



**Universidad de las Ciencias Informáticas
Facultad 6**

Módulo de captura y catalogación de materiales audiovisuales para el
Sistema de Trasmisión de Canales Virtuales

**Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas**

Autor(es): Gretel Benitez Fajardo.
Rafael Cruz Enjamio.

Tutor(es): Ing. Raúl Cabrera Miranda.
Ing. Merlín Milián Díaz.

La Habana, junio 2015
“Año 57 de la Revolución”



DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Firma Autor
Rafael Cruz Enjamio

Firma Autor
Gretel Benitez Fajardo

Firma Tutor
Ing. Merlín Milián Díaz

Firma Tutor
Ing. Raúl Cabrera Miranda



DATOS DE CONTACTO

Ing. Merlín Milián Díaz.

Graduada de la Universidad de las Ciencias Informáticas (UCI) en el año 2011. Pertenece al centro de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6. Labora como Asesora de Planificación y Control de dicho centro.

Correo electrónico: mmilian@correo.uci.cu

Ing. Raúl Cabrera Miranda.

Graduado en la Universidad de las Ciencias Informáticas (UCI) en el año 2013. Pertenece al departamento Integración y Soluciones Informáticas del Centro de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6. Labora como Recién Graduado en Adiestramiento.

Correo electrónico rcmiranda@uci.cu



“El éxito no es ni mágico ni misterioso. El éxito es la consecuencia natural de aplicar con firmexa los principios básicos de la superación personal.”

Jim Rohn.





Dedicatoria

Gretel

A dios, por darme la oportunidad de poder llegar a este momento.

*A mis padres por ser los principales responsables de lo que he logrado en la vida,
por ser mis guías.*

A mi hermana, mis tías, mis abuelas, mis primas y mis tíos.



Dedicatoria

Rafael

Dedico este trabajo:

A mi mamá que ha luchado mucho por verme triunfar en la vida.

A mi papá que siempre me ha hecho sentirme orgulloso de él.

A mis abuelos que siempre que siempre me han dado todo su amor.

A mi hermano que siempre está cuando lo necesito.

A toda mi familia y a aquellos amigos que ya son parte de ella.

A todos ellos; dedico este sueño hecho realidad.



Agradecimientos

Gretel

Los autores de este trabajo de diploma desean expresar su agradecimiento a las siguientes personas que colaboraron con todo el proceso de elaboración, revisión y culminación de este trabajo:

A mis padres (María Rosa y Enrique) por el amor, el esmero y la educación, este es su regalo. Por enseñarme el camino a seguir y dejarme recorrerlo sola.

A mi hermana Glenda.

A mis abuelas Iselda y Belquis, gracias por estar presente y por brindarme su apoyo y amor incondicional en estos años de mi vida.

A mi tía Kenia, por tu dedicación y sacrificio en estos años de mi vida.

A mis tías Margarita, Mirta y Cenia y a mis tíos, por su amor hacia mí.

A mi prima Yanyi, gracias por permitirme tenerte como una hermana.

A Leo, gracias por tu ayuda.

A mis primas y primos en general, los quiero.

A mis tutores (Merlin y Raúl), por la guía y su apoyo ilimitado.

A Tito por estar presente en todas las etapas de mi carrera.

A mis peleonas y queridas amigas Yeni, Nuri y Yaritza, gracias por sus regaños y cariño.



Agradecimientos

Módulo de captura y catalogación de materiales audiovisuales para el Sistema de Trasmisión de Canales Virtuales

A mi familia universitaria: Ale, Carlitos, Angel, Armando, Víctor, Lachi, Yanet, Vladimir, Lixandra, Haidys, Karo, chupi, Luis, White, Yuliet y a los que se quedan, ellos saben quiénes son, gracias por estar siempre ahí.

A mis compañeros de aula.

Al Tribunal y oponente por su valioso aporte con el cual contribuyen a mejorar este trabajo.

A mi compañero de tesis Rafa, por haberme acompañado en este esperado momento de mi vida, muchas gracias.



Agradecimientos

Rafael

Primeramente, agradezco a mis padres por confiar en mí y guiarme por el camino correcto, por enseñarme cuanto hay que esforzarse para alcanzar un sueño.

A mi hermano Raybel, por estar siempre disponibles en mi vida.

A mi esposa Merlin por darme su amor, estar pendiente de mí y ayudarme en todo momento.

A toda la familia: mis abuelos, mis tíos y primos, a Frank y Oliday, a todos gracias por estar siempre a mi lado y que hoy puedan compartir conmigo este importante momento de mi vida.

A Gregorio y Liuba por ser mis amigos y estar siempre pendientes durante mi recorrido en la universidad.

A mi compañera de tesis Gretel por estar disponible siempre que lo necesité y por su ayuda incondicional en todo este tiempo.

A mi tutor Raúl por estar a mi lado incondicionalmente en todo momento que necesite de su ayuda.

A mis amigos Jorge Luis y Deivrys por demostrarme que los amigos se llevan en el corazón.

A mis compañeros del equipo de fútbol por los momentos de alegría que compartimos.



Agradecimientos

Módulo de captura y catalogación de materiales audiovisuales para el Sistema de Trasmisión de Canales Virtuales

A todos mis compañeros de la UCI, de mi municipio, a mis compañeros de clases, a los profesores y a todos aquellos que han estado incondicionalmente siempre que los necesité.

A los miembros del tribunal y oponente por el sacrificio y la abnegación para que cada día fuéramos mejores.

A todos ustedes... Muchas Gracias.



RESUMEN

Actualmente existen empresas emergentes en el contexto audiovisual que se dedican a la transmisión y recepción de la televisión, en las que se ha percibido un aumento en la cantidad de los materiales audiovisuales, originando la necesidad de almacenarlos para su posterior uso. Cuando la cantidad de materiales que se almacenan crece surge la problemática de capturar y catalogar para acceder con mayor facilidad a estos. En el centro GEYSED existe un Sistema de Transmisión de Canales Virtuales que tiene como objetivo automatizar la transmisión, administración y gestión de los canales de radio y televisión existentes en una entidad; este cuenta con un módulo que permite la catalogación y el almacenamiento audiovisual, pero no realiza la gestión de parámetros de catalogación ni la captura de flujos de videos, que por la utilidad que supone tiene gran importancia para las operaciones de gestión del sistema.

El presente trabajo constituye la realización de un módulo que permita la automatización del proceso de captura desde entidades externas al sistema y gestionar los parámetros de catalogación asociados a los materiales audiovisuales, garantizando una amplia descripción de los materiales audiovisuales a utilizar en el sistema.

Palabras clave: captura, catalogación, material audiovisual.



ABSTRACT

There are now emerging companies in the audiovisual context engaged in the transmission and reception of television, which has seen an increase in the amount of audiovisual materials, causing the need to store them for later use. When the amount of materials that are stored growing problem of capture and catalog for easier access to these arises. In the center there is a GEYSED *Sistema de Transmisión de Canales Virtuales* that aims to automate the transmission, administration and management of the existing radio and television entity; this has a module that allows audiovisual cataloging and storage, but not perform management parameters cataloging or capture video streams, which the utility has assumed great importance for operations management system.

This work is the realization of a module that allows the automation of capture from external entities and manage system parameters associated cataloging audiovisual materials, ensuring a comprehensive description of audiovisual materials to be used in the system.

Palabras clave: capture, cataloging, audiovisual material.



Índice de imágenes

Imagen 1: Modelo de dominio.	25
Imagen 2: Diagrama de casos de uso del sistema de catalogación.	33
Imagen 3: Diagrama de casos de uso del sistema de captura.	33
Imagen 4: Diagrama de Clases del Diseño del Caso de uso Gestionar autor de material audiovisual del sistema de catalogación.....	46
Imagen 5: Diagrama de Clase del Diseño del sistema de captura.	47
Imagen 6: Diagrama de componentes catalogación.....	49
Imagen 7: Diagrama de componentes del catalogacionBundle.	50
Imagen 8: Diagrama de componentes captura.....	51
Imagen 9: Diagrama de despliegue.....	53



Índice de tabla

Tabla 1. Descripción del actor del sistema	32
Tabla 2. Descripción del caso de uso Gestionar tipología de material audiovisual.	34
Tabla 3. Resultados de las pruebas de rendimiento del sistema de catalogación.	55
Tabla 4. Caso de prueba para el Caso de Uso Gestionar género: Sección “Adicionar género”.....	57
Tabla 5. Caso de prueba para el Caso de Uso Gestionar género: Sección “Editar género”.	57
Tabla 6. Caso de prueba para el Caso de Uso Gestionar género: Sección “Eliminar género”.....	58
Tabla 7. Caso de prueba para el Caso de Uso Gestionar género: Sección “Mostrar géneros”.....	59



Tabla de contenido

INTRODUCCIÓN.....	1
FUNDAMENTACIÓN TEÓRICA	6
Introducción.....	6
1.1 Objeto de estudio	6
1.1.1 Descripción general del objeto de estudio.....	6
1.1.2 Descripción actual del dominio del problema	9
1.2 Análisis de otras soluciones existentes.....	10
1.2.1 Xstream	10
1.2.2 Catalis.....	11
1.2.3 Tarsys.....	11
1.2.4 Sistema de captura y catalogación de medias	12
1.3 Herramientas y tecnologías para el desarrollo del software.....	13
1.3.1 Metodologías de desarrollo.....	13
1.3.2 Lenguaje de Modelado Unificado (UML)	14
1.3.3 Herramienta CASE.....	15
1.3.4 Lenguaje de Programación	15
1.3.5 Entorno Integrado de Desarrollo	20
1.3.6 Bibliotecas de desarrollo	21
1.3.7 Servidor web.....	22
1.3.8 Sistema Gestor de Base de Datos	22
Conclusiones parciales	23
CARACTERÍSTICAS DEL MÓDULO DE CAPTURA Y CATALOGACIÓN.....	24
Introducción al capítulo.....	24
2.1 Modelo de dominio.....	24
2.1.1 Descripción del flujo del diagrama del modelo de dominio.....	24
2.1.3 Descripción de conceptos del dominio del problema	25
2.2 Especificación de los requisitos de software	26
2.2.1 Técnica de captura de requisitos.....	26
2.2.2 Requisitos Funcionales.....	26



Tabla de contenido

Módulo de captura y catalogación de materiales audiovisuales para el Sistema de Trasmisión de Canales Virtuales

2.2.3	Requisitos No Funcionales	30
2.3	Descripción del sistema	32
2.3.1	Definición de los actores.....	32
2.3.2	Diagrama de Casos de Uso.....	33
2.3.3	Especificación de los Casos de Uso del Sistema.....	34
2.4	Propuesta de la solución	38
	Conclusiones parciales.....	39
DISEÑO DEL MÓDULO DE CAPTURA Y CATALOGACIÓN.....		40
	Introducción al capítulo.....	40
3.1	Diseño	40
3.1.1	Descripción de la arquitectura de software	40
3.2	Patrones	42
3.2.1	Patrones de diseño.....	42
3.2.2	Patrones GoF	44
3.2.3	Patrones J2EE	45
3.3	Diagrama de clases del diseño	45
	Conclusiones parciales	47
IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE CAPTURA Y CATALOGACIÓN.....		48
	Introducción al capítulo.....	48
4.1	Modelo de implementación.....	48
4.1.1	Implementación.....	48
4.1.2	Diagrama de componentes	48
4.1.3	Estándar de codificación	51
4.1.4	Diagrama de despliegue	52
4.1.5	Descripción del diagrama de despliegue.....	53
4.2	Pruebas.....	53
4.2.1	Pruebas de Rendimiento.....	54
4.2.2	Métodos de Pruebas	56
	Conclusiones parciales	59
CONCLUSIONES GENERALES.....		61
RECOMENDACIONES.....		62



Tabla de contenido

Módulo de captura y catalogación de materiales audiovisuales para el Sistema de Trasmisión de Canales Virtuales

REFERENCIAS BIBLIOGRÁFICAS	63
----------------------------------	----



INTRODUCCIÓN

El hombre ha tenido la necesidad de comunicarse con sus semejantes y lo ha hecho a través de diferentes medios de comunicación, los mismos se definen como “la representación física de la comunicación en nuestro mundo; es decir, son el canal mediante el cual la información se obtiene, se procesa y, finalmente, se expresa, se comunica...” (Medina, 2006). La creciente evolución de las Tecnologías de la Información y las Comunicaciones (TIC) y su influencia sobre estos ha provocado una revolución tecnológica a nivel mundial, propiciando el desarrollo de los mismos; esto trae como consecuencia la aparición y el incremento de los materiales audiovisuales.

Audiovisual es un adjetivo que hace referencia conjuntamente al oído y a la vista. La utilización más frecuente del concepto está vinculada al formato de difusión de contenidos que se vale de imágenes ópticas acompañadas por grabaciones acústicas (Definicion.de , 2015), es decir, “...son las imágenes en movimiento y/o a los sonidos grabados, registrados en película, cinta magnética, disco o cualquier otro medio actualmente conocido” (Seapavaa, 2007). Un material audiovisual es visto y oído por el espectador (Definicion.de , 2015). Desde su surgimiento a finales de la década de 1920 su utilización se ha expandido en diferentes esferas sociales: desde grandes empresas e instituciones como: televisoras, productoras de cine, centros educativos hasta las personas comunes, para consumirlos como una forma más de entretenimiento y aprendizaje o para su uso con fines personales.

Actualmente se ha percibido un aumento en la cantidad de materiales audiovisuales en las grandes empresas emergentes en el contexto audiovisual, originando la necesidad de almacenarlos para su posterior uso. Cuando la cantidad de materiales que se almacenan crece surge la problemática de catalogar y capturar los mismos para acceder con mayor facilidad a estos. Con un sistema informático es posible capturar los materiales audiovisuales de manera inteligente; se puede además asociarle metadatos o sea catalogarlos, lo que favorece la búsqueda temática y la realización de las operaciones de gestión de la información de los materiales audiovisuales.

La catalogación de materiales audiovisuales consiste en asociar datos descriptivos a los mismos, lo que puede realizarse de forma manual o mediante un sistema informático que permita el procesamiento y compendio de esta información, con el fin de facilitar la gestión de los materiales que han sido descritos. La catalogación permite contar con la información objetiva y actualizada, depende principalmente de la



forma en que esté organizado todo el conjunto de conocimientos contenidos en diversos materiales documentales, tales como monografías, videos o revistas (ITLA, 2012). Con todo esto se garantiza la perdurabilidad y disponibilidad de la información.

En el centro de desarrollo Geoinformática y Señales Digitales (GEYSED), en la Universidad de las Ciencias Informáticas (UCI), existe una solución que permite automatizar la transmisión, administración y gestión de los canales de radio y televisión que pudieran existir en una entidad: el Sistema de Transmisión de Canales Virtuales (STCV). Se ha concebido como parte del mismo el subsistema de Transmisión, este permite la transmisión automática de la programación a partir de una planificación establecida; el subsistema de Monitorización es el encargado de visualizar en tiempo real la transmisión, así como monitorizar los canales que emite el transmisor; el subsistema de Programación es el encargado de la gestión de todas las planificaciones de transmisión televisiva, también se encarga de la gestión de los usuarios que acceden al sistema; además el STCV cuenta con una Plataforma Interactiva que permite la comunicación entre usuario y sistema.

Aún cuando el STCV cubre las áreas claves de la transmisión de radio y TV, sus usuarios finales experimentan una serie de limitaciones importantes que se describen a continuación:

1. Empleo innecesario de tiempo en la recuperación de materiales audiovisuales, llegando a ser incorrecta en algunos casos. Lo anterior propicia desmotivación en el uso del Sistema y poca confianza en sus resultados por parte de sus usuarios.
2. Imposibilidad de obtener información de materiales audiovisuales con características similares. Esto provoca poca precisión para tomar decisiones de publicación o transmisión e inconformidad por parte de sus usuarios al ver frustrado un intento de localización de determinados materiales audiovisuales.
3. Bajos niveles de usabilidad, ante la engorrosa situación de obtener determinados materiales audiovisuales de otros canales televisivos a través de procesos de descarga o gestión manual como única vía para su posterior inclusión en el STCV. Lo anterior provoca una experiencia de usuario limitada, descontento y hasta abandono del uso del sistema por parte de sus usuarios.

Las limitaciones anteriores afectan considerablemente la introducción del STCV en ambientes reales de despliegue, disminuye el número de clientes potenciales y efectivos del sistema al mismo tiempo que los ingresos obtenidos por concepto de ventas para el Centro GEYSED.



Por todo lo anterior, se plantea como problema de la investigación **¿Cómo disminuir las limitaciones de usabilidad referidas a la identificación, búsqueda y captura de los materiales audiovisuales en el STCV?**

Se define como **objeto de estudio** el proceso de captura y catalogación de materiales audiovisuales, siendo el **campo de acción** la automatización de los procesos de captura y gestión de parámetros de catalogación de materiales audiovisuales en el STCV.

