmente el desarrollo de sistemas de este tipo es muy escaso porque la gestión automática de los metadatos es sumamente compleja. Existen herramientas de este tipo pero son particulares de las empresas las cuales desarrollan su propio sistema de gestión de metadatos de acuerdo a sus

necesidades. Por lo que esta investigación no propone ninguna de las existentes, estas no se adecuan a las necesidades del polo además de tener licencias privativas.

Para la creación de los metadatos del estándar MPEG- 7 se puede utilizar cualquier herramienta de texto o que permita la edición de documentos XML (Microsoft Word, Bloc de Notas, XMLSPY). La cantidad de herramientas existentes para la elaboración y procesamiento de metadatos actualmente no es suficiente para realizar comparaciones; debido a la poca información disponible sobre este tema, y a lo difícil que se hace realizar estas operaciones.

2.5. Utilización y procesamiento de metadatos en el polo de Video y Sonido Digital.

Un sistema de búsqueda eficiente y preciso debe hacer un uso adecuado de los metadatos que describen los archivos que el sistema debe procesar. Sin utilizar un estándar específico y sin tener definido qué tipo de metadatos presenta, un buscador solo puede realizar búsquedas simples atendiendo a las propiedades básicas de los ficheros, como el nombre, la extensión y el tamaño. Estas búsquedas están caracterizadas por responder de forma ineficiente e imprecisa.

En el polo de Video y Sonido Digital, el volumen de información audiovisual está en un diario incremento, por lo que es necesario contar con un sistema que responda de forma precisa a cualquier búsqueda realizada sobre toda esta información. Establecer un estándar para la descripción de los metadatos y un sistema de búsqueda que haga utilización del mismo es el aspecto en el que se propone trabajar, buscando complementar, enriquecer y completar una propuesta de modelo arquitectónico para un sistema de búsqueda de videos digitales a través de sus metadatos.

La entrevista realizada a los líderes de los proyectos del polo de Video y Sonido Digital de la Facultad 9 (Ver Anexo 1), se aplicó con el objetivo de medir el grado de conocimiento sobre los estándares de metadatos y su utilización en el desarrollo de sistemas de búsquedas de contenido en materiales audiovisuales, arrojó como resultado que en el polo no existe un estándar de metadatos definido para la descripción de los videos, además la inexistencia de un sistema de búsqueda que se capaz de recuperar la información a través de metadatos de bajo nivel.

2.6. Resumen del capítulo.

En el presente capítulo se realizó un estudio profundo de los estándares de metadatos más utilizados en el mundo. De acuerdo a la investigación y dada la ausencia de un estándar para la descripción de metadatos en el Polo de Video y Sonido Digital, además de la necesidad del polo de desarrollar un sistema de búsqueda por metadatos se propone utilizar el estándar MPEG-7.

La utilización de este estándar de metadatos proporcionará al polo una descripción profunda de todos los archivos multimedias con los que cuenta, que esto va a facilitar la realización de búsquedas a través de descriptores de contenidos audiovisuales, que pueden ser de alto nivel (autor, formato, descripción) y de bajo nivel (textura, color, forma). Esto traerá consigo que la búsqueda de cualquier material se realice de una forma rápida y eficiente. Algunos de los buscadores más importante del mundo han implementado la utilización de metadatos de este tipo como por ejemplo (Videoma, Mise, Google).

Capítulo III: Propuesta del modelo conceptual del sistema de búsqueda.

Contar con un buscador de contenidos audiovisuales es un reto que presenta el Polo de Video y Sonido Digital. En este capítulo se realizará la descripción del modelo arquitectónico propuesto para el sistema de búsqueda de videos digitales a través sus metadatos. Se abordan temas relacionados con los lenguajes, herramientas, metodologías y tecnologías que serán propuestas para ser utilizadas para la realización del sistema.

3.1. Propuesta del modelo conceptual del sistema de búsqueda.

Un sistema eficiente de búsqueda de información audiovisual debe ser capaz de definir elementos de recuperación que sean significativos en el contexto de la aplicación. Además de proporcionar un método de consulta que permita al usuario especificar de forma natural características selectivas.

Los sistemas de consultas en bases de datos de videos se basan, tradicionalmente, en que realizan un proceso encaminado a obtener un conjunto de descriptores y utilizar estos para la búsqueda en la base de datos. De esta forma se elimina el coste de realizar el análisis de contenidos de los elementos que componen la base de datos en cada consulta, pero limita las búsquedas a aquellas que se pueden realizar con la combinación de los descriptores utilizados. De manera que este tipo de análisis es efectivo para conjuntos de archivos en los que los descriptores satisfacen cualquier consulta.

En los últimos años, varios sistemas prototipos se han propuesto abordar diferentes aspectos de la información contenida en los videos, como son las texturas, similitud de formas y relaciones semánticas²⁵ entre objetos. El objetivo de las técnicas de consulta basadas en contenido es encontrar de forma eficiente los materiales audiovisuales en una base de datos que son similares a la búsqueda indicada.

El sistema que se propone debe ser capaz de realizar búsqueda a través de las dos técnicas de búsquedas existentes: las tradicionales (a través de texto) y las búsquedas a través de ejemplos, basada en descriptores que permitan una aproximación mayor al contenido de los archivos multimedia. Pueden ser derivados del análisis estructural de las imágenes, basándose en la utilización de elementos como el color, la forma y la textura que ya se obtienen en la etapa de análisis de la imagen.

²⁵ Se refiere a los aspectos del significado, sentido o interpretación de un determinado elemento, símbolo, palabra, lenguaje o representación formal.

Capítulo III: Propuesta del modelo conceptual del sistema de búsqueda

La recuperación por contenido cambia la manera de especificar las búsquedas a un paradigma²⁶ basado en lo que se conoce como búsqueda por ejemplos. Así, la consulta típica no se hace solo mediante una cadena de términos sino que también, introduciendo una imagen o dibujando un esbozo, a partir de lo cual el sistema buscará otras con características similares de color, forma o textura.

Analizando las características de los sistemas de búsquedas descritos anteriormente, basados en el nuevo paradigma de búsquedas por contenido, se propone un modelo arquitectónico donde se representan los principales módulos que integran un sistema de recuperación de información.

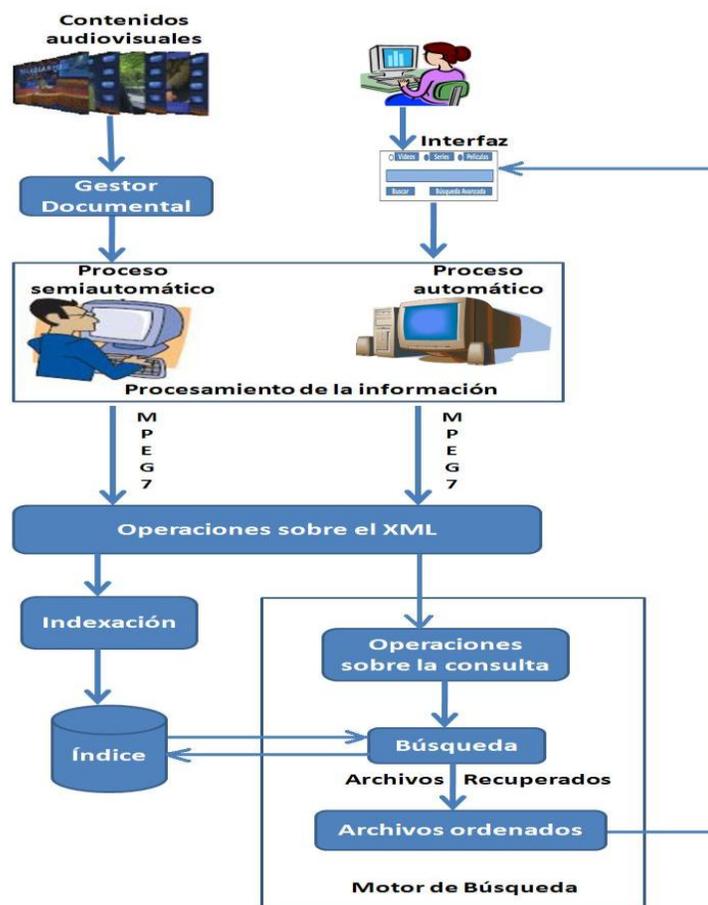


Figura 5 Modelo conceptual del sistema de búsqueda propuesto

²⁶ Conjunto de teorías acerca de una faceta de la realidad, que ofrece la posibilidad de resolver problemas a partir de sus principios fundamentales.

En el modelo existen dos procesos fundamentales el proceso de indexación y el de búsqueda el primero comienza cuando el gestor documental decide que material audiovisual se va a procesar, posteriormente pasa al procesamiento de la información que es donde se extraen los metadatos generados en un XML, de allí se pasa a realizarle las operaciones quedando definido los términos de indexación que se guardaran en una base de datos.

El segundo proceso comienza cuando un usuario especifica la búsqueda en una interfaz, de allí pasa al procesamiento de la información porque el usuario puede haber pasado una imagen como parámetro de búsqueda a lo que hay que extraerle todos los metadatos para poderlo comparar con los índices que están almacenados en la base de datos. Posteriormente pasa al motor de búsqueda que este analiza todos los índices existentes en la base datos ordenándolos de acuerdo a su importancia y mostrándolo en la interfaz finalmente. A continuación se explican todos los componentes del modelo conceptual.

Contenidos audiovisuales: Es el conjunto de materiales a los cuales se le realizará el proceso de indexación (videos, conferencias, series, películas, imágenes, subtítulos, sinopsis).

Gestor documental: Es el encargado de gestionar los archivos audiovisuales que serán procesados.

Procesamiento de la información: constará con los procesos automático y semiautomático.

- **Proceso automático:** Este proceso se realiza a través de un sistema el cual se encarga de realizar el análisis sobre texto (sinopsis, subtítulos), los cuales pasarían a formar términos de indexación.
- **Proceso semiautomático:** Se refiere a la detección de metadatos por parte de un usuario, de las escenas automáticamente detectadas así como el proceso de refinamiento de la detección automática. Permite la adición manual de metadatos, agregando información referencial básica como palabras claves, duración, formato, autor, título, breve descripción del video, información descriptiva como personas u objetos que aparecen.

A continuación se exponen la realización de algunos descriptores que se obtendría del proceso semiautomático.

Forma de la región (*Region Shape*)

La forma de un objeto puede estar compuesta por una o varias regiones como también de zonas vacías como agujeros. *Region Shape* utiliza todos los píxeles dentro de una imagen, que puedan describir

formas. No solo es capaz de describir objetos totalmente compactos como el (a) y (b) de la siguiente figura, sino que también tiene facultades para describir uno con agujeros o zonas disjuntas como los ejemplos (c), (d) y (e) de la figura # 7.

