



TÍTULO: Plataforma de codificación e indexación de video para el Sistema de Captura y Catalogación de Medias

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

AUTOR: Beny Cabrera Martínez – Pinillos

TUTOR: Ing. Jean Michael Suárez Pérez

DEDICATORIA

A mis padres que son el faro que guía mis pasos.

AGRADECIMIENTOS

Quiero agradecer en primer lugar al “Compañero Fidel” por haber creado esta escuela que me ha dado la oportunidad de realizarme como persona y como profesional.

A la dirección de la universidad por haberme dado una segunda oportunidad o de otra forma no estaría escribiendo estos agradecimientos.

A toda mi familia por su apoyo y preocupación y especialmente a mi mamá y mi papá porque han sido mi guía, mi orgullo y mi inspiración y de no ser por ellos no hubiese llegado hasta aquí.

A todos mis profes por transmitirme sus valores y conocimientos que me han formado como profesional.

A todos mis compañeros de los grupos por los que he pasado por soportarme durante todos estos años.

A mi tutor por haberme brindado sus conocimientos y haber confiado en mí.

Y en general a todos aquellos que de una forma u otra han contribuido a mi formación.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Señales Digitales de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autor: Beny Cabrera Martínez – Pinillos

Tutor: Ing. Jean Michael Suárez Pérez

DATOS DE CONTACTO

Tutor: Ing. Jean Michael Suárez Pérez

E-mail: jmsuarez@uci.cu

- Ingeniero en Ciencias Informáticas, Universidad de Ciencias Informáticas, 2009.
- Diploma de Oro.
- Premio Mella.
- Vinculado al Proyecto Productivo UCITeVe y SCCM durante los últimos 3 años de estudios universitarios.
- Líder de Proyecto SCCM (Sistema de Captura y Catalogación de Medias).
- Profesor del Departamento de Señales Digitales, Facultad 6.

OPINIÓN DEL TUTOR

<En este acápite se incluye la opinión del tutor sobre el trabajo de diploma>

RESUMEN

Actualmente la Universidad de las Ciencias Informáticas (UCI) cuenta con soluciones para automatizar los procesos de codificación e indexación de videos. Estos procesos son una pieza fundamental en el flujo de trabajo del Sistema de Captura y Catalogación de Medias, proyecto que se desarrolla actualmente en la universidad, pero los mismos son ineficientes y no son escalables, además de carecer de cualquier tipo de control de flujo y de mecanismos para balancear las cargas de procesamiento.

El presente trabajo de diploma aborda la investigación de los procesos de codificación e indexación de video con tecnologías libres, con el objetivo de desarrollar una plataforma que realice los procesos antes mencionados para el Sistema de Captura y Catalogación de Medias. El desarrollo de la aplicación se llevó a cabo con la utilización de técnicas de programación multihilos aplicando la arquitectura de dos capas y la implementación de plugins. En este documento se recogen las principales funcionalidades del sistema y se muestran las dependencias de los componentes que la integran a través del modelo de implementación. Se describen las herramientas utilizadas y se presentan las pruebas realizadas al sistema.

PALABRAS CLAVES

Codificación, Indexación, Video, Audio, Plugins

ABSTRACT

Currently, the University of Informatics Sciences (UCI) has solutions for automating coding and indexing of videos. These processes are a fundamental part of the workflow of the system for capture and cataloging media files, a project currently underway in the university, but these are inefficient and not scalable, in addition to lacking any type of flow control and mechanisms to balance processing loads

The present dissertation discusses the investigation of the processes of coding and indexing video on open source technologies with the aim of developing a platform of coding and indexing video for the system for capturing and cataloging media files. The application development was performed with the use of multithreading programming techniques applying the architecture based on plugins. On this document sets out the main functionalities of the mentioned processes and shows the dependencies of the components that comprise them through the implementation model. It describes the tools used and shows the tests made to the system.

KEYWORDS

Coding, Indexing, Video, Audio, Plugins

Índice de Tablas

Tabla 1. Descripción de Requisitos Funcionales del Sistema	23
Tabla 2. Software para Estaciones Cliente.....	24
Tabla 3. Software para Servidores.....	24
Tabla 5. Hardware para Servidores	25
Tabla 6. Descripción de Actores del Sistema.....	25
Tabla 7. Descripción Textual del CU Analizar Video.....	27
Tabla 8. Descripción Textual del CU Codificar Video	29
Tabla 9. Descripción Textual del CU Elaborar Resumen Visual de Video	30
Tabla 10. DCP Caja Negra. CU Analizar Video	40
Tabla 11. Descripción de Variables del CU Analizar Video	40
Tabla 12. Resultados de la Prueba al CU Analizar Video	40
Tabla 13. DCP Caja Negra del CU Codificar Video	41
Tabla 14. Descripción de Variables del CU Codificar Video	42
Tabla 15. Resultados de la Prueba al CU Codificar Video.....	42
Tabla 16. DCP Caja Negra. CU Elaborar Resumen Visual de Video.....	43
Tabla 17. Descripción de Variables del CU Elaborar Resumen Visual de Video	44
Tabla 18. Resultados de la Prueba al CU Elaborar Resumen Visual de Video.....	44
Tabla 19. Descripción Textual del CU Reportar Nodo.....	50
Tabla 20. Descripción Textual del CU Actualizar Estado Plataforma.....	51
Tabla 21. Descripción Textual del CU Cancelar Proceso	52

Índice de Figuras

Figura 1. Flujo de eventos	8
Figura 2. Modelo de Dominio	21
Figura 3. Diagrama de Casos de Uso del Sistema	26
Figura 4. Diagrama de Clases de Diseño: Plataforma de Codificación e Indexación.....	32
Figura 5. Diagrama de Clases de Diseño: Análisis	33
Figura 6. Diagrama de Clases de Diseño: Plugin de Análisis.....	33
Figura 7. Diagrama de Clases de Diseño: Codificación	34
Figura 8. Diagrama de Clases de Diseño: Plugin de Codificación.....	34
Figura 9. Diagrama de Clases de Diseño: Indexación.....	35
Figura 10. Diagrama de Clases de Diseño: Plugin de Indexación.....	35
Figura 11. Diagrama de Despliegue	37
Figura 12. Diagrama de Componentes	38

Índice

Introducción	1
CAPÍTULO 1: Fundamentación Teórica	5
1.1. Introducción	5
1.2. Conceptos asociados al dominio del problema	5
1.3. Objeto de Estudio	6
1.3.1. Descripción General	6
1.3.2. Descripción actual del dominio del problema.....	7
1.3.3. Situación Problemática.....	8
1.4. Análisis de otras soluciones existentes	9
1.4.1. Aplicaciones Tedral	9
1.4.2. Aplicaciones Harmonic.....	10
1.5. Conclusiones	11
CAPÍTULO 2: Tendencias y tecnologías actuales.....	12
2.1. Introducción	12
2.2. Metodología de Desarrollo	12
2.3. Lenguaje de Modelado.....	14
2.4. Herramienta Case.....	15
2.5. Lenguaje de Programación	16
2.6. Framework de Desarrollo	17
2.7. Entorno de Desarrollo Integrado	18
2.8. Conclusiones	18
CAPÍTULO 3: Presentación de la solución propuesta.....	20
3.1. Introducción	20
3.2. Modelo de Dominio	20
3.2.1. Descripción General del Modelo de Dominio.....	21
3.2.2. Diagrama de clases del Modelo de Dominio.....	21
3.2.3. Descripción de Clases.....	22
3.3. Requisitos Funcionales	23
3.4. Requisitos No Funcionales.....	23

3.4.1. Fiabilidad.....	23
3.4.2. Eficiencia.....	24
3.4.3. Software.....	24
3.4.4. Hardware.....	25
3.4.5. Interfaz.....	25
3.5. Descripción del Sistema Propuesto.....	25
3.5.1. Descripción de los actores.....	25
3.5.2. Casos de Uso del Sistema.....	26
3.6. Conclusiones.....	31
CAPÍTULO 4. Construcción de la solución propuesta.....	32
4.1. Introducción.....	32
4.2. Diagrama de Clases.....	32
4.2.1. Plataforma de Codificación e Indexación de Video.....	32
4.2.2. Análisis de Video.....	33
4.2.3. Codificación de Video.....	34
4.2.4. Indexación de Video.....	35
4.3. Generalidades de la implementación.....	35
4.4. Modelo de Despliegue.....	37
4.5. Modelo de Implementación.....	38
4.6. Prueba del sistema propuesto.....	38
4.6.1. Diseño de Caso de Prueba Caja Negra. Partición equivalente.....	39
4.7. Conclusiones.....	44
Conclusiones.....	45
Recomendaciones.....	46
Bibliografía.....	47
Anexos.....	50
Anexo I. Descripción del Caso de Uso Reportar Nodo.....	50
Anexo II. Descripción del Caso de Uso Actualizar Estado Plataforma.....	50
Anexo III. Descripción del Caso de Uso Cancelar Proceso.....	51
Glosario.....	53

Introducción

Con el desarrollo de los medios de comunicación y la necesidad de almacenamiento de materiales audiovisuales surge el problema de organizar y catalogar los mismos de forma eficiente, con el objetivo de poder acceder a estos en cualquier momento. Cuando crece el volumen de materiales almacenados se hacen necesarios mecanismos capaces de localizarlos no solo por el identificador del mismo, sino también por características inherentes al audiovisual como pudieran ser: tema principal, personas importantes que intervienen, entre otros (1). Muchas de las empresas necesitadas de almacenar información audiovisual, cuentan con aplicaciones que automatizan los procesos de planificación de las capturas y codificación de las mismas al formato acordado por la institución para ser guardado en el servidor donde se almacenará.

“Es evidente que el panorama actual de los archivos audiovisuales es muy complejo: primero, porque ya existe un amplio abanico de instituciones que trabajan en ellos dentro de la más clara acepción de espacio facilitador para el acceso a documentos audiovisuales, mediante la gestión de actividades de acopio y almacenamiento, registro o catalogación, conservación, promoción y acceso; y segundo, porque de modo tecnológico los archivos evolucionan constantemente a medida que se multiplican los soportes virtuales de registro digital y de sus canales de distribución como el cable, el satélite e Internet, mucho más allá de los soportes materiales clásicos de la cinta, el disco o el casete” (2).

Los sistemas de digitalización de archivos audiovisuales existentes en el mundo necesitan transmitir y recuperar los materiales almacenados, para lo cual deben ser codificados a un formato comprimido que permita su transmisión vía streaming y su distribución en cualquiera de los soportes digitales existentes. Puede decirse que la transcodificación e indexación de medias es uno de los procesos fundamentales de estos sistemas dada su utilidad para la recuperación de materiales. En Cuba, específicamente en la Universidad de las Ciencias Informáticas (UCI), se han implementando soluciones para dar respuesta a las problemáticas antes planteadas, ya que los sistemas existentes son privativos.

Las soluciones implementadas en el Departamento de Señales Digitales de la UCI carecen de un manejo eficiente de los recursos de hardware, así como la escalabilidad necesaria para adaptarse a los constantes cambios en el mundo de los archivos digitales. Por ello este departamento se dio a la tarea de identificar los procesos relacionados con la codificación e indexación de materiales en un sistema de

Introducción

digitalización de archivos audiovisuales, determinando como **problema a resolver** ¿Cómo ejecutar de forma automática los procesos de codificación e indexación de video con tecnologías libres? Se plantea como **objetivo general** de la investigación “*desarrollar una plataforma de codificación e indexación de video sobre tecnologías libres para el Sistema de Captura y Catalogación de Medias*”.

Para dar solución al problema planteado se hace necesario establecer como **objeto de estudio** “los procesos de codificación de video, técnicas de indexación de video y desarrollo de software orientado a plugins”, y el **campo de acción** “la automatización de los procesos de codificación e indexación de video dentro del Sistema de Captura y Catalogación de Medias”. Se plantea como **idea a defender**: el desarrollo de una plataforma de codificación e indexación de video sobre tecnologías libres para el Sistema de Captura y Catalogación de Medias, logrará la ejecución automática de los procesos de codificación e indexación de video.

Para dar cumplimiento al objetivo planteado se trazaron las siguientes **tareas de investigación**:

1. Caracterización de los procesos de codificación e indexación de video.
2. Determinación de los formatos de codificación y técnicas de indexación que necesita el Sistema de Captura y Catalogación de Medias.
3. Descripción del funcionamiento de aplicaciones servidoras para la codificación e indexación de video.
4. Caracterización de los modelos arquitectónicos basados en plugins.
5. Realización de la modelación del negocio.
6. Realización del levantamiento de requisitos funcionales y no funcionales.
7. Realización del Modelo de Casos de Uso del Sistema de la Plataforma de codificación e indexación de video.
8. Realización del Diseño de la Plataforma de codificación e indexación de video.
9. Implementación de la Plataforma de codificación e indexación de video.
10. Realización de las pruebas de caja negra a la Plataforma de codificación e indexación de video.

Introducción

Para dar cumplimiento a las tareas propuestas se hizo necesaria la utilización de los **métodos científicos de investigación** siguientes:

Métodos Teóricos:

Histórico – lógico: Se utilizó para hacer un estudio de la evolución de los procesos de codificación e indexación de video, así como las posibles tendencias a seguir.

Modelación: Se utilizó para lograr una mejor comprensión de los procesos y lineamientos de desarrollo a seguir en cada una de las fases, a través de la representación en diagramas de los procesos de codificación e indexación de video, utilizando para el modelado de la aplicación el Proceso Unificado de Desarrollo (RUP) y el Lenguaje Unificado de Modelado (UML).

Analítico – sintético: Se utilizó para concretar los elementos esenciales necesarios para el desarrollo de la plataforma de codificación e indexación de video. Se realizó un estudio de las soluciones existentes en el mercado, así como de las necesidades reales de una televisora.

Métodos Empíricos:

Observación: Se utilizó para visualizar el comportamiento de los diferentes métodos de codificación de video y determinar el más adecuado para el logro del objetivo planteado.

El presente trabajo de diploma ha sido estructurado en cuatro capítulos:

Capítulo 1: Fundamentación teórica.

Se hace una descripción de los procesos de codificación e indexación de video profundizando en los aspectos relacionados con la situación problemática identificada. Además se analizan algunas de las soluciones existentes en el mundo semejantes a la solución que se plantea en esta investigación.

Capítulo 2: Tendencias y tecnologías actuales.

En este capítulo se exponen las tecnologías, herramientas y metodologías utilizadas en el mundo para la construcción de aplicaciones de escritorio utilizadas en la construcción de la solución que se propone.

Introducción

Capítulo 3: Presentación de la solución propuesta.

