

Universidad de las Ciencias Informáticas

Facultad 6



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

**Título: Módulo para la Gestión de Tipologías Multimedia en
PRIMICIA 2.0**

Autor: Ivan Francisco Hoed Salazar

Tutor: Ing. Adrianet Ramos Velazco

La Habana, julio de 2016

“Año 58 de la Revolución”



“La nueva información hace posible las nuevas ideas”

Zig Ziglar, escritor.

Declaración de Autoría

Declaro ser autor de la presente tesis Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0 y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma.

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

Ivan Francisco Hoed Salazar

Ing. Adrianet Ramos Velazco

Firma del autor

Firma del tutor

Datos de Contacto

Tutor: Ing. Adrianet Ramos Velazco

Centro de trabajo: Universidad de las Ciencias Informáticas

Título de la especialidad del graduado: Ingeniero en Ciencias Informáticas

Año de graduación: 2014

Institución en la que se graduó: Universidad de las Ciencias Informáticas

Correo electrónico: arvelazco@uci.cu

Teléfono particular: 837 2127

Agradecimientos

A mi familia por educarme todos estos años y apoyarme en las decisiones que he tomado en mi vida, gracias a mi mamá, papá, abuela y hermana por estar siempre ahí, soportando junto a mí, los momentos difíciles.

A mis compañeros de brigada y a los del apartamento 107107, compañeros desde el principio de la vida en la universidad Julio, Darayne, Flavia, Yurian, Ivonne, Alfredo, Rojo, Dayron y Yaisel, y el resto de los compañeros de la brigada y del edificio con los que he creado importantes recuerdos en la universidad y que han ido para adelante conmigo.

A los miembros del proyecto Plataforma de Televisión Informativa PRIMICIA, a Yanary la jefa de proyecto por no rendirse conmigo, y mi tutora Adrianet por el apoyo que me brindó en el desarrollo del trabajo de diploma. También al resto por los consejos dados durante los cortes de proyecto.

Resumen

Las organizaciones que trabajan con archivos audiovisuales o multimedia los almacenan en mediatecas, para su posterior uso. La catalogación de medias es un proceso importante para la organización de los archivos que son almacenados. Parte del proceso de catalogación es la clasificación de las medias en tipologías. La gestión de tipologías es el proceso de definir los tipos en que serán clasificadas las medias y los atributos que poseen las medias de cada clase. Los atributos de las medias se pueden encontrar dentro de los archivos, en los metadatos, que son adicionados durante el proceso de creación o edición del archivo.

PRIMICIA en su versión 2.0 cuenta con un módulo para la gestión de medias, que solo son clasificadas en audio, imagen y video, y se define un grupo limitado de atributos. No son definidos otros tipos y atributos para la descripción de las medias, además no se obtienen los que ofrecen los metadatos relacionados a los archivos, lo cual influye en el proceso de catalogación en PRIMICIA. Existen aplicaciones que permiten la gestión de tipologías y otras que manejan los metadatos de las medias, pero no realizan las dos funciones por lo que se realizó la siguiente investigación, cuyo resultado es un módulo para la gestión de tipologías que permite la obtención de metadatos, utilizando para su desarrollo el *framework* para PHP Symfony 2.3 y la biblioteca para la extracción de metadatos getID3. Para verificar su correcto funcionamiento se realizaron pruebas de caja negra e integración.

Palabras clave: archivos de media; catalogación de medias; gestión de medias; metadatos; tipología

Abstract

The organizations that work with audiovisual or multimedia files, stores it in a media library for use it later. The multimedia cataloging is an important process for organizing the files that are stored. Part of the process of cataloging is the classification of multimedia in typologies. Typologies management is the process of defining the types that a multimedia will be classified and attributes that have the multimedia of each class. The multimedia attributes can be found inside the files, in metadata, which are added during the file creation or editing process.

PRIMICIA in version 2.0 has a module for multimedia management, which are only classified in audio, image and video, and it defines a limited group of attributes. are not defined other types and attributes for describing the multimedia, also, attributes that provide metadata related to files are not obtained, which influences the process of cataloging in PRIMICIA. There are applications that allow typologies management and others that handle multimedia metadata, but do not perform the two functions therefore the following research was done, the result is a module for managing typologies that allows obtaining metadata, and was development using the PHP framework Symfony 2.3 and library for metadata extraction getID3. To verify proper operation black box and integration tests were performed.

Keywords: *metadata; multimedia cataloging; multimedia files; multimedia management; typology*

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	6
1.1 Términos relacionados con el dominio del problema	6
1.1.1 Metadatos	6
1.1.2 Gestión de medias	6
1.1.3 Catalogación de medias.....	7
1.1.4 Gestión de tipologías	7
1.2 Soluciones existentes.....	8
1.2.1 Extracción de metadatos en plataformas DAM.....	8
1.2.2 TeVeo Plus v 1.0.....	9
1.2.3 Módulo de Gestión de Tipologías para plataforma VideoWeb 2.0	9
1.3 Selección de metodología, tecnologías y herramientas	10
1.3.1 Metodología de desarrollo AUP-UCI	10
1.3.2 Lenguaje Unificado de Modelado (UML)	11
1.3.3 Herramienta CASE Visual Paradigm v8.0.....	12
1.3.4 Sistema Gestor de Bases de Datos PostgreSQL 9.4.1	12
1.3.5 Lenguajes de programación.....	12
1.3.5.1 HTML 5	13
1.3.5.2 CCS 3.....	13
1.3.5.3 JavaScript.....	13
1.3.5.4 PHP 5.3.....	14
1.3.6 Framework de desarrollo.....	14
1.3.6.1 Symfony 2.3.7	15
1.3.6.2 jQuery 1.9.4.....	15
1.3.6.3 jQueryUI 1.10.3	15

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

1.3.6.4 Boostrap 2.3.2	15
1.3.7 Biblioteca getID3 1.9.12	16
1.3.7 Entorno de Desarrollo Integrado PHPStorm 9.0.....	16
1.4 Conclusiones parciales.....	16
Capítulo 2: Características y diseño del módulo	17
2.1 Modelo de dominio	17
2.1.1 Descripción del modelo de dominio.....	18
2.1.2 Descripción de las clases del modelo de dominio	18
2.2 Requisitos	18
2.2.1 Requisitos funcionales	18
2.2.2 Requisitos no funcionales	20
2.3 Modelo del Sistema.....	21
2.3.1 Especificación de casos de uso	22
2.3.1.1 Caso de uso Gestionar tipología.....	22
2.3.1.2 Caso de uso Gestionar campos.....	26
2.4 Patrón arquitectónico.....	29
2.4.1 Modelo Vista Controlador MVC	29
2.5 Modelo de diseño	30
2.5.1 Patrones de diseño	30
2.5.2 Diagramas de clase del diseño	31
2.5.2.1 Caso de uso Gestionar tipología.....	31
2.5.2.2 Caso de uso Gestionar campos.....	32
2.6 Modelo de datos.....	32
2.7 Conclusiones parciales.....	33
Capítulo 3: Implementación y validación de la solución.....	34
3.1 Modelo de implementación.....	34
3.3 Estándares de codificación.....	35

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

3.2 Modelo de despliegue	36
3.3 Pruebas de software.....	36
3.3.1 Pruebas de caja negra	36
3.3.2 Prueba de integración	38
3.4 Conclusiones parciales.....	38
Conclusiones Generales.....	39
Recomendaciones	40
Referencias Bibliográficas.....	41
Anexos.....	44

Índice de tablas

Tabla 1. Descripción de los actores del sistema	21
Tabla 2. Especificación del caso de uso Gestionar tipología.....	22
Tabla 3: Especificación del caso de uso Gestionar campos.....	26
Tabla 4. Caso de prueba gestionar tipología.....	37
Tabla 5. Descripción de las variables para el caso de prueba gestionar tipología.....	37
Tabla 6. Escenario editar tipología del caso de prueba gestionar tipología	44
Tabla 7. Escenario seleccionar campos de tipologías del caso de prueba gestionar tipología	44
Tabla 8. Escenario eliminar del caso de prueba gestionar tipología	45
Tabla 9. Escenario eliminar múltiples del caso de prueba gestionar tipología	45
Tabla 10. Variables para el caso de prueba gestionar campos de tipologías	46
Tabla 11. Sección insertar campo para el caso de prueba gestionar campos de tipologías	46
Tabla 12. Sección editar campo para el caso de prueba gestionar campos de tipologías	47
Tabla 13. Sección eliminar campo para el caso de prueba gestionar campos de tipologías	48
Tabla 13. Sección eliminar múltiples campos para el caso de prueba gestionar campos de tipologías.....	48

Índice de figuras

Figura 1. Modelo de dominio.....	17
Figura 2. Diagrama de Casos de Uso del Sistema.....	21
Figura 3. Diagrama de Clase del Diseño: Gestionar Tipología.....	31
Figura 4. Diagrama de Clases del Diseño: Gestionar campos de tipología.....	32
Figura 5. Diagrama de Entidad-Relación del Módulo de gestión de Tipologías.....	33
Figura 6. Diagrama de componentes para Gestionar Tipologías.....	34
Figura 7. Diagrama de componentes para Gestionar Campos.....	35
Figura 8. Ejemplo del uso del estándar lowerCamelCase.....	35
Figura 9. Diagrama de despliegue.....	36
Figura 10. Resultados de la iteraciones realizadas al módulo.....	38

Introducción

La comunicación es un tema relacionado a la sociedad y al ser humano como ser social, que intercambia con sus semejantes sus ideas, influyendo en el desarrollo social y personal. La obtención de información es uno de los resultados del intercambio comunicativo, esta permite la adquisición de conocimientos y la toma de decisiones por individuos o grupos. Los medios de comunicación como la prensa, la radio y la televisión son fuentes de información masiva para el público en general. En el caso de la televisión, se apoya en la combinación de texto, audio e imagen para la trasmisión de ideas al público. Las instituciones relacionadas a este medio de comunicación mantienen sus archivos audiovisuales como parte de su patrimonio, tanto los archivos como sus datos son importantes para la organización, ya que pueden ser utilizados cuando sea necesario.

En sus inicios los programas televisivos se grababan en largos carretes magnéticos y su información se almacenaba en papel. Los carretes eran almacenados en bibliotecas de audiovisuales y las tarjetas de índices eran usadas para localizar los recursos. Con la aparición de las primeras computadoras, las tarjetas físicas se comenzaron a guardar en la computadora. Con la evolución de los sistemas de cómputo se comenzó el almacenamiento de archivos de video en plataformas comerciales computarizadas (Engström, 2015).

Los archivos audiovisuales, también conocidos como archivos de media eran almacenados en varios formatos como son cintas magnéticas, casetes de audio o video y fotografías. Con el desarrollo de las tecnologías estos archivos fueron almacenados en formato de disco compacto (CD) y disco versátil digital (DVD). Los atributos de los archivos y los índices utilizados para su búsqueda eran almacenados en tarjetas físicas, y posteriormente en bases de datos digitales. Con el avance de las tecnologías de la información (TIC) y el desarrollo de los ordenadores y las redes, las medias son almacenadas actualmente en servidores y sus atributos son almacenados en bases de datos. Esto permite almacenar un gran número de medias sin que existan las restricciones de espacio de las mediatecas físicas y pueden ser accedidas desde lugares distantes, manejando las medias con un sistema de cómputo.

Las organizaciones que manejan las medias realizan la catalogación de medias para organizar los archivos antes de ser almacenados en la mediateca. Las medias son clasificadas en tipos, también conocidos como tipologías. Estas tipologías ordenan las

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

medias en grupos específicos y pueden ser utilizadas para determinar los atributos que se almacenarán de las medias. Estos tipos son definidos por los objetivos y materiales que maneja la organización.

Los atributos de las medias pueden ser obtenidos a partir de los metadatos, término relacionado inicialmente con la bibliotecología y ahora se ha extendido a los recursos digitales. Este término fue acuñado por Jack Myers en la década de los 60, para describir conjuntos de datos. La definición más popular es la de “datos que describen datos”, refiriéndose a que su función es almacenar las características de otros datos o recursos (Caplan, 1995).

Los archivos multimedia contienen esos metadatos, que *“...proporcionan información resumida sobre el recurso como: autor, título, fecha de creación, palabras claves, entre otros datos. Estos valores no se ven a simple vista y pueden estar disponibles en las propiedades del archivo. Los metadatos son incorporados al material durante su creación y/o posterior edición, como un mecanismo de descripción que facilita la recuperación y el acceso a los propios recursos. Varios reproductores y herramientas de edición de media: organizan, catalogan y localizan estos archivos utilizando sus metadatos”* (Osoria Palma, 2013).

Estos metadatos se pueden encontrar en forma de etiquetas dentro de los archivos de media y son ligados al archivo durante su proceso de creación. Algunos de estos atributos pueden ser consultados con las facilidades que proveen los sistemas operativos de visualizar las propiedades del archivo. También se pueden obtener en reproductores en el caso de audio y video, y en el caso de imagen, en programas de edición, visualización, o ambos.

A nivel profesional se utilizan plataformas que gestionan los procesos de gestión de medias y el manejo de sus metadatos, conocidos como gestores de activos de media (*MAM*, por sus siglas en inglés). Estas plataformas son utilizadas por organizaciones de audio y televisión para la gestión de sus recursos de media y el soporte a los sistemas de transmisión de sus canales.

La Plataforma de Televisión Informativa PRIMICIA es desarrollada en el Centro de Geoinformática y Señales Digitales (GEYSED), de la Universidad de las Ciencias Informáticas (UCI), como parte de los productos de señales digitales que se comercializan. Esta plataforma permite la administración y transmisión de un canal de

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

noticias, manejando los recursos que son utilizados en la creación de la programación que se transmite. Cuenta con un Módulo de Gestión de Medias con el que se manejan los recursos de media.

En este módulo los archivos son clasificados en audio, imagen o video, de cada uno de ellos se obtienen los datos título, autor, género y palabras claves, en el caso de los videos se obtiene además su duración. El resto de los datos que poseen las medias son descartados, se pierde información que puede ser utilizada para la descripción e identificación de la media, o en los procesos de gestión como son la búsqueda y recuperación de archivos o sus datos. Los resultados obtenidos al realizar una búsqueda en PRIMICIA 2.0 pueden devolver un gran número de medias, debido a que en cada grupo definido por las clasificaciones de audio, imagen o video, existe un número considerable de medias. Las medias pueden ser clasificadas en grupos más atómicos acorde a otras tipologías que se definan en la organización donde se despliegue la plataforma, estas clasificaciones permiten restringir las búsquedas y obtener una respuesta con un número de medias más reducido.

Por lo antes mencionado, se determina como **problema de la investigación**: ¿Cómo contribuir a la catalogación y búsqueda de archivos multimedia en PRIMICIA 2.0?

Se define como **objeto de estudio** el proceso de catalogación de medias y como **campo de acción** el proceso de gestión de tipologías en el Módulo de Gestión de Medias de PRIMICIA 2.0.

