

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 9



TÍTULO: Subsistema de Audio y Subsistema de Video del Producto Captura y Catalogación de Medias (CCM). Rol Implementador.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN INFORMÁTICA

AUTOR: José Enrique Tejera Quevedo

TUTOR: Yanio Hernandez Heredia

Ciudad de la Habana, <mes, día de 2010>.

“Año 52 de la Revolución”

AGRADECIMIENTOS

Agradezco a mi madre por el amor, el esmero y la educación, este es tu regalo.

A mi padre por enseñarme el camino a seguir y dejarme recorrerlo solo.

A mi abuelita Rosa, gracias por el cariño y la confianza.

A la Revolución por darme la oportunidad de superarme y llegar a ser un ingeniero.

A la UCI por acogerme y darme las herramientas.

A mi tutor Yanio, a Zori y a Jean por la guía y la ayuda.

A mis compañeros de aula, de apartamento y de proyecto por compartir la vida, los momentos, las alegrías, las tristezas, los problemas, los buenos y los malos momentos.

Agradezco a los que me apoyaron y brindaron la mano en los momentos que los necesitaba, a los que están y a los que se fueron, ustedes son mi segunda familia. Al mello, a Abe, a Jorgito, a Pepe, a Liannet, a Mailen, a todos muchas gracias.

Muchas gracias Adny, sin ti no se qué sería, siempre estás ahí en todo momento, eres la mejor.

Agradezco enormemente a mi tribunal de tesis, me han enseñado mucho en estos pocos meses, muchas gracias.

DATOS DE CONTACTO

Tutor: Ing. Yanio Hernandez Heredia

E-mail: yhernandezh@uci.cu

- Ingeniero en Ciencias Informáticas, Universidad de Ciencias Informáticas, 2007.
- Diploma de Oro.
- Premio Mella.
- Premio Hombre de Futuro.
- Premio del Rector en la Categoría de Profesor Universitario en Adiestramiento más Destacado.
- Vinculado al Proyecto Productivo UCITeVe durante los 5 años de estudios universitarios.
- Líder de Proyecto del Polo Video y Sonido Digital.
- Profesor del Departamento de Técnicas de Programación, Facultad 9.

RESUMEN

En la actualidad muchas Instituciones Nacionales e Internacionales, Jurídicas, Parlamentarias, Universitarias, Gubernamentales, entre otras, generan o necesitan una gran cantidad de información audiovisual y el análisis de esta es primordial en muchos casos para la toma de decisiones en el desarrollo de proyectos y actividades de carácter social. Las Tecnologías de la Informática y las Comunicaciones (TIC) constituyen una herramienta irremplazable de gestión de la información, por tanto su uso es muy útil en las entidades mencionadas anteriormente.

En el presente trabajo se propone el desarrollo de un subsistema que dada una planificación realizada con anterioridad sea capaz de capturar medias desde dispositivos de entrada de video, transcodificar, procesar y transferir de forma automática estas grabaciones hacia los servidores de media para posteriores análisis y el desarrollo de otro subsistema que permita grabar el audio de reuniones, encuentros, vistas judiciales, conferencias y demás actividades de repercusión y permita transcribir esta información para una mejor portabilidad puesto que la Universidad de las Ciencias Informáticas (UCI) no cuenta con un producto informático que tenga la posibilidad de brindar estas funcionalidades.

Este documento recoge los resultados de la investigación realizada en el estudio y de las técnicas de programación. Se describen las funcionalidades de las aplicaciones y como se realizan los procesos antes mencionados. Se muestra la organización de los subsistemas a través del modelo de implementación y de los Diagramas de Componentes. Se prueban las principales funcionalidades implementadas con el objetivo de lograr un software robusto.

Los Subsistema de Audio y Video del Producto Captura y Catalogación de Medias (CCM) hacen uso de las actuales tecnologías libres para procesamiento de imágenes y captura de medias.

PALABRAS CLAVES

Audiovisual, Planificación, Capturar, Medias, Video, Transcodificar, Transferir, Audio, Transcribir



ÍNDICE

INTRODUCCIÓN	3
CAPÍTULO 1: Tendencias Actuales	7
1.1 Introducción	7
1.2 Procesos relacionados con la captura y procesamiento de las medias.	7
1.2.1 Captura de Audio	7
1.2.2 Transcripción.....	8
1.2.3 Captura de Video	8
1.2.3.1 Captura Automática	9
1.2.3.2 Captura Manual.....	9
1.2.4 Indexación	10
1.2.4.1 Extraer Fotogramas	10
1.2.4.2 Fragmentar Videos	11
1.3 Técnicas de programación existentes a nivel nacional e internacional.....	11
1.3.1 Paradigmas de Programación.....	11
1.3.2 Lenguajes de Programación.....	13
1.4 Tendencias y metodologías relacionadas con las técnicas de programación.	16
1.4.1 Programación en Plataformas Libres	16
1.5 Conclusiones.....	18
CAPÍTULO 2: Descripción de la solución propuesta	19
2.1 Introducción	19
2.2 Modelo de Implementación	19
2.2.1 Diagrama de Componentes	20
2.2.1.1 Diagrama de Componentes del Módulo de Audio	21
2.2.1.2 Diagramas de Componentes del Módulo de Video.....	22
• Captura Automática.....	22
• Captura Manual.....	23
• Extracción de Fotogramas	24
• Fragmentar Videos	25
2.3 Estándar de Codificación	25
2.4 Conclusiones.....	30
CAPÍTULO 3: Validación de la Solución Propuesta	31
3.1. Introducción	31
3.2. Pruebas de Software.....	31



3.2.1. Prueba de Caja Blanca.....	31
3.2.2. Prueba de Caja Negra.....	35
3.3. Pruebas de Software a los Subsistemas.....	38
3.3.1. Prueba del Subsistema de Video	38
3.2.1. Prueba del Subsistema de Audio	40
3.3. Conclusiones.....	42
CONCLUSIONES GENERALES.....	43
RECOMENDACIONES.....	44
TRABAJOS CITADOS	45
ANEXOS	47
Anexo 1: Caso de Uso del Módulo Video - Gestionar Video.....	47
Anexo 2: Caso de Uso del Módulo Video - Realizar Captura de Video	50
Anexo 3: Caso de Uso del Módulo Video - Digitalizar	51
Anexo 4: Caso de Uso del Módulo Video - Realizar transcodificación	52
Anexo 5: Caso de Uso del Módulo Video - Registrar datos	54
Anexo 6: Caso de Uso del Módulo Video - Transferir Material	55
Anexo 7: Caso de Uso del Módulo Video - Extraer fotogramas.....	56
Anexo 8: Caso de Uso del Módulo Video - Generar Índices.....	57
Anexo 9: Caso de Uso del Módulo Video - Autenticar Usuario.....	58
Anexo 10: Caso de Uso del Módulo Audio - Autenticar Usuario	59
Anexo 11: Caso de Uso del Módulo Audio - Gestionar Archivo	60
Anexo 12: Caso de Uso del Módulo Audio - Grabar Audio	65
Anexo 13: Caso de Uso del Módulo Audio - Grabación Automática	68
Anexo 14: Caso de Uso del Módulo Audio - Grabación Manual.....	71
Anexo 15: Caso de Uso del Módulo Audio - Reproducir Audio	74
Anexo 16: Caso de Uso del Módulo Audio - Transcribir Audio	75



INTRODUCCIÓN

La Real Academia de la Lengua Española define el vocablo medio como: “*cosa que puede servir para un determinado fin*” y a audiovisual como: “*adjetivo que se refiere conjuntamente al oído y a la vista, o los emplea a la vez. Se dice especialmente de métodos didácticos que se valen de grabaciones acústicas acompañadas de imágenes ópticas.*”

La realidad en un audiovisual se puede percibir con los cinco sentidos que poseen los seres humanos. Específicamente la vista y el oído que perciben en un tiempo y un espacio y son los protagonistas en la comunicación e interpretación de la realidad.

La vista comienza a jugar su papel a través de la percepción visual: lo que se ve es la luz a través del ojo, que solo percibe una parte del espectro electromagnético. La luz llega hasta la retina, que da una imagen invertida. Esta imagen se transmite al cerebro donde se guarda. Este proceso es instantáneo. En la retina se da un fenómeno conocido como persistencia retiniana, que permite crear la sensación de movimiento aunque realmente lo percibido son una sucesión de imágenes fijas a una determinada velocidad. **(1)**

Por su parte el sonido es una vibración que se transmite por un medio físico que llega al oído donde se encuentran unas estructuras (huesecillos) que son capaces de al vibrar, generar las ondas eléctricas que luego se transmiten al cerebro. **(2)**

No se podría hacer referencia a los medios audiovisuales sin hablar de la Radio, su origen se remonta al surgimiento de la corriente eléctrica, el telégrafo y la telefonía. Luego del descubrimiento y la medición de las ondas electromagnéticas por *Heinrich Hertz* fue posible la creación del primer receptor de radio. La llegada de la telegrafía sin hilos, descubrimiento de *Guillermo Marconi*, permitió que las señales sonoras pudieran propagarse a algo menos de 20 Kilómetros de distancia todo un logro en aquella época. Por último A partir del siglo XX cuando *A. Fleming* y *R.A Fessenden* lograron la transmisión de la voz humana, se inició el desarrollo de la radio que hoy conocemos. **(3)**

Por su parte el cine: “Técnica que consiste en registrar imágenes fotográficas sucesivas en una cinta de celuloide, que producen la impresión de movimiento al proyectarse con cierta rapidez” **(4)**, combina imágenes realistas, movimiento y sonido, alcanzando un alto grado de similitud con la realidad por la percepción de formas, colores y movimientos que brinda. Sus primeros pasos fueron a través del cine silente o mudo, donde se proyectaba la película sin audio, primando las expresiones de los actores para un mejor entendimiento de la trama, proceso que iniciaron los hermanos *Lumiere* los cuales crearon el cinematógrafo, que servía lo mismo de cámara como de proyector. La revolución del cine sonoro ocurrió el 23 de octubre de 1927, a partir de ese momento surge el cine



musical, se le da voz y música a las películas y los estudios tienen que realizar grandes inversiones, pues la primera representación de "*The Jazz Singer*", filme sonoro, hablado y cantado, de *Alan Crossland*, marcó la entrada del cine en una nueva edad de su historia. **(5)**

Por su parte la Televisión fue otro medio que revolucionó el mundo y hasta llegó a provocar cambios significativos en las transmisiones y los programas de radio. Las primeras pruebas de transmisión de imágenes son de mediados del siglo XIX, si bien entonces no se hablaba de televisión. La palabra francesa "*télévision*" la utilizó por primera vez el físico ruso *Constantin Perskyi* en una conferencia sobre este tema impartida en el primer Congreso de Electricidad, que se celebró en la Feria Internacional de París de 1900, y pronto fue trasladada a otros idiomas con expresiones muy similares. **(6)** Actualmente la televisión continúa en constante perfeccionamiento, desarrollo e incorporando nuevos paradigmas capaces de cambiar la concepción que se tiene sobre este medio, como la idea de que los usuarios no sean meros espectadores, sino de convertirlo en un medio interactivo.

El video, según la Real Academia de la Lengua Española "Sistema de grabación y reproducción de imágenes, acompañadas o no de sonidos, mediante cinta magnética", se encuentra ligado de forma íntima a la Televisión, pues surge con el propósito de facilitar el trabajo de esta, ya que con la grabación de los programas televisivos se podía planificar una programación con diferentes horarios, permitió el almacenaje de los programas y por tanto su retransmisión, evitando de esta forma que las transmisiones fueran en vivo.

En la actualidad, el desarrollo vertiginoso que se ha alcanzado en la esfera de la Informática y las Comunicaciones, ha permitido un notable avance en el mundo con respecto a los medios audiovisuales, su transmisión, su almacenamiento y las nuevas formas en las que se pueden representar estas. Con la aparición de los formatos digitales, contenedores de cualquier tipo de información, se ha podido capturar y digitalizar el audio y el video desde diferentes fuentes, antenas de radio o televisión, de fuentes analógicas¹ en soportes magnéticos propensos al deterioro por uso, o simplemente crearlo de manera digital en la misma computadora.

La mayoría de estas acciones se ejecutan de forma manual, lo que implica posibles atrasos o errores en dependencia de los fallos que pueda cometer la persona encargada de su ejecución. Con un sistema informático es posible la captura de estas medias de manera automática, tomando en cuenta una planificación previa y luego de ser digitalizadas, pueden ser objeto de varios procesos de acuerdo al tipo o a las necesidades de los usuarios y almacenadas en servidores para su posterior análisis, disminuyendo la ocurrencia de los fallos y logrando de esta manera salvar para las posteriores generaciones el patrimonio cultural, político y nacional que representan los medios

¹ Representación de la información mediante una señal que varía de tamaño o intensidad constantemente



audiovisuales. La implementación de este sistema daría como resultado un software altamente cotizado en el mercado nacional o mundial, pues las soluciones informáticas capaces de automatizar estas acciones han sido desarrolladas en plataformas privadas, lo cual implica un alto costo de adquisición. Además, en su mayoría, estos programas no cuentan con la incorporación de módulos que permitan la planificación de los horarios.

A partir de la situación planteada se ha detectado como **problema a resolver** la inexistencia de los subsistemas que permitan capturar e indexar ficheros de videos de forma automática y obtener y transcribir ficheros de audio en el Producto Captura y Catalogación de Medias (CCM). Para darle solución a este problema se pretende la Implementación de los procesos de captura e indexación de video, así como de captura y transcripción de audio en la obtención del producto para la Captura y Catalogación de Medias y se identificaron como **objeto de estudio** los procesos de captura de medias y las técnicas de indexación de video y de transcripción de audio.

Dándole cumplimiento a lo antes expuesto se propone como **objetivo general**: Implementar los Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias en el Departamento de Señales Digitales, pertenecientes al Centro de Desarrollo de Geoinformática y Señales Digitales. Defendiendo la idea de que si se logra una correcta caracterización de los procesos que intervienen en la captura de audio y video de sistemas de catalogación de medias, se establece un estándar de codificación y se implementan los Subsistemas de Audio y Video se podrá capturar e indexar ficheros de video de forma manual o automática y obtener y transcribir ficheros de audio en el Producto Captura y Catalogación de Medias

Para lograr el objetivo general de la investigación se tendrán en cuenta los siguientes **objetivos específicos**:

- Caracterizar los procesos relacionados con los Subsistemas de Captura y Transcripción de Audio y Captura e Indexación de Video.
- Identificar las distintas técnicas de programación existentes a nivel nacional e internacional.
- Valorar las tendencias y metodologías relacionadas con las técnicas de programación actuales, así como de las plataformas de desarrollo que la soportan.
- Definir un estándar de codificación.
- Implementar el 50% de los casos de uso críticos de los subsistemas.
- Desarrollar casos de prueba que certifiquen el correcto funcionamiento de los algoritmos implementados.

El desarrollo de la investigación estará guiado por métodos científicos, entre los que se encuentran los empíricos que permiten extraer de los fenómenos analizados las informaciones que se necesitan



sobre ellos a través de observaciones, del uso de técnicas opináticas y la propia experimentación y los métodos teóricos donde están comprendidos toda una serie de procedimientos que posibilitan la asimilación teórica de la realidad y que se adecuan a las condiciones en que se va a desarrollar la investigación. **(7)**

Análisis histórico - lógico: Permitirá hacer un estudio de las diferentes técnicas de programación a nivel nacional e internacional, analizando desde sus orígenes, sus diversas utilidades y su evolución, ya que este método está vinculado al conocimiento de las distintas etapas de los objetos en su sucesión cronológica **(7)**

Modelación: Permitirá realizar una modelación para una mejor comprensión de lo que se desea implementar, pues la modelación es justamente el método mediante el cual se crean abstracciones con vistas a explicar la realidad. **(7)**

Análisis y síntesis: Permitirá concretar los elementos más importantes relacionados con el proceso de captura de medias, la transcripción de audio y la indexación de videos, puesto que el análisis permite la división mental del todo en sus múltiples relaciones y componentes y la síntesis establece mentalmente la unión entre las partes previamente analizadas y posibilita descubrir las relaciones esenciales y características generales entre ellas. La síntesis se produce sobre la base de los resultados obtenidos previamente en el análisis. **(7)**



CAPÍTULO 1: Tendencias Actuales

1.1 Introducción

En este capítulo se hace una breve caracterización de los procesos relacionados con la captura y procesamiento de las medias en el producto Captura y Catalogación de Medias y de las técnicas actuales de programación a nivel nacional e internacional haciendo énfasis en las utilizadas para la confección de la aplicación que da origen al desarrollo de este trabajo.

1.2 Procesos relacionados con la captura y procesamiento de las medias.

La captura de medias y su posterior procesamiento, está definida por una serie de procesos, estos se aplicarán en dependencia del tipo de archivo que se disponga a obtener, en específico audio o video. Las fuentes de origen de estos archivos pueden ser señales de antenas, almacenamiento externo o dispositivos analógicos en el caso del video y en el caso del audio se captura la señal de audio obtenida por la tarjeta de audio de la computadora (line-in).

1.2.1 Captura de Audio

En el Subsistema de Audio, encargado de los procesos de grabación y transcripción de audio, se utiliza para la grabación la **API**² (Interfaz de Programación de Aplicaciones), *JavaSound*. Esta *API* provee clases y funcionalidades que se encargan de procesar el audio proveniente desde diferentes entradas de sonido de la computadora, utilizando en este caso la línea de entrada.

Con *JavaSound* es posible manipular el flujo de bytes con información de sonido, que se obtiene a partir de la tarjeta de audio o del componente integrado de la *motherboard* destinada a esta función, siendo posible así construir un fichero de audio a partir de esta información. Lo antes descrito se logra utilizando clases como: *DataLine*, *TargetDataLine*, *SourceDataLine*, *AudioInputStream*, *ByteArrayOutputStream*, con el apoyo de otras nativas de Java: *File* y sus derivadas *FileInputStream* y *FileOutputStream*.

Una vez que se obtiene el fichero de audio, es posible codificar la información que contiene el mismo hacia formato WAV (*formato de elevada calidad, usado en las grabaciones de CDs, que trabaja a 44 Khz y a 16 bits, su principal inconveniente es el elevado peso de los archivos*) (8).

La codificación a WAV, se hace con el objetivo de lograr una entrada válida para el codificador LAME, que es utilizado para obtener ficheros en formato MP3, que logra una mejor compresión de la información y es el establecido en las reglas del negocio para el Sistema Captura y Catalogación de Medias.

² Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software



CAPÍTULO 1: Tendencias Actuales

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

El LAME, (acrónimo recursivo de *LAME Ain't an MP3 Encoder*) es un codificador de *MPEG Audio Layer III* (MP3). Destaca por su rapidez, por la posibilidad de elegir la tasa de bits CBR (tasa de bits constante) y dos tipos de *bitrate* variable, VBR y ABR y el modo *joint stereo*. Este códec tiene una licencia de código abierto y funciona por intérprete de órdenes o bien integrado en otras aplicaciones. **(9)** Es utilizado en la aplicación con el auxilio de la librería Tritonus, el cual nos abstrae de cargar el proceso del LAME con la clase Process de Java y de la introducción de líneas de comando a través del código.