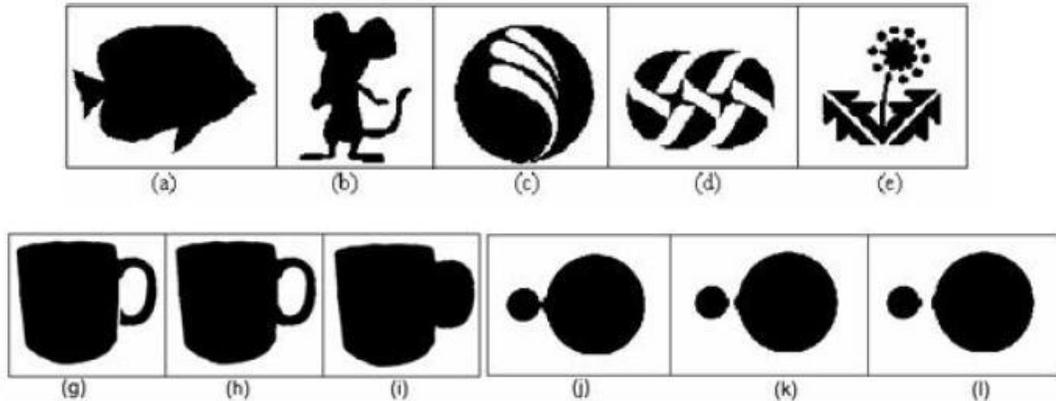


Figura 6 Ejemplo de regiones

Dicho descriptor también se caracteriza por ser robusto al ruido que se introduce inevitablemente en el proceso de segmentación, produce la menor deformación en los bordes del objeto en su reconstrucción.

Otros atributos importantes a resaltar son su pequeño tamaño y un tiempo rápido de extracción. El total de datos de representación debe ocupar 140 bits y la extracción de características es sencilla con el fin de poder tener una bajo nivel de complejidad computacional siendo conveniente para la identificación de objetos en procesado de videos.

Region Shape calcula los coeficientes ART (transformación angular radial), obteniendo una matriz de coeficientes denominados *ArtM*. Estos valores dan información sobre los objetos o formas que constituyen la imagen.

Proceso de creación:

- Definir la estructura.
- Crear las funciones comunes en todos los descriptores.
- Implementar las funciones más importantes del descriptor
- Calcular el punto central de la imagen de entrada.
- Extraer de cada posición el valor y las coordenadas cartesianas correspondientes de la imagen.

- Calcular el valor de la función base ART.
- Hacer el módulo del coeficiente.
- Cuantificar el valor obtenido
- Calcular la similitud entre dos descriptores de entrada.
- Dar respuesta.

Estructura del descriptor de color (*Color Structure Descriptor*)

El *Color Structure* es un descriptor que se caracteriza por clasificar las imágenes teniendo en cuenta la distribución espacial de los colores. Se puede decir que construye una especie de histograma de color de la imagen pero con la particularidad de dar relevancia, no al hecho de que un color se encuentre en un elevado número de píxeles, sino de que aparezca en muchas zonas de la imagen. Su funcionamiento consiste en dividir la imagen de partida en bloques de 8x8 píxeles e investigar dentro de cada uno de estos bloques los colores distintos que aparecen. Este descriptor ofrece una herramienta de comparación entre imágenes que en combinación con descriptores de análisis de contornos es muy utilizado en imágenes de paisajes.

De esta manera, este descriptor permitirá distinguir, por ejemplo, entre dos imágenes cuyo número de píxeles de un color es el mismo, pero la distribución de éste dentro de la imagen es distinta.

En la primera imagen del ejemplo, el color está centrado en una única zona, en cambio, en la segunda, aun existiendo el mismo número de píxeles de este color, su distribución es mucho más uniforme en todo el conjunto; así, al dividir la imagen en sub-imágenes, el número de estas que tendrán el color en su interior, será mucho mayor en la segunda que en la primera, y por lo tanto, también en el segundo caso, el valor del color en el histograma será también mayor.

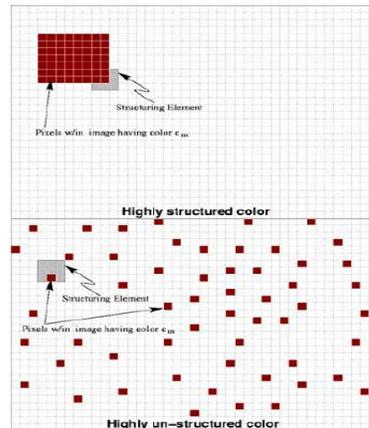


Figura 7 Ejemplo de distribución espacial

Proceso de creación:

- Definir la estructura denominada *VdColorStructure*.
- Construir las funciones comunes del descriptor.
- Definir el espacio de color.
- Cuantificar el valor no uniforme de cada color.
- Dividir la imagen en bloques de 8x8 píxeles.
- Crear la función de similitud.
- Imprimir la similitud por pantalla.

Textura Homogénea (*Homogeneous Textura*)

La textura, representa la regularidad de una imagen en lo que refiere a direccionalidad, tosquedad, regularidad del patrón. El descriptor *Homogeneous Texture*, provee a sus usuarios la capacidad de realizar comparaciones imagen a imagen, realizando una descripción de la textura de estas imágenes a través del valor de la energía y la desviación energética de la señal, extraídas de la distribución frecuencial de esta.

El espacio frecuencial a partir del cual se extraen las características de textura de la imagen, se parte en regiones de 30° en dirección angular y en 5 regiones de octava en dirección radial. De esta manera se obtienen 30 canales como muestra la figura siguiente:

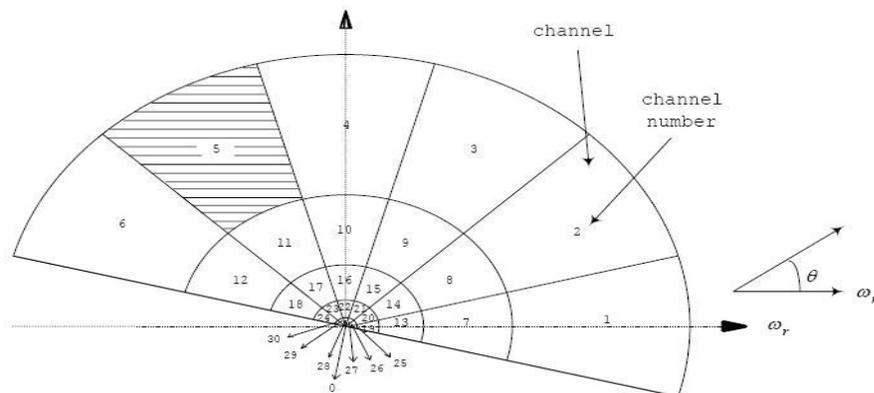


Figura 8 Espacio frecuencial

Proceso de creación:

- Definir la estructura denominada *VdHomoTexture*.
- Construir las funciones comunes del descriptor.
- Definir memoria para 5 imágenes del mismo tamaño que la de entrada.
- Liberar espacio utilizado.
- Calcular distancia entre vectores.
- Se crea una función *VdHomoTextureDistance*.
- Mostrar resultado.

Borde de histograma (*Edge Histogram*)

Edge Histogram es el descriptor que se encarga de representar la distribución espacial de cinco tipos de contornos en regiones locales de una imagen. Es capaz de reconocer cuatro contornos direccionales, es decir, con una dirección claramente marcada, y un contorno de tipo no direccional. El análisis de los bordes juega un papel muy importante en la percepción de las imágenes, es un método que permite la recuperación de imágenes con significado semántico similar.

De este modo el objetivo es encontrar imágenes semejantes a otras imágenes, ya sea a través de ejemplos o de esbozos, especialmente en el caso de imágenes naturales con distribuciones de bordes no

uniformes. En este contexto la eficacia de reconocimiento de las imágenes puede verse significativamente mejorada si dicho descriptor es combinado con otros descriptores, como los descriptores de color.

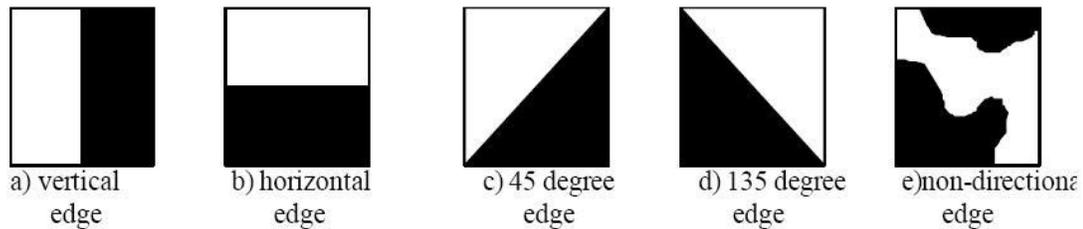


Figura 9 Ejemplo de bordes

Edge Histogram divide la imagen en 16 sub-imágenes (4x4 bloques no solapados), de los cuales tras un posterior procesado se obtiene un total de 80 valores denominados *Bincounts* que forman el histograma de bordes que caracteriza dicha imagen.

Proceso de creación:

- Crear la estructura de tipo descriptor denominada *VdEdgeHistogram*.
- Crear las funciones comunes para un descriptor.
- Crear las funciones específicas del descriptor.
- Calcular las dimensiones que tendrán los 16 bloques.
- Filtrar los bloques con la función denominada *DetectionEdges*.
- Calcular el valor absoluto de cada uno de los píxeles.
- Contar el número de píxeles que forman los contornos de la sub-imagen.
- Normalizar el valor resultante de la suma de píxeles.
- cuantificar el valor obtenido de la normalización
- Ejecutar y visualizar los resultados

XML: Es un lenguaje de marcado que ofrece un formato para la descripción de datos estructurados. Esto facilita declaraciones de contenido más precisas y resultados de búsquedas más significativos en varias plataformas. Se puede definir un conjunto ilimitado de etiquetas; además ofrece un marco de trabajo para etiquetar datos estructurados, lo cual se puede realizar de forma exclusiva, lo que permite que un cliente construya búsquedas específicas. Sin XML, es necesario que la aplicación de búsqueda comprenda el

esquema de cada base de datos, en el que se describe cómo se ha generado. Esto es prácticamente imposible, pues cada base de datos describe sus datos de una forma distinta. Sin embargo, con el lenguaje los archivos se pueden clasificar fácilmente en categorías estándar por autor, título, u otros criterios. De esta forma, se podría realizar búsqueda de una forma coherente. Se propone la estructura de descripción de una escena, basado en el estándar MPEG-7. (**Ver anexo 2**)

Operaciones sobre el XML: Es donde se hace el análisis del XML creado por el procesamiento de la información, generando así los términos de indexación.

Índice: Es la estructura de datos que posibilita un acceso veloz a la colección, una vez procesada, con objeto de obtener los documentos relevantes a una consulta. En este caso se crea un índice por cada descriptor especificado por el estándar MPEG-7 y extraído del documento XML.

Indexación: Es la creación de las estructuras de datos adecuadas para permitir un acceso eficiente y eficaz a los documentos. Es un elemento fundamental para los motores de búsqueda y las bases de datos. Tiene como objetivo elaborar un índice que contenga de forma ordenada la información, con la finalidad de obtener resultados de forma más rápida y relevante al momento de realizar una búsqueda. Sin un índice el motor de búsqueda debería escanear el contenido de cada archivo de forma individual cada vez que se iniciara una búsqueda, lo cual considerando la cantidad de información existente, sería un proceso lento y tedioso.