Se plantea como **objetivo general** desarrollar un módulo que permita la gestión de tipologías de archivos multimedia, y el uso de los metadatos relacionados a las medias para facilitar los procesos de clasificación y obtención de datos de las medias.

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

- ¿Qué herramientas y tecnologías existen en la actualidad para la gestión de tipologías y el manejo de metadatos?
- ¿Cuáles son las características que poseen esas herramientas y tecnologías?
- ¿Cómo influye la gestión de tipologías y los metadatos en la descripción de las medias y los procesos de búsqueda?

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

Para dar respuesta a las preguntas formuladas se proponen las siguientes tareas:

1. Caracterizar las soluciones que realizan gestión de tipologías y el manejo de metadatos.
2. Selección de las herramientas y tecnologías que se utilizarán en el desarrollo del módulo.
3. Generar los artefactos y documentos definidos por la metodología de desarrollo seleccionada.
4. Implementación del módulo de gestión de tipologías.
5. Realización de pruebas al módulo desarrollado.

Para el cumplimiento de las tareas se utilizan los siguientes métodos científicos:

Métodos teóricos:

- Analítico-Sintético:

Este método divide en partes el todo que se estudia con el objeto de analizar cada parte por separado, se reúnen sobre la base de los resultados del análisis las partes en una nueva totalidad (Barchini, 2005 a). Se utiliza para estudiar diferentes contenidos relacionados a los procesos de gestión de medias y manejo de metadatos, para crear los artefactos necesarios para dar solución al problema.

- Histórico-Lógico:

Este método se basa en el estudio histórico del fenómeno, observando la lógica de su desarrollo, explicando la historia de su desarrollo (Hernández León, y otros, 2011). Se utiliza en la descripción de la evolución del proceso de almacenamiento de los archivos audiovisuales, y los procesos para la indización y búsqueda de estos archivos.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

La investigación está conformada por tres capítulos, que son descritos a continuación:

- **Capítulo 1: Fundamentación teórica.**

Son analizados términos relacionados con el tema que se investiga. Se analizan las funciones presentadas por soluciones existentes. Se seleccionan la metodología, herramientas y tecnologías que se utilizan en el desarrollo de la solución.

- **Capítulo 2: Características y diseño del módulo.**

Se especifican los requisitos funcionales y los no funcionales, y sus descripciones. Se identifican los actores y casos de uso para especificar los requisitos del sistema, usados para el diseño, implementación y pruebas de este. Se muestran los diagramas necesarios para entender los procesos que se efectúan en el módulo.

- **Capítulo 3: Implementación y validación de la solución.**

Se describe cómo queda implementado el módulo y cómo se realiza el despliegue del módulo en la plataforma. Se presenta el estándar de codificación utilizado para dar uniformidad al código creado. Se presentan las pruebas realizadas al módulo para verificar su funcionamiento y los resultados que se obtienen de estas.

Capítulo 1: Fundamentación Teórica

En este capítulo son analizados términos relacionados con el tema que se investiga. Se analizan las funciones presentadas por soluciones existentes. Se seleccionan la metodología, herramientas y tecnologías que se utilizan en el desarrollo de la solución.

1.1 Términos relacionados con el dominio del problema

El estudio de los siguientes términos relacionados al tema permite tener conocimientos acerca del objeto de estudio, y los procesos relacionados a la gestión de tipologías y el manejo de metadatos.

1.1.1 Metadatos

Varios autores amplían el concepto de “dato sobre el dato”, al afirmar que los metadatos también deben incluir información sobre el contexto, contenido y control. De este modo, se alcanzan objetivos como describir, identificar y definir un recurso para recuperar, filtrar e informar sobre licenciamiento, condiciones de uso, autenticación y evaluación, preservación e interoperabilidad (Caplan, 1995).

“... se considera como metadato toda información o dato sobre un material multimedia que permita describir el contenido del mismo. La importancia de los metadatos para el trabajo con materiales audiovisuales, es que permiten la clasificación y descripción del contenido de los archivos...” (Rofes Hernández, 2011).

Los metadatos en caso de los archivos multimedia se pueden presentar de dos formas. La primera es en forma de etiquetas incluidas en el contenedor del archivo y la segunda es: almacenados en un archivo independiente del archivo multimedia (Betancourt Rodríguez, 2013).

Por lo anteriormente planteado se puede definir a los metadatos como información o datos sobre los atributos que posee un archivo multimedia, estos se pueden representar dentro de los propios archivos de media o en otro archivo independiente. Pueden ser utilizados en la clasificación, descripción y búsqueda de los archivos, además de ser utilizados para la autenticación y recuperación de recursos de media.

1.1.2 Gestión de medias

La gestión de medias es un término utilizado para varias tareas relacionadas a la postproducción. De manera general cualquier operación relacionada al procesamiento de medias es considerada gestión de media, como la captura, compresión, copia, reubicación o eliminación de archivos de media (Apple, 2011).

La gestión de medias es el proceso de administrar los archivos o recursos multimedia con que cuenta una organización o persona, así como los datos relacionados a estos archivos. Esta gestión permite el

almacenamiento, visualización, modificación o eliminación de las medias o los datos que se posean de ellas.

Existen diferentes aplicaciones y herramientas que permiten la gestión de archivos multimedia, como pueden ser reproductores de audio y video, o programas para la visualización de imágenes, que permiten ciertas características para la gestión de medias. También existen programas que permiten la organización de archivos, entre los cuales se encuentra plataformas para el trabajo con documentos conocidos como Gestores de Activos Digitales (DAM, por sus siglas en inglés), utilizadas a nivel profesional por empresas.

Los DAM que son utilizados para la manipulación de archivos multimedia y sus datos son conocidos como Gestores de Activos de Media, utilizado ampliamente por empresas que necesitan manejar sus recursos multimedia y sus datos, destacándose empresa relacionadas a los medios de comunicación como la radio y la televisión. *“Media Asset Management (MAM) tecnología que recupera contenidos específicos multimedia, sean analógicos o digitales”* (Marcos, 2004).

1.1.3 Catalogación de medias

Según la Real Academia de la Lengua Española la catalogación es una *“Relación ordenada en la que se incluyen o describen de forma individual libros, documentos, personas, objetos, etc., que están relacionados entre sí.”* (RAE, 2016 a). El proceso de catalogación de medias es el proceso de agrupar los diferentes archivos de medias según sus atributos comunes en diferentes clases.

La clasificación, descripción y catalogación de los audiovisuales permiten la recuperación, o sea la búsqueda de material audiovisual dentro de un conjunto de estos materiales, de manera veloz, sencilla y eficaz. (...) Las instituciones deben garantizar estas actividades y entenderlas como el punto donde se inicia la gestión de los documentos audiovisuales (Sosa Vázquez, y otros, 2013).

La catalogación de medias es un proceso relacionado a la gestión de archivos multimedia, cuya función fundamental es la organización y clasificación de las medias para facilitar los procesos de búsquedas que se realizan para recuperar los archivos o sus datos.

1.1.4 Gestión de tipologías

Según la Real Academia de la Lengua Española tipología es un *“Estudio y clasificación de tipos que se practica en diversas ciencias.”* (RAE, 2016 b), por lo que se puede definir como tipología multimedia al estudio y la clasificación de diferentes archivos multimedia en tipos o clases. Este proceso está relacionado a la clasificación que se realiza durante el desarrollo de la catalogación de medias.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

Es un término muy asociado a los metadatos, la gestión de tipologías es una clasificación de los tipos de multimedia basados en un estudio hecho previamente en el ambiente donde se difunden. Dependiendo de la tipología y el estándar de metadato será la información que se almacenará, la cual servirá posteriormente para su búsqueda y organización (Betancourt Rodríguez, 2013)

La gestión de tipologías es el proceso en el cual son definidos los diferentes tipos en que serán clasificados los archivos multimedia. Estas son definidas según el ambiente donde serán utilizadas las medias por la organización donde se realice la gestión de medias, que forma parte del proceso de catalogación que se realiza a los archivos. Se apoya en los metadatos para la definición y obtención de atributos de los archivos para su posterior búsqueda y recuperación.

1.2 Soluciones existentes

Para conocer en qué estado se encuentra el desarrollo de herramientas que permitan los procesos de gestión de tipologías y manejo de los metadatos de una media, dígame extracción y representación de los metadatos, se estudian las siguientes soluciones existentes.

1.2.1 Extracción de metadatos en plataformas DAM

Las plataformas DAM, o MAM, han desarrollado componentes para la extracción de los metadatos de los recursos que estas manejan, pero existe el inconveniente de que están integrados a las plataformas como son *Asset Bank* (Asset Bank, 2015), *Third Light* (Third Light, 2015), *Canto_Cumulus* (Canto, 2015) y *IMatch* (Photools, 2015) por lo que habría que adquirir la plataforma si se desea utilizar sus funcionalidades. También está el caso de la plataforma *Pim Core* (Pim Core, 2015) que es libre y cuenta con funciones para el manejo de metadatos.

En el caso del DAM *Asset Bank*, presenta los metadatos que posee la media además de permitir agregar nuevos campos de datos según las necesidades del usuario. Plantea atributos por defecto como son el título, descripción y palabras claves.

El DAM *Third Light* define los metadatos en primarios y secundarios, utilizándolos en el proceso de inserción de archivos para clasificarlos. En el caso de *Canto_Cumulus* permite la extracción de los metadatos, la creación de nuevos metadatos para la media, así como la reescritura de los metadatos ya existentes.

En el caso de *IMatch* está orientado al trabajo con imágenes, permite la transformación de los metadatos a texto plano, además de permitir agregar etiquetas a los archivos. El DAM *Pim Core* se utiliza para la gestión de archivos de audio y video, obtención y mantenimiento de los metadatos de estos tipos de archivos, además de permitir adicionar nuevas etiquetas, pero no trabaja con archivos de imagen.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

Ninguna de las plataformas mencionadas permite la gestión de tipologías, no está disponible la creación o modificación de las clasificaciones que utilizan. Además, los componentes se encuentran dentro de sus respectivas plataformas por lo que habría que adquirir la plataforma, debido a que la mayoría de estas plataformas son privativas se deben pagar licencias para su uso y no se puede estudiar su código fuente para adaptarlo. A pesar de no ser factibles para realizar la gestión de tipologías, del estudio de las plataformas se ha podido determinar cuáles son las formas de manejar los metadatos aplicadas por las plataformas DAM y MAM

1.2.2 TeVeo Plus v 1.0

TeVeo Plus es un sistema de gestión documental desarrollado por el Centro de Ideoinformática (CIDI), perteneciente a la UCI, el objetivo de la aplicación es facilitar la gestión y documentación de archivos de audio y video (Benítez, 2012). TeVeo Plus permite identificar, catalogar, describir y recuperar los archivos de media a través de metadatos que son incorporados durante el ciclo de vida del archivo de media, este sistema fue desarrollado en el ECM (Gestor de Contenidos Empresariales) Alfresco. Alfresco está escrito en el lenguaje de programación Java y presenta una arquitectura flexible para extensiones o integración con otras aplicaciones, además presenta un alto nivel de seguridad mediante la gestión de usuarios, grupos y roles. Como parte de la gestión de los metadatos de los archivos de audio y video fue desarrollado un módulo para la extracción y representación de los metadatos para el sistema TeVeo Plus (Osoria, 2013).

El sistema desarrollado muestra una forma de representación de los metadatos de audio y video tomando características fundamentales de cada tipo de archivo, así como estándares de metadatos que pueden ser utilizados. Esta solución no maneja los metadatos de imagen y no realiza una gestión de tipologías por lo que no da solución a la situación planteada.

1.2.3 Módulo de Gestión de Tipologías para plataforma VideoWeb 2.0

Este módulo fue desarrollado para la creación de fichas de catalogación de medias para las personalizaciones de la plataforma VideoWeb 2.0. Fue creado en el centro GEYSED, en el proyecto de Captura y Publicación de Medias a partir de propuestas planteadas por el proyecto de Captura y Catalogación de Medias y el proyecto VideoWeb. Está desarrollado en el CMS (Sistema de Gestión de Contenidos) Drupal 7, que permite la conexión a bases de datos de PostgreSQL y además en esa versión cuenta con el módulo *entity* que permite organizar la información ventajosa para la representación de las tipologías de medios audiovisuales. Este sistema permite la generación dinámica y la gestión de tipologías, así como la gestión de los atributos de cada tipología (Ricart Rodríguez, 2013).

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

El sistema desarrollado propone una solución web para gestionar las tipologías de archivos audiovisuales, esta solución puede ser tomada como una guía a la hora de diseñar el componente de gestión de tipologías para el producto PRIMICIA, ya que ejemplifica como debe estar estructurado un módulo de tipologías y los datos que son necesarios, pero no incluye la obtención de los atributos que contienen los metadatos por lo que no resuelve la necesidad de utilizar los metadatos que contienen las medias para describirlas.

Las soluciones estudiadas solo realizan el manejo de metadatos de los archivos medias o la gestión de las tipologías para las medias, pero no aseguran la realización de las dos operaciones, por lo que se decide crear un módulo que permita la gestión de las tipologías y sus campos, manejando los metadatos de los archivos multimedia para obtener los atributos que poseen.

1.3 Selección de metodología, tecnologías y herramientas

Para el desarrollo del módulo es necesario definir una metodología a seguir y las herramientas a utilizar en el diseño y desarrollo del sistema, así como, las tecnologías que están relacionadas. Se ha decidido adoptar la metodología definida por el proyecto PRIMICIA, y las herramientas para el diseño de aplicaciones y el desarrollo web que son utilizadas en el proyecto.

1.3.1 Metodología de desarrollo AUP-UCI

Es una variación de la metodología AUP (Proceso Unificado Ágil) desarrollada por la UCI, engloba dentro de sí tanto metodologías ágiles como robustas. Esta es la metodología aplicada en el proyecto Plataforma de Televisión Informativa PRIMICIA, por lo que se utiliza para el desarrollo del Módulo de Gestión de Tipologías.

La metodología AUP-UCI está dividida en tres fases (Sánchez, 2015 a):

- Inicio: En esta fase se lleva a cabo las actividades relacionadas a la planeación del proyecto.
- Ejecución: Durante el desarrollo de esta fase se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
- Cierre: En esta fase se analizan los resultados del proyecto como es su ejecución y se realiza el cierre formal del proyecto.

Esta metodología propone cuatro disciplinas ingenieriles (Sánchez, 2015 b):

1. Modelo: El objetivo de esa disciplina es entender el negocio de la organización, el problema de dominio que se aborda en el proyecto, y determinar una solución viable para resolver el problema de dominio. Agrupa los flujos Modelado de negocio, Requisitos y Análisis y Diseño.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

2. Implementación. El objetivo de esta disciplina es transformar el modelo en código ejecutable y realizar un nivel básico de las pruebas.
3. Prueba. El objeto de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad.
4. Despliegue. El objetivo de esta disciplina es la prestación y ejecución del sistema y que el mismo esté a disposición de los usuarios finales.