El proceso de grabación se inicia con la autenticación del usuario. Este puede realizar la captura atendiendo una planificación de grabación de audio, elaborada con anterioridad o seleccionar la opción de grabación, en la que primeramente deberá introducir datos necesarios para identificar la grabación, como: tipo de actividad que se desea grabar, hora, día y local donde se encuentra. En ambos casos puede escoger si desea realizar la captura automáticamente, por lo tanto la aplicación comenzará a grabar a la hora especificada, o si desea que se realice de forma manual, en este caso el propio usuario tendrá que iniciar y culminar la grabación.

1.2.2 Transcripción

La transcripción de audio consiste en la escritura del contenido de una grabación de audio, con el objetivo de hacer más portable la información. Este proceso es realizado por operadores de transcripción que mientras escuchan, teclean la información auditiva. El Subsistema de Captura y transcripción de audio provee un editor de texto integrado a un reproductor de audio, con las funcionalidades básicas de reproducción: pausa, reproducción, adelantar y retrasar la reproducción, pues en el proceso de escritura se necesita de las mismas.

Para la reproducción durante la transcripción de audio se hace uso de la librería JVLIC, interfaz para Java de la aplicación *VLC (reproductor multimedia portátil que soporta formatos de audio y vídeo como H.264, Ogg, DivX, MKV, TS, MPEG-2, MP3, MPEG-4, AAC, medios físicos como DVDs, VCD, Audio-CD, streaming. También puede convertir, codificar y actuar como un servidor de streaming en unicast o multicast e IPv4 o IPv6. No necesita ningún códec externo, programa o paquete de códec para trabajar.)* (10), y que posibilita realizar las acciones de reproducción sobre un fichero de audio o video. Entre el conjunto de librerías proporcionadas por JVLIC se utilizan con este fin *MediaPlayer* y *MediaDescriptor*.

1.2.3 Captura de Video

La captura de video tiene como origen varias fuentes en específico y se utiliza hardware especializado, tarjetas de captura de video (dispositivos que contienen los circuitos necesarios



CAPÍTULO 1: Tendencias Actuales

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

para crear una señal de video digital). La captura puede realizarse de manera automática sin la participación o monitoreo de personas o manual, en la cual el usuario será quien guíe los procesos.

1.2.3.1 Captura Automática

La captura se realiza de forma automática a partir de una planificación previamente confeccionada y utilizando tarjetas sintonizadoras de televisión, que permiten capturar señales directamente desde un canal de TV. La aplicación se encarga de conectarse a la base de datos periódicamente y filtrar todas las planificaciones que se han realizado de acuerdo a varios parámetros, la hora, la fecha, el recurso, el canal, con el objetivo de ejecutar las planificaciones que verdaderamente interesan al encontrarse en tiempo para realizar la captura. Al obtener la lista de las capturas con *Mencoder*, que no es más que un codificador de video liberado bajo *GPL (General Public License)*, que incluye el reproductor de video *Mplayer* para Linux y otras plataformas, se capturan los materiales por los dispositivos sintonizadores de TV, apoyándose especialmente en los principales códec de compresión de video y audio, logrando una buena relación en cuanto a calidad y tamaño de los archivos. Este codificador es ejecutado por líneas de comandos, facilitando su ejecución y la recuperación de varios datos que el envía a través de la consola, con el fin de visualizar resultados en tiempo real, por tanto la aplicación hace uso de la clase *QProcess* incluida en la biblioteca de desarrollo *QT*. Luego de iniciada la captura se comprueba periódicamente la hora y fecha de terminada la captura con la fecha y hora del sistema, con el propósito de que cuando ambas sean iguales se detenga la captura y esta media ya codificada al formato correspondiente sea transferida al servidor de medias para su posterior análisis y sea insertada en el servidor de base de datos. Mientras este proceso ocurre la aplicación continúa realizando pedidos a la base de datos ya que el tratamiento de los procesos esta implementado con hilos, los cuales permiten que estos se ejecuten paralelamente.

1.2.3.2 Captura Manual

Como fuentes analógicas de video que el sistema puede capturar y codificar se encuentran, VHS, cámaras, u otros dispositivos que se conecten a la tarjeta de captura de video y como fuente digital, archivos de video que se encuentren en medios extraíbles como memorias flash, discos de almacenamiento externo, entre otros. La aplicación de captura manual inicia pidiéndole al usuario que se registre en el sistema, con el propósito



CAPÍTULO 1: Tendencias Actuales

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

de validar si puede realizar las operaciones solicitadas, de acuerdo a la validez de sus datos y al rol.

Si se escoge que la fuente de origen es Analógica, el sistema tiene la capacidad de capturar la media de este origen con el uso del *Mencoder* y provee al usuario una vista previa de la captura con el uso del *MPlayer*, que se encuentra embebido en la aplicación. Ambos procesos se ejecutan con el uso de la clase *QProcess* incluida en la biblioteca de desarrollo *QT*.

En el caso que la fuente sea Digital, la aplicación analiza el video que se quiere capturar con la utilización del *MPlayer*, si no se encuentra en el formato requerido, usando el *Mencoder* se recodifica. La aplicación muestra una vista previa y el progreso del proceso mientras se ejecuta a través de una Barra de Progreso.

En ambos casos, al concluir el proceso, el video se transfiere hacia el servidor de medias y se adiciona a la base de datos, en donde se insertan datos específicos como el tamaño del video, el nombre, la duración, que son obtenidos de los análisis previos que se les realiza.

1.2.4 Indexación

La indexación es un término utilizado para hacer referencia a diferentes procesos que se le pueden aplicar a los videos. Algunas de las técnicas de indexación más utilizadas son la extracción de fotogramas representativos (*keyframes*), la detección de cambios de escena, la extracción de objetos específicos y de sonidos clave en audio. **(11)** Estos métodos de indexación posibilitarán extraer fotogramas y fragmentar materiales a partir de la generación de índices por concepto de tiempo asociados a los videos, con el fin de brindar prestaciones que permitan optimizar funciones utilizadas por los usuarios para el tratamiento de medias.

1.2.4.1 Extraer Fotogramas

La extracción de fotogramas es un proceso automático que se encuentra en constante comunicación con la base de datos. Su objetivo es conocer, de las medias que se encuentran capturadas, a cuales no se les ha realizado el proceso de extracción de fotogramas. Al identificar el video que no cumple con este requisito, se procede a conocer su tiempo de duración en segundos con el auxilio del *MPlayer*, proceso que se ejecuta a través de la clase *QProcess* del *QT*. De acuerdo a estas características se procede a calcular la cantidad de fotogramas que se deben extraer, los cuales se van guardando en una carpeta nombrada con el mismo nombre del video y cada fotograma se titula de



CAPÍTULO 1: Tendencias Actuales

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

acuerdo al instante que ocupan en la reproducción del video. La interface de la aplicación mostrará los procesos que se están ejecutando en ese preciso momento o si se encuentra en espera.

1.2.4.2 Fragmentar Videos

La fragmentación de video es ejecutada por un proceso que sincrónicamente realizará pedidos a la base de datos en busca de las medias a las que se les necesita realizar esta operación, obteniendo además los valores del índice (tiempo de inicio y tiempo de fin) necesarios para extraer el fragmento del video. Para realizar esta acción se utiliza el *Mencoder*, al cual se le introduce por código, utilizando la clase *QProcess*, la dirección donde se encuentra el archivo de origen y la dirección a la cual se exportará la nueva media obtenida, el formato de archivo y los intervalos de tiempo que se van a extraer al video. La aplicación mostrará un listado de los procesos que se han ejecutado.

1.3 Técnicas de programación existentes a nivel nacional e internacional

Una técnica de programación es una metodología que debe de seguirse y tomarse en cuenta al momento de programar. Deberá entenderse que para la programación hay que asumir ciertas normas que permitan la estandarización de la programación, implicando una disminución de costos, independencia del programador y seguridad.

1.3.1 Paradigmas de Programación

Un paradigma de programación es una colección de modelos conceptuales que juntos modelan el proceso de diseño y determinan, al final, la estructura de un programa. Esa estructura conceptual de modelos está pensada de forma que determinen la forma correcta de los programas y controlen el modo en que se piensan y formulan las soluciones, y al llegar a la solución, ésta se debe de expresar mediante un lenguaje de programación. Para que este proceso sea efectivo las características del lenguaje deben reflejar adecuadamente los modelos conceptuales de ese paradigma. Cuando un lenguaje refleja bien un paradigma particular, se dice que soporta el paradigma, y en la práctica un lenguaje es difícil distinguirlo del propio paradigma, por lo que se identifica con él. **(12)**

1.3.1.1 Programación Estructurada

El programa tiene un diseño modular, los módulos son diseñados de forma descendente y cada módulo se codifica utilizando las tres estructuras de control básicas: secuencia,



CAPÍTULO 1: Tendencias Actuales

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

selección y repetición. Los programas deberán estar dotados de una estructura, incorporando:

- Recursos abstractos: Esto consiste en descomponer una determinada acción compleja en términos de un número de acciones más simples capaces de ejecutarlas o que constituyan instrucciones de computadoras disponibles.
- Diseños descendentes (*top-down*): Es descomponer el problema en una serie de niveles o pasos sucesivos de refinamiento, efectuando una relación entre las diferentes etapas de estructuración de modo que se relacionasen unas con otras mediante entradas y salidas de información. **(13)**

1.3.1.2 Programación Modular

El programa se divide en módulos (partes independientes), los cuales ejecutan una única actividad o tarea y se codifica independientemente de otros módulos. Estos módulos se analizan, codifican y ponen a punto por separado. Cada programa contiene un módulo llamado programa principal, que controla todo lo que sucede; se transfiere el control a submódulos, de modo que ellos pueden ejecutar sus funciones. **(13)**

1.3.1.3 Programación Orientada a Eventos

La programación orientada a eventos no es programar de forma tradicional, donde se sigue un patrón que controla el propio flujo del programa, el control del flujo de un programa orientado a eventos es extremadamente conducido por eventos externos. En vez de esperar por un comando el cual ordene que se proceda determinada información, el sistema esta pre-programado para realizar un ciclo de forma repetida, para verla y realizar una función que desencadena en su procesamiento. **(13)**

1.3.1.4 Programación Funcional

Se enfatiza en la definición de funciones que implementen estados de la máquina, en contraparte con la programación basada en procedimientos, la cual está basada en la ejecución de comandos de forma secuencial. Un programa puramente funcional no usa la mutación, cuando modifica el valor de un estado para producir valores, construye nuevos valores, sin sobre escribirlos, de los existentes. **(13)** Un programa consiste en la definición de una o más funciones. Para la ejecución de un programa, se dan parámetros a una función y el ordenador tiene que calcular el resultado. **(14)**



1.3.1.5 Programación Orientada a Objetos

Se basa en los conceptos de modularidad, abstracción procedimental, abstracción de datos y ocultación de la información de acuerdo al módulo o tipo de dato objeto. Las prioridades fundamentales de un tipo son: encapsulamiento (combinar datos y operaciones que se puedan ejecutar sobre esos datos en un objeto), herencia (es la propiedad que permite a un objeto transmitir sus propiedades a otros objetos denominados descendientes; la herencia permite la reutilización de objetos que se hayan definido con anterioridad) y polimorfismo (es la propiedad que permite decidir en tiempo de ejecución la función a ejecutar, al contrario que sucede cuando no existe polimorfismo, en el que la función a ejecutar se decide previamente y sin capacidad de modificación, en tiempo de compilación). **(13)**

1.3.2 Lenguajes de Programación

Los lenguajes de programación tienen diversas clasificaciones, como por ejemplo según su nivel de abstracción: lenguaje de bajo nivel (es el código fuente de la máquina, es decir el que la máquina puede interpretar); lenguaje de nivel medio (un término entre el lenguaje de la máquina y el lenguaje natural) y lenguaje de alto nivel (los que están compuestos por elementos del lenguaje natural, es decir el humano, especialmente el inglés). También según la forma de ejecución, encontrándonos así con el compilador (programas que permiten traducir un programa del lenguaje natural al lenguaje de bajo nivel) y lenguajes interpretados (los que sólo hacen la traducción de los datos que se van a utilizar en ese momento y no los guarda para usar posteriormente). **(15)**

A partir del análisis previamente realizado se ha llegado a la conclusión que el paradigma de programación que nos brinda mayores ventajas en la implementación de los subsistemas es el orientado a objetos pues: fomenta la reutilización y extensión del código, permite crear sistemas más complejos, relacionar el sistema al mundo real, facilita la creación de programas visuales, agiliza el desarrollo de software, facilita el trabajo en equipo y el mantenimiento del software. Dentro de esta técnica se encuentran varios lenguajes de programación los cuales se caracterizarán a continuación.

1.3.2.1 Lenguaje C

El lenguaje C nació en la década del 70. Es de propósito general y fue creado para desarrollar aplicaciones en el entorno *UNIX*. El lenguaje C está basado en otro más antiguo llamado, originalmente B, pero a diferencia de este, el C introducía el concepto de tipos de datos y creaba código ejecutable, mientras que el B creaba código interpretable. El



CAPÍTULO 1: Tendencias Actuales

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

lenguaje C fue tan importante en la evolución del software que hasta el día de hoy se continúa utilizando. La gran mayoría de los sistemas operativos, como *Linux* o *Windows*, están programados en C o alguna de sus variantes. Permite un control muy preciso de todo el sistema haciéndolo muy poderoso, ya que incorpora en forma nativa herramientas para la administración de memoria y el acceso fácil al hardware sobre el que se ejecutan los programas. También, el C tiene una fuerte presencia en entornos fuera del ámbito de las computadoras personales o servidores, en micro dispositivos de control, dentro de micro controladores; esto se debe a la enorme flexibilidad del lenguaje. Aunque propiamente no sea un lenguaje orientado a objetos existe una enorme cantidad de lenguajes modernos que se basan en C, entre ellos, *C++*, *Visual C++*, *Objective C*, *Java*, *Java Script* y *C#*, entre otros. La similitud de estos lenguajes con el C va desde su sintaxis hasta el nombre de la gran mayoría de las funciones estándar. **(16)**

1.3.2.2 Lenguaje C++

C++ es una versión ampliada del lenguaje C. Incluye todo lo que forma parte de C y añade soporte para la programación orientada a objetos. Además, contiene muchas mejoras y características que sencillamente lo convierten en C mejor, independientemente de la programación orientada a objetos. Una particularidad del C++ es la posibilidad de redefinir los operadores (sobrecarga de operadores), y de poder crear nuevos tipos que se comporten como tipos fundamentales. C++ permite trabajar tanto a alto como a bajo nivel. **(17)**

1.3.2.3 Lenguaje C#

El lenguaje C# se deriva de C y de C++, y fue creado para el desarrollador corporativo dispuesto a sacrificar un poco de poder de C++ por una mayor comodidad y productividad. Un punto importante acerca de C# es que se trata de un lenguaje de programación moderno. Simplifica y moderniza C++ en las áreas de las clases, los espacios de nombres, la sobrecarga de métodos y la manipulación de excepciones. Gran parte de la complejidad de C++ fue eliminada de C# con el fin de facilitar su uso y cometer menos errores. Contribuyendo a su facilidad de uso destaca la eliminación de ciertas características de C++: desaparecen las macros, las plantillas y la herencia múltiple. Se añaden más características en aras de la comodidad, como la seguridad de los tipos, las versiones, la recogida de basura y muchas más. Todas ellas están dirigidas a desarrollar software orientado a componentes. **(18)**



1.3.2.4 Lenguaje Java

Java es un lenguaje de programación desarrollado por *Sun Microsystems* a principios de los años 90. Debido a su semejanza con C y C++, resulta muy fácil aprender por los programadores familiarizados con este. Diseñado como un lenguaje orientado a objetos desde el principio, los objetos agrupan en estructuras encapsuladas, tanto sus datos como los métodos (o funciones) que manipulan esos datos. Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas. Java proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que prescinde por completo de los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria. Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Para acomodar requisitos de ejecución tan variados, el compilador de Java genera *bytecodes*: un formato intermedio indiferente a la arquitectura, diseñado para transportar el código eficientemente a múltiples plataformas hardware y software. La indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas. Estas dos últimas características se conocen como la *Máquina Virtual Java (JVM)*. **(19)**

Debido a las características de cada lenguaje, y a los elementos arrojados por la investigación, el uso del Lenguaje C++ para la implementación del Subsistema de Video y al uso del Lenguaje de Programación Java para desarrollar el Subsistema de Audio es lo más conveniente, debido a las amplias bondades que nos brindan estos lenguajes, como la portabilidad, la fortaleza y teniendo muy en cuenta las características específicas de cada subsistema. Su uso también está motivado por el uso de plataformas Libres para el desarrollo de los mismos, logrando de esta manera la independencia tecnológica que necesita nuestro país.



1.4 Tendencias y metodologías relacionadas con las técnicas de programación.

Las tendencias actuales para el desarrollo de software apuntan a la portabilidad con la que se están concibiendo, la creación de programas que puedan ejecutarse en cualquier plataforma y la reutilización de código. Aunque continúan liderando el mercado las empresas de desarrollo de software privativo, productos de los cuales se desconoce la lógica programada y resultan soluciones caras a los problemas de las empresas y las personas, en los últimos tiempos se ha visto un incremento a la utilización de plataformas libres ya que permite el desarrollo de software con código fuente de acceso libre, fiable, eficaz y que funciona sobre prácticamente cualquier plataforma.

1.4.1 Programación en Plataformas Libres

Linux es posiblemente el sistema operativo para el que existe un mayor número de lenguajes, compiladores, bibliotecas y otras herramientas de desarrollo disponibles, lo cual permite a los programadores elegir lo que más se adecua a sus conocimientos y necesidades de cada proyecto en particular. Linux dispone de varias *shell* completamente programables mediante scripts, también conocidos como guiones. Comparativamente hablando, un script Linux es como un archivo de proceso por lotes del *DOS*, un típico *BAT*, con la diferencia de que las posibilidades son muy superiores. El lenguaje de programación por excelencia en *Linux* es *C*, ya que gran parte del sistema, se ha escrito con ese lenguaje. En cualquier instalación de Linux siempre existe un compilador de *C*, así como archivos de cabecera y bibliotecas estáticas y dinámicas. No es necesario usar *C* o *C++* para acceder a dichas bibliotecas, aunque es lo más habitual, gracias a la existencia de enlaces con otros muchos lenguajes, como *TCL*, *Perl*, *Python* o *Java*. La sofisticación de cara al usuario también ha alcanzado el área de las herramientas de desarrollo, habiendo disponibles múltiples entornos de tipo *RAD* (*Rapid Application Development*) que permiten al programador ahorrar gran cantidad de trabajo y centrarse en la funcionalidad de sus proyectos. Algunas de esas herramientas son comerciales y otras de uso libre, como el propio sistema. La mayor parte de todos los elementos necesarios para programar en *Linux* se encuentran formando parte del sistema, aunque esto depende de cada distribución así como del tipo de instalación realizada, y el resto pueden encontrarse fácilmente en sus respectivos repositorios o en Internet. **(20)**

Cabe señalar que en las plataformas libres se hace difícil la integración de los sistemas con interfaces amigables, por tal razón se ha extendido el uso de frameworks³ que gestionen de forma amena la creación de las interfaces de usuario mencionadas. En la implementación de los

³ Una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.