Figura 10 Proceso de indexación

- **Análisis de las características:** Proceso por el cual las características extraídas del video quedan convertidas en etiquetas.

- **Etiquetas Meta:** Agrupan características con un significado común.
- **Eliminación de características repetidas:** Se eliminan las características que no aportarán información relevante para la realización de la búsqueda.
- **Etiquetas claves:** Se definen las características que contienen la información relevante, que se utilizarán en el proceso de búsqueda (Etiquetas de color, título, autor, formato, duración, textura, formas, palabras claves, descripción del video, información descriptiva como personas u objetos que aparecen).
- **Términos de indexación:** Lo constituyen las etiquetas claves.
- **Indexación:** Es donde se realiza la construcción del índice.
- **Índice:** Estructura de datos que posibilita un acceso veloz a la colección, una vez procesada, con objeto de obtener los documentos relevantes a una consulta.

Interfaz: Es la parte del sistema que permitirá la interacción del usuario con el motor de búsqueda. En ella se especificarán los datos que el usuario necesita para realizar la consulta así como la respuesta.

La interfaz permitirá realizar búsquedas por cadenas de caracteres, tendrá la opción de seleccionar una imagen que constituirá el patrón de búsqueda en el índice, además dará la posibilidad de elegir colores, proponer texturas y formas para realizar la búsqueda.

Motor de búsqueda: Aplicación diseñada para gestionar la información que el usuario especifica en la interfaz, buscando la misma en la base de datos. De esta forma cuando se realiza una consulta, el motor de búsqueda se dirige al índice para localizar los elementos deseados, arrojando así resultados precisos y rápidos.

- **Operaciones sobre la consulta**

Se comprueba si la sentencia de la consulta se realizó correctamente.

- **Búsqueda**

Se realiza la búsqueda sobre la consulta especificada en la interfaz utilizando los métodos de búsquedas implementados en la aplicación.

- **Archivos ordenados**

Después de haber recopilado los archivos que tienen relación con la consulta especificada se ordenan de acuerdo a la importancia de los mismos, mostrando este resultado en la interfaz.

3.2. Propuesta de herramientas a utilizar.

Para llevar a cabo el desarrollo de un sistema de software es necesario que se realice bajo un enfoque disciplinado y sistemático; con la utilización de las herramientas que harían posible desarrollar un producto que responda a las necesidades por la cuales fue necesario su desarrollo. A continuación se hace una propuesta de las herramientas más factibles para llevar a cabo la implementación de un sistema de búsqueda como el que describe el presente trabajo. En caso de que un desarrollador considere necesario hacer uso de otra herramienta que no está incluida en las propuestas, se solicita una profunda argumentación que valide la utilización de la misma.

3.2.1. Metodologías de desarrollo de software.

Las metodologías de desarrollo influyen directamente en el proceso de construcción y se elaboran a partir del marco definido por uno o más ciclos de vida. Piattini define la metodología de desarrollo como un “conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software”. La metodología define Quién debe hacer, Qué, Cuándo y Cómo debe hacerlo, constituye la columna vertebral del proceso de desarrollo de software. (Cataldi, 2000)

Las metodologías guían a los ingenieros de una manera más precisa en su trabajo para lograr productos de software que cumplan con la calidad requerida. Para la realización de los productos de software se hace necesario buscar una metodología que se adapte a las condiciones con las que se cuentan para desarrollar determinados proyectos.

Entre las propuestas metodológicas existentes se encuentran las metodologías ágiles (MAs) y las metodologías pesadas o tradicionales. Las metodologías ágiles están orientadas especialmente a proyectos pequeños y permiten que los programadores se concentren solamente en aquellas funciones que se necesitan inmediatamente. Se realizan entregas al cliente frecuentemente, de las cuales se obtiene retroalimentación constante, posibilitando respuestas rápidas a los cambios en el negocio. (Canós, et al., 2003)

Entre las metodologías ágiles más utilizadas está la Programación Extrema (XP), ágil y especialmente diseñada para la resolución de problemas en el menor tiempo posible y con una alta probabilidad de cambio de requisitos, de ahí que exista una ausencia de énfasis en la arquitectura durante las primeras iteraciones y por tanto un déficit de métodos de diseño arquitectónico. Este elemento es considerado como uno de los aspectos negativos de XP. En su mayoría las MAs defienden la idea de hacer sistemas que debieron estar” para ayer” y es en este punto donde la arquitectura comienza a perder valor. (Letelier, et al., 2006)

Otro caso de este tipo de metodologías es el Método de Desarrollo de Sistemas Dinámicos (DSDM) surgida en 1994 y que tiene definida 5 fases en el ciclo de vida, pero ninguna de ellas atribuye fuerza a la arquitectura de software, pues envuelve a grandes rasgos el diseño y de ahí pasa a la implementación de un producto ejecutable que será limado en próximas iteraciones. Existe realimentación en todas sus fases y se centra en el equipo de trabajo y el usuario como elementos base.

Las metodologías tradicionales ponen gran énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos que se deben producir, incluyendo modelado y documentación detallada. Este enfoque tradicional es conveniente para proyectos que requieran mucho tiempo y recurso, pero no resulta muy adecuado para proyectos donde el entorno del sistema es muy cambiante y es necesario reducir los tiempos de desarrollo.

RUP (*Rational Unified Process*) es un proceso de desarrollo de software basado en la programación orientada a objetos y en el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar dentro de las metodologías tradicionales más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Tiene un carácter dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura. Divide en fases el desarrollo del software (Inicio, Elaboración, Construcción, Transición), cada una de estas fases es desarrollada mediante un ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. (Castellano Báez, et al., 2007)

RUP define una serie de roles para cada integrante del equipo de desarrollo de manera que combinan sus esfuerzos según la fase en la que se encuentren, uno de ellos es el máximo responsable de la

arquitectura: el arquitecto del software. El arquitecto del software debe ser polifacético, poseer madurez, visión y una vasta experiencia que le permita tomar decisiones rápidamente y hacer el juicio adecuado, crítico en ausencia de la información completa. Seguidamente se listan las responsabilidades que establece RUP para el arquitecto de software

Los artefactos fundamentales a desarrollar según RUP son: (Díaz Antón, et al., 2003)

- Modelo de Casos de Uso.
- Modelo de Análisis.
- Modelo de Diseño.
- Modelo de Despliegue.
- Modelo de Implementación.
- Arquitectura de referencia.
- Documento de la Arquitectura.

Evaluando que la propuesta realizada:

- Debe estar documentada adecuadamente con los requisitos necesarios para que los futuros informáticos que deban implementar el sistema, perfeccionarlo o agrandarlo tengan una base sólida por donde guiarse.
- No es necesario entregarlo inmediatamente.

Y teniendo en cuenta que RUP:

- Centra su ciclo de vida en la arquitectura.
- Está diseñada de forma que exista un entendimiento continuo y gradual de todos los implicados en el proceso de desarrollo del software.
- Su aplicación ha arrojado buenos resultados en proyectos nacionales e internacionales.
- Delimita roles esenciales que poseen a su cargo la generación de artefactos y documentación correctos por cada fase y flujo que tributan a un buen producto final.

- Es una metodología implantada y una de las más usadas mundialmente así como en la universidad.

Se determina proponer esta metodología para guiar el desarrollo del software.

3.2.2. Herramienta de modelado.

Como lenguaje para el modelado se propone utilizar **UML**, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, permite visualizar, especificar, construir y documentar un sistema de software. Ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

Dentro de las herramientas más utilizadas para el modelado de diagramas UML se encuentran Rational Rose y Visual Paradigm. A continuación se analizan sus características para luego decidir la herramienta a utilizar en el desarrollo del sistema propuesto.

Rational Rose

Fue desarrollado por los creadores de UML (*Booch, Rumbaugh y Jacobson*), y cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables. Unifica todos sus equipos de desarrollo a través del modelado basándose en UML que agiliza la creación del software. Rational Rose permite visualizar, entender y refinar los requerimientos y arquitectura antes de enfrentar el código, evitando esfuerzos desperdiciados en el ciclo de desarrollo.

Incluye también las siguientes funcionalidades: (Menéndez, 2000)

- Los componentes del modelo se pueden controlar independientemente, lo que permite una gestión y un uso de modelos más granular.
- Soporte para compilación y descompilación de las construcciones más habituales de Java 1.5.
- Generación de código en lenguaje Ada, ANSI C++, C++, CORBA, Java y Visual Basic, con funciones configurables de sincronización entre los modelos y el código.

- Soporte para Enterprise Java Beans 2.0.
- Funciones de análisis de calidad de código.
- Complemento de modelado Web que incluye funciones de visualización, modelado y herramientas para desarrollar aplicaciones Web.
- Modelado en UML para diseñar bases de datos, que integra los requisitos de datos y aplicaciones mediante diseños lógicos y analíticos.
- Integración con otras herramientas de desarrollo de IBM Rational.

Visual Paradigm UML 6.0

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. El modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo.

Esta herramienta ofrece: (Visual paradigm, 2002)

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad en múltiples plataformas.
- Ofrecen un mecanismo general para la organización de los modelos, subsistemas, capas agrupando elementos de modelado.

Teniendo en cuenta las características de las herramientas anteriormente descritas se propone utilizar en el desarrollo del sistema **Visual Paradigm UML 6.0 ya que** además de lo expuesto es una herramienta fácil de utilizar que mejora la comunicación del equipo de trabajo. Es la ideal para el desarrollo de un ambiente de software libre con calidad y perfección, razón por lo que ha logrado una mayor aprobación entre los desarrolladores de software en el mundo.

La versión UML 2.1 para la herramienta de modelado Visual Paradigm proporciona un ambiente de modelado visual que satisface la tecnología del software y ofrece herramientas fáciles de usar y fiables, ejemplo de ello son los diferentes diagramas que hacen posible una mejor modelación, diagramas de clases, de secuencia, de estado, de casos de uso entre otros.

3.2.3. Arquitectura de Software (AS).

Definición de Arquitectura de software

Existen varias conceptos que definen a la arquitectura de software, aun no ha surgido alguna de estas definiciones que sea común para todos los desarrolladores o arquitectos. Entre las más significativas se pueden citar algunas como la de Paul Clements (Clements, 2003):

“La arquitectura de software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se le percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.”

“La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”.

“Arquitectura de software es el estudio de la estructura a gran escala y el rendimiento de los sistemas de software. Aspectos importantes de la arquitectura de un sistema incluye la división de funciones entre los módulos del sistema, los medios de comunicación entre módulos, y la representación de la información compartida.” (Garla, et al., 1994)

Analizando estas y otras definiciones existentes se puede deducir que la arquitectura expone la visión común del sistema en su totalidad en la que el equipo de desarrollo y los usuarios deben concordar, por lo que describe los elementos más importantes del modelo para su construcción como base para una mayor y mejor comprensión, para poder desarrollarlo y producirlo correctamente.