Para realizar la modelación del negocio y el sistema plantea cuatro escenarios, estos escenarios influyen en los tipos de artefactos ingenieriles que se crearán. Estos escenarios están dados por el ambiente en donde se desarrolla el producto (Sánchez, 2015 c).

Al ser PRIMICIA un producto personalizable, no es necesario incluir las responsabilidades del cliente que utiliza el sistema, por lo que no se modelan los casos de uso del negocio, se utiliza el escenario en el que se modela el dominio pero no los casos de uso del negocio. Para describir los conceptos relacionados al negocio se utiliza el modelo de dominio y para el diseño del sistema se utilizan los casos de uso del sistema.

Para el desarrollo del módulo se aplica la metodología AUP en su versión UCI. Creando los artefactos ingenieriles definidos en el proyecto como son el Modelo de Dominio, el Diagrama de Casos de Usos del Sistema, entre otros que permiten llevar a cabo el diseño, desarrollo y prueba de la aplicación.

1.3.2 Lenguaje Unificado de Modelado (UML)

UML es un lenguaje para especificar, visualizar, construir y documentar los artefactos de los sistemas de software, así como para el modelado del negocio y otros sistemas que no son de software (Larman, 2004).

El UML es, tal como su nombre lo indica, un lenguaje de modelado y no un método o un proceso. El UML está compuesto por una notación en el proceso de desarrollo o como parte de una metodología de diseño orientada a objetos (Sparks, 2012).

El UML soporta un conjunto rico en elementos de notación gráficos. Describe la notación para clases, componentes, nodos, actividades, flujos de trabajo, casos de uso, objetos, estados y cómo modelar la relación entre esos elementos. El UML también soporta la idea de extensiones personalizadas a través de elementos estereotipados (Sparks, 2012).

El UML provee beneficios significativos para los ingenieros de software y las organizaciones al ayudarles a construir modelos rigurosos, trazables y mantenibles, que soporten el ciclo de vida de desarrollo de software completo (Sparks, 2012).

1.3.3 Herramienta CASE Visual Paradigm v8.0

Las herramientas CASE son aplicaciones cuyo objetivo es facilitar el diseño del proyecto, el cálculo de los costos, la implementación de parte del código automáticamente, la documentación y detección de errores.

Visual Paradigm v8.0 es la herramienta CASE que se utiliza, pues soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (Visual Paradigm, 2013).

Visual Paradigm es una herramienta CASE multiplataforma que utiliza el lenguaje de modelado gráfico UML. El ambiente de Visual Paradigm proporciona medios intuitivos para realizar el análisis y el diseño orientados a objetos de un sistema, donde se pueden crear diagramas de UML, así como exportarlos a formatos de imagen y pdf. Esta herramienta brinda un entorno de desarrollo amigable para aquel que lo utilice, ya que permite organizar el trabajo a la comodidad de cada persona. Permite la realización de ingeniería directa e inversa, el modelado de procesos de negocio, de requisitos y de bases de datos. Por las facilidades que brinda se selecciona para modelar los diagramas que describen el sistema, su base de datos, entre otros.

1.3.4 Sistema Gestor de Bases de Datos PostgreSQL 9.4.1

Un sistema gestor de bases de datos comunica la aplicación con las bases de datos relacionadas a esta. Su objetivo es permitir al usuario trabajar con los datos que se almacenan sin tener necesidad de actuar directamente con la base de datos.

Para el desarrollo del sistema se selecciona PostgreSQL, que es un poderoso gestor de base de datos con una licencia de código abierto. Funciona en sistemas operativos de tipo Linux. Brinda una amplia gama de operaciones sobre bases de datos relacionales, además de soportar múltiples operaciones concurrentes. (PostgreSQL, 2015)

Se selecciona este gestor de base de datos debido a que la plataforma PRIMICIA 2.0 almacena los datos en una base de datos relacional que utiliza PostgreSQL, lo que facilitará la adición de nuevas tablas y relaciones aprovechando la capacidad de escalabilidad que brinda el gestor.

1.3.5 Lenguajes de programación

Un lenguaje de programación es el conjunto de reglas semánticas y sintácticas que permiten la creación de programas a través de instrucciones y operadores. Es el lenguaje que utilizan los desarrolladores para crear el código que necesita el programa para funcionar en una computadora. En la actualidad existen

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

varios lenguajes que se adaptan a las necesidades del entorno donde se despliegue el programa o la similitud al lenguaje natural que utilizan los seres humanos para comunicarse, los lenguajes pueden ser clasificados en alto o bajo nivel, como en estructurado u orientado a objetos (POO), o para entornos web o de escritorio (TechTerms, 2011).

1.3.5.1 HTML 5

HTML es un lenguaje de marcas para la descripción de documentos web (páginas web). Es un conjunto de etiquetas que describen diferentes contenidos del documento. HTML 5 incluye la creación de etiquetas multimedia para archivos de audio y de video, a diferencia de sus versiones anteriores (w3schools.com, 2014). Este lenguaje permite la estructuración de las páginas web que forman las vistas y las etiquetas multimedia influyen en el manejo de los archivos de media con los que se trabaja en la plataforma por lo que es el utilizado en el desarrollo del módulo.

1.3.5.2 CCS 3

El objetivo de este lenguaje es mejorar la presentación de la página **web**, se utiliza para cambiar los colores, las fuentes de los textos y otras características. Se guardan en archivos con extensión .css que son utilizados por los archivos HTML para darle estilo a lo que se presenta en el navegador.

CSS3 es el último estándar para CSS, es completamente compatible con versiones anteriores de CSS. CSS 3 está dividido en módulos, posee las especificaciones de los anteriores CSS, a las que se adicionan nuevos módulos (w3schools.com, 2013). Se utiliza este lenguaje para dar estilo a páginas pues fue diseñado para funcionar de conjunto con HTML5.

1.3.5.3 JavaScript

JavaScript es un lenguaje de programación comúnmente usado en el desarrollo web. Fue desarrollado originalmente por Netscape para agregar elementos dinámicos e interactivos a los sitios web (TechTerm, 2014)

Es un lenguaje de scripts del lado del cliente, lo que significa que el código fuente es procesado por el navegador web del cliente en lugar del servidor web. Eso significa que las funciones JavaScript puedan ejecutarse una vez que la página web ha sido cargada en el navegador sin necesidad de comunicarse con el servidor (TechTerm, 2014).

Al igual que lenguajes del lado del servidor, como PHP y ASP, el código puede ser insertado en cualquier parte del HTML de la página. Además puede ser referenciado en un archivo separado con extensión .JS, el cual puede ser visto en el navegador (TechTerm, 2014).

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

Por las facilidades que brinda JavaScript de crear funciones para la validación de los formularios y para representar de forma dinámica las páginas web se selecciona este lenguaje para el desarrollo del módulo.

1.3.5.4 PHP 5.3

PHP es un lenguaje de código abierto especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML (The PHP Group, 2006).

Lo que distingue a PHP de lenguajes del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no sabrá el código subyacente (The PHP Group, 2007), relacionado con la seguridad.

PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix, Microsoft Windows, MacOS X, RISC OS y probablemente otros más. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS (Servidor de Información de Internet) y muchos otros (The PHP Group, 2007).

Con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos, o una mezcla de ambas. Una de las características más destacables de PHP es su soporte para varias bases de datos (The PHP Group, 2007), como las de PostgreSQL.

Se utiliza el lenguaje para el desarrollo web del lado del servidor PHP 5.3 por las facilidades que brinda al ser multiplataforma y poder relacionarse con el gestor de base de datos PostgreSQL. Además de que es el lenguaje utilizado por frameworks como Symfony y la biblioteca de PHP *getID3* que se utilizan en la creación de la plataforma Primicia, es ventajoso para la integración del sistema con el módulo de Gestión de Medias existente.

1.3.6 Framework de desarrollo

Un *framework* (marco de trabajo) es una estructura conceptual y tecnológica de soporte definido, normalmente, con artefactos o módulos de software concretos, con base a la cual otro proyecto de software es más fácilmente organizado y desarrollado. Puede incluir soporte para programas, bibliotecas y un lenguaje interpretado, entre otras herramientas que contribuyan a desarrollar y unir los diferentes componentes de un proyecto. Permite a los desarrolladores utilizar funciones o códigos ya creados y probados, evitando que tengan que volver a programarlos (TechTerms, 2013).

1.3.6.1 Symfony 2.3.7

Symfony es un *framework* PHP para el desarrollo de sitios y aplicaciones web. Contiene un conjunto de componentes desacoplados y reutilizables, con los que son construidas las mejores aplicaciones PHP. Está soportado por una amplia comunidad que busca llevar el PHP al siguiente nivel. Su filosofía busca la aplicación y promoción del profesionalismo, buenas prácticas, estandarización y la interoperabilidad de las aplicaciones (Symfony, 2015).

Symfony 2 fue el *framework* utilizado en el desarrollo del sistema Primicia y sus módulos, por lo que es una ventaja utilizar este *framework* con vista a integrar el nuevo módulo la plataforma PRIMICIA. Además Symfony brinda ventajas, ya que es multiplataforma, se puede enlazar con bases de datos de PostgreSQL y se puede utilizar en varios IDE de desarrollo. Symfony está estructurado según la arquitectura MVC por lo que se puede realizar cambios en las plantillas de la vista o en las entidades que manejan la lógica del acceso a datos en las base de datos sin afectar al resto de los componentes de la plataforma. Por lo anteriormente planteado se selecciona este *framework* para el desarrollo del módulo.

1.3.6.2 jQuery 1.9.4

jQuery es una *framework* de JavaScript rápida, pequeña y con varias funciones. Trabaja haciendo recorridos del DOM y manipula sus elementos, maneja eventos, realiza animaciones y Ajax es mucho más simple, con un API fácil de usar que funciona a través de una multitud de navegadores. Con una combinación de versatilidad y capacidad de ampliación (jQuery Foundation, 2015).

1.3.6.3 jQueryUI 1.10.3

jQueryUI es un conjunto de interacciones para la interfaz de usuario, efectos, *widgets*¹ y temas creados sobre la biblioteca jQuery (jQuery Foundation, 2015). Esta biblioteca facilita el desarrollo de la interfaz gráfica de usuario de la aplicación y es la utilizada en el proyecto junto a *Bootstrap* para darle dinamismo las páginas web.

1.3.6.4 Bootstrap 2.3.2

Bootstrap es un *framework* de código abierto utilizado para dar estilo a las páginas web. Es fácil y eficiente, además que permite que los sitios web se adapten a diferentes dispositivos desde teléfonos móviles hasta computadoras de escritorio gracias a que es *web responsive*, capacidad de adaptar el contenido de un sitio web a la pantalla del dispositivo. Con esta biblioteca se tiene acceso a un grupo de elementos comunes de HTML, incluso personalizados, componentes CSS y plugins de jQuery (Bootstrap Core Team,

¹ Herramientas que permite la creación de interfaces gráficas de usuario

2015). Como se mencionaba anteriormente esta biblioteca es utilizada en el proyecto para el diseño de interfaz de usuario.

1.3.7 Biblioteca getID3 1.9.12

getID3 es una biblioteca desarrollada en lenguaje PHP que permite manejar los metadatos de diferentes formatos de archivos multimedia, en específico de los archivos de audio mp3 como lo indica su nombre (getID3, 2015).

Esta biblioteca es de código abierto bajo una licencia pública de GNU, aunque cuenta con una licencia comercial para su uso en software privado. En el caso de la versión 1.9.12 es necesario contar con una versión del lenguaje PHP superior a 5.3.0. (getID3, 2015).

Esta biblioteca permite el manejo de los metadatos de los archivos multimedia, además de soportar los formatos de archivo que son manejados por la plataforma PRIMICIA. Al estar desarrollado en el lenguaje PHP es fácil su integración con el framework de desarrollo Symfony.

1.3.7 Entorno de Desarrollo Integrado PHPStorm 9.0

Un entorno de desarrollo integrado (IDE) es una herramienta que los desarrolladores utilizan para crear programas informáticos. Combina múltiples herramientas de desarrollo en un solo programa, un IDE típico incluye un editor de código, un depurador y un compilador. La mayoría de los IDE proveen además una interfaz de proyecto que permite a los programadores seguir la pista de todos los archivos relacionados a un proyecto. Muchos soportan control de versiones también (TechTerm, 2015).

Se selecciona como IDE a PHP Storm ya que permite el trabajo con Symfony a través del uso de un plugin. Además permite la creación de archivos de JavaScript, CSS y plantillas Twig que utiliza Symfony, permite el trabajo en consola, así permite utilizar la consola para realizar las operaciones de Symfony (JetBrains, 2015). Este es un producto de la empresa JetBrains, por lo que es un software privativo, pero puede ser usado ya que la UCI cuenta con una licencia para la explotación del producto.

1.4 Conclusiones parciales

La gestión de tipologías y el manejo de metadatos son procesos que están relacionados de manera cercana, ya que el primero define las clases y atributos y el segundo brinda esos atributos por lo que se complementan, un buen uso de sus ventajas permite llevar a cabo un mejor proceso de catalogación. A pesar de que las soluciones estudiadas no realizan los dos procesos, su estudio permitió conocer los principales metadatos relacionados a las medias y su forma de representación y el estudio del módulo de gestión de tipologías se convirtió en una guía para el diseño del módulo de gestión de tipologías para PRIMICIA 2.0, que se puede realizar con las metodologías y herramientas seleccionadas.

Capítulo 2: Características y diseño del módulo

Antes de comenzar la implementación de un sistema de software es necesario hacer un estudio previo para un mejor entendimiento de la situación que influye en el desarrollo de la aplicación. Los procesos de ingeniería de software proveen al desarrollador del sistema de un mapa con un destino exitoso (Pressman, y otros, 2015). Es necesario crear los artefactos que plantea la metodología AUP en la etapa de Captura de Requisitos, Análisis y Diseño.

En este capítulo se especifican los requisitos funcionales (RF) y los no funcionales (RNF), y sus descripciones. Se identifican los actores y casos de uso (CU) para especificar los requisitos del sistema, usados para el diseño, implementación y pruebas de este. Se muestran los diagramas necesarios para entender los procesos que se efectúan en el módulo.

2.1 Modelo de dominio

Durante el análisis orientado a objetos, se presta especial atención a encontrar y describir los objetos –o conceptos - en el dominio del problema. La finalidad del análisis orientado a objetos es crear una descripción del dominio desde la perspectiva de la clasificación de objetos. Una descomposición del dominio conlleva una identificación de los conceptos, atributos y asociaciones que se consideran significativas. El resultado se puede expresar en un modelo del dominio, que se ilustra mediante un conjunto de diagramas que muestran los objetos o conceptos del dominio (Larman, 2004). Los modelos de dominio permiten realizar una abstracción del problema, y a partir de él se pueden obtener actores y requisitos del sistema, así como los futuros casos de uso.