CAPÍTULO 1: Tendencias Actuales

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

módulos de audio y video se utilizarán Java y QT respectivamente por ser poderosas plataformas de desarrollo libres, con un uso muy extendido a nivel mundial y porque además poseen una fuerte comunidad de desarrollo y una amplia documentación.

1.4.1.1 Plataforma Java

La plataforma Java es el nombre de un entorno o plataforma de desarrollo originario de Sun Microsystems, capaz de ejecutar aplicaciones desarrolladas usando el Lenguaje de programación Java u otros lenguajes que compilen a bytecode y un conjunto de herramientas de desarrollo, una máquina virtual encargada de la ejecución de aplicaciones, y un conjunto de librerías estándar que ofrecen funcionalidad común.

La plataforma Java consta de dos componentes:

- **La Máquina Virtual de Java (JVM)**

La JVM es una de las piezas fundamentales de la plataforma Java. Básicamente se sitúa en un nivel superior al Hardware del sistema sobre el que se pretende ejecutar la aplicación, y este actúa como un puente que entiende tanto el bytecode, como el sistema sobre el que se pretende ejecutar convirtiendo el código bytecode a código nativo del dispositivo final.

- **La Interfaz de Programación de Aplicaciones de Java (API Java)**

El API Java es una Interfaz de Programación de Aplicaciones (API: por sus siglas en inglés) provista por los creadores del lenguaje Java, y que da a los programadores los medios para desarrollar aplicaciones Java. La API de Java provee de un conjunto de clases utilitarias para efectuar toda clase de tareas necesarias dentro de un programa. Está organizada en paquetes lógicos, donde cada paquete contiene un conjunto de clases relacionadas semánticamente. **(21)**

1.4.1.2 Plataforma QT

Qt es una amplia plataforma de desarrollo que incluye clases, librerías y herramientas para la producción de aplicaciones de interfaz gráfica en C++ que pueden operar en varias plataformas y otros muchos lenguajes de programación. Incluye soporte de nuevas tecnologías como OpenGL, XML, Bases de Datos y programación para redes. Dispone de tres grandes ventajas ante las librerías de ventanas rivales:

1. QT es completamente gratuito para aplicaciones de código abierto



CAPÍTULO 1: Tendencias Actuales

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

2. Las herramientas, librerías y clases están disponibles para casi todas las plataformas Unix y sus derivados (como Linux, MacOS X, Solaris, etc) como también para la familia Windows, por lo que una aplicación puede ser compilada y utilizada en cualquier plataforma sin necesidad de cambiar el código y la aplicación se verá y actuará mejor que una aplicación nativa.
3. Qt tiene una extensa librería con clases y herramientas para la creación de ricas aplicaciones. Estas librerías y clases están bien documentadas, son muy fáciles de usar y tienen una gran herencia de programación orientada a objetos. **(22)**

1.5 Conclusiones

Con la realización de este capítulo se logro conocer como se realizan los diferentes procesos de captura de audio y video, la transcripción del audio obtenido y la indexación del video capturado, los cuales dan una visión más clara y necesaria a la hora de implementar los subsistemas de Audio y Video. También el estudio de las técnicas actuales de programación, las tendencias y metodologías referentes a estas técnicas y los principales lenguajes y paradigmas de programación permitió realizar una comparación entre estos, conocer las ventajas y desventajas de cada uno de ellos y determinar cuál sería el más conveniente a la hora de implementar los módulos.



CAPÍTULO 2: Descripción de la solución propuesta

2.1 Introducción

En este capítulo se hará referencia al modelo de implementación, estructurado a partir de los diagramas de componentes. Todo lo anterior tiene su base en el diseño realizado del Producto Captura y Catalogación de Medias y particularmente de los Módulos de Audio y Video. Se analizan también los estilos y estándares de codificación utilizados en la implementación y que permiten una uniformidad en el código.

2.2 Modelo de Implementación

El modelo de implementación denota la implementación actual del sistema en términos de componentes y subsistemas de implementación. Describe cómo se organizan los elementos de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros. (23)

En el modelo de implementación del Producto Captura y Catalogación de Medias aparecen representados los componentes utilizados para el desarrollo de los subsistemas. Estos componentes son necesarios para la implementación de funcionalidades específicas. Ejemplo de estos son el **jVLC** usado para la reproducción de las medias, el **LAME** para la conversión del audio, el **MPlayer** para la captura y codificación de los videos, ya que además de reproductor, incluye el codificador de video Mencoder.

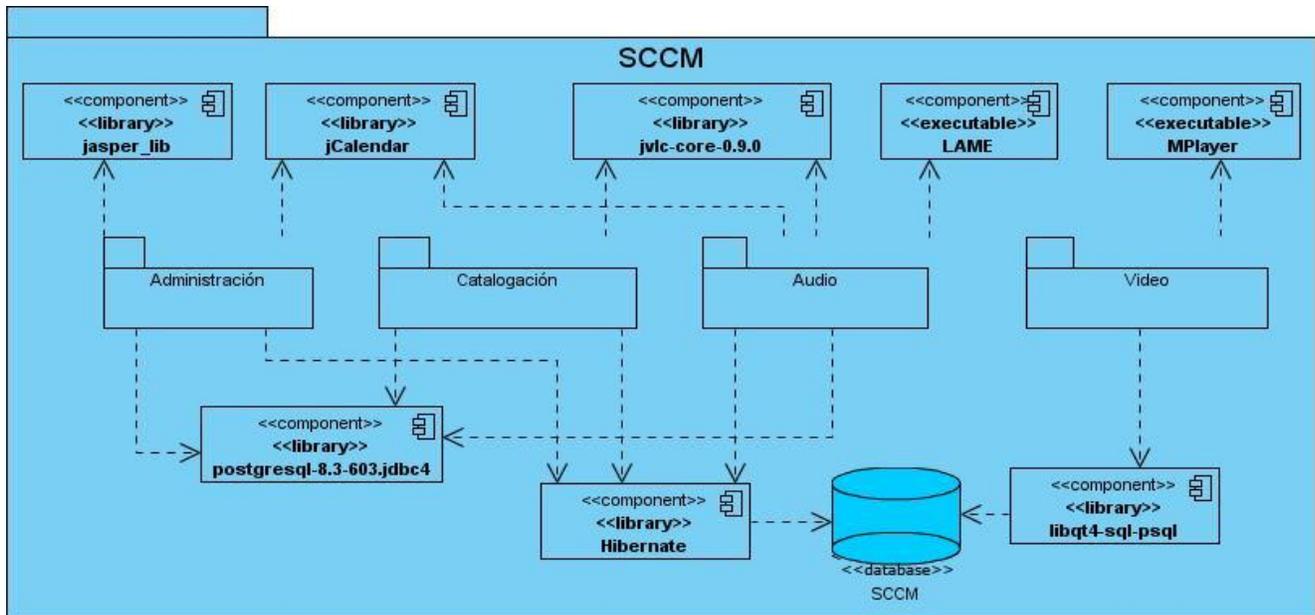


Figura 1: Modelo de Implementación



2.2.1 Diagrama de Componentes

La metodología RUP hace énfasis en el uso de diagramas de componentes para modelar los subsistemas de implementación. El modelo de implementación se estructura a partir de los diagramas de componentes, se debe entender el funcionamiento interno de los módulos para luego asociarlos como un todo dentro de los subsistemas que componen.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Estos diagramas muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de la aplicación informática. Pueden ser simples archivos, bibliotecas cargadas dinámicamente, entre otros. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente. (24)

UML define varios estereotipos estándar que se aplican a los componentes, como pueden ser:

- **<<executable>>**: programas que se ejecutan en un nodo.
- **<<file>>**: ficheros de datos o código fuente.
- **<<library>>**: librerías estáticas o dinámicas.
- **<<table>>**: tabla de base de datos.
- **<<document>>**: documento. (25)



2.2.1.1 Diagrama de Componentes del Módulo de Audio

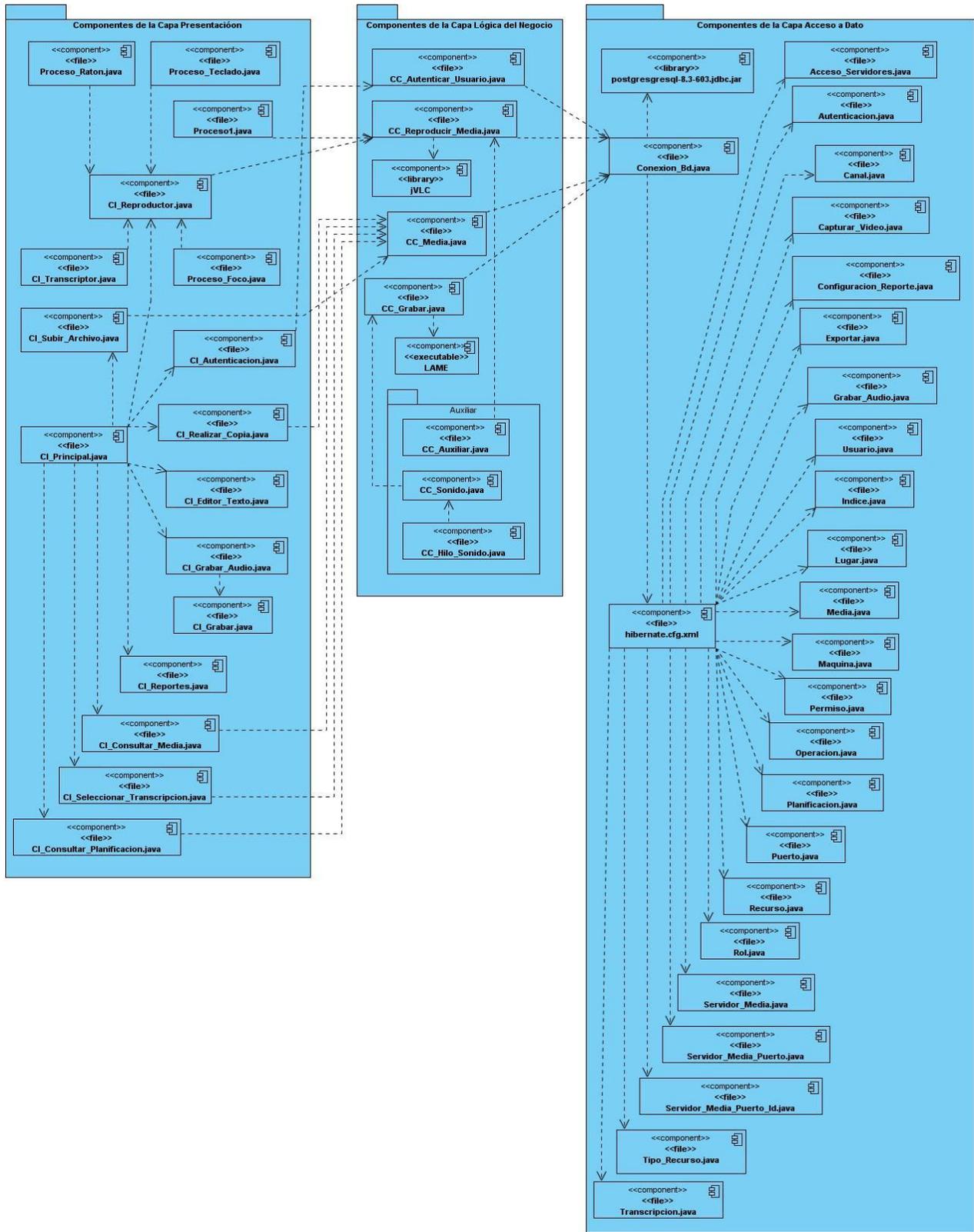


Figura 2: Diagrama de Componentes del Módulo de Audio



2.2.1.2 Diagramas de Componentes del Módulo de Video

El Módulo de Videos está compuesto por cuatro aplicaciones independientes, relacionadas entre sí por la base de datos. A continuación se mostrarán el diagrama de componentes para cada aplicación por separado.

