Universidad de las Ciencias Informáticas

Título: Módulo web para la gestión de tipologías para la plataforma VideoWeb 2.0.

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

Autor: Mahelis Ricart Rodríguez.

Tutores: Ing.Zorilin Alonso Guerrero.

Ing.Yoandri Quintara Rondón.

La Habana, junio de 2013

"Año 55 de la Revolución"

"La gran victoria que hoy parece fácil fue el resultado de pequeñas victorias que pasaron desapercibidas".

Paulo Coelho

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 firmo la presente a los ____ días del mes de _____ del año _____.

Mahelis Ricart Rodríguez	Zorilin Alonso Guerrero
_____	_____
Firma del Autor	Firma del Tutor

DATOS DE CONTACTOS

Tutores:

- **Ing. Zorilin Alonso Guerrero** (zalonso@.uci.cu): Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2009. Profesora de la Universidad de las Ciencias Informáticas y posee categoría docente de profesor Instructor. Pertenece al Centro de Desarrollo de Geoinformática y Señales Digitales (GEYSED) y se desempeña como líder del proyecto Catalogación y Publicación de Medias (CPM) perteneciente al departamento Señales Digitales.

- **Ing. Yoandri Quintana Rondón**: Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2009. Profesor de la Universidad de las Ciencias Informáticas y posee categoría docente de profesor Instructor.

Autor:

- **Mahelis Ricart Rodríguez** (mricart@estudiantes.uci.cu): Estudiante de la Universidad de las Ciencias Informáticas y pertenece al proyecto Catalogación y Publicación de Medias del departamento Señales Digitales.

DEDICATORIA

Dedico mi tesis en especial a mi mamá por darme la vida, por sacrificarse día a día para que yo pudiera terminar mi carrera.

A mi abuela mima, la cual considero mi ídolo, por sus consejos, apoyo y amor incondicional.

A mis hermanos por quererme mucho y confiar en mí, principalmente a mi hermana Yanay.

Finalmente, me gustaría dedicarle este trabajo a un gran amigo, mi fiel consejero Yainier Martínez Rubén, el cual, durante mis cinco años de carrera estuvo ayudándome en todo lo que me hizo falta. Además, gracias a él aprendí que la vida es un costo de oportunidades.

AGRADECIEMIENTOS

Agradezco principalmente a mis tutores Yoandri y Zorilin por ser mis guías, apoyo, amigos y por ser tan pacientes conmigo.

A mi familia en forma general por apoyarme, principalmente a mis padres, mi abuelo Juán, mi tía Olguita y mis primos Yohandri y Yanet.

También a mi novio y amigo Yoendy a quien adoro, por el amor y apoyo incondicional que siempre me ha brindado y por hacerme sentir cada día una persona especial.

A mis amigos de manera general y a todas esas personas que he podido conocer durante mi etapa estudiantil. Especialmente a mis mejores amigas Jennifer, Lieny, Yoarys, Dimersa, Arianny, Leydi, Yanet, Saily y Yanela porque siempre pude contar con ellas. A mis compañeras de apartamento por convivir conmigo durante tanto tiempo y a mis compañeras del edificio 93. A mi amigos Jesús Noguera, Yeni y Maikel a quien le digo Mito.

A mis compañeros del proyecto y profesores que durante mi carrera ayudaron a mi formación, entre ellos Pabel, Yoendry, Sisley, Ángel y Frank, mi profesor de Investigación de Operaciones Tony y el de Ingeniería de Software Carlos Luis.

En general, a todas esas personas que de alguna manera contribuyeron a que yo pudiera realizar mis sueños.

RESUMEN

Como un proceso de estudio y clasificación de tipos de contenidos, la gestión de tipología es de vital importancia para instituciones en las que se maneje un gran cúmulo de multimedia con diferentes fines. A través de esta gestión se logra la estructura adecuada para el almacenamiento de los archivos multimedia, facilitando los procesos de búsqueda y su posterior utilización. La presente investigación se basó en el desarrollo de un módulo de gestión dinámica de tipología de archivos multimedia, para lograr la personalización de la plataforma VideoWeb 2.0 al entorno donde se despliega o desarrolla. El trabajo presentado consta de tres capítulos, en el primero se realiza una caracterización de las soluciones existentes en el ámbito nacional e internacional. Además, se seleccionaron las herramientas y tecnologías adecuadas para el desarrollo de la solución, destacándose entre ellas el CMS[1] Drupal 7, el Sistema Gestor de Base de Datos PostgreSQL 9.1, guiando el proceso de desarrollo la metodología Proceso de Desarrollo Unificado. En la segunda parte de la investigación, se realizó el diseño de la aplicación y se construyó la solución propuesta. Por último, en el tercer capítulo se probó la solución para verificar que esta cumpliera con los objetivos inicialmente propuestos. Con el desarrollo de este módulo se facilitará la gestión de tipologías dinámicas en la plataforma VideoWeb 2.0, de tal forma que se permitirá definir la información que se desea almacenar en dependencia de las necesidades del cliente de manera ágil y personalizada.

Palabras Claves: catalogación, gestión, metadatos, proceso, tipología, web.

[1] *Content Management Systems o* Sistemas de Gestión de Contenidos.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE

INTRODUCCIÓN

Con el apresurado auge de las Tecnologías de la Información y las Comunicaciones (TIC) en los últimos años, se ha evidenciado una creciente necesidad para las personas permanecer informados y comunicados. La información ha logrado alcanzar formas de almacenarse, compartirse, transmitirse y procesarse inimaginables hace 20 años atrás. Esta evolución se ha beneficiado en gran medida de los avances tecnológicos experimentados en todas las épocas, que han ido suprimiendo las barreras que tradicionalmente han limitado la interactividad entre las personas. El manejo de información es un proceso que exige informarse e informar. Es decir, exige construir, primero, una representación de una determinada realidad con los datos que adquirimos de ella para poder darla a conocer, disponiendo esa representación al alcance de los demás o comunicarla. (1)

El uso de nuevos tipos de señales y el desarrollo de nuevos medios de transmisión, adaptados a las crecientes necesidades de comunicación, han sido fenómenos paralelos al desarrollo de la historia. Dentro de este marco cabe mencionar la utilización de los medios audiovisuales en los diferentes sectores de la sociedad como por ejemplo la educación, entretenimiento, entre otros, trayendo consigo la generación de grandes volúmenes de dichos medios. Todo esto ha traído como consecuencia que los materiales que anteriormente se guardaban en formato duro o sólido se hayan digitalizado, y aun así la cantidad siga siendo extremadamente grande.

A la par de dichos acontecimientos se ilustran interrogantes sobre la gestión de los mismos, tal como: ¿Cómo acceder de manera precisa y eficiente a los audiovisuales que se necesita? Este tipo de interrogante conlleva a la necesidad de la organización de las medias para su correcto empleo y obtención de mejores resultados en las diferentes ramas de la sociedad, formando así parte de la citada evolución. Los medios audiovisuales pasan por un proceso en el cual se clasifican de acuerdo a diferentes datos descriptivos que se le asocian, garantizando que la recuperación de los mismos, una vez almacenados, se realice de manera sencilla y rápida. Dicho proceso es nombrado como catalogación de medias, donde el término tipología juega un papel importante, refiriéndose a la clasificación realizada o la información almacenada correspondiente a un tipo de multimedia basado en su ámbito de difusión y contenido. (2)

Para ello muchas organizaciones se rigen por *normas de catalogación*[2] las cuales establecen la estructura de los datos descriptivos. Para lograr que el proceso de recuperación y difusión sea fácil y ágil es necesario que esté respaldado por un buen proceso de gestión de audiovisuales. El punto inicial de la gestión consiste en la adecuada descripción de los contenidos que se almacenan, pues esto garantiza la recuperación posterior.

La catalogación y recuperación de audiovisuales es uno de los objetivos de los sistemas que se implementan en el departamento Señales Digitales del centro Geoinformática y Señales Digitales (GEYSED) de la Universidad de las Ciencias Informáticas (UCI). En los proyectos VideoWeb y el antiguo Captura y Catalogación de Medias (CCM), es donde más resultado se ha obtenido en esas áreas. Dichos proyectos han sido unificados con el propósito de implementar un producto que abarque diferentes oportunidades de negocio. Por la versatilidad de los entornos de aplicación de un sistema de este tipo, los datos de descripción de las medias deben ser modificables fácilmente pues no en todas las instituciones se necesitan los mismos campos de descripción.

Actualmente existen dos módulos para la gestión de tipologías. Uno desarrollado por el proyecto CCM e implementado con la herramienta Symfony 1.4 con Dojo, pero la interfaz de usuario no se llegó a terminar por diversos inconvenientes con el framework Dojo. Otro desarrollado por el proyecto VideoWeb que permite la creación de fichas de descripción personalizadas, pero la interfaz de usuario para la gestión de tipologías es poco intuitiva, no se muestra la información de manera organizada y está implementado con la herramienta Drupal 6. Ambos productos poseen una estructura de base de datos que complejiza el desarrollo, por lo que sigue siendo una problemática lograr adaptar las fichas de catalogación de medias de una manera rápida ante la solicitud de un cliente.

A esto se suma que el equipo de desarrollo ha determinado la necesidad de migrar la plataforma VideoWeb a Drupal 7, puesto que Drupal 6 pronto dejará de tener soporte por la comunidad de desarrollo, así como lo dejó de tener Symfony 1.4 al salir la nueva versión 2.x. (3) Además de que la nueva versión de Drupal incorpora elementos que facilitan el trabajo con tipos de contenido dinámico.

---

[2] Conjunto de reglas que determinan las características que rigen los procesos de catalogación de archivos multimedia y esencialmente los datos que se requieren para su correcta descripción.(4)

A partir de que en la nueva plataforma se trabaja sobre una versión más reciente de Drupal, es recomendable que todos sus módulos se desarrollen sobre la misma, pues esto facilitará el flujo de información en el sistema, así como su despliegue. La nueva versión del sistema debe ser flexible ante la demanda de diversos clientes y uno de los puntos clave para lograr esta flexibilidad lo constituye la modificación del modelo de datos en dependencia de los atributos de descripción que se requieran para la descripción de los materiales audiovisuales que se gestionan.

A partir de la situación planteada se ha identificado como **problema a resolver**: ¿Cómo lograr la adaptabilidad de las fichas de catalogación de medias para las personalizaciones de la plataforma VideoWeb 2.0?

Para dar respuesta a esta interrogante se define como **objeto de estudio**: Los procesos de gestión dinámica de tipologías.

Teniendo como **campo de acción**: Los procesos de gestión dinámica de tipologías para la plataforma VideoWeb 2.0.

Cumpliendo con lo antes expuesto se propone, como **objetivo general**: Desarrollar un módulo web para la gestión de tipologías para plataforma VideoWeb 2.0.

Queda definida como **idea a defender**: Con el desarrollo del módulo web de gestión de tipologías se garantizará la adaptabilidad de las fichas de catalogación de medias para las personalizaciones de la plataforma VideoWeb 2.0.

Para el desarrollo exitoso del presente trabajo se definieron las siguientes tareas de la investigación:

- Descripción del estado del arte relacionado con los procesos de gestión de tipologías de materiales audiovisuales.
- Caracterización de las herramientas y tecnologías seleccionadas para la implementación del módulo de gestión de tipologías para la plataforma VideoWeb 2.0.
- Realización del diseño del módulo de gestión de tipologías para la plataforma VideoWeb 2.0.
- Implementación del módulo de gestión de tipologías para la plataforma VideoWeb 2.0.
- Integración del módulo implementado a la plataforma VideoWeb 2.0.

- Realización de pruebas al módulo implementado.

Se utilizaron métodos científicos como:

- **Análisis histórico-lógico**: Se utilizó para describir y estudiar de forma analítica la trayectoria histórica real, la evolución y desarrollo del proceso de gestión de tipologías en función de la conformación de fichas de catalogación.
- **Analítico-sintético:** Se utilizó para sintetizar lo esencial de la bibliografía consultada sobre el proceso de creación de fichas de catalogación de medias en el departamento Señales Digitales, que se evidencia en el estudio de las investigaciones y aplicaciones realizadas anteriormente.
- **Inductivo–deductivo:** Se empleó con el objetivo de identificar regularidades, particularmente las relacionadas con los requerimientos teóricos y metodológicos exigidos para el proceso de gestión de tipologías para la plataforma VideoWeb.
- **Modelación:** Se hizo usó para representar, de forma estandarizada de acuerdo a los modelos establecidos por la metodología seleccionada, los casos de uso, diseño de clases y otros elementos necesarios en el desarrollo de software.

Descripción de los capítulos:

Capítulo1: Fundamentación teórica. Se brinda información de los diferentes elementos teóricos sobre el proceso para la creación de fichas para la catalogación. Se efectúa un estudio de las soluciones existentes similares al módulo a desarrollar y se analiza la selección de las herramientas y tecnologías que se tendrán en cuenta para la confección de la solución propuesta.

Capítulo 2: Construcción de la solución propuesta. Se representa el diseño de la aplicación propuesta, se muestran los diagramas de clases del diseño, el diagrama de despliegue y el diagrama de componentes. Además se describe el patrón arquitectónico a utilizar así como los patrones de diseño y los elementos referentes a la base de datos a utilizar.

Capítulo 3: Prueba de la solución propuesta. Se define el tipo de prueba a utilizar y se describen los diseños de casos de pruebas. También se exponen los resultados obtenidos en las pruebas al módulo desarrollado.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se brinda una visión general de los aspectos relacionados con la información de los diferentes elementos teóricos sobre el proceso de creación de fichas para la catalogación de medias, así como también se efectúa un estudio de las soluciones existentes similares al módulo a desarrollar. Además se analiza la selección de las herramientas y tecnologías que se tendrán en cuenta para la confección de la solución, con el propósito de garantizar la creación de un software con calidad, robusto y funcional.

1.2 Conceptos asociados al dominio del problema

1.2.1 Catalogación de archivos multimedia

Antes de analizar los conceptos más básicos de la gestión dinámica de tipologías, es preciso conocer elementos que pueden aportar la visión de en qué ámbito es que se utilizan las tipologías y por qué es provechoso que su gestión sea dinámica. Un ejemplo clave es la catalogación de audiovisuales.

Catalogar: Registrar ordenadamente libros o manuscritos en forma de catálogo. (5) Apuntar, registrar ordenadamente libros, documentos, entre otros, formando catálogos de ellos. (6) Catalogar es describir las partes esenciales de un documento para identificar su contenido con el fin de poder recuperarlo, en un momento dado, de entre una colección determinada de documentos. (7)

Catalogación: Acción y efecto de catalogar. El proceso de catalogación consiste en organizar, según diferentes criterios, algún tipo de material y hacer referencias a ellos mediante la organización de sus datos en catálogos, o sea, la relación ordenada en la que se incluyen o describen de forma individual libros, documentos, personas, objetos, entre otros, que están relacionados entre sí (6). Es el proceso de clasificar elementos pertenecientes a un mismo conjunto.

Fichas para la catalogación de medias: es un formulario para almacenar los *metadatos*[3] del material audiovisual, que no es más que el dato sobre el dato. Ejemplo, de la tipología película se almacena el director de la película, título, actores, duración, entre otros datos.

Datos: Información dispuesta de manera adecuada para su tratamiento por un ordenador (6), o sea, la información que será procesada, almacenada y distribuida al ser publicada.

Multimedia: Refiriéndose a cualquier objeto que usa de forma simultánea varios tipos de contenidos, como pueden ser: video, audio, texto, imágenes, animación y otros, en la transmisión de una información. (6)

1.2.2 Gestión de tipologías

Gestión: Acción y efecto de gestionar, hacer diligencias conducentes al logro de un negocio o de un deseo cualquiera. (6)

Tipología: Estudio y clasificación de tipos que se practica en diversas ciencias. (6) Las tipologías de producto multimedia se pueden clasificar de acuerdo a la intencionalidad de la información y de la plataforma informática, es decir, los medios de difusión, en que estarán funcionando las aplicaciones multimedia. (10) Un ejemplo podría constituir las tipologías de archivos audiovisuales que de acuerdo a su contenido pueden ser: películas, documentales, series y videos musicales.