En la figura 1 se muestra el modelo de dominio y las clases conceptuales que lo forman.

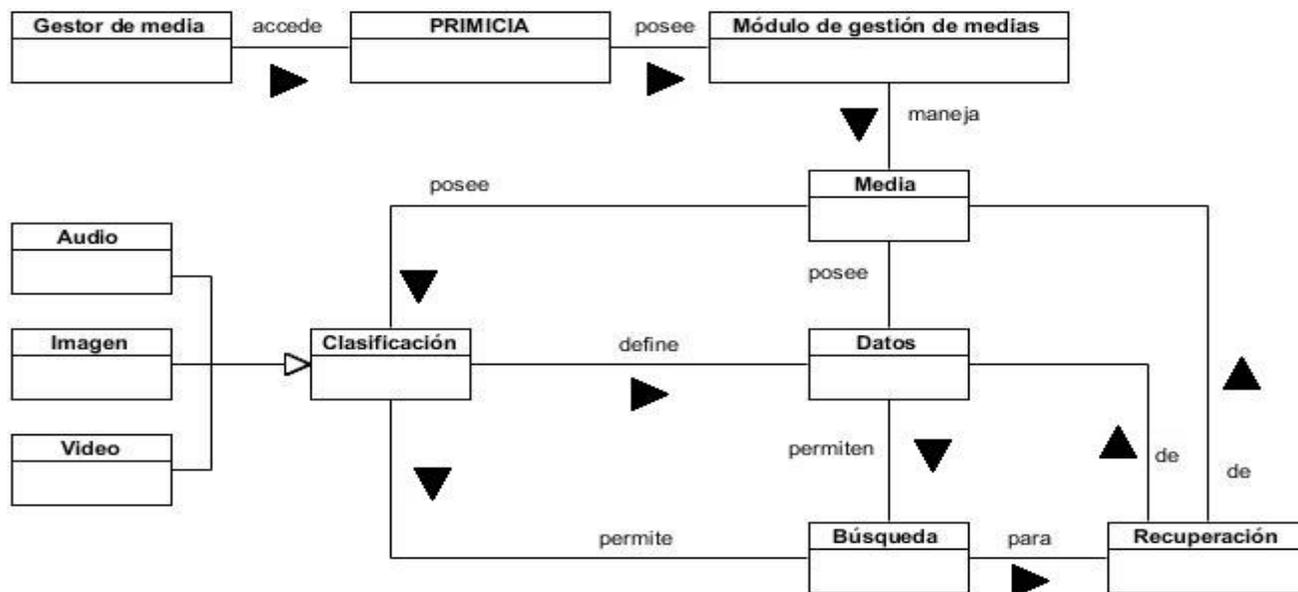


Figura 1. Modelo de dominio

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

2.1.1 Descripción del modelo de dominio

El gestor de medias accede a la plataforma PRIMICIA que posee un módulo para la gestión de medias. Este módulo permite manejar una media o archivo multimedia cuando es adicionada, la media se clasifica en audio, imagen o video. Esta clasificación define los datos que describen al archivo que son título, autor, género, palabras claves y duración para el caso de los videos, que son agrgados durante el proceso de adición. Esta clasificación y los atributos que se definen para la media permiten la posterior búsqueda y obtención del archivo o sus datos.

2.1.2 Descripción de las clases del modelo de dominio

- Gestor de media: Rol que se le asigna al individuo que gestiona las medias en la organización donde se despliega la plataforma PRIMICIA 2.0.
- PRIMICIA: Plataforma de Televisión Informativa PRIMICIA 2.0.
- Módulo de gestión de medias: Es el módulo que permite el manejo de las medias dentro de la plataforma PRIMICIA 2.0
- Media: es el archivo multimedia que se desee almacenar.
- Clasificación: es el proceso de asignar una media a su grupo.
- Audio, Imagen y Video: son los tipos en que son clasificadas las medias.
- Datos: son los atributos que describen a una media.
- Búsqueda: es el proceso para encontrar archivos multimedia que han sido almacenados para poder recuperarlos.
- Recuperación: proceso en donde se obtiene medias o sus datos.

2.2 Requisitos

Los requisitos para un sistema son las descripciones de qué es lo que el sistema debe hacer, los servicios que debe proveer y las condiciones para su trabajo. Estos requerimientos reflejan las necesidades del cliente (Sommerville, 2011). Es necesario conocer lo que desea el cliente, los procesos que se van a informatizar y los detalles a tener en cuenta durante el desarrollo del módulo. Los requisitos pueden ser clasificados en funcionales y no funcionales.

2.2.1 Requisitos funcionales

- **RF1:** Adicionar tipología
Descripción: El sistema debe permitir agregar una tipología y definir su nombre.
Entrada: El nombre de la tipología.
Salida: Mensaje de notificación.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

- **RF2:** Listar tipologías
Descripción: El sistema debe permitir visualizar las tipologías existentes, su nombre y si un archivo multimedia la utiliza.
Salida: Lista de tipologías.
- **RF3:** Visualizar tipología
Descripción: El sistema debe permitir visualizar el nombre una tipología existente, la fecha en que fue creada y la última fecha en que fue modificada.
Entrada: Identificador de la tipología.
Salida: Datos de la tipología.
- **RF4:** Modificar tipología
Descripción: El sistema debe permitir editar el nombre una tipología existente.
Entrada: Identificador de la tipología y el nuevo nombre de la tipología.
Salida: Mensaje de notificación.
- **RF5:** Eliminar una tipología
Descripción: El sistema debe permitir eliminar una tipología existente.
Entrada: Identificador de la tipología.
Salida: Mensaje de notificación.
- **RF6:** Buscar tipología
Descripción: El sistema debe permitir la búsqueda de una tipología.
Entrada: Cadena de caracteres.
Salida: Lista de tipologías que cumplen con el criterio de búsqueda.
- **RF7:** Seleccionar campos de una tipología
Descripción: El sistema debe permitir seleccionar de un grupo de campos definidos cuales utilizará una tipología existente.
Entrada: Identificador de la tipología y campos seleccionados.
Salida: Mensaje de notificación.
- **RF8:** Adicionar campo
Descripción: El sistema debe permitir agregar un campo, define su nombre, tipo de dato que recoge (números enteros, números decimales o caracteres alfanuméricos), la longitud máxima de caracteres que se podrán ingresar y una breve descripción de su función.
Entrada: El nombre del campo, tipo de dato que recoge, longitud máxima de caracteres y descripción de la función del campo.
Salida: Mensaje de notificación.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

- **RF9:** Listar campos
Descripción: El sistema debe permitir visualizar los campos existentes, su nombre, el tipo de dato que recoge y la longitud del campo.
Salida: Lista de campos.
- **RF10:** Visualizar campo
Descripción: El sistema debe permitir visualizar el nombre un campo existente, el tipo de dato que almacena, la longitud del campo y una descripción de la función que realiza.
Entrada: Identificador del campo.
Salida: Datos del campo.
- **RF11:** Modificar campo
Descripción: El sistema debe permitir editar el nombre un campo existente, el tipo de dato que almacena, la longitud del campo y una descripción de la función que realiza. Se debe tener en cuenta que, si el campo es utilizado por una tipología no se puede modificar el tipo de dato y la longitud.
Entrada: Identificador del campo, nuevo nombre del campo, nuevo tipo de dato, nueva longitud del campo y nueva descripción de la función del campo.
Salida: Mensaje de notificación.
- **RF12:** Eliminar campo
Descripción: El sistema debe permitir eliminar un campo existente.
Entrada: Identificador del campo.
Salida: Mensaje de notificación.
- **RF13:** Buscar campo
Descripción: El sistema debe permitir la búsqueda de un campo.
Entrada: Cadena de caracteres
Salida: Lista de campos que cumple con los parámetros de búsqueda.

2.2.2 Requisitos no funcionales

- **RFN1** Interfaz
El módulo debe utilizar la interfaz diseñada para la plataforma PRIMICIA.
- **RFN2** Restricciones de diseño
Para la implementación del componente se tiene en cuenta como restricción del diseño utilizar el framework Symfony 2.3.7, como lenguaje de programación a PHP 5.3, a PostgreSQL 9.4 como gestor de base de datos y la biblioteca getID3 1.9.12 para la extracción de los metadatos de las medias.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

- **RNF3** Requisitos de hardware y software

- **Software:**

- Para la computadora cliente:

- Un navegador Mozilla Firefox 35 o Chrome 32. Sistema operativo: Ubuntu 14.04 o Windows 7 o superior.

- Para el Servidor de Administración:

- Sistema operativo: Ubuntu 14.04. Sistema Gestor de Base de Datos: PostgreSQL 9.2.

- Servidor Web: Apache 2.2.

- **Hardware:**

- Para la computadora cliente:

- 1GB de RAM. Microprocesador: Core 2 Duo a 3.3 GHz. Espacio en disco: 40 GB.

- Para el Servidor Administración:

- Microprocesador: Core 2 Duo a 3.3 GHz. 4 GB de RAM. Espacio en disco: 500 GB.

2.3 Modelo del Sistema

Los modelos del sistema son utilizados para apoyar el proceso de levantamiento de requisitos, durante el diseño se describe el sistema para su implementación, para documentar su estructura y operaciones antes de implementarlo (Sommerville, 2011). Los casos de uso del sistema son obtenidos a partir de las funcionalidades identificadas en los requisitos del software. A través del modelo de sistema es posible comprender el comportamiento de la aplicación mediante su interacción con los usuarios u otra aplicación. Los trece requisitos planteados se agrupan en dos casos de uso y existe un actor gestor de medias, representado en la figura 2.



Figura 2. Diagrama de Casos de Uso del Sistema.

A continuación en la tabla 1, se describe el actor del sistema y el papel que juega en el mismo.

Tabla 1.Descripción de los actores del sistema

Actores	Descripción
Gestor de medias	Es el rol del encargado de manejar las medias dentro de la organización donde se utilice la plataforma PRIMICIA

2.3.1 Especificación de casos de uso

Las especificaciones de casos de uso permiten mostrar cómo se encuentra compuesto cada caso de uso y el flujo de las operaciones que en él se representan. Es una guía para definir el comportamiento del sistema que se desea crear.

2.3.1.1 Caso de uso Gestionar tipología

Tabla 2. Especificación del caso de uso Gestionar tipología.

Objetivo	Listar, adicionar, visualizar, buscar, editar, eliminar tipologías y seleccionar los campos relacionados a estas en el sistema.	
Actores	Usuario: (Inicia) Adiciona, visualiza, edita, busca, elimina tipologías o selecciona campos de una tipología.	
Resumen	El caso de uso inicia mostrando una lista de tipologías existentes en el sistema, el usuario selecciona la opción adicionar, visualizar, editar o eliminar las tipologías, se introducen los datos necesarios según cada opción. Puede seleccionar además los campos que describen los atributos de una media clasificada por esa tipología. El caso de uso termina cuando se adicione, visualice, edite, elimine una tipología o se seleccionen los campos.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado y tener permisos de gestor de medias.	
Poscondiciones	Tipología adicionada exitosamente. Datos de la tipología modificados exitosamente. Tipología visualizada exitosamente. Tipología eliminada exitosamente. Lista de tipologías buscadas. Campos de tipología seleccionados.	
Flujo de eventos		
Flujo básico: Gestionar tipologías		
	Actor	Sistema
1	Selecciona la opción Tipologías en el Módulo de Gestión de Medias.	
2		Muestra una lista de las tipologías existentes en el sistema. Permite realizar varias acciones con una tipología: Adicionar una tipología. Ver Sección 1: "Adicionar tipología". Buscar una tipología. Ver Sección 2: "Buscar tipología". Visualizar una tipología. Ver Sección 3: "Visualizar tipología". Editar una tipología. Ver Sección 4: "Editar tipología". Eliminar una tipología: Ver Sección 5: "Eliminar tipología". Eliminar múltiples tipologías: Ver Sección 6: "Eliminar Múltiples". Seleccionar campos de una tipología. Ver Sección 7: "Campos de tipología".
3	Selecciona una de las opciones.	

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

4		Muestra la interfaz de la opción elegida.
Sección 1: Adicionar tipología		
Flujo básico: Adicionar tipología		
	Actor	Sistema
1.	Inserta los datos de la tipología (nombre).	
2	Pulsa el botón Aceptar.	
3		Valida que los datos insertados sean correctos.
4		Adiciona la tipología
5		Muestra un mensaje de éxito.
6		Actualiza el listado de tipologías. Termina el CU.
Flujos alternos		
2 Cancelar la operación		
	Actor	Sistema
2.1	Pulsa el botón Cancelar.	
2.2		Cancela la operación y muestra el listado de tipologías. Termina el CU.
3 Los datos insertados son incorrectos		
	Actor	Sistema
3.1		Muestra un mensaje de error en el formulario. Regresa al paso 1 del flujo básico Adicionar tipología.
Sección 2: Buscar tipología		
Flujo básico: Buscar tipología		
1	Inserta un texto en el campo de búsqueda.	
2	Pulsa el botón de búsqueda.	
3		Devuelve el listado de tipologías cuyos nombres coincidan con el texto insertado. Termina el CU.
Sección 3: Visualizar tipología		
Flujo básico: Visualizar tipología		
	Actor	Sistema
1		Visualiza los datos de la tipología seleccionada por el usuario (nombre, fecha de creación y fecha de modificación). Termina el CU.
Sección 4: Editar tipología		
Flujo básico: Editar tipología		
1		Verifica si la tipología está en uso.
2		Si no está en uso muestra la interfaz con los datos de la tipología.
3	Modifica los datos que desee de la tipología seleccionada.	
4	Pulsa el botón Aceptar.	
5		Valida que los datos modificados sean correctos.
6		Actualiza los datos de la tipología.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

7		Muestra un mensaje de éxito.
8		Actualiza el listado de tipologías. Termina el CU.
Flujos alternos		
1 La tipología está en uso		
	Actor	Sistema
1.1		Notifica al usuario que lo tipología no puede ser modificada porque está en uso.
4 Cancelar la operación		
	Actor	Sistema
4.1	Pulsa el botón Cancelar.	
4.2		Cancela la operación y muestra el listado de tipologías. Termina el CU.
5 Los datos insertados son incorrectos		
	Actor	Sistema
5.1		Muestra un mensaje de error en el formulario. Regresa al paso 3 del flujo básico Editar tipología.
Sección 5: Eliminar tipología		
Flujo básico: Eliminar tipología		
	Actor	Sistema
1		Verifica si la tipología está en uso.
2		Si no está en uso muestra un mensaje de confirmación de la operación.
3	Pulsa el botón Aceptar.	
4		Muestra un mensaje de éxito.
5		Actualiza el listado de tipologías. Termina el CU.
Flujos alternos		
1 La tipología está en uso		
	Actor	Sistema
1.1		Notifica al usuario que lo tipología no puede ser eliminada porque está en uso.
2 Cancelar la operación		
	Actor	Sistema
2.1	Pulsa el botón cancelar	
2.2		Muestra el listado de tipologías. Termina el CU.
Sección 6: Eliminar múltiples		
Flujo básico: Eliminar múltiples		
1	Selecciona varias tipologías y presiona el botón Eliminar seleccionados.	
2		Muestra un mensaje de confirmación de la operación.
3	Pulsa el botón Aceptar.	
4		Elimina las tipologías seleccionadas que no estén en uso.
5		Actualiza el listado de tipologías. Termina el CU.
Flujos alternos		