3.2.4. Estilos arquitectónicos.

El análisis de los estilos arquitectónicos es de gran importancia, muestran las posibles formas en que el sistema puede estar estructurado y a su vez son previos a la elección de las herramientas y plataforma a

utilizar para el desarrollo del mismo. Los estilos arquitectónicos son arquitecturas de software comunes, marcos de referencias arquitectónicas, formas comunes o clases de sistemas, que están compuestos por una serie de elementos y restricciones arquitectónicas.

Una correcta elección del estilo a utilizar sería la clave del trabajo arquitectónico, por lo que se hace necesario tener en cuenta los estilos existentes para establecer la arquitectura del sistema, y así proponer el que más se ajuste al problema, teniendo en cuenta que el sistema puede integrar varios estilos combinados que de la misma manera darían solución al mismo.

Entre los principales estilos arquitectónicos se encuentran (Abreu Bartomeo, et al., 2008):

Estilos de Flujo de Datos

- Tubería y filtros

Estilos Centrados en Datos

- Arquitecturas de Pizarra o repositorio

Estilos de llamada y retorno

- Modelo-Vista-Controlador (MVC)
- Arquitectura en capas
- Arquitecturas orientadas a objetos
- Arquitecturas basadas en componentes

Estilos de código móvil

- Arquitectura de máquinas virtuales

Estilos heterogéneos

- Sistemas de control de procesos
- Arquitecturas basadas en atributos

Estilos Peer-to-Peer

- Arquitecturas basadas en eventos

- Arquitecturas orientadas a servicios
- Arquitecturas basadas en recursos

Modelo-Vista-Controlador

Separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases. Dicha separación se comporta como sigue:

- **Modelo:** Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado y a instrucciones de cambiar el estado.
- **Vista:** Maneja la visualización de la información.
- **Controlador:** Interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado. Esta forma de separar el trabajo en diferentes niveles facilita testear la aplicación independientemente de la representación visual. Además, si son necesarias modificaciones en el transcurso del trabajo pueden hacerse solo en la capa necesaria sin interferir en las demás. Generalmente los cambios gráficos son de menor esfuerzo y más frecuentes que un cambio en requerimientos o funcionalidades, de aquí que la independencia entre estas tareas sea cómoda.

Entre las ventajas del estilo Modelo-Vista-Controlador están las siguientes:

Soporte de múltiples vistas: Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación Web pueden utilizar el mismo modelo de objetos mostrado de maneras diferentes.

Adaptación al cambio: Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celulares. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Una desventaja que tiene este modelo es el costo de actualizaciones frecuentes: Si el modelo experimenta cambios frecuentes, por ejemplo, podría desbordar las vistas con una lluvia de requerimientos de actualización.

Arquitecturas en Capas

Garlan y Shaw definen el estilo en capas como una organización jerárquica, tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones dadas por la capa inferior. El estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales. Permite realizar optimizaciones y refinamientos enfocando los cambios en un solo lugar. Proporciona amplia reutilización dada la división bien definida de responsabilidades. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa en la medida que soporten las mismas interfaces de cara a las capas adyacentes. Esto conduce a la posibilidad de definir interfaces de capa estándar, a partir de las cuales se pueden construir extensiones o prestaciones específicas (Cuesta, 2007).

Una especialización muy usada de la arquitectura en capas es la arquitectura de tres capas donde se observan muy bien delimitadas las responsabilidades de cada funcionalidad en la aplicación. En una arquitectura de tres capas, cada capa está muy bien delimitada de las demás. Una capa superior interactúa con una capa inferior mediante interfaces que definen las funcionalidades que la misma debe brindar. Las capas de la aplicación pueden residir tanto en el mismo nodo físico como en nodos separados. Las tres capas son:

- Presentación (interfaz de usuario, presentación).
- Lógica de negocio (donde se encuentra modelado el negocio, lógica del negocio)
- Acceso a datos (persistencia, acceso a datos).

Arquitecturas Orientadas a Objetos

Esta arquitectura ha sido incluida, en más de una agrupación de estilos, sin embargo dentro de las Llamada y Retorno ha sido más utilizada en los últimos años. Como su nombre lo indica la definición de este estilo se acoge a las prácticas de programación orientada a objetos: los componentes del estilo son los objetos y estos responden a encapsulamiento, herencia y polimorfismo. Así mismo los objetos constituyen elementos reutilizables en el ambiente de desarrollo y son responsables de preservar la integridad de su propia representación. La principal deficiencia del estilo radica en que la modificación de un objeto implica la modificación de cada elemento (objeto, método) que lo invoca.

Para un posterior desarrollo del sistema de búsqueda se propone la utilización de dos estilos arquitectónicos el Modelo Vista Controlador y la Arquitectura en Capas:

Modelo Vista Controlador (MVC) Separa la lógica de negocio y la presentación, lo que trae como consecuencia que al cambiar este último no sea necesario cambiar la lógica de negocio. Esto permite que el mantenimiento de la aplicación sea más sencillo, garantiza una mayor organización y reutilización de los componentes del sistema obteniendo una solución altamente flexible.

Arquitectura en Capas Permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales. Permite realizar optimizaciones y refinamientos enfocando los cambios en un solo lugar. Proporciona amplia reutilización dada la división bien definida de responsabilidades. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa en la medida que soporten las mismas interfaces de cara a las capas adyacentes.

Una especialización muy usada de la arquitectura en capas es la arquitectura de tres capas donde se observan muy bien delimitadas las responsabilidades de cada funcionalidad en la aplicación. En una arquitectura de tres capas, cada capa está muy bien delimitada de las demás. Una capa superior interactúa con una capa inferior mediante interfaces que definen las funcionalidades que la misma debe brindar.

Patrones arquitectónicos

Los Patrones Arquitectónicos constituyen soluciones reutilizables a problemas específicos en un marco dado. Se implementan a través de un conocimiento específico acumulado por la experiencia en un dominio. Un patrón es una solución a un problema en un contexto, codifica conocimiento específico acumulado por la experiencia en un dominio. Un sistema bien estructurado está lleno de patrones. Surgen como una manera organizativa de brindar una respuesta unificada ante situaciones problemáticas similares. Los Patrones Arquitectónicos proporcionan una arquitectura uniforme que permite una fácil expansión, mantenimiento y modificación de la aplicación. Un patrón se describe esencialmente de la siguiente manera (Abreu Bartomeo, et al., 2008):

- Consecuencias: pros y contras al utilizar la solución propuesta.
- Patrones relacionados: patrones con similar funcionalidad que deben ser estimados como alternativa.

- Descripción del problema: hace referencia al fenómeno que promueve la necesidad de su aplicación.
- Consideraciones: aspectos a considerar dentro de la solución propuesta.
- Solución general: constituye una descripción básica de la solución en sí.

Existen diferentes tipos de patrones en dependencia del nivel conceptual donde se apliquen, ejemplo de ellos son los patrones de diseño, de implementación y de arquitectura. Dependiendo del propósito funcional de los patrones, estos se distinguen en los siguientes tipos:

- Fundamental: construye bloques de otros patrones.
- Presentación: estandariza la visualización de los datos.
- Creación: concepción condicional de objetos.
- Integración: comunicación con sistemas y recursos externos.
- Particionamiento: organización y separación de lógica compleja, conceptos y actores en múltiples clases.
- Estructurales: separación de la presentación, las estructuras de datos, la lógica de negocio y procesamiento de eventos en bloques funcionales.
- Comportamiento: coordina y organiza el estado de los objetos.
- Concurrencia: maneja el acceso concurrente a los distintos recursos.

Patrones de diseño

Cuando se realiza el diseño de un proyecto de software aparecen problemas que es preciso darle solución para que este funcione correctamente. Muchas veces las soluciones son un poco difíciles, aunque siempre existen, provocando consigo el atraso en el desarrollo de las aplicaciones. Para darle solución a estos problemas de manera más fácil existen los patrones de diseño, que son un conjunto de soluciones a problemas que generalmente se encuentran en el diseño de un sistema. Cada patrón explica cómo resolver un determinado problema, bajo determinadas circunstancias, obteniendo como resultado un producto con mayor calidad, más usable y funcional. Usar determinados patrones para el desarrollo de la

aplicación que se propone está en dependencia de los problemas que surjan durante el ciclo de desarrollo de la misma, inicialmente se proponen los siguientes patrones.

Patrones GRASP seleccionados (Abreu Bartomeo, y otros, 2008)

Los patrones GRASP, son patrones que asignan responsabilidades que están presentes en la mayoría de las aplicaciones, generalmente cuando surge algún problema son utilizados por los desarrolladores aun sin saber que están haciendo uso de ellos.

Creador: En las acciones se crean los objetos de las clases que representan las entidades.

Experto: Este es uno de los más utilizados para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

Alta Cohesión: Permite asignar responsabilidades con una alta cohesión, cada clase debe estar formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

Controlador: Es un evento generado por actores externos. Se asocian con operaciones del sistema, como respuestas a los eventos del sistema, tal como se relacionan los mensajes y los métodos. Normalmente un controlador delega en otros objetos el trabajo que se necesita hacer, coordina o controla la actividad. No realiza mucho trabajo por sí mismo.

Bajo Acoplamiento: Asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.) El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema.

Patrones GOF seleccionados (Prieto, 2008)

En la categoría Creacionales:

Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

Ventajas que ofrece:

Acceso controlado a la única instancia; puede tener un control estricto sobre cómo y cuándo acceden los clientes a la instancia. Espacio de nombres reducido; el patrón *Singleton* es una mejora sobre las variables globales. Permite el refinamiento de operaciones y la representación; se puede crear una subclase de *Singleton*. Permite un número variable de instancias; el patrón hace que sea fácil cambiar de opinión y permitir más de una instancia de la clase *Singleton*.

Abstract Factory (Fábrica abstracta): Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando. Cuando el *framework* necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea.

Ventajas que ofrece:

Aísla las clases de implementación: ayuda a controlar los objetos que se creen y encapsula la responsabilidad de creación. Hace fácil el intercambio de familias de productos sin mezclarse, permitiendo configurar un sistema con una de entre varias familias de productos. Fomenta la consistencia entre productos.

En la categoría estructurales:

Decorator (Envoltorio): Añade funcionalidad a una clase, dinámicamente. Permite una mayor flexibilidad que la herencia, incluso puede decorarse un objeto varias veces con el mismo decorador, lo que sería imposible con la herencia. Favorece la definición de interfaces y clases bases ligeras. Sólo se carga con las funcionalidades que se necesitan, no son necesarias megaclases que sean capaces de hacerlo todo.

Composite (Objeto compuesto): Permite tratar objetos compuestos. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, al poseer todos ellos una interfaz común, se tratan de la misma manera.

Ventajas que ofrece:

Define la jerarquía de clase en base a objetos simples y a composiciones. El cliente puede esperar a un De forma general el diseño de patrones se incluyen dentro de las mejores prácticas de la Ingeniería del Software debido a que ofrece numerosas ventajas como:

- Consistencia y calidad en el código fuente.
- Reusabilidad del código fuente.
- Mejor rendimiento de las aplicaciones.
- Aplicaciones más fáciles de mantener.