En este capítulo se reflejan los aspectos centrales de la modelación del negocio. Luego se explican los requisitos funcionales y no funcionales del sistema y se presenta el diagrama de casos de uso del sistema con la descripción de los casos de uso considerados como críticos.

Capítulo 4: Construcción de la solución propuesta.

En este capítulo se abordan temas fundamentales para la construcción del sistema propuesto. Se recogen los diagramas de clases y principios del diseño, así como las generalidades del Flujo de Trabajo de Implementación. Se presentan además, las pruebas realizadas al sistema desarrollado.

CAPÍTULO 1: Fundamentación Teórica.

1.1. Introducción

En este capítulo se hace una descripción de los procesos de codificación e indexación de video profundizando en los aspectos relacionados con la situación problemática identificada. Además se analizan algunas de las soluciones existentes en el mundo semejantes a la solución que se plantea en esta investigación.

1.2. Conceptos asociados al dominio del problema

La definición de los conceptos fundamentales a utilizar en este trabajo constituye una prioridad para lograr la comprensión de los temas a tratar, por lo que fue necesario elaborar la definición operacional de algunos de ellos a partir del análisis de diversos significados abordados en distintas fuentes¹. A continuación se relacionan dichos conceptos:

Formato de compresión de video y sonido digital: Forma de presentación de la información del video y el sonido digital en menor espacio que en su estado original sin comprimir. La característica principal de estos formatos es la compresión con pérdida de calidad e información, aunque generalmente no es perceptible al ojo y el oído humano. Luego de comprimido un fichero nunca se llegará a la información inicial que contenía.

Códec de video: Es un conjunto de algoritmos utilizados para la **codificación** y/o **decodificación** de datos. El proceso de codificación se basa en la compresión del archivo, mientras la decodificación se hace al mismo tiempo que se reproduce el archivo. Los algoritmos de codificación de video están diseñados principalmente para la reducción del tamaño respetando al máximo la calidad del material. En función de

¹ Se consultaron a tal efecto:

- Servicios Informáticos. [En línea] [Citado el: 29 de 11 de 2010.] <http://www.ctisa.com/diccionario.asp>. B60920501.
- **Suárez Pérez, Jean Michael y Estrada Velazco, Aylin.** *Sistema de Captura e Indexación de Video*. [Trabajo de diploma para optar por el título de Ingeniero en Informática] s.l. : Universidad de las Ciencias Informáticas, 2009.
- Glosario Electrotécnico y Electrónico. [En línea] [Citado el: 8 de 12 de 2010.] http://www.natureduca.com/tecno_gloselec_d01.php.
- Mundo DIVX Martes. [En línea] [Citado el: 02 de 12 de 2010.] <http://www.mundodivx.com/glosario.php>.
- **AXIS.** Técnicas de compresion. [En línea] 2004. [Citado el: 02 de 12 de 2010.] www.axis.com/es/documentacion/Tecnicas%20de%20compresion%20de%20video.pdf -.

Capítulo 1: Fundamentación teórica

esto y de la finalidad para la cual será codificado es definido el formato definitivo del fichero, e implícitamente, los algoritmos a utilizar en su compresión.

Codificación de video: La codificación o transcodificación, es la conversión de un sistema de datos a otro distinto. La codificación de video es la compresión de un video en un formato distinto del original o la conversión con uno u otro códec si se tratase del mismo formato.

Indexación de video: Técnicas que se aplican a un archivo audiovisual cuyo objetivo es adjuntar a una imagen o un vídeo una serie de descriptores del contenido, con el fin de medir la similitud con los descriptores correspondientes a una consulta. (3)

1.3. Objeto de Estudio

Para el desarrollo de la investigación se identificó como objeto de estudio: los procesos de codificación de video, técnicas de indexación de video y desarrollo de software orientado a plugins.

1.3.1. Descripción General

En el flujo de procesos de un sistema de digitalización de medias la codificación y la indexación son fundamentales dada la necesidad de la distribución de archivos audiovisuales para su posterior utilización. Estos procesos deben efectuarse de manera que se ajusten a nuevas técnicas, por ello se debe estudiar el mecanismo de desarrollo en plugins. Ello permitirá añadirle funcionalidades adicionales al sistema de tal manera que no habrá que modificar lo ya hecho, además permitirá hacer modificaciones sin afectar al sistema completo.

Partiendo del hecho que la información audiovisual requerida ha sido digitalizada se utilizan técnicas de indexación de video, estas permiten hacer búsquedas más precisas asociándole a cada archivo informaciones como: el tema del material, las personas involucradas, productores, directores, editores, etc; además, existen mecanismos capaces de detectar rostros, objetos, publicidad, entre otros. Hay otras técnicas de indexación que permiten transcribir el audio del archivo a texto añadiendo así más metadatos para la búsqueda, enfocándola en el tema hablado, frases específicas, expresiones de personas determinadas, entre otras.

Capítulo 1: Fundamentación teórica

La codificación de archivos audiovisuales se realiza con el objetivo de brindar la funcionalidad de transmisión de los mismos en una red de datos y su recuperación en formatos de codificación seleccionados por los clientes. La codificación también es necesaria para procesar las medias capturadas por el sistema, con el fin de almacenar los materiales utilizando formatos de alta calidad.

De entre las técnicas de indexación que pueden ser utilizadas en los sistemas de procesamiento audiovisual se encuentra la extracción de un resumen visual, la cual extrae de un video los fotogramas más representativos. El resumen visual es utilizado como base en la realización de la catalogación audiovisual.

1.3.2. Descripción actual del dominio del problema

El proceso de digitalización de un archivo se realiza almacenando el material en un perfil de alta calidad. Para la utilización de estos archivos de manera eficiente se deben extraer características inherentes al audiovisual, como puede ser un resumen visual del material o los logos que identifican el origen del mismo, entre otras. En el proceso de catalogación los materiales son enriquecidos con una mayor cantidad de información relacionada con su contenido en aras de lograr un proceso de búsqueda precisa. Estos archivos se codifican a una copia proxy para permitir a cualquier proceso de edición ser rápido y sencillo, y una vez concluido se realiza el proceso determinado en el archivo original, obteniéndose un material con buena calidad. Por otro lado los usuarios necesitan estos archivos en un formato acorde a las necesidades del proceso de producción y poder llevarse solo el fragmento requerido, para esto se utilizan técnicas de codificación y fragmentación de video (ver figura 1). Los sistemas que automatizan estos procesos son muy utilizados en emisoras de radio y televisión, empresas de video vigilancia, procesos judiciales, entre otros.



Figura 1. Flujo de eventos

1.3.3. Situación Problemática

La codificación e indexación de video son procesos fundamentales para el funcionamiento de un sistema de digitalización de archivos audiovisuales. Ellos posibilitan brindar la mayoría de las funcionalidades de dichos sistemas, por lo cual es necesario que los mismos se ejecuten de manera eficiente y en el menor tiempo posible. En Cuba, específicamente en el departamento de Señales Digitales de la UCI, se han implementando soluciones para dar respuesta a las problemáticas antes planteadas, ya que los sistemas existentes en el mundo son privativos. Para ello el propio departamento, particularmente el proyecto Sistema de Captura y Catalogación de Medias, ha venido trabajando en la solución de estos problemas, pero las soluciones desarrolladas no poseen un nivel de escalabilidad que permita incluir nuevas funcionalidades. Tampoco admite la adición de nuevos perfiles de codificación ni la optimización de los ya existentes. Hacer un cambio en estos sistemas implicaría cambiar el sistema completo, teniendo que ir

directamente al código de la aplicación. No tienen en cuenta la gestión eficiente de los recursos de hardware donde se desplegaría el sistema, repercutiendo en el funcionamiento esperado del mismo.

1.4. Análisis de otras soluciones existentes

En el mundo existen varias compañías que se han dedicado al desarrollo de aplicaciones para la codificación distribuida de materiales audiovisuales, así como su indexación. En este epígrafe se analizan las soluciones con mayor aceptación en el mercado con el propósito de identificar sus ventajas y desventajas y determinar las razones que no hacen posible su utilización.

1.4.1. Aplicaciones Tedral



Tedral es una compañía española que nace de la investigación universitaria con el objetivo de acercar la gestión de procesos de negocios a los activos multimedia, siendo de utilidad para cualquier organización que necesite archivar, buscar o usar grabaciones de vídeo o

audio. La empresa desarrolla sus propios y avanzados productos, estos interactúan con las tecnologías digitales existentes para aportar soluciones globales, fiables y rentables. La empresa produce aplicaciones informáticas para el tratamiento de contenido multimedia, entre ellas se encuentran TD MPM y TD Indexer. (4)

TD MPM es un sistema de gestión de flujos de procesos de media que permite definir, planificar y monitorizar las transferencias y transformaciones de media entre los servidores distribuidos y los sistemas. MPM es un sistema de software que optimiza y automatiza cualquier transferencia de media. Su integración de interfaz probada con equipos de terceros permite construir sistemas homogéneos e integrados usando la mayoría de los equipos existentes y manteniendo lo mejor de los procesos de trabajo del cliente.

TD Indexer es una librería, ofrece un amplio abanico de funcionalidades que permiten transformar, validar/conformar y extraer información del material audiovisual. Es capaz de integrarse con TD MPM ampliando las posibilidades de su aplicación a multitud de posibilidades de casos del mundo audiovisual. Dispone de funcionalidades como: reencapsulación y transcodificación de cualquier formato multimedia,

generación de Storyboard, búsqueda avanzada, conversión de audio a texto, asset virtuales y reconocimiento de patrones. Las funcionalidades conforman un conjunto de programas que analizan los archivos multimedia y extraen de manera automática información fundamental de acceso para catalogación y edición.

Tedial es una de las compañías líderes en el mercado de desarrollo de software para procesamiento de medias. Desarrolla sus productos utilizando licencias de software privativo lo cual constituye un impedimento para su utilización en el proyecto SCCM.

1.4.2. Aplicaciones Harmonic



Harmonic Inc es una compañía establecida en Estados Unidos, cuenta con una línea de productos denominada Rhozet, maneja completamente los procesos de transcodificación. Los productos de Rhozet proveen interfaces fáciles de aprender y de usar, componentes de salvas automáticas y un SDK basado en XML para el control absoluto de la programación de los procesos. Estos productos están diseñados para crecer de acuerdo a las necesidades del cliente, es decir, ofrece escalabilidad de acuerdo al aumento de los requerimientos. Posee además un sistema de control de la calidad automatizado con una herramienta interactiva para la verificación visual de fallas. Utiliza tecnología de alta velocidad, tecnología multi-hilos y tecnología multi-CPU para aumentar las velocidades de conversión. (5)

En la familia de productos Rhozet podemos encontrar a Carbon Coder y Carbon Server, el primero maneja las tareas de codificación y provee en las estaciones clientes funciones adicionales tales como: conversión estándar, inserción de logos, administración de derechos digitales, entre otras. El segundo trabaja en conjunto con el Rhozet Workflow System (WFS) o sistema de flujo de trabajo, el cual maneja una red de nodos compuestos por el motor de Carbon Coder (una granja de codificación), esto posibilita la realización de operaciones de gran volumen a una escala masiva.

Carbon Coder es una aplicación de codificación basada en ficheros, soportando la mayoría de los formatos (actualmente más de 40) y es actualizado constantemente para asimilar nuevos. Puede correr como aplicación en solitario o como parte de una granja de codificación bajo el control de Carbon Server o del WFS. Como parte del proceso de codificación, maneja un amplio rango de operaciones incluyendo

Capítulo 1: Fundamentación teórica

conversión SD/HD, PAL/NTSC, inserción de logos, corrección de color, extracción de Close Caption, entre otras. Puede ser utilizado a través de su interfaz de usuario o utilizando una API que trae incluida basada en XML. (5)

Una granja de codificación puede ser configurada a través de múltiples nodos de Carbon Coder controlados por uno o más Carbon Servers. Permite el procesamiento automático de un gran volumen de tareas de codificación controladas a través del servidor utilizando los nodos de Carbon Coder, al mismo tiempo administra la distribución de trabajos, su priorización, balanceo de carga, transferencias vía FTP, monitoreo de estados y notificaciones. La interacción con el software es a través de una aplicación administrativa que corre en una máquina servidora o a través de la API incluida en el producto. También es posible acceder a una aplicación web desde máquinas remotas. Para un ambiente de codificación más complejo que requiera un gran volumen, la integración de terceras partes o flujos de trabajos sofisticados, se utiliza el WFS.

Estos productos ofrecen la solución más completa para el problema planteado en esta investigación, pero el hecho de haber sido desarrollado en los Estados Unidos, lo cual lo ubica bajo las leyes del bloqueo, y su carácter privativo, impiden su utilización en el proyecto SCCM.

1.5. Conclusiones

En este capítulo se ha realizado una conceptualización de términos y procesos del dominio del problema. Se realizó un estudio sobre el estado actual de las aplicaciones en el mundo para codificación distribuida de materiales audiovisuales, así como su indexación. Dicho estudio arrojó como resultado la imposibilidad de utilizar las aplicaciones estudiadas dado su carácter privativo, pero permitió identificar funcionalidades que el sistema debe brindar. Se recoge además, la identificación de las causas por las cuales es necesario para el departamento de Señales Digitales del centro GEYSED² de la UCI, la creación de sistemas automatizados que posibiliten la codificación e indexación de medias.

² **GEYSED**: Centro de desarrollo de productos, servicios y soluciones informáticas para el procesamiento de Señales Digitales y la Geoinformática.

CAPÍTULO 2: Tendencias y tecnologías actuales.

2.1. Introducción

La salida al mercado de nuevas tecnologías de la información y las comunicaciones con muy poco tiempo entre una y otra, la informatización de la sociedad y la automatización de los procesos de empresas de todo tipo, ha obligado a la industria del software a establecer técnicas, metodologías y herramientas para lograr una aceleración en el proceso de desarrollo de soluciones informáticas. Estos instrumentos, al tiempo que aceleran el proceso de desarrollo, deben asegurar la calidad y la actualización de las soluciones así como su soporte. En este capítulo se exponen las tecnologías, herramientas y metodologías utilizadas en el mundo para la construcción de aplicaciones de escritorio utilizadas en la construcción de la solución que se propone.

2.2. Metodología de Desarrollo

El Proceso Unificado de Desarrollo (RUP) es un proceso para el desarrollo de un proyecto y define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Tiene tres características esenciales: está dirigido por casos de uso, orientan el proyecto a la importancia para el usuario y lo que este quiere; está centrado en la arquitectura, realiza la toma de decisiones e indica cómo tiene que ser construido el sistema y en qué orden; y es iterativo e incremental, divide el desarrollo en mini proyectos donde los casos de uso y la arquitectura cumplen su objetivo de manera más depurada.

