



Universidad de las Ciencias
Informáticas

Proveedor de ficheros de imagen, audio y vídeo para los portales de los medios de prensa cubanos

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

Autores:

Pedro Pablo Bravo Hernández

Felix Pérez Bubaire

Tutores:

Ing. Yankiel Aparicio Pérez

Ing. Paul Rodríguez Leyva

La Habana, Junio 2013

Declaramos ser autores de este trabajo y autorizamos a la Facultad 1 y a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales del mismo, con carácter exclusivo.

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

Firma del Autor

Firma del Autor

Firma del Tutor

Firma del Tutor

Pedro Pablo Bravo Hernández

Correo: ppbravo@estudiantes.uci.cu

La Habana, Cuba

Felix Pérez Bubaire

Correo: fbubaire@estudiantes.uci.cu

La Habana, Cuba

Ing. Yankiel Aparicio Pérez

Ingeniero en Ciencias Informáticas

Correo: yaparicio@uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Ing. Paul Rodríguez Leyva

Ingeniero en Ciencias Informáticas

Correo: pleyva@uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba



*“...De las glorias pasadas se sacan fuerzas para
alcanzar las glorias futuras...”*

José Martí

Pedro Pablo:

A mi abuelo y a mi bisabuela.

*A todos los que sueñan y crean, los que aman y construyen, a los que perseveran
y conquistan.*

Félix:

A mis padres y mis hermanos.

Pedro Pablo:

Agradecimiento especial a la Revolución Cubana por existir y hacer posible la creación de esta universidad donde nos formamos docente, técnica y moralmente para la vida.

A los profesores que día a día brindan lo mejor de ellos para que esto sea posible, principalmente a la ing. Ailyn Estrada, ing. Yankiel Aparicio y Paul Rodríguez Leyva.

Agradezco a mis familiares, a los que ya no están; artífices de de este sueño.

A mi madre materializadora de esta obra y responsable de hacerme un hombre de valores morales y sentimientos puros. A mi abuela Olga y otros.

A mis amigos Grettel Barrio, Lázaro Placeres, Ernesto Antón Peña, amigos del preuniversitario, a mi equipo de balonmano, basket, y voleyball.

A mi novia por ser tan comprensiva a lo largo de este trabajo.

A mi compañero de tesis ejemplo de perseverancia y lucha por lo que uno quiere alcanzar en la vida.

A todos aquellos que de una compartieron conmigo buenos y malos momentos.

A todos gracias.

Félix:

A mi madre por ser lo persona más hermoso que me ha dado la vida, por su apoyo incondicional y la confianza que tuvo siempre en mí.

A mi padre por haberme educado y criado como un hombre de bien.

A mis hermanos por ser fuentes de inspiración para mí.

A mi familia en general que siempre me ha apoyado y siempre confiaron en mí y sabían que al final yo podía lograrlo.

A mi compañero de tesis que ha demostrado ser una persona muy capaz.

A los amigos que siempre han estado ahí en las buenas y las malas a los cuales les deseo éxitos en la vida como futuros ingenieros.

A nuestro tutor Yankiel por su gran ayuda, paciencia y dedicación.

De manera en general a todos los que de una manera u otra han tenido que ver en mi formación como ingeniero.

Resumen

Hoy en día los portales de los medios de prensa cubanos presentan dificultades que interfieren en la rapidez y calidad de las publicaciones en Internet. En la Universidad de las Ciencias Informáticas se desarrolló el Sistema de Gestión Documental de Media para fortalecer la visibilidad, capacidad de respuesta, inmediatez de las publicaciones y cobertura en tiempo real logrando un eficiente uso de los medios disponibles a través de los recursos de hipertexto y multimedia. En el presente trabajo se desarrolló un sistema para centralizar los archivos de media garantizando su publicación en los portales de los medios de prensa cubanos. Para la construcción de dicho sistema se usó el *framework* de desarrollo Symfony 2.1, los lenguajes de desarrollo PHP 5.3, JavaScript 1.8, CSS 3, HTML 5 y el Entorno Integrado de Desarrollo Netbeans 7.3. Como herramienta CASE se utilizó el Visual Paradigm 8.0, el Sistema Gestor de Base de Datos PostgreSQL 9.1 y la metodología de desarrollo de software OpenUp. Como resultado se logró el sistema Proveedor de ficheros de media permitiendo la centralización de los recursos de media. El cual permitió mitigar la duplicación de contenido y contribuyó a difundir y mostrar la verdad de Cuba más allá de las fronteras de la isla, logrando una independencia tecnológica que responderá a las principales necesidades de la prensa en Cuba.

Palabras clave: centralización de media, difusión de archivos, Proveedor de media.

Resumen	8
ÍNDICE DE TABLAS	3
ÍNDICE DE FIGURAS	4
INTRODUCCIÓN	5
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	8
1.1 Sistemas Homólogos.....	8
1.1.1 Sistemas proveedores de audio y vídeo.....	9
1.1.2 Sistemas proveedores de imágenes	11
1.2 Estudio de Estándares	13
1.3 Metodología, tecnologías y herramientas utilizadas en la implementación del sistema...17	
1.3.1 Metodología de desarrollo de software.....	17
1.3.2 Framework de Desarrollo	17
1.3.3 Lenguajes de Programación	19
1.3.4 Sistema Gestor de Bases de Datos	21
1.3.5 Tecnologías del servidor	21
1.3.6 Herramientas	22
1.3.7 Servidor Web Apache	23
1.3.8 Entorno Integrado de Desarrollo	23
Conclusiones parciales.....	24
CAPÍTULO 2: CARACTERIZACIÓN DEL SISTEMA PROPUESTO	25
2.1 Modelo de Dominio.....	25
2.2 Descripción del Sistema	26
2.3 Requerimientos del software	26
2.3.1 Requisitos Funcionales	26
2.3.2 Requisitos no Funcionales	31
2.4 Definición de los Casos de Uso del Sistema.....	32
2.4.1 Identificación de los Actores del sistema	32
2.4.2 Diagrama de caso de uso del Sistema	33
2.4.3 Descripción de casos de uso del sistema.....	33
Conclusiones parciales.....	40
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	41

3.1 Patrones Arquitectónicos	41
3.2 Patrones de Diseño	42
3.2.1 Patrones GRASP	42
3.2.2 Patrones GOF	43
3.3 Diagramas de clases del análisis y del diseño	43
3.3.1 Modelo de Análisis	44
3.3.2 Diagrama de Clase de Análisis	44
3.3.3 Modelo de diseño	45
3.3.4 Diagramas de Interacción	47
3.3.5 Modelo de Datos	48
3.3.6 Modelo de Despliegue	49
Conclusiones parciales	50
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS	52
4.1 Modelo de Implementación del sistema	52
4.1.1 Diagrama de componentes	52
4.2 Estándares de codificación	54
4.2.1 Funciones y Métodos	54
4.2.2 Variables	54
4.2.3 Constantes	55
4.3 Interfaces principales del sistema	55
4.4 Pruebas de software	57
4.4.1 Pruebas funcionales	57
4.4.2 Pruebas de Seguridad	59
4.4.3 Pruebas de Rendimiento	61
Conclusiones parciales	62
Conclusiones Generales	63
Recomendaciones	64
Referencias Bibliográficas	65
Glosario de términos	69
Anexos	70
Anexo 1: Entrevista	70
Anexo 2: Especificación de Requisitos de Software	71
Anexo 3: Especificación de caso de uso	73
Anexo 4: Diagrama de Clase de Análisis	79

Anexo 5: Diagramas de clases del Diseño	80
Anexo 6: Diagramas de Interacción.....	85
Anexo 7: Pruebas Funcionales.....	86

ÍNDICE DE TABLAS

Tabla 1: Requisitos funcionales del sistema	26
Tabla 2: Descripción del RF 16: Subir Imagen	27
Tabla 3: Descripción del RF36: Realizar búsqueda simple del contenido.....	28
Tabla 4: Descripción del RF 37: Realizar búsqueda avanzada del contenido.....	29
Tabla 5: Actores del sistema	32
Tabla 6: Descripción de CU 2: Gestionar Búsqueda.....	33
Tabla 7: Descripción de CU 7: Gestionar Permisos.....	36
Tabla 8: Descripción de CU 3: Visualizar material	38
Tabla 9: Descripción de CU 4: Descargar material	39
Tabla 10: Estereotipos de clases del análisis	44
Tabla 11: Estereotipos Web del Diseño	45
Tabla 12: SC Autenticar usuario	57
Tabla 13: Resultado de las pruebas	59
Tabla 14: Escenario para el CU Autenticar usuario	59

ÍNDICE DE FIGURAS

Figura 1: Representación descriptiva de un recurso RDF	16
Figura 2: Diagrama de Clases del Modelo de Dominio	25
Figura 3: Diagrama de CU del sistema del Usuario Administración.....	33
Figura 4: Diagrama del Patrón Arquitectónico MVC	42
Figura 5: Diagrama de Clases del Análisis Gestionar Búsqueda.....	45
Figura 6: Diagrama de clases de estereotipos web Búsqueda avanzada.....	46
Figura 7: Diagrama de clases de estereotipos web Subir vídeo	47
Figura 8: Diagrama de Colaboración del CU Autenticar usuario.....	48
Figura 9: Modelo físico de datos para PostgreSQL	49
Figura 10: Modelo de Despliegue	50
Figura 11: Diagrama de componentes	53
Figura 12: Ejemplo del estándar de código en <i>lowerCamelCase</i>	54
Figura 13: Interfaz Principal	55
Figura 14: Interfaz de Galería de media.....	56
Figura 15: Interfaz Búsqueda avanzada.....	56
Figura 16: Resultados de las pruebas de rendimiento.....	61

INTRODUCCIÓN

El auge de la sociedad de la información, ha generado la incorporación progresiva de los profesionales de la información y la documentación en los nuevos espacios de tratamiento, gestión y difusión de información digital (Pérez, 2007).

La gestión de contenidos es una actividad que ha cobrado auge en el entorno académico y profesional, y dentro de ella, la “gestión de contenidos Web” donde los contenidos son gestionados en portales Web los cuales distribuyen de forma uniforme y centralizada la información proveniente de diversas fuentes, implementan mecanismos de navegación, integran aplicaciones e incluyen mecanismos de colaboración para el conjunto de usuarios (comunidad) a los que sirve de marco de trabajo (Tramullas, 2005).

En Cuba, debido al creciente desarrollo de portales y la digitalización de los medios de prensa, se trabaja actualmente en la difusión de grandes volúmenes de información en ficheros de imagen, audio y vídeo que forman parte del patrimonio nacional. En la Universidad de las Ciencias Informáticas se ha desarrollado el Sistema de Gestión Documental de Media (Núcleo de la Mediateca). Esta solución permite generar automáticamente diferentes perfiles a partir del fichero original y modela un conjunto de procesos orientados a la descripción y valoración de los contenidos. A pesar de que el sistema garantiza la gestión de los recursos multimedia dentro de las entidades presenta un conjunto de dificultades con la publicación de los archivos de media, las cuales fueron identificadas con la realización de una entrevista al cliente (Ver Anexo 1) donde se describieron las siguientes deficiencias:

- ✓ Los servidores web se encuentran sobrecargados de ficheros de imagen, audio y vídeo dedicando recursos para responder peticiones de los usuarios bajo demanda de estos ficheros cuando pudieran ser destinados a servir contenido web.
- ✓ La información que se publica se encuentra en ocasiones duplicada. Lo cual constituye una experiencia negativa para el usuario pues obtiene los mismos resultados repetidos para diferentes búsquedas realizadas.
- ✓ Existen deficiencias en la representación de los contenidos bajo estándares (NewsML, RDF y JSON) reconocidos internacionalmente.
- ✓ No se garantiza la autenticidad de los contenidos publicados.
- ✓ Los contenidos de media publicados no se encuentran centralizados.

Si bien los elementos mencionados interfieren en la rapidez y calidad de las publicaciones en Internet, en este sentido se hace necesario fortalecer: la visibilidad, la capacidad de respuesta, la inmediatez de las publicaciones y la cobertura en tiempo real logrando un amplio y eficiente uso de los medios disponibles a través de los recursos de hipertexto y multimedia. La solución debe estar orientada a mitigar la

descentralización de los recursos de media, dígame imagen, audio y vídeo que en la actualidad existe en los portales de los medios de prensa cubanos.

Se deriva del análisis anterior el siguiente **problema de investigación**: ¿Cómo lograr la centralización de los archivos de media para su publicación en los portales de los medios de prensa cubanos?

Se define como **objeto de estudio**: la gestión de contenidos de media en Internet. Ajustado al **campo de acción**: la gestión de contenidos de imagen, audio y vídeo en los portales de los medios de prensa cubanos.

Dirigiendo el sentido de la investigación el **objetivo general**: Desarrollar un sistema informático para centralizar la publicación de los archivos de media en los portales de los medios de prensa cubanos.

Desglosado cuidadosamente en **objetivos específicos**:

- ✓ Desarrollar una sistematización teórica sobre los sistemas de gestión de contenidos de imagen, audio y vídeo en Internet.
- ✓ Efectuar el Análisis y el Diseño del sistema proveedor de imagen, audio y vídeo para los portales de los medios de prensa cubanos.
- ✓ Implementar el sistema proveedor de imagen, audio y vídeo para los portales de los medios de prensa cubanos.
- ✓ Validar las funcionalidades del sistema proveedor de imagen, audio y vídeo para los portales de los medios de prensa cubanos.

Con el objetivo de dar cumplimiento a los objetivos planteados se utilizaron los siguientes métodos científicos que determinaron el desarrollo de la investigación:

Métodos Teóricos:

Análítico-Sintético: Método utilizado con el objetivo de analizar, comprender y sintetizar la información documentada relacionada con la gestión de contenidos de media. Este análisis permitió generar una propuesta de solución dando respuesta al problema existente en los medios de prensa cubanos.