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

1 No selecciona ninguna tipología y presiona el botón eliminar seleccionados		
1.1		Muestra un mensaje de error indicando que debe seleccionar al menos una tipología.
1.2	Pulsa el botón Aceptar.	
1.3		Cierra el mensaje de error y muestra el listado de tipologías. Termina el CU.
3 Cancelar la operación		
	Actor	Sistema
3.1	Pulsa el botón Cancelar.	
3.2		Muestra el listado de tipologías. Termina el CU.
4 Hay tipologías seleccionadas están en uso		
4.1		Muestra un mensaje de alerta indicando que algunas tipologías están en uso y no pueden ser eliminadas, las tipologías que no pueden ser eliminadas aparecen en rojo. Termina el CU
Sección 7: Campos de tipología		
Flujo básico: Campos de tipologías		
	Actor	Sistema
1		Muestra un listado de los campos
2	Selecciona los campos que desea y confirma el cambio.	
3		Verifica que se seleccione al menos un campo
5		Modifica la tipología.
6		Muestra un mensaje de éxito. Termina el CU.
Flujos alternos		
1 La tipología está en uso		
	Actor	Sistema
1.1		Muestra un mensaje de error que notifica al usuario que lo tipología no puede ser modificada porque está en uso. Muestra la lista de campos. Termina el CU
4 No existen campos seleccionados		
4.1		Muestra un mensaje de error y notifica al usuario que debe seleccionar al menos un campo. Vuelve al paso 2 del flujo normal.
Relaciones	CU incluidos	No existen.
	CU extendidos	Gestionar campos.
Requisitos no funcionales	RNF1.	
Asuntos pendientes	No procede.	

2.3.1.2 Caso de uso Gestionar campos

Tabla 3: Especificación del caso de uso Gestionar campos

Objetivo	Listar, adicionar, visualizar, buscar, editar y eliminar campos en el sistema	
Actores	Usuario: (Inicia) Adiciona, visualiza, edita, filtra o elimina campos	
Resumen	El caso de uso inicia mostrando una lista de campos existentes en el sistema, el usuario selecciona la opción adicionar, mostrar, buscar, editar o eliminar las campos, se introducen los datos necesarios según cada opción. El caso de uso termina cuando se adicione, visualice, edite, busque o elimine un campo.	
Complejidad	Baja	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado, tener permisos de gestor de medias	
Poscondiciones	Campo adicionado exitosamente Datos del campo modificados exitosamente Campo visualizado exitosamente Campo eliminado exitosamente Lista de campos buscados	
Flujo de eventos		
Flujo básico: Gestionar campos		
	Actor	Sistema
1	Selecciona la opción Gestionar campos.	
2		Muestra una lista de los campos existentes en el sistema. Permite realizar varias acciones con un campo: Adicionar un campo. Ver Sección 1: "Adicionar campo". Visualizar un campo. Ver Sección 2: "Visualizar campo". Editar un campo. Ver Sección 3: "Editar campo". Eliminar un campo: Ver Sección 4: "Eliminar campo". Eliminar múltiples campos: Ver Sección 5: "Eliminar Múltiples". Buscar un campo. Ver Sección 6: "Buscar campo".
3	Selecciona una de las opciones.	
4		Muestra la interfaz de la opción elegida.
Sección 1: Adicionar campo		
Flujo básico: Adicionar campo		
	Actor	Sistema
1	Inserta los datos del campo (nombre, tipo de dato, longitud del campo y descripción de la función del campo).	
2	Pulsa el botón Aceptar.	
3		Valida que los datos insertados sean correctos.
4		Adiciona el campo.
5		Muestra un mensaje de éxito.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

6		Actualiza el listado de campos. Termina el CU.
Flujos alternos		
2 Cancelar la operación		
	Actor	Sistema
2.1	Pulsa el botón Cancelar.	
2.2		Cancela la operación y muestra el listado de campos. Termina el CU.
3 Los datos insertados son incorrectos		
	Actor	Sistema
3.1		Muestra un mensaje de error en el formulario. Regresa al paso 1 del flujo básico Adicionar campo.
Sección 2: Visualizar campo		
Flujo básico: Visualizar campo		
	Actor	Sistema
1		Visualiza los datos del campo seleccionado por el usuario (nombre, tipo de dato, longitud del campo y descripción de la función del campo). Termina el CU.
Sección 3: Editar campo		
Flujo básico: Editar campo		
1		Verifica si el campo es usado por tipologías.
2		Si no está en uso muestra la interfaz con los datos del campo.
3	Modifica los datos que desee del campo seleccionado.	
4	Pulsa el botón Aceptar.	
5		Valida que los datos modificados sean correctos.
6		Actualiza los datos del campo.
7		Muestra un mensaje de éxito
8		Actualiza el listado de campos. Termina el CU.
Flujos alternos		
1 El campo está en uso		
	Actor	Sistema
1.1		Muestra la interfaz con los datos del campo, pero no permite la modificación del campo de longitud.
1.2	Modifica los valores de los campos y continúa en el paso 4 el flujo básico Editar campo.	
4 Cancelar la operación		
	Actor	Sistema
4.1	El usuario pulsa el botón Cancelar.	
4.2		Cancela la operación y muestra el listado de campos. Termina el CU.
5 Los datos insertados son incorrectos		
	Actor	Sistema

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

5.1		Muestra un mensaje de error en el formulario. Regresa al paso 3 del flujo básico Editar campo.
Sección 4: Eliminar campo		
Flujo básico: Eliminar campo		
	Actor	Sistema
1		Muestra un mensaje de confirmación de la operación.
2	El usuario pulsa el botón Aceptar.	
3		Verifica si el campo es usado por tipologías.
4		Si no está en uso muestra un mensaje de éxito.
5		Actualiza el listado de campos. Termina el CU.
Flujos alternos		
1 La tipología está en uso		
	Actor	Sistema
1.1		Notifica al usuario que el campo no puede ser eliminado porque está en uso. Muestra el listado de campos. Termina el CU.
2 Cancelar la operación		
	Actor	Sistema
2.1	El usuario pulsa el botón Cancelar.	
2.2		Muestra el listado de campos. Termina el CU.
Sección 5: Eliminar múltiples		
Flujo básico: Eliminar múltiples		
1	Selecciona varios campos y presiona el botón Eliminar seleccionados.	
2		Muestra un mensaje de confirmación de la operación.
3	Pulsa el botón Aceptar.	
4		Elimina las tipologías seleccionadas que no estén en uso.
5		Actualiza el listado de campos. Termina el CU.
Flujos alternos		
1 No selecciona ningún campo y presiona el botón Eliminar seleccionados		
1.1		Muestra un mensaje de error indicando que debe seleccionar al menos un campo.
3 Cancelar la operación		
	Actor	Sistema
3.1	El usuario pulsa el botón Cancelar.	
3.2		Muestra el listado de campos. Termina el CU.
4 Hay campos seleccionados que están en uso		
4.1		Muestra un mensaje de alerta indicando que algunos de los campos están en uso y no pueden ser eliminados. Los campos en uso aparecen en rojo. Continúa en el paso 5 del flujo normal.

Sección 6: Buscar campo		
Flujo básico: Buscar campo		
	Inserta un texto en el campo de búsqueda.	
	Pulsa el botón de búsqueda.	
		Devuelve el listado de campos cuyos nombres coincidan con el texto insertado. Termina el CU.
Relaciones	CU incluidos	No existen.
	CU extendidos	No existen.
Requisitos no funcionales	RNF1	
Asuntos pendientes	No procede.	

2.4 Patrón arquitectónico

Un patrón de arquitectura de software es un esquema genérico que se ha probado para resolver un problema en particular, que es recurrente en cierto contexto. Este define la estructura, funcionamiento e interacción entre las partes de un sistema, se especifican los componentes, con sus responsabilidades y relaciones.

2.4.1 Modelo Vista Controlador MVC

El framework Symfony seleccionado para desarrollar el componente está desarrollado bajo las restricciones, suposiciones y recomendaciones de la arquitectura Modelo Vista Controlador (MVC). Esta arquitectura sugiere la separación del software en tres componentes: Modelo, Vista y Controlador.

El modelo es la representación de la información que maneja la aplicación. Está constituido por los datos puros y la lógica de los propios datos que puestos en el contexto del sistema proveen de información al usuario y en algunos casos a la propia aplicación.

La vista es la representación del modelo en forma gráfica disponible para la interacción con el usuario, mostrando al usuario los datos que componen al modelo y a través de ella el usuario interactúa con el sistema.

El controlador es la parte encargada de manejar y responder las solicitudes del usuario procesando toda la información necesaria y modificando, en caso de ser necesario, el modelo.

El ciclo de vida del MVC se puede describir a partir de los siguientes pasos, el primero empieza cuando el usuario hace una solicitud al controlador con información sobre lo que él desea realizar. Entonces el controlador decide a quién delegar la tarea, ahí comienza el trabajo del modelo. En esa etapa el modelo se encarga de realizar operaciones sobre la información que maneja para cumplir con lo que le solicita el controlador. Luego de realizar su labor devuelve al controlador la información resultante y este a su vez genera la vista la cual se encarga de mostrar la información de forma visual al usuario (, 2010)

2.5 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación. (Jacobson, 2000)

El modelo de diseño permite la descripción de cómo estará estructurada la aplicación, es una guía para la implementación del módulo. En este modelo se representan las diferentes clases y estructuras que se derivan de los RF, las clases del sistema y la arquitectura, definidas anteriormente.

2.5.1 Patrones de diseño

Un patrón es una descripción de un problema y la esencia de su solución, esa solución puede ser reutilizada en diferentes entornos. El patrón no es una especificación detallada, puede tomarse como una descripción de la sabiduría y la experiencia acumulada, es una solución acertada a un problema común que ya ha sido probada.

Los patrones han tenido un gran impacto en el diseño de software orientado a objetos, además de ser soluciones probadas para problemas comunes se han convertido en parte del vocabulario cuando se habla de diseño. Se puede explicar los diseños que se realizan a partir de la descripción de los patrones que se han usado. (Sommerville, 2011)

El framework Symfony aplica ciertos patrones pertenecientes a dos conocidos grupos los GRASP (Patrones Generales de Software para Asignación de Responsabilidades) y los GOF (Banda de los cuatro).

El patrón alta cohesión se evidencia en la implementación de la clase *Action*, la cual tiene la responsabilidad de definir las acciones para las plantillas y, paralelo a esto colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, está formada por diferentes funcionalidades que están relacionadas estrechamente. (Hidalgo, 2010)

El patrón creador se muestra en las clases mencionadas anteriormente para las operaciones lógicas del negocio en cuestión y la ejecución de cada una de ellas. En las acciones se crean objetos de las clases que representan las entidades evidenciando que la clase *Action* es creadora de dichas entidades. (Hidalgo, 2010)

El patrón controlador, se evidencia en la utilización del controlador frontal que es el punto de entrada único de toda la aplicación y maneja todas las peticiones. Es el encargado de recibir la petición del usuario y en función de ella, envía la solicitud a las distintas clases para que sea procesada. (Hidalgo, 2010)

El patrón estructural GOF *Decorator* (Envoltorio) añade funcionalidad a una clase, dinámicamente. Los archivos de plantillas o *layouts*, almacenan el código HTML que es común en todas las páginas de la aplicación para no tener que repetirlo en cada página. El contenidos de la página se integra en el *layout* o desde otro punto de vista el *layout* decora el contenido de la plantilla.

2.5.2 Diagramas de clase del diseño

Estos diagramas permiten visualizar como se encuentra estructurado el módulo que se desarrolla, las clases, sus atributos y las relaciones que existen entre ellas, además de otros elementos necesarios para el correcto funcionamiento de la aplicación. A continuación se modelará cada caso de uso de manera independiente.