Maneja 6 claves principales:

- **Adaptación de procesos**
- **Balancear prioridades**
- **Colaboración entre equipos**
- **Demostrar valor iterativamente**
- **Elevar el nivel de abstracción**
- **Enfocarse en la calidad**

Las actividades de RUP se centran en crear y mantener modelos utilizando UML. Es un proceso configurable ya que no existe un único proceso apropiado para todos los desarrollos. Se adapta tanto a

Capítulo 2: Tendencias y tecnologías actuales

grupos pequeños como a grandes organizaciones. Basándose en lo que se consideran las mejores prácticas de desarrollo de software, resulta apropiado para una amplia gama de proyectos y organizaciones.

Aplica 6 prácticas principales para el desarrollo de sistemas, estas son:

- **Desarrollo de software en forma iterativa**
- **Gestión de requerimientos**
- **Uso de arquitecturas basadas en componentes**
- **Modelización visual del software**
- **Verificación de calidad del software**
- **Control de cambios (6)**

Beneficios que aporta RUP

- Permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de los riesgos.
- Permite la producción de software que cumpla con las necesidades de los usuarios, a través de la especificación de los requisitos, con una agenda y costo predecible.
- Enriquece la productividad en equipo y proporciona prácticas óptimas de software a todos sus miembros.
- Permite llevar a cabo el proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades críticas.
- Unifica todo el equipo de desarrollo de software y mejora la comunicación al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado y un punto de vista de cómo desarrollar software.
- Optimiza la productividad de cada miembro del equipo al poner al alcance la experiencia derivada de miles de proyectos y muchos líderes de la industria.

2.3. Lenguaje de Modelado

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (7)

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. La estructura estática define los tipos de objetos importantes para un sistema y para su implementación, así como las relaciones entre los objetos. El comportamiento dinámico define la historia de los objetos en el tiempo y la comunicación entre objetos para cumplir sus objetivos. El modelar un sistema desde varios puntos de vista, separados pero relacionados, permite entenderlo para diferentes propósitos.

Características de UML

- Permite adaptarse fácilmente a los usuarios, así como a usuarios de otros métodos.
- Permite modelar nuevos elementos en el proyecto que se desarrolla.
- UML provee una expresividad e integridad holística mejorada, respecto a otros lenguajes de moldeamiento visual.
- UML es fácil de aprender y usar, ya sea respecto a las técnicas más avanzadas, es decir, estereotipos y propiedades, así como algunos cambios en la notación y semánticas.

Capítulo 2: Tendencias y tecnologías actuales

- UML será la elección obvia para realizar nuevos proyectos, especialmente cuando se incremente la disponibilidad de herramientas, libros y cursos.

Las ventajas que posibilita la utilización de UML son varias, por un lado el uso de lenguajes visuales facilita su asimilación y entendimiento por parte del equipo de trabajo; el tiempo invertido en el desarrollo de la arquitectura se minimiza; la detección y resolución de errores se agiliza siempre y cuando se haga uso de herramientas adecuadas de diagnóstico y depuración; y la trazabilidad y documentación del proyecto se confecciona de una forma ordenada y guiada por los casos de uso.

2.4. Herramienta Case

Las herramientas CASE (Computer Aided/Assisted Software/System Engineering) o (Ingeniería de Software Asistida por Computadora) brindan soporte a un enfoque de ingeniería en el desarrollo de software en alguna o en todas las fases de este proceso (8).

Tienen como objetivo:

- Permitir la aplicación práctica de metodologías estructuradas.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes del software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones.

Existen varias herramientas de modelado UML que posibilitan el modelado visual de proyectos de software, viabilizando la comunicación entre los integrantes del equipo de desarrollo. Algunos ejemplos son Rational Rose y Visual Paradigm.

Visual Paradigm para UML es una herramienta CASE, que utiliza UML 2.1 como lenguaje de modelaje y soporta el ciclo de vida completo del desarrollo de software. Es multiplataforma y posibilita la generación de código y documentación. Permite dibujar 13 tipos de diagramas diferentes a través de un intuitivo

Capítulo 2: Tendencias y tecnologías actuales

modelado visual y la aplicación de ingeniería inversa. Admite la importación y exportación de XML e imágenes, la administración de requerimientos, la creación de esquemas de clases a partir de una base de datos y viceversa, además cuenta con un soporte que facilita el trabajo simultáneo sobre un mismo diagrama entre dos desarrolladores en un tiempo real. (9) Teniendo en cuenta lo antes expuesto es por lo que se ha seleccionado a Visual Paradigm como herramienta CASE para el desarrollo de la solución propuesta.

2.5. Lenguaje de Programación

Los lenguajes de programación de alto nivel logran la independencia del tipo de máquina, permiten la ejecución de programas creados por lenguajes cerca del habla humana posibilitando la escritura de instrucciones de forma fácil y entendible. Un ejemplo de los principales lenguajes que presentan estas características son Java, C++, PHP, C#, Visual Basic, entre otros, siendo Java, C++, PHP y C# los más utilizados. Para desarrollar el sistema propuesto se hace necesaria la utilización eficiente de los recursos de hardware de la computadora. El sistema debe ser lo más ligero posible dado el alto consumo de procesamiento y memoria que requerirán sus funcionalidades, por ello el lenguaje seleccionado es C++, lenguaje versátil, potente y general.

C++ mantiene las ventajas de C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original y se le añadieron funcionalidades de las que su antecesor carecía, como la programación orientada a objetos (aunque no es un lenguaje enteramente orientado a objetos), manejo de excepciones y la sobrecarga de operadores. El resultado fue un lenguaje muy ligado al hardware tal como su antecesor, manteniendo una considerable potencia para programación a bajo nivel, con elementos que le permiten también un estilo de programación con alto nivel de abstracción. (10)

Para la creación de sistemas distribuidos, donde el aprovechamiento de los recursos de hardware es uno de los principales fines, es considerable el desarrollo de los mismos en plataformas de escritorio. El desarrollo de una plataforma de escritorio enfoca a la búsqueda de fáciles alternativas de programación en distribuciones operativas libres, teniendo en cuenta que el lenguaje de programación es indiferente al sistema operativo. (10)

Capítulo 2: Tendencias y tecnologías actuales

Por todo lo antes expuesto, C++ ha sido el lenguaje de programación seleccionado para el desarrollo del sistema, tomando en cuenta además, que es el lenguaje nativo del framework de desarrollo Qt, el cual brinda un conjunto de funcionalidades que facilitan y agilizan el desarrollo de la aplicación.

2.6. Framework de Desarrollo

Un framework es un marco de trabajo o conjunto de bibliotecas orientadas a la reutilización a muy gran escala de componentes software para el desarrollo rápido de aplicaciones. Con su utilización se obtiene un desarrollo más rápido de las aplicaciones, la reutilización de componentes software, el uso y la programación de componentes con una política de diseño uniforme. El término framework tiene una acepción más amplia, en donde además de incluir una biblioteca de componentes reutilizables es toda una tecnología o modelo de programación; contiene máquinas virtuales, compiladores, bibliotecas de administración de recursos en tiempo de ejecución y especificaciones de lenguajes. Tal es el caso del framework Qt de Nokia (11). Qt es una framework multiplataforma para desarrollar interfaces gráficas de usuario y también para el desarrollo de programas sin interfaz gráfica como herramientas de consola y servidores. Es producido por la división de software Qt de Nokia. Utiliza el lenguaje de programación C++ de forma nativa, adicionalmente puede ser utilizado con otros lenguajes de programación.

Ofrece además un conjunto de funciones las cuales facilitan el desarrollo de la solución que se propone. Una de ellas es el mecanismo de señales y slots; las señales son emitidas cuando ocurre un evento en particular y el slot es una función que se ejecuta en respuesta a una señal determinada. La posibilidad del desarrollo y carga de plugins es otra de las funcionalidades aportadas por este framework a la implementación de aplicaciones. Con la utilización de plugins se logra una mayor escalabilidad permitiendo al sistema adaptarse a posibles cambios futuros, la inclusión de nuevas funcionalidades, así como el desarrollo de nuevas versiones del sistema sin necesidad de hacer modificaciones en el código del sistema. (12)

Funciona en los principales sistemas operativos, y tiene un amplio apoyo de la comunidad de desarrollo. La Interfaz de Programación de Aplicaciones (API por sus siglas en inglés) de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML, gestión de hilos, soporte de red, una API multiplataforma unificada para la manipulación de archivos y una multitud de otros para el

Capítulo 2: Tendencias y tecnologías actuales

manejo de ficheros, además de estructuras de datos tradicionales. Es distribuido bajo los términos de GNU Lesser General Public License (y otras), Qt es software libre y de código abierto.

2.7. Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado o IDE (del inglés Integrated Development Environment) es un programa compuesto por un conjunto de herramientas que proveen un marco de trabajo amigable para los lenguajes de programación (13). Dentro de las partes fundamentales que lo integran se encuentra el editor de código, compilador, depurador y un constructor de interfaz gráfica de usuario.

Qt Creator es un IDE integrado al framework Qt y es compatible con los sistemas operativos Windows, Linux y Mac OS X. Contiene un sofisticado editor de código que proporciona completamiento de código y ayuda de contexto para el lenguaje C++ y JavaScript. Tiene integrado los sistemas de control de versiones más populares tales como Subversion, CVS, Perforce y Mercurial. Cuenta con dos editores visuales, Qt Designer para diseñar interfaces de usuario a partir de Qt Widgets, y Qt Quick Designer para el desarrollo de interfaces de usuario animadas con el lenguaje JavaScript. Provee soporte para la construcción de aplicaciones de escritorio y dispositivos móviles (12).

Qt Creator es el IDE seleccionado para el desarrollo de la aplicación propuesta dadas las características descritas anteriormente. El mismo cuenta con una amplia documentación incluyendo ejemplos sobre el trabajo con la mayoría de sus funcionalidades. Uno de los principales elementos que inclinaron la balanza en favor de este IDE es que el mismo es software libre con licencia LGPL.

2.8. Conclusiones

En este capítulo se han presentado los instrumentos a utilizar en el proceso de desarrollo de la solución que se propone y sus características los hacen los más adecuados para el proceso de desarrollo. Se llegó a la conclusión de utilizar RUP como metodología de desarrollo ya que mantiene una documentación exhaustiva permitiendo entendimiento entre equipos de trabajo y la utilización futura de la solución propuesta. Como lenguaje de modelado se aplica UML por ser de fácil asimilación y entendimiento, el tiempo invertido en el desarrollo de la arquitectura se minimiza y agiliza la detección y resolución de errores. UML se utiliza con la herramienta Visual Paradigm ya que soporta el ciclo de vida completo del desarrollo de software y es multiplataforma. El lenguaje de programación es C++ gracias a que está muy

Capítulo 2: Tendencias y tecnologías actuales

ligado al hardware con elementos que le permiten un estilo de programación con alto nivel de abstracción. Se utiliza como entorno de desarrollo integrado Qt Creator con las librerías correspondientes al framework Qt de Nokia por ser multiplataforma y ofrecer un conjunto de funcionalidades como el mecanismo de señales y la implementación de plugins.

CAPÍTULO 3: Presentación de la solución propuesta.

3.1. Introducción

En este capítulo se reflejan los elementos centrales de la modelación del negocio representados a través del modelo de dominio. Se explican los requisitos funcionales y no funcionales del sistema y se presenta el diagrama de casos de uso del sistema con la descripción de los casos de uso del sistema considerados como críticos.

3.2. Modelo de Dominio

El Modelo de Dominio muestra las clases conceptuales significativas en el dominio del problema en cuestión, abarcando los ejemplares más importantes de objetos existentes o los eventos que suceden en el entorno donde se desempeñará el sistema. Es considerado un subconjunto del Modelo de Objetos del Negocio.

El modelo de dominio se describe mediante diagramas UML (especialmente mediante diagrama de clases). Estos diagramas muestran las clases del dominio y cómo se relacionan unas con otras mediante asociaciones. Proporciona a los usuarios, clientes, desarrolladores y otros interesados, un vocabulario común.

Las clases del dominio aparecen en tres formas típicas:

- Objetos del negocio que representan elementos que se manipulan en el negocio.
- Objetos del mundo real y conceptos de los que el sistema debe hacer un seguimiento.
- Sucesos que ocurrirán o han ocurrido.

En el transcurso del desarrollo de la solución no se logró relacionar procesos bien definidos en el entorno del negocio. Se hizo difícil especificar los elementos más importantes del sistema y sus interconexiones, así como no se pudo establecer las reglas de su funcionamiento. A pesar de ello es posible identificar personas, eventos, transacciones y objetos involucrados en dicho entorno escasamente perfilado, por lo cual se determinó necesario realizar un modelado del dominio concerniente a la solución.

3.2.1. Descripción General del Modelo de Dominio

Todos los clientes con permisos en el sistema pueden introducir un nuevo material haciendo uso de la herramienta Upload. Esta se encarga de almacenar el material en el servidor de media y notificar a la base de datos que se ha subido al servidor de almacenamiento un nuevo archivo de media. Existen dos programas corriendo, los cuales realizan consultas a la base de datos cada cierto tiempo, para notificar cuándo se ha subido un nuevo material al sistema. Estos son los encargados de ejecutar las tareas de codificación e indexación sobre el archivo luego de conocer la existencia del nuevo fichero.