3.2.5. Lenguaje de programación para una aplicación Web.

Lenguaje Python

Guido van Rossum, más conocido como Guido, creó Python en 1990, un lenguaje de programación de scripts. Es un lenguaje limpio y elegante, permite dividir el programa en módulos reutilizables para otros programas. Viene con una gran colección de módulos estándar que se pueden utilizar como base para los programas.

Python es un lenguaje interpretado, esto permite el ahorro de tiempo en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa. El principal objetivo que persigue este lenguaje es la facilidad, tanto de lectura, como de diseño.

Ventajas y desventajas que ofrece: (Perdomo, 2006)

Ventajas:

- Es un lenguaje muy potente, elegante, claro, comprensible y flexible.
- Tiene óptima potencia y rendimiento.
- Dinámico, permite modificar su código fuente, haciendo aplicaciones de modo sencillo.
- Es un lenguaje Orientado a Objetos e interpretado.

Desventajas:

- Es un lenguaje un poco lento.
- Orientación a objetos impura.
- Falta de buenos Entornos Integrados de Desarrollo (IDE's).

Lenguaje ASP

Microsoft introdujo esta tecnología llamada Active Server Pages en diciembre de 1996, es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente. La definición contextual de Microsoft es que "Las Active Server Pages (ASP) son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para la Web".

Con la aparición de la plataforma .NET se inició una nueva era en el campo de la programación de aplicaciones apareciendo ASP.NET una parte de la plataforma .NET de Microsoft. ASP.NET es una estructura de programación que permite el desarrollo de aplicaciones Web dirigidas a corporaciones. Constituye la forma más rápida y escalable de desarrollar, implementar y ejecutar aplicaciones Web en cualquier navegador o dispositivo. Es usado por programadores para construir sitios Web dinámicos, aplicaciones Web y servicios Web XML. ASP.NET facilita el desarrollo de aplicaciones si se compara con el modelo ASP clásico, por lo que la productividad de los programadores es mejor. Esta plataforma permite dotar de funciones adicionales a una aplicación Web y escribir una menor cantidad de código, entre otras características.

En la actualidad una aplicación ASP.NET puede ejecutarse de dos formas distintas: Aplicaciones cliente/servidor y las aplicaciones que utilizan el navegador estas últimas están caracterizadas por contar con una interfaz rica y muy útil.

Ventajas y desventajas que ofrece: (Acosta Pintado, 2008)

Ventajas:

- Es un lenguaje liviano.
- Se puede utilizar en cualquier computadora que esté conectada a la red y tenga instalado un navegador.
- Es fácil de programar y tiene muchas utilidades que puedes ser modificadas.
- Permite a los proveedores de la Web ofrecer aplicaciones de negocio interactivo.
- Se encarga de detectar el tipo de navegador utilizado por el cliente determinando la versión HTML que este soporta.

Desventajas:

- Tiene que correr en PCs normales que tengan como sistema operativo Windows y un servidor Web.
- Los motores scripting utilizados no son universalmente conocidos.

PHP5

PHP (*Hypertext Preprocessor*) es sencillo, de sintaxis cómoda y similar a la de otros lenguajes como C o C++, es rápido a pesar de ser interpretado, es multiplataforma, es decir, está soportado para cualquier arquitectura, además los scripts no necesitan ser modificados entre una plataforma y otra, esto es una ventaja muy grande respecto a otros lenguajes. Dispone de una gran cantidad de librerías y se le pueden agregar extensiones fácilmente. Es un lenguaje basado en herramientas con licencia de software libre, es decir, no hay que pagar licencias, si es limitada su distribución y se puede ampliar con nuevas funcionalidades si se desea. Es ideal tanto para el que comienza a desarrollar aplicaciones Web como para el desarrollador experimentado. Por estar completamente escrito en C, se ejecuta rápidamente utilizando poca memoria. Puede ser compilado y ejecutado en diversas plataformas, incluyendo diferentes versiones. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al sistema operativo. PHP5 hace un cambio en el manejo de los objetos a diferencia de PHP4, o sea, PHP4 trata los objetos igual que otros tipos de datos básicos como los enteros o los arreglos, de modo que cuando se realizan operaciones sobre un objeto el mismo es copiado íntegramente, mientras que en PHP5 las variables que nombran objetos son en realidad referencias. Otro de los elementos significativos propio de PHP5 es que incluye modificadores de control de acceso para implementar el encapsulamiento.

Ventajas y desventajas que ofrece: (PHP, 2007)

Ventajas:

- PHP soporta prácticamente cualquier plataforma y es independiente del SO. utilizando el mismo código fuente,
- Interactúa con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix y PostgreSQL.

- Gran variedad de módulos.
- PHP es *Open Source*, lo cual significa que el usuario no depende de una compañía específica para solucionar cosas que no funcionan.
- Ampliamente documentado.
- Código fuente disponible.
- Sencillo de aprender y utilizar.
- Seguro (evidentemente tiene errores pero se solucionan mucho antes que otros sistemas propietarios).
- Amplia base de usuarios.

Desventajas:

- Es un lenguaje interpretado, se necesita tener el intérprete de PHP, y esto trae un alto consumo de las aplicaciones.

Se propone el lenguaje de programación PHP específicamente su última versión, este posee un fácil acceso a todos los desarrolladores, tiene buen rendimiento y flexibilidad a la hora de realizar páginas Web dinámicas. Ofrece varias funciones para la explotación de bases de datos sin complicaciones, realizando determinadas acciones de forma fácil y eficaz.

Este lenguaje se puede utilizar en varios sistemas operativos, posee una amplia documentación por lo que resulta fácil de aprender, cuenta con una biblioteca que crece a medida que sus nuevas versiones van surgiendo, además cuenta con disímiles funciones fáciles de utilizar siendo solo necesario hacer llamadas de forma apropiada y especificar los parámetros necesarios para realizar tareas como el acceso a base de datos, encriptación, envío de correo, creación de PDF y XML.

Framework para PHP

Para desarrollar aplicaciones Web complejas no es recomendable utilizar solo PHP, se hace necesario que el código sea ligero, legible y efectivo para asegurar el mantenimiento y las ampliaciones futuras de la aplicación. Se necesita un *framework* que sea maduro, bien documentado y con una gran comunidad de usuarios que lo apoye.

Elegir como *Framework* para PHP a Symfony permite realizar una aplicación sin problemas ni sorpresas, este se basa en la experiencia utilizando las mejores prácticas de desarrollo Web y grandes bibliotecas que permiten reducir el tiempo de desarrollo del software. Proporciona componentes y herramientas para construir aplicaciones Web complicadas con mayor rapidez. Gracias a su versatilidad y sus posibilidades de configuración, Symfony es un *framework* adecuado para cualquier proyecto de aplicación Web, es además un proyecto de software libre exitoso.

Está desarrollado en PHP5 y se puede utilizar en diferentes plataformas como *nix (Unix, Linux), es compatible con diferentes gestores de base de datos como PostgreSQL. Symfony es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones Web. Este emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación Web, su estructura interna se ha diseñado para obtener lo mejor de este patrón y la mayor facilidad de uso. A continuación se detallan las características de este *framework*.

Symfony

Symfony es un *framework* completo diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones Web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja.

Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación Web.

Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas (Unix, Linux) como en plataformas Windows.

Características de Symfony (Symfony, 2005)

- Fácil de instalar y configurar en la mayoría de plataformas
- Independiente del sistema gestor de bases de datos
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos

- Sigue la mayoría de mejores prácticas y patrones de diseño para la Web
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo Symfony.
- Código fácil de leer que incluye comentarios de php.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

3.2.6. Lenguaje de programación para una aplicación de escritorio.

JAVA

El lenguaje de programación Java, fue diseñado por la compañía Sun Microsystems Inc²⁷, con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas (redes de computadoras formadas por más de un tipo de computadora), y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa de Java puede ejecutarse en cualquier máquina o plataforma. La compañía Sun describe el lenguaje Java como "simple, orientado a objetos, interpretado, robusto, portable, de altas prestaciones, multitarea y dinámico". (Hrenández Peñalver, 2008)

Simple: Java es un lenguaje sencillo, con pocas palabras reservadas que hacen que su aprendizaje sea rápido. Además su utilización es independiente del tipo de aplicación que se desee desarrollar por lo que una vez aprendido es fácil aplicarlo a otros tipos de aplicación.

Orientado a objeto: El paradigma orientado a objetos hace que la aplicación se vea como un conjunto de objetos no de celdas de memoria, dando una visión de lo que se programa más cercana al problema a tratar que a la máquina donde se quiere implementar la solución. Además de la encapsulación el paradigma orientado a objetos destaca por otras cualidades como la herencia o el polimorfismo que facilitan el diseño y la implementación.

Interpretado: Al ser un lenguaje interpretado es independiente de la arquitectura en la que se vaya a ejecutar.

²⁷ <http://www.sun.com/>

Multitarea: Java incorpora mecanismos para que el software desarrollado pueda ser multitarea como el uso de diferentes hilos de ejecución (*threads*).

Robusto: Al no trabajar directamente con punteros y posiciones de memoria se evita que el programa intente acceder a zonas que no le corresponden.

Dinámico: Java está en constante evolución, consiguiéndose mejores rendimientos e incorporando nuevas funcionalidades a medida que Sun proporciona nuevas versiones.

Portable: El código generado por el compilador de Java es muy portable pues es el mismo independientemente de la plataforma, puesto que solo depende de la máquina virtual donde se ejecuta.

Seguro: Los programas escritos en Java no acceden directamente a los recursos de la máquina física en la que se ejecutan sino que se ejecutan en una máquina virtual, de manera que el entorno está más controlado.

De arquitectura neutra: Como el programa en java no se ejecuta directamente sobre la máquina sino que se ejecuta en la máquina virtual es indiferente la arquitectura real donde se ejecuta esta.

De altas prestaciones: Java cuenta con gran número de paquetes muy probados para resolver diferentes problemas o satisfacer las necesidades de componentes de cualquier desarrollo. Esto permite un buen rendimiento utilizando elementos estándar del lenguaje.

C#

Fue creado por Anders Heljsberg, quien además creó otros lenguajes y entornos como Turbo Pascal, Delphi o Visual J++. Es un lenguaje de propósito general orientado a objetos creado por Microsoft para su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes. El lenguaje C# fue diseñado para combinar el control a bajo nivel de lenguajes como C y la velocidad de programación de lenguajes como Visual Basic.

Entre las principales características se encuentran la facilidad de uso, además el ambiente de trabajo es muy cómodo, amigable y clásico en las aplicaciones de Windows. Este lenguaje presenta las características necesarias para considerarlo como un lenguaje orientado a objetos, tales son: encapsulación, herencia y polimorfismo, esta forma de programación hace que el código sea reutilizable.

Permite la administración de memoria, la seguridad en el manejo de datos, es compatible con otros lenguajes y permite el uso de operadores.