Métodos Empíricos:

Entrevista: proporcionó adquirir información clara a través de la recopilación de datos para elaborar la aplicación y determinar las características necesarias y funcionalidades para el desarrollo del sistema. Simplificó la realización de las tareas de diseño y la obtención de los requisitos del sistema (Ver Anexo 1).

El presente trabajo se encuentra estructurado por: introducción, cuatro capítulos, conclusiones generales, recomendaciones, referencias bibliográficas y bibliografía de las fuentes consultadas durante el desarrollo del trabajo, el glosario de términos y los anexos que complementan el cuerpo del trabajo.

Capítulo 1: Fundamentación Teórica

Se precisan los diferentes conceptos asociados al dominio del problema que respalda la investigación y desarrollo de la propuesta de solución, a través del estudio y análisis de soluciones existentes en el ámbito nacional e internacional. Se detallan las principales características relacionadas con la gestión de contenidos y su relación existente con la publicación de los contenidos de media. Se describen cada una de las diferentes tecnologías y herramientas a utilizar en el desarrollo del sistema.

Capítulo 2: Caracterización del Sistema Propuesto

Se definen las diferentes características del componente a desarrollar. También se identifican los requisitos funciones y no funcionales del sistema, así como la descripción de los mismos. Además, se elabora el diagrama de casos de uso del sistema.

Capítulo 3: Análisis y Diseño del Sistema

Se describe la arquitectura de *software* y los patrones arquitectónicos y de diseño que se utilizarán en el diseño e implementación de la solución propuesta, modelándose la misma a través de los diagramas de clases del diseño y colaboración, además se representa el diagrama de despliegue.

Capítulo 4: Implementación y Prueba

En este capítulo se muestran los resultados de las pruebas correspondientes del sistema durante su ciclo de vida. Se representa el diagrama de componentes y los diseños de casos de prueba basados en Casos de Uso.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La sociedad es caracterizada como la nueva era de la información digital, dado el alto grado de desarrollo que experimenta la generación del contenido científico-técnico el cual ha crecido a un ritmo acelerado. Este gran cúmulo de datos está disponible en diferentes formatos como: imagen, audio y vídeo, ejerciendo gran influencia en la ciencia y el conocimiento.

Con el objetivo de facilitar el acceso a la información y brindar servicios que posibiliten a los usuarios el manejo de datos e informaciones, siendo este uno de los aspectos más importantes para cualquier entidad, se engloban un grupo de actividades como: recolección, almacenamiento, recuperación y propagación hacia distintas esferas y personas, así como su uso para varias actividades dentro de una organización. Como respuesta a lo antes dicho surge la llamada gestión de contenidos la cual se asocia a un nuevo método para el diseño y desarrollo de portales Web que conlleva a (Galán, y otros, 2005):

- ✓ La inclusión de elementos digitales de diferentes tipos (textuales, gráficos y sonoros).
- ✓ El desarrollo de forma cooperativa y descentralizada.
- ✓ El paso de un modelo estático a otro mucho más dinámico.
- ✓ La reutilización de los contenidos.

Este concepto se asocia también con los nuevos enfoques de Intranet, donde el mayor peso recae en los aspectos relacionados con la identificación de recursos de información internos y externos, su valoración, gestión y tratamiento eficiente. A esto se le une, la necesidad de utilizar tecnologías de la información y sistemas informáticos para el almacenamiento y distribución de información de naturaleza textual.

Debido a los aspectos mencionados la estandarización de la pluralidad tecnológica representa un papel esencial en estos procesos. En Cuba las entidades que tienen a su cargo la manipulación de información, hoy afrontan el reto de alcanzar la difusión del patrimonio que constituyen los ficheros de imagen, audio y vídeo. En la Universidad de las Ciencias Informáticas surge la necesidad de crear un Proveedor de ficheros de media, permitiendo la creación de una herramienta para centralizar los contenidos de media provenientes del Sistema de Gestión Editorial.

En el presente capítulo se precisan elementos teóricos que sustentan la investigación y el desarrollo del tema propuesto, a través del estudio y análisis de soluciones existentes. Se tratan las principales definiciones relacionadas con la gestión de contenidos de media. Se describen las tecnologías, herramientas, lenguajes y metodologías de desarrollo utilizadas para el análisis y personalización del sistema.

1.1 Sistemas Homólogos

Para determinar las condiciones en que se encuentra actualmente la gestión de contenidos de media de

imagen, audio y vídeo, se efectuó la caracterización de un conjunto de aplicaciones similares existentes dentro y fuera del país, donde se incluyen en el ámbito internacional: YouTube, Vimeo, Flickr y Picasa; y en el nacional: TeVeo. A continuación se expone el concepto de proveedor para lograr un mejor entendimiento de la función que desempeñan estos sistemas:

Proveedor de servicios de media: Es una herramienta que permite indexar contenidos y proveer interfaces para la recuperación de información (Lagoze, 2012).

Características principales:

- ✓ La selección y validación de proveedores de datos.
- ✓ El control de flujo de la información.
- ✓ La recolección basada en que los datos y metadatos son transferidos desde la fuente remota al destino en el cual se realizarán los servicios de búsqueda.
- ✓ La detección de contenido duplicado.
- ✓ Interfaz de usuario para la interacción con el público.

1.1.1 Sistemas proveedores de audio y vídeo

YouTube

Es un sitio web en el cual los usuarios pueden subir y compartir vídeos. Brinda la posibilidad de alojar vídeos personales de manera sencilla. Es la comunidad de vídeos más famosa en todo el mundo, en donde se pueden subir las filmaciones propias de cada cual, comentar vídeos, suscribirte a los canales que prefieras y designar algunos como favoritos.

Características principales (YouTube, 2013):

- ✓ **Búsqueda:** Sobre la búsqueda de vídeos y vista de alta resolución YouTube mantiene una logística que permite localizar cualquier vídeo por medio de las etiquetas de metadatos, títulos y descripciones que los usuarios asignan a sus vídeos, pero cuyo uso está restringido únicamente a los vídeos alojados en YouTube.
- ✓ **Calidad de vídeo:** Todos los vídeos deben ocupar un espacio en disco no mayor a 2 GB y deben tener una duración menor o igual a 15 minutos, con una tolerancia extra de 59 segundos de grabación, es decir, una duración total de 15:59 segundos. La totalidad de los vídeos son convertidos a resoluciones de 320×240 y 480×360 píxeles, y a 30 fotogramas por segundo, aunque los vídeos alojados antes de marzo de 2008 sólo están disponibles en la resolución más baja.
- ✓ **Restricción de copias:** Para evitar copias de los archivos de vídeo, éstos están distribuidos en formato *flash* (FLV), propiedad de la empresa *Adobe Flash*.

- ✓ **Ventana principal:** En la ventana principal se pueden encontrar varias clasificaciones de vídeos como los más destacados, los recomendados en distintas temáticas, vídeos patrocinados o los más populares. Además se encuentra una barra de búsquedas que facilita la tarea de realizar las búsquedas de los vídeos y un botón que posibilita la creación o acceso a una cuenta gratuita en la web. Cabe aclarar, que todos los vídeos de la web están ordenados por categorías, de manera que existe la opción de navegar por categorías en vez de hacerlo a través de la barra de búsqueda.
- ✓ **Lista de reproducción:** Brinda la posibilidad de agregar cualquier vídeo a una lista de reproducción, de esta manera, solo se seleccionan los vídeos de la lista y estos se reproducen uno a continuación del otro.
- ✓ **Subir vídeo:** De acuerdo con el manual de usuario: Acepta casi cualquier formato de vídeo para las subidas, sin embargo es recomendable usar el formato de vídeo MPEG4. La duración máxima es de 10 minutos y el tamaño no debe ser mayor a 1 GB. Independientemente del formato en el que fue subido, YouTube procesa todos los vídeos y los convierte al formato *Adobe Flash Vídeo* (cuyos altos niveles de compresión hacen que el archivo sea más "ligero" a la hora de subirse y reproducirse). El tiempo de procesamiento varía dependiendo del tamaño y formato del vídeo original y puede durar entre unos minutos y ocho horas. Una vez procesado, el vídeo está listo para ser compartido con los demás usuarios o incluido en cualquier blog o página de Internet.

Vimeo

Es una red social de Internet que permite compartir y almacenar vídeos digitales para que los usuarios comenten en la página de cada uno de ellos. Los usuarios deben estar registrados para subir vídeos, crear su perfil, cargar avatares, comentar y armar listas de favoritos (Vimeo, 2009).

Función de Vimeo: Ofrece un servicio que consiste en exhibir y acceder productos audiovisuales. Estos servicios son:

- ✓ Controlar notificaciones.
- ✓ Cuenta con un muro para cada perfil que ayuda a la comunicación entre usuarios.
- ✓ Facilita la búsqueda de contenido audiovisual.
- ✓ Cada vídeo puede ser incorporado en un blog si se desea.
- ✓ Segmentación del material audiovisual: Vídeos institucionales y académicos. Proyectos musicales. Proyectos cinematográficos. Vídeos de campañas políticas, animados y cómicos.

Aspectos relevantes:

- ✓ Brinda la posibilidad de importar contactos solo de *Gmail.com* y *Yahoo Mail*.
- ✓ El tamaño ocupado por los vídeos a subir no puede exceder los 500 MB.

- ✓ Permite descargas en formato WMV.
- ✓ La interfaz no soporta el idioma español.
- ✓ Tiene opciones para compartir vídeos a otros sitios por ejemplo desde el perfil de Facebook se pueden visualizar videos que se encuentran en Vimeo.
- ✓ Posibilita descargar los ficheros originales.
- ✓ Permite acceder al código embebido de los vídeos existentes.
- ✓ Posee opciones para decidir quién va a visualizar los vídeos (con mayor o menor grado de privacidad).

TeVeo

Es un sitio web que ofrece servicios de audio y vídeo bajo demanda del Instituto Cubano de Radio y Televisión (ICRT), el cual permite al usuario el acceso a contenidos multimedia de forma personalizada. Ofrece la posibilidad de solicitar y visualizar los contenidos en el momento deseado optimizando la interacción de los usuarios y el sistema (TeVeo, 2012).

Características principales:

- ✓ Los archivos son organizados según su actualización, contenido y la popularidad que hayan alcanzado, en bloques como: recomendaciones, lo más reciente y lo más popular en la aplicación.
- ✓ Permite realizar búsquedas simples mostrando la información de los archivos de audio y vídeo la cual está también disponible en formato RSS adecuado para su consumo a través de un conferenciante de noticias.
- ✓ En conclusión TeVeo brinda servicios de audio y vídeo en el cual se puede realizar una búsqueda de los archivos de media antes mencionados, de fácil uso y posee una interfaz sencilla y fácil de comprender.

1.1.2 Sistemas proveedores de imágenes

Flickr

Permite almacenar, ordenar, buscar, vender y compartir fotografías y vídeos en línea. Actualmente cuenta con una importante comunidad de usuarios que comparte las fotografías y vídeos creados por ellos mismos. Esta comunidad se rige por normas de comportamiento y condiciones de uso que favorecen la buena gestión de los contenidos. La popularidad de Flickr se debe fundamentalmente a su capacidad para administrar imágenes mediante herramientas que permiten al autor etiquetar sus fotografías, explorar y comentar las imágenes de otros usuarios (Cuervo, 2009).

El sistema permite hacer búsquedas de imágenes por etiquetas, por fecha. Otras de las funcionalidades son los canales RSS y *Atom*, la *API* que permite a desarrolladores independientes crear servicios y

aplicaciones vinculados a Flickr. El servicio se basa en las características habituales del HTML y el *HTTP*, permitiendo que sea usable en múltiples plataformas y navegadores. La interfaz de etiquetación y edición de texto utiliza *AJAX*, que también es compatible con la gran mayoría de los navegadores.

La *API* de Flickr expone diferentes mecanismos para realizar las peticiones tales como *REST*, *XMLRPC* y *SOAP* y para los formatos de resultados incluye XML, JSON y PHP. Flickr incluye una serie de *API kits* entre las que se encuentra *Photo Upload API*. Estas *API kits* se desarrollan para facilitar la integración de Flickr en diferentes plataformas y lenguajes como por ejemplo: C, Delphi, .NET, Java, Perl, PHP, Python y Ruby. Flickr no brinda soporte a estas *API kits* por lo que son utilizadas a riesgo de los desarrolladores.

Características principales (Social Media Defined, 2009):

- ✓ Cargas limitadas de fotos (20 MB por foto).
- ✓ Cargas limitadas de vídeos (máximo de 90 segundos, 500 MB cada uno).
- ✓ Capacidad de mostrar vídeo en Alta Definición.
- ✓ Cantidad limitada de almacenamiento.
- ✓ Cantidad limitada de ancho de banda.
- ✓ Archivado de imágenes originales en alta resolución.
- ✓ Capacidad de reemplazar una foto.
- ✓ Posibilidad de publicar cualquier foto o vídeo de hasta 60 murales de grupos.
- ✓ Exploración y uso compartido sin anuncios publicitarios.
- ✓ Posibilidad de ver estadísticas de referenciadores y conteo de visitas.

Picasa

Picasa es un programa gratuito que permite visualizar, editar imágenes y crear álbumes de fotografías personalizados. Entre las características de Picasa más importantes, destaca la posibilidad de añadir etiquetas en fotos de personas con nombres, así como agrupar fotos que contengan caras similares. Además, Picasa se integra con *Google Maps*, de tal forma que ya no es necesario instalar *Google Earth* para añadir una localización geográfica en las fotos. Gracias a estas funcionalidades, se podrá recordar en un futuro dónde fueron tomadas exactamente las fotos (PICASA, 2012).