3.2.2. Diagrama de clases del Modelo de Dominio

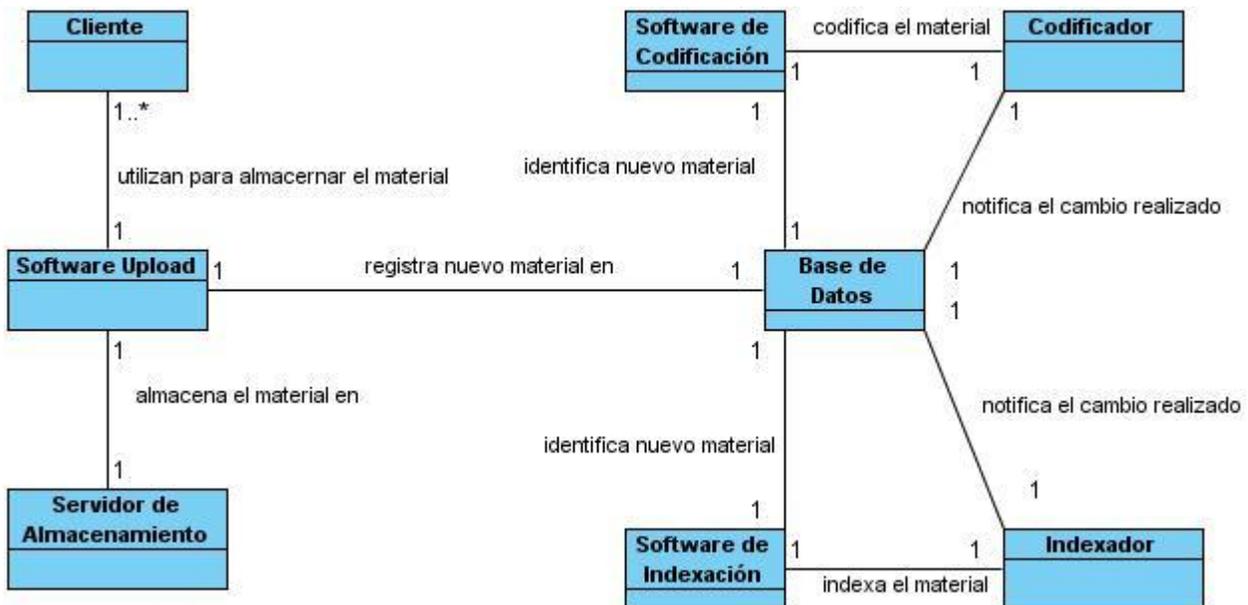


Figura 2. Modelo de Dominio

3.2.3. Descripción de Clases

Cliente: Es todo el personal encargado de subir materiales audiovisuales al sistema haciendo uso de un software.

Software Upload: Programa informático que se encarga de almacenar un material audiovisual en un servidor de almacenamiento y notifica el nuevo material en la base de datos.

Servidor de almacenamiento: Un cómputo con capacidad de disco duro mayor que una computadora común, se usa para almacenar gran cantidad de materiales audiovisuales.

Base de Datos: Registro almacenado en un sistema gestor de base de dato el cual se encuentra accesible en la red para que puedan acceder los programas que tengan permisos. Es en la base de datos donde se almacena toda la información de los materiales audiovisuales.

Software de Codificación: Programa informático que se encarga de detectar cuando ha entrado un nuevo material al servidor de media haciendo uso de la información de la base de datos. Es quien ordena la codificación del archivo.

Codificador: Programa informático que se encarga de codificar (cambiar el formato a un fichero de media) un material cuando el software de codificación se lo provee.

Software de Indexación: Programa informático que se encarga de detectar un nuevo material añadido al servidor de almacenamiento de medias y ejecuta su indexación haciendo uso del Indexador.

Indexador: Programa informático que se encarga de indexar un material audiovisual, el material se lo provee el Software de Indexación.

Capítulo 3: Presentación de la solución propuesta

3.3. Requisitos Funcionales

En este epígrafe se hará referencia a los requisitos funcionales, estos son las capacidades o condiciones que el sistema debe cumplir.

Alias	Requisito	Descripción
RF 1	Analizar video	El sistema debe brindar la funcionalidad de analizar un video con el objetivo de verificar que el formato del material se corresponde con el establecido en el sistema.
RF 2	Codificar video	El sistema debe brindar la funcionalidad de codificar al formato de video en baja resolución para streaming establecido o a un formato de video especificado por el usuario.
RF 3	Elaborar resumen visual de video	El sistema debe permitir realizar un resumen visual de los videos almacenados por sistema.
RF 4	Cortar video	El sistema debe brindar la funcionalidad de cortar un fragmento de video a partir de tiempos especificados por el usuario.
RF 5	Reportar Nodo	El sistema debe brindar la funcionalidad de reportar la existencia de un nodo en la red en el instante que se inicia la ejecución de la aplicación.

Tabla 1. Descripción de Requisitos Funcionales del Sistema

3.4. Requisitos No Funcionales

En este epígrafe se hará referencia a los requisitos no funcionales, estos son las propiedades o cualidades que el producto debe tener. Para ello se tuvo en cuenta la importancia que poseen en función de quienes, clientes y usuarios, puedan valorar cuán usable, seguro, conveniente y agradable será el software.

3.4.1. Fiabilidad

Disponibilidad: el sistema debe estar disponible todas las horas del día.

Tiempo medio de reparación: el sistema no debe estar fuera de servicio más de 2 horas a partir de un fallo.

Capítulo 3: Presentación de la solución propuesta

3.4.2. Eficiencia

3.4.2.1. Eficiencia

El sistema debe ser eficiente a las peticiones realizadas en cada momento, el flujo de trabajo que sigue la aplicación no permite el fallo de ninguna de las partes, pues este influye de manera drástica sobre el próximo paso.

3.4.2.2. Tiempo de respuesta

El tiempo de respuesta promedio de las peticiones realizadas al sistema no debe exceder un tiempo máximo de 5 segundos.

3.4.3. Software

Clientes	
Sistema Operativo	Dependencias
GNU/Linux	<ul style="list-style-type: none">• Framework Qt, Licencia Qt GNU LGPL v. 2.1.• Mencoder, Licencia GPL• Ffmpeg2theora v2.7, Licencia GNU LGPL• Ffmpeg, Licencia GPL/LGPL• MediaInfo, Licencia GPL

Tabla 2. Software para Estaciones Cliente

Servidores		
Sistema Operativo	Aplicación Servidor de Codificación de Datos	Aplicación Servidor de Indexación de Datos
GNU/Linux	<ul style="list-style-type: none">• Biblioteca Ice 3.4.0, Licencia GNU v2.• Framework Qt, Licencia Qt GNU LGPL v. 2.1.• MEncoder, Licencia GPL.• Ffmpeg2theora v2.7, Licencia GNU LGPL.• Ffmpeg, Licencia GPL/LGPL• MediaInfo, Licencia GPL,LGPL	<ul style="list-style-type: none">• Framework Qt, Licencia Qt GNU LGPL v. 2.1.• OpenCV v2.1, Licencia BSD.• Biblioteca Ice 3.4.0, Licencia GNU v2

Tabla 3. Software para Servidores

3.4.4. Hardware

Servidor	Arquitectura	Procesador	Memoria
Codificación	64 bits (x64)	Core2Duo	4 GB
Indexación	64 bits (x64)	Core2Duo	4 GB

Tabla 4. Hardware para Servidores

3.4.5. Interfaz

3.4.5.1. Interfaces de software

El sistema utilizará aplicaciones de apoyo para la realización del procesamiento de video, con las cuales se deberá garantizar la comunicación desde software.

3.4.5.2. Interfaces de comunicación

El sistema requiere de comunicación a través de los protocolos TCP/IP.

3.5. Descripción del Sistema Propuesto

3.5.1. Descripción de los actores

Actor	Descripción
Gestor de Procesos	Procesos que controlan remotamente las peticiones de las aplicaciones y la ejecución de los algoritmos de procesamiento de video.
Usuario Plataforma	Actor que realiza peticiones de procesamiento de video a la aplicación en un ambiente local.
Usuario	Este actor se comporta como una generalización de los actores Gestor de Procesos y Usuario Plataforma
Proceso Automático Plataforma	Proceso que inicia la notificación al sistema remoto de la presencia de un nodo en la red.

Tabla 5. Descripción de Actores del Sistema

3.5.2. Casos de Uso del Sistema

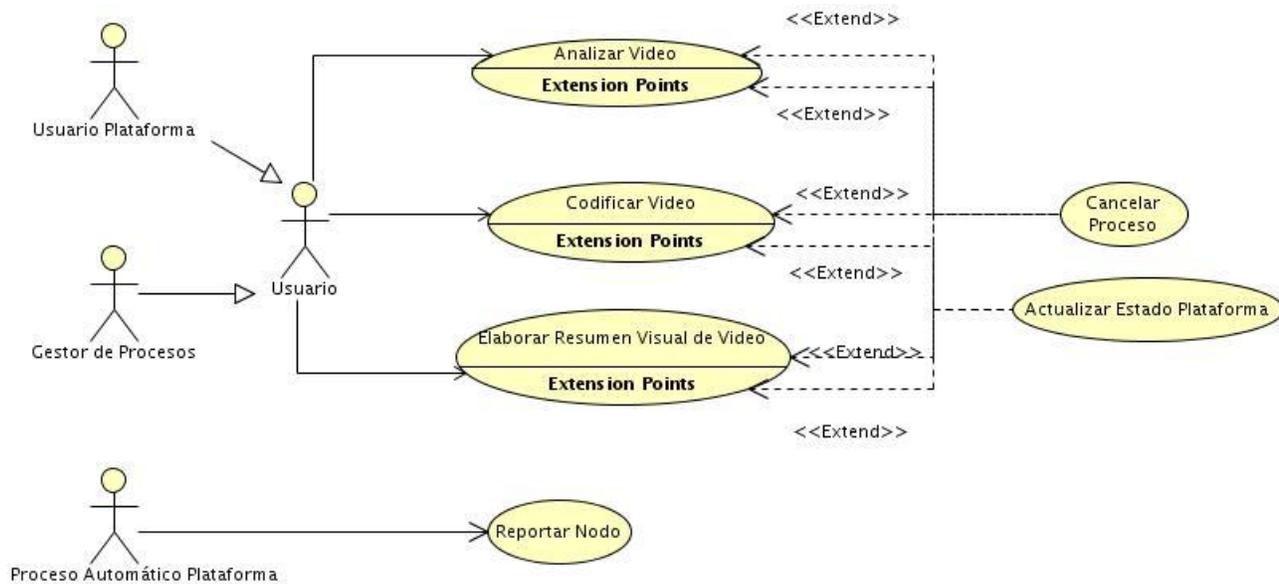


Figura 3. Diagrama de Casos de Uso del Sistema

3.5.2.1. Analizar Video

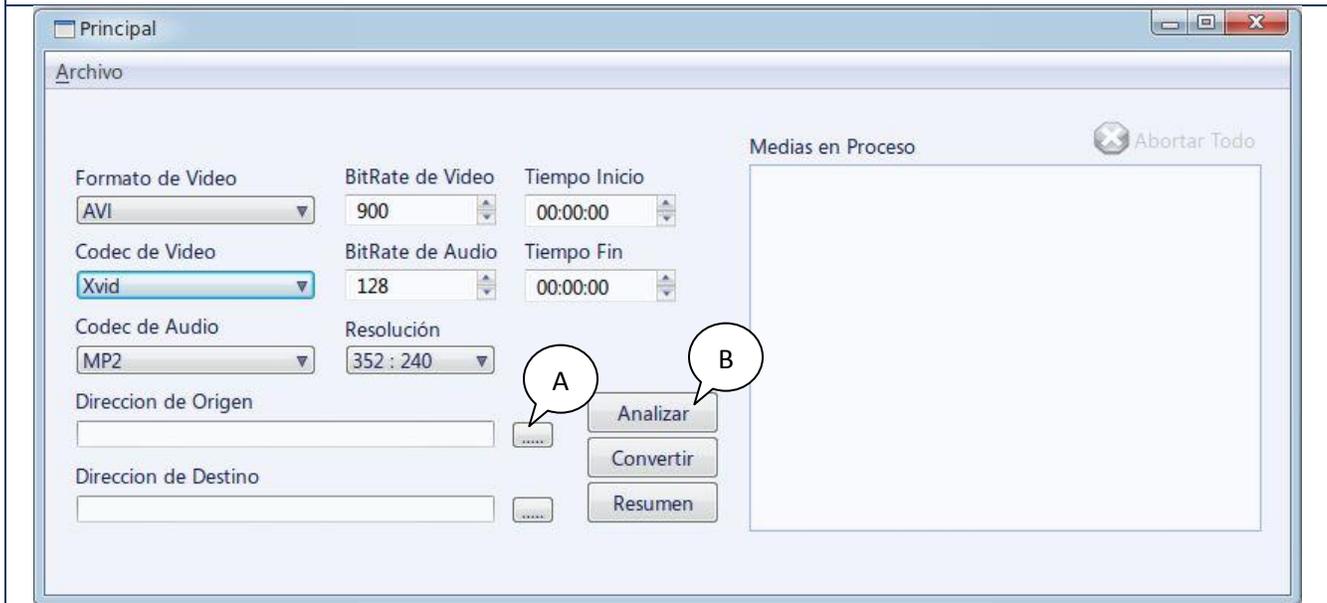
Caso de Uso:	Analizar Video	
Actores:	Usuario	
Resumen:	El caso de uso se inicia cuando se envía la orden de analizar un material con el fin de que cumpla con el perfil establecido para los materiales originales. Implícitamente se le extrae la duración de material.	
Precondiciones:	Se debe haber ordenado el análisis a un nuevo material en el sistema.	
Referencias	RF 1	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	

Capítulo 3: Presentación de la solución propuesta

<p>1. El caso de uso se inicia cuando se recoge el directorio origen (A) del material a analizar que puede ser introducido por un usuario, el cual seleccionará además la opción “Analizar” (B) desde la interfaz de usuario o enviado por la aplicación de control de procesos. Los datos se envían a la estación servidora que realizará el análisis.</p>	<p>1.1. El sistema carga los datos y procede a realizar el análisis del material. 1.2. El sistema verifica que el material cumple con las especificaciones del perfil enviado. 1.3. El sistema extrae la duración del material. 1.4. El sistema notifica el estado de ejecución del proceso al monitor de procesos. 1.5. El sistema notifica la culminación de la ejecución del proceso al monitor de procesos. Culmina el caso de uso.</p>
---	---

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.3. Si el material no cumple con las especificaciones del perfil, se notifica el error correspondiente.