Para dar solución al problema anterior se plantea como **objetivo general**: Desarrollar un módulo para la automatización de los procesos de captura y de gestión de los parámetros de catalogación, que posibilite disminuir las limitaciones de usabilidad en la identificación, búsqueda y captura de materiales audiovisuales en el STCV.

Para dar cumplimiento al objetivo general se definen las siguientes **preguntas científicas**:

¿Cómo se realizan los procesos de captura y catalogación de materiales audiovisuales?

¿Cuáles son las características de los sistemas que permiten capturar y catalogar materiales audiovisuales?

¿Cuáles son las herramientas y tecnologías a utilizar para el desarrollo del módulo propuesto?

¿Cuáles son los elementos que intervienen en el proceso de diseño del módulo de captura y catalogación para el Sistema de Trasmisión de Canales Virtuales?

Se definen las siguientes **tareas de investigación**:

1. Estudio de los procesos relacionados con la captura y catalogación de materiales audiovisuales para la posterior implementación del módulo.
2. Caracterización de soluciones similares existentes a nivel nacional e internacional a fin de identificar funcionalidades que permitan capturar y catalogar materiales audiovisuales.
3. Selección de las herramientas y tecnologías a utilizar para el desarrollo del módulo de captura y catalogación de materiales audiovisuales.



4. Realizar el diseño del módulo de captura y catalogación de materiales audiovisuales para el STCV.
5. Implementar el módulo de captura y catalogación de materiales audiovisuales para el STCV.
6. Evaluación del módulo obtenido aplicando las pruebas de software para validar el correcto funcionamiento del mismo.

Los **métodos científicos** utilizados para el desarrollo de la investigación son:

Métodos Teóricos

- ✓ **Analítico – Sintético:** Análisis: permite la división mental del fenómeno en sus múltiples relaciones y componentes para facilitar su estudio. Síntesis: establece mentalmente la unión entre las partes previamente analizadas, posibilita descubrir sus características generales y las relaciones esenciales entre ellas (Hernández León, 2002). Es utilizado para realizar un estudio de la bibliografía referente a los diferentes conceptos asociados a los procesos de captura y catalogación permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

El presente trabajo de diploma está distribuido en los siguientes capítulos:

Capítulo 1. “Fundamentación teórica del Módulo de captura y catalogación”: En este capítulo, se fundamentan términos técnicos de importancia para la investigación. Se realiza un estudio del arte sobre sistemas que existen en Cuba y el mundo con características similares a las requeridas para la investigación. También se incluyen las herramientas y tecnologías a utilizar.

Capítulo 2. “Características del Módulo de captura y catalogación”: En este capítulo se define el modelo del dominio del sistema que se va a desarrollar. Se describen las clases y flujos del diagrama modelo del dominio. También se identifican los requisitos funcionales y no funcionales. Se definen los casos de usos y actores del sistema.

Capítulo 3. “Diseño del Módulo de captura y catalogación”: En este capítulo se define la arquitectura base del módulo. Además, se describen los artefactos generados en esta fase.

Capítulo 4. “Implementación y prueba del Módulo de captura y catalogación”: En este capítulo se describe la implementación del módulo a través de los diagramas de componentes y de despliegue. Se



Introducción

Módulo de captura y catalogación de materiales audiovisuales para el Sistema de Trasmisión de Canales Virtuales

selecciona el método de prueba y la técnica a utilizar. Además, se describen los casos de prueba a través de los cuales se comprueba la funcionalidad del producto final.



FUNDAMENTACIÓN TEÓRICA

Introducción

En el presente capítulo se identifican los conceptos y elementos teóricos que sustentan la investigación. Se abordan aspectos relacionados con la situación problemática planteada. Además, se realiza un estudio de las soluciones similares existentes. De igual forma se exponen diferentes características de las herramientas y tecnologías a utilizar en el desarrollo de la solución.

1.1 Objeto de estudio

Para la realización de esta investigación es necesario describir los procesos que constituyen el objeto de estudio de la misma: los procesos de captura de materiales audiovisuales y catalogación.

1.1.1 Descripción general del objeto de estudio

Captura: En términos informáticos la captura se define como la transferencia de cualquier tipo de información al disco duro de un ordenador. El proceso de captura de un material audiovisual consiste en la adquisición de todo tipo de materiales audiovisuales a bibliotecas de soporte digital para una posterior gestión de los contenidos de dichos materiales (Tangram solutions, 2014). Existen diferentes formas de realizar la captura, con la utilización de hardware especializado como tarjetas de captura de video, dispositivos que contienen los circuitos necesarios para crear una señal de video digital, o a través de la red, donde se captura un flujo de video.

La captura de materiales audiovisuales y su posterior procesamiento, está definida por una serie de procesos, estos se aplicarán en dependencia del tipo de archivo que se disponga a obtener, en específico audio o video. Para la captura de video es necesario:

- Obtener el video que se está transmitiendo (lo que se quiere capturar).
- Demultiplexar, que es separar los distintos canales de audio y video.
- Descodificar el fichero en dependencia del formato que tenga MPEG, H264, entre otros.
- Descodificar tanto el audio como el video.



- Codificar el audio y el video en el formato que se va a salvar.
- Multiplexar el audio y el video en un único flujo.
- Salvar el fichero en un archivo físico.

El proceso de captura de video se puede realizar de manera automática, sin la participación o monitorización de personas. La captura que se realiza de forma automática se ejecuta a partir de una planificación previamente definida, que permite capturar señales directamente desde un canal de televisión. La manual, se identifica porque son los usuarios quienes guían este proceso.

En el STCV el proceso de captura se utiliza para definir toda aquella operación que se realiza para transferir flujos de videos desde canales externos al sistema o a través de un dispositivo de almacenamiento extraíble hacia el servidor. Para la realización del proceso de captura de video se utilizan tecnologías como *streaming*, VLC¹ y OpenCV². El *streaming* de archivos tipo multimedia es una tecnología que permite transmitir contenido audiovisual desde un servidor hasta uno o varios clientes a través de la red, mediante protocolos de transmisión que posibilitan un uso eficiente del ancho de banda disponible.

Catalogación: La catalogación describe como tal un determinado archivo ya sea: libros, manuscritos, material audiovisual, etc., en sus partes esenciales, esta es una fase del análisis documental que permite la identificación y recuperación de estos, el fin de la misma no es más que un registro bibliográfico o catálogo audiovisual (Clausó García, 1993).

La catalogación tiene como finalidad facilitar la búsqueda y recuperación del propio archivo, además se puede considerar la catalogación como un proceso de generación de metadatos (Metadatos, 2014). Cuando un material audiovisual es archivado en algún medio de almacenamiento es necesario realizar un conjunto de operaciones, como por ejemplo obtener de ellos una serie de datos descriptivos, los cuales son importantes para poder encontrar la información solicitada por un usuario rápidamente.

El documentalista es la persona dedicada a recopilar datos biográficos, informes, noticias, etc., sobre determinado material (RAE, 2014). El documentalista audiovisual es un documentalista en sí mismo, pero

¹ VLC: Es un reproductor multimedia libre y de código abierto multiplataforma y un «framework» que reproduce la mayoría de archivos multimedia, así como DVD, Audio CD, VCD y diversos protocolos de transmisión.

² OpenCV (*Open source Computer Vision library*): Es una librería abierta desarrollado por Intel. Esta librería proporciona un alto nivel funciones para el procesado de imágenes.



especializado, y vinculado al mundo de la imagen en movimiento, el cine y la televisión. Desde el punto de vista terminológico (Documentalistas, 2014). El documentalista audiovisual o catalogador, como es llamado en el Sistema de Trasmisión de Canales Virtuales, es el encargado de identificar, recopilar y almacenar los datos que describirán a los materiales audiovisuales.

La catalogación se centra en tres operaciones destinadas a la representación textual del archivo audiovisual:

- Resumen descriptivo: descripción secuencial, plano a plano, de las imágenes, indicando los tipos de planos, los movimientos de cámara o cualquier otra singularidad que se aprecie. Esta descripción aporta la información precisa y concreta sobre el contenido audiovisual del documento.
- Resumen analítico: resumen del contenido del documento cuya finalidad es la de servir de puente entre la información que proporciona el título y la del resumen descriptivo. En él se indican los sujetos que participan en la acción, el lugar en el que se desarrollan los acontecimientos, el tema principal y los destinatarios del mismo.
- Indización y clasificación: expresión del contenido mediante términos (temáticos/visuales, geográficos y de persona) incluidos en una lista autorizada y clasificación del documento según su ámbito de cobertura (Martínez Tomeu, 2010).

En la presente investigación se propone dinamismo para establecer las entradas de datos sobre los materiales audiovisuales que se deseen catalogar, es decir pudieran emplearse las tres operaciones o ninguna de ellas, esto permite flexibilidad para los clientes, debido a que pueden describir estrictamente un material o no. Se consideró que el uso de estas estuviera definido según las necesidades de la empresa en la cual se empleara el sistema. Para la investigación se definió la utilización del Resumen analítico y la Indización y clasificación, preservando la mayor cantidad de información posible acerca del material audiovisual, lo cual facilitará su recuperación y la gestión de la información contenida sobre los estos.

La investigación se enfoca en la catalogación de materiales audiovisuales, el documento que describe estos será tratado como ficha de catalogación. Los datos descriptivos que se deben considerar para la catalogación vienen determinados por estándares y normas internacionales. Las mismas permiten lograr una adecuada estandarización de este proceso. Por ejemplo las reglas de catalogación IASA (Asociación



Internacional de Archivos Sonoros)³ y FIAT (Federación Internacional Archivos de Televisión)⁴, es una lista de datos mínimos que brindan elementos necesarios que no deben faltar a la hora de realizar la descripción de los archivos de materiales televisivos, ver **Anexo 1**; aunque también pueden adaptarse adicionando campos que se consideren necesarios o eliminando los que no se requieran. De acuerdo a los diferentes tipos de contenidos que se almacenan en el STCV: musicales, documentales, cine, entre otros, para la conformación de la ficha de catalogación del sistema se ha seleccionado una serie de parámetros de la lista de datos descriptivos que ofrece FIAT.

La evolución de los métodos de catalogación de materiales audiovisuales conjuntamente con el desarrollo de la web, ha conllevado a que el proceso sea tan flexible y abarcador como las tecnologías que se utilicen lo permitan. La creación de aplicaciones informáticas es un ejemplo de estos métodos, su uso permite obtener mejor control de las medias, agilizan la descripción textual y ofrecen funcionalidades que aumentan la disponibilidad de estas.

1.1.2 Descripción actual del dominio del problema

En el STCV se ha implementado una aplicación de escritorio para realizar el proceso de catalogación. La misma permite una serie de funcionalidades como: la gestión de los autores, tipologías y géneros de las medias. El usuario realiza la descripción del material audiovisual basándose en una ficha de catalogación que ofrece campos para caracterizar el mismo, introduciendo los datos de acuerdo al contenido que muestra el material audiovisual. La información recogida es almacenada donde puede ser accedida para realizar futuras acciones, tales como su edición y eliminación.

Esta solución se conforma por dos nodos, el nodo Máster: que tiene como funcionalidades la gestión de los materiales audiovisuales, además de responder a peticiones realizadas por los subsistemas del STCV que interactúen con él, y el nodo Servidor: encargado de obtener datos de los servidores que almacenan los materiales audiovisuales. Para garantizar un correcto funcionamiento estos dos nodos deben mantenerse en constante comunicación.

³ <http://www.iasa-web.org>

⁴ <http://www.fiatifta.org/>



1.2 Análisis de otras soluciones existentes

Existen diversas compañías y proyectos nacionales e internacionales que se dedican a la trasmisión y recepción de la televisión, en ellas se han desarrollado aplicaciones para dar solución a las problemáticas de captura y catalogación de materiales audiovisuales, a continuación se describen algunas.

1.2.1 Xtream

La empresa española XTREAM Sistemas de Información Global, cuyos productos están encaminados para dar soluciones de software profesional a necesidades audiovisuales específicas, pertenece al grupo **VITELSA**, este brinda la integración de un completo portafolio de avanzadas soluciones de digitalización, catalogación, gestión, archivo y distribución (IPTV, Web) de contenidos digitales (Grupo VITELSA, 2014).

Dentro de las soluciones de gestión de contenidos Xtream se encuentran: **MEDIABOX** este facilita la catalogación de archivo multimedia (vídeo, audio, documentos electrónicos) desde diferentes tipos de fuentes y distintos formatos. El sistema permite asociar documentos según criterio del usuario (por autor, materia, fechas, entre otros) digitalizando o transcodiando para guardarlos con la calidad deseada (Grupo VITELSA, 2014). Es flexible pues permite la captura de cinta o señal en vivo, brinda la posibilidad de captura continua o planificada. El producto está formado por una serie de módulos cuyas funciones se complementan: captura desde cinta (con magnetoscopio), o desde señal en vivo, captura manual o planificada, posibilidad de captura 24x7, admite diferentes formatos de entrada, posibilidad de marcación de eventos, se puede visualizar lo que se captura, posibilidad de acceso remoto para captura en configuración multicanal (XTREAM, 2014).

La catalogación soporta múltiples tipos de documentos (texto, imágenes, video, audio). Se puede catalogar mientras se codifica, o bien una vez finalizado el almacenamiento. Desde la aplicación de catalogación se pueden generar y borrar *key frames*⁵. El sistema crea metadatos de forma automática cada vez que se crea una plantilla de catalogación. La interfaz muestra un árbol de directorios que define

⁵ Key frames (fotogramas clave) de un video.



un espacio virtual de trabajo para almacenamiento de los assets⁶. Permite catalogar un mismo contenido con múltiples plantillas (XTREAM, 2014).

CICERO es la solución de XTREAM que gestiona de forma integral la grabación, catalogación y distribución de los Juicios Orales celebrados en las salas de vistas de los tribunales, comparencias, etc. (XTREAM, 2014). Dispone de integración con múltiples tecnologías y marcas de los dispositivos de captura de audio y video habitualmente presentes en las salas de vistas. Realiza catalogación, grabación, registro y consulta de audio y/o video para diferentes tipos de juicios y especialidades jurídicas (civil, penal, mercantil, etc.). Realiza la catalogación manual y también automática de las audiencias, permitiendo el registro de cada una de las intervenciones en la sala. Permite la búsqueda de audiencias y declarantes, lo que facilita el acceso individualizado a las declaraciones sin necesidad de localizarlas sobre la línea de tiempo de la audiencia completa utilizando los criterios registrados en la catalogación.

1.2.2 Catalis

Catalis es una herramienta para administrar catálogos, surgió en la Biblioteca Dr. Antonio Monteiro del Instituto de Matemática de Bahía Blanca, dependiente del CONICET y de la Universidad Nacional del Sur, en Argentina. Catalis es distribuido como software de código abierto. Catalis facilita el trabajo con registros MARC 21⁷, ya que su interfaz fue diseñada con el fin de brindar cierta comodidad en el momento de manipular los elementos propios de este formato: campos de datos, campos de control, subcampos, cabecera e indicadores (Catalis, 2014). Brinda la posibilidad de creación dinámica de campos y subcampos. El catalogador puede crear copias de un registro bibliográfico, para agilizar la catalogación de documentos que presenten numerosas características en común.

1.2.3 Tarsys

Tarsys es un software para la gestión de documentos audiovisuales, es una solución de gestión de archivos multimedia. Funciona sobre una base de datos relacional en Oracle y se compone de diversos módulos que pueden funcionar como productos independientes. La base de datos de Tarsys almacena metadatos técnicos, administrativos y de contenidos del material audiovisual. Posee clientes de consulta y

⁶ Los assets son materiales instrucciones independientes y el término puede utilizarse para nombrar un componente o porción de código, una imagen, un código independiente (CSS, JavaScript, etc.), incluso a otro objeto o utilidad.

⁷ Norma de catalogación aplicable a cualquier tipo de materiales.



catalogación y es independiente del gestor de la base de datos. Uno de los módulos por los que se compone Tarsys es Indexer, este brinda dentro de sus funcionalidades la catalogación de videos. Además el gestor multimedia de Tarsys cuenta con una estación de captura de archivos capaz de insertar éstos desde entradas en directo o desde videos controlados de forma remota, así como la transferencia al sistema servidor a partir de la definición de listas de captura. Si se capturan documentos en directo, pueden programarse o bien supervisarse manualmente, mientras que si se hace desde un video es posible definir los puntos de comienzo y fin de los contenidos de las cintas en las listas de captura (Marcos, 2004).