Características:

- ✓ Permite añadir texto como marca de agua a las fotos.
- ✓ Actualiza automáticamente las nuevas fotos añadidas así como los cambios en las existentes (Ajustes, edición, información de etiquetas, etc.).
- ✓ Permite la carga en línea de las imágenes en los álbumes web de Picasa.
- ✓ Sincroniza automáticamente las fotos editadas en los álbumes web de Picasa. Actualiza automáticamente las nuevas fotos añadidas así como los cambios en las existentes (Ajustes,

edición, información de etiquetas, etc). No actualiza los cambios en el orden de los álbumes, de fotos, cambios en las carpetas o en los nombres de archivo.

- ✓ La pantalla completa, de una ventana o de la cámara Web del equipo, imágenes fijas y videos.
- ✓ Pre visualización de fotos con Picasa Photo Viewer. Utilidad para mostrar las fotos como presentación a pantalla completa permitiendo la rotación de las mismas, uso del zoom y la calificación de las fotos mientras se visualizan, así como la orden para cargar las fotos en los álbumes Web.

Luego de haber realizado el análisis de algunas de las soluciones similares existentes en todo el mundo, referente a los sistemas proveedores de media se pudo adquirir conocimientos sobre este tema y sus tendencias actuales, dichas aplicaciones simulan un proceso equivalente al que se desea desarrollar, por lo que se pudieron estudiar los aspectos positivos y negativos de las mismas para tomar lo mejor de ellas y tener en cuenta las desventajas que poseen para tratar de mitigarlas. Esta a su vez no pueden ser utilizadas porque son propietarias y las que se requieren deben ser libres y multiplataforma, pero poseen un conjunto de características que servirán para la posterior implementación del sistema. A continuación se muestran un conjunto de características que poseen para cada archivo de media (imagen, audio y vídeo) entre las que se encuentran:

Característica General: el sistema deberá realizar las búsquedas a partir de las etiquetas de metadatos que cargan la información del contenido que se requiere.

Imagen:

- ✓ Soportar solo formatos como: jpg, jpeg y png.

Audio:

- ✓ Soportar solo formatos como: ogg.

Vídeo:

- ✓ Los vídeos no excederán los 15:59min de duración.
- ✓ La capacidad de los vídeos a subir no puede exceder los 500 MB.
- ✓ Permitir descargas en formato ogg, vp8, webm.
- ✓ Soportar solo formatos como: ogg, vp8, webm.

1.2 Estudio de Estándares

Un estándar establece un sistema común de terminología y definiciones para documentar datos, normando la forma de describirlos. Idealmente las estructuras y definiciones de metadatos deben tener su referencia en un estándar (Maestros del web, 2004).

Características:

- ✓ Los estándares permiten la localización rápida de ciertos elementos. Si se utiliza un estándar, encontrar la información específica en un catálogo de metadatos será mucho más fácil que si no se utiliza ninguno.
- ✓ Posibilitan búsquedas automatizadas.
- ✓ Ayudan a minimizar la duplicación de esfuerzos en la elaboración, recolección, procesamiento o distribución de la información.
- ✓ Un sitio basado en estándares web es más fácil de mantener y actualizar, el código es más simple, de esta forma se elimina la dependencia de un solo desarrollador.

NewsML

Es un estándar multimedia de intercambio de noticias del Consejo de Telecomunicaciones de Prensa Internacional (*IPTC*, por sus siglas en inglés). Como parte de las normas de la familia *IPTC G2*, *NewsML-G2* actúa como una dotación para una o más noticias (por ejemplo, un artículo de texto, una foto o un vídeo clip) o enlaces a noticias, y contiene metadatos para describir la relaciones entre los elementos. En la actualidad, una segunda generación de esta norma está en proceso de desarrollo bajo el nombre *NewsML-G2 Architecture (NAR)*. El objetivo básico de la *NAR* es proporcionar un único modelo genérico para el intercambio de todo tipo de información de interés periodístico, lo que proporciona un marco para una futura familia de normas de intercambio de noticias de *IPTC* (International Press Telecommunications Council, 2012).

Es un método de transmitir, no sólo el contenido de las noticias sino también los datos que describen el contenido de una manera abstracta (metadatos), información acerca de cómo manejar las noticias de una manera adecuada (metadatos de gestión de noticias), la presentación de la información y finalmente indaga acerca de la transferencia técnica en sí. *NewsML-G2* ofrece formatos de intercambio para:

- ✓ Noticias generales: noticias textuales, artículos, fotos, gráficos, audio y vídeo.
- ✓ Un mecanismo flexible para embalaje de noticias de una manera estructurada.

Las ventajas que posee incluyen la mejora de las capacidades de búsqueda y opciones de presentación, navegación e integración con los sistemas Sistema de Administración de Contenidos (*CMS*, por sus siglas en inglés) y una amplia gama de posibles aplicaciones. Esto puede ser un banco de pruebas sobre cómo cerrar la brecha entre la interoperabilidad de metadatos de sintaxis diferentes (Troncy, 2007).

RSS

Really Simple Syndication (*RSS*) es uno de los sistemas más populares para la difusión de contenidos que existen en la web. Facilita la gestión y publicación de información y noticia web. Es una forma estandarizada de distribución de la información de las páginas web a los lectores a través de los Canales *RSS* la cual posee un ciclo periódico de actualización. Los lectores pasan a tener una herramienta útil para

mantenerse informado sobre las noticias web que le resultan de interés, conservando y almacenando toda la información en un solo lugar que se actualiza de manera automática. Brinda al lector un importante ahorro y tiempo en la lectura, ya que sólo con abrir el lector RSS (ya sea un programa, el navegador o un lector web *online*), el usuario podrá ver cuáles son las últimas actualizaciones y noticias que han publicado las diferentes páginas web a las que está suscrito (RSS, 2012).

Características:

- ✓ Las páginas web y blogs distribuyen a través de los canales RSS las últimas actualizaciones de aquellas páginas web que son de su interés.
- ✓ La decisión está del lado del usuario ya que es quien elige a qué páginas web suscribirse y cuando darse de baja de estas páginas web.
- ✓ Está libre de *SPAM*, porque no tienes que dar tu correo electrónico. Esto no ocurre con suscripciones por correo electrónico, en las que además de recibir noticias, podrías recibir también *SPAM* u otra información no deseada. Cuando estás suscrito a las fuentes RSS de una página web, no recibirás otra información que la que se publique en las páginas web que son de tu interés
- ✓ Recibir las fuentes de las páginas web favoritas es totalmente gratuito.

RDF

Resource Description Framework (RDF) es un lenguaje para la representación de la información sobre los recursos en la Web, particularmente dirigido para la representación de los metadatos. Es una base para procesar metadatos; proporciona interoperabilidad entre aplicaciones que intercambian información legible entre las máquinas. Se destaca por la facilidad para habilitar el procesamiento automatizado de los recursos web (Lee, 2009). Puede utilizarse en distintas áreas de aplicación; por ejemplo:

- ✓ En la recuperación de recursos para proporcionar mejores prestaciones a los motores de búsqueda.
- ✓ En la catalogación para describir el contenido y las relaciones de contenido disponibles en un sitio Web o una biblioteca digital particular, por los agentes de *software* inteligentes para facilitar el intercambio y para compartir conocimiento.
- ✓ En la calificación de contenido.
- ✓ En la descripción de colecciones de páginas que representan un "documento" lógico individual.

Es un modelo formal para la representación de las pertenencias y los valores de los recursos de información. Las propiedades se pueden entender como atributos de los recursos y en este sentido corresponden a los pares tradicionales de atributo-valor. Se puede relacionar también con el diseño orientado a objetos donde los recursos corresponden a objetos y las propiedades corresponden a ejemplos de variables. El modelo de datos que propone RDF consiste en tres tipos de objetos:

- ✓ Sujeto: recurso.
- ✓ Predicado: propiedad.
- ✓ Objeto: valor de la propiedad. Puede ser otro recurso o puede ser un literal, es decir, un recurso (especificado por un URI) o una cadena simple o primitiva otro tipo de datos definidos por XML. En la Figura 1 se representa la estructura del RDF.



Figura 1: Representación descriptiva de un recurso RDF

JSON

JavaScript Object Notation (JSON) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, *Standard ECMA-262 3rd Edition* - Diciembre 1999. Es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidas por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos (JSON, 2011).

El mismo está estructurado por:

- ✓ Una colección de pares de nombre/valor. En varios lenguajes es conocido como un objeto, registro, estructura, diccionario, tabla *hash*, lista de claves o un arreglo asociativo.
- ✓ Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales, virtualmente todos los lenguajes de programación las soportan de una forma u otra.

El estudio de los estándares propone superar las limitaciones de la web actual mediante la introducción de descripciones explícitas del significado, la estructura interna y la estructura global de los contenidos y servicios disponibles. Aportando metadatos en las páginas y sitios web, manteniendo el formato compatible de la estructura general de la web y con diversas categorías de páginas, optimizando así la interoperabilidad entre los distintos sistemas informáticos.

1.3 Metodología, tecnologías y herramientas utilizadas en la implementación del sistema

A continuación se muestra el estudio de la metodología, las tecnologías y herramientas que guiarán el proceso de desarrollo del sistema. La Base Tecnológica del Centro de Ideo-Informática define una serie de tecnologías para acotar el desarrollo de los sistemas en una misma dirección, razón por la cual algunas de las tecnologías utilizadas en esta investigación fueron definidas en el marco de dicha base tecnológica.

1.3.1 Metodología de desarrollo de software

OpenUP

Es una metodología que contiene el conjunto mínimo de prácticas que ayudan a los equipos a ser más eficaces en el desarrollo de software. OpenUP abraza una filosofía pragmática y ágil que se centra en la naturaleza colaborativa de desarrollo de software. Es un proceso iterativo que es Mínimo, Completo y Extensible que puede utilizarse tal cual o ampliarse para tratar una amplia variedad de tipos de proyecto. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Está organizada dentro de cuatro áreas principales de contenido: Comunicación y Colaboración, Intención, Solución y Administración (Hernández, 2012).

OpenUP se caracteriza por cuatro principios básicos que se soportan mutuamente:

- ✓ Colaboración para alinear los intereses y un entendimiento compartido.
- ✓ Balance para confrontar las prioridades (necesidades y costos técnicos) para maximizar el valor para los *stakeholders*.

Ventajas de usar OpenUP (Hernández, 2012):

- ✓ Metodología de desarrollo de software de código abierto diseñado para pequeños equipos organizados quienes quieren tomar una aproximación ágil del desarrollo.
- ✓ Proceso iterativo e incremental que es Mínimo, Completo y Extensible donde practicantes de desarrollo de software (desarrolladores, administradores de proyectos, analistas y probadores) trabajan juntos como un equipo de proyecto.
- ✓ Permite detectar errores tempranos a través de un ciclo iterativo.
- ✓ Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP.
- ✓ Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

1.3.2 Framework de Desarrollo

Symfony

Framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón modelo

vista controlador. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Está desarrollado completamente en PHP 5. Para su versión 2.1 se debe utilizar como mínimo la versión de PHP 5.3.3. Es compatible con la mayoría de gestores de bases de datos como: MySQL, PostgreSQL, Oracle y Microsoft SQL Server.

Características generales (Maestros del web, 2012):

- ✓ Fácil de instalar y configurar en la mayoría de plataformas.
- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✓ Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- ✓ Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor y permite un mantenimiento muy sencillo.
- ✓ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Bootstrap 2.0

Es un *framework* que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. Ha sido desarrollado por *Twitter* que recientemente liberó su versión 2.0. Ofrece una serie de archivos con código CSS que se incluyen en las páginas web para aplicar estilos de una manera rápida. La mayor ventaja es que puede crear interfaces que se adapten a los distintos navegadores (*responsive design*) apoyándose en un *framework* potente con numerosos componentes web disminuyendo esfuerzo y tiempo (GenBeta, 2012).

Características principales:

- ✓ Ofrece una serie de plantillas CSS y ficheros JavaScript que permiten integrar el *framework* de forma sencilla y potente en los proyectos web.
- ✓ Permite crear interfaces que se adapten a los diferentes navegadores, tanto de computadoras de escritorio como *tablets* y móviles a distintas escalas y resoluciones.

- ✓ Se integra perfectamente con las principales librerías JavaScript, por ejemplo *jQuery*.
- ✓ Ofrece un diseño sólido usando *LESS* y estándares como CSS3/HTML5.
- ✓ Es un *framework* ligero que se integra de forma limpia al proyecto actual.
- ✓ Funciona con todos los navegadores, incluido Internet Explorer usando HTML *Shim* para que reconozca las etiquetas HTML5.
- ✓ Dispone de distintos *layouts* predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.

1.3.3 Lenguajes de Programación

PHP

Es un lenguaje de programación utilizado para la creación de sitios Web. PHP es un acrónimo recursivo que significa (*PHP Hypertext Pre-processor*). Surgió en 1995, desarrollado por PHP Group. Es un lenguaje interpretado en el lado servidor utilizado para la generación de páginas Web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. No necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. PHP está diseñado específicamente para ser un lenguaje más seguro, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución y siguiendo algunas prácticas correctas de programación. El principal objetivo es mejorar los mecanismos de Programación Orientada a Objetos (POO) para solucionar las carencias de las versiones anteriores. Este avance constituye un paso necesario para conseguir que PHP sea un lenguaje apto para todo tipo de aplicaciones y entornos, incluso los más exigentes (Ciberaula, 2010).

Presenta determinadas características que lo distingue del resto de los lenguajes de programación entre las que se encuentran:

- ✓ Es multiplataforma, puede ser utilizado sobre los sistemas operativos: GNU/Linux, Windows, entre otros.
- ✓ Funcionalidades de conexión con la mayoría de los sistemas de gestión de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- ✓ Soporta la orientación a objeto y sus paradigmas.
- ✓ Su potencial se puede expandir utilizando extensiones.
- ✓ No requiere definición de tipos de variables ni manejo detallado del bajo nivel.
- ✓ Fácil de aprender.
- ✓ Posee una extensa documentación con ejemplos de uso de sus funciones.
- ✓ Es distribuido bajo licencia libre.

