



Universidad de las Ciencias Informáticas

Facultad 6

**COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA
AGORAV**

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

Autor:

Claudia Beatriz González Alonso

Tutores:

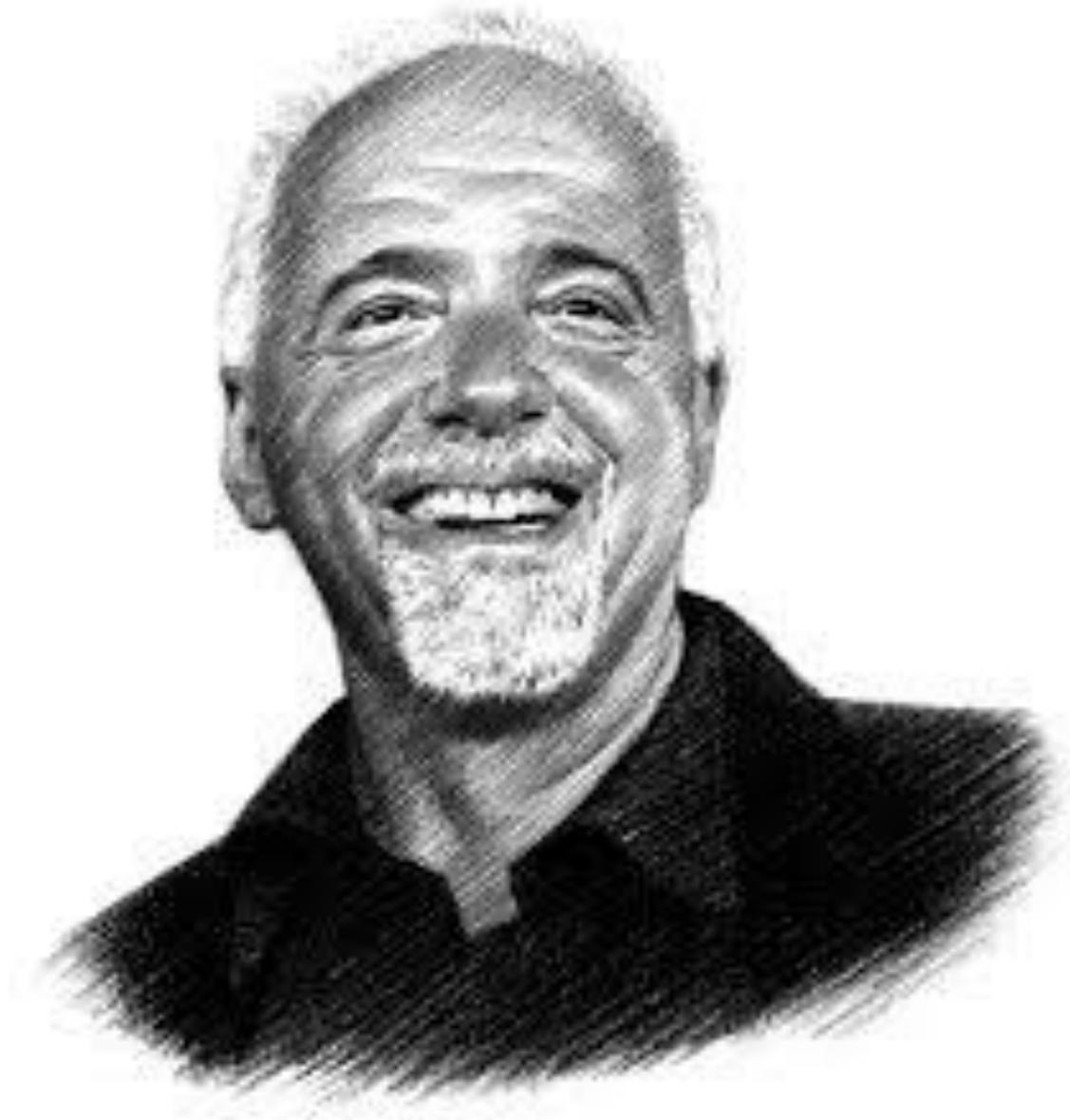
M.Sc. Gerdys Ernesto Jiménez Moya

Ing. Miguel Morciego Varona

Ing. Heidys Dueñas Pérez

La Habana, junio de 2016

“Año 58 de la Revolución”



“Cuando quieres realmente una cosa, todo el Universo conspira para ayudarte a conseguirla”.

Paulo Coelho

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

DEDICATORIA

Le dedico mi trabajo de diploma a mi mamá y a mis papás por su amor y su apoyo incondicional. A mis abuelos y abuelas, por estar siempre cuando los necesito y por su amor incondicional, en especial a mi Mami Nancy por ser una mamá para mí, apoyarme y solapar todas mis travesuras. A mis hermanos, en especial a hermana Patry, por ser más que una, por ser mi vida. Además a mis tíos: Yaimilyn, Roberto Iván y Ernesto. En fin a toda mi familia. Los amo.

AGRADECIMIENTOS

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo, Claudia Beatriz González Alonso, con carnet de identidad 93022434371 soy la autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste, firmamos la presente declaración jurada de autoría en La Habana a los ___ días del mes de junio del año 2016.

Claudia Beatriz González Alonso
Autor

M.Sc. Gerdys Ernesto Jiménez Moya
Tutor

Ing. Miguel Morciego Varona
Tutor

Ing. Heidys Dueñas Pérez
Tutor

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

RESUMEN

La información audiovisual constituye uno de los recursos más aprovechados por las nuevas generaciones de ordenadores y redes de telecomunicación. Una de las aplicaciones que fue concebida para la publicación de audiovisuales es XILEMA AGORAV, la cual fue desarrollada por la Universidad de las Ciencias Informáticas (UCI). Esta plataforma presenta varias limitantes, por ejemplo, no permite la descarga en varios tipos de formatos ni la transcodificación de videos, la descarga de un material audiovisual ante una suspensión eventual del sistema queda interrumpida y además no admite realizar la gestión de subtítulos. El objetivo de este trabajo es desarrollar un componente de *software* para el Sistema XILEMA AGORAV que erradique las limitaciones de codificación y decodificación de los materiales audiovisuales. El resultado obtenido es un componente que se desarrolló empleando la metodología AUP-UCI, el lenguaje de programación C++ y el marco de trabajo para el desarrollo Qt 5.4.1. Las nuevas funcionalidades incorporadas permiten el aumento de la satisfacción de los usuarios de la plataforma, así como de las oportunidades de negocio.

Palabras clave: codificación, decodificación, publicación de audiovisuales.

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

ABSTRACT

Audiovisual information is one of the most used resources by the new generations of computers and telecommunications networks. One application that was designed for the publication of audiovisual is XILEMA AGORAV, which was developed at the University of Information Science (UCI). This platform has several limitations, for example, it does not allow downloading in various types of formats or transcoding videos, the audiovisual material download before a possible suspension of the system is broken and it also does not permit to perform subtitles management. The aim of this work is to develop a software component for the System XILEMA AGORAV that eradicates the limitations of encoding and decoding of audio-visual materials. The result is a component that was developed using the AUP-UCI methodology, the C ++ programming language and framework for the development Qt 5.4.1. The new features allow increasing the satisfaction of users of the platform, as well as business opportunities.

Keywords: *encoding, decoding, publication of audiovisual.*

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA SOBRE LA CODIFICACIÓN Y DECODIFICACIÓN DE AUDIOVISUALES	5
1.1. Introducción	5
1.2. Elementos de la codificación y decodificación de audiovisuales	5
1.3. Ventajas de la codificación digital en los audiovisuales	6
1.4. Principales limitaciones de la codificación y decodificación audiovisual	7
1.5. Importancia y principales aplicaciones de la codificación y decodificación	8
1.6. XILEMA AGORAV	9
Características fundamentales	9
Arquitectura de XILEMA AGORAV	9
Componentes fundamentales de XILEMA AGORAV	10
1.7. La codificación y decodificación en XILEMA AGORAV	11
1.8. Soluciones actuales de codificación y decodificación de audiovisuales	11
"ProMedia Carbon"	11
Tedral MPM	12
Plataforma de codificación de video para el Sistema de Captura y Catalogación de Medias	13
1.9. Metodología de desarrollo de <i>software AUP, en su versión UCI</i>	14
1.10. Tecnologías a utilizar en el desarrollo de la solución	14
Marco de trabajo para el desarrollo	15
Lenguaje de programación	15
Entorno de desarrollo integrado	16
Lenguaje de modelado	17
Herramienta para el modelado	18
1.11. Sistema Gestor de Bases de Datos	18
1.12. Otras herramientas y tecnologías	19
Bibliotecas	19
1.13. Conclusiones parciales	20
CAPÍTULO II: PROPUESTA DE SOLUCIÓN	21
2.1. Introducción	21

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

2.2.	Modelo del Dominio	21
2.3.	Descripción de los Requisitos de <i>Software</i>	22
	Requisitos funcionales	22
	Requisitos no funcionales	23
2.4.	Modelo del sistema	24
	Definición de los Actores del sistema	24
	Patrones de Casos de Uso del Sistema	24
	Diagrama de Casos de Uso del Sistema	24
	Descripción textual de los Casos de Uso del Sistema	25
2.5.	Arquitectura de <i>software</i>	28
	Estilo arquitectónico	28
	Patrón arquitectónico	29
	Patrones de diseño	29
	Patrones Generales de Asignación de Responsabilidades	30
	Patrones del Grupo de los Cuatro	31
2.6.	Modelo de Diseño	31
	Diagrama de Clases del Diseño	31
2.7.	Conclusiones parciales	33

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS -----34

3.1.	Introducción	34
3.2.	Estándar de codificación	34
3.3.	Modelo de Implementación	35
	Diagrama de Componentes	35
3.4.	Modelo de Despliegue	36
3.5.	El Proceso de Pruebas	37
	Pruebas de Integración	38
	Análisis del rendimiento del componente por peticiones	39
3.6.	Conclusiones parciales	41

CONCLUSIONES -----42

RECOMENDACIONES -----43

BIBLIOGRAFÍA -----44

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

GLOSARIO DE TÉRMINOS -----	49
ANEXOS -----	51
ANEXO I: Descripción textual de los Casos de Uso del sistema-----	51

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

ÍNDICE DE FIGURAS

Fig. 1. Componentes fundamentales de XILEMA AGORAV (Martell, y otros, 2015).	10
Fig. 2. Diagrama de clases del modelo de dominio.	21
Fig. 3. Diagrama de Casos de Uso del sistema	25
Fig. 4. Diagrama de clases del diseño.	33
Fig. 5. Diagrama de Componentes	36
Fig. 6. Modelo de Despliegue	37
Fig. 7. Integración del Transcoder con el Subsistema Web.....	38

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

ÍNDICE DE TABLAS

Tabla 1. Descripción general del actor del sistema	24
Tabla 2. Descripción del Caso de Uso “Capturar petición del Subsistema <i>Web</i> ”	25
Tabla 3. Descripción del Caso de Uso “Codificar media”	26
Tabla 4: Pruebas de carga y estrés.....	39
Tabla 5.Descripción del Caso de Uso “Determinar porcentaje de uso del CPU”	51
Tabla 6.Descripción del Caso de Uso “Extraer metadatos”	51
Tabla 7. Descripción del Caso de Uso “Notificar información al Subsistema <i>Web</i> ”	52

INTRODUCCIÓN

En los años treinta, tras la aparición del cine sonoro, surge en Estados Unidos el término audiovisual. Sin embargo, no es hasta la década de los cincuenta, en Francia, cuando esta expresión comienza a ser usada para referirse a las técnicas de difusión simultáneas. A partir de entonces el concepto se amplía y el término se sustantiva (Vega, 2012). En el área de los medios de comunicación de masas se habla de lenguaje audiovisual y comunicación audiovisual. El término audiovisual significa la integración e interrelación plena entre lo auditivo y lo visual para producir una nueva realidad o lenguaje (Kocher, 2013) (López, 2013). Este sector es una de las industrias más complejas y creativas que existen, al mismo tiempo que ha estado sujeta a transformaciones y cambio en los últimos años. La era digital ha dado inicio a una de las mayores revoluciones en los métodos de distribución y producción de medios, se crea así un sector novedoso para el mundo.

En el caso concreto de la televisión, la nueva sociedad de la información y el entretenimiento hace que el medio se enfrente a la digitalización de los sistemas de producción y distribución. Además junto a la creación de la web, se abren nuevas vías de desarrollo para el medio en que se asocia al ordenador con la concepción del uso de la multimedia interactiva, propiciado por la implementación de los servicios *online* (Pérez, 2012) (Toffler, y otros, 2013)(Ferrer, 2013). También se plantea que la sinergia entre los medios de comunicación tradicionales y sus modelos *online*, como nuevo soporte para la difusión de los contenidos audiovisuales, comienzan a ser muy fructíferas y, tras el escepticismo inicial, la mayoría de los *broadcasters* o cadenas de televisión están presentes en la red.

Los sistemas de distribución *online*, son aquellos a los que se pueden acceder mediante un ordenador, la televisión o a través de los dispositivos móviles. Se encargan de rentabilizar el material audiovisual almacenado y facilitan la posibilidad de distribuir un alto número de producciones, que mediante los mecanismos tradicionales, resulta muy difícil su realización.

En la actualidad existen diversos tipos y modelos de dispositivos de reproducción de medias, por ejemplo como son los casos de los llamados MP3, MP4, teléfonos celulares, laptops, entre otros (Martí, y otros, 2011)(López, 2013). Debido a lo anterior, se hace necesario la utilización de programas capaces de convertir los videos a disímiles formatos, lo que provoca la necesidad de alcanzar un alto grado de compatibilidad para poder reproducirlos. Por ello es indispensable ajustar las dimensiones y las especificaciones de cada uno de los videos.

Estos programas utilizados para convertir los videos a distintos formatos, suelen contar con las herramientas necesarias para realizar la conversión (Múnera, y otros, 2014). Por ejemplo, en el caso del Gestor de Procesos de Medias, se utiliza un programa externo que funciona como herramienta para la conversión,

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

INTRODUCCIÓN

otro de los casos es cuando el *software* realiza la conversión por sí mismo y cuentan con bibliotecas que le permiten ejecutar este tipo de acción. En esta variante el usuario sólo tendrá que seleccionar el formato en que desea convertir el video y en qué dispositivo desea reproducirlo.

En el contexto de la informática, la conversión se refiere al proceso de transformación (mayormente conocido como codificación y decodificación) de datos. Este lleva de una representación concreta a otra el archivo multimedia y modifica los *bits* de un formato a otro, para lograr la interoperabilidad de aplicaciones o sistemas diferentes (Hernández, 2013). En términos menos técnicos, la conversión de datos puede ejemplificarse mediante la conversión de un fichero de texto desde una codificación de caracteres a otra.

Un ejemplo de este tipo de sistema, para la conversión, lo constituye XILEMA AGORAV, pues está concebida como plataforma para la publicación de audiovisuales en la *web*, con posibilidades para la captura, publicación, catalogación, consumo en vivo y bajo demanda de contenidos audiovisuales. Este producto es desarrollado en la Universidad de las Ciencias Informáticas (UCI), en el Centro Geoinformática y Señales Digitales (GEYSED). Dos de las posibilidades (publicación, consumo en vivo y bajo demanda) que brinda dicha plataforma están estrechamente relacionadas con los *codecs* (VP 3-9, Theora, MPEG 1-4, H. 264-265, DivX, X 264-265, Xvid, Windows Media Video) y los formatos de video (AVI, MPG, MOV, ASF, Ogg, OGM, RMVB, Matroska, entre otros); lo anterior evidencia sus múltiples posibilidades de interpretación y de uso variado.

Los usuarios de AGORAV, aun cuando está concebida la descarga de los materiales previamente publicados, solamente pueden descargar específicamente en el formato de video *webm* para el consumo posterior del audiovisual fuera de XILEMA AGORAV. Lo planteado anteriormente causa insatisfacción al usuario ante la imposibilidad de reproducir el material en dispositivos que no posean el *codec* (VP 8-9) necesario para su reproducción; los cuales deben codificar el audiovisual fuera de AGORAV y utilizar otra aplicación informática para su conversión.