Se considera Java el lenguaje ideal para el desarrollo de la aplicación, este tiene como principal objetivo el funcionamiento fuera de la plataforma donde se ejecuta, significa que programas escritos en este lenguaje pueden ejecutarse igualmente en cualquier tipo de hardware. Es un lenguaje moderno que elimina la complejidad que existe en otros tipos de lenguajes como C y C++, tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir muchos errores y características que resultan complejas e inseguras como la utilización de punteros, por lo que es considerado un lenguaje más fiable. Es totalmente orientado a objeto, un modo de programación más limpia y segura.

Tiene un nivel de aprendizaje rápido y de forma sencilla, implementa barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real. Soporta sincronización de múltiples hilos de ejecución para crear programas multitareas, posee además varios entornos de desarrollo totalmente gratuitos como NetBeans y Eclipse. Cada vez incorpora más facilidades para la creación y manipulación de gráficos, así como el acceso a bases de datos. Es fiable numerosas empresas han desarrollado excelentes proyectos por su robustez, diseño y su fácil portabilidad.

Framework para Java

Hibernate (Suárez González, 2003)

Es un *framework* de código abierto el cual se puede adaptar a las necesidades del usuario, es una capa de persistencia objeto/relacional y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones y un gran número de tipos de datos. De una manera muy rápida y optimizada se puede generar base de datos en cualquiera de los entornos soportados: Oracle, MySql, PostgreSQL.

Uno de los posibles procesos de desarrollo consiste en, una vez que se tenga el diseño de datos realizado, mapear este a ficheros XML siguiendo la estructura de mapeo de Hibernate. Pudiéndose generar el código de los objetos persistentes en clases Java y también crear base de datos independientemente del entorno escogido. Hibernate se integra en cualquier tipo de aplicación justo por encima del contenedor de datos.

IDE NetBeans

Varias razones hicieron posibles la selección de NetBeans como el entorno de desarrollo a utilizar, es un producto libre, gratuito, sin restricciones de uso y la solución más completa para programar en Java, que fue precisamente el lenguaje seleccionado para la aplicación de escritorio, aunque también puede soportar otros lenguajes de programación. Contiene todos los módulos necesarios para el desarrollo de aplicaciones escritas en lenguaje Java, permitiendo a los desarrolladores comenzar el trabajo inmediatamente.

NetBeans es un IDE que permite la elaboración de potentes aplicaciones de escritorio de forma sencilla, además sin importar el sistema operativo donde se instale puede crear interfaces gráficas de forma visual. La última versión NetBeans IDE 6.5 soporta el desarrollo de aplicaciones empresariales, herramientas de desarrollos visuales, herramientas de esquema XML y modelado UML. NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. (NetBeans, 2007)

3.2.7. Sistema Gestor de Base Datos (SGBD).

Oracle

Oracle es un sistema de gestión de base de datos relacional, fabricado por *Oracle Corporation*. Se considera uno de los sistemas de bases de datos más completos. Es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Para desarrollar en Oracle se utiliza PL/SQL un lenguaje de 5ª generación, bastante potente para tratar y gestionar la base de datos. Es un producto que por su elevado precio solo es utilizado por empresas grandes y multinacionales y aunque su dominio en el mercado de servidores empresariales ha sido casi total, recientemente sufre la competencia de gestores de bases de datos comerciales y otros con licencia Software Libre.

Ventajas y desventajas que ofrece: (Prado, et al., 2004)

Ventajas:

- Es el motor de base de datos relacional más utilizado a nivel mundial.
- Permite el uso de particiones y la administración para la mejora de la eficiencia.
- El software del servidor puede ejecutarse en multitud de sistemas operativos.
- Presenta un aceptable soporte.

Desventajas:

- Sus licencias son excesivamente caras.
- Es importante la necesidad de ajuste para evitar una mal configuración.
- Existe poca bibliografía sobre asuntos técnicos de instalación y administración.

MySQL

MySQL es un sistema de gestión de bases de datos relacionales, o sea que almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén, lo que añade velocidad y flexibilidad, por esta razón proporciona un servidor de base de datos SQL muy rápido, multiusuario, multihilo y robusto. El servidor está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido. Posee gran aceptación debido a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Ventajas y desventajas que ofrece: (Pecos, 2002)

Ventajas:

- Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.
- Seguridad: Determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas en forma de permisos y privilegios, lo que permite compartir datos sin que peligre la integridad y protegiendo determinados contenidos.
- Escalabilidad: Es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- Conectividad: Permite conexiones entre máquinas con distintos sistemas operativos. Servidores Linux o Unix, usando MySQL, pueden servir datos para ordenadores con Windows, Linux, Solaris.
- Recuperación automática ante fallas: Si se da de baja de forma anormal, no suele perder información ni corromper los datos y completa las transacciones que no se terminaron.
- Utiliza el potente lenguaje de consultas SQL.
- Permite implementar sub-consultas a partir de la versión 4.1 y desencadenadores desde la versión 5.0.2.

Desventajas:

- Está prohibida la descarga de MySQL. Este SGBD se hizo software propietario desde enero del 2008 cuando lo compró la empresa norteamericana Sun Microsystems.
- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas por ejemplo Access.
- No permite el modo de autenticación local (seguridad integrada de Windows), sólo el modo estándar.
- No tiene un panel de control gráfico y detallado.

PostgreSQL

Es un SGBD Objeto-Relacional basado en el proyecto POSTGRES, de la universidad de Berkeley. Es una derivación libre de este proyecto, y utiliza el lenguaje SQL. Este proyecto lleva más de una década de desarrollo, siendo hoy día, el sistema libre más avanzado, soportando la gran mayoría de las transacciones SQL y control concurrente. A diferencia de la mayoría de los sistemas de bases de datos que usan bloqueos para el control de concurrencia, *PostgreSQL* mantiene la consistencia de los datos en un modelo multiversión, esto significa que mientras se consulta una base de datos, cada transacción mantiene una imagen o versión de los datos, sin tener en cuenta el estado actual de la información (referencia).

Entre sus principales ventajas se subrayan las siguientes (PostgreSQL, 2006)

- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios Web que atienden un gran número de solicitudes.
- Puede ser instalado un número ilimitado de veces sin temor de sobrepasar la licencia.
- Posee estabilidad y confiabilidad legendarias.
- Es extensible a través del código fuente disponible sin costos adicionales.
- Es multiplataforma, disponible en Linux, Unix, Mac Os X y Windows, entre otros sistemas operativos.
- Permite implementar reglas, vistas, disparadores, sub-consultas y procedimientos almacenados.

- Posee herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL.

Los gestores analizados son muy buenos, dentro de estos se seleccionó como el más apropiado PostgreSQL quien por sus características de código abierto constituye el sistema libre más avanzado. Es el que más se adecua a las necesidades de la aplicación, para almacenar grandes volúmenes de datos y permitir el acceso de diferentes usuarios al mismo tiempo, acreditando permiso a cada uno de ellos. Es un gestor multiplataforma.

3.2.8. Sistema Operativo.

El sistema operativo es el programa (o software) más importante de un ordenador, pues permite el funcionamiento de los demás programas. Es en sí mismo un programa considerado muy especial, complejo e importante. Desempeña 5 funciones básicas en la operación de un sistema informático: suministro de interfaz al usuario, administración de recursos, administración de archivos, administración de tareas y servicio de soporte y utilidades.

Existen distintas tendencias del software actual en cuanto al grado de libertad (software libre y propietario). El software libre viene acompañado de su código fuente por lo que puede ser distribuido, modificado y copiado, a diferencia del software propietario o privativo que ofrece a los usuarios limitadas posibilidades pues su código fuente no está disponible. Se han desarrollado diferentes familias de sistemas operativos la Familia de Windows, Familia Macintosh y la Familia UNIX dentro de las cuales se han desarrollado varios sistemas operativos que han ido evolucionando con el surgimiento de nuevas versiones. A continuación se caracterizan dos de los sistemas operativos más representativos.

Windows

Es una familia de sistema operativos desarrollados y comercializados por Microsoft. Algunas de las distribuciones de Windows son: Windows 3.11, Windows 95, Windows 98, Windows Millenium, Windows XP, Windows NT Server. Las sucesivas versiones han sido una evolución de la original, produciéndose un cambio de mayor envergadura y ofreciendo todas las herramientas necesarias para el trabajo diario con el computador.

Cada vez las versiones son más potentes y proporcionan al usuario plataformas de trabajo más sólidas, estas presentan más funcionalidades que la versión previa, y generalmente son compatibles con las versiones anteriores. Es un sistema operativo propietario.

GNU/Linux

GNU surgió como un proyecto para desarrollar aplicaciones totalmente libres y compatibles con UNIX. La combinación de GNU y Linux llegó a la meta inicial de un sistema operativo libre, actualmente se usa en millones de ordenadores.

Como sistema operativo, GNU/Linux es muy eficiente y tiene un excelente diseño. Es multitarea, multiusuario, multiplataforma y multiprocesador; protege la memoria para que un programa no pueda hacer caer al resto del sistema; carga sólo las partes de un programa que se usan; comparte la memoria entre programas aumentando la velocidad y disminuyendo el uso de memoria; usa un sistema de memoria virtual por páginas; permite usar bibliotecas enlazadas tanto estática como dinámicamente; se distribuye con código fuente; soporta redes tanto en TCP/IP como en otros protocolos. Lo que hace realmente único a GNU/Linux entre otros sistemas operativos es que conlleva una idea del modo en que debería desarrollarse el software.

Aunque el nombre correcto es GNU/Linux muchas personas usan incorrectamente "Linux" para nombrar el sistema operativo, es por eso que Linux se usa más. El sistema GNU/Linux es un componente central, el cual se transforma en muchos productos diferentes: las llamadas variantes o distribuciones, estas cambian la apariencia y funcionamiento del sistema operativo completamente.

Distribuciones de Linux: (Linux online, 2008)

Linux Mint: Es una distribución del sistema operativo GNU/Linux basada en Ubuntu su última versión estable fue lanzada el 30 de junio del 2007 con el nombre Cassandra, posee la propuesta de producir un escritorio elegante y actualizado. Linux Mint es compatible con Ubuntu y comparte los mismos repositorios. En conclusión es una distribución excelente para uso hogareño o personas que se están iniciando en el tema de GNU/Linux, es realmente fácil de instalar y usar. La versión más completa de Linux Mint puede no cumplir con lo que sería GNU, pues contiene Software propietario.

Freespire: Es un sistema operativo basado en Linux, distribuido en formato LiveCD, que combina lo mejor que ofrece el software libre y de código abierto (administrado por la comunidad, distribución gratuita,

código abierto, etc.), pero también ofrece a los usuarios la posibilidad de incluir soporte para formatos propietarios y aplicaciones según sus necesidades.

OpenSUSE: Es el nombre de la distribución y proyecto libre auspiciado por Novell para el desarrollo y mantenimiento de un sistema operativo basado en Linux. Antes de la aparición de openSUSE, el desarrollo de esta distribución, anteriormente conocida como SUSE Linux, se realizaba a puerta cerrada. Ahora, el proceso está abierto a cualquier programador y usuario que desee contribuir al desarrollo de openSUSE.