- ✓ La versión que soporta symfony 2.1 es la 5.3.

HTML

Define la estructura y el contenido de las páginas permitiendo combinar textos, imágenes, sonidos, vídeos y enlaces a otras páginas. Su nivel de complejidad es bajo y además permite embeber dentro de su código, scripts escritos en otros lenguajes como PHP y JavaScript. Pretende proporcionar una plataforma con la que desarrollar aplicaciones web más parecidas a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores. Entre sus características resalta la reducción de la necesidad de complementos externos y un mejor manejo de errores (Alvarez, 2009). La versión más reciente que soporta symfony 2.1 es la cinco.

CSS

Cascading Style Sheets (CSS) u Hojas de Estilo en Cascada es un lenguaje para definir el estilo o la apariencia de las páginas web escritas con HTML o de los documentos XML. Permite separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas web. Su novedad más importante consiste en la incorporación de mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas, sin tener que recurrir a trucos, que a menudo complican el código. CSS es el mecanismo que permite separar los contenidos definidos mediante XHTML y el aspecto que deben presentar esos contenidos (Sierra, 2013). La versión más reciente que soporta symfony 2.1 es la tres.

JavaScript

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Además, es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos, en otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. La integración de JavaScript y XHTML es muy flexible, ya que existen al menos tres formas para incluir código JavaScript en las páginas web (Javier, 2009).

Características principales (Alvarez, 2010):

- ✓ Es simple, no hace falta tener conocimientos de programación para poder hacer un programa.
- ✓ Maneja objetos dentro de una página web y sobre ese objeto se pueden definir diferentes eventos.
- ✓ Estos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades, modificar archivos, entre otros.

- ✓ Es dinámico, ya que responde a eventos en tiempo real, como presionar un botón, pasar el puntero del *mouse* sobre un determinado texto o el simple hecho de cargar la página en un tiempo. Con esto se puede cambiar totalmente el aspecto de una página al gusto del usuario, lo que evita tener en el servidor una página para cada gusto, hacer cálculos en base a variables cuyo valor es determinado por el usuario.
- ✓ La versión a utilizar es la 1.8.

1.3.4 Sistema Gestor de Bases de Datos

PostgreSQL

Sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia *BSD*¹ y con código fuente disponible libremente. Utiliza un modelo cliente-servidor y multiprocesos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. En la versión 9.1 ofrece varias características importantes, entre ellas la eliminación de obstáculos para el despliegue de nuevas aplicaciones y portabilidad en PostgreSQL. Además incluye replicación sincrónica que permite alta disponibilidad con consistencia a través de varios servidores: admite ordenación lingüística por base de datos, tabla o columna y mejora en gran medida el rendimiento de datos efímeros. Tiene la capacidad de conectarse y consultar otras bases de datos de PostgreSQL así como de crear fácilmente, cargar y administrar las características de nuevas bases de datos (Martinez, 2010).

1.3.5 Tecnologías del servidor

Flumotion

Es un sistema de distribución de contenido multimedia *Streaming Media Server* con un diseño modular y distribuido ofreciendo grandes posibilidades de estabilidad y escalabilidad. Permite realizar *streaming* de contenidos usando formatos y estándares libres como Ogg/Theora/Webm. Permite servir ficheros bajo demanda, pasando a través de un proxy reverso montado con apache, el cual tendrá autenticación. Su licencia es libre, con la mayoría de componentes bajo GPL 2+, y el resto bajo Expat y PSF-2. Está compuesto por tres tipos de componentes principales: el "*manager*", el "*worker*" y la interfaz de administración. Un "*manager*" se encarga de repartir tareas a los "*workers*" que estén conectados a él. Las tareas pueden ser codificar un *stream* de vídeo o audio, capturar el *stream* de la entrada *firewire* o *webcam*, multiplexar audio y vídeo, hacer el *streaming* por *HTTP*, etc. La interfaz de administración se conecta al "*manager*" permitiendo al usuario visualizar las tareas que se están realizando en los "*workers*" y poder asignar nuevas. Los componentes pueden residir en una misma máquina o distribuirse en varias

¹ Para sistemas *BSD* (*Berkeley Software Distribution*). Es libre y tiene menos restricciones en comparación con otras, como la GPL, estando muy cercana al dominio público. Permite el uso de *Open Source*.

(Flumotion, 2012). La versión utilizada es la 0.10.

1.3.6 Herramientas

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Garantiza una rápida construcción de aplicaciones con una mayor calidad, y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML. Es una herramienta multiplataforma distribuida bajo licencia privativa en la Visual Paradigm Suite. Es orientada a objetos y se puede integrar con Eclipse, Netbeans, IntelliJ IDEA entre otras herramientas de desarrollo. Proporciona un entorno ágil y eficiente para el diseño. Permite el diseño de base de datos y el diseño del sistema con el diagrama de clases UML. También apoya la generación de bases de datos, Java e Hibernate, así como técnicas de ingeniería inversa de bases de datos. Modelado colaborativo con CVS y Subversion permite el trabajo en equipo, puede soportar la edición simultánea de un mismo proyecto o diagrama. En Visual Paradigm Suite 5.0 se distribuye la versión 8.0 la más reciente de Visual Paradigm para UML. El uso de esta herramienta trae consigo varios beneficios (León, 2010).

Características:

1. Navegación intuitiva entre el modelo visual y el código.
2. Poderosa herramienta de generación de PDF/HTML a partir de diagramas UML.
3. Sincronización entre el código fuente y el modelo en tiempo real o bajo demanda.
4. Entorno visual de modelado superior.
5. Soporte para toda la notación UML.
6. Sofisticados y automáticos diagramas de capas.
7. Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
8. Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
9. Capacidades de ingeniería directa (versión profesional) e inversa.
10. Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
11. Disponibilidad de integrarse en los principales IDE (*Integrated Development Environment*).
12. Disponibilidad en múltiples plataformas.

Apache JMeter

Es una aplicación escrita en Java y diseñada para realizar pruebas de carga a servidores y medir su desempeño funcional. En sus inicios fue pensado para probar aplicaciones web, pero se ha extendido su uso a otros tipos de prueba.

Puede ser utilizado para probar el rendimiento, tanto estático o dinámico de los recursos (ficheros, *servlets*, scripts *Perl*, objetos de Java, bases de datos, servidores de *FTP*). Puede ser utilizado para simular una pesada carga concurrente en un servidor o red y analizar su rendimiento general. Mediante esta herramienta, pueden ser probados servidores como: Web, SOAP, servidor de bases de datos vía LDAP, servidor de correo, entre otros.

Asegura una completa portabilidad. Se pueden visualizar varias estadísticas de carga y proporciona una interfaz para la entrada dinámica de datos de prueba que permite operaciones rápidas en tiempos precisos (Apache JMeter, 2013). Se utilizó la versión 2.3.

1.3.7 Servidor Web Apache

Es flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Entre sus características destacan (Apache, 2012):

Características principales:

- ✓ Multiplataforma.
- ✓ Es un servidor de web conforme al protocolo HTTP.
- ✓ Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- ✓ Basado en hebras en la versión 2.0.
- ✓ Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- ✓ Se desarrolla de forma abierta.
- ✓ Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

1.3.8 Entorno Integrado de Desarrollo

NetBeans

Es un Entorno Integrado de Desarrollo (IDE, por sus siglas en inglés) gratuito y de código abierto para el desarrollo de aplicaciones web, de escritorio y para móviles, disponible en los sistemas operativos Windows, Linux, Mac OS X y Solaris. La base en la que se sustenta su elección es que permite desarrollar aplicaciones utilizando el *framework* Symfony y ejecutar los comandos del mismo directamente desde la interfaz del IDE. Es de fácil instalación y multiplataforma ya que se ejecuta en Windows y GNU/Linux. Además posee varios complementos para integrarlo con Twig (utilizado por Symfony2 para el desarrollo de plantillas Web), PHP 5, HTML 5 y CCS 3 y es sencillo de integrar con el marco de trabajo Symfony2

(NetBeans, 2012). Se utilizó la versión 7.3.

Conclusiones parciales

Para el desarrollo de la propuesta de solución serán utilizados los lenguajes de programación PHP 5.3, HTML 5, CSS 3, JavaScript 1.8. La implementación estará soportada por el *framework* de CSS Bootstrap 2.0 y por el *framework* de desarrollo Symfony 2.1 el cual proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web, además de automatizar las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de la aplicación. Se utilizará como SGBD PostgreSQL 9.1 y servidor web Apache 2. La metodología a utilizar es OpenUP la cual contiene el conjunto mínimo de prácticas que ayudan a los equipos a ser más eficaces en el desarrollo de software. Se empleará como herramientas de desarrollo: Visual Paradigm 8.0 como herramienta de modelado y NetBeans IDE 7.3 por su integración con los diferentes lenguajes de programación.

La investigación permitió establecer un marco común para la construcción de una aplicación capaz de centralizar los ficheros de media para los portales de los medios de prensa cubanos.

CAPÍTULO 2: CARACTERIZACIÓN DEL SISTEMA PROPUESTO

En el presente capítulo se definen las características que deberá cumplir el sistema propuesto a partir de un modelo de dominio; en el que se describen los principales conceptos asociados a la aplicación a desarrollar. Se definen y describen los requisitos funcionales y no funcionales donde se exponen las particularidades y cualidades que el producto debe cumplir, se construye el diagrama de casos de uso del sistema y se describen los casos de uso del sistema.

2.1 Modelo de Dominio

El modelo de dominio muestra los tipos de objetos más importantes en el sistema y puede ser tomado como el punto de partida para el diseño de la aplicación. Es obtenido de la especificación de requisitos mediante la entrevista con el cliente. Comprende y describe las principales clases dentro del contexto de sistema.

En la Figura 2 se muestra el diagrama de modelo del dominio.

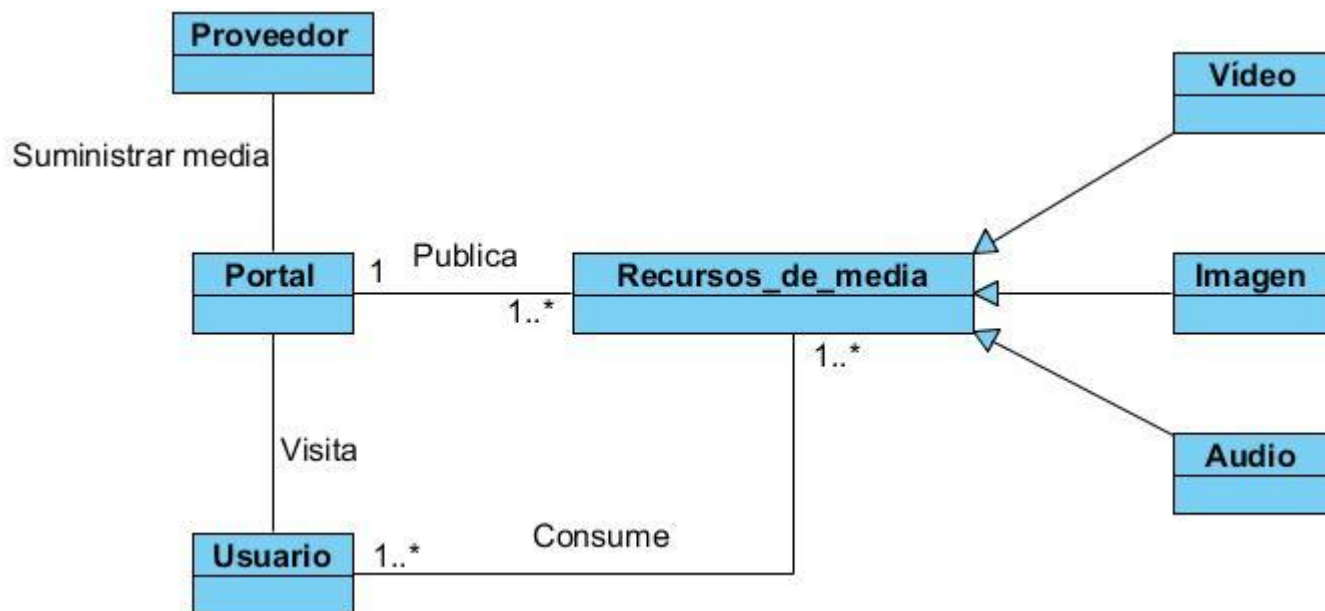


Figura 2: Diagrama de Clases del Modelo de Dominio

Descripción del Diagrama de modelo del dominio

- ✓ **Proveedor:** Es el usuario encargado de suministrar los recursos de media a los Portales de los Medios de Prensa.
- ✓ **Portal:** Encargado de publicar los recursos de media.
- ✓ **Usuario:** Es el usuario que consume los recursos de media.
- ✓ **Recursos de media:** Conjunto de ficheros publicados de media (imagen, audio y vídeo).
- ✓ **Imagen:** Imágenes publicadas para los usuarios.

- ✓ **Audio:** Audios publicados para los usuarios.
- ✓ **Vídeo:** Vídeos publicados para los usuarios.

2.2 Descripción del Sistema

La aplicación web centralizará los contenidos de media y deberá proveer ficheros de media (imagen, audio y vídeo) a los portales de los medios de prensa cubanos. Se encarga de almacenar todos los archivos que le son suministrados, en una base de datos, a la cual se harán las consultas para darle respuesta a las peticiones de los usuarios. Además integra un grupo de funcionalidades las cuales permiten a los usuarios la realización de búsquedas (básicas o avanzadas) de los archivos de media, a partir de etiquetas de metadatos. Muestra una interfaz que es la encargada de mostrar los resultados de acuerdo al criterio introducido por el usuario permitiendo la gestión de los contenidos, logrando así mejorar la interacción entre el sistema y el usuario. Tiene como objetivo principal permitir el acceso rápido y confiable a la información y lograr una distribución de la carga y el procesamiento en los servidores dedicados a la difusión de los ficheros de media.