1.2.4 Sistema de captura y catalogación de medias

En la Universidad de las Ciencias Informáticas el Sistema de captura y catalogación de medias tiene como propósito la organización, catalogación, gestión y recuperación de los archivos de video digitalizados. El sistema está dividido en módulos que agrupan las funcionalidades de un proceso específico que se quiera desarrollar. El sistema contiene una solución desarrollada en Symfony, el módulo de catalogación, que permite realizar búsquedas de materiales audiovisuales ya sean básicas, por clasificación, avanzada, combinadas, y la realización de búsquedas de materiales a catalogar, es decir, el sistema permite la realización de búsqueda de videos que están almacenados en el servidor que no hayan sido catalogados. Además permite realizar la modificación de la ficha de catalogación existente editando los datos de la misma en caso de que el archivo anteriormente haya sido catalogado. La reproducción de medias es otra de las funcionalidades que el sistema ofrece, permite realizar acciones de reproducción con el fin de facilitar la catalogación audiovisual y la consulta.

Conclusiones sobre las soluciones existentes

Con el estudio del estado actual de la temática que se aborda en la investigación se pudo apreciar que las diferentes soluciones existentes en la actualidad ofrecen procesos de captura, catalogación, digitalización, compresión, almacenamiento, consulta de audio y video y otras funcionalidades. Las mismas en su mayoría son desarrolladas con tecnologías privativas, con excepción de Catalis y el Sistema de captura y catalogación de medias que no permiten la captura de flujos de video, además este último no admite la gestión de los datos descriptivos de los materiales audiovisuales en el proceso de catalogación, pues solo permite la modificación de su contenido. Tarsys es una solución tecnológica integral que ofrece todos los



procesos pero no es libre y utiliza el software privativo Oracle. El producto MEDIABOX destinado a la catalogación está desarrollado en escritorio y no es multiplataforma al igual que la solución CICERO. Por las características antes mencionadas sobre las soluciones existentes relacionadas a los procesos de captura y catalogación, se concluye que no son útiles para brindar una solución general al problema planteado en esta investigación.

1.3 Herramientas y tecnologías para el desarrollo del software

Con el objetivo de obtener mejores resultados en el desarrollo del módulo propuesto, es necesario seleccionar las tecnologías y herramientas adecuadas a utilizar. Esta selección se realiza teniendo en cuenta las particularidades del STCV. A continuación se enuncian las características más significativas para la solución propuesta.

1.3.1 Metodologías de desarrollo

Las metodologías de desarrollo de software son el marco de trabajo que colecciona un conjunto de pasos y procedimientos que se deben seguir para organizar, controlar y planear el proceso de desarrollo de un software (Joseph Schmuller , 2015). Hoy día existen significativas metodologías que brindan soluciones a los desarrolladores. Su uso posee una enorme importancia a la hora de gestionar de forma eficiente un proyecto.

Proceso Unificado de Desarrollo de Software (RUP)

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. Utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema de software.

A continuación se enuncian las principales características que ofrece RUP para el desarrollo del sistema propuesto:



- Dirigido por casos de uso: el proceso de desarrollo sigue un hilo por el que avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan y en su final son la fuente a partir de la cual los ingenieros de prueba construyen los casos de prueba. Guían además la arquitectura del sistema y maduran junto a esta según avanza el ciclo de desarrollo del software.
- Centrado en la arquitectura: la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
- Iterativo e incremental: las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple con sus objetivos, el desarrollo continúa con la siguiente, incrementando nuevas funcionalidades y repitiendo nuevamente el proceso (Jacobson, y otros, 2000).

La selección de esta metodología fue determinada ya que el módulo a desarrollar pertenece al Sistema de Trasmisión de Canales Virtuales el cual sigue una línea de trabajo sobre RUP, además es conveniente el uso de esta metodología ya que los artefactos generados pueden ser adaptados por el equipo de desarrollo y dependiendo de las necesidades del sistema a implementar, (Torossi, 2010) expresa que el modelo del análisis es opcional. Lo anterior expuesto, facilitará el trabajo con la metodología seleccionada para el desarrollo de la presente investigación.

1.3.2 Lenguaje de Modelado Unificado (UML)

Para la presente investigación se utiliza UML, este es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software que ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien está involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo. Además tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con



los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue (Schmuller, 2000).

1.3.3 Herramienta CASE⁸

Visual Paradigm for UML 8.0

Para la presente investigación se utiliza Visual Paradigm como herramienta de modelado para representar los artefactos de la aplicación. La herramienta CASE Visual Paradigm puede ejecutarse sobre diferentes sistemas operativos. Soporta el ciclo de vida completo del desarrollo de software: análisis, diseño, construcción, prueba y despliegue. Utiliza UML como lenguaje de modelado, ofreciendo soluciones de software que permiten a las organizaciones desarrollar a las aplicaciones de forma rápida y satisfactoria (Visual Paradigm, 2014). Es una herramienta de modelado visual para todos los tipos de diagramas UML. Es compatible con una amplia gestión de casos de uso, lenguaje de especificación de sistemas para diagramas de requisitos y diseño de BD, además entrega los esfuerzos más eficaces en el análisis y diseño de sistemas.

1.3.4 Lenguaje de Programación

Los lenguajes de programación son la base del desarrollo de un sistema informático. Pueden ser usados para controlar el comportamiento de una máquina, especialmente una computadora. Estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas (Alegsa, 2014). Para la presente investigación se hace necesario caracterizar los diferentes lenguajes de programación que se utilizan para el desarrollo del módulo.

Lenguajes del lado del cliente

HTML 5

En la presente investigación se utiliza HTML⁹, una característica especial de HTML 5 es que es el resultado de agrupar las especificaciones relacionadas al desarrollo de páginas web. Posee nuevas prestaciones por lo que se ha elegido para el desarrollo del módulo, ejemplo de estas son:

⁸ CASE: Ingeniería de Software Asistida por Computación.



- Otorga un excelente soporte para utilizar contenido multimedia como lo son audio y video nativamente.
- Proporciona una mayor optimización de la velocidad y un mejor uso del hardware (HTML 5, 2014).
- HTML 5 agrega elementos como audio y video, estas características permiten incluir y controlar contenido multimedia en la web sin tener que recurrir a plugins y APIs propietarias (Alegsa, 2014).

CSS3¹⁰

Es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla.

CSS3 es una especificación desarrollada por el W3C (*World Wide Web Consortium*) para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra, modificando la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos (Pinto, 2015).

Permite el aumento de la accesibilidad de los usuarios gracias a que pueden especificar su propia hoja de estilo, permitiéndoles modificar el formato de un sitio web según sus necesidades.

Su uso permite:

- Lograr una apariencia uniforme en toda la aplicación.
- Hacer que el código HTML sea más fácil de leer.
- Permitir que las páginas se carguen más rápido.
- Posicionar los elementos de la página de una manera más uniforme (Eguíluz, 2009).

JavaScript 1.8.5

⁹ HTML: Del acrónimo en inglés de HiperText Markup Language (en español se traduce como lenguaje de marcado de hipertexto).

¹⁰ CSS: Cascading Style Sheets (Hojas de Estilo en Cascada).



Se ha seleccionado JavaScript en su versión 1.8.5 para el desarrollo del módulo. Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario (de la Torre , 2006).

Es un lenguaje con muchas posibilidades, permite la programación de pequeños *scripts*, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, entre otros. Entre las características que más utilidades pueden aportar al sistema se encuentran:

- Rapidez en la ejecución del código de la aplicación a desarrollar, ya que este lenguaje es interpretado por el navegador que recibe el programa, no se compila.
- No se declaran los tipos de variables, por lo que se ahorra tiempo en la implementación de algunas funcionalidades del sistema (Eguiluz , 2009).
- Permite la ejecución dinámica, JavaScript responde a eventos producidos por el propio usuario en tiempo real (Lecca Risco, 1998).

Lenguajes del lado del servidor

C++

Se ha seleccionado para la implementación del módulo específicamente para el proceso de captura de materiales audiovisuales C++, este es un lenguaje de programación diseñado a mediados de los años 80 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitieran la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido (C++, 2013). Otra de las particularidades que posee C++ y que puede ser aprovechada en la implementación de dicho sistema es que se caracteriza como robusto y suficientemente bien diseñado ya que proporciona una manera de expresar las ideas en programas de manera cómoda y concisa, lo que permite trabajar tanto a bajo como a alto nivel en lo que se refiere a creación de sistemas complejos. Además tiene gran soporte para la programación orientada a objetos (C++, 2013).



PHP 5.3

PHP, acrónimo de "Preprocesador de Hipertexto" por sus siglas en inglés, es un lenguaje muy potente, mundialmente utilizado en la programación de aplicaciones web del lado del servidor. Se ha utilizado el mismo para el desarrollo del módulo pues sus características como lenguaje brindan un aporte significativo a la aplicación.

- Es rápido, de sintaxis cómoda y su sencillez contribuye a su rápido aprendizaje.
- Es un lenguaje multiplataforma.
- Se le pueden agregar extensiones fácilmente y dispone de una gran cantidad de librerías.
- Se puede utilizar como módulo de Apache, lo que lo hace extremadamente veloz.
- No hay que pagar licencias, no limita su distribución y se puede ampliar con nuevas funcionalidades si se desea (Del Castillo, 2006).
- El código es fácil de entender y mantener.
- PHP unido al servidor web Apache, representan las dos herramientas claves para la creación de este tipo de aplicaciones.

Framework de desarrollo

Un framework es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web (Gutiérrez, 2015).

En la actualidad existen gran variedad de frameworks enfocados en el desarrollo de aplicaciones web. A continuación se exponen las características del utilizado para el desarrollo del módulo.

Symfony 2.1.2



Se elige Symfony en su versión 2.1.2 como framework de desarrollo ya que está diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas basadas en Unix como en plataformas Windows. Cumple con las siguientes características:

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos seleccionado para el desarrollo de las aplicaciones.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa (LibrosWeb, 2015).

Qt 5.2.1

Qt es un framework para el desarrollo de aplicaciones multiplataforma. Se utilizará en el desarrollo del módulo en su versión 5.2.1. Este framework utiliza C++ de manera nativa lo cual representa una ventaja para el desarrollo porque se acopla con el lenguaje seleccionado para la implementación del módulo. Tiene un tratamiento de memoria que permite que el tiempo de ejecución de este sea menor que los programas escritos en Java o C# (SOFTPEI, 2013).



Doctrine 2.3.0

Doctrine es un ORM (Mapeo de Objetos de Base de Datos) de la siglas en inglés *object-relational mapping*. Es una interfaz que traduce la lógica de los objetos a la lógica relacional. Esto permite acceder de manera efectiva a la base de datos desde un contexto orientado a objetos sobre el lenguaje de programación PHP 5.2.3 o superior. Doctrine facilita el trabajo con los datos a los desarrolladores, automatizando la generación de ficheros que responden a las tablas relacionales de la base de datos (Potencier , 2009).

1.3.5 Entorno Integrado de Desarrollo

Un Entorno Integrado de Desarrollo (IDE) es un programa que está compuesto por un conjunto de herramientas para un programador. Estos programas contienen un conjunto de herramientas de programación que permiten el desarrollo de otras aplicaciones en determinado lenguaje. Estos pueden realizar las funciones de un editor de código, un compilador, depurador y hasta un constructor de interfaz gráfica (Lau, 2013). A continuación se describen las características de los utilizados.

NetBeans 7.3

NetBeans es un IDE escrito en Java, pero puede servir para otros lenguajes de programación. Es un producto libre y gratuito sin restricciones de uso. Entre sus principales características están que permite el desarrollo rápido y fácil de aplicaciones Java de escritorio, móviles y aplicaciones web, así como aplicaciones HTML5, JavaScript y CSS. El IDE también proporciona un gran conjunto de herramientas para desarrolladores PHP y C/C++, por lo que se ha seleccionado en su versión 7.3 (NetBeans, 2015).

Qt Creator 3.0.1

Qt Creator es un IDE para la creación de aplicaciones con el framework Qt. Las características claves de Qt Creator permiten a los programadores realizar las siguientes tareas:

- Empezar a desarrollar aplicaciones con Qt rápida y fácilmente con el asistente de proyectos, y acceder rápidamente a proyectos recientes y sesiones.



- Desarrollar aplicaciones con el editor de código avanzado de C++ que provee nuevas y poderosas características para el completamiento de código y viendo el esquema de archivos (que es, la jerarquía de símbolos de un archivo).
- Acceder fácilmente a información con el módulo del sistema de ayuda contextual de Qt (Ubuntu, 2012).

1.3.6 Bibliotecas de desarrollo

Ajax

En el artículo “AJAX: A New Approach to Web Applications”, publicado en febrero del 2005, por Jesse James Garrett, este expresa: “AJAX no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de manera autónoma...” (Eguíluz Pérez , 2005).

AJAX es una técnica para la creación de páginas web rápidas y dinámicas. Permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

AJAX es soportado por muchos exploradores web, entre los que se cuentan los más conocidos: Firefox, Internet Explorer, Chrome, Safari y Ópera. Esta tecnología es de gran utilidad para realizar las funcionalidades sobre la estructura de datos, ya sea para cargar las fichas o guardarlas o evitando que la interfaz visual sobre la que trabaja el usuario se refresque cada vez que realice alguna operación (LibrosWeb, 2015).

JQuery 2.0.2

Se utiliza JQuery ya que es uno de los complementos más esenciales para el desarrollo web usado en la actualidad, ya que facilita el desarrollo de aplicaciones enriquecidas del lado del cliente, en Javascript, compatibles con todos los navegadores, además es compatible con CSS3 (JQuery, 2015).

Entre sus principales ventajas se encuentran:

- Es flexible y rápido para el desarrollo web.
- Excelente integración con AJAX.



- Brinda gran cantidad de plugins adaptables al tipo de proyecto que se esté desarrollando.

1.3.7 Servidor web

Un Servidor Web es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML. La principal característica de los Servidores Web es almacenar los archivos de un sitio y ser publicados en una red para que pueda ser visitado por los usuarios (Carlos Leyva, 2014).

Apache 2.2

Apache es la plataforma de servidores Web de código abierto que posee un conjunto de características que fueron de vital importancia. Se puede utilizar en múltiples sistemas operativos, es personalizable, el servidor puede ser manipulado vía línea comandos, lo que hace la administración remota muy conveniente (APACHE, 2012).

Algunas de sus características son:

- **Fiabilidad:** alrededor del 90% de los servidores con más alta disponibilidad funcionan con Apache.
- **Gratuidad:** es totalmente gratuito y se distribuye bajo la licencia *Apache Software License*, que permite la modificación del código.
- **Extensibilidad:** se pueden añadir módulos para ampliar sus capacidades. Hay una amplia variedad de módulos que permiten desde generar contenido dinámico, monitorizar el rendimiento del servidor, atender peticiones encriptadas por SSL, hasta crear servidores virtuales por IP o por nombre y limitar el ancho de banda para cada uno de ellos. Dichos módulos pueden ser creados por cualquier persona con conocimientos de programación (González, y otros, 2014).

1.3.8 Sistema Gestor de Base de Datos

Un Sistema Gestor de Bases de Datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a estos. El objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos. Es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporcionar acceso controlado a la misma (Slideshare, 2014).



PostgreSQL 9.3

Se utiliza PostgreSQL en su versión 9.3, el mismo es un sistema gestor de base de datos objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution* o Distribución de Software Berkeley) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más usado. (PostgreSQL-es, 2013). Es multiplataforma, permite la integración con los framework utilizados (Symfony y Qt), siendo compatible con estos.

Conclusiones parciales

En este capítulo se precisaron una serie de conceptos relacionados con la problemática planteada, estudiándose el desarrollo y la evolución de los sistemas de captura y catalogación de materiales audiovisuales, tanto a nivel nacional como internacional, analizándose detalladamente sus ventajas y características generales.

- ✓ El estudio de los procesos de captura y catalogación, propició una descripción amplia del objeto de estudio, lo cual garantizó un mejor entendimiento sobre el funcionamiento de estos dos procesos.
- ✓ El análisis de soluciones relacionadas a los procesos de captura y catalogación, permitió concluir que ninguna satisface todos los requisitos deseados, pero sirvió de guía para identificar los posibles parámetros iniciales que tendrá la ficha de catalogación. Se identificaron un conjunto de funcionalidades comunes como: la gestión de parámetros descriptivos y la captura de videos de forma planificada. Además de un grupo de tecnologías que son usadas en los sistemas de gestión audiovisual, entre ellas se destacan: el framework Doctrine y las bibliotecas de desarrollo Ajax y jQuery que sirven de guía para el desarrollo del módulo que se desea implementar.
- ✓ La caracterización de las tecnologías y herramientas propuestas para la elaboración de la solución, permitió conocer las ventajas que ofrecen las mismas para el desarrollo del módulo de captura y catalogación de materiales audiovisuales.



CARACTERÍSTICAS DEL MÓDULO DE CAPTURA Y CATALOGACIÓN

Introducción al capítulo

En el presente capítulo se definen los temas relacionados con las características del módulo que se desea desarrollar. Definiéndose el modelo de dominio con el objetivo de comprender la problemática planteada. También se identifican los requisitos funcionales y no funcionales los cuales constituyen un eslabón fundamental en el resultado final del desarrollo del módulo. Se representan además, los casos de uso y actores del sistema.