1.3 Descripción de objeto de estudio

La clasificación ha jugado un papel vital en cuanto a la historia de los servicios y la administración de la información. La tarea de clasificar se puede definir como la actividad de agrupar los elementos de información de acuerdo a atributos o propiedades comunes entre ellos. Por tanto, definir un sistema de clasificación es elegir en base a qué atributos se van a agrupar los contenidos y cómo se han de organizar estos atributos. (11) Este proceso es llevado a cabo en sistemas donde se almacena mucha información y resulta compleja la administración de la misma.

---

[3] Datos sobre los datos, informaciones sobre datos, datos sobre informaciones, información referente a información, entre otros. (8)

La gestión dinámica de tipologías, como un proceso de estudio y clasificación de tipos de contenidos, es de vital importancia para lograr una estructura adecuada de almacenamiento de la información, principalmente en los sistemas que conservan documentación sujeta a constante consulta y utilización. Ejemplo de esto son las bibliotecas virtuales, mediatecas, televisoras, entre otras. Se han desarrollado en el mundo diversas herramientas de software que tienen el propósito de asistir, con la catalogación de audiovisuales y otro tipo de contenido, la publicación de archivos multimedia como es el caso de Tarsys y Catalis.

Tarsys es un producto mundialmente conocido que se encarga de gestionar archivos multimedia y aplica la gestión de tipologías a través de la creación de carpetas de catalogación con campos de metadatos especializados, de *tesauros*[4] y diccionarios en varios idiomas. (12) Por otra parte, Catalis es una herramienta web para crear catálogos basados en *MARC 21*[5] y *AACR2*[6], que también aplica la gestión de tipologías, donde unas de sus prestaciones es la generación de plantillas para los diferentes tipos de materiales. Las mismas son formularios dinámicos, en el cual el catalogador puede ir agregando o quitando campos y subcampos a medida que sea necesario. (13).

Es propósito general de toda organización que implementa un sistema de ordenamiento de sus archivos, el uso de estándares y normas internacionales que avalen que el trabajo realizado esté completo. En el caso de Catalis se propone el uso de AACR2. Varias organizaciones internacionales se dedican a la elaboración de estándares específicos para el ámbito de la catalogación de archivos multimedia, como por ejemplo:

---

[4] Diccionarios, catálogos. Es una herramienta para el control del vocabulario. Orienta a los indizadores y a los usuarios sobre los términos que pueden utilizar y, así ayuda a mejorar la calidad de la recuperación.(14)

[5] Formato para Registros Bibliográficos. Ha sido diseñado para servir como portador de la información bibliográfica relativa a: materiales textuales impresos y manuscritos, archivos de computador, mapas, música, recursos continuos, materiales visuales y materiales mixtos. (15)

[6] *Anglo-American Cataloguing Rules* (AACR) están diseñados para su uso en la construcción de catálogos y otras listas en las bibliotecas generales de todos los tamaños. (16)

- La Federación Internacional de Archivos de Televisión (FIAT / IFTA), se creó con el fin de impulsar la cooperación entre los más importantes organismos del sector, para estudiar los problemas sobre la conservación del material audiovisual e intercambiar información sobre nuevas tecnologías y para promover la formación profesional teórica de los documentalistas (4). Por la completitud y diversa gama de aplicación que puede tener la lista de datos mínimos ofrecida por la FIAT, el sistema que se elaborará propone su uso, pero siempre sobre la base de que no es un elemento obligatorio y que puede ser adaptable según los requisitos del cliente.
- La Asociación Internacional de Archivos Sonoros y Audiovisuales (IASA) por sus siglas en inglés *International Association of Sound and Audiovisual Archive*, fue establecida en 1969 en Amsterdam para funcionar como un medio para la cooperación internacional entre los archivos que conservan grabados, documentos sonoros y audiovisuales. Tiene entre sus estrategias de trabajo un grupo nombrado Comité de Catalogación y Documentación que se ocupa de las normas y reglas, así como de los sistemas automatizados o manuales, para la documentación y catalogación de estos medios (4).

En síntesis, el proceso de gestión de tipologías dinámicas en el ámbito de los audiovisuales puede resumirse en el siguiente esquema:

Figura 1. Proceso de gestión de tipologías

El diagrama expresa el proceso de gestión de tipologías de manera general, donde el analista de negocio se encarga del estudio de la organización donde se instalará la aplicación final personalizada. El propósito es identificar los tipos de materiales que se gestionan en la institución y los objetivos de la misma, para a partir de dicha información crear la ficha de catalogación. Estas fichas pueden ser o no guiadas por normas o reglas de catalogación, las cuales permiten una estandarización, pero no son obligatorias. A partir de las fichas creadas se procede a personalizar el sistema y luego a desplegarlo en la organización o empresa, garantizando la informatización de sus procesos de una manera ágil y eficaz. La rápida personalización de los sistemas de este tipo permite que los mismos puedan ser competitivos en el mercado y rentables para la entidad desarrolladora.

Con el uso extendido de las aplicaciones web en el entorno de la gestión de audiovisuales y en el departamento Señales Digitales, es necesario desarrollar uno o varios componentes para la gestión de tipologías en aplicaciones web dinámicamente, de forma tal que se permita definir la información que se desea almacenar en dependencia de las necesidades del cliente de manera rápida y personalizada. De esta forma se consume menos tiempo, personal y recursos.

1.4 Situación problemática

La organización lógica de materiales disponibles es indispensable para el trabajo óptimo y el conocimiento real de los recursos que se poseen. En la actualidad existe una notable cantidad de instituciones que se dedican a la recuperación y catalogación de audiovisuales, como es el caso del Instituto Cubano de Radio y Televisión (ICRT), la Oficina de Información de Comité Central del Partido Comunista de Cuba, el Instituto Cubano de Arte e Industria Cinematográfica (ICAIC), entre otros. En el Centro de Desarrollo GEYSED de esta casa de altos estudios, el departamento Señales Digitales se encarga de la elaboración de productos informáticos encaminados a gestionar y procesar materiales audiovisuales. Por la variabilidad de los entornos de aplicación de sistemas de este tipo, los datos de descripción de las medias deben ser modificables fácilmente pues no en todas las instituciones se necesitan los mismos campos de descripción. Algunas veces para un cliente se hace necesario guardar determinada información de las multimedias que no es relevante para otro. A esto se añade que los tipos de contenido que se almacenan también influyen en la información que se utiliza para describirlos, pues normalmente no se necesita guardar los mismos datos de una película que de una conferencia científica, los objetivos de ambos archivos multimedia no son los mismos, por lo que la información que se debe almacenar debe estar en correspondencia con su futura aplicación y con los intereses que se quiera alcanzar con el uso de dicho archivo multimedia.

El proyecto Catalogación y Publicación de Medias tiene como propósito implementar un producto que abarque diferentes oportunidades de negocio, como se ha dicho anteriormente se han unificado varios módulos del antiguo producto Sistema de Captura y Catalogación de Medias y VideoWeb. En el primero de estos dos productos existe un módulo para la gestión de tipologías de audiovisuales implementado en Symfony 1.4 con Dojo, el cual presenta algunas limitantes, por ejemplo: la interfaz de usuario no se llegó a terminar por diversos inconvenientes con el framework Dojo y porque la estructura de base de datos que presenta la aplicación es grande y compleja, lo que hace demasiado lento y engorroso su desarrollo. Por otra parte, la plataforma VideoWeb contiene un módulo de gestión de tipologías que permite la creación de fichas de descripción personalizadas que ha sido implementado al igual que el resto de la plataforma con la herramienta Drupal en su versión 6, este módulo tiene la misma deficiencia del descrito anteriormente en cuanto a su estructura de datos, ya que las consultas se hacen muy trabajosas, siendo la curva de aprendizaje del trabajo con esta estructura bastante alta. También se ha detectado que la interfaz de usuario para la gestión de tipologías es poco intuitiva y no se muestra la información de manera

organizada. Además de lo descrito anteriormente el equipo de desarrollo ha determinado la necesidad de migrar la plataforma a Drupal 7, puesto que Drupal 6 pronto dejará de tener soporte por la comunidad de desarrollo y si se quiere obtener un producto competitivo en el mercado será necesaria su migración, además de que la nueva versión de Drupal incorpora elementos que facilitan el trabajo con tipos de contenido dinámico. Se ha determinado también que es necesaria la incorporación de la gestión dinámica de tipologías en la nueva versión del sistema, pues es uno de los puntos vitales del sistema para lograr personalizaciones más rápidas.

A partir de estas razones se identifica que los módulos de gestión de tipologías con los que se cuenta no satisfacen las funcionalidades que se necesitan para el producto Sistema de Catalogación y Publicación de Medias.

1.5 Análisis de otras soluciones existentes

En la actualidad existen diferentes herramientas de software que asisten la catalogación de audiovisuales y otros tipos de contenidos multimedia. La mayoría de estas herramientas son privativas como por ejemplo Videoma. Por otra parte, se presentan las soluciones de los antiguos proyecto CCM y VideoWeb, los cuales fueron desarrollados en la Universidad de las Ciencias Informáticas.

1.5.1 Videoma Archivo

Videoma Archivo es una solución de software para la gestión digital de video, audio e imagen. Realiza la ingesta de contenido en cualquier formato, catalogación, gestión, publicación y recuperación mediante campos de búsqueda previamente definidos por el usuario. Está diseñado para trabajar en red, a través de múltiples puestos desde los que se pueden efectuar consultas del material almacenado en el servidor central. El sistema es modular y escalable. Videoma Archivo posee la capacidad de capturar y *transcodificar*[7] cualquier tipo de formato disponible en el mercado. (17)

El Módulo API[8] de Videoma Archivo, basado en un servicio web permite tener una interfaz pública integrada en la web y acorde con la imagen corporativa de la Entidad. Videoma trae consigo diversas ventajas como, sistemas de documentación jerarquizado basado en metadatos, importación y exportación

[7] Traducción, conversión, transformación o pasaje de un formato a otro. (18)

[8] Interfaz de programación de aplicaciones.

de metadatos. Además su arquitectura web mejora los flujos de trabajo, tiene administración personalizada de usuarios y grupos de usuarios, y gestión completa de vídeo, audio e imagen. Es un sistema abierto conformado por módulos con funcionalidades complementarias y brinda la posibilidad de adicionar nuevos módulos desarrollados bajo la plataforma Videoma. Su funcionamiento es estable diariamente. También posibilita la publicación de contenido en un portal web y su integración con otras tecnologías del mercado. (17).

Videoma es una solución de software propietario, lo cual dificulta el proceso de adquisición y mantenimiento, esto provoca que no se pueda utilizar como una solución viable, pero luego del análisis de sus funcionalidades se pueden identificar operaciones candidatas a implementar en el módulo de gestión de tipologías. Siguiendo la idea basada en un servicio web que permite tener una interfaz pública integrada y cumpliendo con la característica de tener una administración personalizada de usuarios y grupos de usuarios, se desarrolló una capa de servicios que permiten ofrecer de manera regulada las funcionalidades de gestión de tipologías que se implementaron.

1.5.2 VideoWeb

La Plataforma VideoWeb es un sistema informático formado por componentes de hardware y software que permite la transmisión de video y audio digital utilizando tecnología *streaming*[9]. Brinda servicios de video y audio sobre Internet e incorpora gestores de usuarios, de contenido audiovisual e informativos, los cuales facilitan la gestión de la información audiovisual. El éxito radica en el desarrollo del producto con plataforma completamente libre; además de la gestión dinámica de información asociada a los archivos multimedia, que permite cumplir las exigencias del cliente. Actualmente cuenta con una versión liberada por Calisoft y registrada en el Centro Nacional de Derecho de Autor (CENDA). Se está desarrollando una nueva versión que mantiene las funcionalidades anteriores e incluye nuevas que permiten ampliar el espectro de actuación del sistema.

Este producto consta de varios módulos que permiten que la plataforma sea flexible, como es el caso del módulo de gestión de tipologías. Este módulo tiene como objetivo crear fichas de descripción

[9] Tecnología utilizada para permitir la visualización y la audición de un archivo mientras se está descargando, a través de la construcción de un buffer por parte del cliente. (19)

personalizadas, pero está implementado con Drupal en su versión 6 y actualmente se necesita migrar el producto a Drupal 7.

1.5.3 SCCM

El producto Sistema de Captura y Catalogación de Medias (SCCM) fue desarrollado en la Universidad de las Ciencias Informáticas (UCI), sobre plataformas libres. Está dirigido a empresas e instituciones con control sobre un gran número de materiales multimedia. Presenta como principales funciones, la reproducción, edición y catalogación de medias. También búsquedas basadas en criterios específicos o generales, exportación de medias o cortos almacenados y transcodificación a diferentes formatos de audio o video. Actualmente no existe como proyecto, sin embargo se tienen del mismo un grupo de módulos de software, uno de ellos el de gestión de tipologías. Pero no puede ser utilizado este módulo porque la interfaz de usuario no se llegó a terminar y presenta una estructura de su base de datos estática y compleja haciendo difícil su utilización, ya que no permite crear tablas ni campos de tipologías adicionales cada vez que el usuario necesite ingresar más información. Además ha sido desarrollado en Symfony con Dojo.

1.6 Tecnologías y herramientas para el módulo web

Para el desarrollo del módulo se hizo necesaria la utilización de varias herramientas y tecnologías. Este grupo lo conforman: la metodología para guiar el proceso de desarrollo del sistema, el lenguaje de modelado y la herramienta CASE[10] que soporta dicho modelado. También el sistema de gestión de contenido y el gestor de base datos elegido, además de los lenguajes de programación, el entorno de desarrollo integrado, así como las tecnologías de comunicación y el control de versiones seleccionado.

1.6.1 Metodología de desarrollo de software

Para contar con un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización del desarrollo, es necesaria la aplicación de una metodología con la cual se puede mantener una fácil administración de este proceso. Una metodología de desarrollo de software, surge ante la

[10] *Computer Aided Software Engineering o* Ingeniería de Software Asistida por Computación.

necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto (software). (20)

RUP, resumido así por sus siglas en inglés *Rational Unified Process o* Proceso Unificado de Desarrollo, es un producto del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos. Está dividido en cuatro fases, que se dividen en iteraciones, la fase Inicio (Incepción o Concepción), la fase Elaboración, la fase de Desarrollo (Implementación o Construcción) y la fase de Cierre (Transición). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Es una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo, es decir, es un proceso que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. (21)

Al utilizar esta metodología se garantizará unificar todo el equipo de desarrollo de software y optimizar su comunicación al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado (UML) y un punto de vista de cómo desarrollar software. Por la cantidad de documentación que genera, posibilita mejor desenvolvimiento del proyecto, garantizando que el producto cumpla con las necesidades de los usuarios, a través de la especificación de los requisitos. Además, si ocurre alguna inestabilidad en el equipo de desarrollo, este gran volumen de información ayuda a capacitar al nuevo personal. De esta manera el software se desarrollará con mejor calidad, más rendimiento, reutilización y seguridad.

1.6.2 Lenguaje de modelado

Una de las exigencias de la gran mayoría de instituciones es que los desarrollos de software se formalicen con un lenguaje estándar y unificado. Es decir, se requiere que cada una de las partes que comprende el desarrollo de todo software de diseño orientado a objetos, se visualice, especifique y documente con lenguaje común. Para esto se necesita un lenguaje gráfico, a fin de especificar y documentar un sistema de software de un modo estándar.

UML *Unified Model Language* o Lenguaje de Modelado Unificado se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (22) El uso de UML para el modelado del módulo de tipología

trae como ventaja la posibilidad de visualizar su diseño y compararlo con los requisitos propuestos por el cliente antes de que el equipo de desarrollo comience a implementarlo. De esta manera si es necesario realizar algún cambio, se reducirá el tiempo empleado en la construcción del módulo y se garantizará que su desarrollo sea seguro.