Por otra parte, también genera inconformidad el consumo innecesario del espacio en el disco duro del usuario, al momento de descargar un audiovisual cuando solamente es de interés una parte de este. Igualmente, los administradores del sistema deben codificar el audiovisual completamente sin posibilidades para su edición y descomposición en fragmentos. Otra limitación está constituida por la pérdida de la descarga o codificación de un material audiovisual ante una suspensión eventual del funcionamiento de la plataforma, aun cuando esta sea de un espacio corto de tiempo. Lo anterior produce como consecuencia que el usuario deba comenzar manualmente el proceso de descarga o codificación una vez que se restablezca el servicio.

Para los administradores de la aplicación informática presenta otro conjunto de restricciones que impiden un desenvolvimiento ágil y acertado: el empleo de recursos de *hardware* de manera innecesaria debido al mecanismo de solicitud de peticiones implementado; la imposibilidad de manipular ficheros de subtítulos,

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

INTRODUCCIÓN

que los obliga a utilizar herramientas de terceros. Además de no permitir la definición de los recursos de *hardware* que se utilizan en la codificación o decodificación de un material audiovisual; lo cual repercute negativamente en el rendimiento del dispositivo de cómputo que se utilice, donde de esta manera se pueden obtener fallas de funcionamiento que provocan el reinicio o el cierre inesperado del proceso.

De acuerdo con los elementos antes expuestos se define el siguiente **problema de investigación** ¿Cómo erradicar las limitaciones en la codificación y decodificación de XILEMA AGORAV? A partir del problema propuesto se plantea como **objetivo general** desarrollar un componente de *software* para el Sistema XILEMA AGORAV que erradique las limitaciones de codificación y decodificación de los materiales audiovisuales.

La investigación se enmarca en el ámbito de la codificación y decodificación de audiovisuales lo cual constituye el **objeto de estudio**. Se define como **campo de acción**, precisión del objeto de estudio, la automatización de la codificación y decodificación de audiovisuales en XILEMA AGORAV.

Las siguientes **tareas de la investigación** se definen para dar cumplimiento al objetivo general:

- Caracterizar el proceso de codificación y decodificación de materiales audiovisuales.
- Valorar las soluciones existentes que responden al campo de acción de la investigación en alguna medida, sus limitaciones y fortalezas.
- Caracterizar las principales herramientas, tecnologías, lenguajes y metodologías a utilizar para la construcción de la propuesta de solución.
- Realizar el análisis y el diseño de la solución propuesta.
- Implementar la solución propuesta.
- Desarrollar el proceso de pruebas al módulo implementado.

La investigación utiliza una serie de métodos científicos que guían el proceso investigativo, los cuales se exponen a continuación:

Teóricos:

- **Histórico-Lógico:** el método histórico estudia la trayectoria real de los fenómenos y acontecimientos en el transcurso de su historia. El método lógico investiga las leyes generales del funcionamiento y desarrollo de los fenómenos (Zayas, 1997). Este método se emplea para analizar la evolución de los conceptos asociados a la codificación y decodificación de audiovisuales, lo cual permite formar una definición propia.
- **Analítico-Sintético:** permite la descomposición de un todo complejo en sus partes y cualidades. La síntesis, por su parte, establece la unión entre las partes, previamente analizadas y posibilita descubrir relaciones y características generales entre los elementos de la realidad (Zayas, 1997). Este método se emplea para la valoración de las soluciones existentes que responden al campo de

INTRODUCCIÓN

acción. Además, se utiliza en la selección de las herramientas y tecnologías que se emplean en el desarrollo de la solución propuesta.

Empíricos:

- **Análisis documental:** facilita dentro de la investigación científica el dominio de las técnicas empleadas para el uso de la bibliografía. Permite la creación de habilidades para el acceso a investigaciones científicas, reportadas en fuentes documentales de forma organizada. En la presente investigación se utilizan los procedimientos lógicos y mentales como análisis, síntesis, deducción e inducción; se realiza una recopilación adecuada de datos de varias fuentes para de esta forma permitir el planteamiento de nuevas interrogantes y la sugerencia de nuevas bibliografías para su consulta.

Estructura del documento

La presente investigación está estructurada en tres capítulos:

- Capítulo 1: En este capítulo se exponen los fundamentos teóricos relacionados con los conceptos fundamentales de la codificación y decodificación de audiovisuales, así como el análisis de las soluciones existentes en la actualidad que cuentan con módulos similares. Además, se describe y define la metodología de desarrollo que va a ser empleada, además de las herramientas a utilizar para el diseño e implementación del componente.
- Capítulo 2: En este capítulo se definen los artefactos del diseño y de la implementación de la solución. Se define la estructura del diseño de la aplicación desarrollada, para ello se representa el modelo de dominio, los requisitos funcionales y no funcionales, se describen los casos de uso del sistema y se expone la arquitectura del componente.
- Capítulo 3: En este capítulo se exponen los estándares de codificación utilizados en la implementación de la solución, el diagrama de componentes, el modelo de despliegue y los resultados de las pruebas realizadas a la aplicación para verificar su correcto funcionamiento de acuerdo a los requisitos establecidos.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA SOBRE LA CODIFICACIÓN Y DECODIFICACIÓN DE AUDIOVISUALES

1.1. Introducción

El presente capítulo expone los conceptos fundamentales que se abordan durante la investigación, realizándose una caracterización del objeto de estudio propuesto. Se desarrolla una breve caracterización de las principales herramientas de *software* que en alguna medida aportan información o características relevantes para la solución de la problemática. De igual manera se exponen las herramientas y tecnologías que se utilizan durante la construcción de la propuesta de solución; se documenta la metodología de *software* seleccionada y se describen las principales características de cada herramienta utilizada, además se enfatiza en los elementos de interés para la investigación.

1.2. Elementos de la codificación y decodificación de audiovisuales

La información multimedia es actualmente uno de los principales impulsores del desarrollo y diseño de las nuevas generaciones de ordenadores y redes de telecomunicación. Todo lo que hay detrás del “universo multimedia” es un gran negocio que mueve al año cantidades incalculables en inversiones, investigación, implantación de redes, generación de contenidos, diseño de equipos, servicios y aplicaciones. La información multimedia surge de la integración de distintos tipos de formatos o fuentes de información heterogéneos, tales como la voz, audio, textos, imágenes y videos (Póveda, y otros, 2010).

Los métodos y técnicas usados para la codificación de cada uno de los tipos de fuentes de información pueden ser completamente distintos unos de otros, pero aun así pueden aunarse y formar un bloque de información más completo y homogéneo (López, y otros, 2011).

La codificación digital puede tener como entrada, tanto información analógica como digital y en función a la pérdida de calidad en el proceso puede ser (Sandoval, 2012):

- Codificación sin pérdidas: no hay pérdida de información. Se consigue una compresión alrededor de la mitad del tamaño original.
- Codificación con pérdidas: se pierde información en el proceso pero, no obstante a ello, se consiguen valores de compresión considerablemente altos (MPEG-1 comprime alrededor de 15 veces el tamaño).
- Normalmente se usan conjuntamente, primero con pérdidas y luego sin pérdidas.

La codificación audiovisual se divide en dos procesos (Suárez, y otros, 2013):

- Codificación de fuente: convierte la información multimedia en un fichero o flujo de bits comprimido.

- Codificación de canal: convierte la información digital transmitir o almacenar a las señales adecuadas al medio usado para la transmisión o almacenaje.

Es usual que ambos se implementen conjuntamente para mejorar el comportamiento global. Cuando se agrupa un codificador y un decodificador para proporcionar una comunicación a dos vías se denomina *codec* (Suárez, y otros, 2013).

En la actualidad la codificación/decodificación de audiovisuales se utiliza para lograr la visualización del material audiovisual en diferentes medios, se emplea para ello una amplia variedad de *codec*, tanto para video como para audio, o ambos (Criollo, y otros, 2010). Este proceso, generalmente se realiza en el dispositivo en el cual se pretende visualizar o desde un dispositivo externo a través de la red. Existen gran cantidad de métodos de compresión, ya sean paramétricos o basados en la forma o información de la señal, ya sea en el dominio temporal, espacial o frecuencia.

Pero la codificación digital está sujeta a gran número de compromisos, los cuales dependen del valor o rango de diversos parámetros correspondientes a cada **método de compresión** concreto (Morales Morante 2014), (Schuster, y otros, 2013).

- Eficiencia de codificación: es la relación entre el número de bits necesario para cada elemento multimedia antes y después de la compresión.
- Retardo de compresión: es más delicado en aplicaciones multimedia pero no para la difusión.
- Complejidad: el coste computacional en operaciones por segundo o también el coste de la circuitería de codificación.
- Calidad de la información decodificada: ya sea subjetiva, relativa a capacidades perceptuales humanas u objetiva (SNR).
- *Bit rate* constante o *bit rate* variable: algunos medios o tipos de información pueden conseguir una mayor eficiencia al utilizar para cada escena o fragmento el *bit rate* necesario y mantener la calidad perceptual.
- Flexibilidad: es la capacidad de un flujo de bits comprimidos de poder ser accedidos y decodificados en cualquier punto.

1.3. Ventajas de la codificación digital en los audiovisuales

Al codificar un material audiovisual se obtienen una serie de ventajas como son (Jiménez, y otros, 2015), (Schuster, y otros, 2013):

- Compresión: la eliminación de redundancias, así como las técnicas de enmascaramiento permiten que los flujos de datos digitales pueden ser enviados con mucho menos ancho de banda. Además, a menores necesidades de almacenamiento, los dispositivos se hacen más económicos.

- Flexibilidad de reproducción, además de ajuste de velocidad de reproducción, así como el salto a cualquier punto deseado del flujo de información.
- Copia sin pérdida de calidad: las copias son exactas a los originales.
- Flexibilidad de manipulación, donde se incluye la facilidad del troceado de la información y montaje sin pérdida de calidad, fácil mejora de elementos gráficos y sonoros, mezcla e inserción sin bordes de imágenes infográficas en entornos reales, entre otros.
- Facilidad de búsqueda y recuperación: facilidad de incorporar información de contenido y metadatos que pueden ser obtenidos eficientemente y asociados a los objetos multimedia. Posibilidad de buscar en los contenidos por aspectos tales como parecidos entre objetos, patrones de colores, ritmos y otros.
- Composiciones multimedia: viabilidad sin deterioro de la calidad de integrar cualquier tipo de medio junto con información textual, de control, entre otras, todo ello perfectamente sincronizado en tiempo y espacio.
- Facilidad de intercambio de piezas y composiciones multimedia a través de medios físicos o redes de comunicaciones.

1.4. Principales limitaciones de la codificación y decodificación audiovisual

Existen diferentes limitaciones al utilizar la codificación/descodificación en un audiovisual. Para evaluar y determinar la calidad de video existen dos grupos de métodos denominados métodos objetivos y métodos subjetivos (Lage, y otros, 2013).

Los métodos objetivos buscan a través del desarrollo matemático (algoritmos de evaluación) determinar las diferencias físicas (espaciales y temporales) contenidas en una imagen decodificada, en contraste con la imagen original (imagen sin codificar) (García, 2010). Las principales limitaciones de este método son:

- El método se enfoca en evaluar la información perceptual de movimiento contenida en cada pixel, limitándose a calcular las diferencias entre las secuencias originales y las comprimidas. Esto impide que el método pueda ser utilizado en tiempo real y tecnológicamente representa un alto costo (TVAD, 2011) pues necesita de la imagen original para poder realizar su evaluación.
- Solo es eficiente cuando los errores se comportan como ruido adicional, es decir, que no se encuentra relacionado con la señal, entonces el método no puede diferenciar entre pocos errores de gran amplitud que son percibidos por el observador y muchos errores con baja amplitud que son imperceptibles por el observador (TVAD, 2011) (Lage, y otros, 2013).

Los **métodos subjetivos** se dedican a realizar la evaluación de la calidad de la imagen llevada a cabo por un observador humano. Lo que se logra medir es el impacto que produce la secuencia presentada. La

dificultad que se evidencia en este método radica en no poder contemplarse todos los parámetros que inciden en la calidad de video como (Richardson, 2012):

- La concordancia entre la señal original de referencia y la de prueba.
- Las variaciones de frecuencia entre cuadros o en la resolución.
- El tiempo y los recursos necesarios, el coste y la imposibilidad de automatizar el proceso.
- Las condiciones de visualización (esto es debido a que una prueba realizada en un ambiente cómodo y relajado obtendrá una calificación superior a la misma prueba realizada en un ambiente menos confortable).

De manera general existen tres limitaciones fundamentales (López, y otros, 2013):

- Variedad de formatos tanto de entrada como salida.
- Dependencia total del *codec* necesario para la conversión.
- Alto consumo de recurso de *hardware*.

1.5. Importancia y principales aplicaciones de la codificación y decodificación

Los sistemas de codificación resolvieron desde hace décadas la problemática de enviar mucha información a la velocidad necesaria para sistemas de tiempo real o casi real, así como para el almacenamiento, pero en el campo audiovisual no fue hasta la década de los 90 cuando tuvieron su gran explosión, en la que se abrieron varias líneas tanto por iniciativas privadas (QuickTime, WM) como estandarizadas (MPEG entre otras) (Sánchez, 2015).

Estas líneas han tenido una gran repercusión en los contenidos, que se han podido multiplicar y convertir en multiformato con facilidad. La codificación de vídeo es de primordial importancia para la mayoría de las aplicaciones multimedia, en especial para las aplicaciones que manejan flujos de video en tiempo real y de alta calidad (Fernández, y otros, 2015).

En los últimos años se ha producido un notable avance en las tecnologías digitales, especialmente en aquellas que involucran el procesamiento del video, se destacan la televisión digital, audiovisuales en formato DVD, video bajo demanda, video-conferencias, video-teléfonos, entre otros. El auge y difusión masivos de estas tecnologías dependen en gran medida de la capacidad de compresión que se pueda alcanzar mientras se mantiene un determinado nivel de la calidad del video, además de conservar cierto ancho de banda para las aplicaciones cuya transmisión se realiza en redes, como el video bajo demanda. Actualmente hay múltiples aplicaciones en las que la codificación de vídeo es necesaria, algunos ejemplos son los teléfonos móviles de última generación, la difusión de vídeo en Internet y la televisión de alta definición (García, y otros, 2006).

Existen diferentes proyectos para los cuales es imprescindible el trabajo con las medias y a su vez la transcodificación de estas. Algunos de ellos se encuentran ubicados en la Universidad de las Ciencias Informáticas (UCI), en la Facultad 6, específicamente en el Centro de desarrollo Geoinformática y Señales Digitales (GEYSED) como son: Video Web y Sistema Gestor de Procesos de Medias XILEMA AGORAV.

1.6. XILEMA AGORAV

XILEMA AGORAV es una aplicación informática desarrollada y comercializada por la Universidad de las Ciencias Informáticas. Está diseñada para la publicación, difusión y consumo de contenidos audiovisuales en la Web, con un conjunto de posibilidades adicionales de valor agregado al producto. Dirigida fundamentalmente a redes privadas como universidades, bibliotecas, centros culturales, productoras de música y televisión, centros de negocios y convenciones, hoteles, y otros, AGORAV, además, posee la capacidad de posicionarse en Internet como un espacio de intercambio y consumo de materiales audiovisuales.