- **Captura Automática**

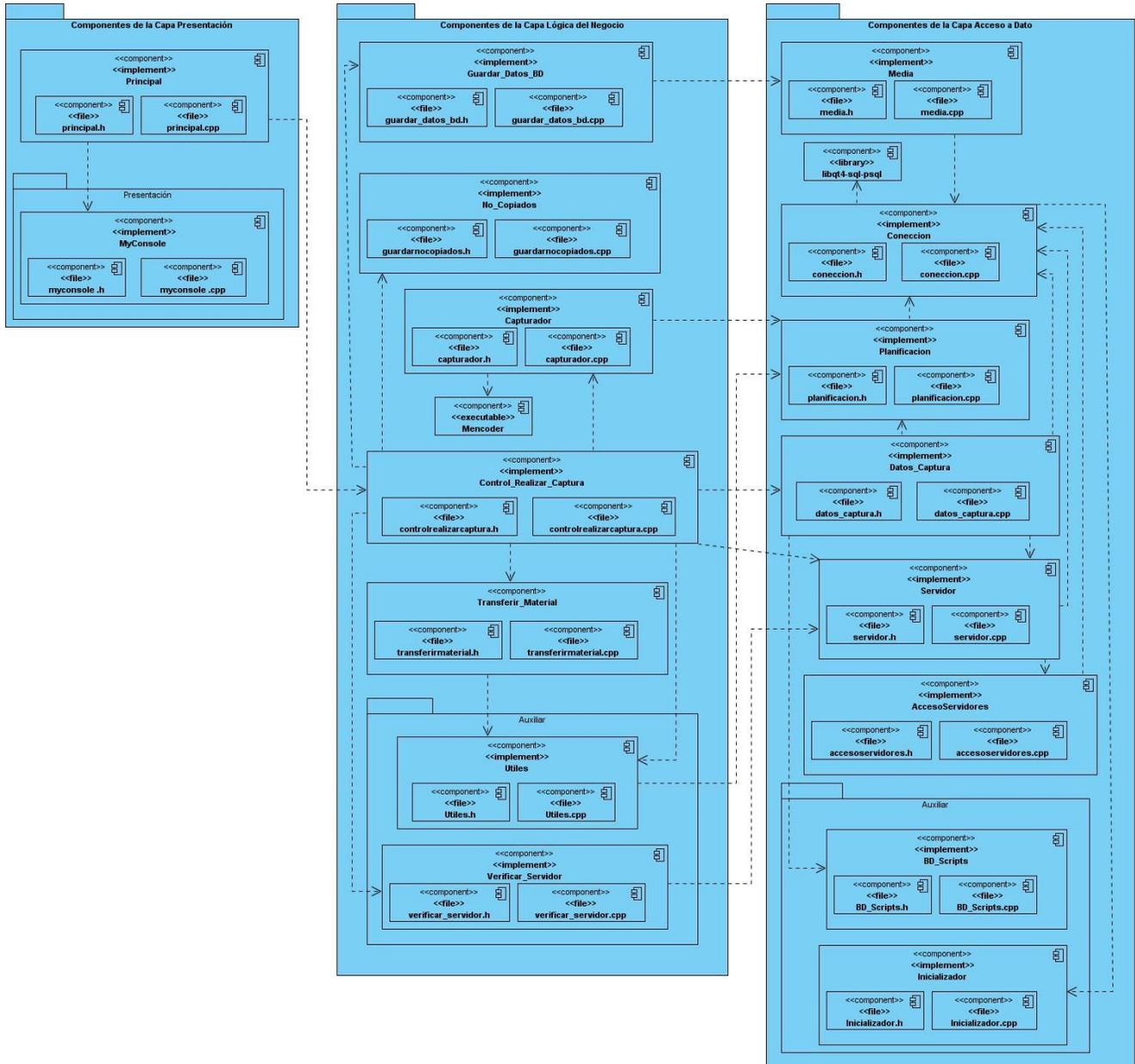


Figura 3: Diagrama de Componentes - Captura Automática



CAPÍTULO 2: Descripción de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

- **Captura Manual**



Figura 4: Diagrama de Componentes - Captura Manual



CAPÍTULO 2: Descripción de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

- Extracción de Fotogramas

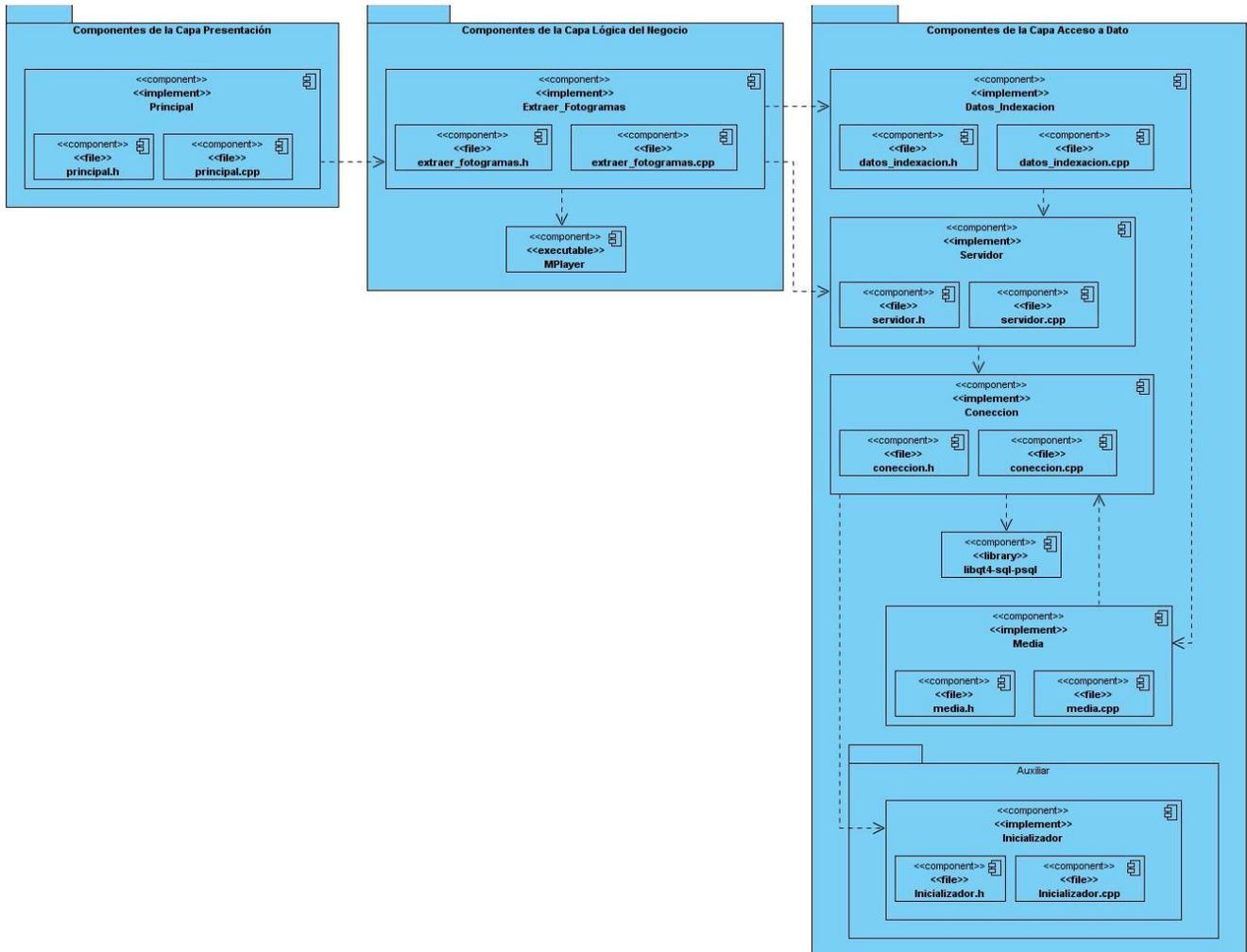


Figura 5: Diagrama de Componentes - Extracción de Fotogramas



- Fragmentar Videos

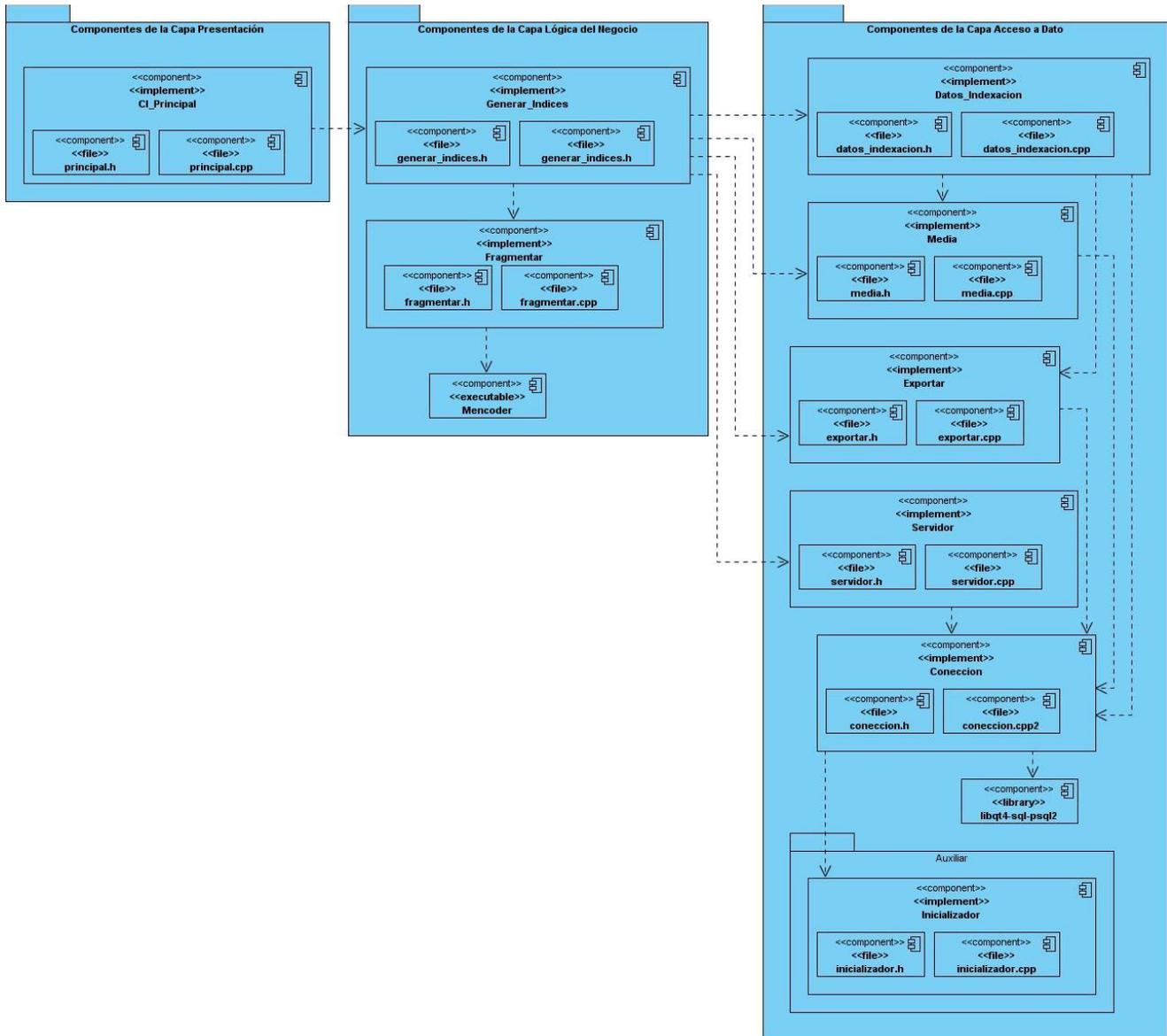


Figura 6: Diagrama de Componentes - Fragmentar Videos

2.3 Estándar de Codificación

Definido por la Real Academia de la Lengua Española como “que sirve como tipo, modelo, norma, patrón o referencia”, un estándar constituye una serie de lineamientos técnicos detallados, destinados a establecer uniformidad. En el caso específico de los estándares de codificación para los lenguajes de programación, las convenciones de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la



CAPÍTULO 2: Descripción de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

lectura, comprensión y mantenimiento del código. Un estándar de codificación completo comprende todos los aspectos de la generación de código, asegurando que todos los programadores del proyecto trabajen de forma coordinada. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento. **(26)**

Con la finalidad de que se cumplan estas premisas en la implementación de los Módulos de Audio y Video del Producto Captura y Catalogación de Medias se ha confeccionado un detallado Estándar de Codificación que permitirá una mayor legibilidad y fortaleza del código fuente. Se ha detallado específicamente para los lenguajes de programación C++ y Java, con los cuales se implementarán el módulo de video y audio respectivamente, lenguajes que aunque presentan una gran similitud, tienen sus diferencias y particularidades, las cuales son especificadas en los casos necesarios.

2.3.1 Criterios de Calidad

Es importante que durante la codificación se consideren permanentemente los siguientes criterios de calidad:

Criterio	Objetivo
Facilidad de Comunicación	Proporcionar al usuario entradas y salidas fácilmente asimilables.
Auto descripción	Proporcionar en el código, explicaciones sobre la implementación realizada.
Simplicidad	La implementación realizada debe hacerse de la forma más comprensible posible.

Tabla 1: Estándar de Codificación - Criterios de Calidad

2.3.2 Principios Generales

- Los nombres de cada uno de los elementos del programa debe explicar en lo posible su uso.
- No manejar en los programas más de una instrucción por línea.
- Declarar cada una de las variables en líneas separadas.
- Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.
- La mayoría de los elementos se deben nombrar usando sustantivos.
- Los atributos deben comenzar con letra minúsculas y los métodos deben comenzar con letra mayúsculas.



CAPÍTULO 2: Descripción de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

- Los parámetros deben comenzar con la letra p y con nombre los mas similar posible al atributo que se refiere.
- Nombramiento de Elementos

Elemento	Reglas de nombramiento
Clases e interfaces	Nombre sustantivo singular, con la primera letra en mayúscula y las demás en minúsculas.
Variables	Nombre sustantivo en mayúsculas. Para separar palabras se usará el guión bajo: _
Paquetes y Directorios y archivos fuente	Nombre sustantivo singular en minúsculas

Tabla 2: Estándar de Codificación - Principios Generales.

2.3.3 Comentarios

Cada programa deberá comenzar con un comentario que incluya:

- Autor
- Fecha
- Objetivo, o problema que resuelve el programa
- Algoritmo.

2.3.3.1 Cada función debe tener un encabezado que contenga:

- Objetivo de la función y no descripción del procedimiento.
- Comentarios de apoyo a variables, llamadas a función o inclusión de archivos que no sean obvios al proceso.
- Explicación de uso de argumentos (parámetros) no obvios.
- Explicación de uso de valores devueltos (de retorno).

2.3.3.2 Tipos de comentarios

- Comentario de Bloque

Ejemplo: /*Comentario en
Bloque*/



CAPÍTULO 2: Descripción de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

- Comentario de una Línea

Ejemplo: //Comentario de una Línea

2.3.4 Nombres de identificadores.

Se considera como identificador a los nombres de variables (arreglos, matrices, apuntadores), funciones, así como cualquier tipo de dato definido por el usuario (estructura, clase). Dichos identificadores deberán seguir las siguientes normas, además de las definidas por el propio lenguaje.

- Deberán tener un nombre significativo para que por su simple lectura, pueda conocerse su función, sin tener que consultar manuales o hacer demasiados comentarios.
- Para nombres que se usen con frecuencia o para términos largos, se recomienda usar abreviaturas estándar para que éstos tengan una longitud razonable. Si usa abreviaturas deben manejar la misma lógica en todo el programa.
- Evitar identificadores que comiencen con uno o dos caracteres de subrayado para evitar que se confundan con los que el compilador selecciona.
- Cada identificador de función, variable o procedimiento deberá ser precedido por la abreviación del tipo de dato de que es la variable.

2.3.5 Identificadores de variables.

Comenzarán siempre con la primera letra minúscula correspondiente a su tipo de dato. Para distinguir palabras dentro del nombre deberá emplearse un guión bajo (_).

Ejemplo:

sNombre_Video

2.3.6 Para el Caso de C++ (Identificadores de punteros (apuntadores))

Su nombre deberá comenzar con la letra **ptr**.

Ejemplo: ptrAlumno

Dónde pAlumno es un puntero que podrá tener la dirección del lugar donde se almacena información de un alumno.

2.3.7 Identificadores de variables dimensionadas (arreglos, matrices)

Su nombre deberá comenzar con las letras **ar**.

Ejemplo: arEstudiantes



Donde arEstudiantes es un arreglo de datos para guardar información de estudiantes.

2.3.8 Identificadores de datos constantes

Serán declaradas en letras mayúsculas.

Ejemplo: `const IVA = 0.15;`

2.3.9 Identificadores de funciones

La primera letra deberá ser mayúscula.

Ejemplo: `void Funcion ();`

2.3.10 Identificadores de tipos definidos por el usuario.

La primera letra será mayúscula.

Ejemplo: `class Clase o struct Estructura`

2.3.11 Líneas y espacios en blanco.

No insertar líneas en blanco entre instrucciones.

```
a = b + c;
```

```
int f;
```

```
f = a;
```

Las declaraciones de datos dentro de una función deberán ir al inicio

Deben incluirse espacios en ambos lados de los operadores binarios.

Ejemplo:

```
y = 50 + 15 - x;
```

2.3.12 Sangrías

Las sangrías tendrán una longitud de tres espacios.

Las llaves abren y cierran con la misma sangría.

Ejemplo: `void Funcion ()`

```
{  
    //Instrucciones de la función  
}
```



CAPÍTULO 2: Descripción de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

2.3.13 Tamaño de Líneas.

Establecer una longitud de línea máxima para el código. Si es necesario partir la línea, la siguiente línea debe alinearse dejando doble sangría.

Ejemplo:

```
if ((Key>=97&&Key<=122)||((Key>=48&&Key<=57)||((Key>=65&&Key<=90)||
    Key==8||Key==13))
```

2.4 Conclusiones

En este capítulo se identificaron elementos importantes que describen la solución propuesta y guían la implementación de los módulos de Audio y Video del Producto Captura y Catalogación de Medias. El modelo de implementación permitió tener una idea más clara de la composición interna de los subsistemas, pues mostró el funcionamiento interno de los módulos a través de los Diagramas de Componentes, conociéndose además los elementos que lo componen. Se especificó además el estándar de codificación utilizado en la implementación de los módulos, lo cual permite una mayor comprensión del código fuente, ya que provee de una herramienta que permite uniformidad a la hora de escribir el programa.



CAPÍTULO 3: Validación de la Solución Propuesta

3.1. Introducción

Para alcanzar la calidad requerida, un producto de software tiene que transitar por un proceso exhaustivo de pruebas. Existen diversos tipos de estas pruebas, cada cual con su objetivo específico y su estrategia bien definida, las que prueban los posibles errores que puedan existir en el software. En este capítulo se abordarán las características de las pruebas de Caja Blanca y Caja Negra, particularizando principalmente en las pruebas que se le realizarán a los módulos de Audio y Video previa Implementación.

3.2. Pruebas de Software

Las pruebas de software, en inglés *testing* o *beta testing* son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un software. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

3.2.1. Prueba de Caja Blanca

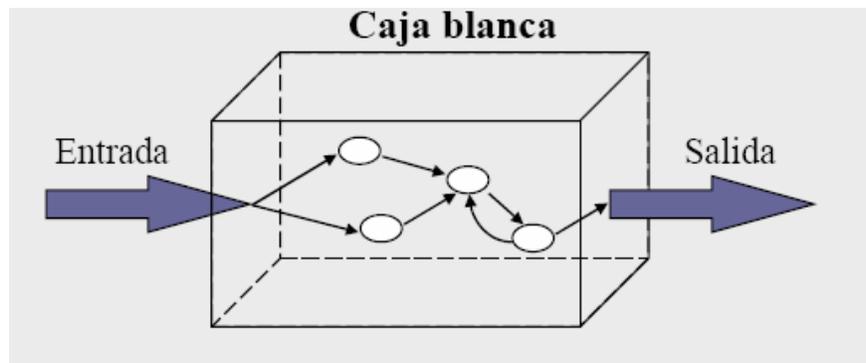


Figura 7: Prueba de Caja Blanca

Se denomina caja blanca a un tipo de prueba de software que se realiza sobre las funciones internas de un módulo. A este tipo de técnicas se le conoce también como Técnicas de Caja Transparente o de Cristal. Este método se centra en cómo diseñar los casos de prueba atendiendo al comportamiento interno y la estructura del programa. Se examina así la lógica interna del programa sin considerar los aspectos de rendimiento. El objetivo de la técnica es diseñar casos de prueba para que se ejecuten, al menos una vez, todas las sentencias del programa, y todas las condiciones tanto en su vertiente verdadera como falsa. **(27)**



CAPÍTULO 3: Validación de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

Algunos de los métodos empleados en las pruebas de caja blanca son los siguientes:

3.2.1.1 Prueba del camino básico: Es una técnica que permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizarán que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

Algunos elementos y conceptos utilizados alrededor de éste método son los siguientes:

- **Grafo de flujo o grafo del programa:** El grafo de flujo se utiliza para representar flujo de control lógico de un programa. Para ello se utilizan los tres elementos siguientes:
 - ✓ Nodos: Representan cero, una o varias sentencias en secuencia. Cada nodo comprende como máximo una sentencia de decisión (bifurcación).
 - ✓ Aristas: Líneas que unen dos nodos.
 - ✓ Regiones: Áreas delimitadas por aristas y nodos. Cuando se contabilizan las regiones de un programa debe incluirse el área externa como una región más.
 - ✓ Nodo predicado: Cuando en una condición aparecen uno o más operadores lógicos (AND, OR, XOR) se crea un nodo distinto por cada una de las condiciones simples. Cada nodo generado de esta forma se denomina nodo predicado.

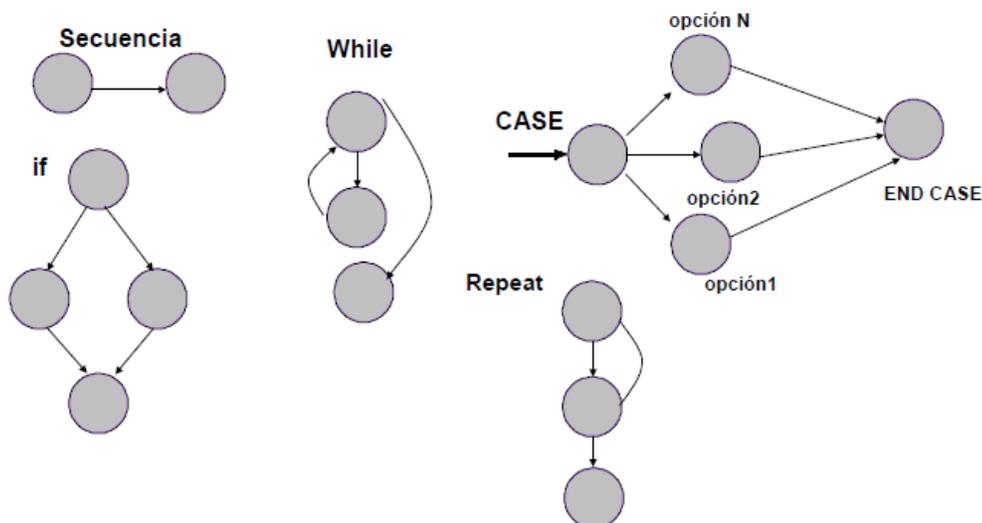


Figura 8: Grafo de flujo de las estructuras lógicas de un programa



CAPÍTULO 3: Validación de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

- **Complejidad ciclomática:** Es una métrica de software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto de las pruebas, el cálculo de la complejidad ciclomática representa el número de caminos independientes del conjunto básico de un programa. Esta medida ofrece al probador de software un límite superior para el número de pruebas que debe realizar para garantizar que se ejecutan por lo menos una vez cada sentencia.

Existen varias formas de calcular la complejidad ciclomática de un programa a partir de un grafo de flujo:

- o El número de regiones del grafo coincide con la complejidad ciclomática, $V(G)$.
- o La complejidad ciclomática, $V(G)$, de un grafo de flujo G se define como
$$V(G) = \text{Aristas} - \text{Nodos} + 2$$
- o La complejidad ciclomática, $V(G)$, de un grafo de flujo G se define como
$$V(G) = \text{Nodos Predicado} + 1 \quad (27)$$

- **Camino independiente:** Cualquier camino del programa que introduce, por lo menos, un nuevo conjunto de sentencias de proceso o una nueva condición.

De forma general, los pasos que se debe seguir para la obtención de los casos de prueba en este método, son los siguientes:

- Emplear el diseño o el código para elaborar el grafo de flujo.
- Determinar la complejidad ciclomática del grafo de flujo.
- Determinar un conjunto básico de caminos linealmente independientes.
- Preparar los casos de prueba que forzarán la ejecución de cada camino del conjunto básico.

3.2.1.2 Prueba de la estructura de control: Dentro de éste tipo de prueba se contempla el método del camino básico mencionado anteriormente pero además existen otras pruebas asociadas que permiten ampliar la cobertura de la prueba y mejorar su calidad.

3.2.1.3 Prueba de condición: Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa. Algunos conceptos empleados alrededor de esta prueba son los siguientes:



CAPÍTULO 3: Validación de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

- **Condición simple:** Es una variable lógica o una expresión relacional ($E1 <$ operador - relacional $> E2$).
- **Condición compuesta:** Está formada por dos o más condiciones simples, operadores lógicos y paréntesis.

En general los tipos de errores que se buscan en una prueba de condición, son los siguientes:

- Error en operador lógico.
- Error en variable lógica.
- Error en paréntesis lógico.
- Error en operador relacional.
- Error en expresión aritmética.

3.2.1.4 Prueba del flujo de datos: Selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.

3.2.1.5 Prueba de bucles: Es una técnica que se centra exclusivamente en la validez de las construcciones de bucles.

- **Bucles simples:** Se les aplica el siguiente conjunto de pruebas:
 - o Pasar por alto totalmente el bucle.
 - o Pasar una sola vez por el bucle.
 - o Pasar dos veces por el bucle.
 - o Hacer m pasos por el bucle con $m < n$ (donde n es el número máximo de pasos permitidos por el bucle).
 - o Hacer $n - 1$, n y $n + 1$ pasos por el bucle.
- **Bucles anidados:** Si se empleara el mismo enfoque de prueba de bucles simples a los bucles anidados, el número de pruebas aumentaría considerablemente por lo cual se sugiere emplear el siguiente enfoque:
 - o Comenzar por el bucle más interior. Establecer o configurar los demás bucles con sus valores mínimos.