Prototipo de Interfaz



Poscondiciones	Se logra identificar el cumplimiento de un material con perfil de video predefinido.
-----------------------	--

Tabla 6. Descripción Textual del CU Analizar Video

Capítulo 3: Presentación de la solución propuesta

3.5.2.2. Codificar Video

Caso de Uso:	Codificar Video
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando se ordena la ejecución de un proceso de codificación a un formato de mayor compresión o cuando se desea codificar a un formato especificado. Dentro de sus funcionalidades implícitas se encuentra la codificación de fragmentos de materiales.
Precondiciones:	Se debe haber ordenado la ejecución de un proceso de codificación de un material tanto a un formato de mayor compresión como a un formato específico.
Referencias	RF 2, RF 4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando se envía a la estación servidora que atenderá el proceso de codificación del material, los datos de directorio origen (A) y destino (B) donde se quiere almacenar el resultado del proceso, el fragmento del material (tiempo de inicio y tiempo de fin) (C) y los parámetros de codificación (D). Estos datos pueden ser seleccionados por un usuario desde la interfaz de usuario, en la que deberá seleccionar además la opción "Convertir" (E) o pueden ser enviados por el monitor de procesos.	1.1. El sistema carga los datos recibidos y comienza la ejecución del proceso, almacenando su resultado en el directorio destino. 1.2. Notifica el estado de ejecución del proceso cargado. 1.3. Notifica la finalización del proceso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.1. En caso de error durante la ejecución del proceso se notifica el mismo.
Prototipo de Interfaz	

Capítulo 3: Presentación de la solución propuesta

Poscondiciones Se obtiene un material codificado según el perfil de codificación

Tabla 7. Descripción Textual del CU Codificar Video

3.5.2.3. Elaborar Resumen Visual de Video

Caso de Uso:	Elaborar Resumen Visual de Video	
Actores:	Usuario	
Resumen:	El caso de uso se inicia cuando se ordena la ejecución de un proceso de extracción de resumen visual de video	
Precondiciones:	Se debe haber ordenado la ejecución de un proceso de extracción de resumen visual de video por la inclusión de un nuevo material al sistema.	
Referencias	RF 3	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	

Capítulo 3: Presentación de la solución propuesta

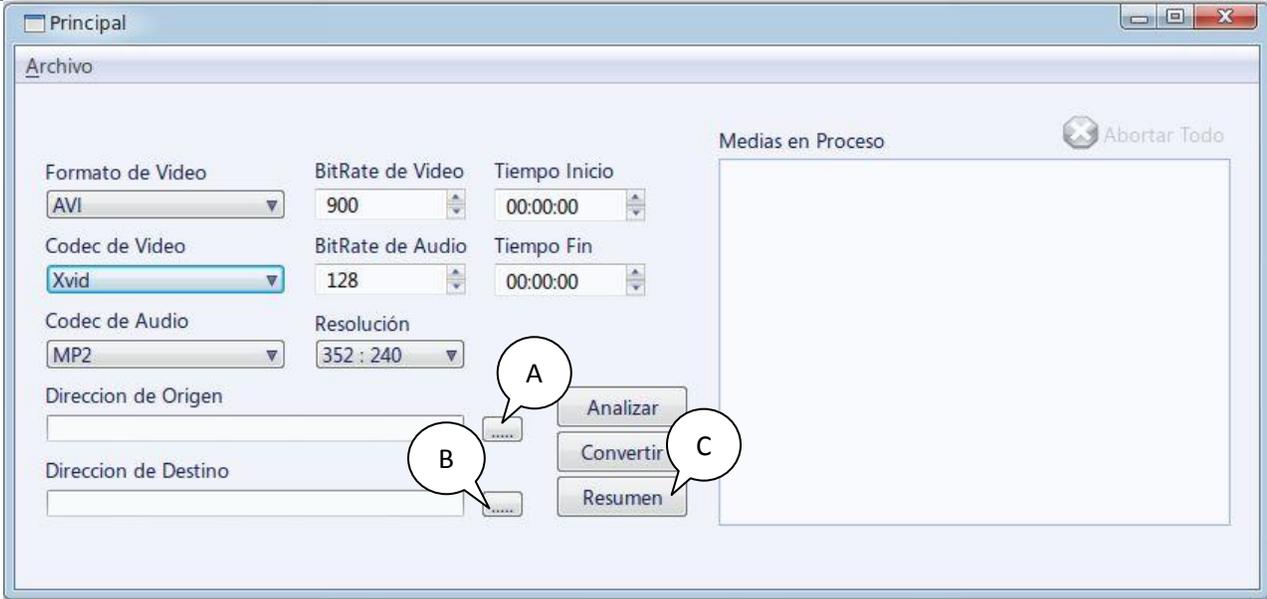
<p>1. El caso de uso comienza cuando se envía a la estación servidora que atenderá el proceso de elaboración de resumen visual de video, los datos de directorio origen (A) y destino (B) donde se quiere almacenar el resultado del algoritmo. Los datos pueden ser introducidos por un usuario desde la interfaz de usuario, donde deberá seleccionar además la opción “Resumen” (C), o ser enviados por el monitor de procesos.</p>	<p>1.1. El sistema carga los datos recibidos y comienza la ejecución del algoritmo, almacenando su resultado en el directorio destino.</p> <p>1.2. Notifica el estado de ejecución del proceso cargado.</p> <p>1.3. Notifica la finalización del proceso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.1. En caso de error durante la ejecución del proceso se notifica el mismo.
Prototipo de Interfaz	
	
Poscondiciones	Se obtiene un compendio de imágenes que representan el resumen visual de un material.

Tabla 8. Descripción Textual del CU Elaborar Resumen Visual de Video

3.6. Conclusiones

En este capítulo se realizó la modelación del negocio a través de un modelo de dominio, a fin de lograr una mejor comprensión de los procesos, para que los usuarios finales y desarrolladores posean un entendimiento común. Se identificaron los requerimientos tanto “funcionales” como “no funcionales” a tener en cuenta para el desarrollo del sistema. Se realizó el diagrama de casos de uso del sistema, así como sus respectivas descripciones, en aras de obtener una visión global de la concepción y funcionamiento del sistema.

CAPÍTULO 4. Construcción de la solución propuesta

4.1. Introducción

En este capítulo se abordan temas fundamentales para la construcción del sistema propuesto. Se recogen los diagramas de clases y principios del diseño, así como las generalidades del Flujo de Trabajo de Implementación. Se presentan las pruebas realizadas al sistema desarrollado.

4.2. Diagrama de Clases

A continuación se muestra el diagrama de clases del diseño de la aplicación, para un mejor entendimiento se han agrupado acorde a sus objetivos funcionales.