Características fundamentales

XILMA AGORAV presenta las siguientes características y potencialidades:

- Introduce nuevas formas de interactividad entre los usuarios al permitirles sugerirse materiales audiovisuales publicados, de esta forma se posibilita la socialización de la información y se emplean diferentes métodos de recomendación.
- Permite las posibilidades de la Web2.0, elemento asociado con un fenómeno social, basado en la interacción que se logra a partir de diferentes aplicaciones y recursos ubicados en la web, que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la red de redes.
- Posibilita la ponderación y voto por los contenidos publicados, lo que establece un sistema de valoración de los materiales ubicados en el entorno.
- Posibilita el comentario de materiales en las propias publicaciones donde contribuye a la retroalimentación sobre el parecer de los usuarios del sistema sobre los materiales publicados.
- Viabiliza la creación de un ambiente personal a la preferencia del usuario con elementos de personalización a partir del historial de navegación o de los votos otorgados a los materiales publicados lo que contribuye a hacer sentir a gusto a los usuarios.

Arquitectura de XILEMA AGORAV

XILEMA AGORAV es un sistema con una arquitectura modular, configurable a cualquier entorno de despliegue relacionado con la organización, catalogación, gestión y recuperación de los materiales audiovisuales digitalizados. AGORAV es capaz de aprovechar al máximo los medios tecnológicos de última generación y a la vez, adaptarse a entornos menos sofisticados en dependencia de las necesidades y recursos de sus usuarios.

Los módulos de la Fig. 1 funcionan de manera independiente lo cual aumenta considerablemente las posibilidades de configuración en el momento del despliegue y se logra un ajuste de este a diferentes entornos tecnológicos.

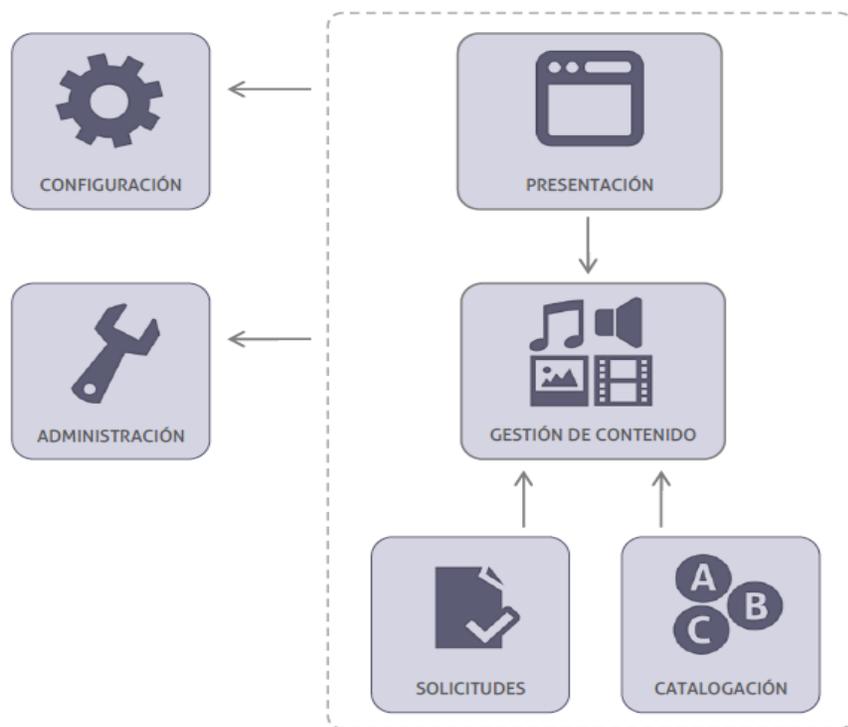


Fig. 1. Componentes fundamentales de XILEMA AGORAV (Martell, y otros, 2015).

Componentes fundamentales de XILEMA AGORAV

Configuración: desde este módulo la plataforma será configurable de acuerdo al ambiente donde se utilizará, podrán ser configurados aspectos como puntos de publicación de *streaming*, puntos de almacenamiento de materiales, uso de aplicaciones terceras y definición de los elementos de comunicación.

Administración: este módulo se encarga del manejo de los usuarios, los roles y los permisos en el sistema. Permite, además, la creación de secciones y categorías (sub-secciones) de contenido con sus respectivas informaciones generales y de interés, las cuales pueden también configurarse desde el módulo. Ofrece un

conjunto de reportes de estado de los servidores *web*, de bases de datos y de *streaming*, reportes de actividad de usuarios, horarios pico de acceso al sistema, contenidos más votados o visitados entre otros.

Presentación: desde una agradable, intuitiva e interactiva interfaz de usuario es posible visualizar las medias, conformar listas de reproducción a partir de las publicaciones, compartir las listas con otros usuarios de la plataforma, comentar las publicaciones y votar las mismas. Además, se puede reproducir materiales audiovisuales en un reproductor embebido en la aplicación o en un reproductor externo y descargar los materiales publicados.

Gestión de contenido: a través de este módulo se adiciona a la plataforma el contenido que se gestionará y pondrá a disposición de los diferentes tipos de usuario posteriormente. Permite la creación de secciones para publicación y de tipologías de contenido personalizadas por tipos de usuario así como la publicación de artículos de contenido asociados a las medias y la creación de señales de televisión en vivo. Este módulo, además, brinda un conjunto de funcionalidades asociadas a la codificación de medias, determinación de fotogramas clave y recuperación de archivos almacenados.

Solicitudes: los usuarios podrán solicitar los materiales audiovisuales que se gestionan en AGORAV aun cuando estos no se encuentren en publicación. A través del módulo pueden realizar búsquedas de estos materiales así como descargar el material solicitado una vez publicado.

1.7. La codificación y decodificación en XILEMA AGORAV

De acuerdo a su objetivo, AGORAV pretende proporcionar accesibilidad completa al contenido audiovisual, de manera que una vez que el contenido es creado, pueda ser consumido y mantenerse accesible para todos sus usuarios desde cualquier terminal cliente (también se incluyen los dispositivos móviles) conectada a la red de AGORAV.

Pero esa diversidad de terminales y redes implican diversas calidades de servicio así como disponibilidad del sistema, al mismo tiempo que suponen distintos formatos de presentación de los diversos tipos de información que pueden formar parte de un contenido audiovisual. En la actualidad AGORAV realiza la transcodificación solo en el momento de la conversión de las medias y únicamente al formato *webm*.

1.8. Soluciones actuales de codificación y decodificación de audiovisuales

Existen varias compañías que se han dedicado al desarrollo de aplicaciones para la codificación y descodificación de materiales audiovisuales. En este epígrafe se analizan las soluciones con mayor aceptación en el mercado con el propósito de identificar características, posibles escenarios de implementación y/o elementos reutilizables.

"ProMedia Carbon"

ProMedia Carbon, anteriormente conocido como “*Carbon Coder*”, componente de la familia *ProMedia* de soluciones de flujos de trabajo optimizados de producción y para múltiples pantallas admite todos los formatos SD y HD más importantes. El mismo puede manejar todas las operaciones cruciales de transcodificación, incluidas las de conversión SD/HD, conversión de PAL/NTSC, inserción de logotipos, conversión de espacios de color, corrección de color, extracción de subtítulos para sordos y muchas otras. Una API abierta permite la creación de flujos de trabajo personalizados o aplicaciones de otros proveedores, lo que aumenta aún más la flexibilidad líder del sector de *ProMedia Carbon* (Kael, 2014).

Elementos identificados de interés para la investigación

- Permite la gestión de subtítulos.
- Permite varios procesos simultáneos.
- Brinda un SDK de interacción – a través de un fichero xml predefinido (Kael, 2014).

Esta última característica no ofrece ventajas pues la problemática de la investigación actual gira en torno a permitir una configuración del video de salida editable en cada caso y no estático.

Elementos que generan limitantes para la investigación:

- No soporta el codec Vp8 para convertir a formato webm.
- Su adquisición no es libre.
- Solo se puede utilizar en los Sistemas Operativos: Windows XP / Windows Vista / Windows 7.
- No tiene en cuenta el consumo de recursos.

Tedial MPM

Tedial 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. Al mismo tiempo, esta importante empresa produce aplicaciones informáticas para el tratamiento de contenido multimedia, entre ellas se encuentran TD MPM y TD *Indexer* (Russell, 2014).

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 al usar la mayoría de los equipos existentes y mantener lo mejor de los procesos de trabajo del cliente (Russell, 2014).

Elementos identificados de interés para la investigación

- Permite varios procesos simultáneos.
- Permite monitorizar el estado de la codificación.
- Presenta compatibilidad de Archivos.
- MPM ofrece la plataforma de interacción para gestionar la tecnología de terceros al utilizar XML, IP, *plugins* y API según sea necesario (incluso si muchos de los sistemas implicados son a menudo incompatibles ya sea por razones técnicas o comerciales).
- Los administradores de medios pueden definir los parámetros clave, como los recursos que se dedican a tareas críticas, el número de operaciones simultáneas y el momento en que el procesamiento no urgente se llevará a cabo.

Elementos que generan limitantes para la investigación

- Su adquisición no es libre.
- Debido a la compatibilidad de archivo no se permiten restricciones de formatos propietarios y envoltorios.

Plataforma de codificación de video para el Sistema de Captura y Catalogación de Medias

Ejecuta de forma automática los procesos de codificación e indexación de video con tecnologías libres mediante la "Plataforma de codificación e indexación de video para el Sistema de Captura y Catalogación de Medias". El sistema brinda la funcionalidad de analizar un video con el objetivo de verificar que el formato del material se corresponde con el establecido. Además brinda la funcionalidad de codificar al formato de video en baja resolución para *streaming* establecido o a un formato de video y la funcionalidad de cortar un fragmento de video a partir de tiempos especificados por el usuario (Cabrera, y otros, 2011).

Elementos identificados de interés para la investigación

- Presenta varias bibliotecas de codificación como: *Ffmpeg*, *mencode* y *Ffmpeg2Theora*, de los cuales en la construcción de la solución se usará *Ffmpeg*.
- Permite la comunicación por protocolos que cargan a través de *plugins* como XMLRPC e ICE.
- Permite la ejecución de varios procesos simultáneos (Cabrera , y otros, 2011).

Elementos que generan limitantes para la investigación

- No tiene implementado un *plugins* para la comunicación con *WebSockets* el cual es usado en la plataforma *web*.
- No gestiona subtítulos.
- No permite dar prioridad a un proceso sobre otro, lo cual afecta el rendimiento del ordenador ya que no se puede limitar el uso de recursos (Cabrera , y otros, 2011).

Otros Convertidores:

- *Total Video Converter* (López, 2015).
- *Freemake Video Converter* (Ferri, 2015).
- *Video to Video Converter* (Ramírez, 2015).
- *Format Factory* (Bihan, 2016).

Estos convertidores son libres de costo y soportan numerosos formatos de archivos, pero no son una solución para el problema ya que no brindan una interfaz de comunicación, lo que hace imprescindible la interacción de un usuario con el *software* directamente. Además, no ofrecen funcionalidades que sean a través de servicios, exigencia indispensable para dar solución a la problemática.

1.9. Metodología de desarrollo de *software AUP*, en su versión UCI

Una metodología de desarrollo de *software* representa un marco de trabajo que tiene entre sus funciones guiar, planificar, estructurar, controlar, manipular y dirigir el proceso de desarrollo de sistemas de información. Surge ante la necesidad de trabajar mediante el uso de procedimientos, técnicas, herramientas y documentos durante el desarrollo de *software* (Pressman, 2010).

Las metodologías se clasifican en dos tipos de procesos: ágiles o ligeras y pesadas o tradicionales. Las ágiles tienen como principios el trabajo en equipo como arma fundamental; el avance del trabajo enmarcándose solamente en los elementos necesarios que este exige. Se enfocan en la constante interacción con el cliente haciéndolo parte del equipo de trabajo y en la posibilidad de cambiar todo lo que debe ser cambiado de forma que se alcance la mayor fiabilidad y calidad en el producto que se desarrolla. (Pressman, 2008)

Debido a que el arquitecto del proyecto XILEMA AGORAV definió como metodología de desarrollo de *software AUP*, en su versión UCI, específicamente el escenario No. 2: Caso de Uso del Sistema (CUS); el componente desarrollado sigue esta, para facilitar su escalabilidad y mantenimiento, de manera que todos los artefactos se puedan integrar correctamente con la documentación del proyecto. La misma describe de manera simple la forma de desarrollar el componente con el uso de técnicas ágiles, como el modelado ágil. El uso de dicha metodología permite una descripción concisa utilizando poca documentación y brinda la posibilidad de utilizar cualquier conjunto de herramientas de Ingeniería de *Software* Asistida por Computadora (por sus siglas en inglés, CASE). Los artefactos ingenieriles que se generan para el desarrollo del componente son: Diagrama de clases del modelo de dominio, Diagrama de Casos de Uso del sistema, Diagrama de Clases del diseño, Diagrama de Componentes, Modelo de Despliegue, Descripción general del actor del sistema y Descripción de los casos de usos definidos.

1.10. Tecnologías a utilizar en el desarrollo de la solución

Marco de trabajo para el desarrollo

Un marco de trabajo o *framework* consiste en un conjunto de bibliotecas, herramientas y normas a seguir que ayudan a desarrollar aplicaciones. Está compuesto por varios segmentos/componentes que interactúan los unos con los otros (Lafosse, 2010). Entre las características que presenta se encuentran la reutilización de código y estandarización del desarrollo, por mencionar algunas. La elección del marco de trabajo forma parte de la estrategia de una empresa ya que será determinante para la calidad, la productividad y la durabilidad de los proyectos (Lafosse, 2010). Dentro de la comunidad de *software* libre existen disímiles marcos de trabajo entre los cuales se encuentra Qt.

Marco de Trabajo: Qt 5.4.1

La tecnología Qt consiste en un *framework* multiplataforma usado en el desarrollo de *software*. El término multiplataforma (*cross-platform*), hace referencia a la habilidad que posee un elemento *software* para poder ser ejecutado en diferentes arquitecturas de computadores o sistemas operativos, como ocurre en el caso de este proyecto.

Para desarrollar la solución de la investigación se escoge Qt por varios motivos entre los que se encuentra hace uso del estándar C++, pero también puede ser usado en otros lenguajes de programación entre los que se encuentran: PHP, Ruby, Python, Java y C++ utilizado de forma nativa. Entre las principales plataformas que soporta se encuentran: Windows, Mac, Linux, Solaris, por mencionar algunos (Digia, 2013). Presenta un módulo *qmake* el cual genera archivos que permiten que el programa sea ejecutado en cualquier entorno de trabajo sin necesidad de modificar el código. Utiliza el mecanismo de señales y ranuras para comunicar los objetos de una aplicación. Las aplicaciones que son desarrolladas en él presentan buena respuesta y uso aceptable en cuanto a memoria.

Lenguaje de programación

Un lenguaje de programación consiste en todos los símbolos, caracteres y reglas de uso que permiten a las personas "comunicarse" con las computadoras. Existen varios cientos de lenguajes y dialectos de programación diferentes. Algunos se crean para una aplicación especial, mientras que otros son herramientas de uso general más flexibles que son apropiadas para muchos tipos de aplicaciones. En todo caso los lenguajes de programación deben tener instrucciones que pertenecen a las categorías ya familiares de entrada/salida, cálculo/manipulación de textos, lógica/comparación y almacenamiento/recuperación. Los mismos se puede clasificar en: lenguaje de alto nivel o lenguaje de bajo nivel. Los lenguajes de programación deben cumplir dos propiedades fundamentales: no presentar ambigüedad (cada elemento del lenguaje debe tener un único significado) y ser computable (los programas deben ser ejecutados por un ordenador) (Berlanga, y otros, 2000).

En el año 1995, se incluyeron algunas bibliotecas de funciones al lenguaje C y con base en ellas, se pudo en 1998 definir el estándar de C++. Este lenguaje proporciona orientación a objetos, esta versión combina la flexibilidad y el acceso de bajo nivel de C con las características de la programación orientada a objetos como abstracción, encapsulación y ocultación. Una consideración importante es que, hasta la publicación de este estándar, C había sido mayormente un subconjunto estricto del C++. Era muy sencillo "actualizar" un programa de C hacia C++ y mantener ese código compilable en ambos lenguajes. Sin embargo, el nuevo estándar agrega algunas características que C++ no admite (Mazaeda, 2014).