CAPÍTULO 3: Validación de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

- Llevar a cabo las pruebas de bucles simples para el bucle más interior, mientras se mantienen los parámetros de iteración de los bucles externos en sus valores mínimos. Añadir otras pruebas para valores fuera de rango o excluidos.
 - Progresar hacia fuera, llevando a cabo pruebas para el siguiente bucle, pero manteniendo todos los bucles externos en sus valores mínimos y los demás bucles anidados en sus valores típicos.
 - Continuar hasta que se hayan probado todos los bucles.
- **Bucles concatenados:** Estos bucles se pueden probar utilizando el enfoque de bucles simples, siempre y cuando cada uno de los bucles sea independiente del resto, de lo contrario se debe emplear el enfoque de bucles anidados.
- **Bucles no estructurados:** Siempre que sea posible estos bucles deben rediseñarse. (28)

3.2.2. Prueba de Caja Negra

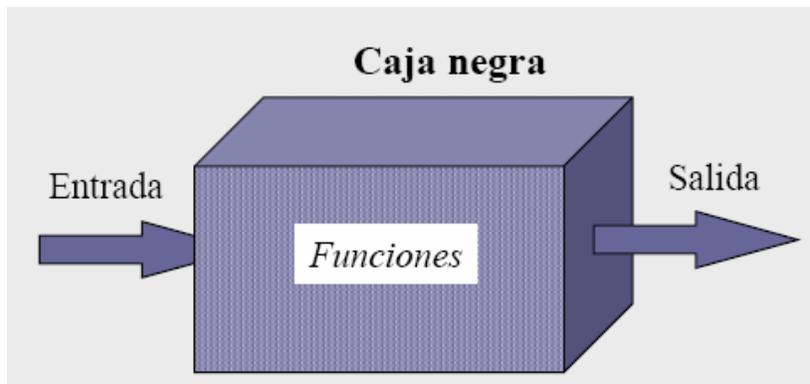


Figura 9: Prueba de Caja Negra

No está basada en el conocimiento del código o diseño interno, determina la funcionalidad del sistema con la entrada de datos y el resultado que estos brindan. Se centra en los requisitos fundamentales del software y permite obtener entradas que prueben todos los requisitos funcionales del programa. Comprobar que la funcionalidad del programa o sistema es completamente operativa. Que la entrada se acepta de forma adecuada y la salida es correcta. Verificar que la integridad de la información interna se mantiene. Con este tipo de pruebas se intenta encontrar:



CAPÍTULO 3: Validación de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación. **(29)**

No obstante, como el estudio de todas las posibles entradas y salidas de un programa sería impracticable se selecciona un conjunto de ellas sobre las que se realizan las pruebas. Para seleccionar el conjunto de entradas y salidas hay que tener en cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en nuestro sistema, y como consecuencia producen una serie de salidas que revelan la presencia de defectos. Entonces, dado que la prueba exhaustiva es imposible, el objetivo final es encontrar una serie de datos de entrada cuya probabilidad de pertenecer al conjunto de las que causan dicho comportamiento erróneo sea lo más alto posible. **(27)**

Para confeccionar los casos de prueba de Caja Negra existen distintos criterios, algunos de ellos son:

3.2.2.1 Grafo Causa-Efecto: Consiste en crear un grafo causa-efecto a partir de las especificaciones, y seleccionar suficientes casos de prueba como para asegurar la cobertura del grafo.

- Se llama causas a las características de los datos de entrada.
- Los efectos son las clases de salidas que puede proporcionar el programa. **(30)**

A partir del grafo causa-efecto se construye una tabla de decisión que refleje las dependencias entre causas y efectos. Lo que se hace entonces es reducir la tabla de decisión y seleccionar sólo un caso de prueba para todas las causas que producen el mismo efecto, o para cada columna de la tabla de decisión. **(31)**

3.2.2.2 Partición equivalente: Presenta la partición equivalente como un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la



CAPÍTULO 3: Validación de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. El objetivo de partición equivalente es reducir el posible conjunto de casos de prueba en uno más pequeño, un conjunto manejable que evalúe bien el software. Se toma un riesgo porque se escoge no probar todo. Así que se necesita tener mucho cuidado al escoger las clases.

La partición equivalente es subjetiva. Dos probadores quienes prueban un programa complejo pueden llegar a diferentes conjuntos de particiones. En el diseño de casos de prueba para partición equivalente se procede en dos pasos:

- **Se identifican las clases de equivalencia:** Las clases de equivalencia son identificadas tomando cada condición de entrada (generalmente una oración o una frase en la especificación) y repartiéndola en dos o más grupos.

Es de notar que dos tipos de clases de equivalencia están identificados: las clases de equivalencia válidas representan entradas válidas al programa, y las clases de equivalencia inválidas que representan el resto de los estados posibles de la condición (es decir, valores erróneos de la entrada).

- **Se define los casos de prueba:** El segundo paso es el uso de las clases de equivalencia para identificar los casos de prueba. El proceso es como sigue: se asigna un número único a cada clase de equivalencia. Hasta que todas las clases de equivalencia válidas han sido cubiertas por los casos de prueba, se escribe un nuevo caso de prueba que cubra la clase de equivalencia válida. Y por último hasta que los casos de prueba hayan cubierto todas las clases de equivalencia inválidas, se escribe un caso de la prueba que cubra una, y solamente una, de las clases de equivalencia inválidas descubiertas.

3.2.2.3 Análisis de valores límite: Los errores tienden a darse más en los límites del campo de entrada que en el centro. Por ello, se ha desarrollado el análisis de valores límites (AVL) como técnica de prueba. El análisis de valores límite es una técnica de diseño de casos de prueba que completa a la partición equivalente. En lugar de



CAPÍTULO 3: Validación de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la elección de casos de prueba en los extremos de la clase. En lugar de centrarse solamente en las condiciones de entrada, el AVL obtiene casos de prueba también para el campo de salida. **(32)**

Las pautas para desarrollar casos de prueba con esta técnica son:

- Si una condición de entrada especifica un rango de valores, se diseñarán casos de prueba para los dos límites del rango, y otros dos casos para situaciones justo por debajo y por encima de los extremos.
- Si una condición de entrada especifica un número de valores, se diseñan dos casos de prueba para los valores mínimo y máximo, además de otros dos casos de prueba para valores justo por encima del máximo y justo por debajo del mínimo.
- Aplicar las reglas anteriores a los datos de salida.
- Si la entrada o salida de un programa es un conjunto ordenado, habrá que prestar atención a los elementos primero y último del conjunto. **(27)**

- **Condiciones sublímites:** Las condiciones límites normales son las más obvias de descubrir. Estas son definidas en la especificación o son evidentes al momento de utilizar el software. Algunos límites, sin embargo, son internos al software, no son necesariamente aparentes al usuario final pero aún así deben ser probadas por el probador. Estas son conocidas como condiciones sublímites o condiciones límites internas. **(32)**

3.3. Pruebas de Software a los Subsistemas

A los programas que componen los Módulos de Audio y Video del Producto Captura y Catalogación de Medias se le aplicarán específicamente las Pruebas de Caja Negra con la intención de conocer las respuestas que nos brindan después de ser sometidos a diferentes entradas o condiciones, su rendimiento y comportamiento.

3.3.1. Prueba del Subsistema de Video

Como resultado de las pruebas de Caja Negra realizadas al Módulo de Video del Producto Captura y Catalogación de Medias del Departamento de Señales Digitales, se detectaron varios problemas los cuales son reflejadas en una serie de no conformidades reflejados en esta epígrafe del documento.



CAPÍTULO 3: Validación de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

3.3.1.1. Elementos probados.

Se les realizaron pruebas a los casos de uso:

- Autenticar usuarios
- Gestionar video
- Realizar captura de video
- Realizar transcodificación
- Registrar datos
- Transferir material
- Extraer fotogramas
- Generar índices

La descripción de los casos de uso se encuentra en los anexos.

3.3.1.2. Elementos no probados y causas

- No se probó el caso de uso: Digitalizar pues no se cuenta con los dispositivos necesarios como tarjetas de captura de video.
- No se probó el escenario: Subir material digital con el formato correcto, perteneciente a la sección: Subir material, del caso de uso: Gestionar video, porque está asociada al caso de uso: Digitalizar.
- El caso de uso: Realizar captura no se le probaron todas las funcionalidades porque no se cuenta con tarjeta de captura de video en el proyecto.
- No se probó el escenario: Realizar transcodificación de la digitalización en curso, perteneciente al caso de uso: Realizar transcodificación, porque no se cuenta con los dispositivos necesarios.

3.3.1.3. Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Clasificación	Estado NC
En el Caso de uso: <u>Realizar captura de video</u> (ver Anexos) al almacenar en el servidor la media, si ocurre algún problema de conexión o que el servidor está lleno.	1	No se muestra mensaje al usuario.	Al almacenar en el servidor la media, si ocurre algún problema de conexión o que el servidor está lleno, se debe notificar al usuario de lo que sucede.	S: de tipo Excepciones	16/abril/201 0 PD



CAPÍTULO 3: Validación de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

En el Caso e uso: <u>Transferir material al</u> almacenar en un material y no hay capacidad en el servidor.	2	No está implementada esta funcionalidad en el Módulo de captura manual.	Al almacenar en el servidor la media, el servidor está lleno, se debe lanzar un reporte de error y almacenar el material en un directorio local. Esto no ocurre para el Módulo de captura manual.	S: de tipo Funcionalidad	16/abril/201 0 PD
En el Caso de uso: <u>Extraer Fotogramas</u> si no se especifica una media.	3	No está implementada esta funcionalidad.	Si no se especifica una media no se notifica en la consola la constancia del error y no se escribe en la tabla operaciones.	S: de tipo Funcionalidad	16/abril/201 0 PD

Tabla 3: Pruebas Subsistema de Video - Defectos y dificultades detectados

3.2.1. Prueba del Subsistema de Audio

A partir de las Pruebas de Caja Negra realizadas al Módulo de Audio del Producto Captura y Catalogación de Medias del Departamento de Señales Digitales, se han obtenido una serie de no conformidades y se han detectado errores, a los cuales se les hará referencia en este epígrafe.

3.2.1.1. Elementos probados.

Se les realizaron pruebas a los casos de uso:

- Autenticar usuario
- Gestionar Archivo
- Grabar Audio
- Grabación Automática
- Grabación Manual
- Reproducir Audio
- Transcribir Audio

La descripción de los casos de uso se encuentra en los anexos.

3.2.1.2. Elementos no probados y causas

Se probaron todos los casos de uso críticos del Subsistema de Audio del Producto Captura y Catalogación de Medias (SCCM).



CAPÍTULO 3: Validación de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

3.2.1.3. Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Clasificación	Estado NC
En el caso de uso: <u>Gestionar Archivo</u> , en la sección " <u>Subir archivo</u> " cuando culmina la transferencia del audio	1	No limpia el visual para poder introducir nuevos datos si se necesita subir otro archivo de audio	Al terminar de transferir el archivo de audio hacia el servidor de medias.	S: de tipo Funcionalidad	27/abril/2010 PD
En el caso de uso: <u>Gestionar Archivo</u> , en la sección " <u>Subir archivo</u> " cuando culmina la transferencia del audio	2	No informa al usuario cuando se trata de volver a adicionar el archivo anteriormente subida	Al intentar subir una media que ya se había transferido en el paso anterior.	S: de tipo Excepción	27/abril/2010 PD
En la Sección " <u>Pausar/Continuar Grabación</u> " del caso de uso <u>Grabación Automática</u>	3	Luego de pausar la Grabación, al reiniciarla, se pierde el audio capturado con anterioridad.	Mientras se realiza una grabación al pausar la misma y reanudarla.	S: de tipo Funcionalidad	27/abril/2010 PD
En el caso de uso <u>Grabación Automática</u> luego de comenzada una grabación	4	No se encuentra implementado el control de Volumen.	Mientras se realiza una grabación se necesita controlar el volumen de la misma.	S: de tipo Funcionalidad	27/abril/2010 PD
En la Sección " <u>Pausar/Continuar Grabación</u> " del caso de uso <u>Grabación Manual</u>	5	Luego de pausar la Grabación, al reiniciarla, se pierde el audio capturado con anterioridad.	Mientras se realiza una grabación al pausar la misma y reanudarla.	S: de tipo Funcionalidad	27/abril/2010 PD
En el caso de uso <u>Grabación Manual</u> luego de comenzada una grabación	6	No se encuentra implementado el control de Volumen.	Mientras se realiza una grabación se necesita controlar el volumen de la misma.	S: de tipo Funcionalidad	27/abril/2010 PD



CAPÍTULO 3: Validación de la Solución Propuesta

IMPLEMENTACIÓN: Subsistemas de Audio y Video del Producto Captura y Catalogación de Medias

En el caso de uso: <u>Transcribir Audio</u> cuando se selecciona ver una transcripción	7	Se abre el editor de textos sin la transcripción. Hay que ir al botón abrir desde servidor para que se cargue.	Se abre el editor de texto con el reproductor que permite transcribir el audio que se reproduce.	S: de tipo Funcionalidad	27/abril/2010 PD
---	---	---	---	-----------------------------	-------------------------

Tabla 4: Pruebas Subsistema de Audio - Defectos y dificultades detectados

3.3. Conclusiones

Luego de haber conocido algunas de las diversas técnicas existentes para realizar las pruebas del software y haber aplicado una en específico a los Módulos de Audio y Video se pudieron encontrar una serie de no conformidades y defectos que presentan las aplicaciones obtenidas, lo que nos permitirá mejorar y así obtener una versión más potente y libre de fallos. Con esto se logra una mayor calidad en el producto obtenido, pues, al ser sometidos a las pruebas pertinentes que confirmen las funcionalidades previstas, se logra que los subsistemas sean lo más robustos posible.



CONCLUSIONES GENERALES

A través del desarrollo de la investigación se ha hecho un análisis exhaustivo de cómo se realizan los diferentes procesos de los subsistemas, la captura de las medias, los procesos de indexación, fragmentación y transcripción. Se estudiaron las tendencias y técnicas de programación actuales, las cuales brindaron una herramienta cognoscitiva y sirvieron de base a la hora de implementar los subsistemas. Se realizó además el Modelo de Implementación, apoyado por los diferentes diagramas de componentes, los cuales brindaron una idea general de cómo estaba integrado cada subsistema y cada aplicación y la relación de dependencia entre las diferentes clases y los componentes que intervienen en la confección de los mismos. Se definió un Estándar de Codificación con el cual se logra uniformidad y estética en el código escrito, posibilitando una mayor legibilidad. Y por último se validó de la solución propuesta al realizarle pruebas que certificaron las funcionalidades implementadas y permitieron encontrar fallas y corregir errores. Como resultado de dicha investigación se logró la implementación de 2 subsistemas que incorporan procesos automáticos que de una forma u otra optimizan las acciones y reducen errores humanos que anteriormente se presentaban por el seguimiento continuo de las planificaciones que en muchas ocasiones podían ejecutarse en horarios de sueño.



RECOMENDACIONES

Se recomienda la profundización en el funcionamiento de las API que utilizan los software de apoyo, para de esta forma ganar en independencia y factibilidad del sistema. Además es necesario indagar más sobre las técnicas de indexación y la detección automática de patrones en las medias, con el fin de brindar una mayor gama de prestaciones. Por último se recomienda el estudio de el envío de señales entre aplicaciones con tecnologías como Corba⁴ o Ice, pues actualmente todo los procesos se realizan leyendo constantemente de la Base de Datos, lo que enlentece la conexión y puede llegar a saturar la red.

⁴ (Arquitectura Común de Intermediarios en Peticiones a Objetos), es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.



TRABAJOS CITADOS

1. Pascual, Miguel Ángel Martín. **Universidad Autónoma de Barcelona.** [En línea] [Citado el: 8 de diciembre de 2009.]
http://www.cac.cat/pfw_files/cma/premis_i_ajuts/treball_guanyador/Menci__Miguel_A__Martin.pdf.
2. **Universidad de Oviedo.** [En línea] [Citado el: 7 de diciembre de 2009.]
http://www.uniovi.es/mdelbuey/documentos/documentos/area%20academica/bases/cu_rso%202001-2002/deficiencia%20auditiva%2001-02.PDF.
3. Enrich, Juan Juliá. **Radio Historia y Técnica . s.l. : MARCOMBO, 2004. ISBN: 9788426713469 .**
4. **SensAgent.** [En línea] [Citado el: 1 de noviembre de 2009.]
<http://dictionnaire.sensagent.com/cine/es-es/>.
5. Sadoul, Georges. **Historia del Cine Mundial: Desde los Orígenes. s.l. : Siglo XXI editores, S.A. de C.V, Decimonovena Edición en Español, 2004. ISBN 968-23-0533-0.**
6. Sanjuán, Olga Pérez. **Recordando la Historia: El concepto "televisión" en sus orígenes. 2007.**
7. Zayas, Dr. Cs. Carlos Alvarez de. **Metodología de la Investigación Científica. Santiago de Cuba : Universidad de Oriente, 1995.**
8. **Universidad Don Bosco.** [En línea] [Citado el: 7 de diciembre de 2009.]
<http://www.udb.edu.sv/Academia/Laboratorios/informatica/internet/guia3llnva.pdf>.
9. **LAME MP3 Encoder.** [En línea] [Citado el: 7 de diciembre de 2009.]
<http://lame.sourceforge.net/about.php>.
10. **VLC.** [En línea] [Citado el: 7 de diciembre de 2009.] <http://www.videolan.org/vlc/>.
11. Bimbo, Alberto del. **Visual Information Retrieval. California : Morgan Kaufmann Publishers, 1999. 1-55860-624-6.**
12. **Universidad de Murcia.** [En línea] [Citado el: 18 de Marzo de 2010.]
<http://www.um.es/docencia/barzana/lAGP/lagp3.html>.
13. **Plataforma.** [En línea] [Citado el: 8 de diciembre de 2009.]
<http://plataforma.cbtis122.net/mod/resource/view.php?id=1195>.
14. Fokker, Jeroen. **Departamento de Informática. Universidad de Utrecht.** [En línea] 1996. [Citado el: 20 de Abril de 2010.] <http://people.cs.uu.nl/jeroen/courses/fp-sp.pdf>.
15. Lanzillotta, Analía. **Definición de Leguajes de Programación. Master Magazine.** [En línea] [Citado el: 8 de diciembre de 2009.]
<http://www.mastermagazine.info/termino/5560.php>.
16. Ruiz, Marcelo Hernán. **Programación C. Ciudad de Buenos Aires, Argentina : MP Ediciones, 2003. ISBN 987-526-155-6.**
17. Schildt, Herbert. **C++ Guía de Autoenseñanza. Madrid : McGRAW-HILL, 1995. ISBN: 0-07-882025-1.**
18. Wille, Cristoph. **C #. 1ra Edición en Español. Madrid : PEARSON EDUCACION, S.A.,, 2001. pág. 3. ISBN: 0-672-32037-1.**
19. Marañón, Gonzalo Álvarez. **Características del lenguaje Java.** [En línea] [Citado el: 8 de diciembre de 2009.] <http://www.iec.csic.es/CRIPTONOMICON/java/quesjava.html>.
20. Charte Ojeda, Francisco. **Programación en GNU / LINUX . Madrid : Anaya, 2003. ISBN 84-415-1544-1.**