4.2.1. Plataforma de Codificación e Indexación de Video

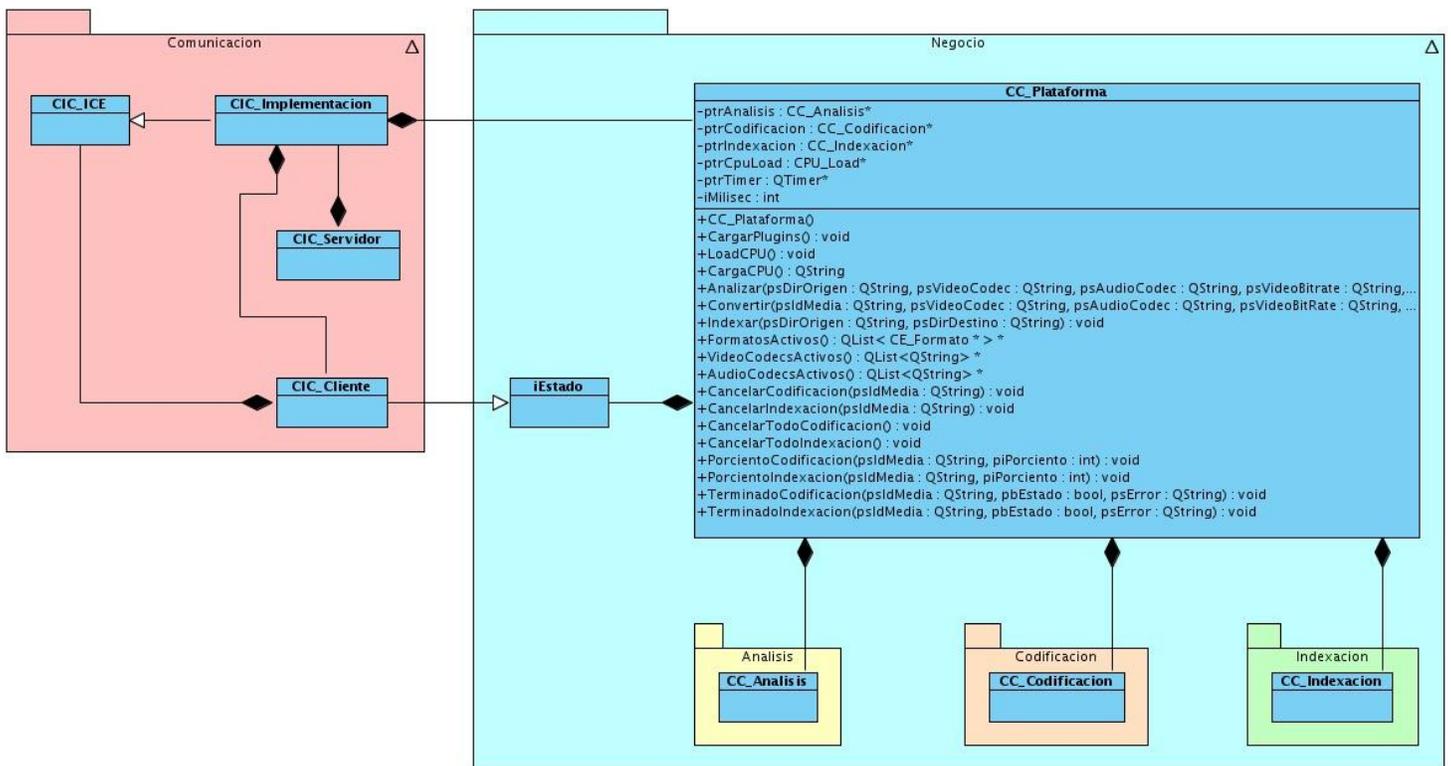


Figura 4. Diagrama de Clases de Diseño: Plataforma de Codificación e Indexación

4.2.2. Análisis de Video

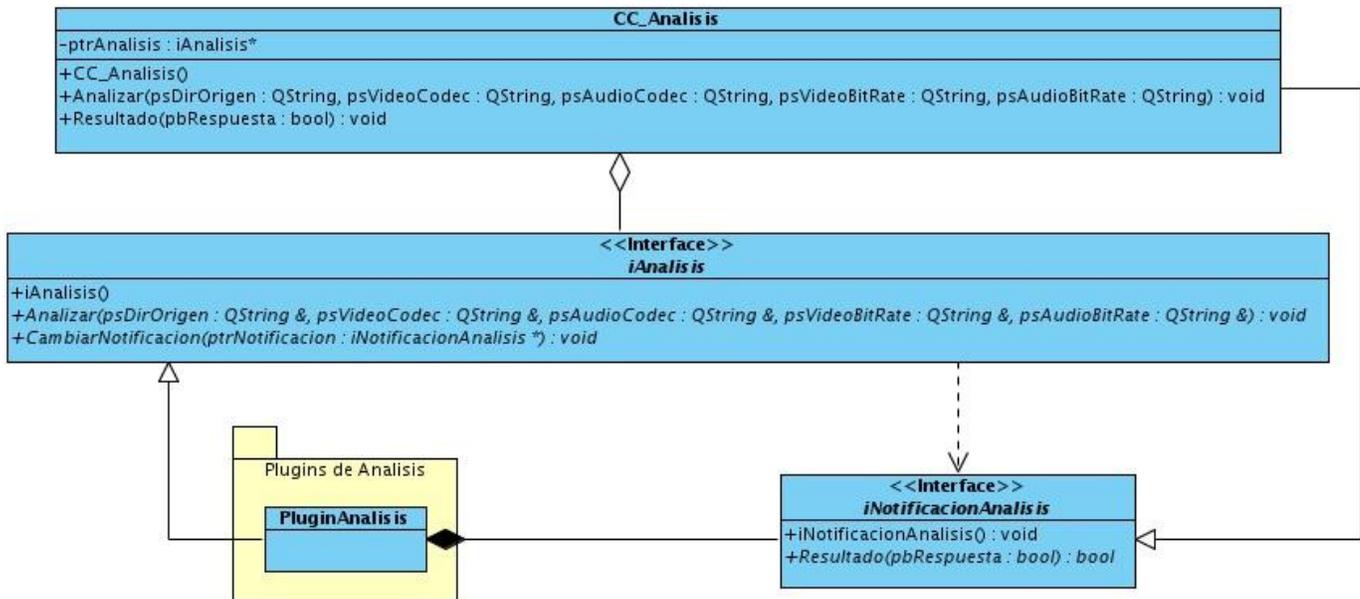


Figura 5. Diagrama de Clases de Diseño: Análisis

4.2.2.1. Plugin de Análisis de Video

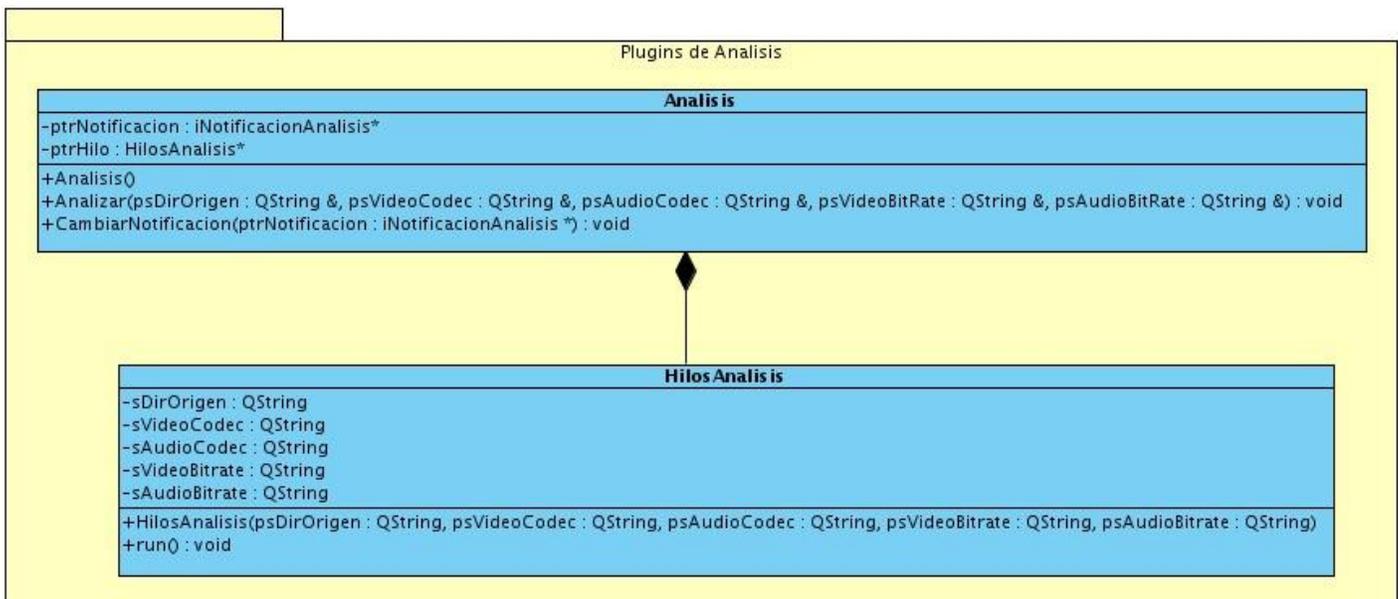


Figura 6. Diagrama de Clases de Diseño: Plugin de Análisis

4.2.3. Codificación de Video

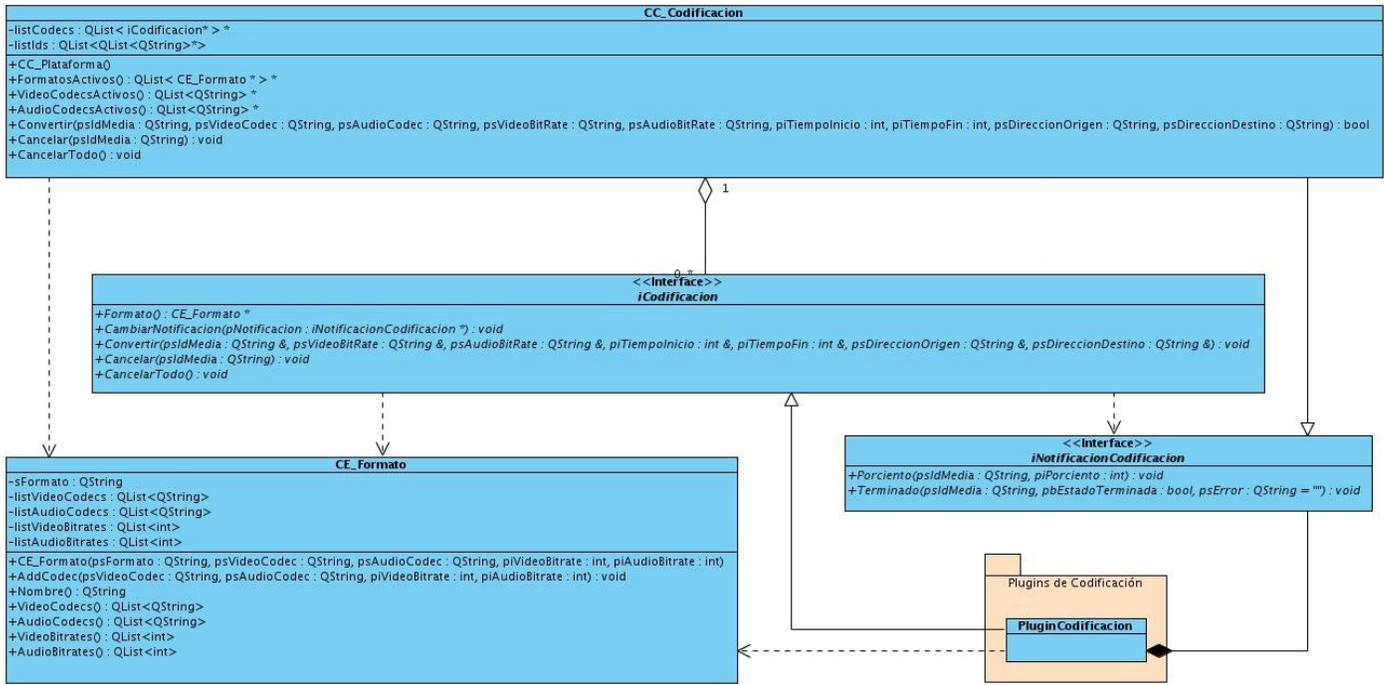


Figura 7. Diagrama de Clases de Diseño: Codificación

4.2.3.1. Plugin de Codificación de Video

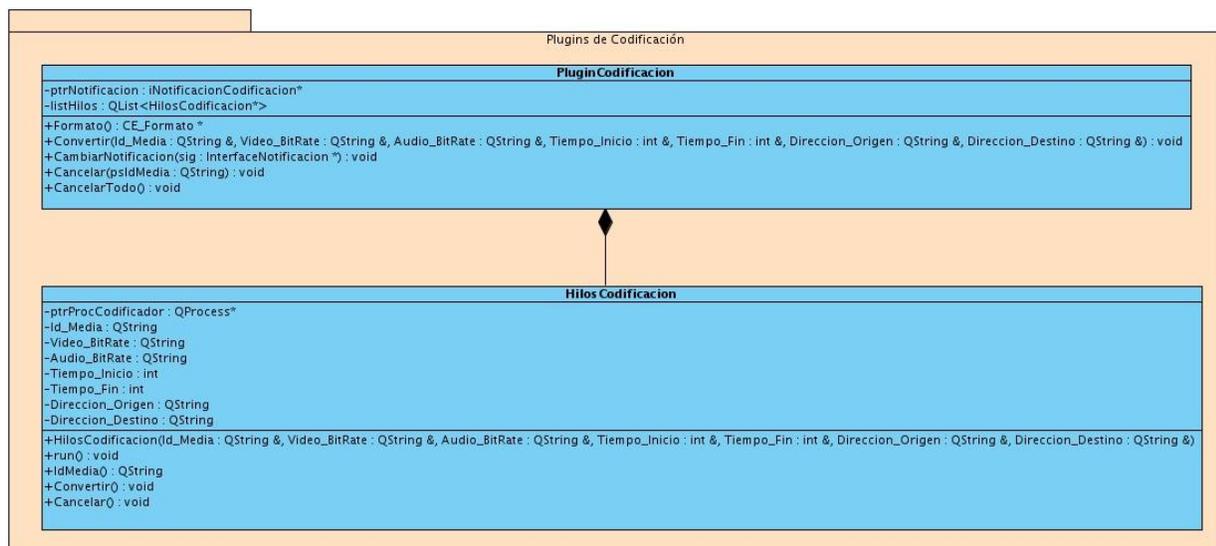


Figura 8. Diagrama de Clases de Diseño: Plugin de Codificación

4.2.4. Indexación de Video

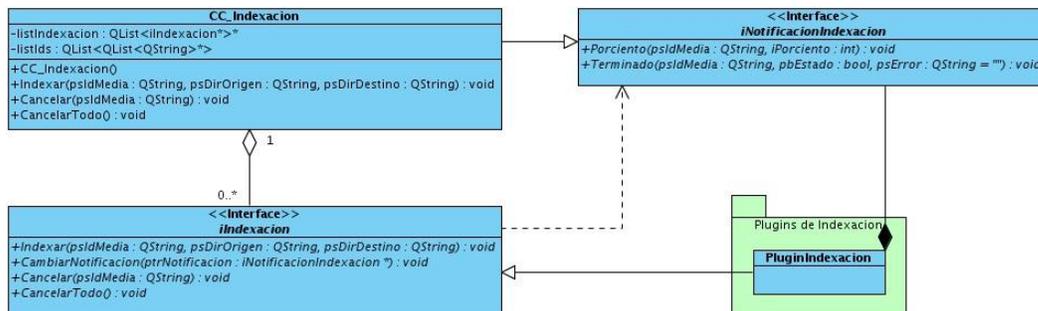


Figura 9. Diagrama de Clases de Diseño: Indexación

4.2.4.1. Plugin de Indexación de Video

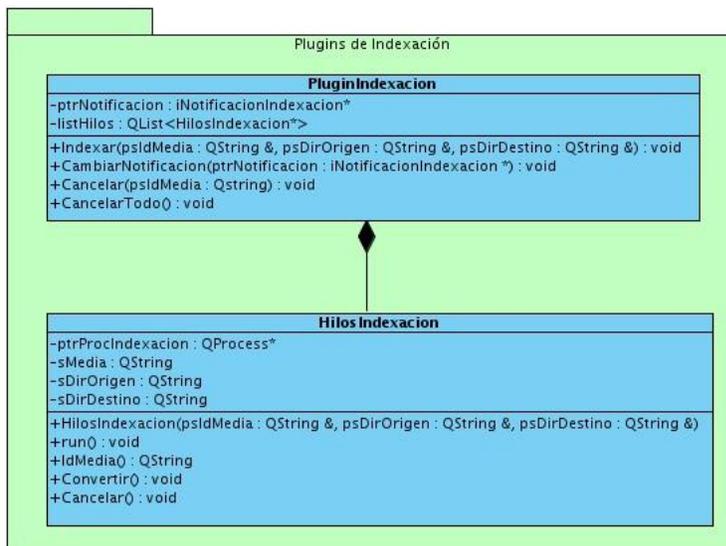


Figura 10. Diagrama de Clases de Diseño: Plugin de Indexación

4.3. Generalidades de la implementación

Los constantes avances en la rama de las TIC a la par de las crecientes necesidades de sus usuarios, han provocado que la industria de software se plantee nuevos mecanismos para lograr desarrollar aplicaciones en menor tiempo y con un mayor nivel de escalabilidad. Uno de estos mecanismos es el desarrollo de sistemas utilizando complementos, también conocidos como plugins. Un plugin es una aplicación que se relaciona con otra para aportarle nuevas funcionalidades, esta aplicación adicional es ejecutada por la

Capítulo 4: Construcción de la solución propuesta

aplicación principal e interactúan a través de APIs. La aplicación principal o host provee servicios que los complementos pueden utilizar incluyendo un método para registrarse ellos mismos y un protocolo para el intercambio de datos, de esta manera los plugins dependen de los servicios prestados por el host y no suelen funcionar por sí mismos. Por otro lado la aplicación principal funciona independientemente de ellos, lo que permite a los usuarios finales añadir y actualizar los complementos de forma dinámica sin necesidad de hacer cambios a la aplicación. Qt ofrece soporte para el desarrollo de aplicaciones basadas en plugins además de un conjunto de ejemplos y documentación sobre el tema.

El sistema desarrollado necesita ser escalable y eficiente, es por ello que para su implementación se utilizó el mecanismo descrito anteriormente integrándose a una arquitectura de dos capas: Presentación – Comunicación y Negocio. Los procesos de codificación e indexación de video requieren de un alto consumo de procesamiento por lo cual los plugins desarrollados para este tipo de tareas están basados en programación multihilos. La programación multihilos utilizando las librerías de Qt, aseguran un eficiente consumo de recursos de hardware, proporcionando además la posibilidad de la realización de varias tareas al mismo tiempo. De igual forma se hizo necesaria la utilización de varios patrones de diseño, es el caso del patrón Singleton: el sistema tiene un único punto de entrada utilizando para ello la misma instancia de la clase controladora para todas las peticiones; el patrón Observador: el sistema permite a los objetos captar dinámicamente las dependencias entre objetos, de tal forma que un objeto notificará a los objetos dependientes de él cuando cambia su estado, siendo actualizados automáticamente. (14)

Para garantizar la reutilización y entendimiento de los códigos desarrollados se realizó la programación siguiendo estándares de codificación previamente definidos, donde por ejemplo, el nombre de métodos y clases comienzan en mayúscula con notación “camel case” (GetNombre), para el casos de las variables se identifica primero el tipo de dato y luego un nombre referente a la información para el que están destinados (sNombre), en el caso específico de los parámetros se antepone la letra ‘p’ (SetNombre(string psNombre)). Las clases reflejarán el tipo de uso que se le da, por ejemplo las clases controladoras se nombran con dos letras ‘C’ separadas del nombre de la clase por un separador (CC_MiClase), las mismas se encuentran distribuidas por carpetas, separadas de acuerdo a la estructura definida por capas en el modelo de clases del diseño, logrando así la utilización y optimización de códigos y clases implementadas sin importar el módulo al cual correspondan.

4.4. Modelo de Despliegue

El Modelo de Despliegue provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema propuesto. Muestra dónde y cómo se desplegarán los componentes. Es un mapa específico de la instalación física del sistema. Un diagrama de despliegue ilustra el despliegue físico del sistema en un ambiente de producción (o prueba). Muestra dónde se ubicarán los componentes, en qué servidores, máquinas o hardware. Puede ilustrar vínculos de red, ancho de banda de LAN, etc. (15)

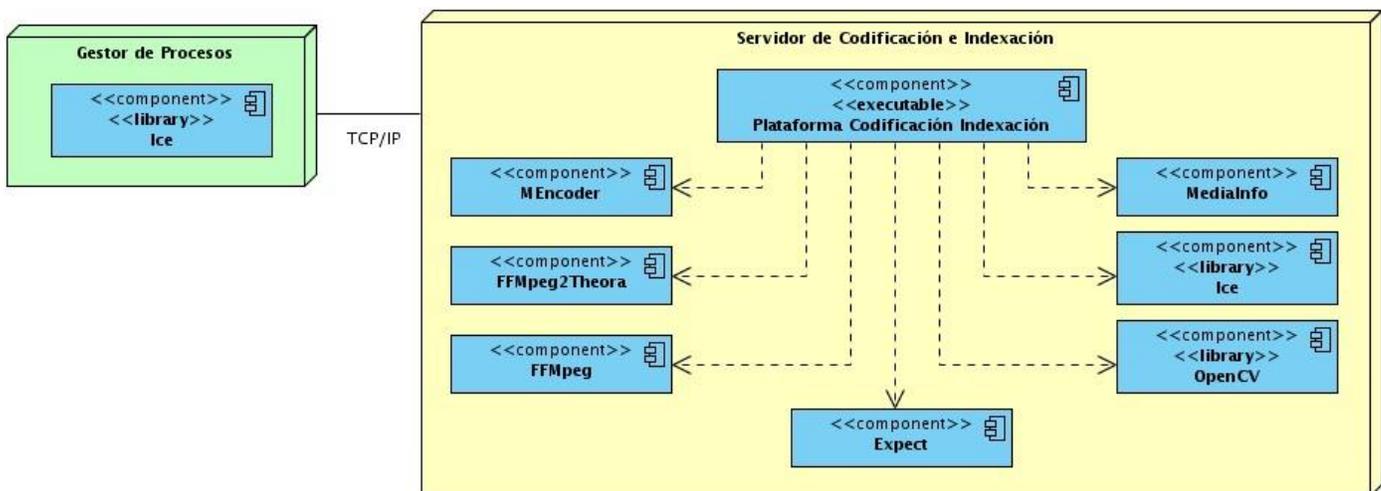


Figura 11. Diagrama de Despliegue

El diagrama presentado muestra la organización óptima que debe tener el despliegue del sistema de acuerdo a las particularidades de la entidad que lo utilice. Pueden existir variaciones al utilizarse como aplicación en solitario, en tal caso no será necesario el Gestor de Procesos.