Lenguaje de programación: C++

C++ es un lenguaje de programación que se utiliza para realizar cualquier tipo de aplicación independientemente del entorno o sistema operativo. Permite el trabajo con varios paradigmas de programación entre los que se encuentran: programación estructurada, programación orientada a objetos, programación modular, abstracción de datos y programación genérica (Hernández, y otros, 2012). C++, cuando es necesario, permite prever la duración de ejecución de una aplicación (Guérin, 2015). Entre los compiladores C++ más utilizados está GNU g++ que se encuentra bajo la licencia GPL y es prácticamente indispensable para compilar el núcleo de Linux. Este compilador es uno de los más utilizados para programar C++ en GNU/Linux.

El lenguaje C++ es seleccionado ya que al estar estandarizado permite compilar un mismo código fuente en diversas plataformas. Una característica importante es que es un lenguaje rápido debido al hecho de ser compilado y no interpretado. El trabajo con punteros permite hacer referencias a espacios en memoria y no copias innecesarias de la información en dicha ubicación, lo cual evita que se cargue la memoria innecesariamente.

Entorno de desarrollo integrado

Un Entorno de Desarrollo Integrado (del inglés *Integrated Development Environment*, IDE), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.

IDE: QtCreator 3.1.1

QtCreator es un IDE creado por Trolltech, multiplataforma, diseñado para hacer que el desarrollo en C++ de la aplicación Qt sea más rápido y fácil. En el año 2008 Nokia compra Qt a Trolltech y luego en el 2011 tras su alianza con *Microsoft* decide vender la licencia comercial de Qt a Digia (Zugasti, 2014).

QtCreator no quiere ser un reemplazo de *Eclipse ni Visual Studio*, debido a que es un IDE creado por Trolltech para el desarrollo de aplicaciones con las bibliotecas Qt. Los sistemas operativos que soporta en forma oficial son (Zugasti, 2014):

- GNU/Linux 2.6.x, para versiones de 32 y 64 bits con Qt 4.x instalado. Además, hay una versión para Linux con gcc 3.3.
- Mac OS X 10.4 o superior, requiere Qt 4.x.
- Windows XP, Vista y 7, requiere el compilador *MinGW* y Qt 4.4.3 para *MinGW*.

Al ser un IDE, contiene un editor de texto que le permite (Zugasti, 2014):

- Escribir código y formato.
- Anticipar lo que se va a escribir y completar el código.
- Visualización en línea de errores y mensajes de advertencia.
- Navegar semánticamente por las clases, funciones y símbolos.
- Recibir ayuda sensible de las clases, funciones y símbolos.

Entre los aspectos por los que se escoge se encuentra su característica de ser multiplataforma, el cual es el entorno integrado de desarrollo del marco de trabajo seleccionado. El hecho de poseer un avanzado editor de código C++ con un gran completamiento de código que facilita la programación y brinda alta compatibilidad con el lenguaje seleccionado, es una ventaja que se tiene en cuenta. Una característica a destacar y que influye en la selección es que el uso de la memoria es menor respecto a otros IDE, como por ejemplo *Eclipse* o *NetBeans*.

Lenguaje de modelado

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML se utiliza para modelar distintos tipos de sistemas: sistemas de *software*, sistemas de *hardware*, y organizaciones del mundo real. UML ofrece diagramas en los cuales modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos de negocio.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Componentes para modelar componentes.

- Diagramas de Implementación para modelar la distribución del sistema.

UML, al prescribir una notación estándar, permite que un diseño orientado a objetos, previamente modelado con cualquiera de la docena de metodologías populares, lo que causa a los revisores tener que aprender las semánticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí, pueda ser entendido por diferentes diseñadores que modelen diferentes sistemas. (Company Overview of Popkin Software & Systems Inc, 2015).

Herramienta para el modelado

Una herramienta CASE se define como un *software* que se utiliza para ayudar a las actividades del proceso del *software* como la ingeniería de requerimientos, el diseño, el desarrollo de programas y las pruebas (Sommerville, 2010).

Este tipo de herramienta incluye editores de diseño, diccionarios de datos, compiladores, depuradores, herramientas de construcción de sistemas, entre otros. La presente tecnología facilita el proceso de desarrollo del *software* lo que aumenta la productividad y reduce el tiempo que se emplea. Proporciona ayuda al proceso de desarrollo del *software* donde se automatizan algunas de sus actividades (Sommerville, 2010).

Visual Paradigm para UML 8.0

Visual Paradigm for UML es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de *software* y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un *software* (Paradigm, 2015).

Algunas de las ventajas que presenta esta herramienta son (Sierra, 2006):

- Disponibilidad en múltiples plataformas: Esta característica es importante pues se necesita de una herramienta que sea compatible con los sistemas GNU/Linux y Windows.
- Documenta todo el trabajo desarrollado sin tener que recurrir a otras herramientas.
- Generación de código: Entre los diferentes lenguajes para los que realiza esta herramienta los diagramas a código se encuentra C++, el cual es el lenguaje de programación escogido para realizar la implementación del módulo.

Por lo antes expuesto se demuestra que esta herramienta CASE cuenta con las características que se necesitan para ser escogida.

1.11. Sistema Gestor de Bases de Datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de

datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

Sistema Gestor de Bases de Datos: PostgreSQL 9.4

PostgreSQL es un Sistema Gestor de Bases de Datos relacional y orientado a objetos, desarrollado en la Universidad de California. Es un gestor de Bases de Datos de código abierto, brinda un control de concurrencia multi-versión que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación (Cerón , 2010).

Es usado en la presente investigación debido a que la base de datos de XILEMA AGORAV usa PostgreSQL 9.4. Además de que posee características significativas del motor de datos, entre las que se pueden incluir las subconsultas y los valores por defecto. El código fuente se encuentra disponible sin costo. Está disponible para 34 plataformas con la última versión estable. Posee una integridad referencial e interfaces nativas para lenguajes como C, C++, PHP y PYTHON. Funciona en todos los sistemas operativos como *Linux* y *Windows*. Debido a la liberación de la licencia, PostgreSQL se puede usar, modificar y distribuir de forma gratuita para cualquier fin, ya sea privado, comercial o académico.

1.12. Otras herramientas y tecnologías

Bibliotecas

***Ffmpeg* 2.5.8 para la codificación e incrustación de metadatos**

Ffmpeg es una colección de *software* libre que puede grabar, convertir (transcodificar) y hacer *streaming* de audio y vídeo. Incluye *libavcodec*, una biblioteca de *codecs*. *Ffmpeg* está desarrollada en GNU/Linux, pero puede ser compilado en la mayoría de los sistemas operativos, donde se incluye *Windows*. *Ffmpeg* está liberado bajo una licencia GNU (*Lesser General Public License 2.1+*) o GNU (*General Public License 2+*) (en dependencia de las bibliotecas que estén incluidas). Los desarrolladores recomiendan utilizar el último *snapshot* de *SubVersion* ya que mantienen constantemente una versión estable (Biurrun, 2009).

Ffmpeg es utilizado en la presente investigación porque programa bastante sencillo y muy fácil de usar, orientado tanto a personas con conocimientos avanzados como usuarios novatos. Es capaz de elegir el códec con sólo escribir la extensión. Por ejemplo, *Ffmpeg* usará x264 si se elige .mp4, mpeg4 si se elige .avi, VP8 si se elige .webm.

***MedialInfo* 0.7.82 para extraer metadatos**

MedialInfo es un pequeño pero poderoso programa que puede dar acceso a toda la información localizada en archivos de audio y video. Con este programa se puede localizar, desde información fácilmente accesible

acerca de propiedades, hasta los detalles más precisos acerca de codificadores específicos y la manera en que el archivo fue creado (Team, 2016).

MediaInfo ofrece dos maneras de controlar la recopilación de datos, este es uno de los aspectos por los que se escoge este programa para la extracción de metadatos. Los principiantes van a apreciar la presencia de una interfaz gráfica de todas las funciones que puede mostrar todo lo que deseen, pero los usuarios profesionales pueden usar una interfaz de comandos mucho más poderosa que puede funcionar con scripts personalizados, extraer información de grandes archivos multimedia, y volcar los datos en bases de datos de tu elección. La aplicación muestra fácilmente toda la información disponible en cinco categorías básicas de metadatos – General, Video, Audio, Texto y Capítulos. Los formatos de archivos soportados son numerosos, que cubren todos los codificadores más populares que son utilizados por serios recolectores de datos multimedia y usuarios generales de PC (Team, 2016). Estas notables ventajas condujeron a la selección de dicha biblioteca como herramienta para la obtención de metadatos.

1.13. Conclusiones parciales

La codificación y decodificación de audiovisuales es un proceso clave para elevar los índices de aceptación y las posibilidades de comercialización de la aplicación informática XILEMA AGORAV, debido al impacto en la calidad que ellos representan. El análisis de las herramientas existentes que realizan la codificación y decodificación de audiovisuales permitió identificar un conjunto de funcionalidades candidatas a ser incluidas dentro de la propuesta de solución, al mismo tiempo que corroboraron la necesidad de la presente investigación. La definición del conjunto de herramientas y tecnologías satisface las necesidades en el desarrollo, garantiza una mayor usabilidad de la aplicación al mismo tiempo que impulsa los principios de soberanía tecnológica impulsados por la Universidad. Todos los elementos seleccionados se adecuan a las políticas de calidad e implementación de la UCI.

CAPÍTULO II: PROPUESTA DE SOLUCIÓN

2.1. Introducción

En el presente capítulo se especifica la fase inicial de análisis del sistema de acuerdo con la metodología propuesta, se describe el modelo de dominio, la propuesta de requisitos funcionales y no funcionales, y finalmente se define la arquitectura del sistema y los patrones de diseño utilizados.

2.2. Modelo del Dominio

De acuerdo con lo que establece (Jacobson, y otros, 2000), el modelo del dominio representa las clases más significativas del sistema y su objetivo fundamental es facilitar la comprensión del contexto y de los requisitos del sistema, a través de la utilización del diagrama de clases del dominio en notación UML.

Se emplea un modelo de dominio pues no se logra determinar procesos de negocio que presenten fronteras bien establecidas donde se puedan identificar claramente las personas que lo inician y los beneficiados con cada proceso. Además, no se conoce quiénes son las personas que desarrollan las actividades de cada uno de estos procesos. Otro aspecto a tener en cuenta para su selección es que las clases del dominio son obtenidas de la base del conocimiento de varios expertos del dominio o de conocimiento asociado con sistemas similares como son los analizados en las soluciones existentes.

Descripción general del modelo de dominio

Un usuario realiza una petición a la plataforma de descargar, publicar o reproducir una media.

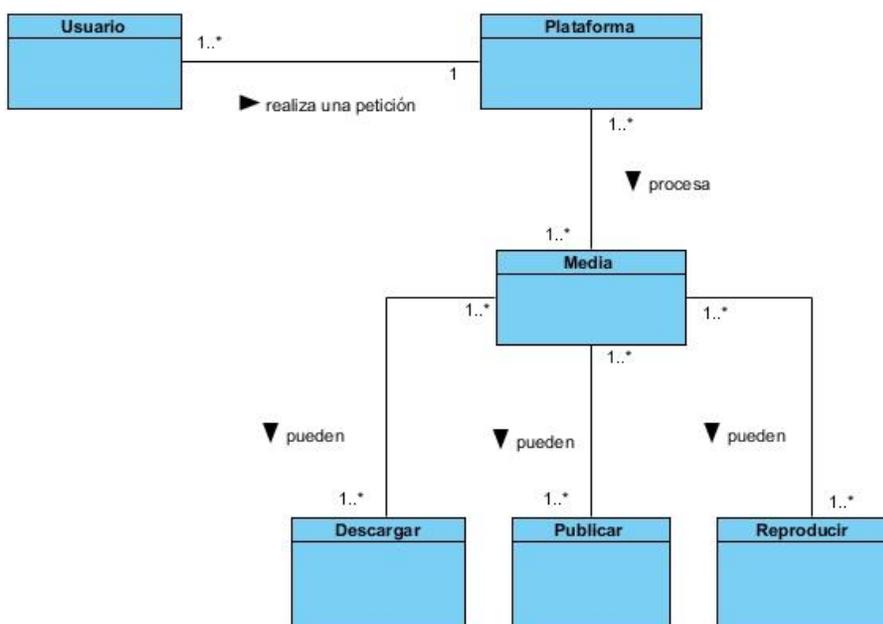


Fig. 2. Diagrama de clases del modelo de dominio.

Descripción de las clases

Usuario: Representa a la persona que va a realizar la descarga, publicación o reproducción de una media.

Plataforma: Sistema que se encarga del conjunto de procesos (descarga, publicación o reproducción de una media) y relaciones que se establecen entre ellos.

Media: Material audiovisual que se va a ser procesado.

Descargar: Descarga de la media solamente en formato *.webm*.

Publicar: Publicación de la media solamente en formato *.webm*.

Reproducir: Reproducción de la media solamente en formato *.webm*.

2.3. Descripción de los Requisitos de Software

Según Pressman en su libro “Ingeniería de *Software*. Un enfoque práctico”, establece que los requisitos de *software* constituyen las necesidades de los clientes, las funcionalidades que deben cumplir el sistema *software* y las restricciones que debe poseer. Generalmente estos requisitos pueden clasificarse en funcionales y no funcionales.

Requisitos funcionales

Pressman establece que un requisito funcional no es más que una capacidad o condición que el sistema debe cumplir. A continuación, se detallan los requisitos funcionales de la investigación:

RF 1: Capturar petición del Subsistema Web.

Descripción: El sistema debe permitir capturar la petición enviada por el Subsistema *Web*. Los tipos de petición son: petición de codificación, de porcentaje de codificación, de porcentaje de uso de CPU (del inglés *Central Processing Unit*), de Pausar proceso de codificación, de Detener proceso de codificación, de Cancelar proceso de codificación, de lectura de metadatos y de cantidad de medias codificando.

RF2: Notificar información al Subsistema Web.

Descripción: El sistema debe permitir notificar al Subsistema *Web* la información procesada de la petición. Las informaciones que se pueden notificar son: fin de codificación de la media, error de codificación de la media, porcentaje de codificación de la media, porcentaje de uso de CPU, proceso pausado de codificación, proceso detenido de codificación, proceso cancelado de codificación, metadatos de la media y cantidad de medias codificando.

RF3: Iniciar proceso de codificación de media.

Descripción: El sistema debe permitir codificar una media cuando se envía una petición de codificación. Para ello se toma de dicha petición los parámetros: dirección origen de la media, dirección destino de la media, atributos de la media, dirección origen de subtítulo, atributos de subtítulo, se incrusta o no subtítulo,

dirección origen de audio, atributos de audio y se incrusta o no audio. Los atributos de subtítulo son tamaño, color, estilo y tipo de letra y los del audio son *bitrate*, volumen y *códec*. En caso de que no se establecen atributos de la media se tomarán los siguientes valores: ancho=720, alto=480, tiempo_inicio=100, tiempo_fin=200, *bit_rate*=1500. Cuando no se especifique origen o destino de la media no se codificará.

RF4: Pausar proceso de codificación de media.

Descripción: El sistema debe permitir pausar la media que se encuentra codificando cuando se envía una petición de pausar proceso de codificación.

RF5: Detener proceso de codificación de media.

Descripción: El sistema debe permitir detener la media que se encuentra codificando cuando se envía una petición de detener proceso de codificación.

RF6: Cancelar proceso de codificación de media.

Descripción: El sistema debe permitir cancelar la codificación de la media que se encuentra codificando cuando se envía una petición de cancelar proceso de codificación.

RF7: Determinar porcentaje de codificación de media.

Descripción: El sistema debe permitir determinar el porcentaje de codificación de la media cuando se envía una petición de porcentaje de codificación.

RF8: Determinar cantidad de medias codificando.

Descripción: El sistema debe permitir determinar la cantidad de medias procesando cuando se envía una petición de cantidad de medias procesando.