2.1 Modelo de dominio

El modelo del dominio muestra clases conceptuales significativas en un dominio del problema. Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés (Larman, 2003).

2.1.1 Descripción del flujo del diagrama del modelo de dominio

El Sistema de Trasmisión de Canales Virtuales integra varios subsistemas (Programación, Trasmisión, Monitorización) y una Plataforma Interactiva que realizan varias operaciones para posibilitar la correcta transmisión de canales de radio y televisión. En el Módulo de gestión y almacenamiento audiovisual, el catalogador es el encargado de copiar los materiales audiovisuales al servidor, donde a cada uno de ellos se le asocian datos descriptivos.

2.1.2 Diagrama de Modelo de dominio

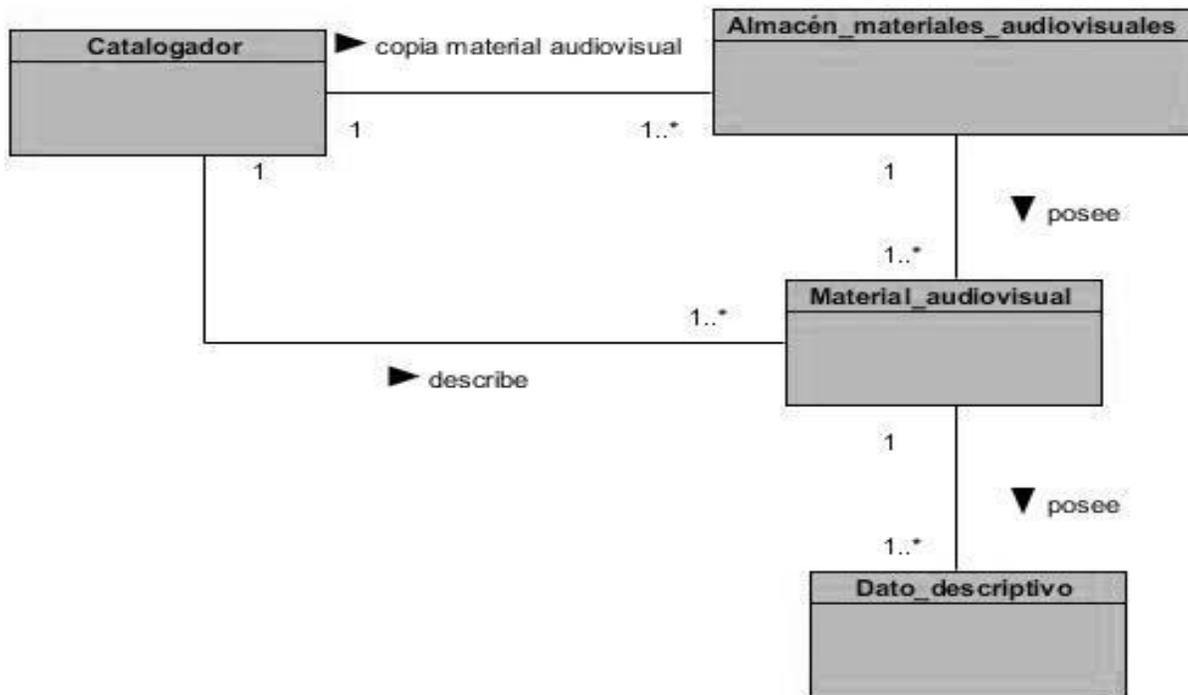


Imagen 1: Modelo de dominio.

2.1.3 Descripción de conceptos del dominio del problema

Catalogador: Usuario con permisos de catalogación encargado de gestionar los datos descriptivos asociados a cada material audiovisual. Además de copiar los materiales audiovisuales al almacén de materiales audiovisuales.

Material audiovisual: Media física sobre la cual se realizan las operaciones de catalogación en el sistema.

Dato descriptivo: Parámetro definido para realizar la catalogación al material audiovisual.

Almacén de materiales audiovisuales: Es donde se almacenan los materiales audiovisuales para su posterior uso y procesamiento en el sistema.



2.2 Especificación de los requisitos de software

Un requisito es una “condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado”. También se aplica a las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación. Estos se clasifican en funcionales y no funcionales (Laguna, 2013). La especificación de los requisitos de software describe el comportamiento del sistema que se desea desarrollar donde incluye los requisitos funcionales y los requisitos no funcionales.

2.2.1 Técnica de captura de requisitos

Para la captura de requisitos del módulo de captura y catalogación de materiales audiovisuales se utilizó la técnica de recopilación de información:

Entrevista: La misma se define como una conversación planificada entre el investigador y el entrevistado para obtener información. La entrevista no estructurada, no lleva un cuestionario rígido y puede variar de una persona a otra, es flexible. Se aplica a especialistas en el tema, es una forma de obtener criterios de expertos (Hernández León, 2002).

Su utilización permitió extraer los requisitos de una forma más segura, debido a que en ocasiones los usuarios pueden tener dificultad para describir sus tareas o dejar información importante sin especificar.

2.2.2 Requisitos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Los requisitos funcionales son aquellos requisitos que describen la interacción entre el sistema y su ambiente, independientemente de su implementación. El ambiente incluye al usuario y cualquier otro sistema externo que interactúa con el sistema (Quiroga, 2013).

A continuación se enuncian los requisitos funcionales del módulo de captura y catalogación de materiales audiovisuales:

Nota: Se utiliza el prefijo RF en su nomenclatura.



RF1: Autenticar usuario: El sistema debe brindar la posibilidad de autenticar a los usuarios, identificando el mismo y autorizando el acceso a la aplicación según su rol para realizar las funciones definidas para el mismo.

RF2: Realizar búsqueda básica: El sistema permitirá la realización de búsqueda que se basará en acotar los elementos de la lista sobre la que se aplique en dependencia de los caracteres introducidos por el usuario.

RF3: Realizar búsqueda por filtros: El sistema permitirá realizar búsquedas definiendo filtros relativos a la descripción del material, lo que posibilitará la búsqueda más específica de materiales audiovisuales.

RF4: Adicionar género a un material audiovisual: El sistema debe ser capaz de adicionar un género para los materiales audiovisuales.

RF5: Editar género a un material audiovisual: El sistema debe ser capaz de editar el género de un material audiovisual. Esto permitirá modificar la información del género almacenada hasta el momento en el sistema.

RF6: Eliminar género de un material audiovisual: El sistema debe ser capaz de eliminar un género de los materiales audiovisuales.

RF7: Listar género de los materiales audiovisuales: El sistema debe ser capaz de mostrar el listado de los géneros de los materiales audiovisuales existentes en el sistema.

RF8: Adicionar tipología a un material audiovisual: El sistema debe ser capaz de adicionar una tipología de los materiales audiovisuales.

RF9: Editar tipología a un material audiovisual: El sistema debe ser capaz de editar la tipología de un material audiovisual. Esto permitirá modificar la información de la tipología almacenada hasta el momento en el sistema.

RF10: Eliminar tipología de un material audiovisual: El sistema debe ser capaz de eliminar una tipología de materiales audiovisuales.



RF11: Listar tipología de los materiales audiovisuales: El sistema debe ser capaz de mostrar el listado de las tipologías de los materiales audiovisuales existentes en el sistema.

RF12: Adicionar autor a un material audiovisual: El sistema debe ser capaz de adicionar un autor de los materiales audiovisuales.

RF13: Editar autor a un material audiovisual: El sistema debe ser capaz de editar el autor de un material audiovisual. Esto permitirá modificar la información del autor almacenado hasta el momento en el sistema.

RF14: Eliminar autor de un material audiovisual: El sistema debe ser capaz de eliminar un autor de un material audiovisual.

RF15: Listar autores de los materiales audiovisuales: El sistema debe ser capaz de listar los autores de los materiales audiovisuales.

RF16: Adicionar ficha de catalogación de archivo: El sistema debe permitir introducir los datos descriptivos de un material audiovisual local.

RF17: Editar ficha de catalogación de archivo: El sistema debe permitir editar los datos descriptivos de un material audiovisual local.

RF18: Eliminar ficha de catalogación de archivo: El sistema debe permitir eliminar los datos descriptivos de un material audiovisual local.

RF19: Visualizar ficha de catalogación de archivo: El sistema permitirá mostrar la ficha de catalogación seleccionada. Esto permitirá mostrar la información almacenada de los materiales audiovisuales en su ficha de catalogación correspondiente.

RF20: Listar fichas de catalogación de archivos: El sistema debe ser capaz de listar las fichas de catalogación existentes de los materiales audiovisuales locales.

RF21: Adicionar parámetros de catalogación: El sistema debe permitir adicionar parámetros de catalogación, es decir aumentar los datos descriptivos por los que se cataloga un material audiovisual.



RF22: Editar parámetros de catalogación: El sistema debe permitir editar los parámetros de catalogación que hayan sido añadidos, es decir modificar los datos descriptivos por los que se cataloga un material audiovisual.

RF23: Eliminar parámetros de catalogación: El sistema debe permitir eliminar parámetros de catalogación que hayan sido añadidos, es decir eliminar los datos descriptivos por los que se cataloga un material audiovisual.

RF24: Listar parámetros de catalogación: El sistema debe permitir listar los nuevos parámetros de catalogación que hayan sido añadidos.

RF25: Adicionar ficha de catalogación de flujos de video: El sistema debe permitir adicionar una ficha de catalogación de flujos de video, en la cual se reflejará una planificación mostrando: hora de inicio y fin, dirección ip y la fecha, dichos datos serán consultados a través de la base de datos por el sistema de captura.

RF26: Editar ficha de catalogación de flujos de video: El sistema debe permitir editar una ficha de flujos de video ya almacenada, la cual será modificada por el usuario cuando este considere necesario.

RF27: Eliminar ficha de catalogación de flujos de video: El sistema debe permitir eliminar una ficha de captura ya almacenada, la cual será eliminada por el usuario cuando este considere necesario.

RF28: Listar fichas de catalogación de flujos de video: El sistema debe ser capaz de listar las fichas de catalogación de flujos de video existentes.

RF29: Capturar flujo de video: El sistema permitirá capturar un flujo de video que se esté transmitiendo en ese momento en un canal virtual externo al STCV. Con el objetivo de su posterior catalogación.

RF30: Obtener planificación: El sistema de captura permitirá obtener las planificaciones almacenadas en la base de datos, estas planificaciones se actualizarán cada vez que se inserten, editen o eliminen los datos referentes a alguna de ellas.



2.2.3 Requisitos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. Por lo tanto, pueden especificar el rendimiento del sistema, la protección, la disponibilidad y otras propiedades emergentes. Se identificaron en este trabajo los siguientes requisitos no funcionales:

Nota: Se utiliza el prefijo RNF en su nomenclatura.

- **Requisitos de usabilidad**

RNF1: El sistema deberá ser manejado por usuarios con conocimientos básicos en informática.

Requisitos de software

En las estaciones clientes:

RNF2: Windows 7/8 o Ubuntu desde 12.04 hasta 14.04.

RNF3: Navegador: Mozilla Firefox 34.

En los servidores:

RNF4: Sistema Operativo: Ubuntu 14.04.

RNF5: Sistema Gestor de Base de Datos: PostgreSQL 9.1.

RNF6: Aplicación Servidor Web: Apache 2.2.



- **Requisitos de hardware**

En las estaciones cliente:

RNF7: Procesador recomendado: Core2Duo.

RNF8: RAM: 1 GB (mínimo recomendado).

RNF9: Velocidad de conexión recomendada: 100 Mbps.

En los servidores:

RNF10: Procesador recomendado: Core2Duo o superior.

RNF11: Memoria RAM: 2 GB (mínimo recomendado).

- **Requisitos en el diseño y la implementación**

RNF12: El lenguaje de programación a utilizar en la implementación del módulo será C++ y PHP 5.3 del lado del servidor, JavaScript 1.8.5, CSS3 y HTML 5 del lado del cliente. Como IDE de desarrollo NetBeans 7.3 y Qt Creator 3.0.1; el framework Symfony 2.1.2 y Qt 5.2.1.

- **Requisitos de apariencia o interfaz externa**

RNF13: El sistema debe tener la identidad marcaría representada en el STCV.

RNF14: Todos los mensajes deben estar escritos en idioma español, con letra inicial mayúscula y punto final.

- **Requisitos de seguridad**

Confidencialidad:

RNF15: El sistema debe proteger la información y datos de manera que los usuarios no autorizados no puedan leerlos o modificarlos.

Integridad:

RNF16: La información solo puede ser modificada por quien está autorizado y de manera controlada, en este caso los usuarios con privilegio de catalogador.



2.3 Descripción del sistema

Luego de la identificación de los requisitos funcionales del Módulo de captura y catalogación de materiales audiovisuales estos se agruparon en casos de uso (CU) donde posteriormente se representan en el diagrama de casos de uso del sistema. Los casos de uso engloban un conjunto de acciones relacionadas con el actor del sistema, este último inicializa dichas acciones.

2.3.1 Definición de los actores

Un actor representa un conjunto coherente de roles que los usuarios de los casos de uso juegan al interactuar con el sistema. Normalmente, un actor representa un rol que es jugado por una persona, un dispositivo de hardware, una BD o incluso otro sistema que interactúe (NEYEM, 2013). Los actores identificados para el Módulo de captura y catalogación de materiales audiovisuales se describen a continuación.

Tabla 1. Descripción del actor del sistema

Actor	Descripción
Catalogador	Es el encargado de realizar todas las operaciones de catalogación sobre los materiales audiovisuales.
Reloj	Es el encargado de obtener la planificación existente para la realización del proceso de captura.



2.3.2 Diagrama de Casos de Uso

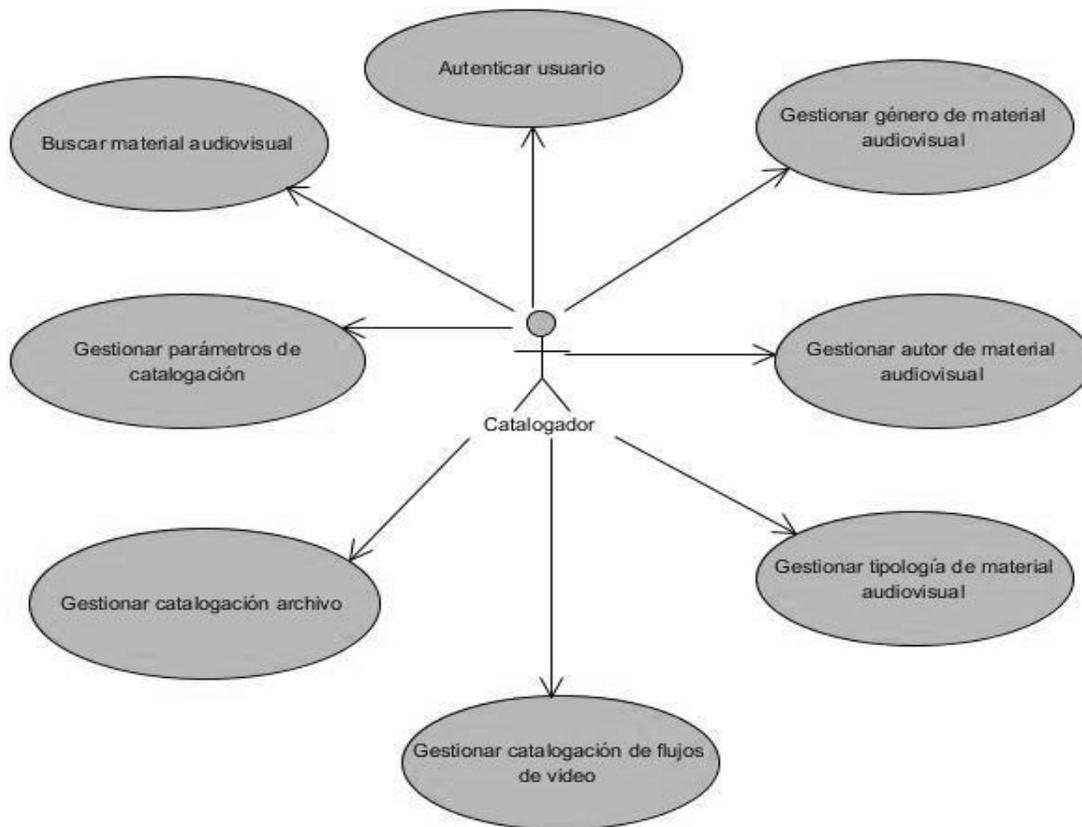


Imagen 2: Diagrama de casos de uso del sistema de catalogación.