1.6.3 Herramienta CASE

Las herramientas CASE permiten Modelar los Procesos de Negocios de las empresas y desarrollar los Sistemas de Información Gerenciales. Además, permite que los sistemas (especialmente los complejos), se tornen más flexibles, comprensibles y mejoran la comunicación entre los participantes. (22)

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML, ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de Sistemas que están interesados en construcción de aplicaciones informáticas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. (23)

La misma ha sido seleccionada por su fácil uso, aparte de que tiene soporte multiplataforma. También soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Además tiene como ventajas la generación de documentación en formatos HTML y PDF, código a partir de los diagramas e ingeniería inversa. Exporta e importa diagramas en el estándar XML y como imágenes, y tiene la capacidad de crear el esquema de clases a partir de una base de datos y viceversa.

1.6.4 Sistema de gestión de contenido

El sistema de gestión de contenido, también son conocidos como gestores de contenido web (Web Content Management o WCM), es un software que se utiliza principalmente para facilitar la gestión mediante la web. Proporciona un entorno que posibilita la actualización, mantenimiento y ampliación de la web con la colaboración de múltiples usuarios. (24)

Drupal (v.7) es un CMS que se distribuye como software libre bajo licencia GNU GPL (*General Public License*) versión 2 o superior. Puede ser modificado y distribuido libremente, pero siempre se debe hacer bajo la misma licencia, de forma que también puedan ser modificados y redistribuidos libremente por otras personas. (25) Drupal es fácilmente escalable y adaptable a las necesidades de los usuarios gracias a la

gran cantidad de módulos disponibles. El código de Drupal está optimizado, por lo que es un sistema relativamente rápido teniendo en cuenta lo amplio de su espectro de aplicación. (26)

Se utiliza Drupal en su versión 7 ya que facilita la creación y publicación de contenidos y la administración básica. También brinda la posibilidad de trabajar con diferentes sistemas de gestores de bases de datos como PostgreSQL. Además, supone ventajas para un módulo de gestión de tipologías de contenido audiovisual como el que se quiere desarrollar, debido a que esta versión propone una nueva forma de organizar la información a partir del módulo *Entity*[11]. El mismo proporciona una forma unificada para hacer frente a las entidades y sus propiedades, permitiendo soluciones más ligeras y flexibles. Además, proporciona un controlador *CRUD*[12] entidad, que ayuda a simplificar la creación de nuevos tipos de entidad. (27) En versiones anteriores se utilizaban los nodos para guardar diversos tipos de información con la finalidad de mostrarla al usuario. Con la filosofía que propone *Entity* ahora es posible crear nuevos tipos de contenido más específicos, las entidades, que son unidades lógicas de almacenamiento de información.

Las entidades pueden dividirse en subentidades o tipos de entidad, y estas últimas pueden agruparse en paquetes que constituyen especializaciones en cuanto a estructura y comportamiento de las entidades, funciona en la práctica como una herencia desde las entidades hasta las agrupaciones inferiores permitiendo especializar cada vez más los contenidos a través de sus campos de información y funcionalidades. Esta filosofía permite que la creación de contenido dinámico se realice de forma más sencilla.

1.6.5 Gestor de base de datos

Un Sistema Gestor de Bases de Datos (SGBD), en inglés *Data Base Management System* (DBMS) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos

---

[11] Entidad: Instancia de un tipo de entidad en particular, como un comentario, término de taxonomía o perfil de usuario o de un grupo, como un blog, un artículo o producto. (27)

[12] (*Create, Read, Update, Delete*) o (Crear, Listar, Modificar, Eliminar) entidades.

y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (28)

PostgreSQL (v.9.1) es un SGBD objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution*), con su código fuente disponible libremente y multiplataforma. Permite procesar o trabajar con grandes volúmenes de datos y una alta concurrencia de usuarios accediendo a la vez al sistema, soporta diversos tipos de datos e incluye la herencia entre sus tablas y tiene soporte para el control de versiones. (29) Además es el utilizado por la plataforma VideoWeb 2.0, por tanto es conveniente seleccionarlo para garantizar la compatibilidad del módulo con la misma.

1.6.6 Lenguaje de programación

Existen lenguajes y técnicas de desarrollo web, que ofrecen a los desarrolladores características interesantes para añadir a sus sitios web, como son los lenguajes del lado del cliente (HTML, CSS, JavaScript) y los del lado del servidor (PHP). La diferencia más básica entre ellos es que los lenguajes de programación del lado del cliente se ejecutan en el navegador web, en cambio en el lenguaje de programación de lado servidor, se ejecutan en el servidor sin tener en cuenta el navegador web del cliente.

HTML (v5), *HyperText Markup Language o* lenguaje de marcado de hipertexto, es un lenguaje compuesto de una serie de etiquetas o marcas que permiten definir el contenido y la apariencia de las páginas web.(30) Se basa en el metalenguaje SGML (*Standard Generalized Markup Language),* estándar internacional para la definición de métodos de representación de texto en forma electrónica no ligados a ningún sistema ni a ningún dispositivo, lo que quiere decir que todo irá encerrado entre dos etiquetas, normalmente, una que indica el comienzo de algún elemento del documento y otra que indica el final del mismo. (31)

Se utiliza para definir la estructura lógica del documento, o sea, la forma en la que debe aparecer en el navegador el texto, las imágenes y los demás elementos que contengan el módulo, en la pantalla del ordenador. Además, cuando el navegador del cliente solicita algunas páginas, permitirá hacer hiperenlaces, facilitando la relación con otras fuentes de información.

CSS (v3), *Cascading Style Sheets* u Hojas de Estilo en Cascada, es la tecnología desarrollada por el *World Wide Web Consortium* (W3C)[13] con el fin de separar la estructura de la presentación. (32) Con la utilización de CCS se definirá el "estilo visual" del módulo sin necesidad de hacerlo etiqueta por etiqueta, para cada una de las páginas, o sea, se diseñará el estilo de forma independiente al documento HTML y se podrá aplicar a todo el documento o algunos de sus niveles. Esto evitará hacer a los archivos demasiado pesados, reduciendo de esta manera, el tiempo de carga de las páginas web y facilitará la consistencia y homogeneidad en el diseño y la imagen del módulo de tipologías.

JavaScript es un lenguaje *script*[14] usado para incluir código en el cliente que es interpretado por el navegador. (33) Está basado en objetos pero no es orientado a objeto, es interpretado por el cliente y no requiere de compilación. Se utiliza ya que permite realizar validación de los datos de entrada de los clientes, permitiendo crear páginas dinámicas y seguras, lo que las hace más atractivas para el usuario. Su uso proporcionará la conformación de formularios con un mejor acabado para la gestión de tipologías, lo que le otorgará al software mejores características de usabilidad.

PHP (v5.3) en su inicio significaba *Personal Home Page*, pero con el tiempo al ser desarrollado por otro grupo se llama PHP *Hypertext preprocesor*. (35) Es un lenguaje interpretado, multiplataforma y sólo se necesita un servidor que lo interprete para ejecutarlo. Este está difundido gracias a su código abierto y se utiliza principalmente para la elaboración de sitios web que presenten una fuerte gestión y consultas a base de datos como es el caso de la plataforma VideoWeb a la cual pertenece el módulo a desarrollar. Además, permite las técnicas de Programación Orientada a Objetos (POO) y brinda la posibilidad de conectarse a la mayoría de los gestores de base de datos como PostgreSQL.

---

[13]El *World Wide Web Consortium* (W3C) es la organización que desarrolla los estándares para normalizar el desarrollo y la expansión de la Web y la que publica las especificaciones relativas al lenguaje HTML. (32)

[14]Un script es un tipo de programa que consiste de una serie de instrucciones que serán utilizadas por otra aplicación. (34)

1.6.7 AJAX

AJAX es el acrónimo inglés para *Asynchronous JavaScript and XML* (JavaScript y XML asíncrono). No es una tecnología, sino la unión de varias tecnologías. Se utiliza por ser una técnica de desarrollo web que genera aplicaciones interactivas, respondiendo a las interacciones del usuario rápidamente. Permite la comunicación asincrónica a la base de datos evitando la recarga de las páginas, actualizando porciones de la página en vez de la página completa. De esta manera, a la hora de cargar una ficha de catalogación o guardarlas y buscar tipologías evitará que la interfaz visual sobre la que trabaja el usuario se refresque cada vez que realice una de estas operaciones. Además es soportado por muchos exploradores web, entre los que se encuentra Mozilla FireFox. (36)

1.6.8 Tecnologías de comunicación

Las funcionalidades de gestión de tipologías son útiles en diversos productos del Departamento Señales Digitales, para que otros sistemas puedan utilizar estas funciones se decidió utilizar servicios web. Un servicio web es cualquier servicio que está disponible a través de Internet, utiliza un sistema de mensajería XML estandarizada, y no está ligada a ningún sistema operativo o lenguaje de programación. (37) En el caso del módulo de gestión de tipologías se decidió utilizar como tecnologías de comunicación SOAP y *XML-RPC*[15]. SOAP es un protocolo de comunicación, basado en XML para intercambiar información entre ordenadores, en el caso de XML-RPC, es un protocolo simple que utiliza mensajería XML para realizar llamadas a procesamientos remotos, donde las solicitudes son codificadas con XML y se envía a través de HTTP POST. Ambos son independientes de la plataforma y del lenguaje. (37) Drupal 7 permite la creación de servicios web a partir de estas dos tecnologías, además, son las más utilizadas en los productos del departamento y permiten el intercambio de datos de manera efectiva.

1.6.9 Control de versiones

Un sistema de control de versiones es un sistema de gestión de archivos y directorios, cuya principal características es que mantiene la historia de los cambios y modificaciones que se han realizado sobre ellos a lo largo del tiempo. De esta forma, el sistema es capaz de recordar las versiones antiguas de los

---

[15] XML: *eXtensible Markup Language* o Lenguaje de marcado extensible.
RPC: *Remote Procedure Call* o Llamadas a procesamientos remotos.

datos, lo que permitirá examinar el histórico de cambios o recuperar versiones anteriores de un fichero, incluso aunque haya sido borrado. (38)

Subversion, también conocido como SVN, es un sistema de control de versiones libre y de código abierto. Está preparado para funcionar en red, y se distribuye bajo una licencia libre de tipo Apache. SVN posee las características de tener soporte tanto de ficheros de texto como de binarios y mantiene versiones no sólo de archivos, sino también de directorios, así como las versiones de los metadatos asociados a los directorios. Además de los cambios en el contenido de los documentos, se mantiene la historia de todas las operaciones de cada elemento, incluyendo la copia, cambio de directorio o de nombre. (38)

1.6.10 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (IDE), también conocido como entorno de diseño integrado, entorno integrado de depuración o entorno de desarrollo interactivo, es una aplicación de software que proporciona servicios integrales a los programadores de computadoras para el desarrollo de software. (39)

NetBeans IDE se utiliza ya que es de código abierto, libre y gratuito sin restricciones de uso. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación como PHP, Javascript, HTML y CSS. Además escanea todos los proyectos que se crean o se cargan, reconociendo funcionalidades, clases, atributos, propiedades. Facilita un completamiento de código eficiente para los desarrolladores. Tiene soporte para el sistema de control de versiones, AJAX, PHP y modelado UML. (40)

1.7 Conclusiones parciales

El estudio de los conceptos asociados al dominio del problema facilitó una mejor comprensión sobre el proceso de gestión de tipologías para los medios audiovisuales. Se describieron soluciones ofrecidas a la problemática presentada, pero se determinó que ningunas de ellas pueden ser utilizadas porque no satisfacen todos los requisitos de la aplicación que se desea implementar. A pesar de esto, se toman las funcionalidades más relevantes para un mejor desarrollo del producto, haciéndolo más completo e intuitivo para el usuario. Las herramientas y tecnologías seleccionadas para el desarrollo del módulo permitirán que la solución sea flexible y adaptable a la plataforma VideoWeb 2.0.

CAPÍTULO 2. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

2.1 Introducción

El presente capítulo se refiere al diseño de la aplicación propuesta, muestra los diagramas de clases del diseño, el diagrama de despliegue y el diagrama de componente, así como sus respectivas descripciones. Se especifica el patrón arquitectónico a utilizar así como los patrones de diseño y los elementos referentes a la base de datos a utilizar como el diagrama entidad-relación y la descripción de las tablas, además del estándar de codificación manejado en la aplicación.

2.2 Descripción del sistema

A partir del análisis realizado en productos anteriores desarrollados en el departamento se está en condiciones de realizar un nuevo diseño de clases teniendo en cuenta que los requisitos no han cambiado, por lo que permiten conocer cuáles son las funcionalidades que se deben implementar en esta nueva versión, con la diferencia de que se realizará con nuevas tecnologías.

El nuevo módulo de tipologías tiene el objetivo permitir gestionar las tipologías del sistema así como sus campos de una manera más rápida ante la solicitud del cliente. Posibilitará gestionar tantas tipologías de archivos multimedia como sean necesarias, permitiéndole a la plataforma la cualidad de ser más flexible a la hora de personalizarla.

Se ha creado una capa de servicios web en la que se exponen las funcionalidades principales que implementa el módulo. A través de los servicios, otras aplicaciones desarrolladas en el departamento pueden consumir la lógica implementada y reutilizarla, siendo de utilidad para el resto de las soluciones que requieren la gestión de tipologías dinámica.

El desarrollo de estas funcionalidades otorgará a la plataforma una rápida personalización del sistema para diversos clientes. Permite a los usuarios y desarrolladores personalizar la aplicación en dependencia de su entorno. Como resultado del trabajo de los miembros del proyecto y de tesis anteriores ya se han obtenido los requisitos y casos de uso a los que debe dar respuesta el software que se implementará, los mismos pueden ser consultados en los Anexos 1 y 2, para que el lector tenga una visión más amplia de las características del módulo.

2.3 Diagrama de clase del diseño

El Diagrama de Clases del Diseño (DCD) describe gráficamente las especificaciones de las Clases de Software y las Interfaces en una aplicación, o sea, muestra cómo quedaría implementada la aplicación en términos lógicos. Para un mejor entendimiento del mismo, se muestra el diagrama de clases del diseño de cada caso de uso del módulo de tipología.

2.3.1 DCD para el caso de uso ¨Gestionar tipologías¨

Para implementar el caso de uso Gestionar tipologías que describe el proceso de adicionar, editar y eliminar tipologías en el sistema, se diseñó el diagrama de clases que se muestra en la Figura 2.

El mayor peso recae sobre las clases:

tipologia_controlador: en esta clase se encuentra el método *Mostrar_Formulario_Editar_Tipologia()* que tiene como objetivo mostrar el formulario para crear y editar las tipologías. Además contiene el método *Mostrar_Formulario_Eliminar_Tipologia()* que se encarga de mostrar el formulario para que el usuario confirme si quiere eliminar una tipología determinada.

tipologia_vista: cuyos métodos *Formulario_Editar_Tipologia()* y F*ormulario_Eliminar_Tipologia()* son los encargados de mostrar las formularios para editar la tipología y eliminarla.

tipología_modelo: esta clase se encarga de consultar la base de datos para obtener el resultado según la selección del usuario, a través de los métodos *Tipologia_Delete_Confirm_Submit()* y *Eliminar_Tipologia()* para realizar el proceso de eliminación de la tipología.