RF9: Determinar porcentaje de uso de CPU.

Descripción: El sistema debe permitir determinar el porcentaje de uso de CPU cuando se envía una petición de porcentaje de uso de CPU.

RF10: Extraer metadatos.

Descripción: El sistema debe permitir extraer los metadatos de una media cuando se envía una petición de lectura de metadatos. Los metadatos que se pueden extraer son nombre de la media, autor, duración, álbum y CD.

Requisitos no funcionales

Según (Jacobson, y otros, 2000) son requisitos no funcionales aquellos que imponen restricciones en el diseño, la implementación o los estándares de calidad, son propiedades o cualidades que el producto debe tener.

RNF1: Seguridad: La información que se gestiona en el componente estará protegida del acceso no autorizado.

RNF2: Restricciones de diseño: El diseño de la aplicación estará sujeto a las especificaciones de la dirección de producción de la Universidad de las Ciencias Informáticas, específicamente a la marca XILEMA.

RNF3: Software:

Sistema operativo:

GNU-Linux: Ubuntu 15.4

Windows 7 o superior.

RNF4: Hardware:

40 GB de capacidad en disco duro o más.

2 GB de RAM o más.

Procesador con tecnología Intel Dual Core P8700 a 2.53 GHz o más.

2.4. Modelo del sistema

“...los considerados padres de la Ingeniería de Software opinan que El Modelo de Casos de Uso del Sistema colabora con el cliente, los usuarios y los desarrolladores en establecer un lenguaje común para la futura implementación de un sistema software...” Cada tipo de usuario del sistema a desarrollar se representa mediante un actor. Todos los actores y casos de uso conforman el Modelo de Casos de Uso del Sistema (en lo adelante CUS) (Jacobson, y otros, 2000).

Definición de los Actores del sistema

Tabla 1. Descripción general del actor del sistema

Actor	Descripción
Subsistema Web	Se refiere a la aplicación web XILEMA AGORAV , encargada de solicitar todas las peticiones al componente en desarrollo.

Patrones de Casos de Uso del Sistema

Los patrones de CUS ayudan a describir lo que el sistema debe hacer, describen el uso del sistema y cómo interactúa con los usuarios. El uso de estos patrones constituye una buena práctica en el modelado de CUS (Larman, 1999). A continuación, se describen los patrones de casos de uso empleados en la investigación:

Extensión concreta: La extensión concreta consiste en una relación de extensión entre dos Casos de Uso como se evidencia en la Fig. 4, en el Caso de Uso Establecer comunicación con el subsistema *web* y el resto de los casos de uso del diagrama.

Diagrama de Casos de Uso del Sistema

CAPÍTULO II

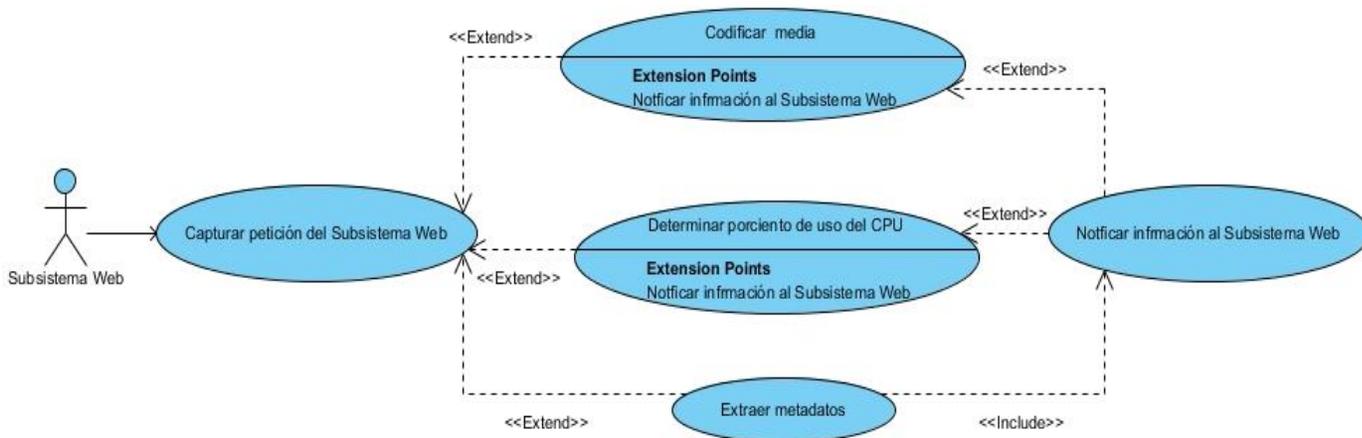


Fig. 3. Diagrama de Casos de Uso del sistema

Descripción textual de los Casos de Uso del Sistema

La Descripción textual de un Caso de Uso ayuda a identificar los objetos y operaciones en los sistemas. Cada Caso de Uso se describe utilizando una descripción en lenguaje natural. Según (Sommerville, 2005), esto ayuda a los diseñadores a identificar los objetos en el sistema y les permite comprender el comportamiento que tendrá el mismo.

A continuación, se muestra la descripción textual del Caso de Uso Codificar Media. El resto de las descripciones textuales de cada Caso de Uso se pueden consultar en el Anexo I.

Tabla 2. Descripción del Caso de Uso “Capturar petición del Subsistema Web”

Caso de Uso	Capturar petición del Subsistema <i>Web</i> .
Actores	Subsistema <i>Web</i>
Resumen	El caso de uso inicia cuando el Subsistema <i>Web</i> envía al sistema una petición.
Precondiciones	
Postcondiciones	Una petición fue atendida por el sistema.
Referencias	RF1.
Prioridad	Crítica.
Flujo Normal de Eventos	
Actor	Sistema
1. El Subsistema <i>Web</i> envía una petición al sistema.	2. El sistema captura la petición.
	3. Si es una petición de codificación, de porcentaje de codificación, de pausar proceso de codificación, de detener proceso de codificación, de cancelar proceso de codificación

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

CAPÍTULO II

		o de cantidad de medias codificando ir al Caso de Uso Codificar media . Si es una petición de porcentaje de uso de CPU, ir al Caso de Uso Determinar porcentaje del CPU . Si es una petición de lectura de metadatos ir al Caso de Uso Extraer metadatos .
		4. Termina el Caso de Uso.
Flujos alternos		
Actor		Sistema
Relaciones	CU Incluidos	
	CU Extendidos	Codificar media, Determinar porcentaje de uso del CPU y Extraer metadatos.

Tabla 3. Descripción del Caso de Uso “Codificar media”

Caso de Uso	Codificar media.
Actores	
Resumen	El caso de uso inicia cuando se recibe una petición de codificación, de porcentaje de codificación, de pausar proceso de codificación, de detener proceso de codificación, de cancelar proceso de codificación o de cantidad de medias codificando. La petición es atendida y finalmente el caso de uso termina cuando: una media es codificada, pausada, detenida o cancelada, es determinado el porcentaje de codificación o la cantidad de medias codificando.
Precondiciones	
Postcondiciones	Una de estas acciones es realizada: una media es codificada, pausada, detenida o cancelada, es determinado el porcentaje de codificación o la cantidad de medias codificando.
Referencias	RF3, RF4, RF5, RF6, RF7, RF8.
Prioridad	Crítica.
Flujo Normal de Eventos	
Actor	Sistema
	<ol style="list-style-type: none"> 1. Si es una petición de codificación ir a la Sección 1. 2. Si es una petición de pausar proceso de codificación ir a la Sección 2. 3. Si es una petición de detener proceso de codificación ir a la Sección 3. 4. Si es una petición de cancelar proceso de codificación ir a la Sección 4.

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

CAPÍTULO II

	<p>5. Si es una petición de porcentaje de codificación ir a la Sección 5.</p> <p>6. Si es una petición de cantidad de medias codificando ir a la Sección 6.</p>
Sección 1: “Codificar media ”	
Actor	Sistema
	1. El sistema especifica de la media dirección origen, dirección destino y atributos.
	2. El sistema especifica del subtítulo dirección origen, atributos y si se incrusta o no el subtítulo.
	3. El sistema especifica del audio dirección origen, atributos y si se incrusta o no el audio.
	4. El sistema codifica la media, ir al Caso de Uso Notificar información al Subsistema Web.
	5. Termina el Caso de Uso.
Flujos alternos	
Actor	Sistema
	1.1 Si no se especifican la dirección origen o la dirección destino el sistema no codifica la media. Termina el Caso de Uso.
	1.2 Si no se especifican los atributos de la media el sistema toma los siguientes valores por defecto para los atributos: ancho=720, alto=480, tiempo_ inicio=100, tiempo_ fin=200, bit_rate=1500. Termina el Caso de Uso.
Sección 2: “Pausar codificación”	
Actor	Sistema
	1. El sistema pausa la media que se encuentra codificando. Termina el Caso de Uso.
Sección 3: “Detener codificación”	
Actor	Sistema
	1. El sistema detiene la media que se encuentra codificando. Termina el Caso de Uso.
Sección 4: “Cancelar codificación”	
Actor	Sistema
	1. El sistema cancela la media que se encuentra codificando. Termina el Caso de Uso.
Sección 5: “Porcentaje de codificación”	

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

CAPÍTULO II

Actor		Sistema
		1. El sistema determina el porcentaje de codificación, ir al Caso de Uso Notificar información al Subsistema Web .
		2. Termina el Caso de Uso.
Sección 6: "Cantidad de medias codificando"		
Actor		Sistema
		3. El sistema determina la cantidad de medias que se encuentran codificando, ir al Caso de Uso Notificar información al Subsistema Web .
		4. Termina el Caso de Uso.
Relaciones	CU Incluidos	
	CU Extendidos	Notificar información al Subsistema Web.

2.5. Arquitectura de software

La arquitectura de un sistema consiste en la descripción de los subsistemas y componentes de un sistema informático y las relaciones que se establecen entre ellos (Pressman, 2005). Resulta necesaria la arquitectura pues permite en un sistema organizar su desarrollo, promover la reutilización y hacerlo evolucionar en la medida que se desarrollen nuevas versiones. Para definir la arquitectura es necesario definir la infraestructura (computadoras, dispositivos, sistemas operativos, servidores) e identificar las propiedades del sistema que se encuentra en construcción. Otro aspecto importante es seleccionar y combinar patrones arquitectónicos además de identificar subsistemas para integrarlos a la arquitectura.

Para la construcción del sistema se propone una arquitectura orientada a la publicación de servicios donde se emplean los principios de *Service Oriented Architecture* (SOA); de forma tal que brinde sus funcionalidades a través de estos, usando un protocolo de comunicación optimizado para el intercambio de información con servidores *web*. Estas funcionalidades esencialmente enfocadas al procesamiento de archivos multimedia se implementarán usando bibliotecas con alto nivel de aprovechamiento de recursos de *hardware*, que sean capaces de cubrir la amplia variedad de formatos y *codec* de videos que existen. Además de que, mediante actualizaciones, incorpore los nuevos que se creen. De esta manera se logra que el sistema procese ficheros audiovisuales nuevos sin la necesidad de modificar el código fuente de la aplicación, simplemente actualizando la biblioteca.

Estilo arquitectónico

“Un estilo arquitectónico es una transformación impuesta al diseño de todo un sistema. El objetivo es establecer una estructura para todos los componentes del sistema” (Pressman, 2005). Los estilos definen

los patrones que posiblemente se utilicen en los sistemas, evaluando así la arquitectura que más se ajuste a los requisitos que tiene definido el *software*. Los estilos pueden apreciarse como una forma de organizar los distintos modelos de sistemas, además sirven para sintetizar estructuras de soluciones.

Para la implementación de la propuesta de solución se utiliza el estilo arquitectónico **Llamada y retorno**. Este estilo tiene características favorables para el desarrollo de programas con estructuras relativamente fáciles de modificar y adaptar a nuevos entornos, lo cual contribuye a la flexibilidad del subsistema en cuestión.

Patrón arquitectónico

Un patrón arquitectónico al igual que un estilo impone una transformación en el diseño de una arquitectura. No obstante, el patrón no se concentra en toda la arquitectura sino en un aspecto; impone sobre la arquitectura una regla pues explica la forma en que el *software* manejará al nivel de infraestructura algún aspecto referente a su funcionalidad (Pressman, 2005). Los patrones abarcan del comportamiento dentro del contexto de la arquitectura aspectos específicos. Junto a un estilo arquitectónico los patrones son usados para determinar de un *software* la forma de su estructura general. Según (Pressman, 2005) un patrón arquitectónico en capas permite organizar el sistema en diferentes conjuntos de abstracción, lo cual garantiza entre varios aspectos proporcionar amplia reutilización y admitir refinamientos y optimizaciones. Entre las ventajas que presenta este patrón arquitectónico se encuentran (Pressman, 2005): “Permite aislar a la tecnología que implementa la base de datos, de forma que sea fácil cambiar esta tecnología” y se ajusta a las prácticas orientadas a objetos donde el procesamiento ocurre a través de mensajes.

Para la realización del Componente Transcoder se emplea el patrón **En capas**, específicamente en tres capas: *Communication*, *Logic* y *Data Access*. La capa *Communication* es la responsable del intercambio de información a través de la publicación de servicios con otras aplicaciones. La capa *Logic* se encarga de prácticamente todo el procesamiento lógico de información que tiene lugar en el subsistema y la *Data Access* garantiza la persistencia de datos y el acceso a ellos, dígase en bases de datos o en ficheros físicos. Se escoge el patrón debido a que distribuye por capas el trabajo donde cada una provee de información a las otras, además si cambiara la forma de trabajo de alguna de estas capas sus modificaciones no afectarían a las demás. Un ejemplo de esto sería si se decidiera implementar una nueva forma de comunicación solo habría que cambiar elementos en la capa *Communication*. La arquitectura en tres capas permite hacer más sencillo el desarrollo de las aplicaciones para luego brindarle un mejor mantenimiento.

Patrones de diseño

Los patrones de diseño elevan la calidad de los sistemas además de brindar una solución probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares. Existen clasificaciones para estos patrones entre los que se encuentran los Patrones Generales de *Software* para Asignar Responsabilidades y los Patrones del Grupo de los Cuatro. A continuación, se presentan los patrones de diseño que son utilizados para el desarrollo de la solución.

Patrones Generales de Asignación de Responsabilidades

Los patrones generales de *software* para asignar responsabilidades (por sus siglas en inglés GRASP¹) permiten asignar responsabilidades a objetos de forma tal que estos conozcan el rol que le corresponde. Son descritos los principios fundamentales expresados en forma de patrones de la asignación de responsabilidades (Larman, 1999). Los patrones que se utilizan son:

Experto: Asigna una responsabilidad a la clase que cuenta con la información necesaria para cumplirla. Constituye un principio básico que frecuentemente se utiliza en el diseño orientado a objetos. Las ventajas que presenta este patrón son: la conservación del encapsulamiento debido a que los objetos para hacer lo que se les pide se valen de su propia información, además de posibilitar que el sistema sea robusto y de fácil mantenimiento. En la solución el patrón es utilizado en las tres capas, ejemplo de ello es la clase *DataAccess* que es la única que posee la configuración de la base de datos y la forma de comunicarse con ella.

Creador: Asigna la responsabilidad a una clase de crear objetos, práctica muy frecuente en los sistemas orientados a objetos. Entre sus beneficios se encuentra el brindar soporte a un bajo acoplamiento lo que trae como consecuencia una menor dependencia respecto al mantenimiento y la reutilización en mayor medida. En la solución el patrón se refleja en las tres capas, ejemplo de ello es la clase *Media* que es la encargada de crear objetos de *Subtitle*.

Controlador: Asigna la responsabilidad a una clase de administrar un mensaje de los eventos del sistema. “*Un evento del sistema es un evento de alto nivel generado por un actor externo; es un evento de entrada externa. Se asocia a operaciones del sistema: las que emite en respuesta a los eventos del sistema*” (Larman, 1999). En la solución se refleja el patrón en la capa lógica de negocio, específicamente en la clase *Controller* la cual maneja gran parte de las acciones que se realizan en el sistema.