4.5. Modelo de Implementación

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas y mostrar las relaciones entre sus elementos. Es un diagrama que muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. (16)

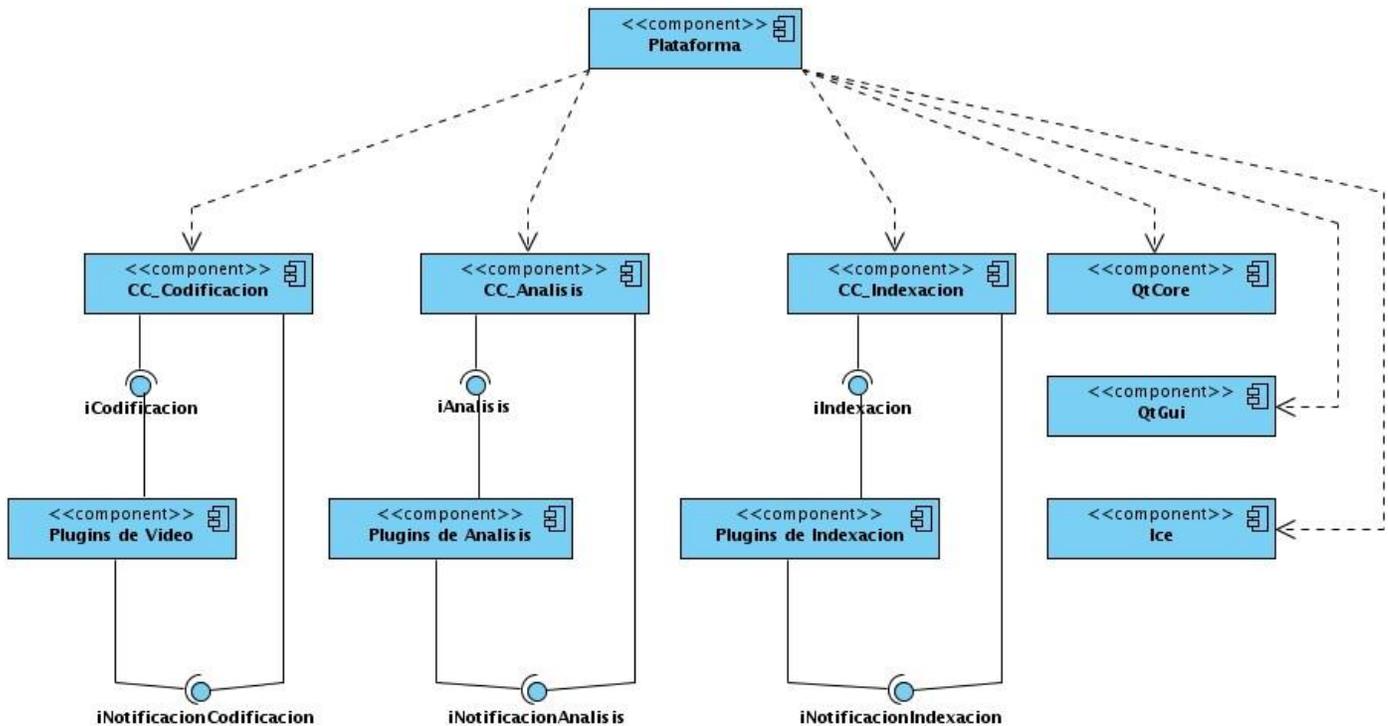


Figura 12. Diagrama de Componentes

4.6. Prueba del sistema propuesto

La prueba es un elemento crítico para la garantía de la calidad del software, es el proceso que permite verificar y revelar la calidad de este tipo de producto. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas. Con la realización de estas pruebas se pretende encontrar y documentar los defectos que puedan afectar la calidad del sistema desarrollado y validar los requisitos que debe cumplir el mismo y a su vez verificar si estos fueron implementados correctamente.

Capítulo 4: Construcción de la solución propuesta

Las pruebas a llevar a cabo sobre el sistema desarrollado responden al método de prueba de caja negra utilizando específicamente la técnica de partición equivalente, cuyo propósito es evaluar los requisitos funcionales del programa. Se realiza para verificar el comportamiento del componente sin tener en cuenta cómo se implementa. En el caso particular de esta investigación, se aplicarán estas pruebas a los casos de uso considerados críticos para el desarrollo del sistema.

4.6.1. Diseño de Caso de Prueba Caja Negra. Partición equivalente

Nombre del Caso de Uso: Analizar Video

Descripción General: El caso de uso se inicia cuando se envía la orden de analizar un material con el fin de que cumpla con el perfil establecido para los materiales originales. Implícitamente se le extrae la duración de material

Condiciones de Ejecución: Se debe haber ordenado el análisis a un nuevo material en el sistema.

Secciones a probar en el Caso de Uso.

Nombre de la Sección	Escenarios de la Sección	Descripción de la Funcionalidad	Flujo Central
SC1: Analizar Video	EC 1.1: Analizar Video exitosamente	<ol style="list-style-type: none">1. El sistema carga los datos y procede a realizar el análisis del material.2. El sistema verifica que el material cumple con las especificaciones del perfil enviado.3. El sistema extrae la duración del material.4. El sistema notifica el estado de ejecución del proceso al monitor de procesos.5. El sistema notifica la culminación de la ejecución del proceso al monitor de procesos.	Ejecutar la acción "Añadir". Seleccionar el material a analizar. Ejecutar la acción "Aceptar".

Capítulo 4: Construcción de la solución propuesta

	EC 1.2: Analizar Video fallido	Si el material no cumple con las especificaciones del perfil, se notifica el error correspondiente.	Ejecutar la acción "Añadir". Seleccionar el material a analizar. Ejecutar la acción "Aceptar".
--	--------------------------------	---	---

Tabla 9. DCP Caja Negra. CU Analizar Video

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Dirección Origen	Campo de Texto	No	El usuario introducirá en este campo la dirección correspondiente al material que desea analizar

Tabla 10. Descripción de Variables del CU Analizar Video

Sección	Resultado esperado	Resultado de la prueba
Analizar Video	El sistema debe mostrar las características del material analizado siempre que se entren datos válidos a la variable. En caso de datos inválidos se notifica el error: "El material especificado no se encuentra".	Satisfactorio

Tabla 11. Resultados de la Prueba al CU Analizar Video

Nombre del Caso de Uso: Codificar Video

Descripción General: El caso de uso se inicia cuando se ordena la ejecución de un proceso de codificación a un formato de mayor compresión o cuando se desea codificar a un formato especificado. Dentro de sus funcionalidades implícitas se encuentra la codificación de fragmentos de materiales.

Condiciones de Ejecución: Se debe haber ordenado la ejecución de un proceso de codificación de un material tanto a un formato de mayor compresión como a un formato específico.

Capítulo 4: Construcción de la solución propuesta

Secciones a probar en el Caso de Uso.

Nombre de la Sección	Escenarios de la Sección	Descripción de la Funcionalidad	Flujo Central
SC1: Codificar Video	EC 1.1: Codificar Video exitosamente	<ol style="list-style-type: none"> 1. El sistema carga los datos recibidos y comienza la ejecución del proceso, almacenando su resultado en el directorio destino. 2. Notifica el estado de ejecución del proceso cargado. 3. Notifica la finalización del proceso. 	Ejecutar la acción "Añadir". Seleccionar el material a convertir. Ejecutar la acción "Aceptar". Ejecutar la acción "Convertir". Seleccionar parámetros de codificación. Ejecutar acción "Comenzar".
	EC 1.2: Codificar Video fallido	En caso de error durante la ejecución de proceso se notifica el mismo.	Ejecutar la acción "Añadir". Seleccionar el material a convertir. Ejecutar la acción "Aceptar". Ejecutar la acción "Convertir". Seleccionar parámetros de codificación. Ejecutar acción "Comenzar".

Tabla 12. DCP Caja Negra del CU Codificar Video

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Dirección Origen	Campo de Texto	No	El usuario introducirá en este campo la dirección correspondiente al material que desea Codificar.
2	Dirección Destino	Campo de Texto	No	El usuario introducirá en este campo la dirección de salida correspondiente al material que desea Codificar. El sistema asumirá una dirección de salida por defecto en caso de que el usuario no especifique una.

Capítulo 4: Construcción de la solución propuesta

3	Formato	Lista Desplegable	No	El usuario seleccionará un formato de codificación.
4	Códec de Video	Lista Desplegable	No	El usuario seleccionará un códec de video.
5	Códec de Audio	Lista Desplegable	No	El usuario seleccionará un códec de audio.
6	Resolución	Lista Desplegable	No	El usuario seleccionará una resolución de video.
7	Bitrate de Video	Campo de Texto	No	El usuario introducirá el valor numérico correspondiente al bitrate de video.
8	Bitrate de Audio	Lista Desplegable	No	El usuario seleccionará un valor numérico correspondiente al bitrate de audio.
9	Tiempo de Inicio	Campo de Texto	No	El usuario introducirá un valor correspondiente al instante de inicio de codificación del video.
10	Tiempo de Fin	Campo de Texto	No	El usuario introducirá un valor correspondiente al instante final de codificación del video.

Tabla 13. Descripción de Variables del CU Codificar Video

Sección	Resultado esperado	Resultado de la prueba
Codificar Video	El sistema debe cargar correctamente los datos de codificación (ver CU Analizar Video). El sistema debe mostrar el porcentaje de codificación del material y notifica la finalización del proceso.	Satisfactorio

Tabla 14. Resultados de la Prueba al CU Codificar Video

Capítulo 4: Construcción de la solución propuesta

Nombre del Caso de Uso: Elaborar Resumen Visual de Video

Descripción General: El caso de uso se inicia cuando se ordena la ejecución de un proceso de extracción de resumen visual de video

Condiciones de Ejecución: Se debe haber ordenado la ejecución de un proceso de extracción de resumen visual de video por la inclusión de un nuevo material al sistema.

Secciones a probar en el Caso de Uso.

Nombre de la Sección	Escenarios de la Sección	Descripción de la Funcionalidad	Flujo Central
SC1: Elaborar Resumen Visual de Video	EC 1.1: Elaborar Resumen Visual de Video exitosamente	<ol style="list-style-type: none">1. El sistema carga los datos recibidos y comienza la ejecución del algoritmo, almacenando su resultado en el directorio destino.2. Notifica el estado de ejecución del proceso cargado.3. Notifica la finalización del proceso.	Ejecutar la acción "Añadir". Seleccionar el material a realizar resumen visual. Ejecutar la acción "Aceptar". Ejecutar la acción "Resumen". Introducir el valor del porciento de sensibilidad para el resumen. Ejecutar la acción "Aceptar".
	EC 1.2: Elaborar Resumen Visual de Video fallido	En caso de error durante la ejecución del proceso se notifica el mismo.	Ejecutar la acción "Añadir". Seleccionar el material a realizar resumen visual. Ejecutar la acción "Aceptar". Ejecutar la acción "Resumen". Introducir el valor del porciento de sensibilidad para el resumen. Ejecutar la acción "Aceptar".

Tabla 15. DCP Caja Negra. CU Elaborar Resumen Visual de Video

Capítulo 4: Construcción de la solución propuesta

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Dirección Origen	Campo de Texto	No	El usuario introducirá en este campo la dirección correspondiente al material que desea realizar el resumen visual.
2	Dirección Destino	Campo de Texto	No	El usuario introducirá en este campo la dirección de salida correspondiente al material que desea realizar el resumen visual.
3	Sensibilidad	Campo de Texto	No	El usuario introducirá en este campo el valor correspondiente al porcentaje de sensibilidad del resumen.

Tabla 16. Descripción de Variables del CU Elaborar Resumen Visual de Video

Sección	Resultado esperado	Resultado de la prueba
Elaborar Resumen Visual de Video	El sistema debe cargar correctamente los datos para realizar el resumen. El sistema debe mostrar el porcentaje de ejecución del resumen del material y notifica la finalización del proceso.	Satisfactorio

Tabla 17. Resultados de la Prueba al CU Elaborar Resumen Visual de Video

4.7. Conclusiones

Durante el capítulo se expusieron los diferentes diagramas de clases correspondientes al diseño y se identificó una arquitectura adecuada para el desarrollo del sistema, que contribuyó a la adecuada implementación del mismo. Se abordaron las generalidades para el desarrollo de la aplicación, teniendo en cuenta la arquitectura seleccionada y los componentes utilizados, constatando que la aplicación de patrones de diseño asegurará la escalabilidad y robustez de la solución. Además, con la creación del modelo de despliegue se obtuvo una vista de la distribución física del sistema logrando una mayor explotación de sus funcionalidades. Se realizaron pruebas de caja negra al sistema arrojando resultados satisfactorios.

Conclusiones

Conclusiones

En la presente investigación se profundizó en los conocimientos sobre los procesos de codificación e indexación de video, procedentes de fuentes diversas en diferentes publicaciones digitales logrando organizarlos en función de la fundamentación del objetivo propuesto.

Se logró desarrollar la Plataforma de codificación e indexación de video con tecnologías libres, con la característica de ser más eficiente por la utilización de un lenguaje de programación cercano al hardware y de técnicas de programación multihilos que posibilita una mejor distribución de los recursos. Además se asegura la escalabilidad del sistema utilizando el mecanismo de plugins, lo cual permite actualizar el software de manera dinámica.

Las pruebas realizadas a la aplicación con el objetivo de validar los requisitos funcionales determinados para su desarrollo, arrojaron resultados favorables y reveló un sistema sólido y eficiente.

Los resultados de este trabajo posibilitan la factibilidad tecnológica de la utilización de la Plataforma de codificación e indexación de video en el departamento de Señales Digitales de la UCI.

Recomendaciones

En aras de lograr un manejo eficaz de la aplicación desarrollada se recomienda la elaboración de un manual de ayuda para el usuario, así como capacitar al personal para la utilización del software.

Desarrollar plugins de codificación y de análisis basados en la utilización directa de librerías de códecs, con el propósito de elevar la eficiencia y agregar nuevas funcionalidades al sistema.

Desarrollar plugins de indexación que permitan la elaboración de resumen visual de los videos codificados con las técnicas de codificación más recientes.