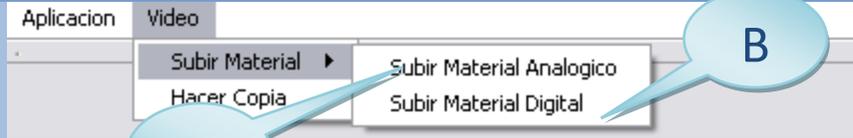


21. Universidad Nacional Autónoma de México. [En línea] [Citado el: 20 de enero de 2010.] http://profesores.fi-b.unam.mx/carlos/java/java_basico1_5.html.
22. Universidad Pampeu Fabra. [En línea] [Citado el: 8 de Febrero de 2010.] <http://microe.udea.edu.co/proyectos/DMA/enlaces/articulo/Selección de una Plataforma para el Desar.pdf>.
23. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 1999.
24. Letelier Torres, Patricio. *Desarrollo de Software Orientado a Objeto usando UML*. s.l. : Universidad Politécnica de Valencia.
25. Castilla, Universidad de. Departamento de Sistemas Informáticos. [En línea] [Citado el: 2 de Febrero de 2010.] <http://www.info-ab.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.
26. Microsoft. [En línea] [Citado el: 9 de Febrero de 2010.] [http://msdn.microsoft.com/es-es/library/aa291591\(VS.71\).aspx](http://msdn.microsoft.com/es-es/library/aa291591(VS.71).aspx).
27. Universidad de Sevilla. *Departamento de Lenguajes y Sistemas Informáticos*. [En línea] [Citado el: 18 de Marzo de 2010.] <http://www.lsi.us.es/docencia/get.php?id=361>.
28. Universidad Distrital Francisco José de Caldas. *Pruebas de Caja Blanca*. [En línea] [Citado el: 16 de Marzo de 2010.] <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node26.html>.
29. Universidad Carlos III de Madrid. [En línea] [Citado el: 18 de Marzo de 2010.] http://ocw.uc3m.es/informatica/ingenieria-de-software-iii/materialclase/ISIII_09_PRUE.pdf.
30. Universidad Politécnica de Madrid. *Facultad de Informática* . [En línea] [Citado el: 17 de Marzo de 2010.] <http://lml.ls.fi.upm.es/~jramirez/ed2/pruebas.ppt>.
31. Instituto Tecnológico Leonardo Da Vinci. [En línea] [Citado el: 18 de Marzo de 2010.] www.ldv.com.ar/alumnos/calidad/cap2.doc.
32. Universidad Distrital Francisco José de Caldas. *Pruebas de Caja Negra*. [En línea] [Citado el: 17 de Marzo de 2010.] <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node28.html>.



Anexo 1: Caso de Uso del Módulo Video - Gestionar Video.

Caso de Uso:	Gestionar Video	
Actores:	Usuario de Captura	
Resumen:	Mediante este caso de uso el usuario de captura puede subir ficheros al servidor y hacer copias de estos fuera del servidor.	
Precondiciones:	Que exista un dispositivo disponible para realizar la captura o un fichero para almacenar en el servidor.	
Referencias	R2	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario necesita subir o hacer una copia de un material.	1.1 Se muestra una interfaz que brinda la posibilidad de ejecutar las siguientes acciones: a) Si decide subir fichero al servidor (A) ir a la sección "Subir material" c) Si decide hacer copia (B) de un fichero del servidor ir a la sección "Hacer copia"	
<i>Prototipo de Interfaz</i>		
Sección "Subir material"		
Acción del Actor	Respuesta del Sistema	
1. El usuario de captura solicita la opción de subir un material al servidor.	1.1 Se muestra una interfaz en la que se pide al usuario la selección de subir un material analógico o digital. a) Si decide subir material analógico (A) ver sección "Subir material analógico". b) Si decide subir material digital (B) ver sección "Subir material digital".	
<i>Prototipo de Interfaz</i>		



Sección "Subir material analógico"

Acción del Actor	Respuesta del Sistema
	1.1. Se cargan los dispositivos de captura del ordenador y se procede a la digitalización (Ver Caso de Uso Digitalizar).

Prototipo de Interfaz

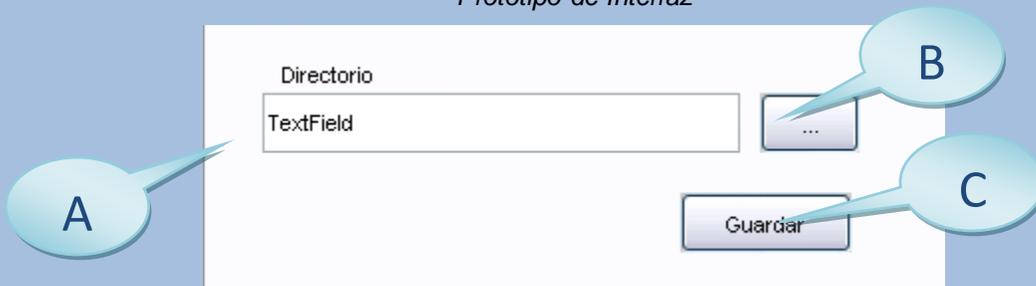
Sección "Subir material digital"

Acción del Actor	Respuesta del Sistema
	1.1. Se muestra la interfaz con el campo dirección de origen del archivo que desea agregar al servidor (A).
2. El usuario de captura indica la dirección de origen del fichero (B) y acepta (C).	2.1 El sistema comprueba que el formato del fichero MPEG2. 2.2 Guarda en el servidor de media y registra los datos en el servidor de BD. (Ver Casos de Usos Transferir material y Registrar datos).

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	2.1 Si el archivo que se quiere subir no tiene formato requerido por el sistema se transcodifica el mismo al formato deseado y se sube al servidor. (Ver Caso de Uso Realizar transcodificación).

Prototipo de Interfaz

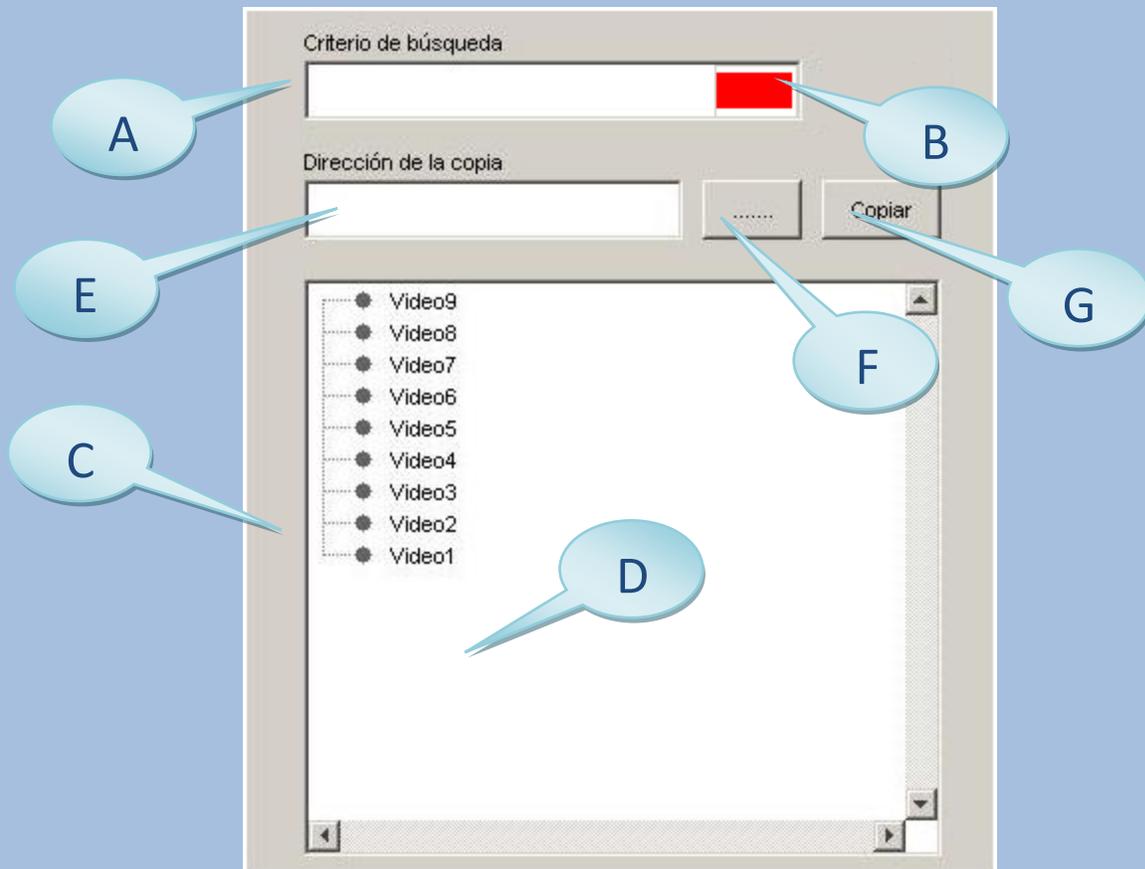


Sección "Hacer Copia"



Acción del Actor	Respuesta del Sistema
	1.1 Se muestra una interfaz en la que se pide al usuario seleccionar la fecha en que se capturo el del fichero (A)
2. El usuario la fecha	2.1. El sistema muestra los resultados de la búsqueda (C) basado en la fecha seleccionada por el usuario.
3. El usuario de captura selecciona el video (D), indica el destino de la copia (E, F) que desea y ejecuta la acción de copiar (G).	3.1 El sistema realiza una copia del fichero existente en el servidor en la dirección indicada por el usuario.

Prototipo de Interfaz



Flujos Alternos

Acción del Actor	Respuesta del Sistema
	1.1 Muestra un mensaje informando al usuario que no encuentra ningún video con las características solicitadas en la búsqueda.

Tabla 5: Caso de Uso del Módulo Video - Gestionar Video.



Anexo 2: Caso de Uso del Módulo Video - Realizar Captura de Video

Caso de Uso:	Realizar captura de video	
Actores:	Proceso Automático de Captura	
Resumen:	El caso de uso se inicia siguiendo los tiempos definidos por la planificación descrita por el servidor de BD. Realiza una conversión del material a partir de las especificaciones del formato de video al que se desea transcodificar.	
Precondiciones:	Debe existir una planificación para ser capturada en la PC de captura especificada.	
Referencias	R2	
Prioridad	Crítico	
<i>Flujo Normal de Eventos</i>		
<i>Sección "Realizar captura de video"</i>		
	<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
1.	Lee desde el servidor de BD los datos pertinentes a la captura de video en dicha estación (canal, dispositivo, fecha, tiempo inicio, tiempo fin, nombre del material).	
2.	Manda a ejecutar la captura de video a partir de la fecha planteada, siguiendo los tiempos leídos.	2.1. Ejecuta la captura de video y transcodifica el material (ver descripción del Caso de Uso Realizar transcodificación).
<i>Prototipo de Interfaz</i>		
<i>Flujos Alternos</i>		
	<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
<i>Prototipo de Interfaz</i>		
Poscondiciones	Se obtiene un material codificado en un directorio local de la máquina de captura.	

Tabla 6: Caso de Uso del Módulo Video - Realizar Captura de Video



Anexo 3: Caso de Uso del Módulo Video - Digitalizar

Caso de Uso:	Digitalizar
Actores:	CU Gestionar video
Resumen:	El usuario digitaliza un material provisto por un medio de almacenamiento no digitalizado.
Precondiciones:	Debe existir un material en un soporte analógico.
Referencias	R2
Prioridad	Crítico

Flujo Normal de Eventos

Sección "Digitalizar"

Acción del Actor	Respuesta del Sistema
	1.1. Se muestra una interfaz con el campo correspondiente al dispositivo (B) con el que se va a digitalizar.
2. Se escoge el dispositivo deseado y se solicita la digitalización del material (C).	2.1. Realiza el proceso de digitalización del material. 2.2. Visualiza una pantalla (A) con el flujo que se está digitalizando. 2.3. Transcodifica el material digitalizado. (Ver Caso de Uso Realizar transcodificación).
3. Solicita la acción de parar la digitalización (E).	3.1. Para la digitalización y transfiere el material al servidor de media (Ver Caso de Uso Transferir material).

Prototipo de Interfaz



Flujos Alternos

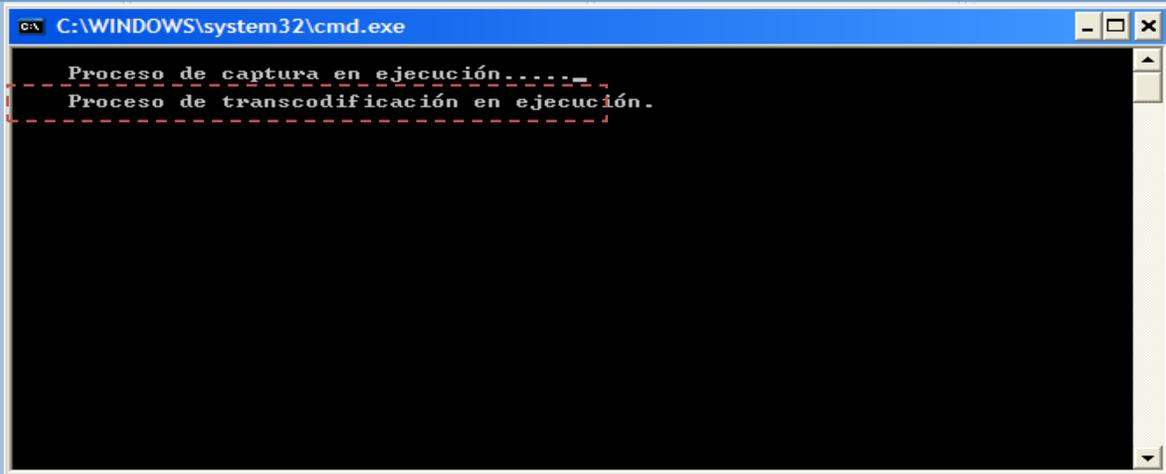


Acción del Actor		Respuesta del Sistema
<i>Prototipo de Interfaz</i>		
Poscondiciones	Se obtiene un material que proviene de fuentes externas no digitales.	

Tabla 7: Descripción de Casos de Uso - Digitalizar

Anexo 4: Caso de Uso del Módulo Video - Realizar transcodificación

Caso de Uso:	Realizar transcodificación	
Actores:	CU Realizar captura de video, CU Digitalizar, CU Gestionar video	
Resumen:	El caso de uso se realiza cuando es necesario que los materiales a almacenar tengan el formato especificado por el sistema.	
Precondiciones:	Se debe haber comenzado una captura de video, un proceso de digitalización o existir un material en un formato no deseado.	
Referencias	R3	
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección "Realizar transcodificación"		
Acción del Actor		Respuesta del Sistema
		<ol style="list-style-type: none"> 1. Se realizará la transcodificación atendiendo a: <ol style="list-style-type: none"> a) El comienzo del proceso de captura de video desde una señal de televisión. b) El comienzo del proceso de digitalización de video. c) La entrada de un material para subir que no contiene el formato especificado por el sistema. 2. Se guardan los datos de las medias (Ver Caso de Uso Registrar datos). 3. Traslada los materiales al servidor (Ver CU Transferir material).
<i>Prototipo de Interfaz</i>		



Sección “Realizar transcodificación en el proceso de captura de video”

Acción del Actor	Respuesta del Sistema
1. Lee desde el servidor de BD los datos pertinentes a la transcodificación (formato).	1.1. Se realiza la transcodificación siguiendo los tiempos de la captura en curso.

Sección “Realizar transcodificación en el proceso de digitalización de un material”

Acción del Actor	Respuesta del Sistema
1. Lee del servidor de BD los datos pertinentes a la transcodificación (formato).	1.1. Se realiza la transcodificación siguiendo los tiempos de la digitalización en curso.

Sección “Realizar transcodificación en un material con formato incorrecto”

Acción del Actor	Respuesta del Sistema
2. Lee del servidor de BD los datos pertinentes a la transcodificación (formato).	2.1. Se realiza la transcodificación.

Flujos Alternos

Acción del Actor	Respuesta del Sistema

Prototipo de Interfaz

Poscondiciones	Se obtiene un material codificado.
----------------	------------------------------------

Tabla 8: Caso de Uso del Módulo Video - Realizar Transcodificación



Anexo 5: Caso de Uso del Módulo Video - Registrar datos

Caso de Uso:	Registrar datos.	
Actores:	CU Realizar transcodificación.	
Resumen:	Se identifican los principales parámetros de un video (formato, tamaño, duración, etc.) además del tiempo y fecha en que se realizó la incorporación del mismo. Con esto se garantiza el estado inicial de los ficheros obtenidos y de sus propiedades.	
Precondiciones:	Debe haber terminado el proceso de transcodificación o se debe haber subido al servidor un material con el formato requerido por el sistema.	
Referencias	R4	
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección "Registrar datos"		
Acción del Actor	Respuesta del Sistema	
1. Analiza el material y solicita enviar los datos al servidor de BD.	1.1. Guarda los datos correspondientes al material en la BD.	
<i>Prototipo de Interfaz</i>		
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
<i>Prototipo de Interfaz</i>		
<pre> C:\WINDOWS\system32\cmd.exe Proceso de captura en ejecución....._ Proceso de transcodificación en ejecución. Guardando datos en el \$B...!.._ </pre>		
Poscondiciones	Datos iniciales en la BD de las medias capturadas	

Tabla 9: Caso de Uso del Módulo Video - Registrar Datos



Anexo 6: Caso de Uso del Módulo Video - Transferir Material

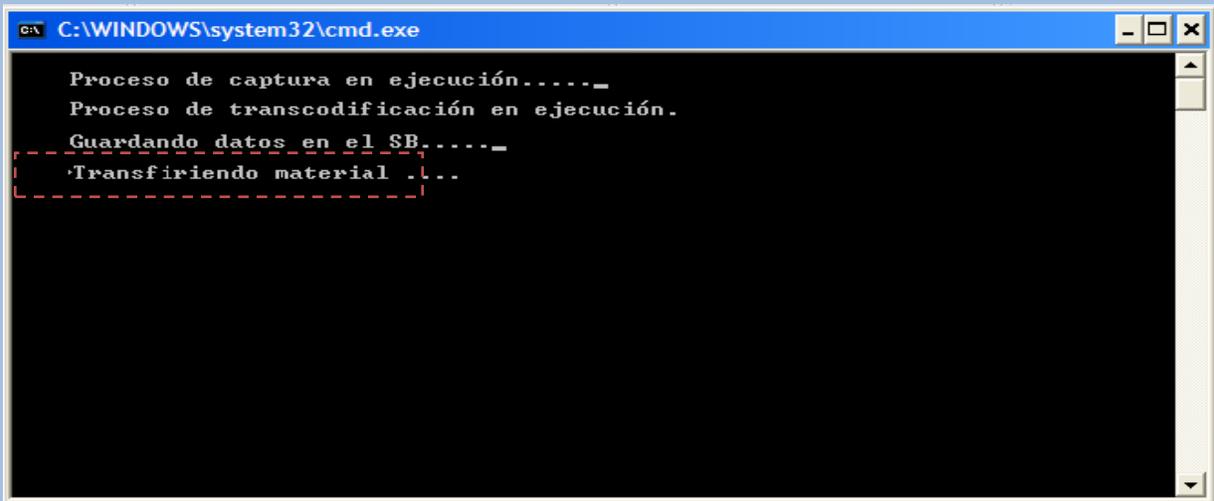
Caso de Uso:	Transferir material
Actores:	CU Realizar transcodificación, CU Extraer fotograma, CU Gestionar video, CU Generar índices.
Resumen:	El caso de uso gestiona la transferencia de un fichero hacia un destino especificado.
Precondiciones:	Debe existir un directorio destino y un material en un directorio origen.
Referencias	R5
Prioridad	Crítico

Flujo Normal de Eventos

Sección "Transferir material"

Acción del Actor	Respuesta del Sistema
1. Lee de la BD los datos de la dirección a donde tiene que transferir el material (directorio destino).	
2. Ejecuta la acción de transferir el material.	2.1. Verifica la disponibilidad de almacenamiento del directorio especificado. 2.2. Ejecuta la transferencia del material

Prototipo de Interfaz



Flujos Alternos

Acción del Actor	Respuesta del Sistema
	2.1 Lanza un reporte de capacidad insuficiente para realizar la transferencia al administrador del sistema. 2.2 Transfiere el material a una dirección local.