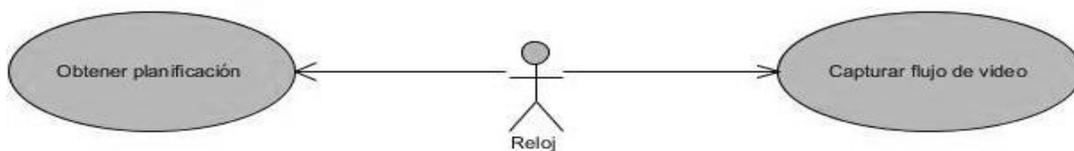


Imagen 3: Diagrama de casos de uso del sistema de captura.



2.3.3 Especificación de los Casos de Uso del Sistema

La descripción de los casos de uso facilita la comprensión de las diferentes funcionalidades del sistema. A continuación se describe el caso de uso Gestionar tipología de material audiovisual. Las descripciones de los restantes CU pueden encontrarse en el **Anexo 2** de la presente investigación.

Tabla 2. Descripción del caso de uso Gestionar tipología de material audiovisual.

Objetivo	Gestionar las Tipologías de los materiales audiovisuales del sistema.	
Actores	Catalogador	
Resumen	El Caso de Uso se inicializa cuando el catalogador selecciona la opción Gestionar/Tipología de material audiovisual, ya sea para adicionar u obtener el listado de las tipologías existentes en el sistema.	
Complejidad	Media	
Prioridad	Crítica	
Precondiciones	El usuario debe estar autenticado en el sistema.	
Postcondiciones	Las tipologías quedan gestionadas.	
Flujo de eventos		
Flujo básico Gestionar tipología de material audiovisual		
	Actor	Sistema
1	El Caso de Uso se inicializa cuando el usuario selecciona la opción Gestionar/Tipología.	
2		El sistema muestra una lista de las diferentes opciones del menú seleccionado que contiene: <ul style="list-style-type: none"> • Adicionar. • Mostrar.
3	El catalogador escoge una de las opciones que brinda el sistema para gestionar tipología.	
4		El sistema ejecuta la operación seleccionada por el usuario: <ul style="list-style-type: none"> • Si selecciona la opción Adicionar ir a la sección 1. • Si selecciona la opción Mostrar ir a la sección 2.
5		El Caso de Uso termina cuando son gestionadas las tipologías de los materiales audiovisuales.
Sección 1: “Adicionar tipología de un material audiovisual”		
Flujo básico Adicionar tipología de un material audiovisual		
	Actor	Sistema
1.	El usuario selecciona la opción Adicionar.	
2.		Muestra la interfaz de Adicionar tipología, con el campo Nombre.
3.	Inserta el Nombre de la tipología que desea añadir. Ver prototipo de interfaz 1.	
4.		Se validan los datos de la inserción de la tipología. En el



		campo Nombre sólo deben existir letras y espacios. (Ver la acción 1a de los Flujos Alternos.) Volver al Flujo básico Gestionar tipología de material audiovisual.
5.	El usuario selecciona la opción Guardar.	
6.		El sistema guarda los datos. (Ver la acción 1b de los Flujos Alternos.)

Prototipo de Interfaz 1.



Flujos alternos

Nº Evento 1a: Datos Incorrectos.

	Actor	Sistema
1.	El usuario incumple con alguno de los caracteres establecidos para la inserción del Nombre de la tipología o el campo esté vacío.	
2.		El sistema muestra un mensaje de error donde precisa que el dato de entrada no es correcto o el campo está vacío.

Flujos alternos

Nº Evento 1b: Opción Volver al listado.

	Actor	Sistema
1.	El usuario selecciona la opción Volver al listado.	
2.		El sistema cancela la acción y regresa a la interfaz del listado de tipologías.

Sección 2: "Mostrar"

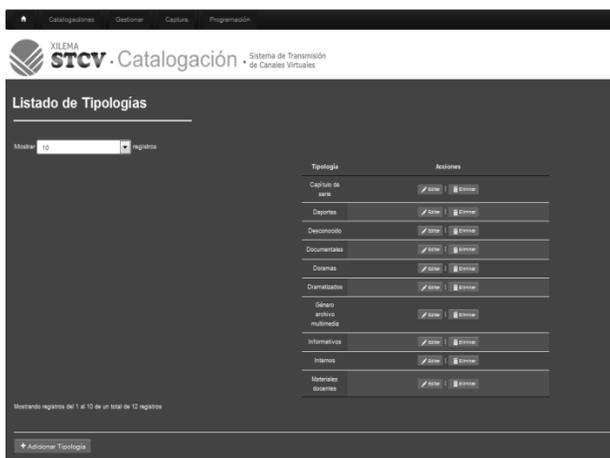
Flujo básico Mostrar

	Actor	Sistema
1.	El usuario selecciona la opción Mostrar.	
2.		El sistema muestra una interfaz con todas las tipologías existentes en el sistema. Además de las opciones Editar y Eliminar.
3.	El usuario selecciona una de las siguientes opciones: <ul style="list-style-type: none"> - Editar - Eliminar 	



	Ver prototipo de interfaz 2.	
4.		El sistema realiza la operación seleccionada por el usuario: 1. Si selecciona la opción Editar ir a la sección 2.1. 2. Si selecciona la opción Eliminar ir a la sección 2.2.

Prototipo de Interfaz 2.



Sección 2.1: “Editar”

Flujo básico Editar

	Actor	Sistema
1.	El usuario selecciona la opción Editar.	
2.		El sistema muestra una interfaz con la tipología seleccionada.
3.	El usuario modifica la tipología seleccionada. Ver prototipo de interfaz 3.	
4.		El sistema verifica los datos modificados (Ver la acción 2.1a de los Flujos Alternos.)
5.	El usuario selecciona la opción Guardar.	
6.		El sistema guarda los datos. (Ver la acción 2.1b de los Flujos Alternos.) Volver a la sección 2.

Prototipo de Interfaz 3.



Flujos alternos

Nº Evento 2.1a: Datos Incorrectos.

	Actor	Sistema
1.	El usuario incumple con alguno de los caracteres establecidos para la edición del Nombre de la tipología o el campo esté vacío.	
2.		El sistema muestra un mensaje de error donde precisa que el dato de entrada no es correcto o el campo está vacío.

Flujos alternos

Nº Evento 2.1b: Opción Volver al listado.

	Actor	Sistema
1.	El usuario selecciona la opción Volver al listado.	
2.		El sistema cancela la acción y regresa a la interfaz del listado de tipologías.

Sección 2.2: "Eliminar"

Flujo básico Eliminar

	Actor	Sistema
1.	El usuario selecciona la opción Eliminar.	
2.		El sistema muestra un mensaje de confirmación.
3.	El usuario confirma la acción, presionando en el botón Aceptar.	
4.		El sistema elimina la tipología seleccionada. (Ver la acción 2.2a de los Flujos Alternos.) Volver a la sección 2.

Flujos alternos

Nº Evento 2.2a: Opción Cancelar.

	Actor	Sistema
1.	El usuario cancela la acción, presionando en el botón Cancelar.	
2.		Cancela la operación realizada, cierra el formulario. Termina el CU.

Relaciones	CU Incluidos	No procede.
	CU Extendidos	No procede.

Requisitos no funcionales	
----------------------------------	--



Requisitos funcionales	RF8, RF9, RF10, RF11
Asuntos pendientes	No procede.

2.4 Propuesta de la solución

El Módulo de captura y catalogación de materiales audiovisuales para el STCV, pretende con los requisitos identificados realizar los procesos de captura y catalogación y que estos permitan un correcto manejo de la información y almacenamiento en el sistema de los materiales audiovisuales. El catalogador como usuario autorizado podrá realizar el proceso de catalogación donde gestionará todos los datos descriptivos relacionados con un material audiovisual: gestionará las tipologías, géneros, autores pertenecientes a las medias. Además posibilitará añadir, modificar y eliminar los parámetros de catalogación que hayan sido añadidos por el usuario. En la ficha de catalogación se muestran los parámetros escogidos del listado que propone FIAT: Título (Nombre), Fecha de emisión (Año), Lugar de emisión (País), Duración y Contenido (Descripción) además de otros como: Género, Tipología, Autor, Capturado y VOD. Se tienen en cuenta estos parámetros, debido a que en el análisis de las soluciones existentes varios son comunes en las fichas de catalogación de los sistemas: Sistema de captura y catalogación de medias y del Módulo de gestión y almacenamiento audiovisual para el STCV.

Las fichas creadas estarán en correspondencia al tipo de obtención del material audiovisual. Para las medias capturadas las fichas de catalogación mostrarán además los siguientes parámetros: Dirección Ip, Fecha, Hora Inicio y Hora Fin, los cuales representarán la planificación del proceso de captura. Mientras que para las medias obtenidas a través de un medio de almacenamiento extraíble se mostrará la opción Examinar, desde la cual se podrá acceder al directorio especificado y cargar la media deseada. Podrá realizarse en el sistema una búsqueda básica introduciendo caracteres o una búsqueda a través de filtros, permitiendo la obtención de forma rápida del material audiovisual deseado. Para la realización de la captura de flujos de videos se desarrolla una aplicación de escritorio que obtendrá una planificación de la ficha de catalogación almacenada en base de datos y ejecutará dicho proceso utilizando la biblioteca gstreaming-1.0 perteneciente al framework Streaming.



Conclusiones parciales

En este capítulo se precisaron las principales características del módulo de captura y catalogación y se logró obtener una visión general de las mismas. La generación de los artefactos que propone la metodología seleccionada permitió entender detalladamente el contexto del sistema arribando a las siguientes conclusiones:

- ✓ Para un mejor entendimiento del funcionamiento actual del sistema y entender la situación problemática planteada y por lo tanto contribuir a la comprensión de los requisitos que se desglosan de este contexto se definió el modelo de dominio, el cual proporcionará al equipo de trabajo utilizar un lenguaje común durante el desarrollo del sistema. Se describieron los principales conceptos asociados al mismo.
- ✓ La especificación de requisitos permitió describir el comportamiento del sistema que se desea desarrollar, lográndose con esto una mejor organización en las funcionalidades, creando así las bases para el desarrollo de la aplicación.
- ✓ Con la identificación y descripción de los actores del sistema y la especificación de casos de uso se agruparon los requisitos funcionales, permitiendo una vista clara de su interacción con el sistema y la relación de los mismos con los actores.



DISEÑO DEL MÓDULO DE CAPTURA Y CATALOGACIÓN

Introducción al capítulo

Se decidió prescindir de la fase de análisis, debido a que RUP es una metodología configurable en cuanto a sus artefactos, los requisitos a implementar son conocidos y se cuenta con una comprensión de los mismos, lo que permite prescindir del refinamiento de los requisitos e ir directo al diseño. En el presente capítulo se explica cómo transformar los requisitos en los productos de trabajo que especifica el diseño del software. Además se realiza el modelado de los artefactos que se deben generar en esta etapa del desarrollo de software, los cuales se encargan de describir cómo funcionan los casos de uso del sistema, dando paso a la implementación del mismo.

3.1 Diseño

Según Pressman, el diseño del software es realmente un proceso de muchos pasos pero que se clasifican dentro de uno mismo. Es donde se modela el sistema y su forma, en general, se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software (Puebla, 2009).

El diseño de la arquitectura de software ocurre inmediatamente después de la especificación de los requerimientos de software y considera como elementos principales los siguientes: componentes de software, propiedades de dichos componentes y la comunicación entre ellos. El diseño detallado se lleva a cabo justo antes de la codificación, y forma parte de las primeras tareas del desarrollador; describe la lógica, el control jerárquico, estructura de datos, empaquetado de componentes, entre otros (Suárez Hernández., 2008).

3.1.1 Descripción de la arquitectura de software

La arquitectura de software permite representar de forma concreta la estructura y funcionamiento interno de un sistema, lo que posibilita su entendimiento, la organización del desarrollo del mismo. “La arquitectura de software de un sistema es la estructura o estructuras del sistema, lo cual abarca componentes de software, las propiedades visibles externamente de esos componentes, y las relaciones entre ellas” (Suárez Hernández., 2008).



Arquitectura Cliente-Servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. En la presente investigación se define esta arquitectura, donde la capa vista en la aplicación web, será la encargada de realizar peticiones al servidor, comprendido por el modelo, la capa controladora y el sistema de captura, dando respuesta a las peticiones realizadas, comunicándose a través de la base de datos donde se guardarán la planificaciones existentes.

Arquitectura Modelo-Vista-Controlador

Teniendo en cuenta las características de la presente investigación, se utiliza para el desarrollo de la aplicación web el patrón arquitectónico Modelo-Vista-Controlador (MVC), ya que el uso del framework Symfony se identifica con este patrón. MVC pertenece al conjunto de estilos en llamada y retorno. Se basa además en la reutilización de código y la separación de conceptos, características que buscan facilitar las tareas de desarrollo en la aplicación y su posterior mantenimiento. El mismo se compone de tres componentes, el modelo, la vista y el controlador, permitiendo separar la lógica de negocio, la interfaz de usuario y las entidades encargadas de gestionar las comunicaciones.

Modelo o capa de datos: contiene los componentes que representan y gestionan los datos manejados por la aplicación, contiene mecanismos para acceder a la información y también para actualizar su estado.

Vista o capa de presentación: los componentes de esta capa son responsables de mostrar al usuario el estado actual del modelo de datos y presentarle las distintas acciones disponibles, contiene el código de la aplicación que va a producir la visualización de las interfaces de usuario.

Capa de control: conoce los métodos y atributos del modelo, recibe y realiza lo que el usuario quiere hacer ya que contiene los componentes necesarios para responder a las acciones que se solicitan en la aplicación, además gestiona la aplicación de la lógica de negocio sobre el modelo de datos, y determina qué vista debe mostrarse a continuación (MVC, 2015).

Arquitectura N capas



Se seleccionó para la aplicación de escritorio el patrón en N capas por las facilidades que brinda en el desarrollo del sistema. Esta arquitectura pertenece al estilo de llamada y retorno. En Garlan y Shaw definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior (Garlan, y otros, 1994). Las capas suelen ser entidades complejas, compuestas de varios paquetes o subsistemas.

En la aplicación se utilizará la arquitectura 2 capas, variante de la arquitectura N capas que consiste en separar un proyecto en la Capa lógica del negocio y la Capa de Acceso a Datos.

Capa de Negocio: es donde se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todos los procesos que deben realizarse en el sistema.

Capa de Acceso a Datos: es donde se realiza todo el almacenamiento de datos. Recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

3.2 Patrones

El patrón es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. Muchos patrones ofrecen orientación sobre cómo asignar las responsabilidades a los objetos ante determinada categoría de problemas. Expresado lo anterior con palabras más simples, el patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas (Schmuller, 2003).

3.2.1 Patrones de diseño

Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. GRASP (patrones generales de software para asignar responsabilidades) es un acrónimo que significa *General Responsibility Assignment Software Patterns*. El



nombre se eligió para indicar la importancia de captar (*grasping*) estos principios, si se quiere diseñar eficazmente el software orientado a objetos (Schmuller, 2003). A continuación se exponen los patrones que se evidencian en el módulo propuesto.

- ✓ **Patrón Experto:** Tiene como objetivo asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. En la aplicación web cada clase mantiene este principio, por ejemplo la entidad *Autor.php* es la experta en conocer todo lo relacionado con los autores, por tanto es aquí donde se realizan las acciones referentes al mismo.
- ✓ **Patrón Creador:** El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. Es decir, guía la asignación de responsabilidades relacionadas con la creación de objetos, permitiendo instanciar y crear las clases que le son necesarias para cumplir sus funcionalidades. En la aplicación web es aplicado por las clases controladoras, como por ejemplo *GeneroController*, en esta, se crean los objetos que representan la entidad *ArchivoMultimediaGenero.php*, lo cual evidencia que las clases controladoras son las encargadas de instanciar dichas entidades.
- ✓ **Patrón Controlador:** El propósito fundamental de este patrón es asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Este patrón se pone de manifiesto en todo el sistema ya que cada uno de los eventos generados por el usuario catalogador, es redirigido a las clases controladoras para que realicen las operaciones solicitadas, manteniendo siempre la alta cohesión.
- ✓ **Patrón Alta cohesión:** Tiene como objetivo asignar una responsabilidad de modo que la cohesión siga siendo alta. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme (Schmuller, 2003). Se utilizó Alta cohesión para asignar responsabilidades a clases que estén altamente relacionadas con la catalogación garantizando la contribución entre estas para realizar tareas de elevada complejidad en vez de utilizar una sola clase para ello. Por ejemplo la generación de la ficha es responsabilidad del *CatalogarController* pero quien tiene los datos para generar la ficha de catalogación es *ArchivoMultimedia.php*. De esta manera se evidencia la relación que debe existir entre ambas clases



ya que la primera solo se encarga de generar la ficha, utilizando para ello, los datos que devuelven los métodos de *ArchivoMultimedia.php*.

- ✓ **Patrón Bajo acoplamiento:** Su propósito fundamental es asignar una responsabilidad para mantener bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. Permite la reutilización aumentando la productividad, el diseño de las clases más independientes y reduce el impacto de los cambios (Schmuller, 2003). Este patrón es utilizado por el framework Symfony, y por ende en el sistema, al no asociar las clases del modelo con las de la vista o el controlador, la dependencia entre las clases, en este caso, se mantiene baja.