Bibliografía

1. **Almeida Maldonado, Enrique y Baby González, Ronald.** *Sistema de Catalogación de Medias*. [Trabajo de diploma para optar por el título de Ingeniero en Informática] s.l. : Universidad de las Ciencias Informáticas, 2009.
2. **Ministerio de Educación y Ciencia de España.** <http://ares.cnice.mec.es>. [En línea] [Citado el: 26 de 05 de 2011.] <http://ares.cnice.mec.es/informes/13/index.htm>.
3. **Manzanera, Antoine.** *Indexación de imágenes y video*. Bogotá : ENSTA – ParisTech, 2008.
4. **Tedial.** Tedial. [En línea] 2010. [Citado el: 13 de 12 de 2010.] <http://www.tedial.com>.
5. **Harmonic Inc.** Rhozet. [En línea] 2011. [Citado el: 16 de 12 de 2010.] <http://www.rhozet.com>.
6. **JHONATAN MINA, F. M., ANDRES FDO PEREZ.** *Resumen Ejecutivo RUP*. 2008.
7. **I. Jacobson, G. Booch, J. Rumbaugh.** *El Lenguaje Unificado de Modelado. Manual de Referencia*. s.l. : Pearson Education S.A., 2000.
8. **Kendall, K.** *Análisis y Diseño de Sistemas*. 2005.
9. UML, BPMN and Database Tool for Software Development. [En línea] Visual Paradigm. [Citado el: 22 de 03 de 2011.] <http://www.visual-paradigm.com>.
10. Instituto Universitario de Tecnología del Estado Portuguesa. [En línea] Instituto Universitario de Tecnología del Estado Portuguesa. [Citado el: 22 de 03 de 2011.] <http://www.iutep.tec.ve/>.
11. **Herrera Diaz, Francisco Paul.** Paulhdccp's Blog. [En línea] 12 de 04 de 2010. [Citado el: 22 de 03 de 2011.] <http://paulhdccp.wordpress.com>.
12. **Corporation, Nokia.** Qt - Cross-platform application and UI framework. [En línea] Nokia Corporation, 2008-2011. <http://qt.nokia.com>.
13. **Bucefalo.** *Bucefalo*. [En línea] 2008. <http://bucefalo.com.mx>.
14. **Martínez Juan, Francisco Javier.** *Guía de Construcción de Software en Java con Patrones de Diseño*. Oviedo : ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA, 2000.
15. Enterprise Architect - Herramienta de Modelado UML. [En línea] Sparx Systems Pty Ltd, 2000-2007. [Citado el: 23 de 03 de 2011.] http://www.sparxsystems.com.ar/resources/tutorial/physical_models.html.

Bibliografía

16. MCC. [En línea] Universidad Nacional Autónoma de México. [Citado el: 23 de 03 de 2011.] <http://uxmcc1.iimas.unam.mx/~cursos/Objetos/Cap7/cap7.html>.
17. **I. Jacobson, G. Booch, J. Rumbaugh.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Pearson Education S.A., 2000.
18. **García de Jalón, Javier y Ignacio Rodríguez, José.** *Aprenda C++ como si estuviera en primero*. San Sebastián : Escuela Superior de Ingenieros Industriales, 1998.
19. **Sanjuán, Olga Pérez.** *Recordando la Historia: El concepto "televisión" en sus orígenes*. 2007. pág. 160.
20. **Jack, Keith.** *Dictionary of Video and Television Technology*. United States of America : Elsevier Science., 2002. 1-878707-99-X.
21. **Beck, K.** *Extreme Programming Explained. Embrace Change*. s.l. : Pearson Education, 1999.
22. **Schildt, Herbert.** *C++ guía de autoenseñanza*. España : McGRAW-HILL, 1995. ISBN: 0-07-882025-1.
23. **Lanzillotta, Analía.** Definición de Leguajes de Programación. *Master Magazine*. [En línea] Analía Lanzillotta. [Citado el: 05 de 03 de 2011.] <http://www.mastermagazine.info/termino/5560.php>.
24. Glosario Electrotécnico y Electrónico. [En línea] [Citado el: 8 de 12 de 2010.] http://www.natureduca.com/tecno_gloselec_d01.php.
25. IBM. [En línea] 2002. [Citado el: 25 de 1 de 2011.] <http://publib.boulder.ibm.com/iseres/v5r2/ic2924/index.htm?info/apis/concept.htm>.
26. Los Archivos AVI. [En línea] [Citado el: 02 de 02 de 2011.] http://www.megaservice.com.ar/Info_espe/Los%20archivos%20AVI.htm.
27. MPEG home page. [En línea] 2010. [Citado el: 02 de 12 de 2010.] <http://www.chiariglione.org/mpeg/>.
28. MPEG:ORG. [En línea] 2010. [Citado el: 02 de 12 de 2010.] <http://www.mpeg.org/MPEG/mpeg-pointers-and-resources/>.
29. MPlayer The Movie Player. [En línea] 2008. [Citado el: 02 de 12 de 2010.] <http://www.mplayerhq.hu/design7/info-es.html>.
30. Mundo DIVX Martes. [En línea] [Citado el: 02 de 12 de 2010.] <http://www.mundodivx.com/glosario.php>.
31. Ogg. [En línea] 2008. [Citado el: 02 de 12 de 2010.] <http://www.xiph.org/ogg/>.

Bibliografía

32. Servicios Informáticos. [En línea] [Citado el: 29 de 11 de 2010.] <http://www.ctisa.com/diccionario.asp.B60920501>.
33. xiph.org. [En línea] 2008. [Citado el: 03 de 12 de 2010.] <http://www.xiph.org/vorbis/>.
34. **AXIS**. Técnicas de compresion. [En línea] 2004. [Citado el: 02 de 12 de 2010.] www.axis.com/es/documentacion/Tecnicas%20de%20compresion%20de%20video.pdf -.
35. **FFmpeg**. FFmpeg. [En línea] 2009. [Citado el: 03 de 12 de 2010.] www.ffmpeg.org.
36. **Suárez Pérez, Jean Michael y Estrada Velazco, Aylin**. *Sistema de Captura e Indexación de Video*. [Trabajo de diploma para optar por el título de Ingeniero en Informática] s.l. : Universidad de las Ciencias Informáticas, 2009.
37. **XTREAM**. XTREAM. Sistemas de información Global. [En línea] XTREAM, 2010. [Citado el: 26 de 05 de 2011.] <http://www.xtreamsig.com/index.php>.
38. **The Digital FAQ**. The Digital FAQ. [En línea] 2010. [Citado el: 26 de 05 de 2011.] <http://www.digitalfaq.com/guides/video/introduction-encode-convert.htm>.
39. **Amado, José Augusto López**. *Subsistemas de Administración y Catalogación del Sistema de Captura y Catalogación de Medias (CCM)*. [Trabajo de diploma para optar por el título de Ingeniero en Informática] s.l. : Universidad de las Ciencias Informáticas, 2010.
40. **Alcover Alcover, Adrian Alberto**. *Ampliación del Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria*. [Trabajo de diploma para optar por el título de Ingeniero en Informática] Universidad de las Ciencias Informáticas : s.n., 2010.

Anexos

Anexo I. Descripción del Caso de Uso Reportar Nodo

Caso de Uso:	Reportar Nodo	
Actores:	Proceso Automático Plataforma	
Resumen:	El caso de uso se inicia cuando se inicia la ejecución de la aplicación	
Precondiciones:	Debe haberse iniciado la aplicación	
Referencias	RF5	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1.1. Se envían los datos del nodo (estación de trabajo) hacia el Gestor de Procesos.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
<i>Prototipo de Interfaz</i>		
Poscondiciones	El nodo queda registrado en el Gestor de Procesos	

Tabla 18. Descripción Textual del CU Reportar Nodo

Anexo II. Descripción del Caso de Uso Actualizar Estado Plataforma

Caso de Uso:	Actualizar Estado Plataforma	
Actores:	CUS Analizar Video, CUS Codificar Video, CUS Elaborar Resumen Visual de Video	
Resumen:	El caso de uso se inicia cuando el proceso en ejecución emite una señal de actualización	
Precondiciones:	El proceso debe estar ejecutándose	
Referencias	RF1, RF2, RF3, RF4	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	

Anexos

	1. El caso de uso se inicia al iniciarse cualquier procesamiento sobre los materiales audiovisuales. En la medida que se ejecuta el proceso la plataforma deberá notificar el estado y porcentaje de ejecución del mismo hacia el monitor de procesos. Culmina el caso de uso.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1. En caso de error durante la ejecución del proceso se notifica el mismo.
Prototipo de Interfaz	
Poscondiciones	El Gestor de Procesos recibe los datos de la notificación

Tabla 19. Descripción Textual del CU Actualizar Estado Plataforma

Anexo III. Descripción del Caso de Uso Cancelar Proceso

Caso de Uso:	Cancelar Proceso
Actores:	CUS Analizar video, CUS Codificar video, CUS Elaborar resumen visual de video.
Resumen:	El caso de uso se inicia cuando se ha ordenado cancelar un proceso que se encontraba en ejecución.
Precondiciones:	Se debe haber ordenado cancelar la ejecución de un proceso ya iniciado.
Referencias	RF2, RF3, RF4
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia al enviarse la señal de cancelación de un proceso en ejecución que contendrá el identificador del proceso en cuestión. Estos datos pueden ser seleccionados por un usuario desde la interfaz de usuario, en la que deberá seleccionar la opción "Cancelar" (A) o pueden ser enviados por el monitor de procesos.	1.1. El sistema cancela la ejecución del proceso que se le ha ordenado cancelar. 1.2. El sistema elimina los ficheros temporales creados en la dirección de destino del proceso durante su ejecución hasta cancelarse.

Anexos

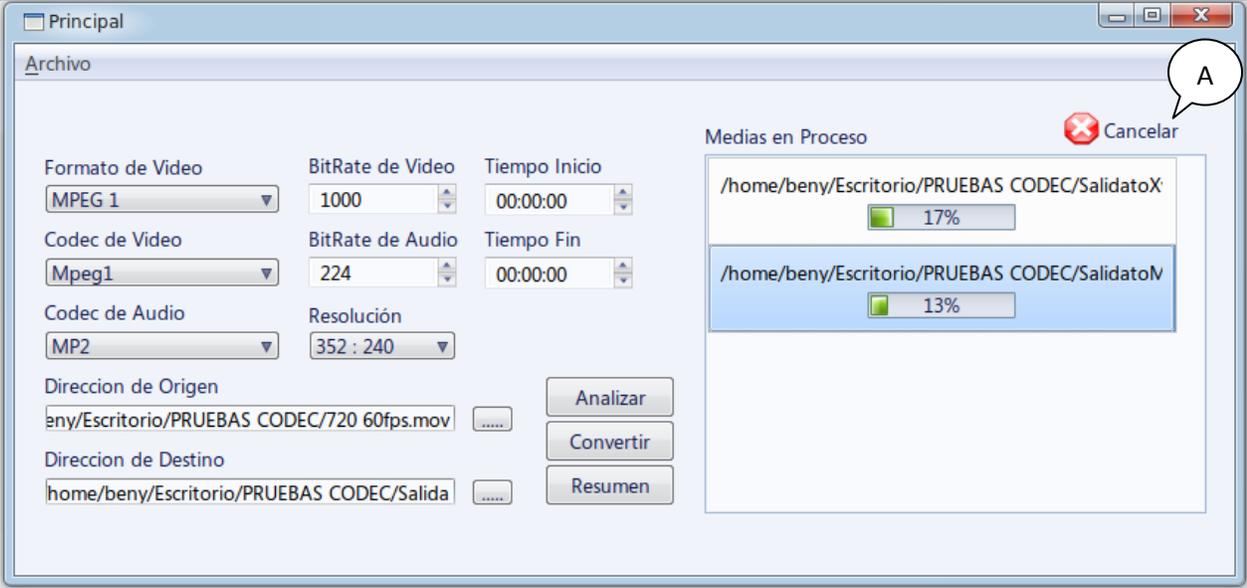
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
<i>Prototipo de Interfaz</i>	
	
Poscondiciones	Se cancela la ejecución de un proceso y se eliminan los ficheros temporales que se habían creado durante la ejecución del mismo.

Tabla 20. Descripción Textual del CU Cancelar Proceso

Glosario

Formato de compresión de video y sonido digital: Forma de presentación de la información del video y el sonido digital en menor espacio que en su estado original sin comprimir. La característica principal de estos formatos es la compresión con pérdida de calidad e información, aunque generalmente no es perceptible al ojo y el oído humano. Luego de comprimido un fichero, nunca se llegará a la información inicial que contenía.

Códec: Dispositivo o programa capaz de codificar y decodificar en un signo o una corriente de datos digitales.

Códec de video: Es un conjunto de algoritmos utilizados para la **codificación** y/o **decodificación** de datos. El proceso de codificación se basa en la compresión del archivo, mientras la decodificación se hace al mismo tiempo que se reproduce el archivo. Los algoritmos de codificación de video están diseñados principalmente para la reducción del tamaño respetando al máximo la calidad del material. En función de esto y de la finalidad para la cual será codificado es definido el formato definitivo del fichero, e implícitamente, los algoritmos a utilizar en su compresión.

Codificación de video: La codificación o transcodificación, es la conversión de un sistema de datos a otro distinto. La codificación de video es la compresión de un video en un formato distinto del original o la conversión con uno u otro códec si se tratase del mismo formato.

Indexación de video: Técnicas que se aplican a un archivo audiovisual cuyo objetivo es adjuntar a una imagen o un vídeo una serie de descriptores del contenido, con el fin de medir la similitud con los descriptores correspondientes a una consulta. (3)

Compresión de Video: Es una forma de compresión de datos diseñada para reducir el tamaño de los archivos de video.

Digitalización: Proceso de cuantificar una señal analógica, convertirla y representarla en forma digital. La digitalización de las señales analógicas consigue un significativo aumento de la velocidad de proceso, así como de la precisión durante la reproducción.

Glosario

Media: Película, imagen o cualquier otro material audio visual que requiere de un uso especial de equipamiento para visualizarlo.

Plugin: Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

Video: Es la tecnología de la captación, grabación, procesamiento, almacenamiento, transmisión y reconstrucción por medios electrónicos digitales o analógicos de una secuencia de imágenes que representan escenas en movimiento.

Multimedia: El término multimedia se utiliza para referirse a cualquier objeto o sistema que utiliza múltiples medios de expresión (físicos o digitales) para presentar o comunicar información.

Multiplataforma: Se utiliza el término para denominar a los programas, lenguajes de programación u otra clase de software que pueden brindar sus prestaciones funcionando sobre diversas combinaciones de hardware y software.

Servidor: Software u ordenador que provee servicios a otros programas o equipos denominados clientes.

Streaming: Consiste en la distribución de audio o video por Internet.

Metadatos: Literalmente “sobre datos”, son datos que describen otros datos.

Aplicación: Es una clase de programa informático creado para facilitar al usuario un determinado tipo de trabajo. Esto lo caracteriza frente a otros programas como los sistemas operativos, las utilidades y los lenguajes de programación.

Audio: Señal, técnica o sistema electrónico relacionado con la grabación, reproducción y transmisión del sonido que normalmente se encuentra acotado al rango de frecuencias audibles para los seres humanos.

Copia proxy: Copia que se le hace a un material audiovisual la cual tiene una mayor compresión y menor resolución para facilitar trabajos de procesamiento de video.