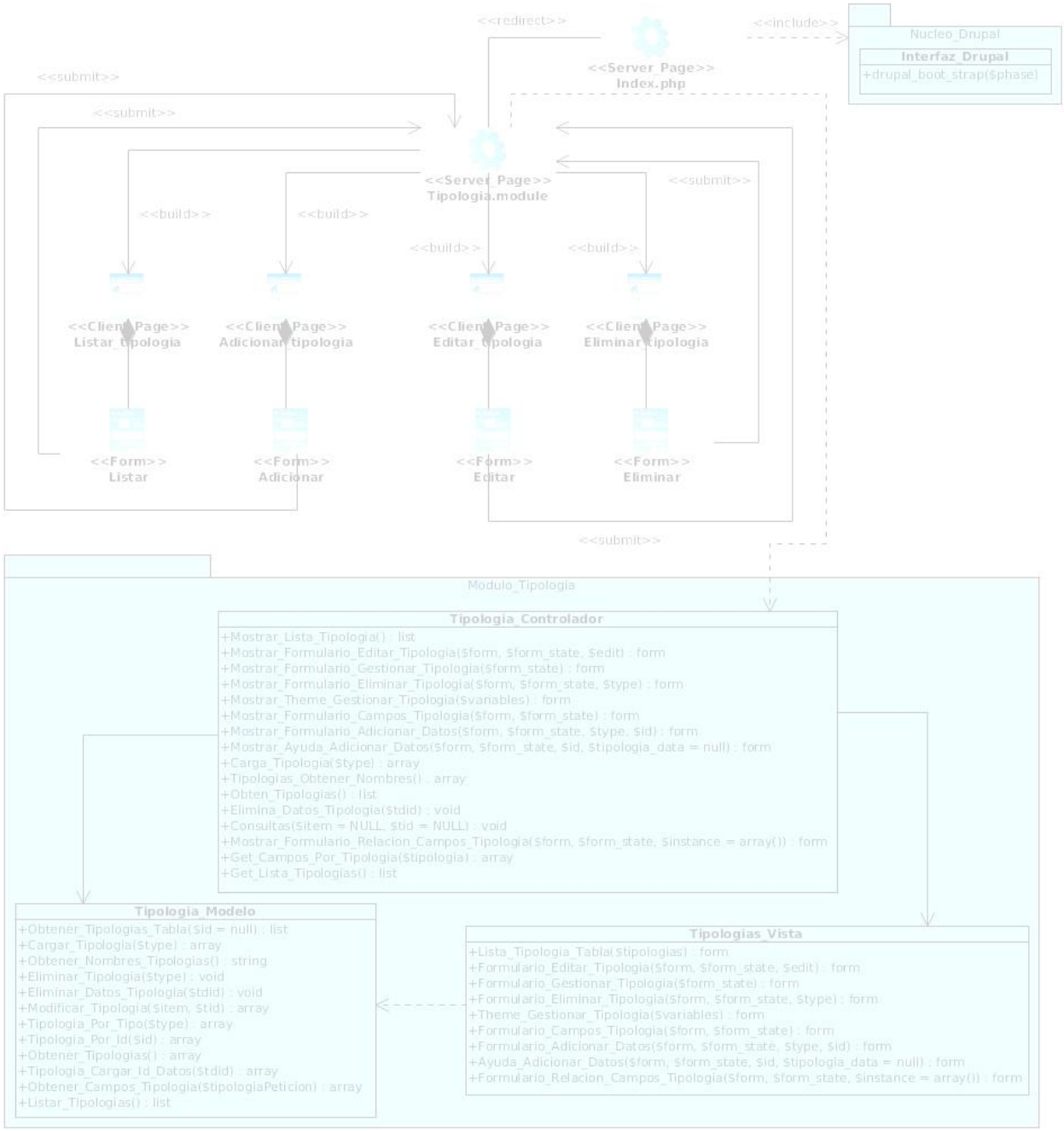


Figura 2. DCD del caso de uso Gestionar tipologías

2.3.2 DCD para el caso de uso ¨Gestionar campos de tipologías¨

Al ser dinámico el módulo de tipología, da la opción de crear nuevos campos a medida que vayan siendo necesarios a través del modelo *Entity*. El caso de uso Gestionar campos de tipologías es el encargado del proceso de creación de campos para las tipologías. A continuación se representa en la Figura 3, el DCD para este caso de uso.

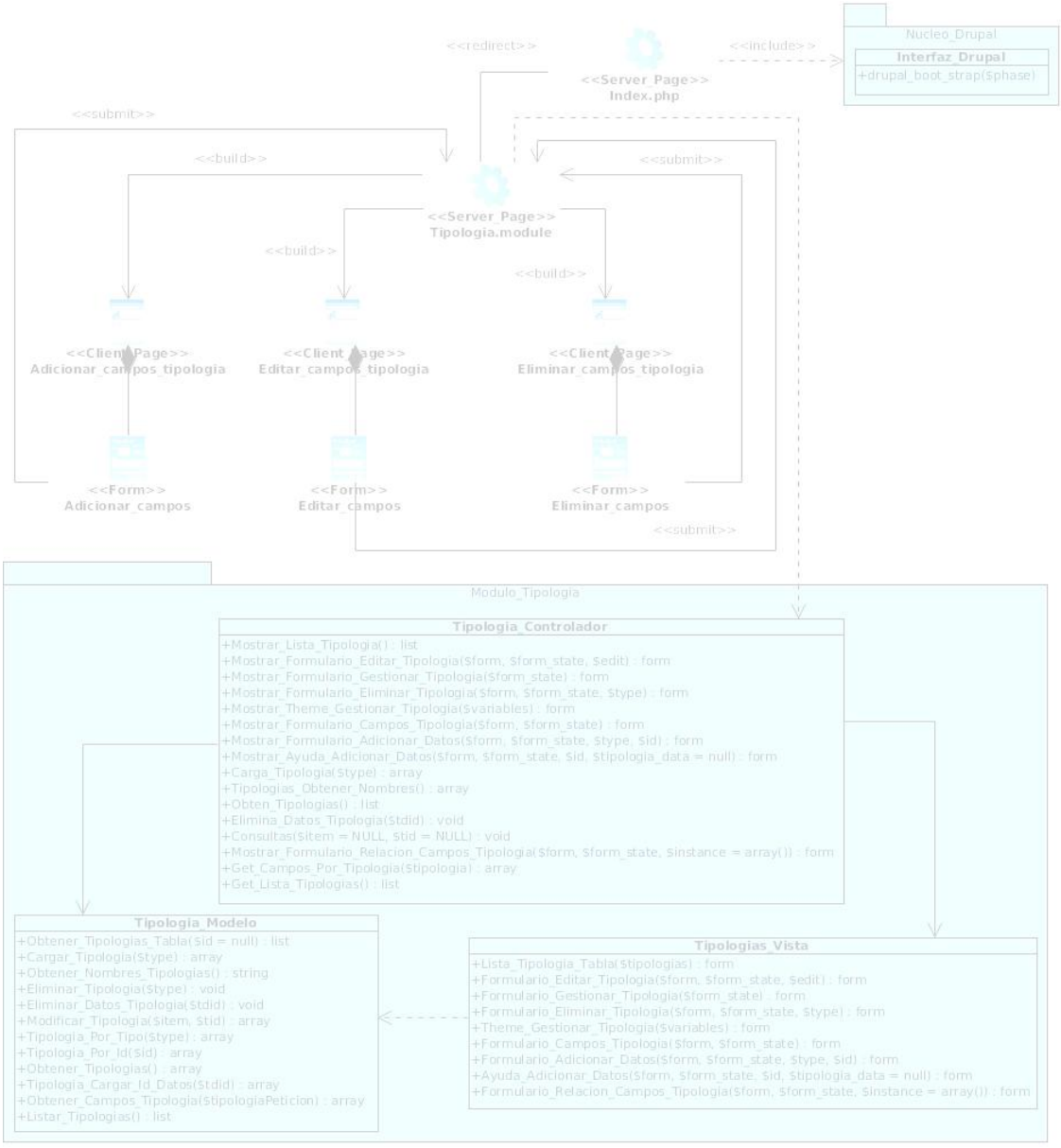


Figura 3. DCD del caso de uso Gestionar campos de tipología

2.4 Patrón arquitectónico

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software. (41)

Un patrón de arquitectura de software es un esquema genérico probado para solucionar un problema particular, el cual es recurrente dentro de un cierto contexto. Este esquema se especifica describiendo los componentes, con sus responsabilidades y relaciones (42), por ejemplo el patrón Modelo- Vista- Controlador.

2.4.1 Modelo – Vista– Controlador

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista. (43) Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas. (44)

Definición de las partes:

El **Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo. (44)

La **Vista** es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo. (44)

El **Controlador** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo. (44) La finalidad del controlador es mejorar la reusabilidad por medio del desacople entre la vista y el modelo. (43)

El módulo será implementado utilizando de este patrón arquitectónico ya que, permite establecer una estructura para todos los componentes del módulo y determina la relación entre ellos. Además se podrán realizar cambios en cada una de sus partes (vista, modelo y controladora), sin afectar a las demás, garantizando el desarrollo de manera independiente de estas.

2.5 Patrón del diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software (45) Estos patrones proporcionan una estructura conocida por todos los programadores, de manera que la forma de trabajar no resulte distinta entre los mismos. Además permiten tener una estructura de código común a todos los proyectos que implemente una funcionalidad genérica, así el software construido es más fácil de comprender, mantener y extender.

2.5.1 GRASP

GRASP (*General Responsibility Assignment Software Patterns*), es un acrónimo que significa Patrones Generales de Software para Asignar Responsabilidades. (46) En esta investigación se han definido los siguientes patrones de asignación de responsabilidades:

Experto: Este patrón se encarga de asignar una responsabilidad al experto en información, es decir, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad. (46) En la solución se evidencia la utilización de este patrón en la clase **tipologia_modelo**, la cual se centra en el acceso a los datos y en ella recae la responsabilidad de manipularlos, o sea, tiene la información referente a las tipologías ya que las funcionalidades para manipularlas (modificarlas o eliminarlas) están implementadas en esta clase.

Creador: Este patrón ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. Es el encargado de asignarle a la clase B la responsabilidad de crear una instancia de clase A. B es un creador de los objetos A. (46) Se puede percibir el patrón creador en la clase

tipologia_controlador en la función *Mostrar_Lista_Tipologia()*, donde se crea una instancia de la clase **tipologia_modelo** (*$modelo_tipologia = new tipologia_modelo();*)

Controlador: Este patrón se utilizó ya que asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas facilitando la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, permitiendo aumentar la reutilización de código y a la vez tener un mayor control. (47) En la implementación del módulo se puede ver que la clase controladora **tipologia_controlador** se encarga de dirigir las funcionalidades del sistema que están distribuidas en las demás clases como las de gestionar las tipologías así como sus campos. La función *Mostrar_Lista_Tipologia()* perteneciente a esta clase, es la encargada de asignarle la responsabilidad a la clase **tipologia_modelo** de obtener la lista de tipología existente en la base datos y a la clase **tipologia_vista** que la muestre.

Alta Cohesión: Este patrón se utilizó con el propósito de que la información que almacena una clase sea coherente y deba estar (en la medida de lo posible) relacionada con la clase, o sea,todas sus características y operaciones están relacionadas entre sí. (46) En este caso, un ejemplo claro son las clases **tipologia_controlador**, **tipologia_modelo** y **tipologia_vista**, con responsabilidades estrechamente relacionadas que realizan un trabajo adecuado y ajustado a la clase.

Bajo Acoplamiento: Este patrón se utilizó para establecer la menor cantidad de clases ligadas entre sí, de esta forma, si ocurre alguna modificación en algunas de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. (46) Esto se evidencia en la estructura que se ha empleado (MVC), donde la vista, el modelo y la controladora están en clases diferentes y cada una cumple con una función específica.

2.5.2 GOF

GOF conocido como *Gang of Four* o Pandilla de los Cuatro, formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Son los autores del famoso libro "*Design Patterns: Elements of Reusable Object Oriented Software*". (47) Los patrones utilizados fueron:

Instancia Única (*Singleton*): Este patrón creacional garantiza una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. (47) El uso de dicho patrón se evidencia en

todos los módulos que posee Drupal y en los que se creen nuevos, los cuales contienen conjunto de *hooks*[16] y funciones que hacen que cada uno de ellos sean independientes (se puede entender como el funcionamiento de una clase).

Puente (*Bridge*): La capa de abstracción de bases de datos de Drupal se aplica de una forma similar a este patrón de diseño estructural, el cual desacopla una abstracción de su implementación. (47) Los módulos pueden ser definidos de forma que es independiente del sistema de bases de datos que se está utilizando, y proporciona la capa de abstracción para ello. Si se modifica el sistema gestor de bases de datos o se instala dicho módulo en un sistema diferente, la capa de abstracción definida por el API de bases de datos que incorpora el núcleo de Drupal se comporta como puente entre el origen y el destino del módulo en cuestión sin necesidad de modificar el código de este.

Accion (*Command*): Muchos de los *hooks* de Drupal utilizan este patrón de comportamiento, el cual encapsula una operación en un objeto, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma. (47) Para reducir el número de funciones que son necesarias para la aplicación, pasando la operación como un parámetro, junto con los argumentos. El propio sistema de *hooks* de Drupal utiliza este modelo, a fin de que los módulos no tengan que definir cada *hook* existente, sino sólo los que necesitan, o sea, se definen solamente los *hooks* que utiliza en el módulo de tipología y no todos los que implementa Drupal.

2.6 Diseño de la base de datos

Uno de los principales pasos en la elaboración de una base de datos, es la realización de una buena descripción de la misma, de este modo se logrará un mejor entendimiento y visualización de lo que se desea construir. Para esto se recurre al *modelo de datos*[17], el cual determina de qué manera los datos van a ser guardados, organizados y manipulados en un sistema de base de datos. Partiendo de lo anterior, se muestra a continuación el Modelo Entidad-Relación del módulo de gestión de tipologías de la plataforma VideoWeb 2.0.

---

[16] Ganchos

[17] Determinan la estructura de la información, con el objetivo de mejorar la comunicación y la precisión en aplicaciones que usan e intercambian datos. (48)

En la Figura 4 se representa solo una parte del modelo de datos, que constituye físicamente la base de datos del producto en general. En la misma se reflejan las tablas relacionadas con el proceso de gestión de tipologías que se implementa en el módulo, tales como t**ipologia**, **tipología_data**,**archivo_multimedia**, **relacion_am_tipologia**, **almacenamiento**, **almacen_local**, **field_config_instance** y **field_config**.