2.5.2.1 Caso de uso Gestionar tipología

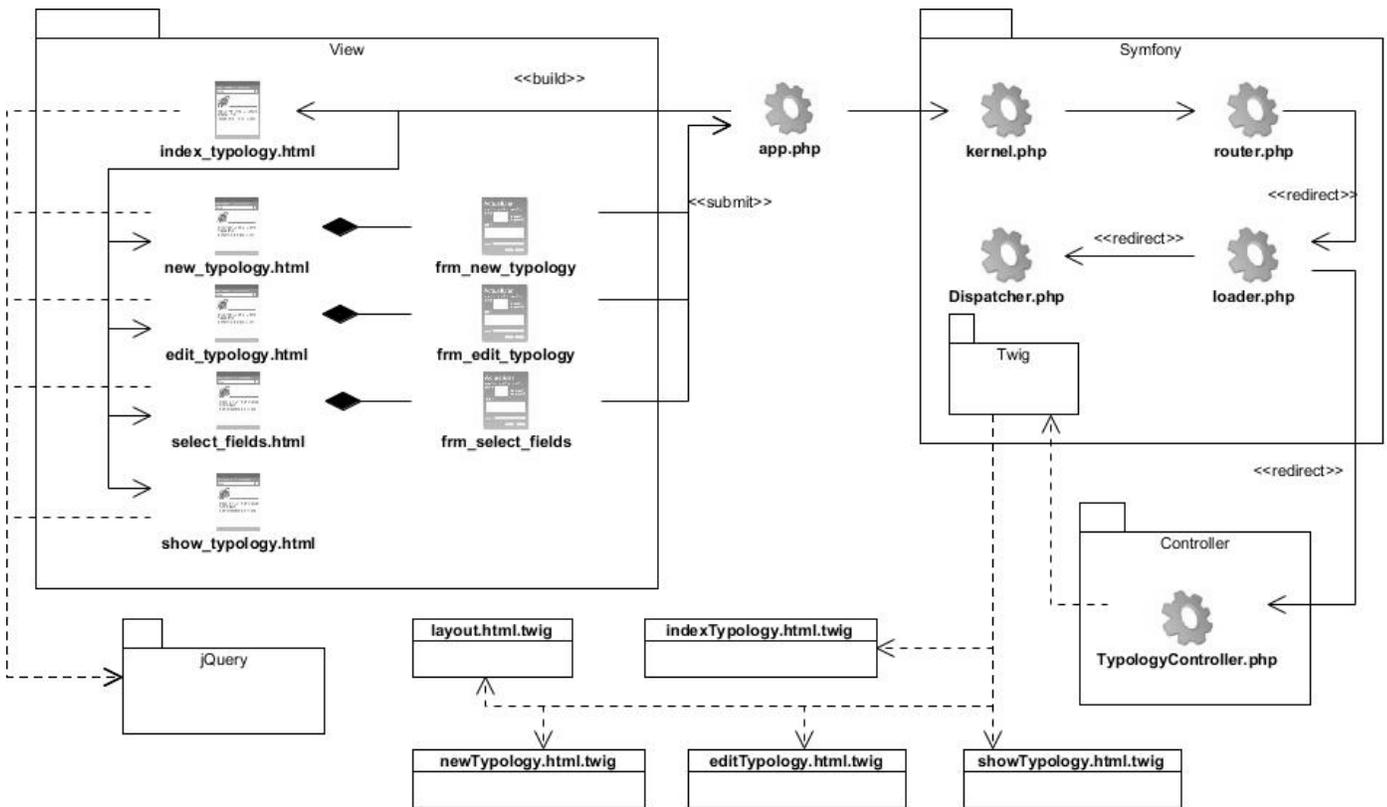


Figura 3. Diagrama de Clase del Diseño: Gestionar Tipología

2.5.2.2 Caso de uso Gestionar campos

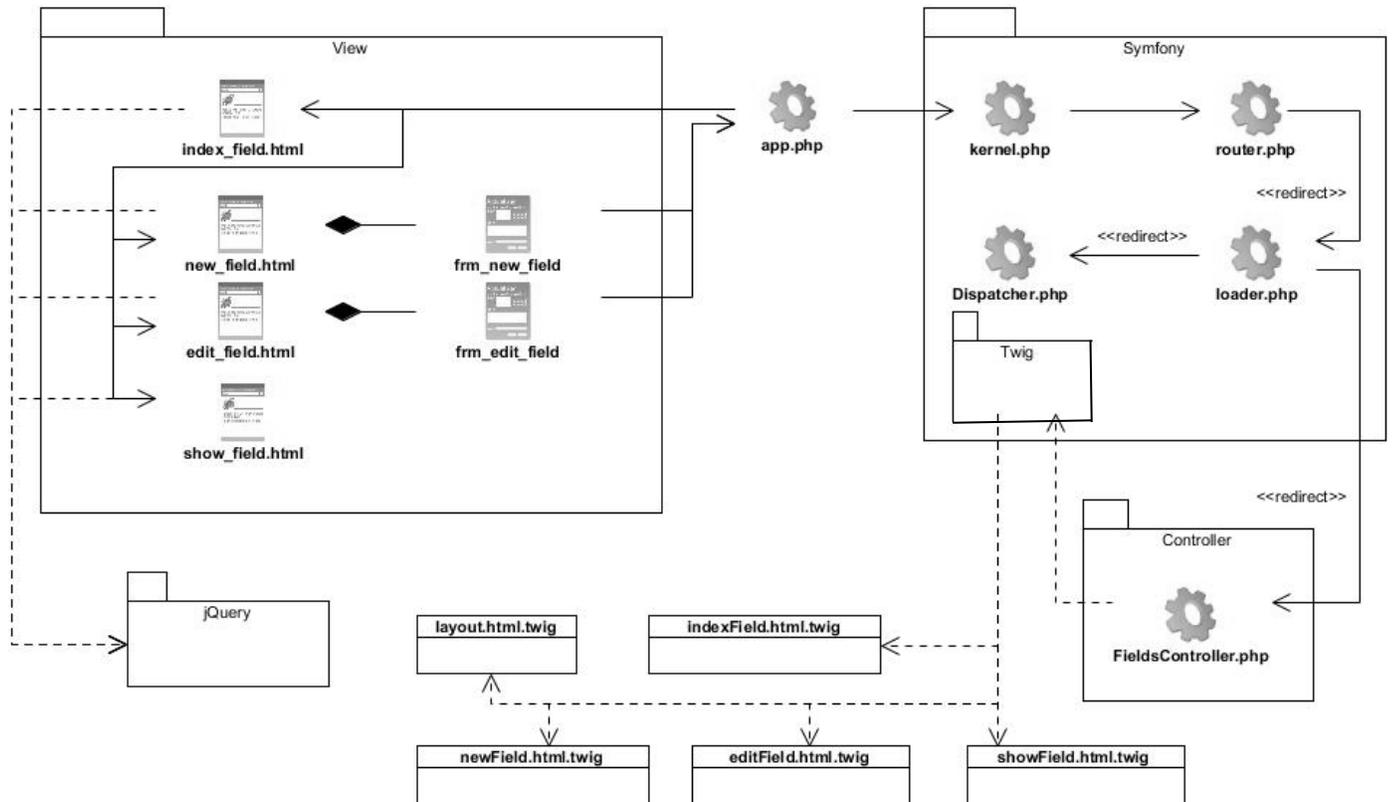


Figura 4. Diagrama de Clases del Diseño: Gestionar campos de tipología

2.6 Modelo de datos

El modelo de datos permite representar los elementos y los atributos, que se desean almacenar durante un tiempo determinado. Además representa las relaciones que existen entre estos atributos. Se representan a través de los diagramas de entidad-relación modelando las entidades atributos, relaciones, cardinalidad y llaves, que formarán la base de datos que utilizará el módulo.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

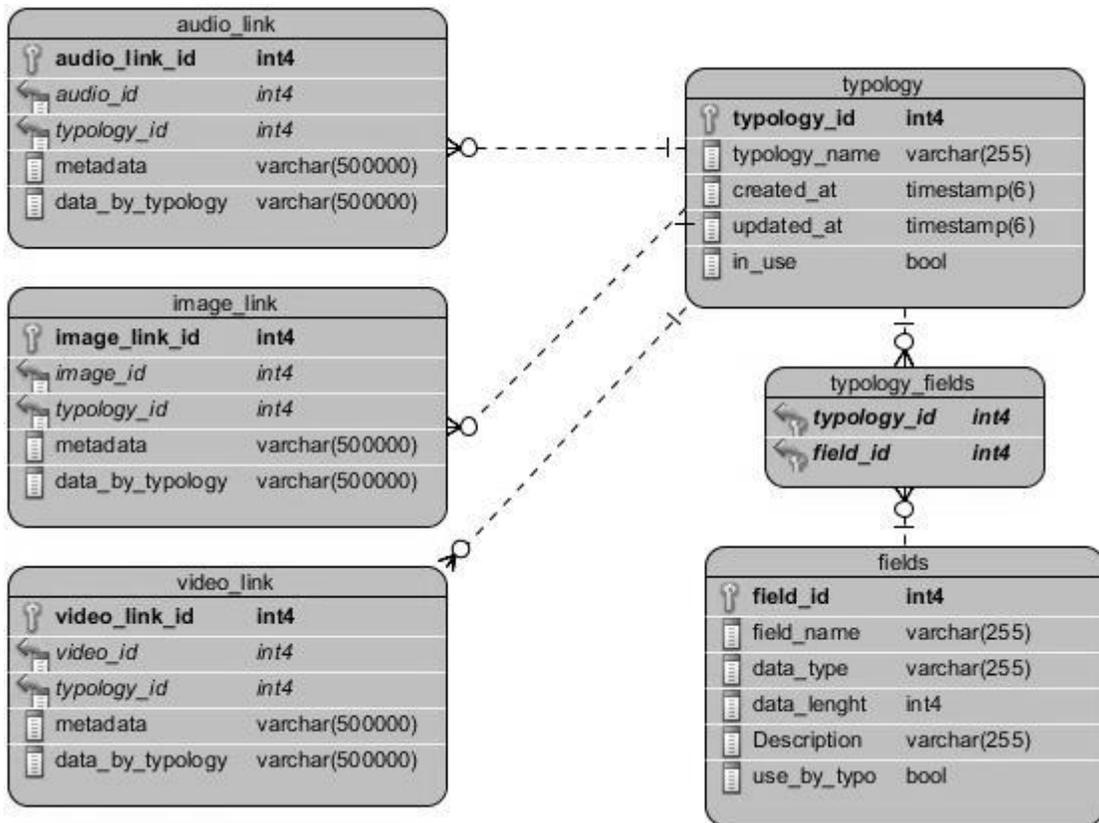


Figura 5. Diagrama de Entidad-Relación del Módulo de gestión de Tipologías

Las entidades audio_link, imagen_link y video_link representadas anteriormente se encuentran relacionadas a las entidades audio, imagen y video respectivamente que forman parte del Módulo de Gestión de Medias de PRIMICIA 2.0, las clases link son utilizadas para enlazar los tipos de medias existentes con las tipologías y para almacenar los datos relacionados a estas, así como almacenar los metadatos originales de las medias

2.7 Conclusiones parciales

El modelo de dominio permitió comprender como son manejadas las medias en el módulo de gestión de medias de PRIMICIA 2.0 y comprender el entorno en que se incluirá el módulo de gestión de tipologías. La definición de los requisitos funcionales y no funcionales definen las condiciones que debe cumplir el módulo a desarrollar. A través de la descripción del diagrama del sistema y los casos de uso que lo conforma se determina la estructura del módulo.

Capítulo 3: Implementación y validación de la solución

En este capítulo se describe cómo queda implementado el módulo y cómo se realiza el despliegue del módulo en la plataforma. Se presenta el estándar de codificación utilizado para dar uniformidad al código creado. Luego se presentan las pruebas realizadas al módulo para verificar su funcionamiento y los resultados que se obtienen de estas.

3.1 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes (base de datos, ejecutables, módulos o ficheros). Describe como se organizan los componentes según el entorno en donde se implementan y el lenguaje de programación que se utiliza. Permite visualizar como esta está construido el sistema y las relaciones que existen entre los diferentes componentes.

A continuación se muestran los diagramas de componentes para los casos de uso Gestionar Tipologías y Gestionar Campos.

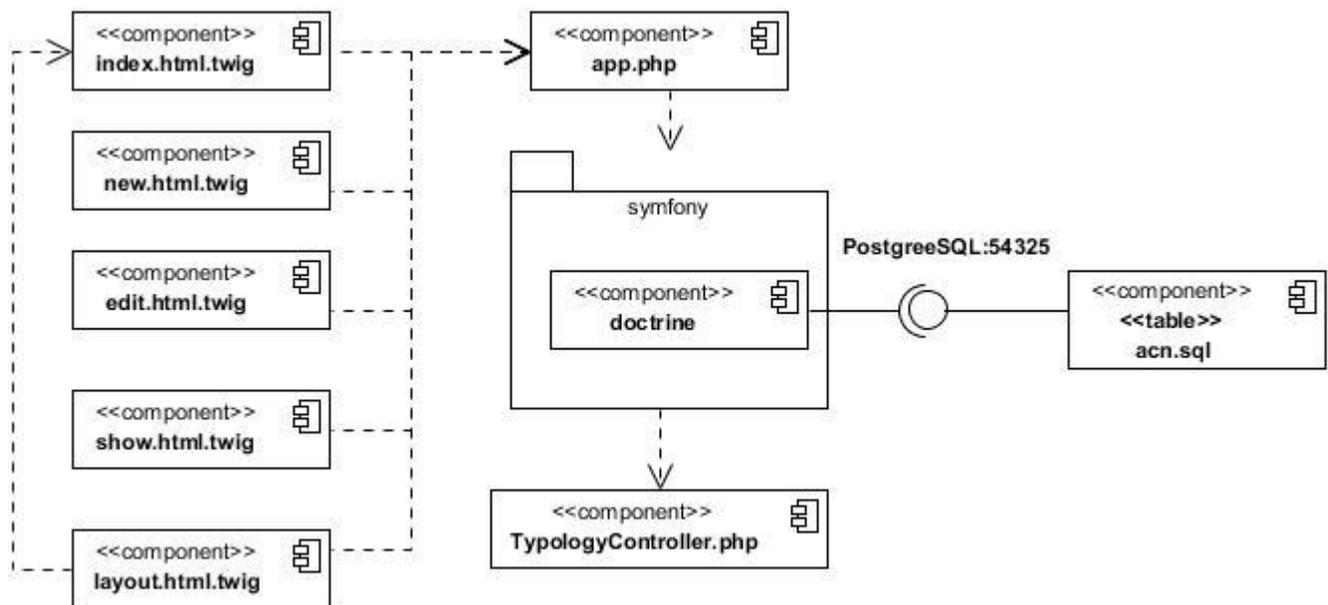


Figura 6. Diagrama de componentes para Gestionar Tipologías

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

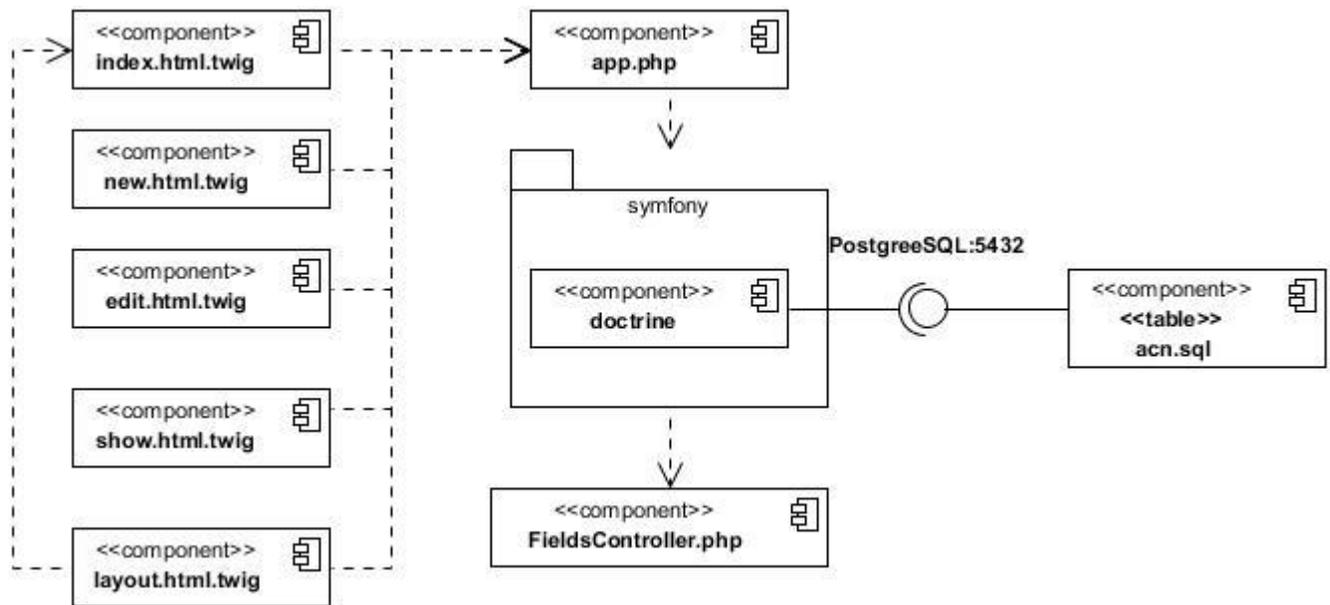


Figura 7. Diagrama de componentes para Gestionar Campos

En los diagramas anteriores se observa como las plantillas *twig* que generan las vistas se comunican con el controlador frontal, quien llama a la controlador relacionado a las plantillas utilizando los componentes de Symfony, luego el componente Doctrine se conecta a la base de datos para manejar los datos almacenados o ingresar nuevos.