3.2.2 Patrones GoF

GoF (Grupo de Cuatro) es el acrónimo de *Gang of Four* estos patrones representan soluciones técnicas basadas en la programación orientada a objetos, permitiendo la reutilización de código. Los patrones GoF se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (Mestras Pavón , 2004). Para definir dichos patrones, se tuvo en cuenta el framework utilizado (Symfony 2.1.2), el cual está basado en el patrón arquitectónico Modelo-Vista-Controlador e incluye diferentes patrones de diseño que facilitan el trabajo a los desarrolladores.

Patrón estructural envoltorio (*decorator*): Añade funcionalidad a una clase, dinámicamente. El archivo *layout.php*, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación web, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el *layout*, o si se mira desde el otro punto de vista, el *layout* decora el contenido de la plantilla.

Patrón estructural fachada (*facade*): Su objetivo fundamental es la creación de una interfaz unificada de alto nivel que represente un subsistema y, a su vez, facilite el uso del mismo. Su principal ventaja parte de la separación de los componentes del sistema y el cliente, pues de esta forma se reduce el número de objetos con los que este último trata, facilitando así la utilización y acceso a las funcionalidades proporcionadas por la aplicación (Larman, 2003). La aplicación web cuenta con una interfaz principal, desde donde se puede acceder a todas las funcionalidades que ofrece el sistema.



3.2.3 Patrones J2EE

Otros de los patrones que se evidencia en la solución propuesta es el **Patrón Controlador Frontal (*Front Controller*)** perteneciente al conjunto de patrones J2EE (*Java 2 Enterprise Edition*), dicho patrón permite determinar la vista apropiada para manejar la solicitud del usuario (Deepak, 2003). En el sistema web este patrón se evidencia en el archivo *app_dev.php*, el cual se encarga de enrutar todas las peticiones realizadas por el usuario hacia las clases controladoras.

3.3 Diagrama de clases del diseño

Los diagramas de clases del diseño constituyen un elemento esencial en la concepción de la aplicación que se propone, ya que sirven de guía a los desarrolladores para la construcción de una aproximación del sistema que se desea implementar (Frómata Lescay, 2011). Además describen gráficamente las especificaciones de las clases de software y las interfaces en una aplicación, permitiendo formalizar el conocimiento del dominio del sistema. Muestra elementos tales como: clases, interfaces, con sus operaciones y constantes, información sobre los tipos de atributos, navegabilidad y dependencia. Se utilizan para modelar la vista de diseño estática de un sistema (Schmuller, 2003). A continuación se muestra el diagrama de clases para el CU Gestionar autor de material audiovisual. Los restantes diagramas de clases se pueden encontrar en el **Anexo 3** de la presente investigación.

El diagrama de clases presentado muestra la estructura del patrón arquitectónico utilizado en la solución web propuesta (MVC), donde la vista contiene todas las interfaces y los formularios con los que interactúa el usuario para insertar, editar y mostrar los autores de los materiales audiovisuales, (*Index.html.twig*, *new_autor.html.twig*, *index_autor.html.twig*, *edit_autor*, *Form_new*, *Form_delete*). El controlador contiene la clase controladora *AutorController* y un controlador frontal *SP_ControladorFrontal*, el cual se encarga de enrutar las peticiones del usuario hacia la clase controladora, esta a su vez hace uso de la entidad *Autor* que se encuentra representada en el modelo.

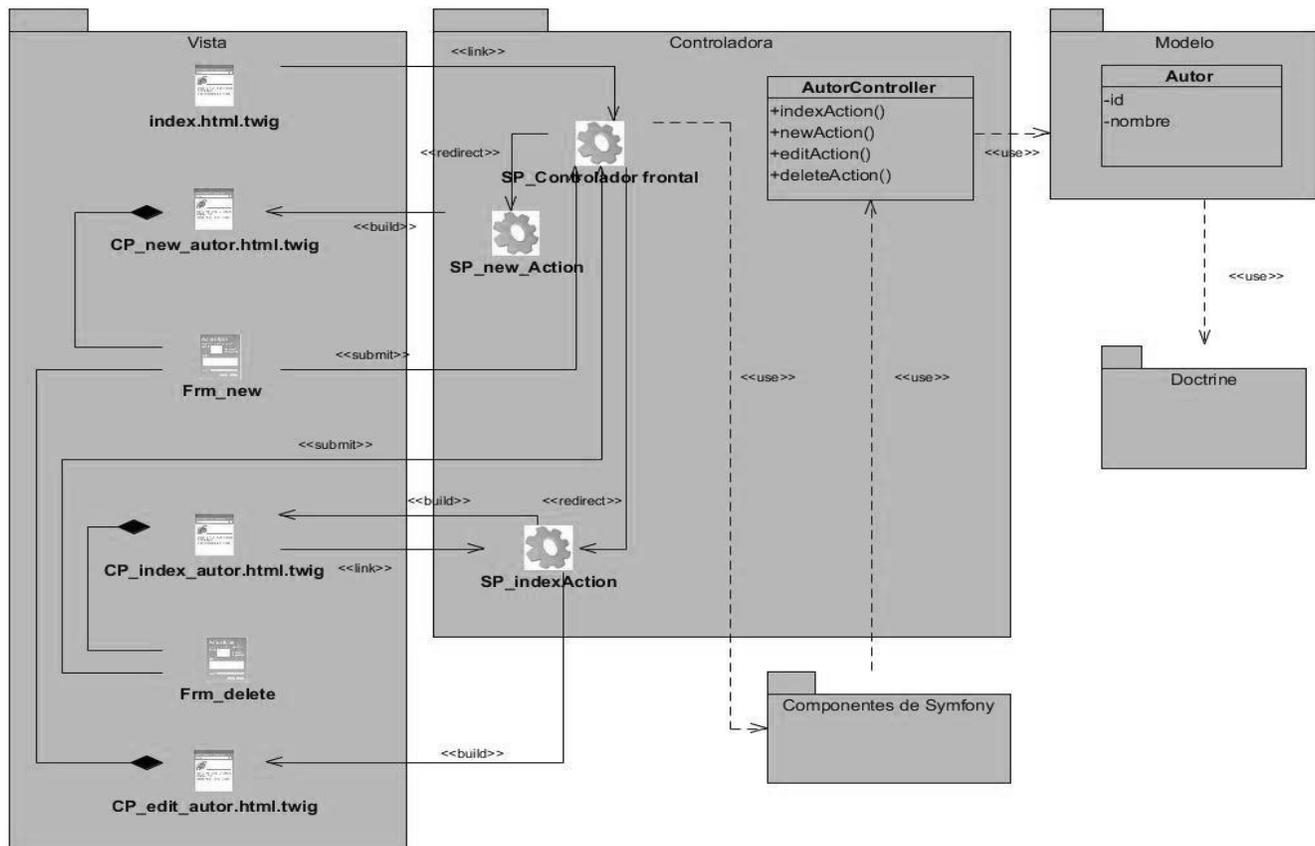


Imagen 4: Diagrama de Clases del Diseño del Caso de uso Gestionar autor de material audiovisual del sistema de catalogación.

La siguiente imagen representa el diagrama de clases del diseño del sistema de captura, el mismo está representado por dos paquetes representando las capas: Acceso a datos y Negocio, donde se agrupan las clases implementadas en el sistema. En el **Anexo 3** se encuentra el diagrama íntegro.



IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE CAPTURA Y CATALOGACIÓN

Introducción al capítulo

En el presente capítulo se expondrán los elementos fundamentales para la implementación del módulo de captura y catalogación, los diagramas de componentes y de despliegue como punto de partida para el flujo de trabajo de implementación. También se describen las pruebas a realizar, definiéndose el tipo de pruebas, técnicas y métodos a utilizar con el objetivo de validar el sistema desarrollado y detectar posibles fallos.

4.1 Modelo de implementación

El modelo de implementación está comprendido por un conjunto de componentes y subsistemas que representan la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. Este artefacto describe cómo se implementan los componentes, congregándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre éstos (Letelier, 2014).

4.1.1 Implementación

En la implementación, los elementos del diseño, fundamentalmente las clases, se implementan en términos de componentes, como ficheros de código fuente, librerías, entre otros. También se describe cómo dependen los componentes unos de otros. El flujo de trabajo Implementación enmarca el inicio de la fase de construcción del software.

4.1.2 Diagrama de componentes

Los diagramas de componentes permiten describir los elementos físicos del sistema y sus relaciones. Modelan la vista estática de la aplicación permitiendo mostrar dependencias lógicas entre un conjunto de componentes de software (Larman, 2003).



A continuación se muestra el diagrama de componentes, donde se integra el sistema de catalogación:

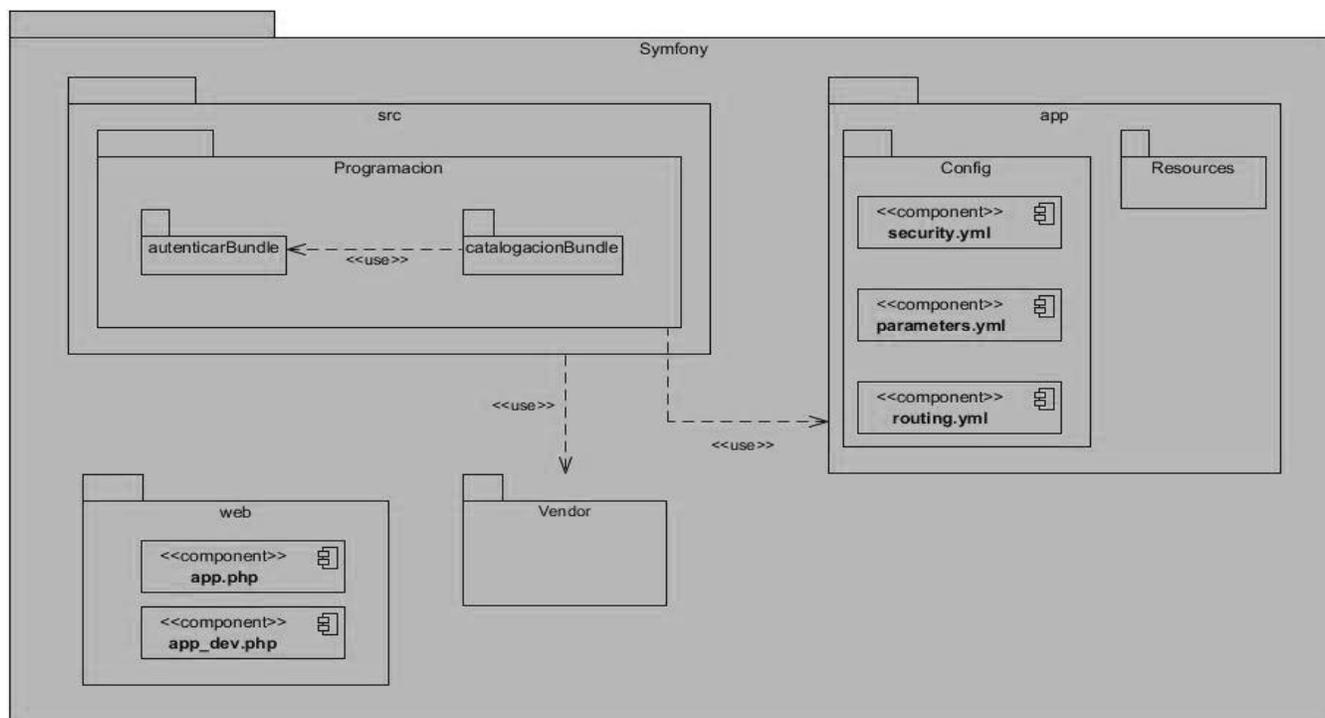


Imagen 6: Diagrama de componentes catalogación.

El sistema desarrollado web se encuentra empaquetado en dos *bundles*¹¹, donde se encuentran distribuidos tres paquetes: modelo, vista y controlador, de tal forma que exista correspondencia con el patrón arquitectónico MVC.

En el paquete controlador se encuentran las clases controladoras, encargadas de manejar las peticiones de los usuarios a través de los métodos que tienen implementados. El paquete modelo agrupa las clases de la lógica y las entidades del sistema, a través de las cuales se realiza el acceso a la base de datos. El paquete vista, agrupa los archivos que permiten visualizar las respuestas que devuelven los controladores al usuario.

A continuación se muestra el diagrama de componentes para el `catalogacionBundle`, donde se encuentran los principales componentes del sistema a desarrollar.

¹¹ Bundle: Es un directorio que contiene todo tipo de archivos dentro una estructura jerarquizada de directorios. Suelen contener clases PHP y archivos web (JavaScript, CSS e imágenes). No existe ninguna restricción sobre lo que se puede incluir dentro de un bundle.

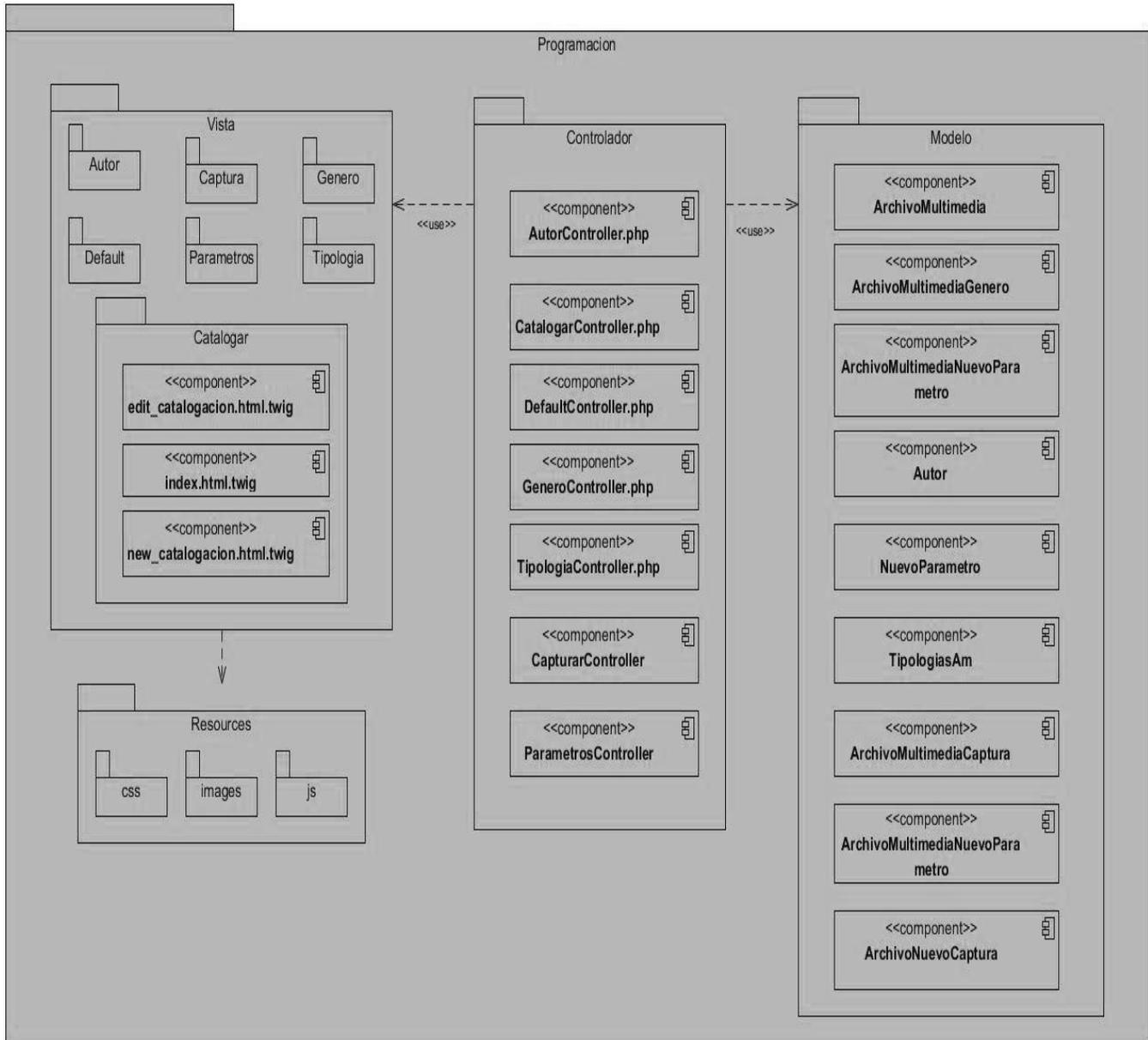


Imagen 7: Diagrama de componentes del catalogacionBundle.