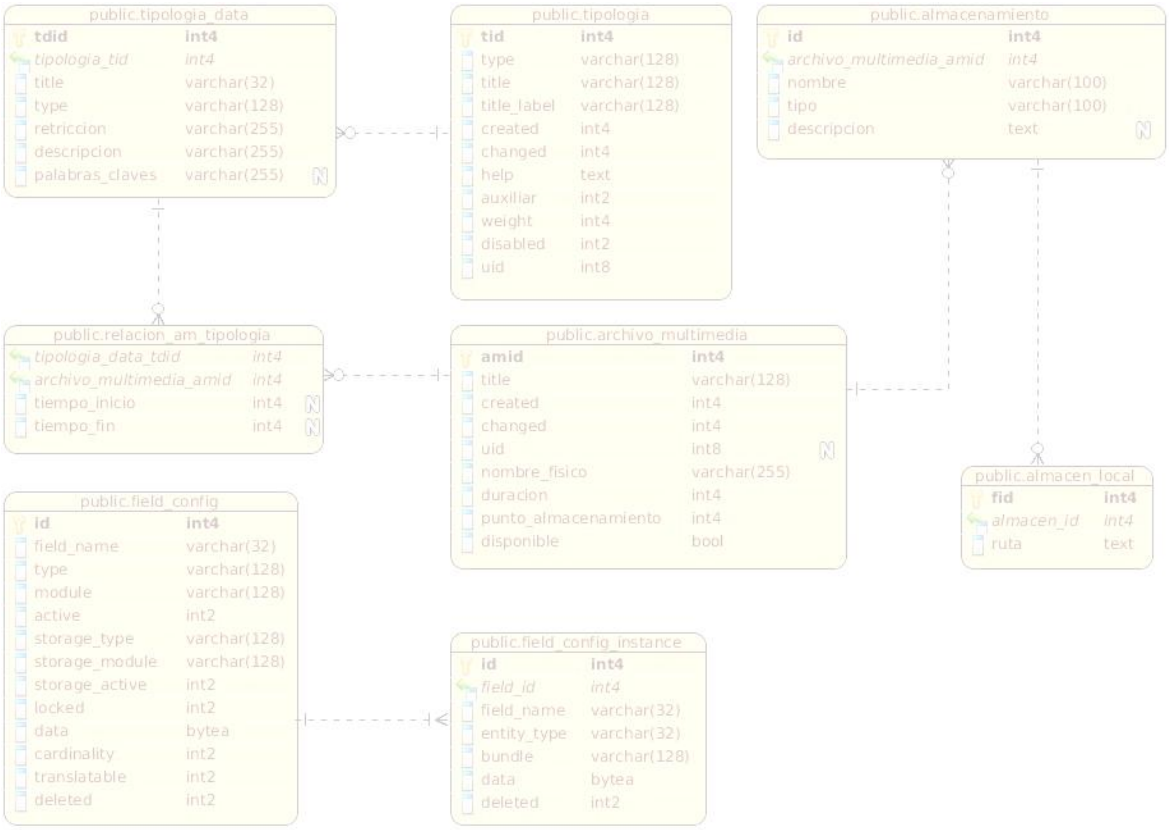


Figura 4. Diseño de la base de datos

En **tipología** se almacena todas las tipologías existentes las cuales van a tener como atributos el identificador de cada tipología (tid), el título de la tipología (title), tipo de tipología (type), ayuda o descripción (help), fecha de creación (created) y de modificación (changed de la misma, la variable auxiliar (auxiliar) que es para el caso de que la tipología no puede ser asociada a los archivos multimedia, entre otros parámetros. Esta tabla tiene una relación de uno a mucho con la tabla **tipología_data**. La misma

contiene los datos básicos de todos los materiales que se quieren gestionar, en este caso archivos multimedias. Esta tabla presenta atributos como: el identificador de la tipología (tid), el título de la tipología (title), tipo de tipología (type), la restricción, descripción y las palabras claves.

Por otra parte en la tabla **archivo_multimedia** se gestionan los datos propios de los archivos multimedia como su identificador (amid), el título del archivo multimedia (title), fecha de creación (created) y de modificación (changed), la duración de la media (duración), el nombre físico del lugar donde está almacenado (nombre_fisico), donde está almacenado, entre otros. La misma tiene una relación de mucho a mucho con la tabla *tipología_data*, por lo que se genera la tabla **relacion_am_tipologia**. Esta contiene el identificador del archivo mutimedia (amid) y el identificador de los datos asociados a él (tdid), así como el tiempo de inicio (tiempo_inicio) tiempo de finalización (tiempo_fin) de la media, ya que un archivo multimedia puede tener varios datos descriptivos, además del campo descripción y palabras claves.

En la tabla **field_config_instance** se guarda toda la información referente a la instancia realizada por un campo, como el nombre del campo, nombre de la entidad, entre otras. En el caso de **fiel_config**, al ser dinámico el módulo de tipología, se pueden crear varios campos así se requiera. En esta tabla se almacenan todos los datos configurables de los campos de tipología, por ejemplo: el nombre del archivo, el tipo, a que módulo pertenece, tipo de almacenamiento, entre otras.

2.7 Diagrama de despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. (49)

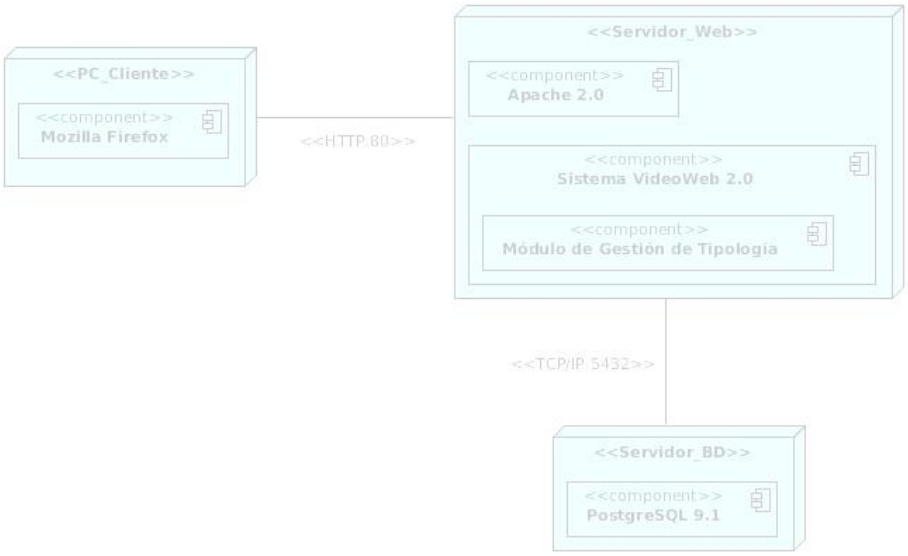


Figura 5. Diagrama de despliegue

Descripción de los Nodos:

PC Cliente: Nodo cliente por el cual el actor accederá a la aplicación utilizando un navegador web (Mozilla Firefox) y se conectará mediante el protocolo HTTP por el puerto de comunicación 80 al Servidor Web.

Servidor Web: Nodo en el cual se encontrará instalado el servidor Web Apache 2.0 el que alojará la aplicación web que contiene el componente.

Servidor de Base de Datos: En el nodo servidor de base de datos estará ejecutándose el sistema gestor de Base de Datos PostgreSQL 9.1. La comunicación es a través de las clases del modelo implementadas en el módulo de gestión de tipologías.

Descripción del protocolo de comunicación utilizado:

TCP/IP (Protocolo de control de transmisión/Protocolo de Internet): Es un conjunto de protocolos de red en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. (50)

HTTP (Protocolo de transferencia de hipertexto): Permite la transferencia de archivos (principalmente, en formato HTML) entre un navegador (el cliente) y un servidor web localizado mediante una cadena de caracteres denominada dirección URL. (51)

2.8 Modelo de implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se podrá 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. Para representar los diagramas del Modelo de Implementación se puede emplear el diagrama de UML de Componentes. (52)

El diagrama de componentes (ver Figura 6) muestra la distribución e interacción entre los componentes que conforman el módulo web de gestión de tipologías, teniendo en cuenta el diseño y la arquitectura establecida para el mismo.

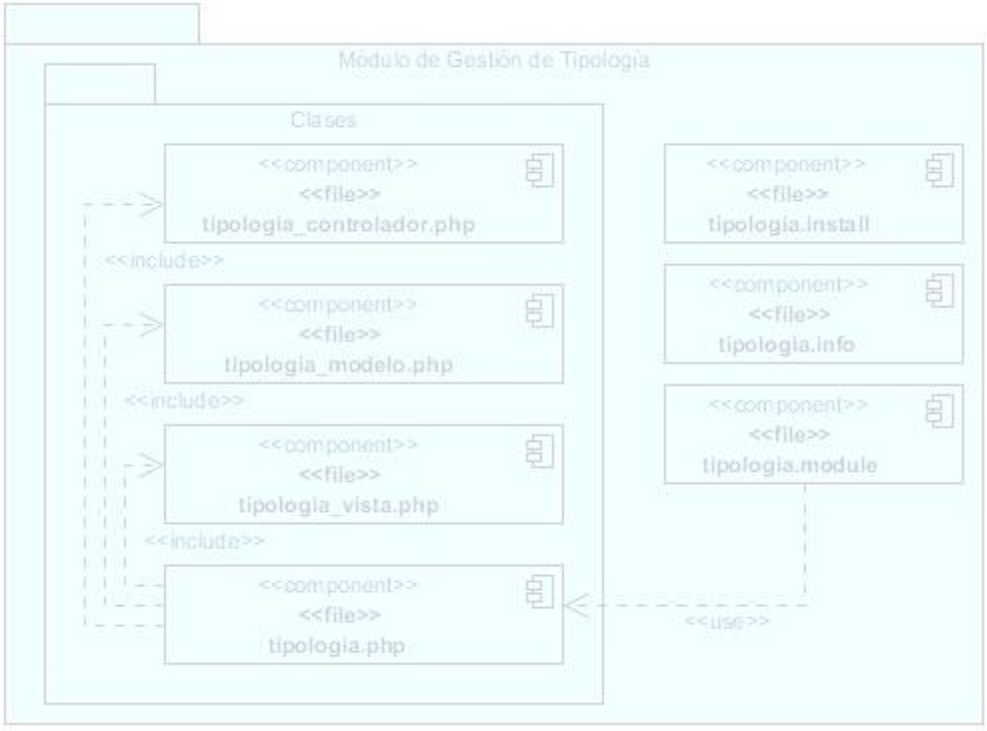


Figura 6. Diagrama de componentes

2.9 Estándares de codificación

Los estándares de codificación son reglas y descripciones que se siguen para la escritura del código fuente. El seguir un estándar de codificación en el código de programación es muy importante en cualquier

proyecto, principalmente si este involucra muchos desarrolladores, ya que sirve como punto de referencia para los programadores, lográndose mayor eficiencia en la reimplementación del código fuente y provee un estilo de programación.

A continuación se muestran principios generales del estándar de codificación seleccionado:

- Los nombres de cada uno de los elementos del programa deben ser significativos; su nombre debe explicar en lo posible el uso del elemento.
- No manejar en los programas más de una instrucción por línea.
- Declarar las variables en líneas separadas.
- Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.
- La mayoría de los elementos se deben nombrar usando sustantivos.
- Los atributos deben comenzar con letra minúsculas y los métodos deben comenzar con letra mayúsculas.
- Para distinguir palabras dentro del nombre deberá emplearse un guión bajo (_).

2.10 Conclusiones parciales

La utilización de patrones (arquitectónicos y de diseño de clases) permitió crear diagramas de clase del diseño íntegros para cada caso de uso del módulo web. Por otra parte, los diagramas de clases del diseño y el diseño de la base de datos, sirvieron de guía para la implementación del módulo de gestión dinámica de tipologías. En el caso del diagrama de despliegue permitió indicar cómo se ubicarán las partes dentro del entorno computacional físico que soportará el sistema.

A su vez, el diagrama de componentes se utilizó para modelar la vista estática de un sistema y el estándar de codificación como punto de referencia para los programadores, logrando un lenguaje común. De esta forma se garantiza una mejor compresión de la solución implementada y se han sentado las bases para que en futuros desarrollos se pueda efectuar el mantenimiento e implementación de nuevas funcionalidades de una manera más sencilla.

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de que se cometan errores son comunes, conllevando a que el producto final no sea lo esperado por el cliente. Debido a esto, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad y el cumplimiento de los objetivos propuestos. En el presente capítulo se abordará sobre la validación de la solución propuesta donde se dejan expuestos los resultados obtenidos en las pruebas de caja negra al módulo desarrollado.

3.2 Prueba de software

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. (53) Los dos objetivos principales del proceso de pruebas son:

- Maximizar el número de errores detectados (cobertura).
- Reducir al mínimo el número de casos de prueba (costo).

Existen varios tipos de prueba, cada una enfocada a un objetivo único, entre ellas se encuentran las pruebas de integración y de funcionalidad. Dichas pruebas, se le realizarán al módulo desarrollado para garantizar su correcto funcionamiento de acuerdo a los requisitos establecidos.

Las **pruebas de integración**, se llevan a cabo durante la construcción del sistema y consisten en el proceso de combinar y probar múltiples componentes juntos. Tiene como objetivo encontrar fallos en la respuesta de un módulo cuando su operación depende de los servicios prestados por otro(s) módulo(s), basándose en la especificación de requisitos del usuario. (53) Por lo que se realizaron pruebas de integración para comprobar que el módulo desarrollado se integrar correctamente con el resto de los módulos de la plataforma VideoWeb, en este caso Archivo Multimedia y Catalogación.

Dentro de las pruebas de funcionalidad se encuentra el método de **pruebas de caja negra**, conocidas también por pruebas de caja opaca, funcionales, de entrada/salida o inducidas por los datos, se centran

en lo que se espera del módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ende el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro. (53)

3.2.1 Prueba de integración

Se realizaron las pruebas de integración del módulo de gestión de tipologías con el módulo de servicios de Catalogación y el módulo de Archivos Multimedia. Al ejecutarse dichas pruebas, se detectó una no conformidad con respecto a la información que devolvía el servicio web a través del cual el módulo de servicios de Catalogación obtiene el listado de tipologías existentes, fue necesario refinar la respuesta ofrecida por este servicio. La integración con el módulo Archivos Multimedia funcionó correctamente.

3.2.2 Prueba de caja negra

Idealmente, se determinaron un conjunto de casos de prueba tales que su ejecución exitosa implique que no hay errores en el software desarrollado. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto. (53)

Tabla 1. Caso de prueba ¨Gestionar tipologías¨

Caso de prueba: Gestionar tipologías.					
Nombre de la sección	**Escenarios de la sección**	**Descripción de la funcionalidad**	**Flujo central**	**Resultados Esperados**	**Resultados Obtenidos**
SC 1: Crear Tipologías.	EC 1.1: Crear Tipologías satisfactoriamente.	El usuario selecciona la opción de Adicionar Tipologías, introduce los datos necesarios para la creación de la tipologías, luego presiona el botón “Guardar” y se muestra un mensaje	VideoWeb 2.0/Estructura/Tipología/Opción Adicionar Tipología/Inserta todos los datos de la tipología/ Presiona el botón “Guardar”.	El sistema crea la tipología, muestra un mensaje indicando que la tipología ha sido creada satisfactoria mente y actualiza la lista de tipología.	El sistema crea la tipología, muestra un mensaje indicando que la tipología ha sido creada satisfactoriamente y actualiza la lista de tipologías.

		indicando se ha creado satisfactoriamente la tipología.			
	EC 1.2: Crear Tipologías fallida.	El usuario presiona el botón "Guardar" antes de completar todos los datos obligatorios el sistema muestra un mensaje de error indicando que existen campos obligatorios por llenar.	VideoWeb 2.0/Estructura/Tipología/Opción Adicionar Tipología/Inserta todos los datos de la tipología/ Presiona el botón "Guardar".	El sistema muestra un mensaje indicando que existen campos obligatorios por llenar.	El sistema muestra un mensaje indicando que existen campos obligatorios por llenar.
	EC 1.3: Crear Tipologías existente.	El usuario completa todos los datos obligatorios y presiona el botón "Guardar". Luego el sistema muestra un mensaje de error indicando que existe la tipología.	VideoWeb 2.0/Estructura/Tipología/Opción Adicionar Tipología/Inserta todos los datos de la tipología/ Presiona el botón "Guardar".	El sistema muestra un mensaje indicando que existe la tipología.	El sistema muestra un mensaje indicando que existe la tipología.
SC 2: Editar Tipologías.	EC 2.1: Editar tipología satisfactoriamente.	El usuario selecciona la tipología que desea modificar en el sistema, y presiona la opción "editar". El sistema carga los	VideoWeb 2.0/Estructura/Tipología/Opción editar/Editar todos los datos de la tipología/ Presiona el botón	El sistema modifica la tipología, muestra un mensaje indicando que la tipología ha sido modificada	El sistema modifica la tipología, muestra un mensaje indicando que la tipología ha sido modificada satisfactoriamente y actualiza la

		datos de la tipología para su posterior modificación por parte del usuario. Al finalizar la modificación, el usuario presiona el botón "Guardar" y se muestra un mensaje indicando que los datos han sido modificados correctamente.	"Guardar".	satisfactoria mente y actualiza la lista de tipología.	lista de tipología.
	EC 2.2: Editar tipología fallida.	El usuario presiona el botón "Guardar" antes de completar todos los datos obligatorios el sistema muestra un mensaje de error indicando que existen campos obligatorios por llenar.	VideoWeb2.0 /Estructura/Ti pología/Opci ón editar/Editar todos los datos de la tipología/ Presiona el botón "Guardar".	El sistema muestra un mensaje indicando que existen campos obligatorios por llenar.	El sistema muestra un mensaje indicando que existen campos obligatorios por llenar.
SC 3: Eliminar Tipologías.	EC 3.1: Eliminar tipología satisfactori amente.	El usuario selecciona la tipología que desea eliminar y presiona el botón "eliminar". El sistema muestra un mensaje de	VideoWeb 2.0/Estructur a/Tipología/O pción eliminar/Elimi na todos los datos de la tipología/ Presiona el botón	El sistema muestra un mensaje de verificación, eliminar la tipología satisfactoria mente y actualiza la lista de	El sistema muestra un mensaje de verificación, eliminar la tipología satisfactoriamen te y actualiza la lista de tipología.

		confirmación de la acción, si el usuario presiona el botón "Eliminar" se eliminan la tipología y se muestra un mensaje indicando que se ha realizado la operación.	"Guardar".	tipología.	
	EC 3.2: Eliminar tipología fallida.	El usuario selecciona la tipología que desea eliminar y presiona el botón "eliminar". El sistema muestra un mensaje de confirmación de la acción, si el usuario presiona el botón "Cancelar" el sistema no elimina y muestra la lista de tipología.	VideoWeb 2.0/Estructura/Tipología/Opción eliminar/Elimina todos los datos de la tipología/ Presiona el botón "Guardar".	El sistema no elimina y muestra la lista de tipologías.	El sistema no elimina y muestra la lista de tipologías.