2.3 Requerimientos del software

El propósito fundamental de la captura de los requisitos es guiar el desarrollo hacia el sistema correcto. Durante la fase de levantamientos de requisitos se aplicó una entrevista (Ver Anexo 1) al cliente como una habilidad para la obtención de las funcionalidades del sistema, una vez obtenidos las mismas se incluyeron en el documento de Especificación de Requisitos de Software. En esta sección se especifican los requisitos funcionales y no funcionales definidos para satisfacer las necesidades del cliente.

2.3.1 Requisitos Funcionales

Tabla 1: Requisitos funcionales del sistema

RF-1 Autenticar usuario.	RF-23 Eliminar vídeo.
RF-2 Solicitar cuenta de usuario.	RF-24 Bloquear vídeo.
RF-3 Adicionar cuenta de usuario.	RF-25 Desbloquear vídeo.
RF-4 Editar cuenta de usuario.	RF-26 Descargar vídeo.
RF-5 Eliminar cuenta de usuario.	RF-27 Reproducir vídeo.
RF-6 Habilitar cuenta de usuario.	RF-28 Subir vídeo.
RF-7 Deshabilitar cuenta de usuario.	RF-29 Editar metadato.
RF-8 Auditar cuenta de usuario.	RF-30 Visualizar metadato.

RF-9 Visualizar cuenta de usuario.	RF-31. Exportar metadato (PDF, JSON).
RF-10 Filtrar cuenta de usuario.	RF-32 Editar permiso.
RF-11 Eliminar imagen.	RF-33 Asignar permisos a un usuario.
RF-12 Bloquear imagen.	RF-34 Visualizar permiso de un usuario.
RF-13 Desbloquear imagen.	RF-35 Realizar búsqueda simple del contenido.
RF-14 Descargar imagen.	RF-36 Realizar búsqueda avanzada del contenido.
RF-15 Visualizar imagen.	RF-37 Visualizar resultado de la búsqueda.
RF-16 Subir imagen.	RF-38 Servir contenidos mediante <i>streaming</i> .
RF-17 Eliminar audio.	RF-39 Mostrar código para referenciar los contenidos de media publicados.
RF-18 Bloquear audio.	RF-40 Generar reporte sobre acciones a los contenidos.
RF-19 Desbloquear audio.	RF-41 Formulario de Contacto.
RF-20 Descargar audio.	RF-42 Ayuda en línea.
RF-21 Reproducir audio.	RF-43 Servicio de RSS de los últimos contenidos añadidos al sistema.
RF-22 Subir audio.	RF-44 Compartir en Redes Sociales (<i>Twitter</i> y <i>Facebook</i>).

Especificación de los Requisitos de Software

Tabla 2: Descripción del RF 16: Subir Imagen

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 16	Subir imagen.	La acción inicia cuando un usuario Proveedor desea subir una imagen al sistema, el mismo muestra la interfaz correspondiente con la opción Subir imagen.	Alta	Alta
Prototipo				

Campos	Tipos de Datos	Reglas o Restricciones
1. Título.	1. Texto.	Introducir el título del archivo.
2. Dirección física del archivo.	2. File.	Que indique la dirección del archivo.
3. Tipo de archivo.	3. Imagen.	Solo se pueden subir archivos con formato: jpg, jpeg y png.
Observaciones	En caso de dejar campos vacíos el sistema muestra un mensaje: "Por favor, rellene este campo".	

Tabla 3: Descripción del RF36: Realizar búsqueda simple del contenido

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 36	Realizar búsqueda simple del contenido.	El usuario una vez dentro del sistema introduce en el campo de texto (Búsqueda) el criterio de búsqueda.	Alta	Alta

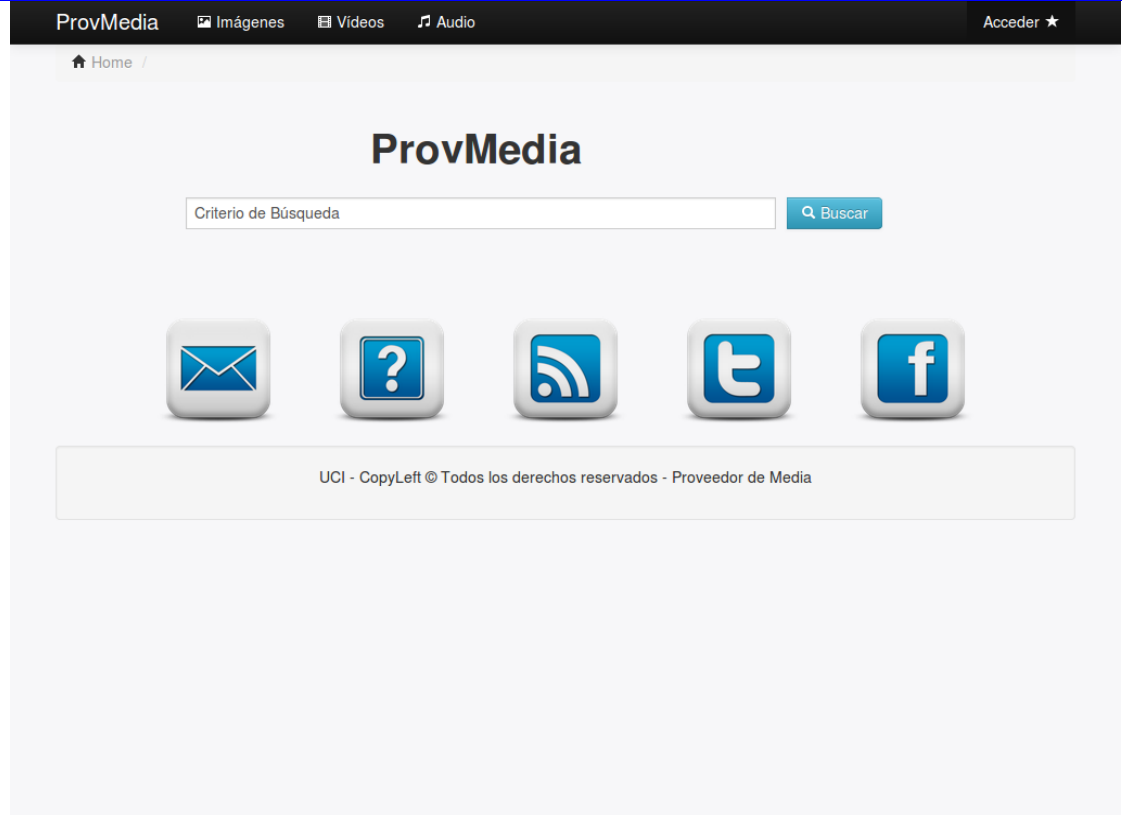
Prototipo		
 <p>The screenshot shows the ProvMedia website interface. At the top, there is a navigation bar with 'ProvMedia', 'Imágenes', 'Vídeos', 'Audio', and 'Acceder'. Below this is a search bar with the placeholder text 'Criterio de Búsqueda' and a 'Buscar' button. Underneath the search bar are five social media icons: an envelope, a question mark, RSS, Twitter, and Facebook. At the bottom of the page, there is a footer with the text 'UCI - CopyLeft © Todos los derechos reservados - Proveedor de Media'.</p>		
Campos	Tipos de Datos	Reglas o Restricciones
1. Campo de texto.	Texto.	Que el usuario introduzca un criterio de búsqueda.
Observaciones	En caso de introducir incorrectamente el criterio de búsqueda se muestra un mensaje: "No hay resultados para este criterio".	

Tabla 4: Descripción del RF 37: Realizar búsqueda avanzada del contenido

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 37	Realizar búsqueda avanzada del contenido.	El usuario autenticado selecciona el vínculo Búsqueda avanzada, luego se muestra una interfaz que permite realizar una búsqueda avanzada (audio, video, imagen).	Alta	Alta
Prototipo				

Campos	Tipos de Datos	Reglas o Restricciones
1. Nombre.	1. Texto.	Introducir el nombre del archivo
2. Título.	2. Texto.	Introducir el título del archivo
3. Autor.	3. Texto.	Introducir el autor
4. Fecha de creación.	4. Fecha.	Introducir la fecha de creación
5. Fecha de modificación.	5. Fecha.	Introducir fecha de modificación
6. Temática.	6. Texto	Introducir temática
7. Colaborador.	7. Texto	Introducir el colaborador
8. Palabras clave.	8. Texto	Introducir las palabras clave
9. Editor.	9. Texto	Introducir el editor
10. Fuente.	10. Texto	Introducir la fuente
Observaciones	En caso de introducir incorrectamente el criterio de búsqueda se	

	muestra un mensaje: “Debe introducir el criterio de búsqueda correctamente”.
--	--

Las descripciones correspondientes al resto de los requisitos se encuentran en el Anexo 2.

2.3.2 Requisitos no Funcionales

Usabilidad

- ✓ El sistema podrá ser usado por una persona que posea conocimientos básicos en informática.
- ✓ El sistema debe contar con una interfaz sencilla para el cliente.
- ✓ Utilizar el idioma español para los textos de la interfaz.

Fiabilidad

- ✓ El sistema debe estar disponible las 24 horas.
- ✓ Las recursos de media van a ser visualizados desde un servidor *Streaming*, en caso de fallo del servidor van a ser visualizados desde el mismo sistema.

Seguridad

- ✓ El sistema debe controlar los permisos de acceso según el rol que desempeñe el usuario.
- ✓ Solo si el usuario autenticado se encuentra registrado se autoriza el acceso.

Software

- ✓ El sistema debe desarrollarse con la tecnología Symfony 2.1.8, el cual utiliza la arquitectura Modelo-Vista-Controlador.
- ✓ Para los servidores usar como sistema operativo Ubuntu 12.04 LTS.
- ✓ El sistema debe ser compatible con las versiones actuales de los navegadores (Mozilla Firefox 19.0 y Chromium 20).

Hardware

- ✓ Servidores: Core i3 3.1, 2 GB RAM, HDD (200 a 500 GB).
- ✓ Cliente: Pentium 4 micro 2.20 GHz, 512 MB RAM o superior.

Requerimientos de apariencia o interfaz externa

- ✓ El sistema debe poseer una interfaz amigable para el usuario, basada en Web, brindando facilidades que permitan interactuar con el sistema de forma fácil y rápida.

Restricciones del diseño y la implementación

- ✓ Se deben procesar imagen, audio y vídeo con los siguientes formatos: jpg, png, jpeg, webm, ogg, ogx y ogv.
- ✓ Las interfaces para al usuario deben ser compatibles con las versiones actuales de los navegadores (Mozilla Firefox 19.0 y Chromium 20).

- ✓ Las PC clientes deben tener un navegador web Mozilla Firefox preferentemente.
- ✓ Desarrollar la aplicación con herramientas *Open Source*.

Soporte

- ✓ El soporte y/o mantenimiento de la aplicación no debe detener el servicio.

Rendimiento

- ✓ El sistema debe mostrar una respuesta eficiente en el intercambio de información cliente-servidor.

Legales

- ✓ Las herramientas seleccionadas para el desarrollo del sistema deben estar respaldadas por licencias.

2.4 Definición de los Casos de Uso del Sistema

2.4.1 Identificación de los Actores del sistema

Tabla 5: Actores del sistema

Actor	Objetivo
Usuario Anónimo.	Es un usuario anónimo que puede realizar operaciones en el sistema como visualizar contenidos de media, realizar búsqueda básica.
Usuario Autenticado.	Es un usuario autenticado, el cual puede realizar operaciones en el sistema de acuerdo al permiso que posea.
Proveedor.	Es el usuario que provee los archivos de media al sistema.
Administrador.	Es un usuario autenticado en el sistema con permiso para ejecutar las funcionalidades de administración.