Alta cohesión: Asigna las responsabilidades en función de mantener la complejidad dentro de los límites posibles. La cohesión es una medida que refleja cuán enfocadas y relacionadas se encuentran las responsabilidades en una clase. Las clases que presentan una alta cohesión se caracterizan por poseer responsabilidades estrechamente relacionadas que no hagan un enorme trabajo; sin embargo, una clase

¹ GRASP: *General Responsibility Assignment Software Patterns*

con baja cohesión realiza muchas cosas no afines a un trabajo excesivo. Entre los beneficios que presenta el patrón está la mejora de la facilidad y claridad para comprender el diseño. En la solución el patrón se refleja en las tres capas, ejemplo de ello es la clase *Subtitle* debido a que las funcionalidades que se implementan en dicha clase tienen fines similares.

Bajo acoplamiento: Asigna la responsabilidad a las clases, lo cual garantiza que las mismas se encuentren lo más independiente posibles, lo que trae consigo una dependencia escasa y un aumento en la reutilización. En la solución el patrón se refleja en las tres capas, ejemplo de ello es la clase *Audio* que no tiene dependencias de clase alguna.

Patrones del Grupo de los Cuatro

Los patrones del Grupo de los Cuatro (por sus siglas en inglés GOF²) se dividen en tres categorías: patrones de creación, patrones estructurales y patrones de comportamiento (Larman, 1999).

A continuación, se describe el patrón que se evidencia en la solución:

Instancia única o *Singleton*: es un patrón de creación que garantiza que una clase posea una única instancia, a la vez que provee un punto de acceso global a ella. Permite por su diseño restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Este patrón se utiliza en la capa lógica en la clase *Controller* y en la capa de acceso a datos en la clase *Connection*, ambas clases tienen su constructor privado y un método que crea o brinda una instancia de dichas clases llamado *getInstance ()*, lo que garantiza que se tenga de ellas una única instancia.

2.6. Modelo de Diseño

Es la representación física de los Casos de Uso mediante un modelo de objetos; centra su atención en el impacto que tiene en el sistema, los requisitos tanto funcionales como no funcionales y algunas otras restricciones del entorno de implementación (Jacobson, y otros, 2000), sirve de abstracción para la implementación y como entrada fundamental de la misma.

Diagrama de Clases del Diseño

Según (Jacobson, y otros, 2000), un diagrama de clases corresponde a la realización de un Caso de Uso, donde se muestran las clases, subsistemas y las relaciones involucradas en el mismo, también tiene en cuenta los requisitos que influyen sobre determinada clase, objetos o subsistemas.

² GOF: *Gang Of Four*

El diagrama de clases del diseño del “Componente de codificación y decodificación para XILEMA AGORAV” se encuentra dividido en tres capas: *Communication*, *Logic* y *Data Access*.

La capa *Communication* está compuesta por las clases *Communication_Controller* (responsable de controlar que todas las peticiones que se realicen en la clase *Server_WS* sean atendidas por la clase *Controller* de la capa *Logic*), *Server_WS* (responsable de gestionar los clientes y las peticiones de los mismos) y *Client_WS* (almacena la información referente a un cliente). La capa *Logic* está compuesta por varias clases, las mismas son: *Controller* (encargada de realizar las peticiones que llegan a través de la clase *Server_WS*), *Media* (almacena la información referente a una media y gestiona instancias de la clase *Subtitle*), *ThreadCpuUsage* (clase hilo que se encarga del cálculo del uso de CPU), *CpuUsageCalculator* (responsable de instanciar la clase *ThreadCpuUsage* para aislar el cálculo del uso del CPU del hilo central de la aplicación), *Communs* (encargada de realizar funcionalidades sencillas, comunes en otras clases), *Subtitle* (almacena la información referente a un cliente) y *Audio* (almacena la información referente a un cliente). La capa *Data Access* tiene como objetivo principal el acceso y persistencia de los datos en la base de datos. En ella se encuentran las clases *Data_Access* (encargada de realizar las consultas a la Base de Datos a través de una instancia a la clase *Conexion*) y *Conexion* (brinda un objeto de comunicación con la Base de Datos).

A continuación, se muestra el diagrama de clases del diseño del “Componente de codificación y decodificación para XILEMA AGORAV”:

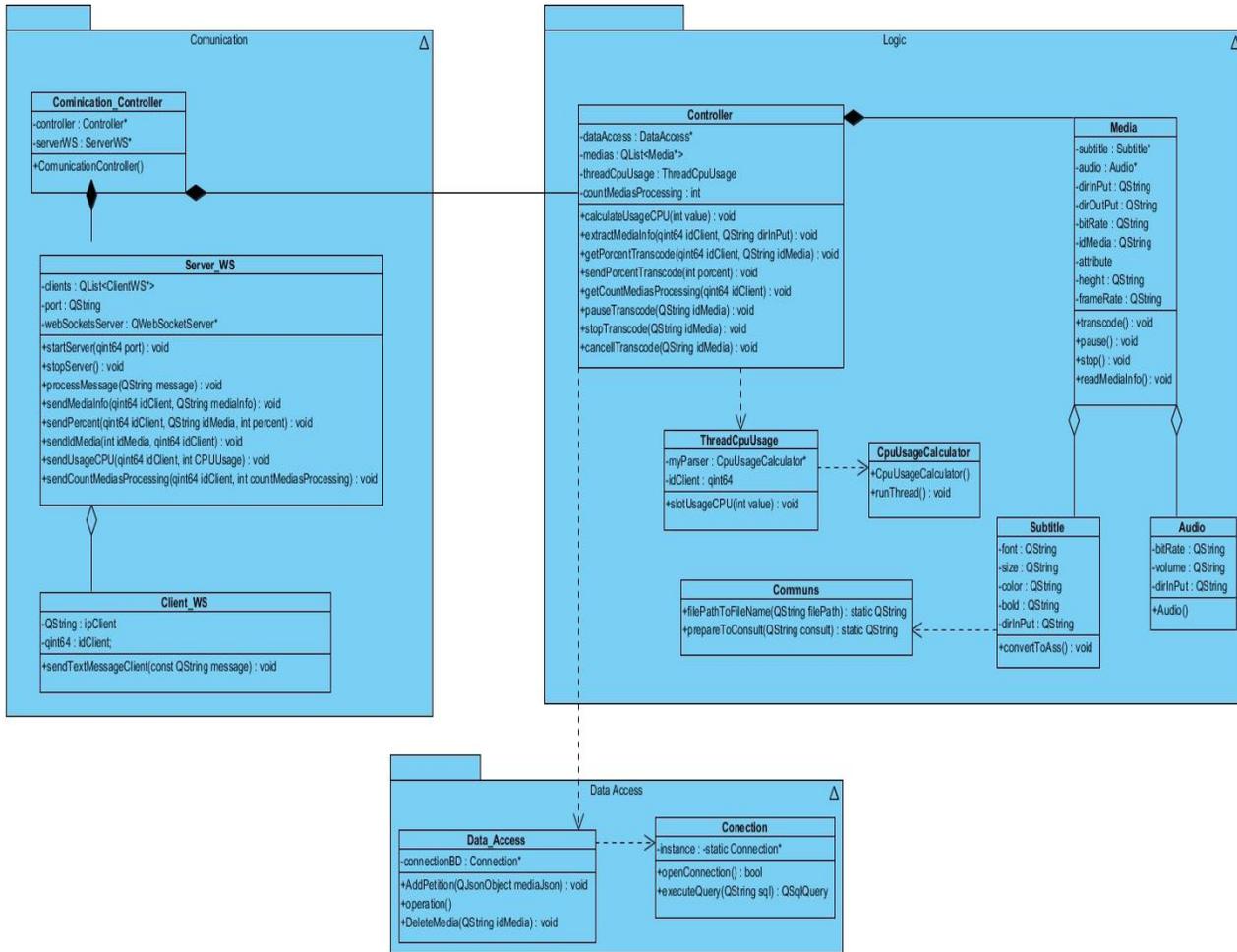


Fig. 4. Diagrama de clases del diseño.

2.7. Conclusiones parciales

La definición de los elementos asociados al dominio de la solución, la identificación y especificación de cada uno de los requisitos funcionales y no funcionales y la modelación del sistema, propician el entendimiento común y acertado entre el equipo de desarrollo y los clientes del componente a desarrollar. Lo anterior representa una ventaja para la correcta ejecución de las restantes etapas del desarrollo de la investigación.

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS

3.1. Introducción

En el presente capítulo se presentan los diagramas correspondientes a los modelos de diseño, implementación, componentes y despliegue. Finalmente, se presenta el proceso de pruebas a la solución propuesta.

3.2. Estándar de codificación

Los estándares de codificación son reglas y descripciones que son establecidas para facilitar la comprensión y el mantenimiento del código. El estándar de codificación debe seguirse desde el comienzo del software. El empleo de los estándares permite obtener un código legible, reduce la posibilidad de cometer errores, mejora la comunicación entre los programadores y brinda una guía bien documentada para el mantenimiento del *software*. En la presente investigación se propone el uso del siguiente estándar de codificación para cumplir con las políticas establecidas por el departamento Desarrollo de Componentes perteneciente al centro GEYSED para la programación en C++ con algunas personalizaciones.

Apariencia de atributos:

- Se escriben en minúscula y su nombre debe tener relación con el valor que almacenan.

Apariencia de funciones:

- Se escriben en mayúscula y su nombre debe tener relación con la operación que realizan.
- En caso de que el nombre sea compuesto se utiliza la notación *Camel Casing* (las palabras compuestas se escriben con mayúscula excepto la primera palabra).

Apariencia de clases:

- El nombre debe indicar con un sufijo la capa arquitectónica a la que pertenece y tener relación con el objetivo de la clase.

Separadores:

- Se utiliza el separador “_” para nombrar las clases.

Líneas y espacios en blanco:

- No contener más de una instrucción por línea.
- Dejar una línea en blanco entre las declaraciones de cada función.
- Las declaraciones de datos dentro de una función deben ir al inicio y separadas de las instrucciones ejecutables de la función a través de una línea en blanco.

Estilo *Allman*

Se trata de crear una nueva línea para las llaves, e indentar el código debajo de ellas. La llave de cierre tiene el mismo indentado que la de inicio.

```
ClientWS* ServerWS::clientById(qint64 id)
{
    | for (int i = 0; i < clients.length(); ++i)
        {
            | if (clients.at(i)->getIdClient()==id)
                {
                    | return clients.at(i);
                }
        }
    | return NULL;
}
```

Este estilo mantiene un código limpio y claro. Las llaves de inicio y fin coinciden en la misma columna, lo cual hace más fácil la identificación de cada bloque.

3.3. Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo del diseño, se implementan en términos de componentes (ficheros de código fuente, ejecutables, entre otros). El modelo de implementación también describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o los lenguajes de programación utilizados, y cómo dependen los componentes unos de otros (Jacobson, y otros, 2000).

Diagrama de Componentes

“Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño” (Jacobson, y otros, 2000). Entre los estereotipos estándar de componentes se encuentran: *executable* (programa que puede ser ejecutado en un nodo), *file* (fichero que contiene datos o código fuente), *library* (librería estática o dinámica) y *document* (documento). Los estereotipos pueden ser modificados durante la creación de componentes en un entorno de implementación particular. Un diagrama de componentes describe la estructura física del sistema y posibilita la representación de las dependencias de los componentes tanto en tiempo de ejecución como de compilación.

A continuación, se muestra el diagrama de componentes donde se aprecia la división del sistema en componentes y las dependencias que se generan entre los mismos. Primeramente, se presenta el *Transcoder* que consiste en un ejecutable y depende de los módulos *QtSql*, *QtXml*, *QtWebSockets* y *QtCore*, además de las bibliotecas *FFmpeg* y *MediaInfo*. El *QtSql* maneja el intercambio de información con la base de datos a través de las clases *QSql*, *QSqlQuery*, entre otras. *QtXml* permite el trabajo con los ficheros XML mediante instancias de la clase *QDomDocument*. En el caso de *QtWebSockets* se emplea para la

comunicación del componente con el subsistema *web* con la utilización de las clases *qwebsocketserver* y *qwebsocket*. *QtCore* permite el trabajo con objetos de tipo *QList*, *QString*, *QStringList*, por mencionar algunos. La biblioteca estática *FFmpeg* se emplea para todo el proceso de transcodificación de los materiales audiovisuales y en el caso de *MediaInfo* posibilita la obtención de los metadatos de dichos materiales.

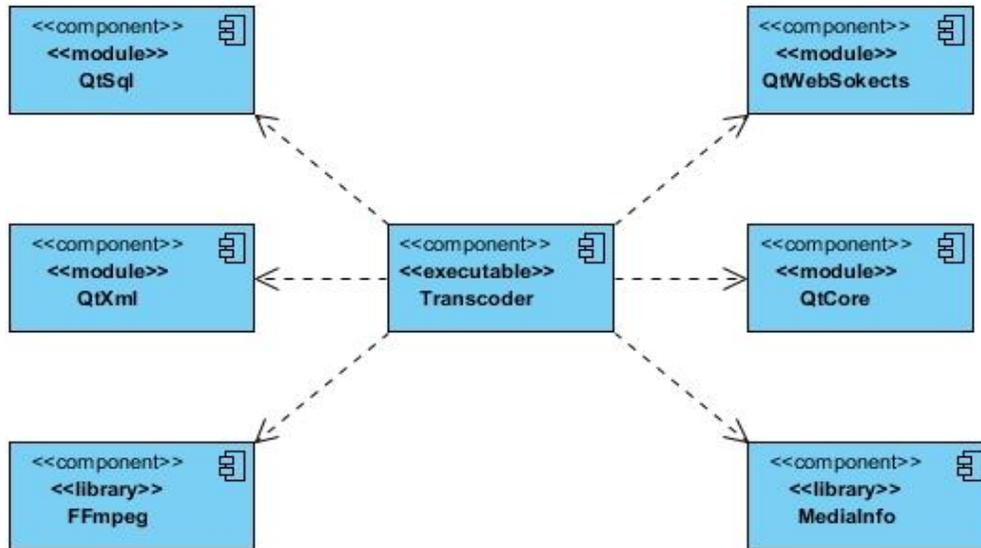


Fig. 5. Diagrama de Componentes

3.4. Modelo de Despliegue

“El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos del cómputo” (Jacobson, y otros, 2000).

Este modelo es primordial en las actividades de diseño e implementación pues la distribución que posee el sistema tiene mucha influencia en su diseño. El modelo de despliegue representa en sí mismo una correspondencia entre la arquitectura *software* y la arquitectura del sistema (*hardware*). Un nodo es un objeto físico que existe en tiempo de ejecución y que representa algún tipo de recurso computacional, en él se ejecutan los componentes y además posee relaciones que representan medios de comunicación entre ellos. Los componentes son los elementos que intervienen en la ejecución del sistema.

A continuación, se muestra el modelo de despliegue del *Transcoder*. En este modelo existe un nodo computadora servidor donde se encuentra la Plataforma XILEMA AGORAV con un módulo *WebSockets* para realizar las peticiones a la computadora que funciona como nodo de codificación el cual ejecuta el *Transcoder* que también tiene un módulo de comunicación (*QtWebSockets*). Se muestra además una computadora servidor que aloja el servidor de bases de datos, en el cual se guardan las direcciones físicas de las medias y la información de cada petición. También se puede apreciar una computadora que funciona

como servidor de almacenamiento de medias para facilitar el acceso y almacenaje de los materiales audiovisuales.

El flujo de información entre dichas computadoras comienza por el servidor *web* donde se realiza la petición de transcodificación de una media, que se guarda en el servidor de almacenamiento y la dirección en el servidor de bases de datos, luego realiza la petición al nodo de codificación y este extrae dicha media del servidor de almacenamiento y escribe en el de bases de datos los parámetros de la petición, finalmente la media transcodificada se almacena en el servidor de almacenamiento y se cambia el estado de la petición en la base de datos. Toda esta comunicación se realiza a través del protocolo TCP/IP.