A continuación se muestra el diagrama de componentes del sistema de captura propuesto:

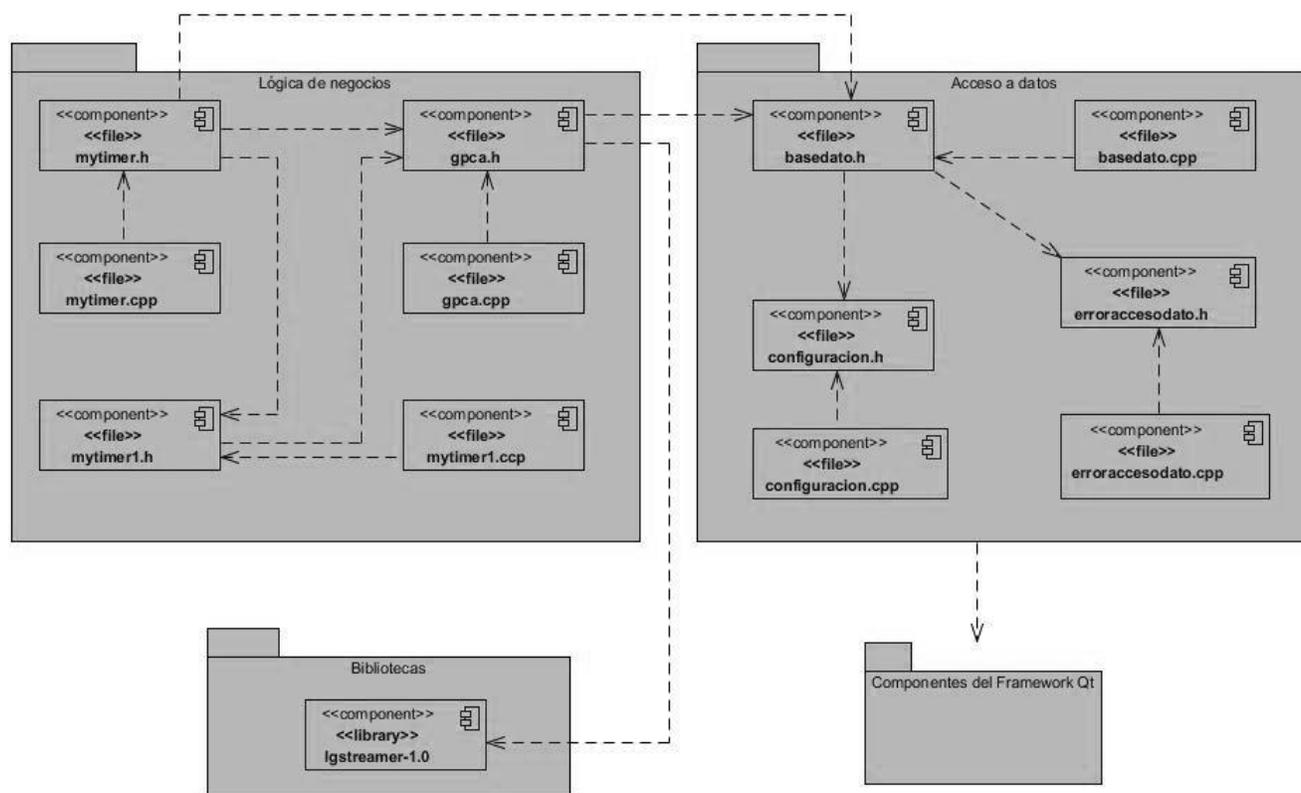


Imagen 8: Diagrama de componentes captura.

4.1.3 Estándar de codificación

Un aspecto que es muy importante a considerar en la fase de implementación es la definición de un estándar de codificación de los programas, establecer estándares para la programación de un sistema es tan significativo como cualquiera de los diagramas vistos anteriormente, más aún si su implementación involucra a un grupo de desarrolladores. Esto asegura que el código sea de alta calidad y sea fácil de mantener.

A continuación se muestran algunos de estos estándares para el lenguaje PHP utilizados en Symfony2.

CamelCase: Es una forma de notación de texto que sigue el patrón de palabras en minúscula sin espacios y con la primera letra de cada palabra en mayúsculas exceptuando la primera palabra. Este estándar fue utilizado para la definición de clases y métodos en el código de la aplicación.



Existen dos tipos de CamelCase:

- UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula. Ejemplo: *EjemploDeUpperCamelCase*.
- lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula. Ejemplo: *ejemploDeLowerCamelCase* (Auquilla Lopez, 2014).

En la aplicación se utilizó el UpperCamelCase para nombrar las clases existentes y el *ejemploDeLowerCamelCase* para la nomenclatura de los métodos.

En la estructura se definieron los siguientes estándares:

- Las líneas podrían tener 80 caracteres o menos.
- Las llaves de apertura deben ir en la siguiente línea y la llave de cierre debe ir en la siguiente línea después del cuerpo.
- Las llaves de apertura en las estructuras de control debe ir en la misma línea y las llaves de cierre deben de ir después del cuerpo.
- Los paréntesis en las estructuras de control no deben usar espacios antes o después.
- Añadir un solo espacio después de cada limitador de coma.
- Añadir un solo espacio alrededor de los operadores (==, &&, ...).
- Usa llaves para indicar el control de la estructura sin tener en cuenta el número de declaraciones que el grupo pueda contener.
- Define una clase por fichero.
- Declara las propiedades de clase antes que los propios métodos de clase (Garay, 2012).

4.1.4 Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación (Huallpara, y otros, 2014). A continuación se muestra el diagrama de despliegue propuesto para el módulo de captura y catalogación de materiales audiovisuales, el cual cuenta con cinco nodos que responden a la distribución física del sistema.

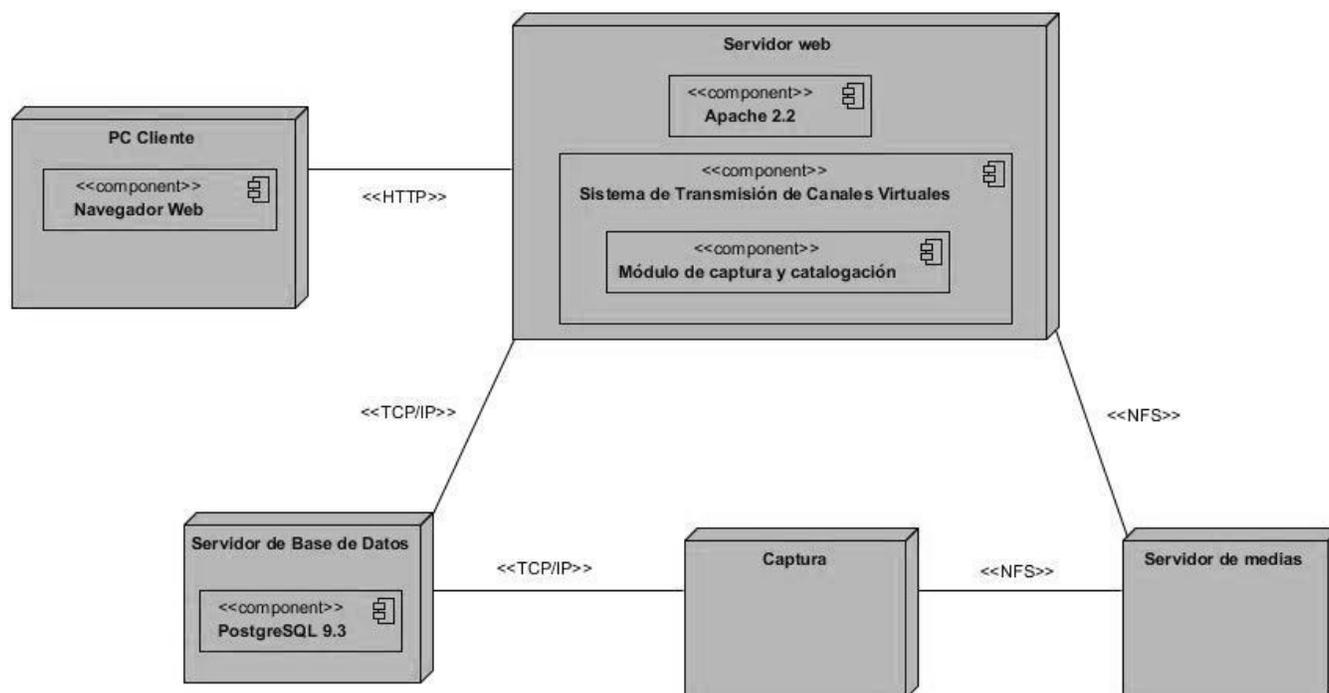


Imagen 9: Diagrama de despliegue.

4.1.5 Descripción del diagrama de despliegue

El nodo PC cliente representa los ordenadores que permitirán la interacción del usuario con la aplicación y que establece una comunicación mediante el protocolo HTTP con los nodos donde se encuentra el Servidor web. Desde este último se tiene acceso al Servidor de medias para el almacenamiento de los materiales audiovisuales, mediante el protocolo NFS y al Servidor de base de datos a través del protocolo TCP/IP. El Servidor de base de datos es donde se almacena toda la información gestionada por el sistema. Mantendrá una comunicación con el nodo Captura, sistema de captura de flujos de videos, a través del protocolo TCP/IP. El nodo Captura permitirá una comunicación directa a través del protocolo de comunicación NFS, para el almacenamiento de los flujos de videos capturados, con el Servidor de medias.

4.2 Pruebas

La importancia de los costos asociados a los errores, promueve la definición y aplicación de un proceso de pruebas minuciosas y bien planificadas. Las pruebas permiten validar y verificar el software, entendiendo como validación del software el proceso externo al equipo de desarrollo, que determina si el software



satisface los requisitos, y verificación como el proceso interno que determina si los productos de una fase satisfacen las condiciones de dicha fase (Pressman , 2002).

4.2.1 Pruebas de Rendimiento

Las pruebas de rendimiento tienen que diseñarse para asegurar que el sistema pueda procesar su carga esperada. Esto normalmente implica planificar una serie de pruebas en las que la carga se va incrementando regularmente hasta que el rendimiento del sistema se hace inaceptable (Sommerville, 2005). Para medir el rendimiento en la aplicación web se realizan diferentes tipos de pruebas, donde a continuación se mencionan las utilizadas:

❖ Pruebas de carga

Este es el tipo más sencillo de pruebas de rendimiento. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperadas. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación.

❖ Prueba de estrés

Esta prueba se utiliza normalmente para hacer colapsar la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores a determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

Resultados de las pruebas de Rendimiento para el sistema de catalogación

Se ejecutaron una serie de funcionalidades que brinda el sistema con el objetivo de evaluar el tiempo de respuesta del proceso de catalogación. La ejecución de las pruebas se realizó sobre el entorno de trabajo que presenta como características: Microprocesador Intel Core 2 Duo E4500 a 2,20 Ghz con 2 GB de RAM. Se utilizó la herramienta de software libre, JMeter, que ofrece la posibilidad de realizar pruebas de carga sobre diferentes aspectos de una aplicación web, inicialmente diseñada para pruebas de estrés en las mismas. Utilizar JMeter supone una mayor efectividad en el proceso y en la fiabilidad de los resultados. Dispone de varios componentes que facilitan la elaboración de los escenarios de prueba con la



ventaja de simular para cada uno de esos escenarios miles de usuarios. De esta manera se verifica el rendimiento del sistema mediante las pruebas de Carga y Estrés. Las pruebas arrojaron como resultados que para la cantidad promedio esperada de usuarios conectados concurrentemente (entre 30 y 50) se obtuvieron tiempos de respuesta entre 1 y 3 segundos, valorados como aceptables, mientras que más allá del límite establecido (más de 100 peticiones concurrentes) la aplicación se mantuvo sin fallos.

Tabla 3. Resultados de las pruebas de rendimiento del sistema de catalogación.

Usuarios	Error %	Rendimiento (tiempo de respuesta)
30	0,00	2,5 /seg
50	0,00	2,9 /seg
100	0,00	5,2 /seg

Resultados de las pruebas de Rendimiento para el sistema de captura

Para la observación de la carga que pudiera soportar el sistema en la realización del proceso de captura de flujos de videos, se desarrollaron las pruebas de rendimiento en dos escenarios teniendo en cuenta un número de fichas de catalogación de flujos de videos con planificaciones en un mismo instante de tiempo y desde diferentes fuentes de obtención del material audiovisual.

Para un **primer escenario** se realizó el proceso teniendo en cuenta 6 fichas de captura de flujos de videos planificadas en un mismo instante de tiempo y desde las siguientes fuentes de obtención: Se planificaron 2 capturas desde un canal externo al STCV y 4 de una pc desde donde se transmitía un video a través del reproductor VLC. Sin mostrar ningún error en la realización del proceso.

En el **segundo escenario** se realizó el proceso teniendo en cuenta 8 fichas de captura de flujos de videos planificadas en un mismo instante de tiempo y desde las siguientes fuentes de obtención: Se planificaron 3 capturas para una cámara IP, 2 desde un canal externo al STCV y 3 de una pc desde donde se transmitía un video a través del reproductor VLC. Mostrando al igual que en el escenario anterior resultados satisfactorios.



4.2.2 Métodos de Pruebas

Existen métodos de prueba independientemente de la técnica que se utilice o el nivel en que se enmarquen estas técnicas. Estos métodos proporcionan distintos criterios para generar casos de prueba que provoquen fallos en los programas, agrupándose en:

- Método de Caja Blanca o Estructural: Se basa en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.
- Método de Caja Negra o Funcional: Se realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar (Moreno , y otros, 2006).

Pruebas de Caja Negra

Estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. Se centran principalmente en los requisitos funcionales del software (Moreno , y otros, 2006). Para seleccionar el conjunto de entradas y salidas sobre las cuales trabajar, hay que tener en cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en el sistema, y como consecuencia producen una serie de salidas que revelan la presencia de defectos. El objetivo final es encontrar una serie de datos de entrada cuya probabilidad de pertenecer al conjunto de entradas que causan dicho comportamiento erróneo sea lo más alto posible (Moreno , y otros, 2006).

En la aplicación desarrollada se realizaron las pruebas de caja negra utilizando la técnica de partición equivalente, la cual divide el campo de entrada en clases de datos a partir de las cuales pueden derivarse casos de prueba. La técnica de partición equivalente consiste en definir un caso de prueba que descubra varias clases de errores, reduciendo así el número de casos de prueba que deben desarrollarse. Se definen un conjunto de estados válidos y no válidos para las condiciones de entrada y se verifica que los datos de salida sean los esperados (Pressman, 2010).

A continuación se muestra el caso de prueba perteneciente al CU Gestionar género de material audiovisual. El resto de los casos de prueba pueden ser consultados en el **Anexo 4**.

Caso de Prueba # 1 “CU Gestionar género de material audiovisual”.



Tabla 4. Caso de prueba para el Caso de Uso Gestionar género: Sección “Adicionar género”.

Sección “Adicionar género”				
Escenario	Descripción	Nombre	Flujo central	Respuesta del sistema
EC 1.1 Insertar datos del género correctamente	Se adiciona correctamente el género.	V	1. Se selecciona la opción “Adicionar”, en la página principal del sistema. 2. Se llena el campo correctamente. 3. Se selecciona la opción “Guardar”. 4. Se muestra un mensaje de éxito.	Se inserta un nuevo género en el sistema.
		Cualquier combinación de letras. Ejemplo: Comedia.		
EC 1.2 Dejar campos vacíos	Se intenta insertar un género dejando el campo vacío.		1. Se selecciona la opción “Adicionar”, en la página principal del sistema. 2. Se llena el campo correctamente. 3. Se selecciona la opción “Guardar”. 4. Se muestra un mensaje de error.	Muestra el mensaje de error: “Por favor, llene este campo”.

Tabla 5. Caso de prueba para el Caso de Uso Gestionar género: Sección “Editar género”.

Sección “Editar género”				
Escenario	Descripción	Nombre	Flujo central	Respuesta del sistema
EC 2.1 Insertar datos del género correctamente	Se guardan correctamente los cambios realizados.	V	1. Se selecciona la opción “Editar” en la página Listado de géneros.	Se guardan correctamente los cambios realizados. Muestra el mensaje de éxito: “Género editado”



		Cualquier combinación de letras. Ejemplo: Comedia.	2. Se edita el campo nombre. 3. Se selecciona la opción "Guardar". 4. Se muestra un mensaje de éxito.	satisfactoriamente".
EC 2.2 Dejar campos vacíos	Se intenta insertar un género dejando el campo vacío.		1. Se selecciona la opción "Editar" en la página Listado de géneros. 2. Se edita el campo nombre. 3. Se selecciona la opción "Guardar". 4. Se muestra un mensaje de error.	Muestra el mensaje de error: "Por favor, llene este campo".

Tabla 6. Caso de prueba para el Caso de Uso Gestionar género: Sección "Eliminar género".