Fedora: El objetivo de este proyecto es conseguir un sistema operativo de propósito general y basado exclusivamente en software libre con el apoyo de la comunidad Linux. Los ingenieros de Red Hat continúan participando en la construcción y desarrollo de este proyecto e invitan y fomentan la participación de miembros de la comunidad Linux.

GNU/Linux o simplemente Linux es el sistema operativo de mayor crecimiento en los últimos años. Su éxito no es casualidad. Sus ventajas distintivas son las siguientes:

- Es un software libre, no hay que pagar nada por la instalación del sistema, es gratuito.
- Es ideal para las redes, fue diseñado en internet.
- Es 100% configurable.
- Es un sistema más seguro y confiable, dispone de un código fuente.
- Existe muchísima documentación en varios idiomas.
- Soporte de grandes empresas como IBM, Motorola, Sun entre otras.
- Su licencia permite la copia, modificación y distribución del software.
- Es un sistema operativo caracterizado por su robustez, pueden pasar años sin necesidad de apagar o reiniciar el equipo.
- Su instalación no es sencilla pero permite personalizar los paquetes que se quieran instalar.
- Ofrece una alta compatibilidad con el hardware ofreciendo actualizaciones frecuentes.
- Libre de virus, aún no se conoce ningún virus para Linux.

- Ideal para la programación, se puede programar en Linux para distintas plataformas.

Ubuntu: Es una distribución GNU/Linux que ofrece un sistema operativo predominante enfocado a ordenadores personales, aunque también proporciona soporte para servidores. Es una de las más importantes distribuciones de GNU/Linux a nivel mundial. Se basa en Debian y concentra su objetivo en la facilidad y libertad de uso, la fluida instalación y los lanzamientos regulares cada 6 meses. Entre las ventajas que presenta Ubuntu se encuentra la rapidez con la que enciende el ordenador, el rendimiento en redes es más rápido y con mayor estabilidad, la instalación es muy fácil y viene por defecto programas instalados. Su última versión es Ubuntu 8.10 y fue lanzada el 30 de octubre del 2008, la misma, oferta varias mejoras entre ellas la interacción escalable entre escritorio y dispositivo móvil, soporte dinámico del módulo Kernel y mejora la conectividad.

Basándose en el principio de llegar a implantar un software libre que permita a países subdesarrollados como Cuba migrar a una tecnología independiente y adquirir software a costos muy bajos o gratuitos, contribuyendo de esta forma al desarrollo de la economía, se seleccionó el sistema operativo GNU/Linux en su versión Ubuntu 8.10.

Este sistema operativo es la mejor alternativa frente a los demás, pues permite traducir el software a cualquier lenguaje maximizando el número de personas que pueden usarlo. Se tiene el código fuente lo que permite un control estricto del software, da libertad al usuario de estudiarlo, modificarlo, ejecutarlo, distribuirlo y mejorarlo o sea tener un libre uso del mismo. Se propone utilizar la distribución Ubuntu, un software libre sin ninguna restricción en su licencia.

3.3. Resumen del capítulo.

En el capítulo se realizó la propuesta del modelo conceptual del sistema de búsqueda, en la cual se identifican los principales componentes (gestor documental, procesamiento de la información, indexación, motor de búsqueda) para la realización de una aplicación de búsqueda de videos digitales a través de los metadatos definidos en el estándar MPEG-7. Además, teniendo en cuenta las características de la Universidad de las Ciencias Informáticas en cuanto a las tecnologías que se utilizan, se proponen las principales herramientas que permitirían el desarrollo de la aplicación (RUP, UML, Visual Paradigm, PHP, Symfony, PostgreSQL).

Conclusiones

El modelo propuesto surge luego de analizar las tendencias actuales de la búsqueda de contenidos audiovisuales, las particularidades y necesidades de un sistema de búsqueda de videos digitales a través del contenido en el Polo de Video y Sonido Digital de la Facultad 9, y de comprender cuán importante resulta el desarrollo de un sistema de este tipo para la gestión de el gran volumen de información audiovisual manejado en el polo. Se puede concluir que:

- El modelo propuesto será el punto de partida para desarrollar un sistema de búsqueda de videos digitales basado en el contenido de los mismos.
- La búsqueda automática por contenido en materiales audiovisuales no está totalmente desarrollada, pues no es una tarea fácil para los desarrolladores, al ser muy costosa, y computacionalmente compleja.
- Al constituir un archivo de texto, los metadatos son la vía ideal para el acceso al contenido de los videos digitales. La indización de estos metadatos permitirá la obtención de mejores resultados en las búsquedas.
- El MPEG-7, es la propuesta indicada para la descripción de contenido de materiales audiovisuales en el polo de Video y Sonido Digital, pues está especialmente diseñando para facilitar la recuperación de estos materiales en sistemas de búsquedas.

Recomendaciones

Este trabajo propone un modelo para desarrollar un sistema de búsquedas de archivos de video digital, basado en la utilización de metadatos. Se recomienda:

- Adoptar en el Polo de Video y Sonido Digital de la Facultad 9, el estándar de metadatos MPEG-7 para la descripción de los materiales audiovisuales que se manejen en el polo.
- Desarrollar el sistema de recuperación de videos digitales a través de la implementación del modelo propuesto.
- Desarrollar una herramienta que permita la extracción de metadatos y descriptores de bajo nivel de los archivos audiovisuales.

Referencias Bibliográficas.

Abreu Bartomeo, Yanedi, y otros. 2008. Teleformación. [En línea] Universidad de las Ciencias Informáticas, 2008. [Citado el: 28 de 3 de 2009.] <http://teleformacion.uci.cu/mod/resource/view.php?id=14077>.

Acosta Pintado, Eliana. 2008. Lenguaje de programación para web: ASP. [En línea] 3 de 2008. [Citado el: 25 de 4 de 2009.] <http://www.scribd.com/doc/2413026/Lenguaje-de-programacion-para-Web-ASP>.

Amengual Galdón, Sebastian. 2006. Motores de Búsqueda para contenidos audiovisuales. [En línea] 10 de 2006. [Citado el: 15 de 01 de 2009.] http://www.gti.ssr.upm.es/~eeg/docs/comunicacionesI+D_06.pdf.

Benavides, A y Silva, L.C. 2001. El enfoque bayesiano. [En línea] 26 de 2 de 2001. [Citado el: 8 de 1 de 2009.] <http://lcsilva.sbhac.net/EI%20enfocoque%20bayesiano.pdf>.

Canós, José H, Letelier, Patricio y Penadés, María del Carmen. 2003. Metodologías Ágiles en el Desarrollo de Software. [En línea] Universidad Politécnica de Valencia, 12 de 11 de 2003. [Citado el: 26 de 4 de 2009.] <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf#page=9>.

Casares, Javier. 2003. Manual de Posicionamiento. [En línea] 2003. [Citado el: 2 de 12 de 2008.] <http://www.manualdeposicionamiento.com/guia-de-referencia-seo/seo-para-videos/>. ISBN 13: 978-84-611-8192-6.

Castellano Báez, Maidileydys, Cervantes Rodon, Damian y Espinosa Ronquillo, Dayris. 2007. Teleformación. [En línea] Universidad de las Ciencias Informáticas, 2007. [Citado el: 27 de 3 de 2009.] <http://teleformacion.uci.cu/mod/resource/view.php?id=11402>.

Cataldi, Zulma. 2000. Metodología de diseño, desarrollo y evaluación de software educativo. [En línea] 2000. [Citado el: 26 de 3 de 2009.] <http://laboratorios.fi.uba.ar/lsi/cataldi-tesisdemagistereninformatica.pdf>. ISBN 960-34-0204-2.

Clements, Paul. 2003. Published Software Architecture Definitions. [En línea] 2003. [Citado el: 27 de 3 de 2009.] http://www.sei.cmu.edu/architecture/published_definitions.html.

Cuesta, Carlos E. 2007. Ingeniería del software. [En línea] Universidad Rey Juan Carlos., 2007. [Citado el: 30 de 3 de 2009.] <http://kybele.escet.urjc.es/documentos/ISI/Arquitecturas%20de%20SW.pdf>.

- De Jong, Annemieke. 2003.** Los metadatos en el entorno de la producción audiovisual. Una introducción. [En línea] Federación Internacional de Archivos de Televisión, 2003. [Citado el: 26 de 1 de 2009.] http://archivesatrisk.org/restricted/standards/Metadata_version_Es_2003.pdf.
- Delcor Ballesteros, Jordi y Pérez Noriega, Verónica. 2006..** DESCRIPCIÓN, INDEXACIÓN, BÚSQUEDA Y ADQUISICIÓN DE SECUENCIAS DE VÍDEO MEDIANTE DESCRIPTORES MPEG-7. [En línea] 2006. [Citado el: 24 de 2 de 2009.] <http://upcommons.upc.edu/pfc/bitstream/2099.1/3855/1/54955-1.pdf>.
- Díaz Antón, María Gabriela, Pérez, María Angélica y C Grimmán, Anna. 2003.** Propuesta de una Metodología de desarrollo de software bajo un enfoque de calidad sistémica. [En línea] Universidad Simón Bolívar (USB), 2003. [Citado el: 27 de 3 de 2009.] http://165.98.8.15/~oneyda/utilidades/met_soft/metod_USB.pdf.
- Echemendía Tourt, Fernando y Rivera Suárez, Yoel. 2008.** Sistema para la Detección y Extracción de Textos en Videos Digitales. [En línea] 6 de 2008. [Citado el: 8 de 1 de 2009.] <http://biblioteca.uci.cu>.
- Estivill, Assumpció, y otros. 2005.** Uso de metadatos Dublin Core en la descripción y recuperación de artículos de revistas digitales. [En línea] 2005. [Citado el: 17 de 01 de 2009.] <http://temaria.net/estivill2005es.pdf>.
- Fernández Lunas, Juan Manuel. 2001.** Modelos de recuperacion de informacion basados en redes de creencias. [En línea] 2001. [Citado el: 18 de 1 de 2009.] <http://dialnet.unirioja.es/servlet/tesis?codigo=12095>.
- García Albacete, Antonio. 2002.** Gestión Completa de Vídeo. [En línea] 11 de 2002. [Citado el: 10 de 1 de 2009.] <http://www.ntdigitales.com/docs/VIDEOMA.pdf>.
- García de León, Alicia y Garrido Días, Adriana. 2001.** Recuperacion de información en Internet: Google, un buscador singular. [En línea] 12 de 2001. [Citado el: 8 de 12 de 2008.] <http://www.cinfo.cu/Userfiles/file/Cinfo/cinfo2001/v32n3a2001/recuperacion.htm>.
- Garla, David y Shaw, Mary. 1994.** Facultad de ingeniería. Universidad de Buenos Aires. [En línea] 1 de 1994. [Citado el: 28 de 3 de 2009.] <http://materias.fi.uba.ar/7510/review0000.pdf>.