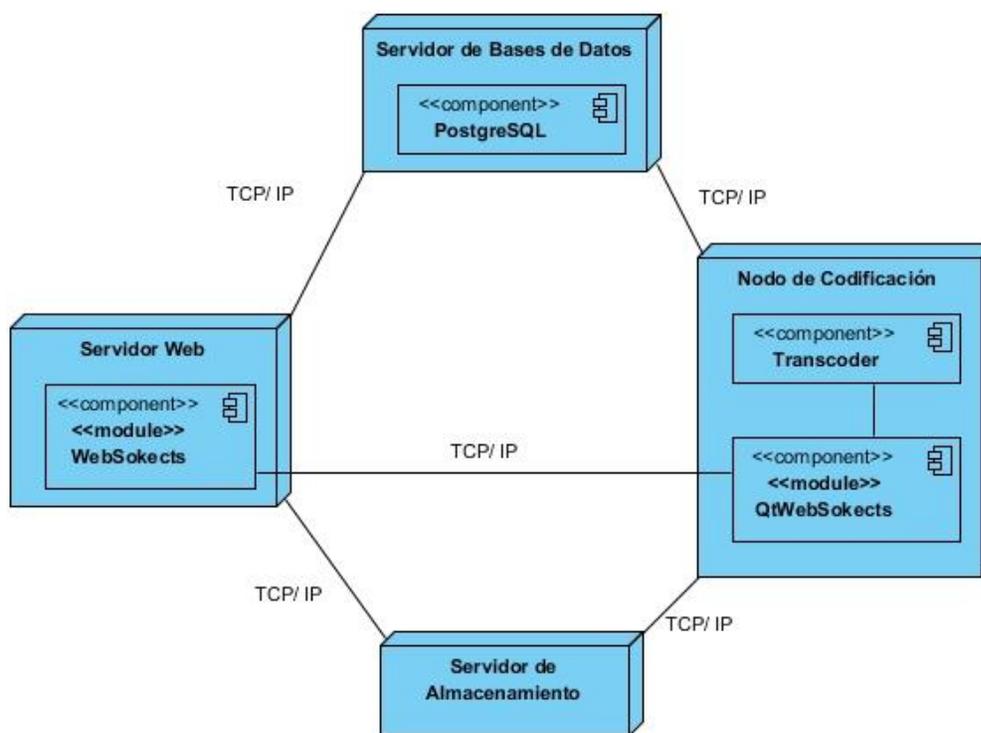


Fig. 6. Modelo de Despliegue

3.5. El Proceso de Pruebas

El proceso de pruebas no es más que el proceso de ejecución de un programa con la intención de descubrir un error, una prueba tiene éxito si descubre un error no detectado hasta entonces. Las pruebas demuestran hasta qué punto las funciones del *software* parecen funcionar de acuerdo con las especificaciones. En este proceso se ejecutan pruebas dirigidas a componentes del *software* o al sistema de *software* en su totalidad (Pressman, 2010).

La estrategia que se ha de seguir a la hora de evaluar dinámicamente un sistema *software* debe permitir comenzar por los componentes más simples y más pequeños y avanzar progresivamente hasta probar todo

el *software* en su conjunto. Las pruebas unitarias están dirigidas a probar cada componente individualmente para asegurar que funcione de manera apropiada como unidad (Pressman, 2010).

Pruebas de Integración

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras al mismo tiempo, se lleva a cabo pruebas para detectar errores asociados con la interacción. Se clasifican en integración incremental (ascendente o descendente) e integración no incremental. La primera se combina el módulo que se debe probar con el conjunto de módulos que ya han sido probados, mientras que la segunda consiste en probar cada módulo por separado y luego integrarlos todos a la vez para de esta manera probar el sistema completo (Oterino, 2014).

En la presente investigación es necesario realizar este tipo de pruebas debido a que la aplicación desarrollada no inicializa de forma independiente sus funcionalidades, por tal motivo se necesita de un sistema externo que realice las peticiones pertinentes. Además, para demostrar el cumplimiento del objetivo general, es necesario comprobar que el componente implementado erradica las limitaciones de codificación y decodificación identificadas en XILEMA AGORAV, planteadas en la descripción de la problemática. En la siguiente figura se muestra el flujo del proceso que se debe cumplir para lograr una exitosa integración entre la Plataforma Web de XILEMA AGORAV y el *Transcoder*.



Fig. 7. Integración del Transcoder con el Subsistema Web.

En la Fig. 8 se representa el esquema de integración del Transcoder con el Subsistema Web, cuando un cliente con permiso de publicación accede a la funcionalidad de publicar contenido y selecciona un video en formato (AVI, MPG, MOV, ASF, WMV, MKV) distinto a webm. El subsistema web muestra un mensaje informándole al usuario que el archivo se encuentra en proceso de codificación, lo que implica que el nodo de codificación procesa en ese momento la petición. El usuario con el objetivo de ver el estado de la conversión se dirige a la interfaz de Gestionar archivos multimedia en la pestaña de conversión y aprecia el porcentaje de conversión del video, lo que significa que el *Transcoder* envía en ese instante el porcentaje de conversión.

En la interfaz de Gestionar archivos multimedia que tiene las funcionalidades de pausar, detener, reanudar y cancelar; el usuario acciona en cada una de estas y el Transcoder realiza la acción correspondiente, lo cual se evidencia en el cambio de estado que refleja el Subsistema Web. Un ejemplo de lo descrito anteriormente es al usuario presionar la opción pausar, donde dicho botón cambia el nombre a reanudar y se muestra el proceso pausado, luego lo reanuda y se aprecia que la transcodificación continúa.

La plataforma cuenta con un módulo para la distribución automática de tareas de codificación, con el cual también se comprobó la integración. Este módulo brinda al usuario información de los nodos de codificación que obtiene a través de peticiones que le realiza a estos nodos. Entre las informaciones que se muestran se encuentran: cantidad de medias codificando, porcentaje de uso de CPU y porcentaje total de conversión; en una prueba donde se fueron aumentando el número de medias procesando paulatinamente, estos valores mostraban exactamente el estado del nodo en cuanto a ellos.

Una vez terminadas las pruebas de integración realizadas por cada uno de los requisitos funcionales identificados, se concluye que todas las funcionalidades implementadas en el Transcoder son accesibles desde el Subsistema Web; por tanto, queda demostrada la correcta integración entre el Transcoder y el Subsistema Web.

Análisis del rendimiento del componente por peticiones

Se realiza este tipo de prueba a la aplicación, cuyo objetivo es verificar y validar el desempeño de un elemento de un sistema bajo diferentes condiciones de carga, además estas pruebas son importantes cuando los sistemas deberán soportar un gran volumen de usuarios o transacciones concurrentes. En esta investigación las pruebas no están orientadas a la cantidad de usuarios sino a la cantidad de peticiones. Se debe tener en cuenta que estas pruebas deben ser realizadas bajo condiciones controladas para asegurar la precisión de las medidas tomadas. Por tales razones estas pruebas se realizaron en entornos diferentes y con cantidades de peticiones variables, los resultados de las mismas se muestran a continuación en la tabla 5:

Tabla 4: Pruebas de carga y estrés

Cantidad de peticiones	Propiedades de la PC					Tiempo de Respuesta (segundos)	Sistema Operativo
	Modelo CPU	Núcleos CPU	Hilos CPU	Frecuencia CPU	RAM		

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

CAPÍTULO III

10	Intel Quad Core N3530	4	4	2.16 GHz	4 Gb	1.08	Windows 8.1
20	Intel Quad Core N3530	4	4	2.16 GHz	4 Gb	1.32	Ubuntu 15.4
10	Intel Dual Core P8700	2	2	2.53 GHz	6Gb	1.98	Windows 10
20	Intel Dual Core P8700	2	2	2.53 GHz	6Gb	2.12	Windows 10
10	Intel Core i3 2120	4	4	3.3 GHz	4Gb	0.90	Windows 8.1
20	Intel Core i3 2120	4	4	3.3 GHz	4Gb	0.98	Windows 8.1
10	Intel Core i3 2120	4	4	3.3 GHz	2Gb	0.92	Ubuntu 15.4
20	Intel Core i3 2120	4	4	3.3 GHz	2Gb	0.94	Windows 10
10	Intel Core i7 4720HQ	4	8	3.6 HGz	8Gb	0.32	Windows 10
30	Intel Core i7 4720HQ	4	8	3.6 HGz	8Gb	0.41	Ubuntu 16.4

Significado de los elementos de la tabla 5:

1. **Cantidad de peticiones:** hace referencia a la cantidad de usuarios que se analizaron.
2. **Propiedades de la PC:** muestra las propiedades de las computadoras donde se ejecutó el *componente* desarrollado.
 - 2.1. **Modelo CPU:** muestra el modelo del CPU.

- 2.2. **Núcleos CPU:** muestra la cantidad de núcleos físicos del CPU.
- 2.3. **Hilos CPU:** muestra la cantidad de procesadores lógicos del CPU.
- 2.4. **Frecuencia CPU:** muestra la frecuencia máxima que puede alcanzar el CPU.
- 2.5. **RAM:** Muestra la cantidad de memoria RAM de la computadora.
- 3. **Tiempo de Respuesta (segundos):** tiempo que demora el componente en aceptar n peticiones.
- 4. **Sistema Operativo:** sistema operativo de la computadora donde se ejecutó el componente.

Después de realizadas las pruebas de carga y estrés, se arriba a la conclusión de que el componente desarrollado es capaz de aceptar un número de peticiones variables sin aumentar considerablemente el tiempo en hacerlo. Lo anterior se determina a pesar de haberse ejecutado el componente en computadoras con distintas propiedades, sistemas operativos y cantidad de peticiones. Como se puede observar en la Tabla 5, los tiempos de retorno no varían a gran escala cuando se analiza la misma cantidad de peticiones. Además, ninguna de las computadoras en que se probó, presentó inestabilidad en el sistema operativo lo cual pudiera ser provocado por el uso exhaustivo de recursos de *hardware*.

3.6. Conclusiones parciales

La realización del diagrama de componentes permite identificar las dependencias que posee la solución del sistema, así como librerías, los procesos y operaciones, donde a su vez muestra las interacciones y dependencias entre estos componentes. La confección del modelo de despliegue permite implementar el sistema en base a la distribución física del mismo. El estándar de codificación definido que sigue las políticas del proyecto XILEMA AGORAV sirve para un mejor entendimiento y mantenimiento de la aplicación. Con la realización de las pruebas de integración y estrés se identificaron no conformidades, erradicadas posteriormente, lo cual valida el correcto funcionamiento del *Transcoder*, además de su flexibilidad.

CONCLUSIONES

Una vez realizada la investigación se arribaron a las siguientes conclusiones:

- El estudio de las soluciones actuales relacionadas con la transcodificación permitió obtener un grupo de características a tener en cuenta para el desarrollo de la propuesta de solución, así como sus principales limitantes.
- La utilización de las bibliotecas *Ffmpeg* y *Mediainfo* facilitaron la implementación de un componente que permite la transcodificación desde y hacia disímiles formatos de archivos multimedia.
- El componente desarrollado permitió erradicar las limitaciones relacionadas con la codificación y decodificación de los materiales audiovisuales en el Sistema XILEMA AGORAV, además este puede ser integrado a otros sistemas de este tipo que empleen una arquitectura similar a la empleada en la propuesta.
- Las herramientas y tecnologías seleccionadas para el desarrollo del componente garantizan los principios de soberanía tecnológica que lleva a cabo el país.
- La realización de las pruebas de integración permitió comprobar la comunicación y el funcionamiento sistémico entre el componente propuesto y el subsistema web de XILEMA AGORAV.
- El análisis del rendimiento del componente demostró que el tiempo de respuesta para gestionar las tareas de transcodificación se mantiene estable a pesar de variar las características del *hardware* empleado.

RECOMENDACIONES

Luego de haber analizado los resultados del presente trabajo de diploma, resulta factible proponer las siguientes recomendaciones:

- Agregar procesamiento inteligente de video para que una petición del usuario pueda mostrar un resumen del video y detectar diferentes situaciones como movimiento, rostros, objetos, entre otros; con el objetivo de mejorar el proceso de catalogación de medias en la plataforma XILEMA AGORAV.
- Mejorar el algoritmo para determinar el porcentaje de uso del CPU.

BIBLIOGRAFÍA

1. *H.264 y MPEG-4*. Richardson, I. 2013. 2013.
2. **López, Jorge R, y otros. 2013.** *Limitaciones de los métodos objetivos de la medición de la calidad de video de acuerdo a la norma H.264*. 2013.
3. **Aguilar, A. Gómez. 2011.** Nuevos modelos de negocio: transformación en la cadena de valor en el audiovisual TIC. . 2011.
4. **Ariel Álvarez Monzoncillo, J.M. (2011). 2011.** *La televisión etiqueta. Nuevas audiencias, nuevos*. Barcelona : s.n., 2011.
5. **Berlanga, Rafael Llavori y Rodríguez, Julio. 2000.** *Introducción a la programación con Pascal*. s.l. : Universidad Jaume, 2000. 8480213051.
6. **Biurrún, Diego. 2009.** BLACKDUCK | Open Hub. *BLACKDUCK | Open Hub*. [En línea] 18 de 10 de 2009.
7. **Cerón , Miracle Mercadé. 2010.** *Estudi i avaluació de les funcionalitats de PostgreSQL*. 2010.
8. *Codificación de Contenidos Multimedia*. Dpto. de Ingeniería de Telecomunicación Área de Ingeniería Telemática. Universidad de Jaén E. P. S. de Linares.
9. **Company Overview of Popkin Software & Systems Inc. 2015.** Modelado de Sistemas com UML. 2015.
10. **Digia. 2013.** Qt digia. *Qt digia*. [En línea] 19 de 11 de 2013. <http://qt.digia.com/Product..>
11. **edición, Ian Sommerville. Novena edición. 9na. 11 may. 2015.** *Ingeniería del software*. 11 may. 2015.
12. **García, J. Mireles. 2010.** *Descripción del nuevo estándar de video H.264 y comparación de su eficiencia de codificación con otros estándares*. 2010.
13. **Guérin, Brice-Arnaud. 2015.** *Lenguaje C++*. 2015.
14. **Hernández, Enrique Orallo y Hernández, José Orallo. 2012.** *C++ Estándar*. 2012.
15. **Hernández, Raquel. 2013.** Digitalización, compresión y descompresión de videos para aplicación informática en entorno multimedia. 2013.
16. **Incera, Ramiro García y José.** *Transmisión eficiente de flujos de video ante la convergencia digital*. s.l. : Instituto Tecnológico Autónomo de México.
17. **informe), FERNANDO DE GARCILLÁN PRIETO(Gerente del Cluster ICT-Audiovisual de Madrid y editor del informe). y LUIS RODRÍGUEZ GARCÍA(Coordinador del. ESTADO DEL ARTE DE LAS TECNOLOGÍAS AUDIOVISUALES. s.l. : Un proyecto realizado por Xpertia Soluciones Integrales en colaboración con el Cluster ICT-Audiovisual de Madrid.**