Sección "Eliminar género"			
Escenario	Descripción	Flujo central	Respuesta del sistema
EC 3.1 Eliminar género correctamente	Se elimina un género correctamente.	1. Se selecciona la opción "Eliminar" en la página Listado de géneros. 2. Muestra el mensaje de confirmación: "Realmente desea eliminar el género". 3. Se selecciona la opción "OK".	Elimina el género del sistema y de la base de datos. Muestra el mensaje de éxito: "Género eliminado satisfactoriamente".
EC 3.2 No eliminar un género correctamente	No se elimina un género correctamente.	1. Se selecciona la opción "Eliminar" en la página Listado de géneros. 2. Muestra el mensaje de confirmación: "Realmente desea eliminar el género". 3. Se selecciona la opción "OK".	Muestra el mensaje de éxito: "No se puede eliminar el género".



Tabla 7. Caso de prueba para el Caso de Uso Gestionar género: Sección “Mostrar géneros”.

Sección “Mostrar género”				
Escenario	Descripción	Nombre	Flujo central	Respuesta del sistema
EC 3.1 Listar géneros correctamente	Se muestran los géneros correctamente.	V	1. Se selecciona la opción “Mostrar”, en la página principal del sistema.	Se muestra una lista de géneros almacenados en el sistema.
		Muestra el listado de todos los géneros.		

Ejecución y resultados de las pruebas de caja negra

Durante la fase de prueba se realizaron 2 iteraciones, las mismas arrojaron los siguientes resultados:

Primera iteración: Se encontraron 8 No conformidades de las cuales se identificaron 6 de redacción, 1 de interfaz y 1 de idioma.

Segunda iteración: Corregidas las No Conformidades pertenecientes a la primera iteración, se ejecutaron nuevamente los diseños elaborados encontrando 2 No conformidades de las cuales 2 eran de Redacción. Para ver una mejor descripción de las No conformidades puede ser consultado el **Anexo 5**.

Las No conformidades fueron corregidas arrojando resultados satisfactorios.

Conclusiones parciales

En este capítulo se describe la estructura y el funcionamiento del módulo propuesto, con el objetivo de lograr una mejor implementación y obtener una aplicación que cumpla con las funcionalidades establecidas, alcanzándose los siguientes resultados:

- ✓ La estructura de las clases del diseño en paquetes de componentes permitió mostrar la organización y las dependencias entre los componentes que conforman el sistema.



- ✓ Con la realización del diagrama de despliegue se logró la representación, a través de cinco nodos, de la distribución física del módulo.
- ✓ Con las pruebas realizadas al módulo desarrollado, se obtuvieron un total de 8 casos de prueba. En la primera iteración se identificaron 6 no conformidades. En la segunda iteración se hallaron 2 no conformidades. En la tercera iteración no se encontraron no conformidades, por lo que se puede concluir que la aplicación desarrollada cumple con la calidad requerida en cuanto a las entradas y salida de los datos gestionados. Para medir el rendimiento del sistema se realizaron pruebas de Carga y Estrés mediante la herramienta JMeter, lo que propició observar el comportamiento del módulo bajo una cantidad de peticiones esperadas.



CONCLUSIONES GENERALES

El presente trabajo de diploma da cumplimiento a las tareas de investigación propuestas inicialmente, logrando brindar mayor extensibilidad y acceso a las funcionalidades del módulo de captura y catalogación para el Sistema de Trasmisión de Canales Virtuales, permitiendo arrojar los siguientes resultados:

- ✓ Con la caracterización de los procesos de captura y catalogación, se logró una descripción amplia del objeto de estudio, garantizando un mejor entendimiento sobre el funcionamiento de estos dos procesos.
- ✓ La metodología utilizada, permitió guiar todo el proceso de desarrollo del software, generando los principales artefactos que permitieron obtener una arquitectura sólida para el sistema.
- ✓ Las pruebas realizadas facilitaron la validación de la especificación de los requisitos funcionales aumentando la calidad del módulo obtenido.
- ✓ El módulo desarrollado permite ampliar la descripción de los materiales audiovisuales mediante la gestión de parámetros, además ofrece la posibilidad de obtener flujos de videos de canales externos al sistema, garantizando una correcta gestión de la información de los materiales audiovisuales a utilizar en el STCV.



RECOMENDACIONES

Los resultados obtenidos, luego del desarrollo de la presente investigación satisfacen los requisitos definidos, no obstante se recomienda:

- Continuar el desarrollo de la investigación con el objetivo de perfeccionar y aumentar las funcionalidades de la aplicación como por ejemplo:
 - Realizar el proceso de captura de forma distribuida, permitiendo dividir la carga de trabajo en varias estaciones para agilizar el proceso de captura de flujos de video.
 - Incorporar un módulo que realice un control sobre el espacio disponible en el servidor donde se almacenan los materiales audiovisuales.



REFERENCIAS BIBLIOGRÁFICAS

Eguiluz , Javier. 2009. Introducción a Javascript . [En línea] 2009.

Visual Paradigm. 2014. BPMN and Database Tool for Software Development . [En línea] 2014.
<http://www.visual-paradigm.com>.

2014. Alegsa. *ALEGSA.com.ar* . [En línea] 2014. www.alegsa.com.ar/Dic/metadato.php.

Alegsa. 2014. Definiciones-de.com. *DEFINICIONES-DE.com*. [En línea] Diccionario de ALEGSA - Santa Fe, Argentina, 2014. <http://www.definiciones-de.com/Definicion/de/catalogar.php>.

ALFONSO, ALEX. 2010. Servicios Web. [En línea] 2010. <http://www.diigo.net/profile/sesser>.

Álvarez, Miguel Angel. 2001. DesasarrolloWeb. [En línea] 2001. DesasarrolloWeb.com.

ANALISIS Y DISEÑO DE SISTEMAS II. [En línea]

APACHE. 2012. APACHE. APACHE. Versión 2.2 del servidor HTTP Apache . [En línea] 2012.
<http://httpd.apache.org/docs/2.2/>.

Aquilla Lopez, Kristina . 2014. ESTANDAR DE CODIFICACIÓN. [En línea] 2014.
<http://es.scribd.com/doc/246920425/Estandar-de-Codificacion#scribd>.

C++. 2013. LENGUAJE DE PROGRAMACIÓN C++. [En línea] 26 de marzo de 2013.

Carlos Leyva. 2014. SERVIDOR WEB. *Brending Designers*. [En línea] 2014.
<http://www.brendingdesigners.com.ar/articulos/servidorweb.html>.

Catalis. 2014. Catalis. Catalis Una herramienta web para crear catálogos basados en MARC 21 y AACR2 . [En línea] 2014. <http://inmabb.criba.edu.ar>.

Clauso García, Adelina. 1993. Análisis documental: el análisis formal. Madrid : s.n., 1993.

ClubEnsayos. 2013. [En línea] 2013. <http://www.clubensayos.com/Temas-Variados/Catalogacion/1022340.html>.

de la Torre , Aníbal. 2006. Lenguajes del lado servidor o cliente. 2006.

Deepak, Alur. 2003. *CORE J2EE PATTERNS: BEST PRACTICES AND DESIGN STRATEGIES*. s.l. : PRENTICE-HALL INTERNATIONAL EDITION, 2003. 9780131422469.



Definición de CSS. 2014. Definición de CSS. Definición de CSS - ¿Qué son las hojas de estilo o cascading style sheets? [en línea]. hojas de estilo o cascading style sheets.htm. [En línea] 2014. hojas de estilo o cascading style sheets.htm.

Definición. 2014 . definicion.org. . [En línea] 2014 . www.definicion.org/lenguaje-de-programacion.

Definicion.de . 2015. Definición de audiovisual - Qué es, Significado y Concepto. *Definicion.de* . [En línea] 2015. <http://definicion.de/audiovisual/>.

Del Castillo, Alvaro. 2006. Webs dinámicas con PHP. Webs dinámicas con PHP. [En línea] 2006.

Documentalistas. 2014. Documentalistas. DOKUMENTALISTAS . [En línea] 2014. www.dokumentalistas.com.

Edmonson, Ray. 1998. UNA FILOSOFIA DE LOS ARCHIVOS AUDIOVISUALES. París : s.n., 1998.

Eguíluz Pérez , Javier. 2005. Introducción a AJAX. [En línea] 2005.

Eguíluz, Javier. 2009. Introducción a CSS. España : s.n., 2009.

Frómata Lescay, Roberto. 2011. Subsistema de Control de Ingesta del Sistema. La Habana : s.n., 2011.

Gallo Hancock, Michael William. 2002. Networking Explained. 2002.

Garay, Joseba . 2012. Convenciones en código Symfony 2. [En línea] 2012. <http://www.nireleku.com/2012/10/convenciones-en-codigo-symfony-2/>.

Garlan, David y Shaw, Mary. 1994. "An introduction to software architecture". s.l. : CMU Software Engineering Institute Technical Report, 1994.

González, Guillermo y Saco, Vallés. 2014. Master online en LAMP (Linux - Apache - MySQL - PHP) – Masters en LAMP (Linux - Apache - MySQL - PHP) – On-line Ciberaula. [En línea] 2014. http://www.ciberaula.com/curso/lamp/que_es/.

Grupo VITELSA. 2014. Grupo VITELSA. *Vitelsa-Integración de soluciones de software* . [En línea] 2014. <http://www.vitelsa.es>.

Gutiérrez, Javier J. 2015. ¿Qué es un framework web? [En línea] 2015.

Hernández León, Rolando Alfredo. 2002. *EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA*. Ciudad de la Habana : s.n., 2002.

HTML 5. 2014. Aprender internet. Qué es HTML 5 – Conoce qué es HTML 5 y cuáles son sus características principales. [En línea] 2014. <http://aprenderinternet.about.com/od/Glosario/g/Que-Es-Html-5.htm>.



Hualpara, Marca y Quisbert, Hugo Michael. 2014. ANALISIS Y DISEÑO DE SISTEMAS II TRABAJO DE INVESTIGACIÓN Y EXPOSICIÓN “Diagrama de Despliegue”. [En línea] 2014.

ITLA. 2012. Biblioteca. Instituto Tecnológico de las Américas - ITLA. [En línea] 2012. http://www.itla.edu.do/biblioteca/secundaria.php?act=sistema_organizacion.

Jacobson, I. , Booch, G. y Rumbaugh., J. 2000. El proceso unificado de desarrollo de. 2000.

Joseph Schmuller . 2015. Aprendiendo UML En 24 Horas, Intercambios Virtuales. [En línea] 13 de marzo de 2015. [/www.intercambiosvirtuales.org/libros-manuales/aprendiendo-uml-en-24-horas-joseph](http://www.intercambiosvirtuales.org/libros-manuales/aprendiendo-uml-en-24-horas-joseph).

JQuery. 2015. DesarrolloWeb.com.Manuales.Manuales de jQuery. [En línea] 2015. <http://www.desarrolloweb.com/manuales/manual-jquery.html>.

—. 2015. JQuery. [En línea] 2015. <http://jquery.com/>.

Laguna, Miguel A. 2013. Requisitos. Ingeniería de Software 1. [En línea] 2013. [//www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf](http://www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf).

Larman, Craig. 2003. Leyendo UML y Patrones. *Leyendo UML y Patrones 2da Edición*. 2003.

—. 2003. Prentice. UML y Patrones. 2003.

Lau, Ricardo. 2013. Vigo:Unión Webmaster Libres. *Ide (Entornos de Desarrollo Integrado)*. [En línea] 2013. <https://sites.google.com/site/vigomiciudad/otras-cosa-interesantes/ide-entornos-de-desarrollo-integrado>.

Lecca Risco. 1998. Qué es JAVA Script ? [En línea] 1998. <http://www.pablin.com.ar/computer/cursos/cursojs/js1.htm>.

Letelier, P. 2014. Rational Unified Process (RUP). [En línea] 2014.

LibrosWeb. 2015 . LIBROSWEB. [En línea] 2015 . http://librosweb.es/libro/symfony_1_4/capitulo_1/symfony_en_pocas_palabras.html.

Marcos, Mary Carmen. 2004. Tarsys, un software para la gestión de documentos audiovisuales. El profesional de la información. [En línea] February de 2004. <http://www.elprofesionaldelainformacion.com>.

Martínez Tomeu, Virginia. 2010. El documentalista audiovisual del Centro de Documentación de Barcelona Televisión. Barcelona: Facultad de Biblioteconomía y Documentación. Barcelona : Facultad de Biblioteconomía y Documentación, 2010.

MCNarte. 2015. MCNarte.Tu wed de arte. [En línea] 2015. <http://www.mcnarte.com/app-arte/do/show?key=catalogacion>.

Medina, Cervantes. 2006. Los medios de comunicación y la organización del tiempo libre de los jóvenes en Guadalajara. Revista de la Universidad de Valle de Atemajac. 2006.



- Mestras Pavón , Juan . 2004.** Patrones de diseño orientado a objetosDep. Ingeniería del Software e Inteligencia Artificial. Madrid : s.n., 2004.
- Metadatos. 2014.** Metadatos. *INTRODUCCIÓN A LOS METADATOS: ESTÁNDARES Y APLICACIÓN.* [En línea] 2014. /www.sedic.es/autoformacion/metadatos/tema1.htm.
- 2013.** MODELO DE IMPLEMENTACION . [En línea] 2013. <http://mayraunidad5.blogspot.com/>.
- Moreno , Juristo Natalia y Vegas Sira. , Ana Maria. 2006.** *Técnicas de evaluación de software.* 2006.
- MVC. 2015.** Qué es MVC. . [En línea] 2015. <http://www.desarrolloweb.com/articulos/que-es-mvc.html>.
- NetBeans. 2015.** NetBeans. *www.netbeans.org.* [En línea] 2015. https://netbeans.org/index_es.html.
- NEYEM, ANDRÉS. 2013.** INSTRUCTIVO PARA EL MODELADO CON CASOS DE USO. 2013.
- Pinto, Joseph . 2015 .** Joseph Pinto Portafolio de Programación. [En línea] 2015 . <http://josephapl1989.esy.es/PortafolioProgramacion2c2/css/>.
- Potencier , Fabien . 2009.** Symfony. [En línea] 11 de June de 2009. <http://symfony.com/blog/new-in-symfony-1-3-what-s-up-with-propel-and-doctrine>.
- Pressman , Roger S. 2002.** Ingeniería del Software, un enfoque práctico. . 2002.
- PRESSMAN y PRESSMAN, Roger S. 2010.** *Ingeniería del Software, un enfoque práctico.* 2010.
- Pressman, Roger S. 2010.** "*Software Engineering*". New York : Higher Education, 2010.
- Puebla. 2009.** Centro Interactivo de Recursos de Información y Aprendizaje. *Universidad de las Américas Puebla. CIRIA.* [En línea] 2009. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/fuentes_k_jf/capitulo2.pdf.
- Quiroga. 2013.** Requerimientos funcionales y no funcionales. [En línea] 2013. <http://sistemas.uniandes.edu.co>.
- RAE. 2014.** Diccionario de la Lengua Española. Real Academia Española. [En línea] 2014. www.rae.es/obras-academicas/diccionarios/diccionario-de-la-lengua-espanola.
- Schmuller, Joseph. 2000.** APRENDIENDO UML EN 24 HORAS. s.l. : PEARSON EDUCACION, 2000.
- . **2003.** Aprendiendo UML en 24 horas. [aut. libro] Prentice Hall. *UML y Patrones.* México : Editorial Dirección Computación., 2003.
- Seapavaa. 2007.** SOUTHEAST ASIA-PACIFIC AUDIOVISUAL ARCHIVE ASSOCIATION. [En línea] 2007. <http://www.seapavaa.org>.



Slideshare. 2014. Slideshare. [En línea] 2014. <http://es.slideshare.net/hugofern/sistema-gestin-de-bases-de-datos>.

Sobre PostgreSQL. 2014. Sobre PostgreSQL. PostgreSQL-es . [En línea] 2014. http://www.postgresql.org.es/sobre_postgresql.

SOFTPEI. 2013. Inicio rápido con el Framework Qt. . *SOFTPEI | Ingeniería y Sistemas*. [En línea] 2013.

Sommerville, Ian. 2005. *Ingeniería del software*. Madrid : s.n., 2005. ISBN.

Suárez Hernández., José de Jesús. 2008. Arquitectura de software: importancia de su ciclo de vida. Arquitectura de software: importancia de su ciclo de vida. [En línea] 2008. <http://www.enterate.unam.mx/Articulos/2006/febrero/arquitect.htm>..

Tangram solutions. 2014. Tangram solutions. *Tangram solutions*. [En línea] 2014. <http://tangramsolutions.es/soluciones/ingesta-y-captura/>.

Ubuntu. 2012 . lightweight integrated development environment (IDE) for Qt. [En línea] 2012 . <https://apps.ubuntu.com/cat/applications/qtcreator/>.

XML-RPC. 2012. XML-RPC. XML-RPC. [En línea] 2012. XML-RPC.Com.

XTREAM. 2014. XTREAM. XTREAM Sistemas de información global . [En línea] 2014. <http://www.xtreamsig.com>.