2.4.2 Diagrama de caso de uso del Sistema

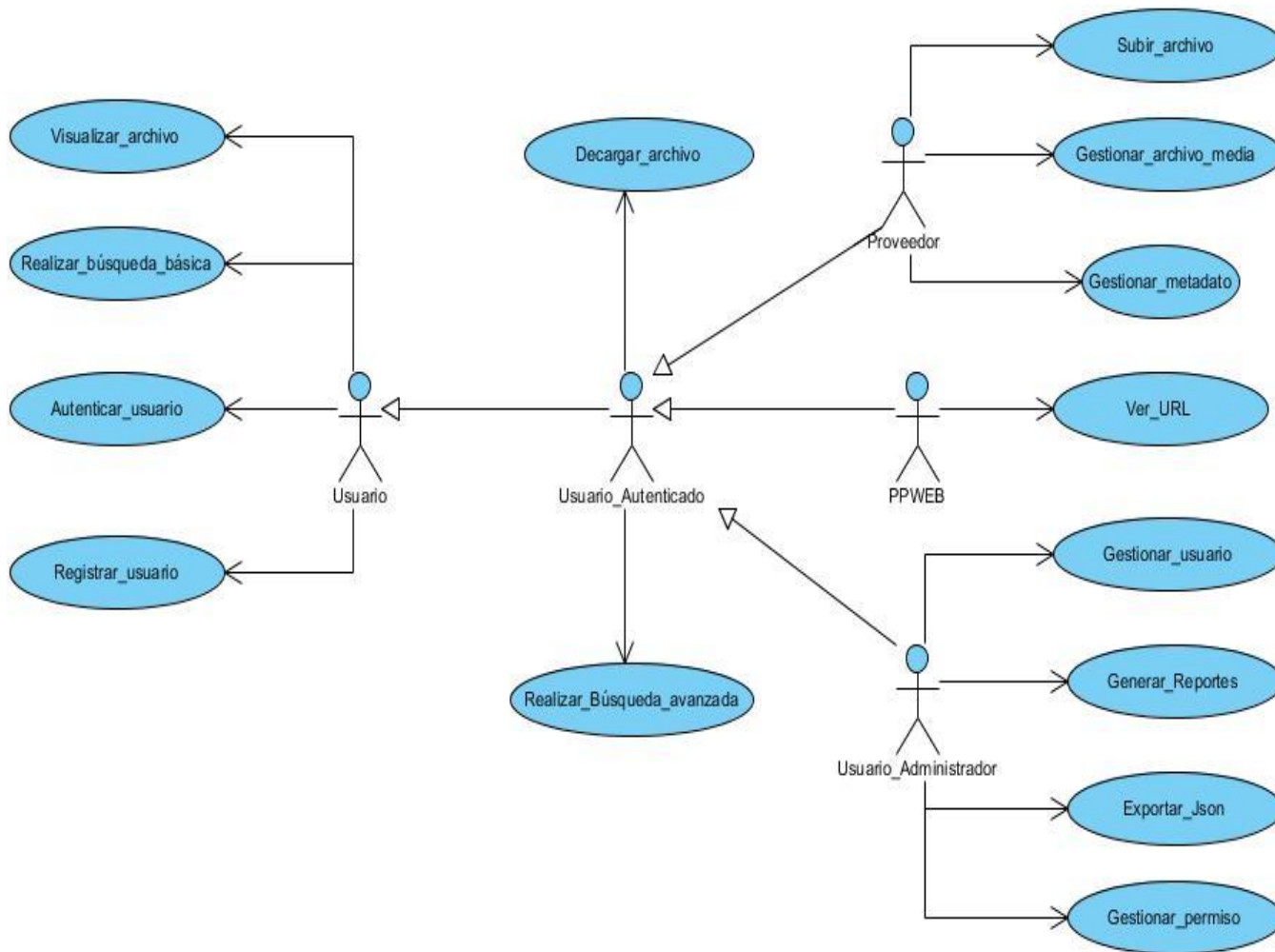


Figura 3: Diagrama de CU del sistema del Usuario Administración

2.4.3 Descripción de casos de uso del sistema

En las Tablas 6, 7, 8 y 9 se muestran las descripciones de los Casos de Uso del Sistema que fueron modelados:

Tabla 6: Descripción de CU 2: Gestionar Búsqueda

Objetivo	Realizar Búsqueda
Actores	Usuario Autenticado.
Resumen	El caso de uso se inicia cuando se selecciona la opción de búsqueda de materiales. El sistema permite realizar dos tipos de búsqueda: básica y avanzada, el resultado de la búsqueda se muestra teniendo en cuenta los filtros introducidos. El caso de uso culmina cuando se muestran los resultados obtenidos a partir de los filtros introducidos por el usuario.

Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	El usuario debe estar autenticado y poseer permisos para la realización de búsqueda de materiales.	
Poscondiciones	Se obtiene el listado de los materiales de acuerdo a los filtros establecidos.	
Flujo de eventos		
Flujo básico realizar búsqueda		
	Actor	Sistema
1	1. El caso de uso se inicia cuando el usuario selecciona la opción de búsqueda de material.	1.1 El sistema muestra la interfaz de búsqueda de materiales.
2	2. El usuario activa el área correspondiente para introducir opcionalmente los filtros de búsqueda de acuerdo al tipo de búsqueda que desea realizar, las que pueden ser: a) Búsqueda Avanzada, ir a la sección "Búsqueda avanzada" b) Búsqueda Básica, ir a la sección "Búsqueda básica"	
Sección "Búsqueda avanzada"		
	Acción del Actor	Respuesta del Sistema
		1.1 El sistema activa el área de Búsqueda avanzada, permitiendo introducir datos para una búsqueda exacta de los archivos.
	2. El usuario introduce los datos a los filtros correspondientes a la búsqueda avanzada: <ul style="list-style-type: none"> ✓ Filtro asociado al formato de los ficheros: <ul style="list-style-type: none"> • Tipo de formato (imagen, audio y vídeo). ✓ Filtros avanzados de datos generales: 	

	<ul style="list-style-type: none"> • Texto (opcional)(J) • Respetar ortografía (opcional) (L) 	
Flujos alternos		
	Acción del Actor	Respuesta del Sistema
		3.1 Si se seleccionó el filtro por formato y no se seleccionó el tipo de formato, el sistema muestra el siguiente mensaje: "Este valor es necesario". El sistema da la posibilidad de seleccionarlo. Si el tipo de formato que selecciona es incorrecto el sistema muestra el mensaje: "El valor especificado no es válido". El sistema da la posibilidad de corregirlo.
Sección: "Búsqueda Básica"		
	Acción del Actor	Respuesta del Sistema
1		1.1 El sistema activa el área de Búsqueda básica, generando dinámicamente una tabla con los filtros de búsqueda básica obtenidos de la base de datos.
2	<p>2. El usuario introduce los filtros correspondientes a la búsqueda básica:</p> <ul style="list-style-type: none"> ✓ Título del material (opcional) (A) ✓ Palabras claves (opcional) (B) ✓ Procedencia (opcional) (C) ✓ Descripción (opcional) (D) 	2.1 El sistema verifica que los datos introducidos estén correctos.
Flujos alternos		
	Acción del Actor	Respuesta del Sistema
		2.1 Si el actor seleccionó uno de los campos de fecha y seleccionó incorrectamente el formato, el sistema muestra el mensaje: "El valor especificado no es válido.". El sistema da la posibilidad de corregirlo.

Tabla 7: Descripción de CU 7: Gestionar Permisos

Objetivo	Realizar la Gestión de Permisos.	
Actores	Usuario Administración.	
Resumen	El caso de uso se inicia cuando el usuario realiza las acciones de insertar, modificar o eliminar los permisos de los roles existentes en el sistema, culmina el mismo al realizarse una de ellas.	
Complejidad	Alta	
Prioridad	Secundario	
Precondiciones	El usuario debe estar autenticado y poseer permisos para la gestión de usuarios.	
Poscondiciones	Se insertan, modifican o eliminan los datos de los permisos de acuerdo a las acciones del actor.	
Flujo de eventos		
Acción del Actor		
	Actor	Sistema
1	<p>1. El caso de uso se inicia cuando el usuario selecciona una de las opciones en el menú general para la gestión de permisos, puede seleccionar:</p> <ul style="list-style-type: none"> a) Adicionar Permiso, ir a la sección “Adicionar Permiso” b) Modificar Permiso, ir a la sección “Modificar Permiso” c) Eliminar Permiso, ir a la sección “Eliminar Permiso” 	
Sección: “Adicionar Permiso”		
	Acción del Actor	Respuesta del Sistema
2		<p>1.2 El sistema verifica que exista conexión con la Base de Datos.</p> <p>1.3 El sistema muestra la interfaz de gestión de permisos.</p>
3	<p>2. El usuario selecciona la opción “nuevo” de la interfaz de gestión de permisos e introduce los datos necesarios para la creación de un permiso:</p> <ul style="list-style-type: none"> ✓ Nombre del Permiso (obligatorio) (A) ✓ Contenido del Permiso (obligatorio) (B) ✓ Descripción del Permiso (opcional) (C) 	<p>2.1 El sistema habilita el botón “Guardar” (E) si se han introducido todos los datos obligatorios.</p>

	✓ Tipo de Permiso(obligatorio) (D)	
4	3. El usuario presiona el botón “Guardar” (E)	3.1 El sistema muestra un mensaje de notificación de la acción: “El permiso ha sido adicionado correctamente” 3.2 El sistema actualiza el árbol de permisos existentes incorporando los datos del nuevo permiso y culmina el caso de uso.
Flujos alternos		
	Acción del Actor	Respuesta del Sistema
1		1.2 Si no se puede establecer conexión con la Base de Datos, el sistema muestra el siguiente mensaje: “No se puede establecer conexión con los datos.” Culmina el caso de uso.
Sección: “Modificar Permiso”		
	Acción del Actor	Respuesta del Sistema
1		1.1 El sistema verifica que exista conexión con la Base de Datos. 1.2 El sistema muestra la interfaz de gestión de permisos.
2	2. El usuario selecciona el permiso que desea modificar en el árbol de permisos del sistema y posteriormente selecciona la opción “Modificar” (F) en la interfaz de gestión de permisos.	2.1 El sistema carga los datos del permiso en la interfaz de gestión de permiso donde pueden ser editables.
3	3. El usuario edita los datos del permiso que desea modificar: <ul style="list-style-type: none"> ✓ Identificador del Permiso (Obligatorio) (A) ✓ Nombre del Permiso (Obligatorio) (B) ✓ Descripción del Permiso (Obligatorio) (C) ✓ Tipo de Permiso (Obligatorio) (D) 	3.1 El sistema habilita el botón “Guardar” (E) en caso de que se hayan realizado cambios en los datos del permiso y los datos requeridos del permiso estén completos.
4	4. El usuario presiona el botón “Guardar” (E)	4.1. El sistema muestra un mensaje: “Se ha modificado el permiso”. Culmina el caso de uso.
Flujos alternos		
	Acción del Actor	Respuesta del Sistema

1		1.1 Si no se puede establecer conexión con la Base de Datos, el sistema muestra el siguiente mensaje: “No se puede establecer conexión con los datos.” Culmina el caso de uso.
	Sección: “Eliminar Permiso”	
	Acción del Actor	Respuesta del Sistema
1		1.1 El sistema verifica que exista conexión con la Base de Datos. 1.2 El sistema muestra la interfaz de gestión de permisos.
2	2. El usuario selecciona el o los permisos que desea eliminar en el árbol de permisos del sistema y posteriormente selecciona la opción “Eliminar” (G) en la interfaz de gestión de permisos.	2.1 El sistema pide confirmación de la acción.
	3. El usuario selecciona la opción “Aceptar”	3.1 El sistema elimina los datos del permiso.
	Flujos alternos	
	Acción del Actor	Respuesta del Sistema
1		1.1 Si no se puede establecer conexión con la Base de Datos, el sistema muestra el siguiente mensaje: “No se puede establecer conexión con los datos.” Culmina el caso de uso.

Tabla 8: Descripción de CU 3: Visualizar material

Objetivo	Visualizar Material.
Actores	Usuario.
Resumen	El caso de uso se inicia cuando se selecciona el material que se desea visualizar. El caso de uso culmina cuando el usuario realiza alguna operación de finalizar.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	El usuario debe estar autenticado, debe poseer permisos para realizar la reproducción de materiales. Se debe haber realizado la copia en baja resolución de los materiales originales, el resumen visual de los mismos y estar

	publicados en el servidor de <i>Streaming</i> .	
Poscondiciones	Se realizan las acciones de reproducción seleccionadas por el usuario sobre el material que se reproduce.	
Flujo de eventos		
	Acción del Actor	Respuesta del Sistema
1	1. El caso de uso se inicia cuando el usuario selecciona el material que va a reproducir.	1.1 El sistema visualiza el material seleccionado. El material comienza a reproducirse por defecto.
2	2. El usuario puede realizar las siguientes acciones de reproducción: ✓ Pausar/Continuar reproducción, ir a la sección "Pausar/Continuar reproducción". ✓ Parar reproducción, ir a la sección "Parar reproducción".	
Sección: "Pausar/Continuar Reproducción"		
	Acción del Actor	Respuesta del Sistema
3	3. El usuario presiona el botón "Pausar/Continuar" reproducción (A/B). Si el material está reproduciéndose el botón estará activo con la función de pausa, si el material está pausado el botón estará activo con la función de continuar la reproducción.	3.1 Se pausa o continúa la reproducción del material. Culmina el caso de uso.
Sección: "Parar Reproducción"		
	Acción del Actor	Respuesta del Sistema
1	1. El usuario presiona el botón "Parar" (C).	1.1 El reproductor sitúa la reproducción en pausa en el inicio del material. Culmina el caso de uso.

Tabla 9: Descripción de CU 4: Descargar material

Objetivo	Descargar Material
Actores	Usuario Autenticado.
Resumen	El caso de uso se inicia cuando el usuario con privilegios decide descargar los materiales que han sido solicitados. Se verifica que la solicitud se haya aprobado, se selecciona la solicitud y se inserta la ruta para la descarga. Culmina el caso de uso cuando ha sido descargado el material.
Complejidad	Media

Prioridad	Crítico	
Precondiciones	Que el usuario esté autenticado, tenga permiso para descargar material y se haya aprobado la solicitud.	
Poscondiciones	Es indicada la dirección de destino para la descarga de materiales y se ejecuta la descarga terminando así el caso de uso.	
Flujo de eventos		
	Acción del Actor	Respuesta del Sistema
1		1. El caso de uso se inicia cuando el sistema muestra la interfaz para descargar material.
2	2. El usuario introduce la dirección de destino para la descarga (Obligatorio). (A)	2.1. El sistema habilita el botón "Descargar" (B)
3	3. El usuario presiona el botón "Descargar" (B).	3.1 El sistema muestra la zona donde se realizará la descarga. Culmina el caso de uso.
Flujo Alternativo		
	Acción del Actor	Respuesta del Sistema
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Las descripciones correspondientes al resto de los Casos de Uso del Sistema se encuentran en el Anexo 3.

Conclusiones parciales

En el presente capítulo se determinaron las características principales del sistema para lograr un mayor entendimiento del mismo. Se llevó a cabo el levantamiento de los requerimientos del sistema resumidos en CU para el posterior desarrollo de las funcionalidades. Se determinó el comportamiento de las acciones de los usuarios a través de la confección del diagrama de CU del sistema. La información expuesta y descrita incluida en los artefactos obtenidos, permitió determinar la estructura y organización del sistema a desarrollar, y de esta forma servir de apoyo para el análisis y el diseño a desarrollar posteriormente.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

En el presente capítulo se analiza la estructura de la aplicación, teniendo como punto de partida la Fase de Análisis y Diseño donde se generan los diagramas de clases del análisis, de interacción y los diagramas de clases del diseño respectivamente. Se estudian y se seleccionan los patrones arquitectónicos y de diseño a emplear en el sistema a desarrollar, se genera además el modelo de despliegue.