Tabla 2. Caso de prueba ¨Gestionar campos de tipologías¨

Caso de prueba: Gestionar campos tipologías.					
Nombre de la sección	**Escenarios de la sección**	**Descripción de la funcionalidad**	**Flujo central**	**Resultados Esperados**	**Resultados Obtenidos**
SC 1: Crear campos de Tipologías.	EC 1.1: Crear campo de la	El usuario puede crear un nuevo campo o seleccionar	VideoWeb 2.0/Estructura/Tipología/Opción	El sistema crea el campo de la tipología y	El sistema crea el campo de la tipología y muestra un

	tipología satisfactoriamente.	uno ya existente, personaliza los parámetros del campo y presiona el botón “Guardar”. Luego configura los metadatos del campo y presiona el botón “Guardar las opciones”. El sistema muestra un mensaje confirmando que el campo fue creado.	Adicionar Tipología/Inserta todos los datos de la tipología/ Presiona el botón “Guardar”.	muestra un mensaje de confirmación.	mensaje de confirmación.
	EC 1.2: Crear campo de la tipología fallida.	El usuario presiona el botón “Guardar” antes de completar todos los datos obligatorios el sistema muestra un mensaje de error indicando que existen campos obligatorios por llenar.	VideoWeb 2.0/Estructura/Tipología/Opción gestionar campos/Inserta todos los datos de los campos de la tipología/ Presiona el botón “Guardar”.	El sistema muestra un mensaje indicando que existen campos obligatorios por llenar.	El sistema muestra un mensaje indicando que existen campos obligatorios por llenar.
	EC 1.3: Crear campo de la tipología existente.	El usuario puede crear un nuevo campo con el nombre del sistema igual a uno ya existente y	VideoWeb 2.0/Estructura/Tipología/Opción gestionar campos/Inserta todos los	El sistema muestra un mensaje de error indicando que ya existe el campo.	El sistema muestra un mensaje de error indicando que ya existe el campo.

		presiona el botón “Guardar”. El sistema muestra un mensaje de error.	datos de los campos de la tipología/ Presiona el botón “Guardar”.		
SC 2: Editar campos de Tipologías.	EC 2.1: Editar campos de tipología satisfactoriamente.	El usuario selecciona el campo de la tipología que desea modificar en el sistema, modifica los parámetros del campo y presiona el botón “Guardar”. Luego modifica los metadatos del campo y presiona el botón “Guardar las opciones”. El sistema muestra un mensaje confirmando que el campo fue modificado.	VideoWeb 2.0/Estructura/Tipología/Opción editar/Editar todos los datos del campo/ Presiona el botón “Guardar”.	El sistema modifica los campos de la tipología, muestra un mensaje indicando que el campo ha sido modificado y actualiza la lista de campos de la tipología.	El sistema modifica los campos de la tipología, muestra un mensaje indicando que el campo ha sido modificado y actualiza la lista de campos de la tipología.
	EC 2.2: Editar campos de tipología fallido.	El usuario presiona el botón “Guardar” antes de completar todos los datos obligatorios el sistema muestra un	VideoWeb 2.0/Estructura/Tipología/Opción editar/Editar todos los datos del campo/ Presiona el botón	El sistema muestra un mensaje indicando que existen campos obligatorios por llenar.	El sistema muestra un mensaje indicando que existen campos obligatorios por llenar.

		mensaje de error indicando que existen campos obligatorios por llenar.	“Guardar”.		
	EC 2.3: Editar campo de la tipología existente.	El usuario modificar el campo con el nombre del sistema igual a uno ya existente y presiona el botón “Guardar”. El sistema muestra un mensaje de error.	VideoWeb 2.0/Estructura/Tipología/Opción editar/Editar todos los datos del campo/ Presiona el botón “Guardar”.	El sistema muestra un mensaje de error indicando que ya existe el campo.	El sistema muestra un mensaje de error indicando que ya existe el campo.
SC 3: Eliminar campo de Tipología.	EC 3.1: Eliminar campo de la tipología satisfactoriamente.	El usuario selecciona el campo que desea eliminar y presiona el botón “eliminar”. El sistema muestra un mensaje de confirmación de la acción, si el usuario presiona el botón “Eliminar” se eliminan el campo y se muestra un mensaje indicando que se ha realizado la operación.	VideoWeb 2.0/Estructura/Tipología/Opción eliminar/Elimina todos los datos del campo/ Presiona el botón “Guardar”.	El sistema muestra un mensaje de verificación, eliminar la tipología satisfactoriamente, actualiza la lista de tipología y muestra un mensaje de información indicando que el campo fue eliminado.	El sistema muestra un mensaje de verificación, eliminar la tipología satisfactoriamente, actualiza la lista de tipología y muestra un mensaje de información indicando que el campo fue eliminado.
	EC 3.2:	Si el usuario	VideoWeb	El sistema no	El sistema no

	Eliminar campo de la tipología fallida.	presiona el botón “Cancelar” el sistema no elimina y muestra la lista de campos.	2.0/Estructura/Tipología/Opción eliminar/Elimina todos los datos del campo/ Presiona el botón “Guardar”.	elimina y muestra la lista de tipologías.	elimina y muestra la lista de tipologías.

3.2.3 Resultados de las pruebas Caja Negra

Para la realización de la pruebas de caja negra, fueron probados dos casos de uso, los que representan el cien por ciento del total. Para cada caso de uso se tuvo en cuenta los diferentes escenarios que pudieran existir, los cuales implican cada una de las posibles interacciones del usuario o combinaciones de situaciones posibles con el fin de evaluar el comportamiento del sistema ante cada funcionalidad. El resultado de cada prueba coincide con el resultado esperado lo que demuestra la correcta implementación del módulo.

3.3 Conclusiones parciales

Al concluir este capítulo se logró la validación de las funcionalidades del módulo a partir de la realización de pruebas de integración y de caja negra. La integración fue probada con dos módulos de la plataforma VideoWeb, siendo ajustada la implementación del módulo de acuerdo a las no conformidades encontradas. Las pruebas de caja negra se realizaron a partir de los casos de prueba que permitieron explorar cada uno de los escenarios del módulo. Se obtuvo como resultado, un módulo web de gestión de tipologías que responde correctamente según los objetivos planteados.

CONCLUSIONES

Con la realización de la investigación se logró dar cumplimiento a los diferentes objetivos propuestos:

- Se elaboró un estudio de los conceptos más significativos relacionados con la gestión de tipología que facilitó una mejor comprensión sobre el proceso de gestión de tipologías para los medios audiovisuales.
- Para desarrollar el módulo se caracterizaron las herramientas y tecnologías lo que permitió sentarlas bases para el posterior trabajo con las mismas.
- Los requisitos funcionales, los no funcionales y los casos de uso con sus descripciones anteriormente descritos en otras versiones, sirvieron de guía para definir las características del sistema.
- Para comprobar la calidad y correcto funcionamiento del sistema, se realizaron pruebas de integración y de caja negra, obteniendo resultados satisfactorios, demostrando el cumplimiento de los requisitos establecidos.
- El módulo implementado garantizará la adaptabilidad de las fichas de catalogación de medias para las personalizaciones de la plataforma VideoWeb 2.0.

RECOMENDACIONES

Al concluir el presente trabajo se recomienda:

- Implementar una funcionalidad que permita que el módulo de gestión de tipologías pueda atender simultáneamente el proceso de diferentes sistemas clientes. De esta manera se podrá dar respuesta a los requisitos de diferentes personalizaciones con un solo servicio desplegado.

REFERENCIA BIBLIOGRÁFICA

1. **Lozano, Mayra.** Nuevas Tecnologias. MANEJO DE LA INFORMACIÓN. [En línea] 15 de Junio de 2012. http://nuevastecnologias1724.blogspot.com/2012/06/manejo-de-informacion.html.

2. **Benki, Julio Ramiro Quevedo.**Componente para la gestión dinámica de tipologías para la catalogación de medias. La Habana : s.n., 2011.

3. Symfony.es. [En línea] 2013. http://www.symfony.es.

4. **Moya, Iván Hernández.**Migración del Módulo de Catalogación de materiales audiovisuales del producto Captura y catalogación de medias. Ciudad de La Habana : s.n., 2011.

5. *Definicion.org.* [En línea] http://www.definicion.org/catalogar.

6. Real Academia Española. *DICCIONARIO DE LA LENGUA ESPAÑOLA - Vigésima segunda edición.* [En línea] 2001. http://lema.rae.es/drae/.

7. **Lapuente, María Jesús Lamarca.** Hipertexto: El nuevo concepto de documento en la cultura de la imagen. [En línea] [Citado el: 19 de Febrero de 2013.] http://www.hipertexto.info/documentos/catalog_h.htm.

8. **Heredia, Daliana Noa.**Gestión de metadatos asociados a las tipologías documentales definidas en el Gestor de Documentos Administrativos eXcriba. La Habana : s.n., 2012.

9. Comunidad Astalaweb. *Diccionarios en la Red.* [En línea] 2007. http://diccionarios.astalaweb.com/Consultas/Diccionario%20de%20inform%C3%A1tica.asp.

10. WordPress.com. *Tproduccion multimedia's Blog.* [En línea] http://tproduccionmultimedia.wordpress.com/12-concepto-de-tipologias-multimedia/.

11. **Francisco, Yusef Hassan Montero Fernández y Jesús Martín.** No Solo Usabilidad journal. Sistemas de Clasificación de Información. [En línea] 14 de Febrero de 2004. http://www.nosolousabilidad.com/articulos/sistemas_clasificacion.htm. ISSN 1886-8592.

12. **Tedial.** Tedial. *Workflow optimisation and media asset management for the broadcast industry.* [En línea] 2012. http://www.tedial.com/products/tarsys/product-overview.

13. **Fernando J. Gómez, CONICET y Instituto de Matemática de Bahía Blanca (INMABB).** Catalis. [En línea] 2005. http://inmabb.criba.edu.ar/catalis/catalis.php?p=caract.

14. **Gastaminza, Félix del Valle.** Lenguajes documentales. *Los tesauros Facultad de Ciencias de la Información.* [En línea] 2004. http://pendientedemigracion.ucm.es/info/multidoc/prof/fvalle/tesauro.htm.

15. **Oficina de Desarrollo de Redes y Normas MARC de la Biblioteca del Congreso.** El Formato Bibliográfico MARC 21 LITE. [En línea] 10 de Abril de 2006. http://www.loc.gov/marc/bibliographic/litespa/introgen.htm.

16. American Library Association, Canadian Library Association, and the Chartered Institute of Library and Information Professionals.AARC. [En línea] 2006. www.aacr2.org/about.html.

17. ISID - Videoma. Gestión Inteligente de Activos Multimedia. [En línea] http://www.isid.com/Spanish/about/historia.htm..

18. **Guardia, Diego.** [En línea] 15 de Septiempre de 2008. http://guardiadiego.blogspot.com/2008/09/trancodificacin.html.

19. **Instituto Superior de Formación y Recursos en Red para el Profesorado.** Diseño de Materiales Multimedias Web 2.0. [En línea] 2008. http://www.ite.educacion.es/formacion/materiales/107/cd/video/video0103.html.

20. **Isaías Carrillo Pérez, Rodrigo Pérez González y Aureliano David Rodríguez Martín.** Metodologias de desarrollo del software. [En línea] 9 de Octubre de 2008. http://www.slideshare.net/geurquizo/metodologias-de-desarrollo-del-software.

21. **Vilma Quispe Carita, Dante Harry Huamantuco Solorzano Y Jose Luis Vargas Yupanqui.***Universidad Nacional del Altiplano - Facultad de Ingeniería Mecánica Eléctrica, Electrónica y Sistemas.* Peru : s.n., 2001.

22. **David Miguel Angeles Lara, Eduardo Santoyo Figueroa y Daniel Sierra Cruz.** Slideshare - Present yourself. *Exposicion Herramientas Automatizadas IPN UPIICSA - Visual Paradigm For Uml .* [En línea] 15 de Noviembre de 2007. [Citado el: 19 de Febrero de 2013.] http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml. .

23. Targetware. software.com.ar. [En línea] http://www.software.com.ar/visual-paradigm-para-uml.html.

24. **Cuerda, Xavier García.** Mosaic – Tecnologías y comunicación multimedia. [En línea] 29 de Noviembre de 2004. http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/.

25. **Ramírez, Luiyi Jiménez.**Aprende Drupal con Forcontu | Experto en Drupal 7 | Nivel inicial. s.l. : Forcontu S.L, 2011.

26. **Mateos, Juan Félix.** SlideShare. *Drupal CRTF ¨Las Acacias¨.* [En línea] Abril de 2011. http://www.slideshare.net/jecol59/drupal-7-2.

27. **Manjit Singh, leonard, Elijah Lynn.** Drupal. *An Introduction to Entities.* [En línea] 8 de Abril de 2013. http://drupal.org/node/1261744.

28. CAVSI- Computer Audio Video Systems Integrator. [En línea] http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/..

29. **Cesar Cartes, Omar Sepúlveda y Jorge Gajardo.** PDFOnline. Sistema Gestor de Bases de Datos PostgreSQL. *Instituto Profesional Santo Tomas Chillán.* [En línea] http://share.pdfonline.com/9363202103524565923 64f118373f9d8/trabajo%20postgresql.htm.

30. **Mora, Sergio Luján.**Programación de aplicaciones web: historia, principios básicos y clientes web. s.l. : Club Universitario, 202. 978-84-8454-206-3.

31. Internet, el instrumento esencial de la diplomacia del siglo XXI. [En línea] www.un.org/spanish/Depts/dpi/seminario/pdf/AnexoI.pdf.

32. **Lapuente, M. J. Lamarca.** Hipertexto: El nuevo concepto de documento en la cultura de la imagen. *Catalogación de hipertextos.* [En línea] 10 de Febrero de 2013. http://www.hipertexto.info/documentos/html.htm.

33. **Pérez, Javier Eguíluz.** Scribd. *Introducción a JavaScript.* [En línea] 7 de Junio de 2008. [Citado el: 11 de Febrero de 2013.] http://es.scribd.com/doc/8734596/Introduccion-Javascript-Libros-Web.

34. **Lerner, Michael.** Learn the Net. com. [En línea] 2013. http://www.learnthenet.com/spanish/glossary/scripts.htm.

35. **Aguila, Yoandry Pacheco.** monografias.com. [En línea] 8 de Febrero de 2007. [Citado el: 11 de Febrero de 2013.] http://www.monografias.com/trabajos43/ajax/ajax2.shtml.