<i>Prototipo de Interfaz</i>	
Poscondiciones	Se guarda el material en el servidor de media, para posterior análisis.

Tabla 10: Caso de Uso del Módulo Video - Transferir Material

Anexo 7: Caso de Uso del Módulo Video - Extraer fotogramas

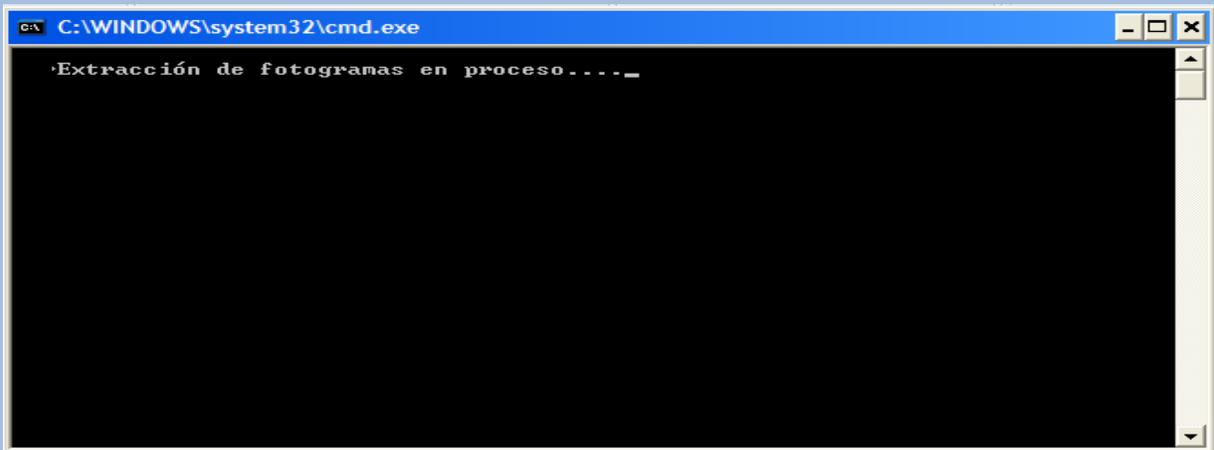
Caso de Uso:	Extraer fotogramas
Actores:	Proceso automático de Indexación
Resumen:	Se realiza la extracción de los fotogramas claves del material, con el fin de agilizar posteriores análisis.
Precondiciones:	Debe existir al menos un material en el directorio referenciado por el proceso automático de indexación.
Referencias	R6
Prioridad	Crítico

Flujo Normal de Eventos

Sección "Extraer fotogramas"

Acción del Actor	Respuesta del Sistema
1. Verifica que exista algún material para indexar.	
2. Manda a ejecutar la extracción.	2.1. Se realiza la extracción de los fotogramas, asociando estos a los tiempos del material. 2.2. Se guarda la ubicación del directorio que contiene los fotogramas en la BD. 2.3. Se transfiere el material indexado (Ver CU Transferir material).

Prototipo de Interfaz



Flujos Alternos



Acción del Actor		Respuesta del Sistema
<i>Prototipo de Interfaz</i>		
Poscondiciones	Se obtiene un directorio asociado al material donde se encuentran los fotogramas claves del mismo.	

Tabla 11: Caso de Uso del Módulo Video - Extraer Fotogramas

Anexo 8: Caso de Uso del Módulo Video - Generar Índices.

Caso de Uso:	Generar Índices	
Actores:	Proceso automático de Indexación.	
Resumen:	Realiza marcas virtuales en un material atendiendo a intervalos de tiempos.	
Precondiciones:	Debe existir al menos un material en el directorio referenciado por el proceso automático de indexación.	
Referencias	R6	
Prioridad	Secundario	
Flujo Normal de Eventos		
Sección "Generar Índices"		
Acción del Actor		Respuesta del Sistema
1. Lee del servidor de BD los intervalos de tiempo a generar en los materiales especificados.		
2. Se manda la ejecución de la fragmentación virtual para crear una nueva sección que haga referencia al video original.		2.1. Se extrae un fragmento del material sin modificar el original y se guardan las referencias del fichero virtual obtenido en la BD. 2.2. Se transfiere el material original (Ver CU Transferir material).
<i>Prototipo de Interfaz</i>		
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
<i>Prototipo de Interfaz</i>		
Poscondiciones	Se obtiene un corte del video, atendiendo a las peticiones de otros sistemas.	

Tabla 12: Caso de Uso del Módulo Video - Generar Índices.



Anexo 9: Caso de Uso del Módulo Video - Autenticar Usuario

Caso de Uso:	Autenticar Usuario.
Actores:	Usuario de Captura.
Resumen:	El caso de uso se inicia cuando el actor introduce su usuario y contraseña para acceder al sistema, los mismos se verifican contra la base de datos. Finaliza cuando se habilita la entrada al usuario con los permisos asignados a este o se deniega su acceso.
Precondiciones:	
Referencias	R1
Prioridad	Secundario

Flujo Normal de Eventos

Sección "Autenticar Usuario"

Acción del Actor	Respuesta del Sistema
1. El actor inserta su nombre de usuario (A) y contraseña (B) en los campos correspondientes para acceder al sistema.	
2. El actor presiona el botón Aceptar (C), para acceder al sistema.	2.1. El sistema busca en la base de datos el nombre del usuario. 2.2. Compara la contraseña correspondiente a ese usuario. 2.3. Se buscan los permisos a los que tiene acceso el actor. 2.4. Se le permite el acceso al sistema con los permisos que tiene asignado el rol al que pertenezca. 2.5. Se construye la interfaz principal de acuerdo a los permisos obtenidos.

Prototipo de Interfaz





Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1. Si el nombre del usuario introducido no existe en la base de datos o la contraseña proporcionada no coincide se muestra un mensaje de "Usuario o contraseña incorrecto". 2.2. Se ofrece la oportunidad de volver a introducir sus datos.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1. Si el usuario no tiene permisos para utilizar la aplicación se muestra un mensaje de "No tiene suficientes privilegios, contacte al administrador." 2.2. Se ofrece la oportunidad de volver a introducir sus datos.
Prototipo de Interfaz	
Poscondiciones	Se obtiene la autenticación de un usuario con privilegios de subir un material al servidor o hacer una copia de algún material en un directorio local.

Tabla 13: Caso de Uso del Módulo Video - Autenticar Usuario

Anexo 10: Caso de Uso del Módulo Audio - Autenticar Usuario

Caso de Uso:	Autenticar Usuario	
Actores:	Usuario de Audio	
Resumen:	Mediante este caso de uso el usuario puede acceder a las operaciones a las que tiene permiso en el sistema según su rol.	
Precondiciones:	-	
Referencias:	RF 9 [1]	
Prioridad:	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario introduce su nombre de usuario (A) y contraseña (B) y presiona aceptar (C).	1.1. El sistema busca el nombre de usuario. 1.2 Si existe el usuario entonces el sistema compara la contraseña. 1.3 Si son correctos los datos, el sistema le muestra al usuario las opciones a las que tiene acceso según su rol.	



Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.2. Si no existe el usuario el sistema emite un mensaje de error. 1.3. Si la contraseña es incorrecta se emite un mensaje de error.
Poscondiciones	El usuario accede a las operaciones a las que tiene asignadas según su rol dentro del sistema.

Tabla 14: Caso de Uso del Módulo Audio - Autenticar Usuario

Anexo 11: Caso de Uso del Módulo Audio - Gestionar Archivo

Caso de Uso:	Gestionar Archivo
Actores:	Usuario de audio
Resumen:	Mediante este caso de uso el usuario de audio puede subir ficheros al servidor, hacer copias de estos fuera del servidor o consultarlos.
Precondiciones:	Que el usuario esté autenticado, que el usuario pertenezca a un rol con permisos de gestión de archivos y que exista al menos un fichero almacenado en el servidor o alguno fuera de este en soporte digital que se desee añadir.
Referencias	RF 8 [1]
Prioridad	Crítico

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción de gestión de archivos en el menú Gestión de archivos de la ventana principal. Puede ser una de las siguientes opciones: a) Subir Archivo.	2. El sistema ejecuta alguna de las siguientes acciones: a) Si se decide subir fichero al servidor ir a la sección "Subir Archivo" b) Si se decide hacer copia de un fichero del servidor ir a la sección "Realizar Copia"



- b) Realizar Copia.
- c) Consultar Archivo.

c) Si se decide consultar archivo del servidor ir a la sección "Consultar archivo"

Prototipo de Interfaz

Subir Archivo

Archivo: ...

Información del Archivo	Destino
Título: <input type="text"/>	Servidor de Media: <input type="text" value="Item 1"/> ▾
Actividad: <input type="text"/>	<input type="button" value="Adicionar"/> <input type="text"/>
Fecha: <input type="text"/> <input type="button" value="📅"/>	

Realizar Copia

Parámetros de Búsqueda

Tipo de archivo <input checked="" type="radio"/> Audio <input type="radio"/> Transcripción	Origen <input type="checkbox"/> Grabado <input type="checkbox"/> Gestionado	Local <input type="text" value="Todos"/> ▾	Tipo de Actividad <input type="text" value="Todas"/> ▾
Duración Hora: <input type="text" value="0"/> ▾ Min: <input type="text" value="0"/> ▾ Seg: <input type="text" value="0"/> ▾	Transcripción <input type="checkbox"/> Terminada <input type="checkbox"/> Parcial	Entre fechas Desde: <input type="text"/> <input type="button" value="📅"/> Hasta: <input type="text"/> <input type="button" value="📅"/>	

Resultados

Destino ...

Transferencia al servidor:



Consultar Archivo

Parámetros de Búsqueda

Tipo de archivo

Audio Transcripción

Origen

Grabado Gestionado

Local

Todos ▼

Tipo de Actividad

Todas ▼

Duración

Hora: ▲ ▼ Min: ▲ ▼ Seg: ▲ ▼

Transcripción

Terminada Parcial

Entre fechas

Desde: 📅 Hasta: 📅

Resultados

Sección "Subir archivo"

Acción del Actor	Respuesta del Sistema
1. El usuario solicita la opción "Subir Archivo".	1.1. Se muestra una interfaz en la que se pide al usuario la dirección de origen del archivo que desea agregar al servidor.
2. El usuario selecciona la dirección de origen del fichero y acepta.	2.1. El sistema comprueba que el formato del fichero es mp3. 2.2. El sistema muestra detalles del fichero seleccionado como: tamaño del fichero y fecha y hora de la selección. 2.2. El sistema habilita un cuadro de texto para que se especifique el título del audio y habilita la lista de servidores de archivos disponibles para subir el fichero.
3. El usuario teclea un título para el archivo (no es obligatorio) y selecciona un servidor de archivos (no es obligatorio) y finalmente presiona el botón "Adicionar". <i>Nota: De no ser especificados por el usuario el título o el servidor de medias en ambos casos el sistema utiliza valores por defecto, en el caso del título emplea el nombre del fichero y en el caso de los servidores selecciona el primer servidor de la lista.</i>	3.1 El sistema inicia la transferencia del fichero al servidor seleccionado, y muestra el por ciento de transferencia del fichero hacia el servidor. 3.2 Una vez culminada satisfactoriamente la transferencia completa del fichero seleccionado hacia el servidor el sistema procede a registrar los datos primarios correspondientes al fichero de audio a través del CUS Registrar Datos (ver Caso de Uso Registrar Datos) y muestra un mensaje informando de la



	culminación satisfactoria de la operación.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>2.1. Si el archivo que se quiere subir es formato wav se transcodifica el mismo a este formato (Ver Caso de Uso Realizar Transcodificación) y continua con 2.2. En caso de no ser un fichero wav el sistema muestra un mensaje de error informando que el formato del fichero no es soportado por el sistema.</p> <p>3.1 En caso de no ser posible la transferencia de ficheros a servidor el sistema muestra un mensaje informando del error ocurrido.</p> <p>3.2 En caso de no ser posible el registro de los datos primarios del fichero el sistema muestra un mensaje informando del error ocurrido.</p>
Sección “Consultar Archivo”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción “Consultar Archivo”.	1.1 El sistema ofrece la posibilidad de filtrar los datos de los ficheros de audio almacenados en el servidor en cuanto a los datos: origen, transcripción, fecha, duración, tipo de actividad y local.
2. El usuario selecciona los filtros según los que desea buscar los ficheros y presiona el botón filtrar.	2.1. El sistema muestra los ficheros del servidor que cumplen con los filtros de búsqueda establecidos por el usuario y ofrece la posibilidad de seleccionar uno de los ficheros resultantes de la búsqueda o de limpiar los resultados de la búsqueda a través del botón “Limpiar resultados”.
3. El usuario selecciona uno de los ficheros mostrados como resultado de la búsqueda.	3.1 En caso de haberse seleccionado uno de los resultados de la búsqueda el sistema habilita los siguientes botones: “Reproducir Audio” y “Ver Transcripción”, y en caso de haberse presionado el botón “Limpiar resultados”, el sistema borra todos los resultados de la búsqueda y establece los filtros de búsqueda en sus valores por defecto.



<p>4. El usuario presiona uno de los botones “Reproducir Audio” o “Ver Transcripción”</p>	<p>4.1 En el caso de “Reproducir Audio” el sistema mostrará una ventana que ofrecerá prestaciones básicas para la reproducción del audio seleccionado (Ver Caso de Uso Reproducir Audio)</p>
<p>Flujos Alternos</p>	
	<p>4.1 En el caso de “Ver Transcripción” el sistema mostrará una ventana con la transcripción del audio. <i>Nota: No permitirá edición</i></p>
<p>Sección “Realizar Copia”</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
<p>1. El usuario selecciona la opción “Realizar Copia”.</p>	<p>1.1 El sistema ofrece la posibilidad de filtrar los datos de los ficheros de audio almacenados en el servidor en cuanto a los datos: origen, transcripción, fecha, duración, tipo de actividad y local.</p>
<p>2. El usuario selecciona los filtros según los que desea buscar los ficheros y presiona el botón filtrar.</p>	<p>2.1. El sistema muestra los ficheros del servidor que cumplen con los filtros de búsqueda establecidos por el usuario y ofrece la posibilidad de seleccionar uno de los ficheros resultantes de la búsqueda o de limpiar los resultados de la búsqueda a través del botón “Limpiar resultados”.</p>
<p>3. El usuario selecciona uno de los ficheros mostrados como resultado de la búsqueda o presiona el botón “Limpiar resultados”.</p>	<p>3.1 En caso de haberse seleccionado uno de los resultados de la búsqueda el sistema habilita un cuadro de texto y su botón correspondiente para seleccionar el destino donde se desea crear la copia y también habilita el botón copiar, y en caso de haberse presionado el botón “Limpiar resultados”, el sistema borra todos los resultados de la búsqueda y establece los filtros de búsqueda en sus valores por defecto.</p>
<p>4. El usuario selecciona la carpeta destino deseada donde se creará la copia del fichero seleccionado y presiona el botón copiar.</p>	<p>4.1 El sistema inicia la copia del fichero seleccionado, y muestra el por ciento de copia del fichero a la carpeta seleccionada. 4.2 Una vez culminada satisfactoriamente la copia completa del fichero el sistema muestra un mensaje informando de la culminación satisfactoria de la operación.</p>
<p>Flujos Alternos</p>	



	4.1 En caso de no ser posible la copia de ficheros a la carpeta destino el sistema muestra un mensaje informando del error ocurrido.
Poscondiciones	Se realiza una copia de un fichero del servidor, se incluye un fichero de audio en este o se consulta un audio o transcripción, según la decisión del usuario, y estas operaciones son registradas.