- Herranz Arribas, Luis. 2003.** Colegio Oficial Asociación Española. Ingeniero en Telecomunicación. [En línea] 7 de 2003. [Citado el: 24 de 3 de 2009.] http://www.coit.es/pub/ficheros/p067_resumen_nokia_0567680c.pdf.
- Hrenández Peñalver, Gregorio. 2008.** Departamento de Matematicas Aplicadas. [En línea] Universidad Politécnica de Madrid., 2008. [Citado el: 18 de 4 de 2009.] <http://www.dma.fi.upm.es/gregorio/JavaGC/Cconvexo/teoriaJava.html>.
- INFLANET. 1998.** International Federation of Library Association and Institution . [En línea] 21 de 8 de 1998. [Citado el: 3 de 12 de 2008.] <http://www.ifla.org/IV/ifla64/007-126s.htm>.
- ISO. 2004.** International Standards for Business, Government and Society. [En línea] 2004. [Citado el: 2 de 2 de 2009.] http://www.iso.org/iso/support/faqs/faqs_widely_used_standards/widely_used_standards_other/date_and_time_format.htm.
- . **2002.** International Standards for Business, Government and Society. [En línea] 7 de 2002. [Citado el: 2 de 2 de 2009.] <http://www.iso.org/iso/search.htm?qt=639&searchSubmit=Search&sort=rel&type=simple&published=on>.
- . **1999.** International Standards for Business, Government and Society. [En línea] 3 de 1999. [Citado el: 2 de 2 de 2009.] http://www.iso.org/iso/search.htm?qt=3166&published=on&active_tab=standards.
- Letelier, Patricio y Penadés, María del Carmen. 2006.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Ciencia y Técnica Administrativa*. [En línea] 06 de 2006. [Citado el: 26 de 3 de 2009.] <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. ISSN 1666-1680.
- Linux online. 2008.** Linux online. [En línea] 2008. [Citado el: 11 de 5 de 2009.] <http://www.linux.org/>.
- López Guzmán, Clara. 2000.** Biblioteca Digital Universitaria DGSCA. [En línea] 3 de 2000. [Citado el: 2 de 12 de 2008.] http://www.bibliodgsca.unam.mx/tesis/tes7c1lg/sec_20.htm.
- Martínez, José M. 2004.** MPEG-7 Overview. [En línea] 10 de 2004. [Citado el: 24 de 2 de 2009.] <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>.
- Menéndez, Rosa. 2000.** Rational Software Corporation. [En línea] 2000. [Citado el: 29 de 4 de 2009.] <http://www.usmp.edu.pe/publicaciones/boletin/fia/info36/proyectos.html>.

- Montaño, César, Cortazzo, Joseline y Seroubian, Mabel. 1999.** Centro Regional de Nuevas Tecnologías de Información. [En línea] 1999. [Citado el: 12 de 12 de 2008.] <http://www.crnti.edu.uy/05trabajos/normas1/bus.ppt>.
- Morales Sánchez, Julieta. 2005.** El formato Dublin Core como sistema de catalogación electrónico. [En línea] 06 de 12 de 2005. [Citado el: 16 de 01 de 2009.] http://www.mati.unam.mx/index.php?option=com_content&task=view&id=105&Itemid=51..
- Navarrete, Toni y Blat, Josep. 2003.** Biblioteca de la Universidad Complutense. [En línea] 2003. [Citado el: 13 de 12 de 2008.] <http://europa.sim.ucm.es/compludoc/AA?articuloid=235898&donde=castellano&zfr=0.13866710>.
- NetBeans. 2007.** NetBeans. [En línea] 2007. [Citado el: 27 de 4 de 2009.] <http://www.netbeans.org/>.
- Olivé Peig, Enric. 2003.** [En línea] 11 de 2003. [Citado el: 16 de 02 de 2009.] http://www.tdr.cesca.es/TESIS_UPF/AVAILABLE/TDX-0316104-132946//tepo1de1.pdf.
- Orozco García, Alejandra. 2008.** La construcción de metadatos en el proceso de organización, análisis documental y recuperación de la información en los archivos de imágenes en movimiento. [En línea] 26 de 11 de 2008. [Citado el: 16 de 01 de 2009.] <http://www.patrimoniofilmico.org.co/docs/metadatos.pdf>.
- Paulus Velázquez, Cristian. 2005.** Universidad de Chile DCC. [En línea] 7 de 2005. [Citado el: 2 de 12 de 2008.] <http://www.dcc.uchile.cl/~cvasquez/introehistoria.pdf>.
- Pecos, Daniel. 2002.** PostGreSQL vs. MySQL. [En línea] 7 de 6 de 2002. [Citado el: 25 de 4 de 2009.] http://www.netpecos.org/docs/mysql_postgres/x57.html.
- Perdomo, Roberto. 2006.** Python, un lenguaje de programación Ágil. [En línea] 2006. [Citado el: 26 de 4 de 2009.] <http://maracay.velug.org.ve/descargas/PonenciaPython.pdf>.
- PHP. 2007.** PHP. [En línea] 9 de 2007. [Citado el: 7 de 5 de 2009.] <http://www.php.net/manual/es/intro-whatcando.php>.
- PostgreSQL. 2006.** PostgreSQL. [En línea] 2006. <http://www.postgresql.org/>.
- Prado, Bibiana, Taboada, Roberto y Copca, Agustín. 2004.** Oracle. [En línea] 2004. [Citado el: 26 de 4 de 2009.] <http://www.uaem.mx/posgrado/mcruz/cursos/miic/oracle3.ppt>.

Prieto, Félix. 2008. Programación III.I.T.I. de Sistemas Patrones de diseño. [En línea] 2008. [Citado el: 26 de 4 de 2009.] http://www.infor.uva.es/~felix/datos/priii/tr_patrones-2x4.pdf.

Sánchez Calas, Juan Carlos. 2002. Uportal. *Que son los metadatos. Bibliotecario Documentalista REUNA.* [En línea] 2002. [Citado el: 01 de 12 de 2009.] http://www.uportal.cl/siel/siel_docs/estandarizacion/metadatos_SIEL.pdf.

Siri, Laura. 2000. Internet: Búsqueda y buscadores. [En línea] 2000. [Citado el: 7 de 12 de 2008.] <http://books.google.com/books?hl=es&lr=&id=GT4Ex85fNxEC&oi=fnd&pg=PA21&dq=buscador&ots=84stqTGRiA&sig=yGOtK4V1WIVOykumRHsWmx6Znlq#PPP1,M1..> ISBN 9879334795, 9789879334799.

Suárez González, Héctor. 2003. Manual Hibernate. [En línea] 21 de 3 de 2003. [Citado el: 14 de 5 de 2009.] http://www.javahispano.org/contenidos/es/manual_hibernate/.

Symfony. 2005. Open-Source PHP Web Framework. [En línea] 2005. [Citado el: 6 de 5 de 2009.] <http://www.symfony-project.org/>.

Vetro, Anthony. 2008. MPEG-7 Video Browsing and Summarization. [En línea] 13 de 6 de 2008. [Citado el: 16 de 1 de 2009.] <http://www.merl.com/projects/video-browsing/>.

Vidal Bordés, Francisco Javier. 2001. Solapamiento en herramientas de búsqueda de información en la World Wide Web. [En línea] 6 de 2001. [Citado el: 12 de 12 de 2008.] <http://ibersid.eu/ojs/index.php/scire/article/view/1143/1125>.

Vilches, Lorenzo. 2001. Tecnologías digitales al servicio de los archivos de imágenes. [En línea] 2001. [Citado el: 15 de 12 de 2008.] http://www.archivo-semiotica.com.ar/TECNOLOGIAS_IMAGEN_.html.

Visual paradigm. 2002. Visual paradigm. [En línea] 2002. [Citado el: 29 de 4 de 2009.] <http://www.visual-paradigm.com/>.

Anexos

Anexo # 1 Preguntas fundamentales del cuestionario de la entrevista realizada a los líderes de polo y proyectos, así como al arquitecto del polo.

¿Cuál es la misión del Polo de Video y Sonido Digital?

¿Cuál es la visión del Polo de Video y Sonido Digital?

¿Qué sistema o algoritmo utilizan para la recuperación de materiales audiovisuales en el polo de Video y Sonido Digital? ¿Cómo realizan la recuperación de materiales audiovisuales en el polo?

¿Conocen algún estándar de metadatos para la descripción de materiales audiovisuales?

¿Utilizan algún estándar de metadatos para la descripción de los materiales audiovisuales que maneja el polo? ¿Cuál o cuáles?

Anexo # 2

Ejemplo de un documento XML de la descripción de una escena:

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video id="1">
        <Media Time>
          <Media TimePoint>T00:00:00:0F30000</Media TimePoint>
          <MediaDuration>PT16M33S11772N30000F</MediaDuration>
        </Media Time>
        <TemporalDecomposition gap="false" overlap="false">
          <VideoSegment id="shot1_1">
            <Media Time>
              <Media TimePoint>T00:00:00:0F30000</Media TimePoint>
              <MediaDuration>PT3S22112N30000F</MediaDuration>
            </Media Time>
            <TextAnnotation confidence="0.500000">
              <Free TextAnnotation> Tiroteo en una calle de Chicago. Tres
                personas situadas detrás de dos coches disparan a otra
                resguardada en una ventana </Free TextAnnotation>
            </TextAnnotation>
          </VideoSegment>
          <VideoSegment id="shot1_2">
            <Media Time>
              <Media TimePoint>T00:00:03:22112F30000</Media TimePoint>
              <MediaDuration>PT9S18288N30000F</MediaDuration>
            </Media Time>
            <TextAnnotation confidence="0.500000">
              <Free TextAnnotation>
                Llega la funeraria y se forra
              </Free TextAnnotation>
            </TextAnnotation>
          </VideoSegment>
        </TemporalDecomposition>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

Glosario de Términos.

XML: es un lenguaje abierto que se ha diseñado y optimizado para mejorar la funcionalidad de la Web.

UML: es un lenguaje que permite especificar, documentar, visualizar y construir los artefactos de un sistema que involucra gran cantidad de software.

Resource Description Framework: desarrollado bajo los auspicios de W3C, es una infraestructura que posibilita la codificación, el intercambio, la reutilización de los metadatos estructurados.

HTML: (HyperText Markup Language). Es el lenguaje estándar para describir el contenido y la apariencia de las páginas en la Web.

JPG: formato de compresión de imágenes, utilizado para crear logos, banners... Reduce el peso de la imagen, acelerando el tiempo de descarga

GIF: (*Graphic Interchange Format*). Un formato de imagen, actualmente muy difundido en la Web. Permite reducir el tamaño de la imagen mediante una elevada tasa de compresión.

PDF: (*Portable Document Format*). Un formato de Adobe que permite distribuir un documento en diferentes sistemas conservando su presentación (.pdf).

URL: (*Uniform Resource Locator*). La dirección Internet de un recurso Web (página, elementos incorporados, etc.) entendido e interpretado por los navegadores.

DGN: fichero de dibujo y diseño.

GeoTiff: estándar de metadatos de dominio público que permite que información georreferenciada sea encajada en un archivo de imagen de formato TIFF.

WAV: (*Waveform Audio Format*) formato de audio digital.

MP3: formato de audio digital comprimido.

IDE: Entorno de Desarrollo Integrado.

PNG: (*Portable Network Graphics*). Gráficos Portables de Red. Formato gráfico muy completo especialmente pensado para redes.