36. LibroWeb. *Capítulo 1. Introducción a AJAX.* [En línea] http://librosweb.es/ajax/capitulo_1.html.

37. **Cerami., Ethan.**Web Services Essentials. Distributed Applications with XML-RPC, SOAP, UDDI & WSDL. O'Reilly First Edition. 2002. 0-596-00224-6.

38. **García, Luis.** Instituto Nacional de Tecnologías Educativas y Formación del Profesorado. *OBSERVATORIO TECNOLÓGICO. Ministerio de Educación, Cultura y Deporte.* [En línea] 17 de Enero de 2008. http://recursostic.educacion.es/observatorio/web/ca/software/software-general/548-luis-garcia.

39. **Blanco, Carlos.** CarlosBlanco.pro. *Entornos de Desarrollo Integrado (IDE´s) – Introducción.* [En línea] 8 de Abril de 2012. http://carlosblanco.pro/2012/04/entornos-desarrollo-integrado-introduccion/.

40. NetBeans. [En línea] 2013. https://netbeans.org/index_es.html.

41. Targetware. *software.com.ar.* [En línea] http://www.software.com.ar/visual-paradigm-para-uml.html..

42. Introducción a los Patrones (Diseño y Arquitectura). **Ochoa, Sergio.** 2005.

43. **Sebastián, Juan.** Comusoft.com – Seguridad de la información, software y tecnología. [En línea] 13 de Noviembre de 2010. http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas.

44. **Catalani, Exequiel A. Exequiel.** Exequiel Catalani – Desarrollo. [En línea] 20 de Agosto de 2007. http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/.

45. Microsoft Developer Network. [En línea] Marzo de 2013 de 2013. http://msdn.microsoft.com/es-es/library/bb972240.aspx.

46. **Grosso, Andrés.** Prácticas de Software - Experiencias sobre la Ingeniería y Management del Software. [En línea] 21 de Marzo de 2011. [Citado el: 14 de Marzo de 2013.] http://www.practicasdesoftware.com.ar/2011/03/patrones-grasp/.

47. Benneth Christiansson (Ed.), Mattias Forss,Ivar Hagen,Kent Hansson,Johan Jonasson,Mattias Jonasson,Fredrik Lott,Sara Olsson and Thomas Rosevall.*GoF Design Patterns - with examples using Java and UML2.* 2008.

48. definición.De. [En línea] 2013. http://definicion.de/modelo-de-datos/.

49. Sparx Systems Argentina - SOLUS S.A. Sparx Systems. [En línea] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

50. Kioskea.net. [En línea] http://es.kioskea.net/contents/282-tcp-ip.

51. UNAM. *Universidad Nacional Autónoma de México.* [En línea] http://www.vpnuniversitaria.unam.mx/usos.html.

52. **Kristal, Carlos David Marrero y Kiberley.** MeRinde. Metodología de la Red Nacional de integración y Desarrollo de Software Libre. *Centro Nacional de Tecnologías de Información.* [En línea] [Citado el: 6 de Abril de 2013.] http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.

53. **2, Departamento de Ingenieria de Software.** Entorno Virtual de Aprendizaje . [En línea] 25 de Abril de 2011. http://eva.uci.cu/file.php/160/Curso_2010-2011/Semana_9/Conferencia_7/Materiales_Basicos/Sobre_la_disciplina_de_Prueba.pdf.

BIBLIOGRÁFICA CONSULTADA

1. **Benki, Julio Ramiro Quevedo.**Componente para la gestión dinámica de tipologías para la catalogación de medias. La Habana : s.n., 2011.

2. Benneth Christiansson (Ed.), Mattias Forss,Ivar Hagen,Kent Hansson,Johan Jonasson,Mattias Jonasson,Fredrik Lott,Sara Olsson and Thomas Rosevall.*GoF Design Patterns - with examples using Java and UML2. .* 2008.

3. **González, Enrique Almeida Maldonado y Baby Ronald.***Sistema de catalogación de medias.* La Habana : s.n., 2009.

4. **Rodríguez, Iván Betancourt.**Desarrollo de un componente para la gestión de tipologías de archivos multimedia en aplicaciones web. La Habana : s.n., 2011.

5. **Moya, Iván Hernández.**Migración del Módulo de Catalogación de materiales audiovisuales del producto Captura y catalogación de medias. Ciudad de La Habana : s.n., 2011.

6. **James Rumbaugh, Ivar Jacobson y Grady Booch. James Rumbaugh Cupertino.***El Lenguaje Unificado de Modelado. Manual de Referencia.* California : s.n., 1998.

7. **Ramírez, Luiyi Jiménez.**Aprende Drupal con Forcontu | Experto en Drupal 7 | Nivel inicial. s.l. : Forcontu S.L, 2011.

8. **Mora, Sergio Luján.***Programación de aplicaciones web: historia, principios básicos y clientes web.* s.l. : Editorial Club Universitario, 2002. ISBN 978-84-8454-206-3.

9. **Vilma Quispe Carita, Dante Harry Huamantuco Solorzano y Jose Luis Vargas Yupanqui.***MONOGRAFIA - Metodología RUP (Rational Unified Process).* UNO-PERU : Universidad Nacional Del Altiplano - Facultad De Ingeniería Mecánica Eléctrica, Electrónica y Sistemas., 2011.

10. Benneth Christiansson (Ed.), Mattias Forss,Ivar Hagen,Kent Hansson,Johan Jonasson,Mattias Jonasson,Fredrik Lott,Sara Olsson, and Thomas Rosevall.*GoF Design Patterns -with examples using Java and UML2.* 2008.

11. **Glenford J. Myers, John Wiley & Sons Sons.***The Art of Software Testing.Second Edition.* 2004.

12. **Steven R. Rakitin, Artech House.**Software Verification and Validation for Practitioners and Managers. Second Edition Edition E. 2001.

13. **Sommerville, Ian.**Ingeniería de software. Séptima edición. ISBN:84-7829-074-5.

14. **Alvaréz, Adrían Pérez.***Guía básica de Drupal 7.* Cuba : s.n., 2012.

15. **Mora, Sergio juján.**Programacion de aplicaciones web: historia, principios básicos y clientes web. s.l. : Club Universitario , 2002. ISBN:84-8454-206-8.

ANEXOS

Anexo 1. Requisitos Funcionales y no Funcionales

Requisitos Funcionales (RF)

RF1. Adicionar tipología.

Descripción: El sistema debe permitir a los usuarios con permisos adicionar una nueva tipología de archivo multimedia y definir los datos que va a incluir dicha tipología. Las tipologías definen los tipos de contenido que gestiona el sistema y los datos de descripción que poseerán, por ejemplo: La tipología Películas, tendrá los campos Director, Año, Género, etc.

Entrada: Nombre, descripción y si es tipología o no.

Salida: Mensaje de notificación.

RF2. Editar tipología.

Descripción: El sistema debe permitir editar los datos de una tipología ya existente.

Entrada: Nombre, descripción y si es tipología o no.

Salida: Mensaje de notificación.

RF3. Listar tipología.

Descripción: El sistema debe permitir listar todas las tipologías que existan en el sistema.

Salida: Lista de tipologías.

RF4. Eliminar tipología.

Descripción: El sistema debe permitir a los usuarios con permisos eliminar una tipología de archivo multimedia.

Entrada: Identificador de la tipología que se desea eliminar.

Salida: Lista de tipologías actualizada.

RF5. Adicionar campos a una tipología existente.

Descripción: El sistema debe permitir a los usuarios con permisos adicionar campos a las tipologías existentes. Estos campos serán usados para almacenar los datos asociados a las tipologías. Por ejemplo de la tipología Películas, serán útiles los campos: Director, Año, Género, etc.

Entrada: Nombre del campo, tipo de dato del campo, descripción y longitud máxima de los textos.

Salida: Mensaje de notificación.

RF6. Editar campos de tipología.

Descripción: El sistema debe permitir a los usuarios con permisos editar los valores de los campos adicionados de las tipologías existentes.

Entrada: Nombre del campo, tipo de dato del campo, descripción y longitud máxima de los textos.

Salida: Mensaje de notificación.

RF7. Listar campos de tipología.

Descripción: El sistema debe permitir listar los campos de una tipología seleccionada.

Entrada: Identificador de tipología seleccionada.

Salida: Lista de campos que pertenecen a una tipología.

RF8. Eliminar campos de tipología.

Descripción: El sistema debe permitir a los usuarios con permisos eliminar campos a las tipologías existentes.

Entrada: Identificador del campo de tipología.

Salida: Mensaje de notificación.

Requisitos no Funcionales (RNF)

RnF1. Usabilidad

1.1 Tipo de usuario final

Luego de un adiestramiento, la aplicación debe ser manejada de forma fácil por los administradores del sistema.

1.2 Tipo de aplicación informática

Aplicación Informática: WEB.

1.3 Finalidad

Proveer a un sistema de gestión de medias de las funcionalidades básicas para la gestión de tipologías de contenido audiovisual, que permitirá una efectiva catalogación y recuperación de este tipo de contenido.

1.4 Ambiente

Estos requisitos están en función de la dimensión del entorno donde se despliegue, estos son los requisitos mínimos con los que deberá contar la solución que se propone:

Estaciones clientes: Memoria RAM: 1 GB, Procesador: Core 2 Duo, Monitor: 17 ".
Servidor web: Arquitectura 64 bits (x64), Procesador: 1 QuadCore, Memoria RAM: 4 GB.
Servidor de BD: Arquitectura 64 bits (x64), Procesador: 1 QuadCore, Memoria RAM: 4 GB.

RnF2. Soporte

Estaciones clientes: Todas las estaciones clientes tienen que tener interfaz de red, cualquier Sistema Operativo, cualquier navegador excepto Internet Explorer, recomendado navegador Mozilla Firefox 10 o superior.
Servidores: En todos el Sistema Operativo Ubuntu 12.04. Servidor de BD: PostgreSQL 9.1. Servidor web: Drupal 7, apache 2.

RnF3. Interfaz

3.1. Interfaces de usuario del módulo de gestión de tipologías.

Las interfaces gráficas implementadas para el modulo deben concebirse con un ambiente sencillo y de navegación fácil para el usuario. Los colores serán convenientemente utilizados dada la funcionalidad y objetivo del sistema, siendo claros en la mayor parte de la aplicación logrando una vista agradable a los usuarios y resaltando con otras tonalidades los mensajes de interacción de los que dependen las funcionalidades críticas.

3.2. Interfaces de Comunicación del módulo de gestión de tipologías.

El sistema requiere interactuar con el módulo de servicios de catalogación y el resto de los sistemas del departamento Señales Digitales que lo requieran, por lo que debe proveerse una capa de servicios para exponer las principales funcionalidades del mismo. La capa de servicios será implementada haciendo uso de los protocolos SOAP y XML-RPC.

Anexo 2. Modelo del sistema

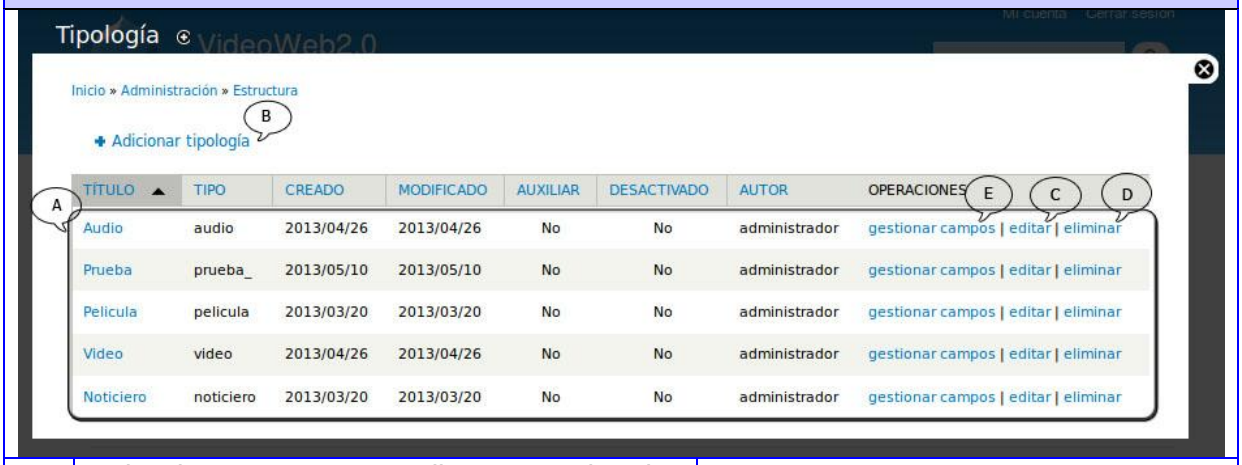
2.1 Diagrama de Casos de Usos

Figura 7. Diagrama de casos de usos

2.2 Descripción de Caso de Uso

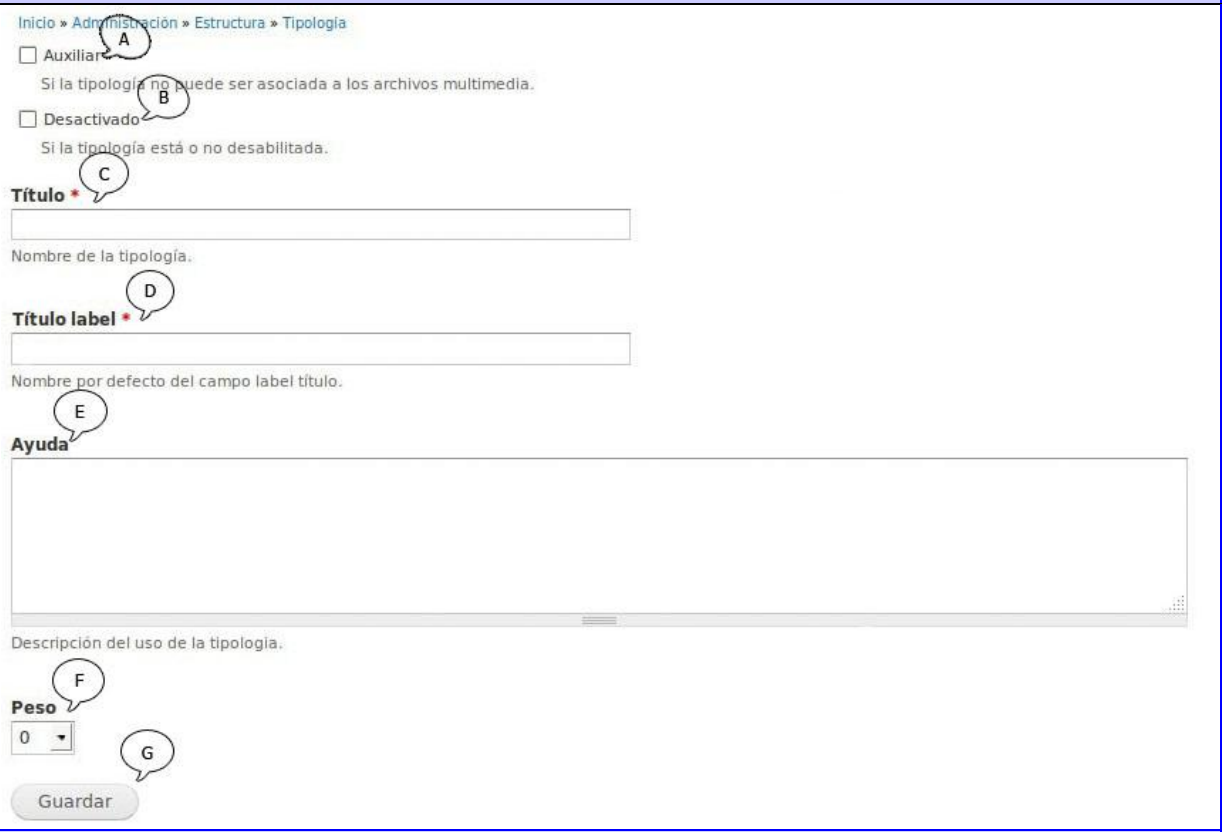
Tabla 3. Descripción de Caso de Uso: Gestionar tipologías