Tabla 15: Caso de Uso del Módulo Audio - Gestionar Archivo

Anexo 12: Caso de Uso del Módulo Audio - Grabar Audio

Caso de Uso:	Grabar Audio
Actores:	Usuario de audio
Resumen:	Mediante este caso de uso usuario puede capturar audio desde la aplicación, utilizando como fuente la línea de entrada.
Precondiciones:	Que el usuario esté autenticado, que el usuario pertenezca a un rol con permisos de grabación de audio y que exista alguna actividad a grabar en la planificación de trabajo o se necesite realizar una grabación en un momento determinado.
Referencias	R 1 [1]
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario puede seleccionar en el Menú Grabación las opciones siguientes: <ul style="list-style-type: none"> a) Consultar planificaciones. b) Grabar audio. 	1.1 El sistema puede realizar las siguientes acciones: <ul style="list-style-type: none"> a) Si el usuario selecciona la opción “Consultar planificaciones”, ver sección “Consultar Planificación” b) Si el usuario selecciona la opción “Grabar Audio”, ver sección “Grabar Audio”.
Prototipo de Interfaz	



Consultar Planificaciones

Parámetros de Búsqueda

Hora
Desde: 0 0 Hasta: 0 0 Local: Todos

Tipo de Actividad: Todas Entre fechas: Desde: Hasta:

Buscar

Resultados

Cancelar Atender

Datos de la Grabación

Nombre: Reunión Ordinario Octubre
Local: Salón 1
Actividad: Reunión de Brigada
Aceptar

Controles

Destino: Comenzar Pausar Terminar

Sonido
Volumen: Servidor: Lista de Servidores

Información del Sistema:
Espacio disponible: 12.3 Gb / 80 Gb
Estado: Grabando

Información
Creación de archivo WAV
Transcodificación a MP3
Transferencia al Servidor

Sección "Consultar planificación"

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción "Consultar Planificación".	1.1 El sistema muestra una interfaz para seleccionar los filtros de la planificación que desea atender. El sistema muestra una interfaz que ofrece la posibilidad de filtrar los datos de las planificaciones de grabación de audio en cuanto a los datos: fecha,



	hora, tipo de actividad y local.
2. El usuario establece los filtros que considera y presiona el botón “Filtrar”.	2.1 El sistema muestra las planificaciones que cumplen con los parámetros introducidos por el usuario.
3. El usuario selecciona una de las planificaciones obtenidas como resultado de la búsqueda y presiona el botón “Atender”.	3.1 El sistema muestra la interfaz de grabación de audio, que muestra los datos de la Planificación seleccionada (Nombre, Hora de inicio, Hora de Fin, Local y Actividad), muestra además Información sobre el sistema (Espacio Disponible y Estado). Además de dicha información el sistema ofrece la posibilidad de seleccionar una de dos opciones: “Grabación Automática” y “Grabación Manual”.
4. El usuario selecciona la opción “Grabación Automática”	4.1 Véase Caso de Uso Grabación Automática.
Flujos Alternos	
4. El usuario selecciona la opción “Grabación Manual”.	4.1 Véase Caso de Uso Grabación Manual.
Sección “Grabar Audio”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción “Grabar Audio”.	1.1 El sistema muestra la interfaz de Grabación de Audio, donde deben introducirse los siguientes datos sobre la grabación a realizar: Nombre, Local y Actividad.
2. El usuario introduce los datos requeridos (Nombre, Local y Actividad) y presiona el botón “Aceptar”.	2.1 El sistema verifica que los datos introducidos sean válidos, y en caso de serlo habilita el resto de los controles de la interfaz y deshabilita los controles que recogen los datos introducidos por el usuario. Una vez habilitados los controles el sistema continúa según establece el Caso de Uso Grabación Manual.
Flujos Alternos	
	2.1 En caso de que los datos no sean válidos el sistema muestra un mensaje informando de que los datos no son válidos
Postcondiciones	Se establecen los datos necesarios para comenzar una grabación, ya sea automática o manual.

Tabla 16: Caso de Uso del Módulo Audio - Grabar Audio



Anexo 13: Caso de Uso del Módulo Audio - Grabación Automática

Caso de Uso:	Grabación Automática
Actores:	Usuario de audio
Resumen:	Mediante este caso de uso usuario puede capturar audio desde la aplicación, a partir de una planificación existente es decir de forma automática se iniciará la captura de audio, teniendo como datos aquellos establecidos en la planificación.
Precondiciones:	Que el usuario esté autenticado, que el usuario pertenezca a un rol con permisos de grabación de audio y que exista alguna planificación de grabación de audio.
Referencias	R 1 [1]
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción “Grabación Automática”.	1.1 El sistema muestra una interfaz que permite la selección de dos opciones: “Comenzar Automáticamente” y “Finalizar Automáticamente” y ofrece además un grupo de controles para controlar la captura de audio(cuadro de texto para especificar destino del fichero capturado aparece con un valor por defecto, una lista de servidores disponibles (servidor donde será almacenado, aparece un servidor por defecto), botón explorar para cambiar la carpeta destino (habilitado), botones comenzar (deshabilitado), pausar (deshabilitado) y terminar (deshabilitado) y control de volumen y opción mute (ambos habilitados)).
2. El usuario desmarca solo la opción “Finalizar Automáticamente” y presiona el botón Aceptar.	2.1 El sistema habilita el botón pausar y terminar, deshabilita la selección de opciones y establece en el Panel de Información del sistema el Estado: “En espera”, que muestra entre paréntesis una cuenta regresiva del tiempo (HH:MM:SS) restante para que se inicie la grabación de forma automática. En el momento en que el tiempo restante sea (00:00:00) el sistema inicia la captura de audio desde la línea de entrada, estableciendo como Estado en el Panel de Información del Sistema el Estado: “Grabando”, que



	muestra entre paréntesis el tamaño del fichero creado a partir del audio capturado en la carpeta destino especificada y el tiempo transcurrido desde el inicio de la grabación en el formato HH:MM: SS.
3. El usuario presiona el botón Pausar.	3.1 Véase la Sección Pausar/Continuar Grabación
4. El usuario presiona el botón Reiniciar.	4.1 Véase la Sección Pausar/Continuar Grabación
5. El usuario presiona el botón Terminar	5.1 Véase la sección Terminar Grabación.
Flujos Alternos	
2. El usuario desmarca solo la opción “Comenzar Automáticamente” y presiona el botón Aceptar.	2.1 El sistema habilita el botón pausar y comenzar, deshabilita la selección de opciones y el botón terminar y establece en el Panel de Información del sistema el Estado: “En espera”, que muestra entre paréntesis el tiempo (HH:MM:SS) con los valores (00:00:00). Véase la Sección Comenzar Grabación y Pausar/Continuar.
2. El usuario desmarca ambas opciones: “Comenzar Automáticamente” y “Finalizar Automáticamente” y presiona el botón Aceptar.	2.1 El sistema habilita el botón pausar, comenzar, terminar y deshabilita la selección de opciones y el botón terminar y establece en el Panel de Información del sistema el Estado: “En espera”. Véanse las secciones Comenzar Grabación y Pausar/Continuar y Terminar Grabación.
2. El usuario presiona el botón Aceptar.	2.1 El sistema habilita el botón pausar y deshabilita el resto de los controles, deshabilita la selección de opciones y establece en el Panel de Información del sistema el Estado: “En espera”, que muestra entre paréntesis una cuenta regresiva del tiempo (HH:MM:SS) restante para que se inicie la grabación de forma automática. En el momento en que el tiempo restante sea (00:00:00) el sistema inicia la captura de audio desde la línea de entrada, estableciendo como Estado en el Panel de Información del Sistema el Estado: “Grabando”, que muestra entre paréntesis el tamaño del fichero creado a partir del audio capturado en la carpeta destino especificada y el tiempo transcurrido desde el inicio de la grabación en el formato HH:MM: SS. Cuando coincida el tiempo del



	sistema con la hora de fin de la planificación el sistema continúa según establece la sección Terminar Grabación.
2. El usuario presiona el botón Cancelar.	2.1 El sistema cierra la ventana activa.
3. El usuario presiona el botón Terminar.	3.1 Véase la sección Terminar Grabación.

Prototipo de Interfaz

Sección "Pausar/Continuar Grabación"

Acción del Actor	Respuesta del Sistema
1. El usuario presiona el botón Pausar.	1.1 El sistema establece en el Panel de Información del sistema el Estado: "En pausa", y cambia el texto del botón Pausar por "Continuar". Además el sistema deja de capturar audio al fichero creado.

Flujos Alternos

1. El usuario presiona el botón Continuar.	1.1 El sistema establece en el Panel de Información del sistema el Estado: "Grabando", y cambia el texto del botón Pausar por "Pausar". Además el sistema continúa la capturar audio al fichero creado.
--	---

Sección "Comenzar Grabación"

Acción del Actor	Respuesta del Sistema
1. El usuario presiona el botón Comenzar	1.1 El sistema inicia la captura de audio desde la línea de entrada, estableciendo como Estado en el Panel de Información del Sistema el Estado: "Grabando", que muestra entre paréntesis el tamaño



	del fichero creado a partir del audio capturado en la carpeta destino especificada y el tiempo transcurrido desde el inicio de la grabación en el formato HH:MM:SS.
Sección "Terminar grabación"	
Acción del Actor	Respuesta del Sistema
1. El usuario presiona el botón Terminar.	<p>1.1 El sistema crea un fichero WAV a partir del fichero capturado y muestra mediante la barra de progreso "Creación de Archivo WAV" el porcentaje de creación del fichero. En caso de ocurrir algún error el sistema muestra una ventana informando del error ocurrido.</p> <p>1.2 Una vez culminado este proceso el sistema crea un fichero MP3 a partir del fichero WAV obtenido en el paso anterior y muestra mediante la barra de progreso "Transcodificación a MP3" el porcentaje de creación del fichero. En caso de ocurrir algún error el sistema muestra una ventana informando del error ocurrido.</p> <p>1.3 Una vez culminado el paso anterior el sistema inicia la transferencia del fichero MP3 al servidor seleccionado y muestra mediante la barra de progreso "Transferencia al Servidor" el porcentaje de creación del fichero. En caso de ocurrir algún error el sistema muestra una ventana informando del error ocurrido.</p>
Poscondiciones	Se almacena en el servidor un fichero de audio MP3 correspondiente a la señal de audio recibida por la línea de entrada.

Tabla 17: Caso de Uso del Módulo Audio - Grabación Automática

Anexo 14: Caso de Uso del Módulo Audio - Grabación Manual

Caso de Uso:	Grabación Manual
Actores:	Usuario de audio
Resumen:	Mediante este caso de uso usuario puede capturar audio desde la aplicación, teniendo como datos aquellos establecerá en el momento en que decida capturar audio.
Precondiciones:	Que el usuario esté autenticado y pertenezca a un rol con permisos de grabación de



	audio
Referencias	R 1 [1]
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción “Grabar Audio”.	1.1 El sistema muestra una interfaz que permite al usuario establecer los valores iniciales de la grabación (Nombre, Local y Actividad). El resto de los componentes de la interfaz aparecen deshabilitados.
2. El usuario presiona el botón Aceptar	2.1 El sistema habilita los Controles de Grabación, permitiendo al usuario la selección de la carpeta destino y del Servidor (en caso de que el usuario no establezca estos elementos ambos toman valores por defecto)
3. El usuario presiona el botón Comenzar	3.1 Véase la sección Comenzar Grabación.
4. El usuario presiona el botón Pausar	4.1 Véase la sección Pausar/Continuar Grabación
5. El usuario presiona el botón Terminar	5.1 Véase la sección Terminar Grabación.

Flujos Alternos

	2.1 En caso de que los datos proporcionados por el usuario no sean válidos el sistema muestra un mensaje informando del error en la entrada de datos.
--	---

Prototipo de Interfaz

Sección “Pausar/Continuar Grabación”

Acción del Actor	Respuesta del Sistema
1. El usuario presiona el botón Pausar.	1.1 El sistema establece en el Panel de Información



	del sistema el Estado: “En pausa”, y cambia el texto del botón Pausar por “Continuar”. Además el sistema deja de capturar audio al fichero creado.
Flujos Alternos	
1. El usuario presiona el botón Continuar.	1.1 El sistema establece en el Panel de Información del sistema el Estado: “Grabando”, y cambia el texto del botón Pausar por “Pausar”. Además el sistema continúa capturando el audio al fichero creado.
Sección “Comenzar Grabación”	
Acción del Actor	Respuesta del Sistema
1. El usuario presiona el botón Comenzar	1.1 El sistema inicia la captura de audio desde la línea de entrada, estableciendo como Estado en el Panel de Información del Sistema el Estado: “Grabando”, que muestra entre paréntesis el tamaño del fichero creado a partir del audio capturado en la carpeta destino especificada y el tiempo transcurrido desde el inicio de la grabación en el formato HH:MM:SS.
Sección “Terminar grabación”	
Acción del Actor	Respuesta del Sistema
1. El usuario presiona el botón Terminar.	1.1 El sistema crea un fichero WAV a partir del fichero capturado y muestra mediante la barra de progreso “Creación de Archivo WAV” el porcentaje de creación del fichero. En caso de ocurrir algún error el sistema muestra una ventana informando del error ocurrido. 1.2 Una vez culminado este proceso el sistema crea un fichero MP3 a partir del fichero WAV obtenido en el paso anterior y muestra mediante la barra de progreso “Transcodificación a MP3” el porcentaje de creación del fichero. En caso de ocurrir algún error el sistema muestra una ventana informando del error ocurrido. 1.3 Una vez culminado el paso anterior el sistema inicia la transferencia del fichero MP3 al servidor seleccionado y muestra mediante la barra de progreso “Transferencia al Servidor” el porcentaje de creación del fichero. En caso de ocurrir algún error el sistema



	muestra una ventana informando del error ocurrido.
Poscondiciones	Se almacena en el servidor un fichero de audio MP3 correspondiente a la señal de audio recibida por la línea de entrada.

Tabla 18: Caso de Uso del Módulo Audio - Grabación Manual

Anexo 15: Caso de Uso del Módulo Audio - Reproducir Audio

Caso de Uso:	Reproducir Audio	
Actores:	Usuario de audio	
Resumen:	Mediante este caso de uso se puede reproducir un fichero almacenado en el servidor de archivos.	
Precondiciones:	Que el usuario esté autenticado, pertenezca a un rol con permisos de reproducción de Audio e inicie el caso de uso Gestionar Archivo en su Sección Consultar Archivo o el Caso de Uso Transcribir Audio.	
Referencias	R 4 [1], R5 [1]	
Prioridad	Crítico	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El usuario presiona el botón Reproducir Audio.	2. El sistema muestra una interfaz que permite las siguientes opciones básicas de un reproductor de audio: pausa, stop, play, adelantar, y retrasar reproducción. Operaciones que podrán realizarse mediante control por teclado.
Prototipo de Interfaz		
Poscondiciones	El audio es reproducido para consultas realizadas por el usuario.	

Tabla 19: Caso de Uso del Módulo Audio - Reproducir Audio.



Anexo 16: Caso de Uso del Módulo Audio - Transcribir Audio

Caso de Uso:	Transcribir Audio	
Actores:	Usuario de audio	
Resumen:	Mediante este caso de uso usuario de audio puede transcribir audio desde la aplicación.	
Precondiciones:	Que el usuario esté autenticado, pertenezca a un rol con permisos de transcripción y que exista audio no transcrito en el servidor.	
Referencias	R 4, R 7, R 5 [1]	
Prioridad	Crítico	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El usuario selecciona la opción “Transcribir Audio”.	1.1 El sistema ofrece la posibilidad de filtrar los datos de los ficheros de audio almacenados en el servidor en cuanto a los datos: origen, transcripción, fecha, duración, tipo de actividad y local.
	2. El usuario selecciona los filtros según los que desea buscar los ficheros y presiona el botón filtrar.	2.1. El sistema muestra los ficheros del servidor que cumplen con los filtros de búsqueda establecidos por el usuario y ofrece la posibilidad de seleccionar uno de los ficheros resultantes de la búsqueda o de limpiar los resultados de la búsqueda a través del botón “Limpiar resultados”.
	3. El usuario selecciona uno de los ficheros mostrados como resultado de la búsqueda.	3.1 En caso de haberse seleccionado uno de los resultados de la búsqueda el sistema habilita el botón: “Transcribir Audio” y en caso de haberse presionado el botón “Limpiar resultados”, el sistema borra todos los resultados de la búsqueda y establece los filtros de búsqueda en sus valores por defecto.
	4. El usuario presiona el botón “Transcribir Audio”.	4.1 El sistema muestra una interfaz con un editor de texto para transcribir y un reproductor de audio para escuchar el audio que se transcribe.



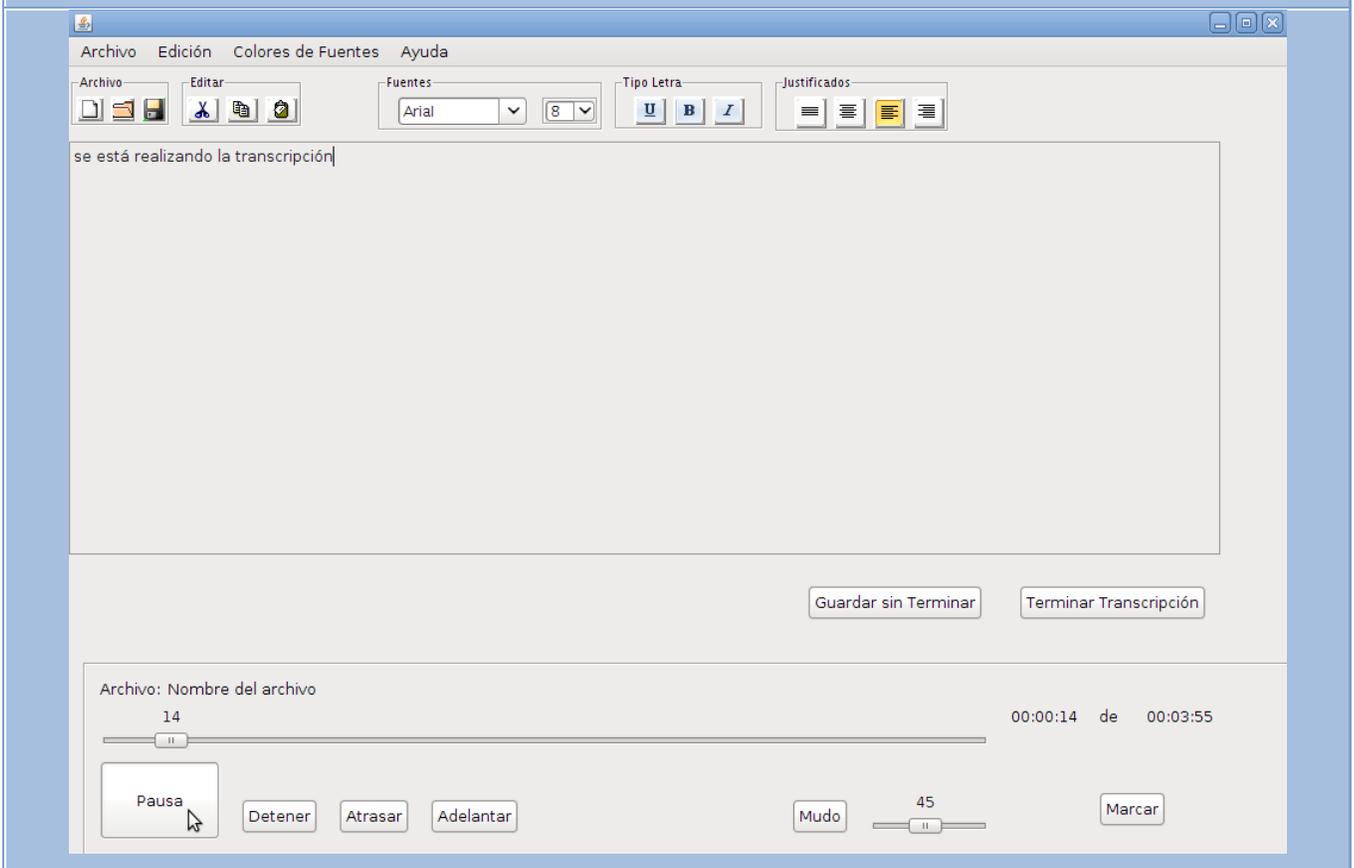
5. El usuario de audio puede realizar las siguientes operaciones:

- a) Elegir guardar la transcripción sin completarla.
- b) Guardar la transcripción completa marcando la opción Terminada.

5.1. El sistema puede ejecutar las siguientes acciones:

- a) Si el usuario decide guardar la transcripción como incompleta el sistema guarda la transcripción hecha hasta el momento identificándola como “no completada”.
- b) Si el usuario decide guardar la transcripción como completa el sistema la guarda y es marcada como “Terminada”.

Prototipo de Interfaz



Poscondiciones

Se obtiene el fichero de transcripción terminada o no del audio correspondiente.

Tabla 20: Caso de Uso del Módulo Audio - Transcribir Audio