18. *INTERNET Y LA SOCIEDAD RED*. **Castells., Manuel.** s.l. : Programa de Doctorado sobre la Sociedad de la Información y el Conocimiento. Universitat Oberta de Catalunya.
19. **Lafosse, Jerome. 2010.** El framework de desarrollo de aplicaciones Java EE. Barcelona : ENI, 2010. 2010.
20. **Larman, Craig. 1999.** *UML y PATRONES. Introducción al análisis y diseño orientado a objetos.* México : Prentice Hall, 1999. 536. 970-17-0261-1.
21. **Martí, J. y Adelantado, E. 2011.** Contenidos audiovisuales y televisivos para dispositivos móviles: Una aproximación al mercado español. 2011.
22. **MERCADÉ., MIRACLE CERÓN. 2010..** *Estudi i avaluació de les funcionalitats de PosgreSQL.* 2010.
23. **Mujal, Ramón María. 2014.** Las técnicas multimedia en las enseñanzas a distancia. 2014.
24. **Múnera, Natalia Macias y Ovidio, Jorge Arboleda. 2014.** Software para la conversión de formatos. 2014.
25. **Oterino, Ana M. del Carmen García. 2014.** Tipos de pruebas. 2014.
26. *Paralelización del codificador H.264 con estimación de movimiento adaptativa en clusters de PCs1.* **A. Rodríguez, M. Pérez, A. González, J. Peinado, J.C. Fernández.** Universidad Politecnica de Valencia, Dept. de Informática de Sistemas y Computadores ;Universidad Jaume I, Dpt. d'Enginyeria i Ciència dels Computadors. : s.n.
27. **Pérez, Gedisa de Silva. 2000.** *La televisión ha muerto. La nueva producción audiovisual en la era de Internet: la tercera revolución industrial.* Barcelona : s.n., 2000.
28. **Pressman, R. S. 2008.** *Ingeniería de Software: Un enfoque práctico. 5ta Edición.* Madrid : McGraw Hill, 2008. 765 p.
29. **Pressman, Roger S. 2010.** *Software engineering: a practitioner's approach.* . 2010.
30. **Proyecto XILEMA AGORAV.Centro de GEYSED. FUNCIONALIDADES FUNDAMENTALES DE LA SOLUCIÓN INFORMÁTICA PROPUESTA.**
31. **Richardson, I. 2012.** *Video Codec Design.* 2012.
32. **Sánchez, Tamara Rodríguez. 2015.** *Metodología de desarrollo para la Actividad productiva de la UCI.* 2015.
33. —. **2015.** *Metodología de desarrollo para la Actividad productiva de la UCI. Versión: 1.2.* 2015.
34. **Schuster, Guido M. y Katsaggelos, Aggelos. 2013.** *Rate-Distortion based video compression: optimal video frame compression and object boundary encoding.* 2013.
35. **Sierra, María. 2006.** *Trabajando con Visual Paradigm for UML.* Cantabria : s.n. : s.n., 2006.
36. **Sommerville, Ian. 2010.** *Ingeniería del Software. 7ma edición.* 2010.
37. **Team, MediaInfo. 2016.** FileHorse. *FileHorse.* [En línea] 17 de 2 de 2016.

38. **Toffler, A. y Toffler, H. 2013.** La revolución de la riqueza. 2013.
39. **TVAD, Grupo técnico del foro de la televisión de alta definición en España. 2011.** *Calidad de video en alta definición.* 2011.
40. **Vladimir Martell, Miguel Morciego. 2015.** *Libro blanco de XILEMA AGORAV.* 2015.
41. **Zugasti, Andrés Piccininno. 2014.** *Qt Mobile: compilación cruzada para distintos sistemas operativos móviles - PICCININNO - QT Mobile: compilación cruzada para distintos sistemas operativos móviles.* 2014.
42. **ALAN LE BIHAN, 2016.** Format Factory. *Softonic* [en línea]. [Consulta: 11 mayo 2016]. Disponible en: <http://format-factory.softonic.com/>.
43. **CABRERA MARTÍNEZ-PINILLOS, B. y SUÁREZ PÉREZ, J.M., 2011.** Plataforma de codificación e indexación de video para el Sistema de Captura y Catalogación de Medias. [en línea], [Consulta: 29 noviembre 2015]. Disponible en: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_04371_11.
44. **CRIOLLO, A., PÉREZ, T., WILSON, R. y EDGAR, A., 2010.** Estudio de la aplicabilidad de un sistema de comprensión MPEG-4 para la optimización del ancho de banda para la distribución de CATV en la ciudad de Cuenca. [en línea], [Consulta: 3 mayo 2016]. Disponible en: <http://localhost:8080/xmlui/handle/123456789/1063>.
45. **DPTO. DE INGENIERÍA DE TELECOMUNICACIÓN ÁREA DE INGENIERÍA TELEMÁTICA, 2015.** Codificación de Contenidos Multimedia. [en línea]. [Consulta: 29 noviembre 2015]. Disponible en: <http://www4.ujaen.es/~jccuevas/data/AATTAA2/Transparencias/Tema%20%20Codificacion%20de%20Contenidos%20Multimedia%20.pdf>.
46. **FABRIZIO FERRI-BENEDETTI, 2015.** Freemake Video Converter. *Softonic* [en línea]. [Consulta: 11 mayo 2016]. Disponible en: <http://freemake-video-converter.softonic.com/>.
47. **FERNÁNDEZ, J., RODRÍGUEZ, A., PÉREZ, M., GONZÁLEZ, A. y PEINADO, J., 2005.** Paralelización del codificador H.264 con estimación de movimiento adaptativa en clusters de PCs - jp2k5.pdf. [en línea]. [Consulta: 29 noviembre 2015]. Disponible en: <http://atc.umh.es/gatcom/Ficheros/Articulos/jp2k5.pdf>.
48. **GARCÍA, R. y INCERA, J., 2006.** Transmisión eficiente de flujos de video ante la convergencia digital. *AMCIS 2006 Proceedings* [en línea], Disponible en: <http://aisel.aisnet.org/amcis2006/505>.
49. **IVÁN RAMÍREZ, 2015.** Video to Video Converter. *Softonic* [en línea]. [Consulta: 11 mayo 2016]. Disponible en: <http://video-to-video-converter-1.softonic.com/>.
50. **JIMÉNEZ ÁLVAREZ, P. y YÁÑEZ ANLLO, M., 2015.** Guía Didáctica. Patrimonio audiovisual y remezcla en vivo: cuidados y usos creativos del archivo digital. [en línea], [Consulta: 3 mayo 2016]. Disponible en: <http://dspace.unia.es/handle/10334/3588>.
51. **JOSÉ MARÍA LÓPEZ, 2016.** Total Video Converter. *Softonic* [en línea]. [Consulta: 11 mayo 2016]. Disponible en: <http://total-video-converter.softonic.com/>.

52. **KAEL, J., 2014.** ProMedia Carbon. *Harmonic Inc* [en línea]. [Consulta: 29 noviembre 2015]. Disponible en: <http://www.harmonicinc.com/es/product/encoding-transcoding-distribution/promedia-carbon>.
53. **KAEL, J., 2014.** ProMedia Carbon. *Harmonic Inc* [en línea]. [Consulta: 29 noviembre 2015]. Disponible en: <http://www.harmonicinc.com/es/product/encoding-transcoding-distribution/promedia-carbon>.
54. **KOCHER, M., 2013.** History & Evolution; audio visual communication. [en línea]. [Consulta: 27 abril 2016]. Disponible en: <https://prezi.com/vkpv8w1xcosh/history-evolution-audio-visual-communication/>.
55. **LAGE, F.J., CATALDI, Z., LOPEZ, J.R., REBASA, M.P. y GARRO, D.M., 2013.** Limitaciones de los métodos objetivos de la medición de la calidad de video de acuerdo a la norma H.264. *Proceedings of International Conference on Engineering and Computer Education*, vol. 8, no. 0, pp. 465-468. ISSN 2317-4145. DOI 10.14684/icece.v8.465-468.
56. **LÓPEZ, S.R., 2013.** Los contenidos audiovisuales en internet y su impacto en la televisión. *Razón y palabra*, no. 83, pp. 682-692.
57. **LÓPEZ RABADÁN, P. y VICENTE MARIÑO, M., 2011.** MÉTODOS Y TÉCNICAS DE INVESTIGACIÓN DOMINANTES EN LAS REVISTAS CIENTÍFICAS ESPAÑOLAS SOBRE COMUNICACIÓN (2000-2009). [en línea]. [Consulta: 3 mayo 2016]. Disponible en: <http://www.revistacomunicar.com/pdf/2011-04-Lopez-Vicente.pdf>.
58. **MAZAEDA ECHEVARRÍA, R., 2014.** Programación concurrente en C++ v11. [en línea], [Consulta: 3 mayo 2016]. Disponible en: <http://uvadoc.uva.es:80/handle/10324/12044>.
59. **MORALES MORANTE, F., 2014.** *Montaje audiovisual: Teoría, técnica y métodos de control*. S.I.: Editorial UOC. ISBN 978-84-9029-791-9.
60. **PÓVEDA-LÓPEZ, I.-C., CALDERA-SERRANO, J. y POLO-CARRIÓN, J.-A., 2010.** Definición del objeto de trabajo y conceptualización de los Sistemas de Información Audiovisual de la Televisión. *Investigación bibliotecológica*, vol. 24, no. 50, pp. 15-34. ISSN 0187-358X.
61. **RUSSELL, 2014.** MPM Media IT Platform. *Tedial* [en línea]. [Consulta: 29 noviembre 2015]. Disponible en: <http://www.tedial.com/modules-and-architecture/mpm-media-it-platform>.
62. **SAETTLER, P., 2015.** Historical overview of audio-visual communication. *Audiovisual communication review*, vol. 2, no. 2, pp. 109-117. ISSN 0001-2890, 1556-6501. DOI 10.1007/BF02713269.
63. **SÁNCHEZ GALIANO, G., 2015.** *ESTUDIO COMPARATIVO DE SERVIDORES MULTIMEDIA*. S.I.: 3Ciencias. ISBN 978-84-943486-5-5.
64. **SANDOVAL RUIZ, C., 2012.** Codificador RS(n,k) basado en LFCS: caso de estudio RS(7,3). *Revista Facultad de Ingeniería Universidad de Antioquia*, no. 64, pp. 68-78. ISSN 0120-6230.
65. **SUÁREZ, Ì. y RAÚL EDUARDO, 2013.** Procedimiento y realización de un audiovisual, basado en emulación de arquitecturas sonoras y de imagen. *Biblioteca USB Bogotá, Colección Tesis CD T.IS*

BIBLIOGRAFÍA

760009-06 / S939p CD-ROM [en línea], [Consulta: 3 mayo 2016]. Disponible en: <http://bibliotecadigital.usb.edu.co/jspui/handle/10819/1510>.

66. **VEGA, E., 2012.** Audiovisuales - 01.pdf. [en línea]. [Consulta: 27 abril 2016]. Disponible en: <http://www.eugeniovega.es/assignaturas/audio/01.pdf>.

GLOSARIO DE TÉRMINOS

- **Transcodificación:**

Según Manovich, el constituyente más relevante del proceso de informatización de los nuevos medios de comunicación es la TRANSCODIFICACION. A lo que hoy en día llamamos nuevos medios, no son nada más ni nada menos que los viejos medios que han sido digitalizados. En simples palabras “transcodificar” significa traducir algo a un formato diferente, en este caso hablamos de traspasarla a una pc. Esta transformación solo es ejecutable gracias a la codificación numérica y la organización modular. Estos medios, son traducidos a un formato descifrable por el ordenador, pudiendo así ser almacenados, y otro de sus puntos a tener en cuenta es que pueden ser transmitidos a cualquier parte del planeta en tan solo segundos. Dentro de los nuevos medios de comunicación diferenciamos la capa cultural y la informática. Generalmente la capa informática actúa modificando la capa cultural de manera que nos vemos sumergidos en un mundo digital que avanza y no tiene límites.

- **Online:**

1. En telecomunicaciones, on-line (en línea) se utiliza para designar a una computadora que está contactada al sistema, está operativa, está encendida o accede a internet.
2. Estado de un usuario cuando está conectado a internet.

- **Codecs**

La palabra *codec* se traduce de las palabras codificador y decodificador. Un *codec* no es ni más ni menos que una serie de funciones algorítmicas necesarias para comprimir un archivo, a este proceso de compresión se le denomina "codificación" y descomprimir o decodificar los datos de audio y vídeo. Esto quiere decir que si queremos reproducir un vídeo digital y no tenemos instalado el códec con el que se han codificado los datos no podremos visualizarlo correctamente en nuestro PC (Sigla de *Personal Computer*," Computadora Personal"), deberemos buscar los codecs y tenerlos instalados.

- **Formatos de video:**

Los videos digitales se pueden guardar en archivos de distintos formatos. Cada uno se corresponde con una extensión específica del archivo que lo contiene. Existen muchos tipos de formatos de video. Aquí se citan algunos de los más utilizados. Asimismo cada tipo de archivo admite en cada momento un códec de compresión distinto.

- **Webm:**

Formato multimedia abierto y libre desarrollado por Google y orientado a usarse con HTML5. Es un proyecto de software libre, bajo una licencia permisiva similar a la licencia *BSD (Berkeley Software Distribution)*.

- **CPU**(*Central Processing Unit*, “Unidad Central de Proceso”):

Es un componente básico de la computadora personal u ordenador que procesa datos y realiza cálculos matemáticos-informáticos. Proporciona la capacidad de programación, y junto con la memoria y los dispositivos de entrada/salida es de los componentes computacionales que encontramos presente en toda la historia de las computadoras. Los microprocesadores de un chip han ido reemplazando a los CPU hasta llegar a los días actuales en que usualmente cuando se hace referencia a este término se habla de los microprocesadores.

ANEXOS

ANEXO I: Descripción textual de los Casos de Uso del sistema

Tabla 5.Descripción del Caso de Uso “Determinar porcentaje de uso del CPU”

Caso de Uso	Determinar porcentaje de uso del CPU.	
Actores		
Resumen	El caso de uso inicia cuando se recibe una petición de porcentaje de uso del CPU. El sistema determina el porcentaje de uso del CPU y así termina el caso de uso.	
Precondiciones		
Postcondiciones	El porcentaje de uso del CPU es determinado.	
Referencias	RF9	
Prioridad	Secundario.	
Flujo Normal de Eventos		
Actor		Sistema
		1. El sistema determina el porcentaje de uso del CPU, ir al Caso de Uso Notificar información al Subsistema Web.
		2. Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	Notificar información al Subsistema Web.

Tabla 6.Descripción del Caso de Uso “Extraer metadatos”

Caso de Uso	Extraer metadatos.	
Actores		
Resumen	El caso de uso inicia cuando se recibe una petición de lectura de metadatos de una media. El sistema extrae los metadatos y así termina el caso de uso.	
Precondiciones		
Postcondiciones	Los metadatos son extraídos de la media.	
Referencias	RF10	
Prioridad	Secundario.	
Flujo Normal de Eventos		
Actor		Sistema
		1. El sistema extrae los metadatos de la media. Los metadatos a extraer son: nombre de la media, autor,

COMPONENTE DE CODIFICACIÓN Y DECODIFICACIÓN PARA XILEMA AGORAV

ANEXOS

		duración, álbum y CD, ir al Caso de Uso Notificar información al Subsistema Web.
		2. Termina el caso de uso.
Relaciones	CU Incluidos	Notificar información al Subsistema Web
	CU Extendidos	

Tabla 7. Descripción del Caso de Uso “Notificar información al Subsistema Web”

Caso de Uso	Notificar información al Subsistema Web.	
Actores		
Resumen	El caso de uso inicia cuando se notifica al Subsistema Web recibe una petición de lectura de metadatos de una media. El sistema extrae los metadatos y así termina el caso de uso.	
Precondiciones		
Postcondiciones	Se notifica al Subsistema Web.	
Referencias	RF2.	
Prioridad	Crítica.	
Flujo Normal de Eventos		
	Actor	Sistema
		<p>1. El sistema notifica la información al Subsistema Web de acuerdo al tipo de petición:</p> <p>Si es una petición de codificación se notifica fin de codificación de la media.</p> <p>Si es una petición de porcentaje de codificación se notifica el porcentaje de codificación de la media.</p> <p>Si es una petición de cantidad de medias codificando se notifica la cantidad de medias codificando.</p> <p>Si es una petición de porcentaje de uso del CPU se notifica el porcentaje de uso del CPU.</p> <p>Si es una petición de lectura de metadatos se notifica los metadatos de la media.</p>
		2. Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	