Objetivo	Adicionar, listar, editar y eliminar tipologías en el sistema.	
Actores	Administrador: (Inicia) Adiciona, edita y elimina tipologías.	
Resumen	El caso de uso inicia mostrando una lista de tipologías existentes en el sistema, el usuario selecciona la opción de adicionar, editar o eliminar las tipologías, para ello introduce los datos necesarios para realizar estas operaciones. El caso de uso termina cuando se adicione, edite o se elimine una tipología.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario está autenticado y tiene permisos de administrador.	
Postcondiciones	Tipología adicionada, editada o eliminada.	
Flujo de eventos		
Flujo básico: Gestionar tipologías.		
	Actor	Sistema
	Ejecuta el módulo de tipologías.	
		Muestra una lista de las tipologías existentes (A) con las acciones que se pueden realizar sobre cada una de

		ellas (adicionar, editar, eliminar y gestionar sus campos). (Ver interfaz 1).
Interfaz 1		
		
	Selecciona con un clic una de las operaciones siguientes: Adicionar tipología. (B) Editar tipología. Ver sección 1 “Editar tipología “. (C) Eliminar tipología. Ver sección 2 “Eliminar tipología”. (D) Gestionar campos de la tipología. Ver la DCU correspondiente. (E)	
		Muestra una interfaz para insertar los datos necesarios para una tipología.
	Inserta los datos teniendo en cuenta los siguientes parámetros: Auxiliar. (A) Desactivado. (B) Titulo (obligatorio). (C)	

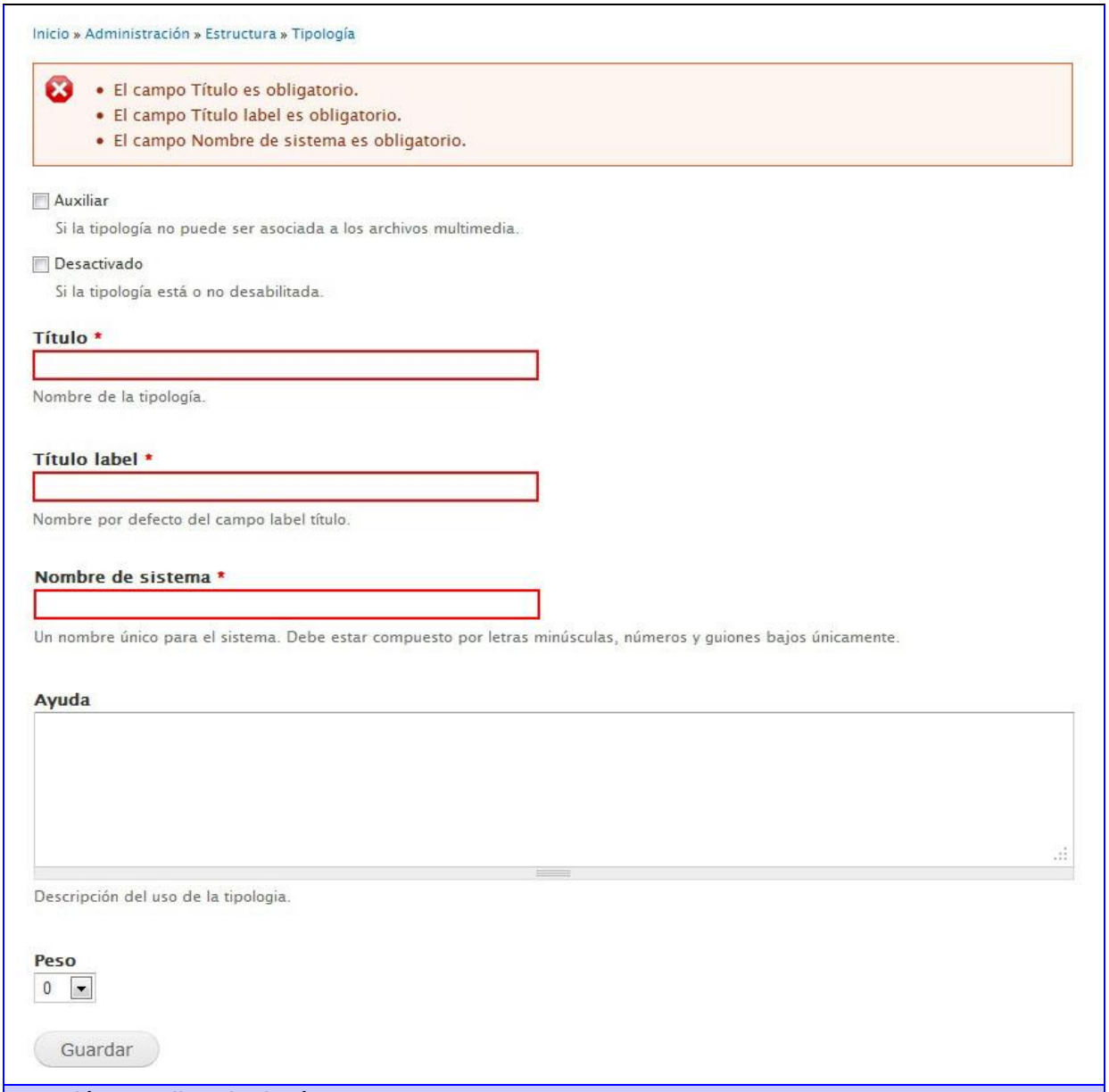
	Titulo label (obligatorio). (D) Ayuda. (E) Peso. (F) Selecciona la opción “Guardar” (G) para guardar los cambios (Ver Interfaz 2).	
		Valida que no existan campos obligatorios vacíos.
		Inserta los datos en la base de datos.
		Muestra un mensaje de información. Termina el CU (Ver interfaz 3).

Interfaz 2



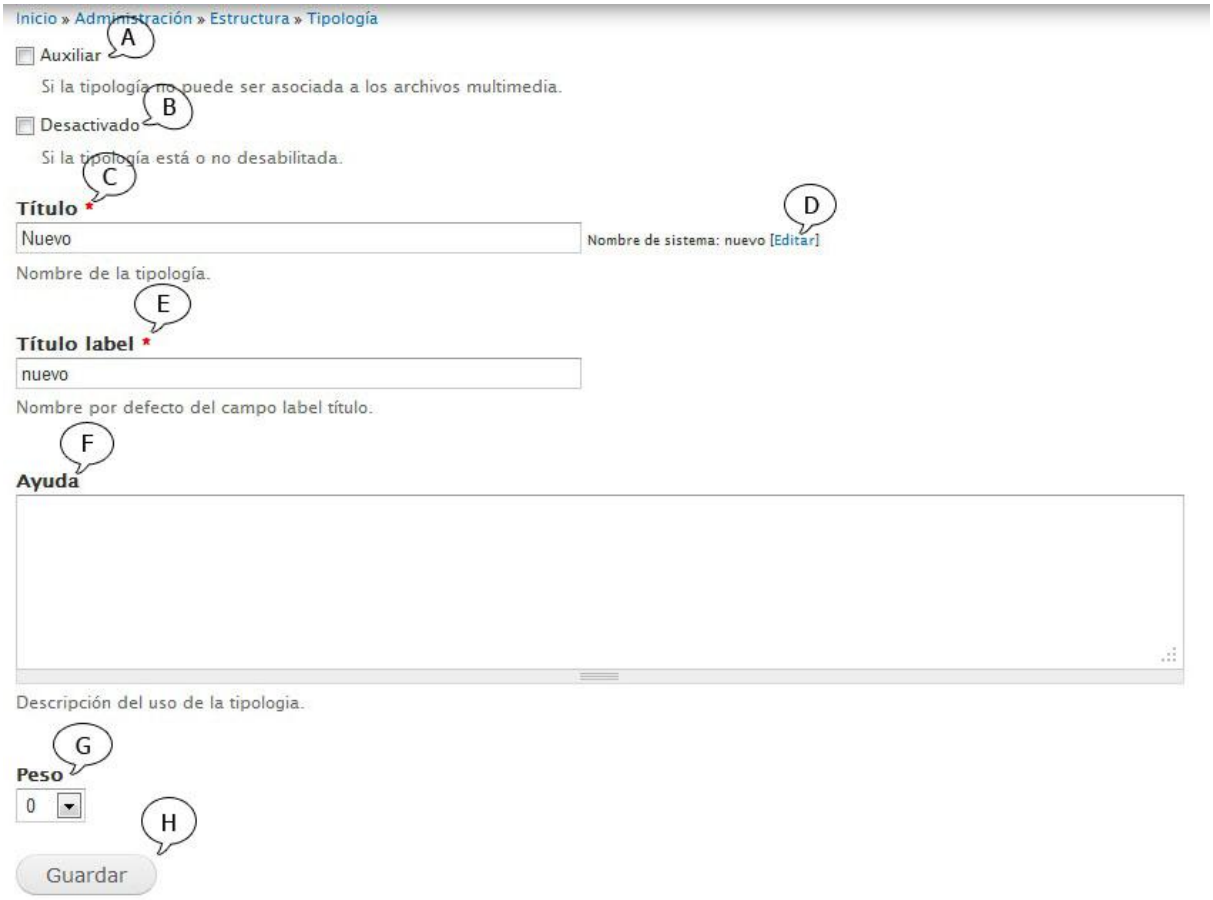
Interfaz 3

Flujos alternos		
6a Evento: Campo obligatorio vacío.		
	Actor	Sistema
		Muestra un mensaje de error indicando que hay campos vacíos (ver interfaz 4). Regresa al paso 5 del flujo básico.
Interfaz 4		



Sección 1: Editar tipología		
Flujo básico: Editar tipología.		
	Actor	Sistema
	Selecciona con un clic la tipología que desea editar.	
		Busca los datos de la tipología

		seleccionada en la base de datos y carga la interfaz con los datos guardados de dicha tipología.
	Modifica los parámetros deseados: Auxiliar. (A) Desactivado. (B) Titulo. (obligatorio). (C) Nombre del sistema (Automático, Modificación opcional). (D) Titulo label. (obligatorio). (E) Ayuda. (F) Peso. (G) Selecciona la opción “Guardar” (H) para guardar los cambios (Ver Interfaz 5).	
		Valida que no existan campos obligatorios vacíos.
		Valida que los datos estén correctos.
		Valida que el nombre del sistema no exista en la base de datos.
		Busca en la base de datos la tipología y modifica los datos que fueron cambiados.
		Muestra un mensaje indicando que la tipología fue modificada correctamente (Ver interfaz 7). Termina el CU.
Interfaz 5		



Interfaz 6

Flujos alternos

5a Evento: Campo obligatorio vacío.

	Actor	Sistema
5a.		Muestra un mensaje de error indicando que hay campos vacíos (ver interfaz 4). Regresa al paso 4 del flujo básico.

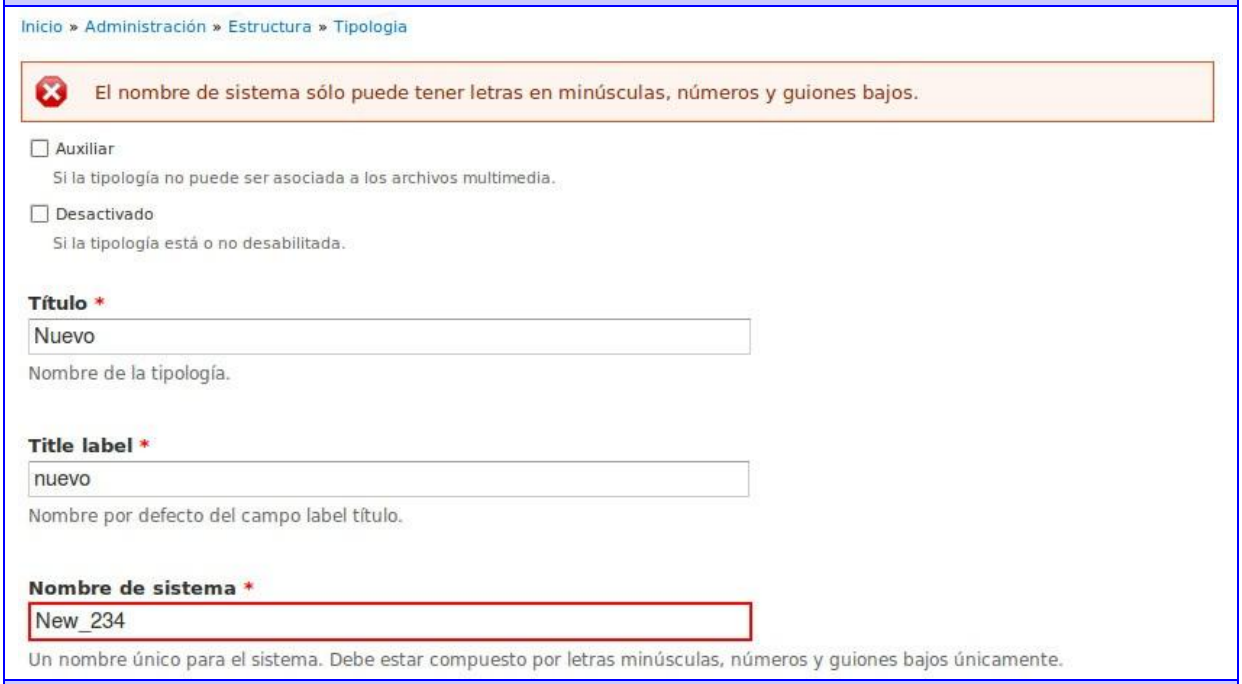
Flujos alternos

6a Evento: Datos incorrectos.

	Actor	Sistema

6a.		Muestra un mensaje de error indicando existen datos incorrectos (Ver interfaz 7). Regresa al paso 4 del flujo básico.

Interfaz 7



Flujos alternos

7a Evento: Nombre del sistema existente en la base de datos.

	Actor	Sistema
		Muestra un mensaje de error indicando que ese nombre ya existe (Ver interfaz 8). Regresa al paso 4 del flujo básico.

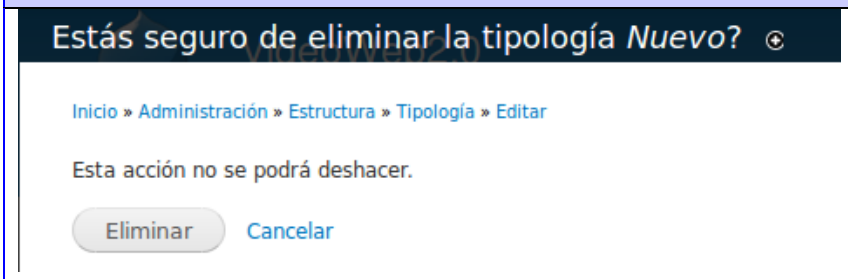
Interfaz 8

Sección 2: Eliminar tipología.

Flujo básico: Eliminar tipología.

	Actor	Sistema
	Selecciona la tipología que desea eliminar.	
		Muestra un mensaje de confirmación para eliminar la tipología seleccionada. (Ver interfaz 8)

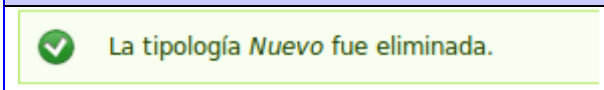
Interfaz 8



	Selecciona la opción “Eliminar”. (Ver Interfaz 8)	
		Busca la tipología seleccionada en la

		base de datos y la elimina del sistema.
		Muestra un mensaje de información (Ver interfaz 9). Termina el CU.

Interfaz 9



Flujos alternos

4a Evento: Selecciona la opción “Cancelar”.

	Actor	Sistema
4a.		Cancela la operación. Termina el CU.

Relaciones	CU Incluidos	No existen.
	CU Extendidos	Gestionar Campos de Tipología. Ver CU Gestionar Campos de Tipología.
Requisitos funcionales	RF1, RF2, RF3, RF4.	
Asuntos pendientes		