3.3 Estándares de codificación

Un estándar de codificación comprende los aspectos de creación de código. El código fuente de una aplicación debe ser uniforme como si fuese programado por un único programador de una sola vez. Que exista una uniformidad en la forma de representar el código posibilita que sea comprendido fácilmente y facilita el proceso de mantenimiento.

En el desarrollo del módulo se aplicó el estándar de codificación *lowerCamelCase* para la representación de los métodos y las variables, el cual define que para conformar un nombre la primera letra de la palabra debe ser minúscula y el resto de las palabras que conforman el nombre su primera letra debe ser mayúscula, simulando las jorobas de un camello. A continuación se muestra un ejemplo.

```
public function updateFieldsTypologyAction(Request $request)
{
    $data = $request->query->get('datos');
    $data = json_decode($data, true);
    $campos = $data['campos'];
    $id = $data['id'];
    $translator = $this->get('translator');
    $em = $this->getDoctrine()->getManager();
    $tipo = $em->getRepository('TipologiaBundle:Typology')->findOneBy(array('id' => $id));
    $eliminarCampos = $tipo->getFields();
}
```

Figura 8. Ejemplo del uso del estándar lowerCamelCase

3.2 Modelo de despliegue

El modelo de despliegue muestra los nodos físicos o equipos de hardware durante la ejecución del módulo. En este caso el módulo se encuentra relacionado al Módulo de Gestión de Medias, dentro del Subsistema de Administración por lo que se encuentra ubicado en el servidor web para la administración y la base de datos.

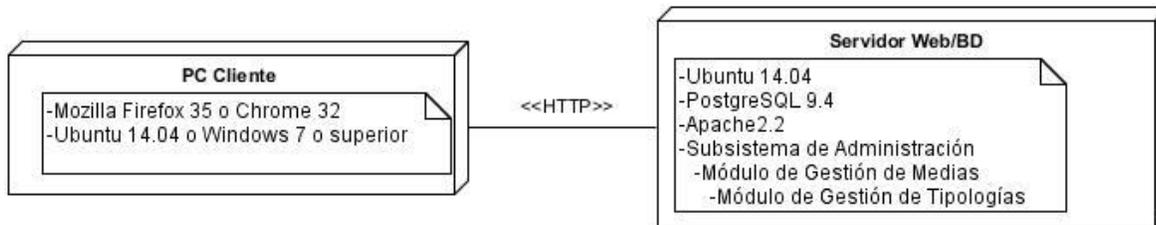


Figura 9. Diagrama de despliegue

3.3 Pruebas de software

La prueba de software es uno de los elementos principales relacionados a la verificación y validación de la solución. La verificación se refiere a un conjunto de tareas que asegura que se implemente de manera correcta una función específica. La validación se refiere a los diferentes grupos de tareas que verifican que el módulo construido cumple con los requerimientos del cliente. Las pruebas permiten asegurar la calidad de la aplicación y determinar la existencia de errores (Pressman, y otros, 2015).

La metodología AUP-UCI plantea la realización de tres niveles de prueba: pruebas internas, pruebas de liberación y pruebas de aceptación (Sánchez, 2015). Para probar el correcto funcionamiento del módulo se realizan pruebas internas que son llevadas a cabo por el desarrollador para verificar el correcto funcionamiento del código implementado. Se realizan pruebas de caja negra para comprobar el funcionamiento de la lógica del módulo y pruebas de integración para verificar que funciona de manera conjunta con el módulo de gestión de medias.

3.3.1 Pruebas de caja negra

Las pruebas de caja negra se realizan sobre la interfaz de usuario del sistema su objetivo es validar el correcto funcionamiento del módulo, verifica que las entradas sean correctas, las validaciones y los datos que inserta el usuario funcionen correctamente.

Se diseña un caso de prueba para cada caso de uso del sistema a continuación se muestra el escenario insertar una tipología del caso de prueba para Gestión de tipologías, en las tablas 4 y 5.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

Tabla 4. Caso de prueba gestionar tipología

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo central
EC 1.1 Adicionar tipología exitosamente	El usuario adiciona el nombre de la tipología	V	Muestra un mensaje que la tipología se adicionó exitosamente y actualiza el listado de tipologías	Seleccionar opción Adicionar. Insertar el nombre de la tipología. Dar clic en el botón "Aceptar"
		Videoclip		
EC 1.2 Adicionar tipología con errores	El usuario adiciona un nombre de la tipología no válido	I	Muestra un mensaje de error y se mantiene el formulario de adicionar tipología	Seleccionar opción Adicionar. Insertar el nombre de la tipología. Dar clic en el botón "Aceptar"
		Video-clip		
EC 1.3 Adicionar tipología con campos vacíos	El usuario no adiciona el nombre de la tipología	NA	Muestra un mensaje de error y se mantiene el formulario de adicionar tipología	Seleccionar opción Adicionar. Insertar el nombre de la tipología. Dar clic en el botón "Aceptar"
EC 1.4 Adicionar tipología ya existente	El usuario adiciona un nombre de tipología que ya está almacenada	V	Muestra un mensaje de error y se mantiene el formulario de adicionar tipología	Seleccionar opción Adicionar. Insertar el nombre de la tipología. Dar clic en el botón "Aceptar"
		Documental		
EC 1.5 Cancelar operación de adición	El usuario selecciona la opción de cancelar la operación	NA	Muestra el listado de tipologías	Seleccionar opción Adicionar. Dar clic en el botón "Cancelar"

Tabla 5. Descripción de las variables para el caso de prueba gestionar tipología.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Se debe introducir caracteres alfanuméricos, letras y números.
2	Campos	CheckBox	No	Marcar la opción

En la figura 10 se recogen los resultados obtenidos, en la primera iteración realizada al módulo fueron detectadas 5 no conformidades, las cuales fueron resueltas. Durante la segunda iteración no fueron detectadas no conformidades.

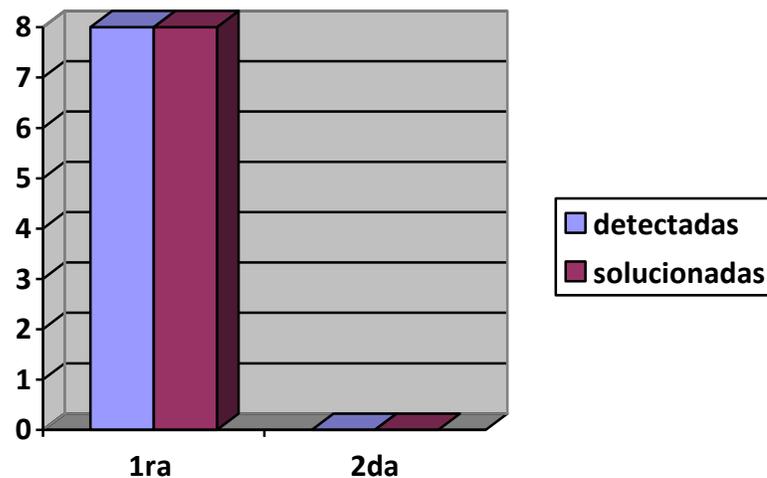


Figura 10. Resultados de la iteraciones realizadas al módulo.

3.3.2 Prueba de integración

La prueba de integración está orientada a verificar de forma sistemática que la estructura del programa y los errores que surjan de la interacción de las partes. Su objetivo es tomar los módulos y construir una estructura de programa que esté acorde al diseño (Pressman, y otros, 2015).

Se realiza la prueba de integración descendente, a partir del módulo de medias que existía se probó la integración con el nuevo módulo de tipologías verificando cada cambio que se realizaba en la conexión entre ambos módulos. Además la capacidad de Symfony de adicionar nuevos módulos o *bundles* al proyecto y el acceso que brinda el controlador frontal facilitan la integración del módulo de gestión de tipologías a PRIMICIA 2.0. Cuando se probó el funcionamiento del producto terminado se comprobó que este funciona de conjunto con el módulo de gestión de medias de manera correcta.

3.4 Conclusiones parciales

Con el diagrama de componentes se pudo observar cómo está compuesta la estructura de la aplicación, y las relaciones que existen entre cada componente. La utilización del estándar de codificación *lowerCamelCase* permite la creación de un código uniforme fácil de interpretar. La realización de las pruebas permitió verificar el correcto funcionamiento de los casos de uso Gestionar Tipología y Gestionar Campos, y la correcta integración del módulo de gestión de tipologías al módulo de gestión de medias de PRIMICIA 2.0.

Conclusiones Generales

Al finalizar el proceso de investigación se concluye que:

- La creación de la aplicación permitió la gestión de tipologías para la clasificación y la definición de nuevos datos para cada media, que pueden ser obtenidos a partir de los metadatos que contienen los archivos.
- La gestión de tipologías y el manejo de metadatos en conjunto permitió mejorar el proceso de catalogación y búsqueda, contribuyendo a la clasificación, descripción y búsqueda de los archivos de media que se realiza en PRIMICIA 2.0.
- Con las pruebas de software se validó que el módulo de gestión de tipologías cumple con las necesidades planteadas por el cliente y se integra de manera satisfactoria al módulo de gestión de medias de PRIMICIA 2.0.

Recomendaciones

Al concluir el presente trabajo se recomienda:

- Actualizar la versión de la biblioteca para la extracción de metadatos getID3, teniendo en cuenta la aparición de nuevos formatos de archivos y metadatos para la descripción de medias, así como facilidades que nuevas versiones pueden proveer.

Referencias Bibliográficas

Apple. 2011. Final Cut Pro 7 User Manual. *Media Management*. [En línea] 2011. [Citado el:18 de febrero de 2016.]

<https://documentation.apple.com/en/finalcutpro/usermanual/index.html#chapter=87%26section=1%26tasks=true>.

Asset Bank. 2015. AssetBank. [En línea] 2015. [Citado el: 2015 de octubre de 30.] <http://www.assetbank.co.uk>.

Barchini, Graciela Elisa. 2005. Métodos "I+D" de la Informática. s.l. : Revista de Informática Educativa y Medios Audiovisuales, 2005. 2(5), págs. 16-24. ISSN 1667-8338.

Betancourt Rodríguez, Iván. 2013. *Componente para la Gestión de Tipologías de archivos multimedia en aplicaciones web*. Universidad de las Ciencias Informáticas. La Habana, Cuba : Ediciones Futuro, 2013. págs. 32-39, Serie Científica de la Universidad de las Ciencias Informáticas, Vol 6, No. 11.

Bootstrap Core Team. 2015. Bootstrap. [En línea] 2015. [Citado el: 15 de marzo de 2016.] <http://getbootstrap.com>.

Canto. 2015. Canto_Cumulus. [En línea] 2015. [Citado el: 20 de noviembre de 2015.] <http://www.canto.com/cumulus/portals>.

Caplan, Priscilla. 1995. You call it corn, we call it syntax-independent metadata for document like objects. *The public Access Computer System review*. 1995. Vol. 4, 6.

Engström, Jonas. 2015. The evolution of the MAM systems. *Changing escenaries Changin Roles Part VII Setting the standard in second generation MAM systems and metadata. Selected papers from the FIAT/IFTA Media Management Seminar* . Glasgow : s.n., 2015.

getID3. 2015. getID3. [En línea] 2015. [Citado el: 18 de marzo de 2016.] <http://getid3.sourceforge.net/>.

Hernández León, Rolando Alfredo y Coello González, Sayda. 2011. *El proceso de Investigación científica*. La Habana : Editorial Universitaria Cubana, 2011. ISBN 978-959-16-1307-3.

JetBrains. 2015. PHPStorm. [En línea] 2015. [Citado el: 19 de marzo de 2016.] <http://www.jetbrains.com/phpstorm>.

jQuery Foundation. 2015. jQuery projects. [En línea] 2015. [Citado el: 15 de marzo de 2016.] <https://jquery.org/projects>.

Larman, Craig. 2004. *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Segunda Edición. Madrid : Pearson Education. Inc Publishing as Prentice Hall, 2004.

Marcos, Mari Carmen. 2004. Tarsys un software para la gestión de documentos audiovisuales. 2004. Vol. 13, 1, págs. 70-73.

Osoria Palma, Dayamí. 2013. *Módulo para la extracción y representación de los metadatos en el Sistema de Gestión Documental de audio y video digitales TeVeo Plus V1.0.* Universidad de las Ciencias Informáticas. La Habana, Cuba : s.n., 2013. Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas.

Photoools. 2015. IMatch. [En línea] 2015. [Citado el: 20 de noviembre de 2015.] <http://www.photoools.com/3720/exporting-metadata-attributes-text-exporter>.

Pim Core. 2015. Pim Core. [En línea] 2015. [Citado el: 20 de noviembre de 2015.] <http://www.pimcore.org>.

PostgreSQL. 2015. PostgreSQL. [En línea] 2015. [Citado el: 20 de noviembre de 2015.] <http://www.postgresql.org/about/>.

Pressman, Roger S. y Maxim, Bruce R. 2015. *Software Engineering. A practitioner's approach.* Eighth edition. New York : McGraw-Hill Education, 2015. ISBN 978-0-07-802212-8.

Figueroa Hidalgo, Daira, Vega Ortiz, Yurisbel y Martell Hernández, Vladimir. 2010. *Propuesta de diseño para proyectos informáticos que utilizan Symfony como Framework.* 5, 2010, Serie Científica. Universidad de las Ciencias Informáticas., Vol. 3.

RAE. 2016. Diccionario de la RAE. 2016.

Rofes Hernández, Rafael Néstor. 2011. *Procesamiento de metadatos para la descripción y recuperación de información en archivos de audio y video.* Universidad de las Ciencias Informáticas. La Habana, Cuba : s.n., 2011. Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas.

Sánchez, Tamara Rodríguez. 2015. *Metodología de desarrollo para la actividad productiva en la UCI.* Universidad de las Ciencias Informáticas. La Habana, Cuba : s.n., 2015.

Sommerville, Ian. 2011. *Software Engineering.* Ninth Edition. s.l. : Pearson Education, Inc. Addison-Weasley, 2011.

Sosa Vázquez, Sisley, Alonso Guerrero, Zorilín y Díaz Alés, Nilo Tomás. 2013. Sistema para la catalogación de audiovisuales para la web. s.l. : Revista Digital Sociedad de la Información, 2013. 40.

Sparks, Geoffrey. 2012. Sparx System. *Una introducción al UML. El modelo de Casos de Usos.* 2012.

Symfony. 2015. What is Symfony? [En línea] 2015. [Citado el: 20 de febrero de 2016.] <https://symfony.com/what-is-symfony>.

TechTerm. 2015. IDE definition. [En línea] 2015. [Citado el: 18 de marzo de 2016.] <http://techterms.com/definition/ide>.

TechTerms. 2014. JavaScript definition. [En línea] 2014. [Citado el: 28 de noviembre de 2015.] <http://techterms.com/definition/javascript>.

TechTerms. 2013. Framework definition. [En línea] 2013. [Citado el: 18 de febrero de 2016.] <http://techterms.com/definition/framework>.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

TechTerms. 2011. TechTerms. *Programming Language Definition*. [En línea] 2011. [Citado el: 12 de diciembre de 2015.] http://techterms.com/definition/programming_language.

The PHP Group . 2007. ¿Qué puede hacer PHP? [En línea] 2007. [Citado el: 28 de noviembre de 2015.] <http://php.net/manual/es/intro-whaticando.php>.

The PHP Group. 2006. ¿Qué es PHP? [En línea] 2006. [Citado el: 30 de noviembre de 2015.] <http://php.net/manual/es/intro-what-is.php>.

Third Light. 2015. Third Light. [En línea] 2015. [Citado el: 30 de octubre de 2015.] <http://www.thirlight.com>.

w3schools.com. 2013. w3schools.com. *CCS3 Introduction*. [En línea] 2013. [Citado el: 8 de diciembre de 2015.] http://www.w3schools.com/css/css3_intro.asp.

w3schools.com. 2014. w3schools.com. *HTML5 Introduction*. [En línea] 2014. [Citado el: 8 de diciembre de 2015.] http://www.w3schools.com/html/html5_intro.asp.

Anexos

Tabla 6. Escenario editar tipología del caso de prueba gestionar tipología

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo central
EC 1.1 Editar tipología exitosamente	El usuario inserta el nombre de la tipología	V	Muestra un mensaje que la tipología se modificó exitosamente y actualiza el listado de tipologías	Seleccionar opción Editar de una tipología del listado. Insertar el nombre de la tipología. Dar clic en el botón "Aceptar"
		Videoclip		
EC 1.2 Editar tipología con errores	El usuario inserta un nombre de la tipología no válido	I	Muestra un mensaje de error y se mantiene el formulario de editar tipología	Seleccionar opción Editar de una tipología del listado. Insertar el nombre de la tipología. Dar clic en el botón "Aceptar"
		Video-clip		
EC 1.3 Editar tipología con campos vacíos	El usuario no inserta el nombre de la tipología	NA	Muestra un mensaje de error y se mantiene el formulario de editar tipología	Seleccionar opción Editar de una tipología del listado. Insertar el nombre de la tipología. Dar clic en el botón "Aceptar"
EC 1.4 Editar tipología ya existente	El usuario inserta un nombre de tipología que ya está almacenada	V	Muestra un mensaje de error y se mantiene el formulario de editar tipología	Seleccionar opción Editar de una tipología del listado. Insertar el nombre de la tipología. Dar clic en el botón "Aceptar"
		Documental		
EC 1.5 Cancelar operación de edición	El usuario selecciona la opción de cancelar la operación	NA	Muestra el listado de tipologías	Editar de una tipología del listado. Dar clic en el botón "Cancelar"

Tabla 7. Escenario seleccionar campos de tipologías del caso de prueba gestionar tipología

Escenario	Descripción	Variable 2	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar campos de tipología exitosamente	El usuario selecciona los campos de la tipología	NA	Muestra un mensaje que la tipología se modificó exitosamente y muestra el listado de tipologías	Seleccionar opción Seleccionar campos de una tipología del listado. Seleccionar los campos. Dar clic en el botón "Aceptar"
		NA		
EC 1.2 La tipología está relacionada a una media	El usuario selecciona los campos de una tipología relacionada a una media	NA	Muestra un mensaje de error que dice que la tipología no puede ser modificada porque está en uso y muestra el listado de tipologías	Seleccionar opción Seleccionar campos de una tipología del listado. Seleccionar los campos. Dar clic en el botón "Aceptar"
EC 1.3 No se	El usuario no	NA	Muestra un mensaje de	Seleccionar opción Seleccionar

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

<i>selecciona ningún campo</i>	<i>selecciona ningún campo</i>		<i>error que dice que se debe seleccionar al menos un campo y muestra el listado de tipologías</i>	<i>campos de una tipología del listado. Seleccionar los campos. Dar clic en el botón "Aceptar"</i>
<i>EC 1.4 Cancelar operación</i>	<i>El usuario elige la opción cancelar</i>	NA	<i>Muestra el listado de tipologías</i>	<i>Seleccionar opción Seleccionar campos de una tipología del listado. Dar clic en el botón "Cancelar"</i>
<i>EC 1.5 Gestionar campos</i>	<i>El usuario elige gestionar los campos de tipologías existentes</i>	NA	<i>Muestra el listado de campos</i>	<i>Seleccionar opción Seleccionar campos de una tipología del listado. Dar clic en el botón "Gestionar campos"</i>

Tabla 8. Escenario eliminar del caso de prueba gestionar tipología

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo central
<i>EC 1.1 Eliminar tipología exitosamente</i>	<i>El usuario selecciona eliminar una tipología del listado</i>	NA	<i>Muestra un mensaje que la tipología se eliminó exitosamente y actualiza el listado de tipologías</i>	<i>Seleccionar opción Eliminar de una tipología del listado. Dar clic en el botón "Aceptar"</i>
<i>EC 1.2 Eliminar tipología relacionada a una media</i>	<i>El usuario selecciona eliminar una tipología del listado que está en uso</i>	NA	<i>Muestra un mensaje de error y se muestra el listado de tipologías</i>	<i>Seleccionar opción Eliminar de una tipología del listado. Dar clic en el botón "Aceptar"</i>
<i>EC 1.3 Cancelar operación</i>	<i>El usuario selecciona la opción de cancelar la operación</i>	NA	<i>Muestra el listado de tipologías</i>	<i>Seleccionar opción Eliminar de una tipología del listado. Dar clic en el botón "Cancelar"</i>

Tabla 9. Escenario eliminar múltiples del caso de prueba gestionar tipología

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo central
<i>EC 1.1 Eliminar tipologías seleccionadas exitosamente</i>	<i>El usuario selecciona las tipología a eliminar y selecciona la opción eliminar múltiples</i>	NA	<i>Muestra un mensaje que las tipologías se eliminaron exitosamente</i>	<i>Seleccionar las tipologías a eliminar y la opción Eliminar múltiples. Dar clic en el botón "Aceptar"</i>
<i>EC 1.2 Eliminar tipologías seleccionadas que estén en uso</i>	<i>El usuario selecciona tipologías que están relacionadas a una media</i>	NA	<i>Muestra un mensaje de error que dice que la tipología no puede ser eliminada porque está relacionada a una media y muestra el listado de tipologías</i>	<i>Seleccionar las tipologías a eliminar y la opción Eliminar múltiples. Dar clic en el botón "Aceptar"</i>

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

EC 1.3 Eliminar tipologías seleccionadas y que algunas estén en uso	El usuario selecciona algunas tipologías que están relacionadas a una media	NA	Muestra un mensaje de error que dice que la tipología no puede ser eliminada porque está relacionada a una media y muestra el listado de tipologías	Seleccionar las tipologías a eliminar y la opción Eliminar múltiples. Dar clic en el botón "Aceptar"
EC 1.4 No se selecciona ninguna tipología	El usuario no selecciona ninguna tipología	NA	Muestra un mensaje que notifica que se debe seleccionar al menos una tipología	Seleccionar las tipologías a eliminar y la opción Eliminar múltiples. Dar clic en el botón "Cancelar"
EC 1.5 Cancelar operación	El usuario selecciona la opción de cancelar la operación	NA	Muestra el listado de tipologías	Seleccionar las tipologías a eliminar y la opción Eliminar múltiples. Dar clic en el botón "Cancelar"

Tabla 10. Variables para el caso de prueba gestionar campos de tipologías

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Se debe introducir caracteres alfanuméricos, letras y números.
2	Tipo de dato	RadioButton	No	Marcar la opción
3	Longitud del campo	Campo de texto	No	Se debe introducir caracteres numéricos, números.
4	Descripción	Campo de texto	No	Se debe introducir caracteres alfanuméricos, letras y números.

Tabla 11. Sección insertar campo para el caso de prueba gestionar campos de tipologías

Escenario	Descripción	Variable 1	Variable 2	Variable 3	Variable 4	Respuesta del sistema	Flujo central
EC 1.1 Adicionar campo exitosamente	El usuario adiciona el nombre de la tipología	V	V	V	V	Muestra un mensaje que el campo se adicionó exitosamente y actualiza el listado de campos	Seleccionar opción Adicionar. Insertar el nombre del campo. Dar clic en el botón "Aceptar"
		Temática	Texto	255	Es la temática que trata una media		

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

EC 1.2 Adicionar campo con errores	El usuario adiciona datos no válidos	I	V	I	I	Muestra un mensaje de error y se mantiene el formulario de adicionar campo	Seleccionar opción Adicionar. Insertar el nombre del campo. Dar clic en el botón "Aceptar"
		Tem-atica	Texto	ercc	.%/		
EC 1.3 Adicionar campo con campos vacíos	El usuario no adiciona el nombre de la tipología	I	V	I	I	Muestra un mensaje de error y se mantiene el formulario de adicionar campo	Seleccionar opción Adicionar. Insertar el nombre del campo. Dar clic en el botón "Aceptar"
			Texto				
EC 1.4 Adicionar tipología ya existente	El usuario adiciona un nombre de tipología que ya está almacenada	V	V	V	V	Muestra un mensaje de error y se mantiene el formulario de adicionar campo	Seleccionar opción Adicionar. Insertar el nombre del campo. Dar clic en el botón "Aceptar"
		Álbum	Texto	255	Prueba		
EC 1.5 Cancelar operación de adición	El usuario selecciona la opción de cancelar la operación	NA	NA	NA	NA	Muestra el listado de campos	Seleccionar opción Adicionar. Dar clic en el botón "Cancelar"

Tabla 12. Sección editar campo para el caso de prueba gestionar campos de tipologías

Escenario	Descripción	Variable 1	Variable 3	Variable 4	Respuesta del sistema	Flujo central
EC 1.1 Editar campo exitosamente	El usuario insert el nombre del campo	V	V	V	Muestra un mensaje que el campo se editó exitosamente y actualiza el listado de campos	Seleccionar opción Editar de un campo del listado. Insertar el nombre del campo. Dar clic en el botón "Aceptar"
		Temáticas	255	Es la temática que trata una media		
EC 1.2 Editar campo con errores	El usuario adiciona datos no válidos	I	I	I	Muestra un mensaje de error y se mantiene el formulario de editar	Seleccionar opción Editar de un campo del listado.

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

		Temáticas	ercc	.%/	campo	Insertar el nombre del campo. Dar clic en el botón "Aceptar"
EC 1.3 Editar campo con campos vacíos	El usuario no adiciona el nombre del campo	l	i	i	Muestra un mensaje de error y se mantiene el formulario de adicionar campo	Seleccionar opción Editar de un campo del listado. Insertar el nombre del campo. Dar clic en el botón "Aceptar"
EC 1.4 Editar campo ya existente	El usuario inserta un nombre de campo que ya está almacenado	V	V	V	Muestra un mensaje de error y se mantiene el formulario de editar campo	Seleccionar opción Editar de un campo del listado. Insertar el nombre del campo. Dar clic en el botón "Aceptar"
		Álbum	255	Prueba		
EC 1.5 Cancelar operación de edición	El usuario selecciona la opción de cancelar la operación	NA	NA	NA	Muestra el listado de campos	Seleccionar opción Editar de un campo del listado. Dar clic en el botón "Cancelar"

Tabla 13. Sección eliminar campo para el caso de prueba gestionar campos de tipologías

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo central
EC 1.1 Eliminar campo exitosamente	El usuario selecciona eliminar un campo del listado	NA	Muestra un mensaje que el campo se eliminó exitosamente y actualiza el listado de campos	Seleccionar opción Eliminar de una campo del listado. Dar clic en el botón "Aceptar"
EC 1.2 Eliminar campo relacionada a una tipología	El usuario selecciona eliminar un campo del listado que está relacionado a una tipología	NA	Muestra un mensaje de error y se muestra el listado de campos	Seleccionar opción Eliminar de una campo del listado. Dar clic en el botón "Aceptar"
EC 1.3 Cancelar operación	El usuario selecciona la opción de cancelar la operación	NA	Muestra el listado de campos	Seleccionar opción Eliminar de un campo del listado. Dar clic en el botón "Cancelar"

Tabla 13. Sección eliminar múltiples campos para el caso de prueba gestionar campos de tipologías

Módulo para la Gestión de Tipologías Multimedia en PRIMICIA 2.0

Escenario	Descripción	Variable 2	Respuesta del sistema	Flujo central
<i>EC 1.1 Eliminar campos seleccionados exitosamente</i>	<i>El usuario selecciona las tipología a eliminar y selecciona la opción eliminar múltiples</i>	<i>NA</i>	<i>Muestra un mensaje que los campos se eliminaron exitosamente</i>	<i>Seleccionar los campos a eliminar y la opción Eliminar múltiples. Dar clic en el botón "Aceptar"</i>
<i>EC 1.2 Eliminar campos seleccionados que estén relacionados a una tipología</i>	<i>El usuario selecciona tipologías que están relacionadas a una media</i>	<i>NA</i>	<i>Muestra un mensaje de error que dice que el campo no puede ser eliminado porque está relacionada a una tipología y muestra el listado de campos</i>	<i>Seleccionar los campos a eliminar y la opción Eliminar múltiples. Dar clic en el botón "Aceptar"</i>
<i>EC 1.3 Eliminar campos seleccionados y que algunos estén relacionados a una tipología</i>	<i>El usuario selecciona algunas tipologías que están relacionadas a una media</i>	<i>NA</i>	<i>Muestra un mensaje de error que dice que el campo no puede ser eliminado porque está relacionada a una tipología y muestra el listado de campos</i>	<i>Seleccionar los campos a eliminar y la opción Eliminar múltiples. Dar clic en el botón "Aceptar"</i>
<i>EC 1.4 No se selecciona ningún campo</i>	<i>El usuario no selecciona ningún campo</i>	<i>NA</i>	<i>Muestra un mensaje que notifica que se debe seleccionar al menos un campo</i>	<i>Seleccionar los campos a eliminar y la opción Eliminar múltiples. Dar clic en el botón "Cancelar"</i>
<i>EC 1.5 Cancelar operación</i>	<i>El usuario selecciona la opción de cancelar la operación</i>	<i>NA</i>	<i>Muestra el listado de tipologías</i>	<i>Seleccionar las tipologías a eliminar y la opción Eliminar múltiples. Dar clic en el botón "Cancelar"</i>