3.1 Patrones Arquitectónicos

El patrón arquitectónico elegido para el tratamiento de las interfaces de usuarios de la aplicación es el Modelo Vista Controlador (MVC) utilizado por el *framework* de desarrollo Symfony. Trata de realizar un diseño que desacople los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos, con la finalidad de mejorar la reusabilidad. De esta forma, las modificaciones en las vistas impactan en menor medida la lógica de negocio. El MVC separa el modelado del dominio, la presentación y las acciones basadas en datos integrados por el usuario en tres clases diferentes (Potencier, 2012):

- ✓ **Modelo:** Esta es la representación específica de la información sobre la cual funciona el sistema. Responde a las peticiones que realizan las vistas sobre su estado y permite su actualización normalmente desde el controlador. Representa cada una de las clases que son generadas según la estructura de la base de datos.
- ✓ **Vista:** Representa el modelo en un formato adecuado para interactuar, usualmente interfaz de usuario. Interpreta las acciones del usuario, accediendo a las operaciones de negocio de la aplicación y modificando a partir de los resultados, el estado del modelo y la navegación entre vistas.
- ✓ **Controlador:** Este responde a eventos, usualmente acciones del usuario, que son la lógica de la aplicación, e invoca peticiones al modelo y a la vista.

En la Figura 4 se muestra el diagrama del patrón MVC:



Figura 4: Diagrama del Patrón Arquitectónico MVC

3.2 Patrones de Diseño

Los patrones de diseño constituyen la base para la búsqueda de soluciones a problemas comunes en el desarrollo del *software* y otros ámbitos referentes al diseño de interacción o interfaces. Para el desarrollo de la solución se evidencia el uso de los Patrones Generales de *Software* para la Asignación de Responsabilidades (GRASP, por sus siglas en inglés) utilizados como buenas prácticas de la Programación Orientada a Objetos. De la misma forma se listan los patrones Banda de los Cuatro (*GOF*, por sus siglas en inglés) que son evidenciados en el proceso de implementación de la interfaz del sistema, así como en los marcos de trabajo empleados como apoyo a dicho proceso (Pressman, 2002).

3.2.1 Patrones GRASP

- ✓ **Experto:** El patrón experto se evidencia en el marco de trabajo, ya que cada función del mismo está bien definida y agrupada en clases por la funcionalidad común que representan. Como ejemplo se tienen las funciones para el almacenamiento y recuperación de la información desde los repositorios.
- ✓ **Bajo acoplamiento:** Se garantiza en la arquitectura del marco de trabajo la separación de las funcionalidades en clases con baja dependencia entre ellas y definición de interfaces, lo que representa una ventaja a la hora de realizar modificaciones en algún componente sin repercusiones en el resto de la plataforma.
- ✓ **Alta cohesión:** Una de las características principales de Symphony es la organización del trabajo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. Por ejemplo las clases con sufijo *Default_Controller* contienen varias funcionalidades estrechamente relacionadas entre ellas, teniendo un sentido común y un propósito único, siendo las encargadas de controlar las acciones de cada entidad a la que pertenecen y por lo tanto

pertenecen a la capa del Controlador dentro de la arquitectura Modelo-Vista-Controlador. Este hace posible que el software sea flexible a cambios sustanciales con efecto mínimo.

- ✓ **Controlador:** Todas las peticiones web son manejadas por un solo controlador frontal (app.php) el cual corresponde al archivo raíz de la web, que es el punto de entrada de la aplicación en un entorno determinado.
- ✓ **Creador:** Identifica quien debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. Puede soportar un bajo acoplamiento, mayor claridad, encapsulamiento y reutilización. La nueva instancia deberá ser creada por la clase que tiene la información necesaria para realizar la creación del objeto. Usa directamente las instancias creadas del objeto. Almacena o maneja varias instancias de la clase.

3.2.2 Patrones GOF

- ✓ **Decorator (decorador):** Es un patrón estructural, permite añadir funcionalidades dinámicamente a una clase con la creación de otra clase sin necesidad de la herencia. Este método pertenece a la clase abstracta *sfView*, padre de todas las vistas, que contienen un decorador para permitir agregar funcionalidades dinámicamente. El archivo nombrado *base.html.twig* es el que contiene el diseño de la página. Este es el archivo principal para una dependencia a tres niveles del código HTML quedando en un primer nivel *base.html.twig*, que define el encabezado y cuerpo de las páginas HTML. En un segundo nivel el *Fronted.html.twig* y el *Backend.html.twig*, para fijar el maquetado que es visible en todas las páginas del sistema y evitando repetirlo en cada página. Por último los archivos de plantillas con extensión *html.twig* donde se estructura el contenido de cada una de las páginas. El contenido de la plantilla se integra en el diseño decorando a la plantilla.

3.3 Diagramas de clases del análisis y del diseño

El flujo de trabajo Análisis y Diseño tiene como objetivo traducir los requisitos a una especificación que describe cómo implementar el sistema:

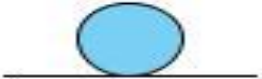
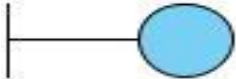

- ✓ El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales.
- ✓ El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos.

Durante el análisis, son descritos los requisitos, refinándolos y estructurándolos. Todas estas actividades se realizan para conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema incluyendo su arquitectura.

3.3.1 Modelo de Análisis

El modelo de análisis contiene clases del análisis y sus objetos organizados en paquetes que colaboran entre sí. Estas clases se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Estas clases contienen atributos y entre ellas se establecen relaciones. UML propone clasificar las clases en:

Tabla 10: Estereotipos de clases del análisis

Nombre	Características	Representación
Entidad	Modelan información que posee larga vida y que es a menudo persistente.	 CE_Ejemplo
Interfaz	Modelan la interacción entre el sistema y sus actores.	 CI_Ejemplo
Control	Coordinan la realización de uno o unos pocos CU coordinando las actividades de los objetos que implementan la funcionalidad del CU.	 CC_Ejemplo

3.3.2 Diagrama de Clase de Análisis

En la Figura 5 se muestra el diagrama de clases de análisis los diagramas correspondientes se encuentran en el Anexo 4:

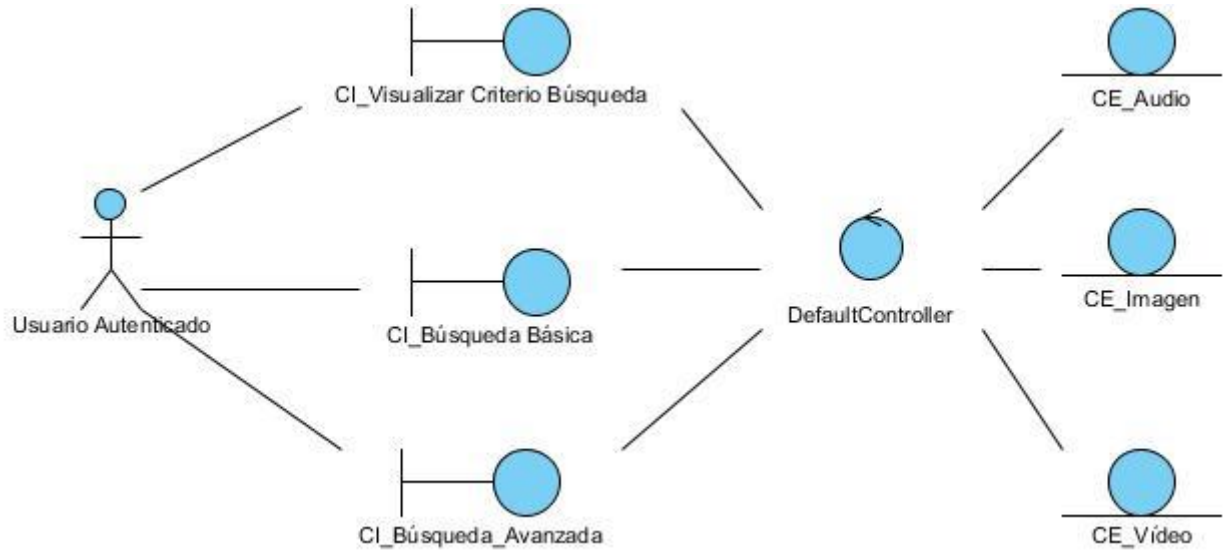




Figura 5: Diagrama de Clases del Análisis Gestionar Búsqueda

3.3.3 Modelo de diseño

Es un modelo de objetos que describe la realización de los CU, 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 de diseño sirve de abstracción de la implementación y el código fuente.

En la Tabla 11 se muestran los estereotipos utilizados en los diagramas de clases del diseño:

Tabla 11: Estereotipos Web del Diseño

Clases	Estereotipos	Función
<i>Server Page</i> [SP] (en español página del lado del servidor)		Representa la página web que tiene el código en el servidor.
<i>Client Page</i> [CP] (en español página del lado del cliente)		Representa la página web con formato HTML.

<p>Form (en español formulario HTML).</p>		<p>Colección de elementos de entrada que son parte de una página cliente.</p>
---	---	---

En las Figuras 6 y 7 se muestran los diagramas de Estereotipos Web del sistema, los diagramas correspondientes se encuentran en el Anexo 5:

3.3.3.1 Diagrama de Clases de Estereotipos Web

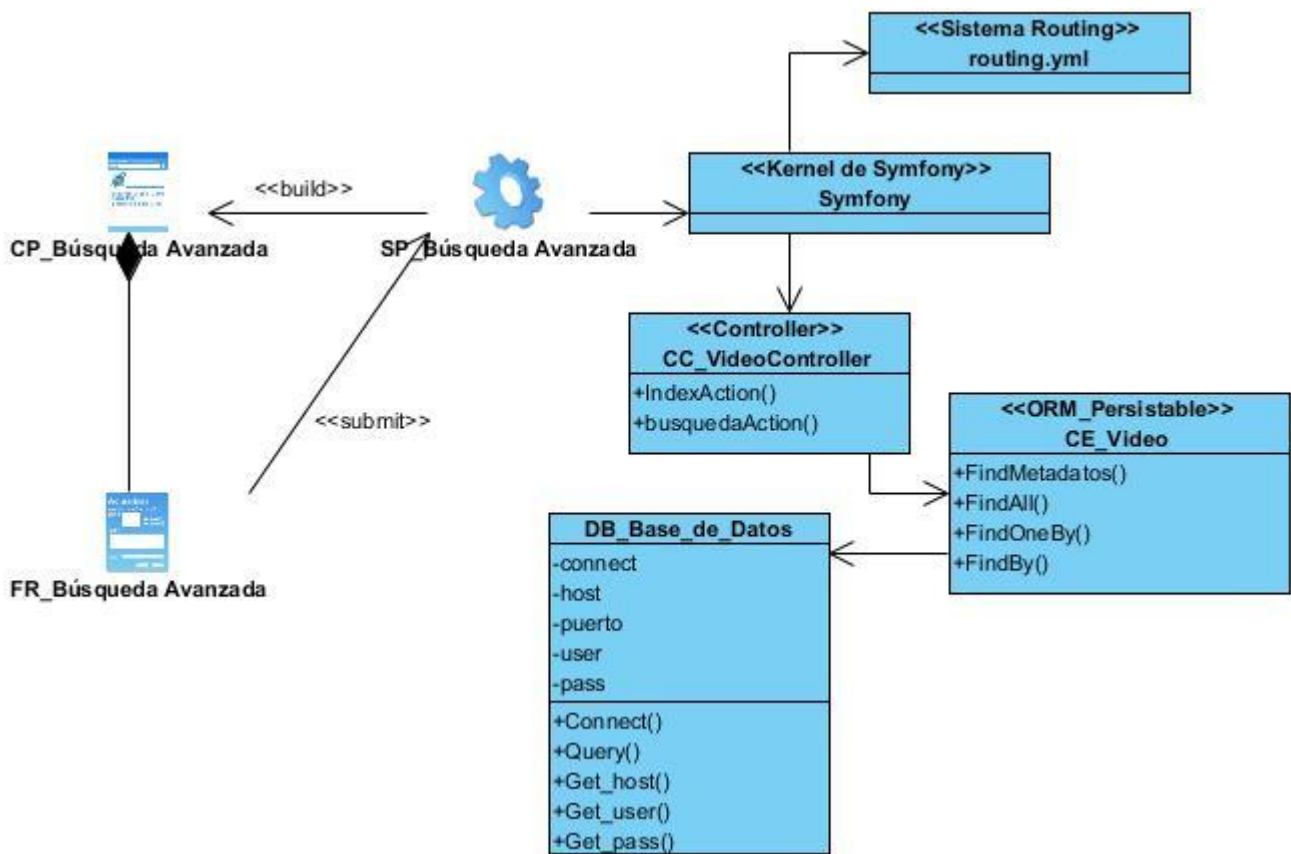


Figura 6: Diagrama de clases de estereotipos web Búsqueda avanzada

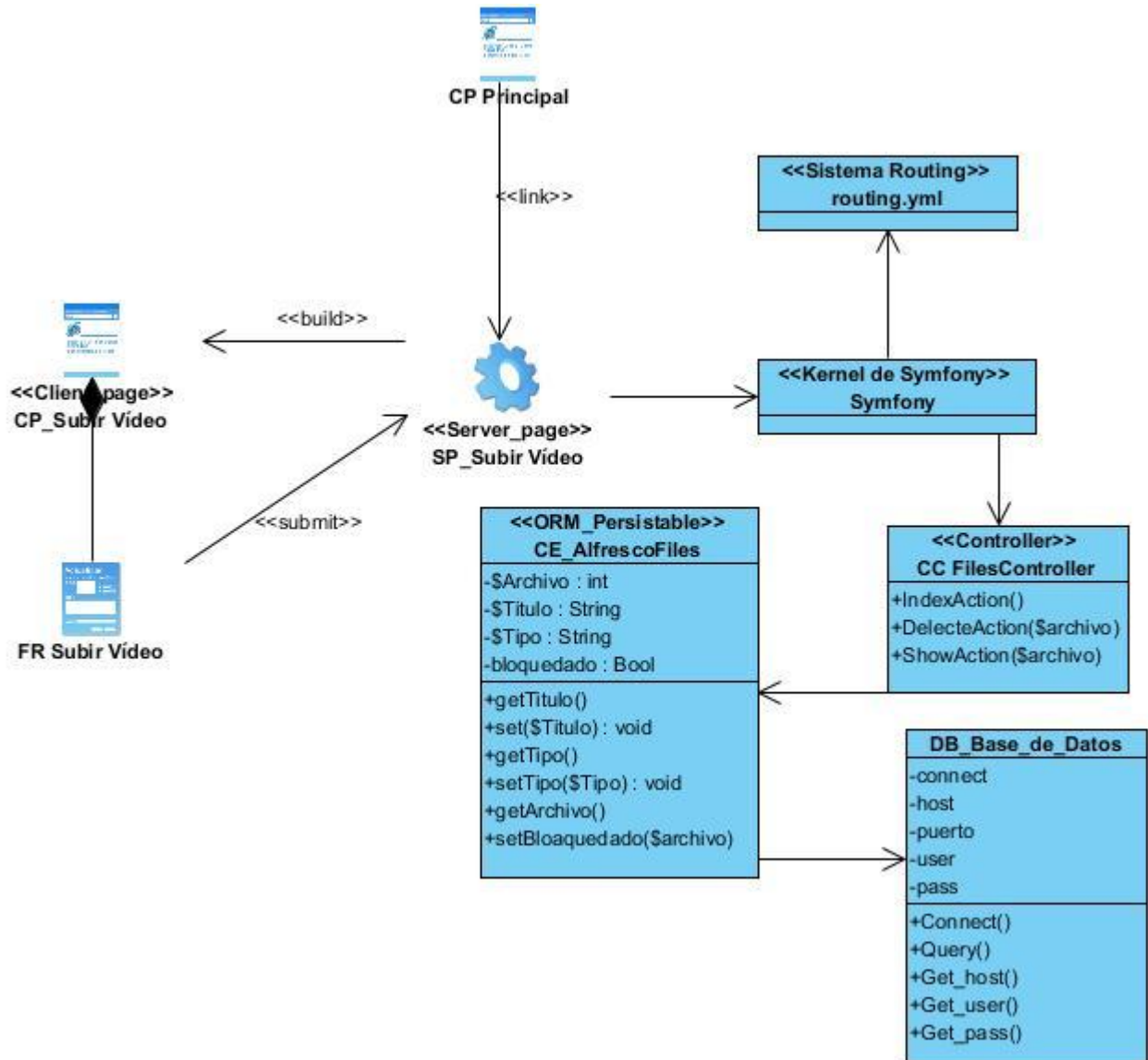


Figura 7: Diagrama de clases de estereotipos web Subir vídeo

3.3.4 Diagramas de Interacción

Fue utilizado para modelar el comportamiento dinámico de operaciones, se describe la interacción entre objetos a través de mensajes para cumplir las tareas en el sistema. Para representar dicho comportamiento se utilizó el diagrama de colaboración.

En la Figura 8 se muestra el diagrama de colaboración para el caso de uso **Autenticar Usuario**. Los diagramas correspondientes se encuentran en el Anexo 6:

Diagrama de Colaboración

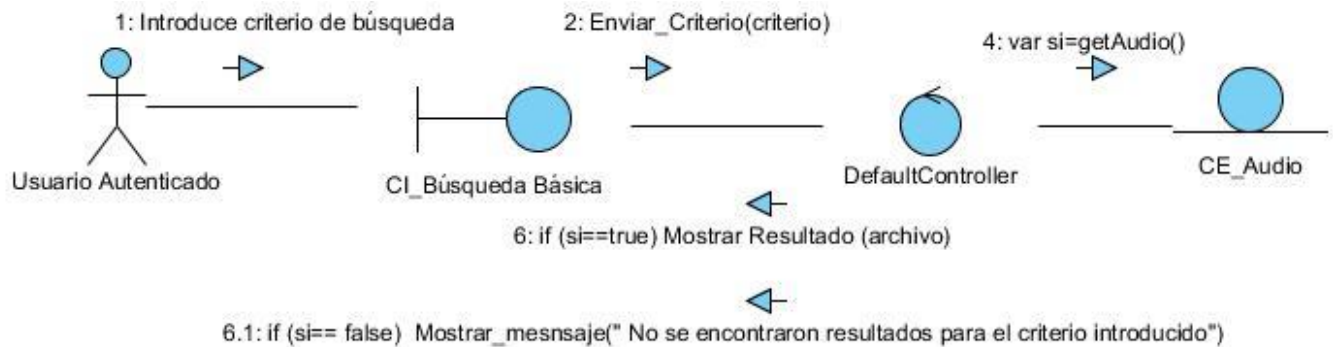


Figura 8: Diagrama de Colaboración del CU Autenticar usuario

3.3.5 Modelo de Datos

Un modelo de datos es aquel que describe la representación lógica y física de los datos persistentes en el sistema (Jacobson, y otros, 2000). En el desarrollo de la solución se ha hecho necesario garantizar la persistencia de los datos.

El sistema propuesto hace uso de una base de datos ya que es necesario persistir información relacionada con los contenidos de media.

En la Figura 9 se muestra el diagrama del Modelo físico de datos para la Base de Datos PostgreSQL:

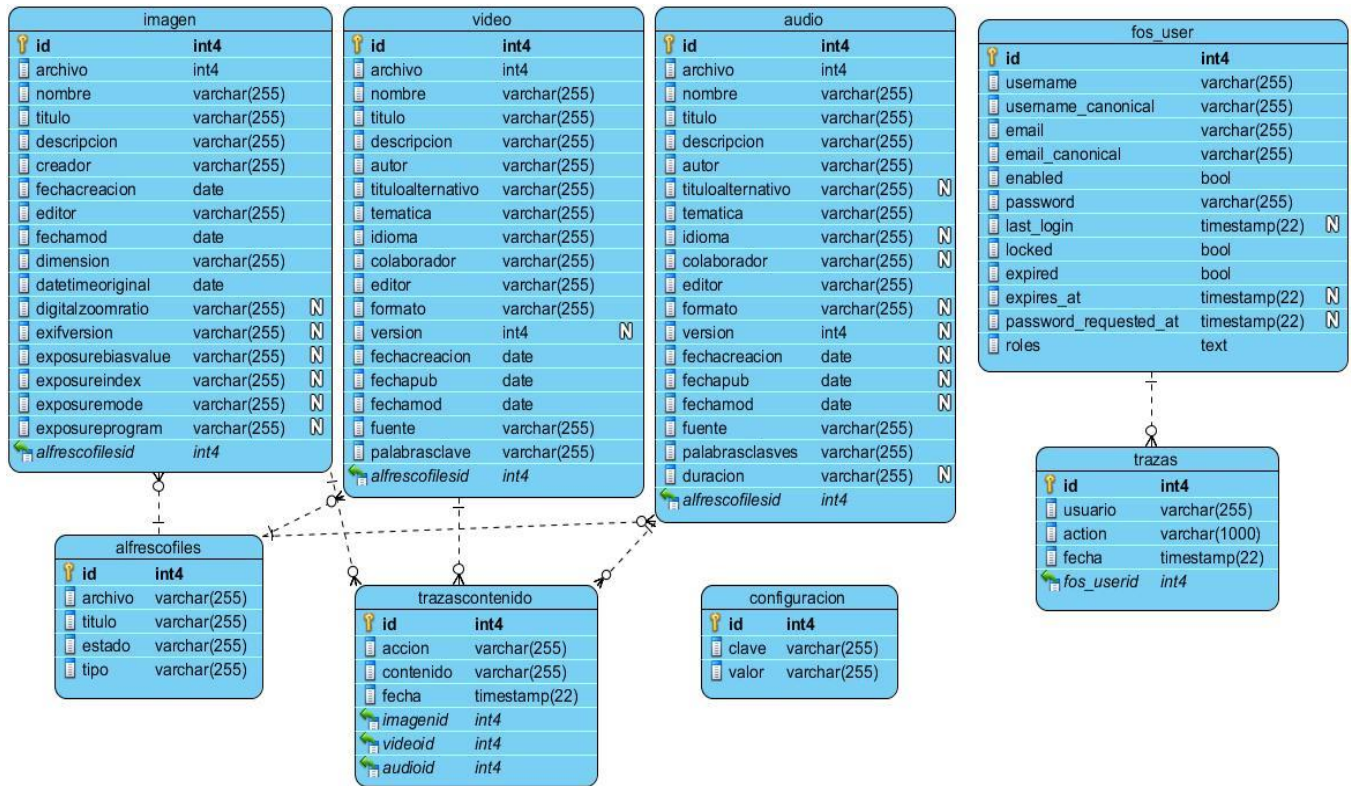


Figura 9: Modelo físico de datos para PostgreSQL

Descripción del modelo de datos

imagen: es la entidad encargada de almacenar todos los metadatos de las imágenes.

audio: es la entidad encargada de almacenar todos los metadatos de los audios.

vídeo: es la entidad encargada de almacenar todos los metadatos de los vídeos.

fosuser: entidad que guarda la información acerca de los usuarios del sistema.

trazas: es la entidad encargada de almacenar la información de las acciones que se realiza el usuario sobre los contenidos.

trazascontenido: es la entidad encargada de almacenar la información acerca de las acciones sobre los contenidos.

alfrescofiles: es la entidad encargada de gestionar los contenidos de medias que son cargados en el sistema.

configuración: es la entidad que guarda los valores para la conexión con el servidor *streaming*.

3.3.6 Modelo de Despliegue

El modelo de despliegue se utiliza para capturar los elementos de configuración del procesamiento y las conexiones entre los elementos físicos. También se utiliza para visualizar la distribución de los componentes de *software* en los nodos físicos.

Dicho modelo consiste en:

- ✓ **Nodos:** elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos.
- ✓ **Dispositivos:** nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.
- ✓ **Conectores:** expresa el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.

En la Figura 10 se muestra el diagrama de despliegue:

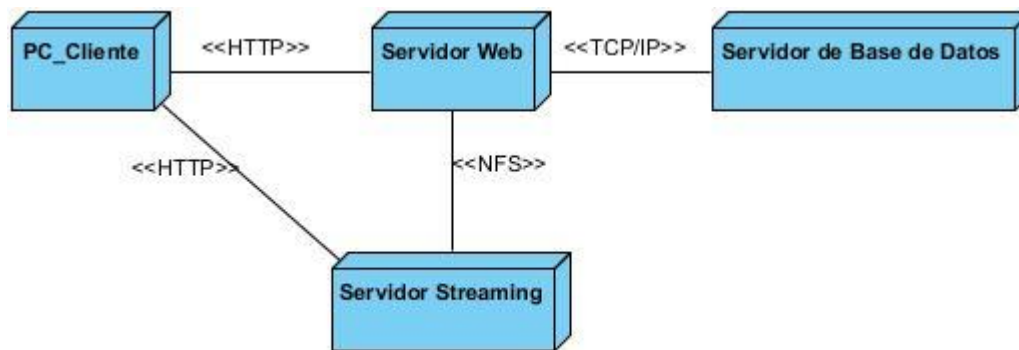


Figura 10: Modelo de Despliegue

Descripción del diagrama de Despliegue

- ✓ **Nodo PC_Cliente** es donde los usuarios pueden consultar y actualizar la información que se encuentra en el Servidor Web. Para acceder al Sistema Proveedor de ficheros de media, la PC_Cliente utiliza una aplicación cliente, en este caso, un navegador web. La comunicación entre las PC Cliente y el Servidor Web se establece a través del protocolo de comunicación HTTPS.
- ✓ **Nodo Servidor Web**, se ubican íntegramente las capas de presentación, lógica del negocio y de acceso a datos del sistema, así como los servicios que se brindan.
- ✓ **Nodo Servidor de Base de Datos** es donde se encuentra toda la información persistente asociada a los contenidos que son adicionados, actualizados y consultados por los usuarios en el sistema Proveedor de ficheros de media.
- ✓ **Nodo Servidor_Streaming** permite la posibilidad de realizar *streaming* de contenidos usando formatos y estándares libres como *Ogg/Theora/Webm*. Puede integrarse en el mismo servidor web, también como el sistema presenta una arquitectura escalable puede estar en un servidor separado y conectarse a los archivos de forma NFS.

Conclusiones parciales

En el presente capítulo se utilizaron los patrones GRASP garantizando una arquitectura sólida del sistema.

Se refinaron y estructuraron los requerimientos. Se describió la propuesta de solución con la representación de los diagramas de clases de análisis y los diagramas de clases del diseño. Se representó el diagrama de despliegue mostrando la distribución de los componentes de software en los nodos físicos. Se definieron los principios que guiarán la posterior implementación del sistema.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

El capítulo aborda los flujos de trabajo de Implementación y Prueba. La implementación se comienza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, *scripts*, ficheros de código binario, ejecutables y similares (Jacobson, 2000). Se desarrolló el diagrama de componentes el cual muestra como está dividido en componentes el sistema y las dependencias entre ellos. Finalmente concluye con la puesta a prueba de la solución obtenida, mediante la realización de pruebas funcionales, pruebas de seguridad y pruebas de carga y estrés con el objetivo de validar el sistema desarrollado y detectar posibles fallos.

4.1 Modelo de Implementación del sistema

El propósito de la fase de Elaboración es el establecimiento de la línea base de la arquitectura del sistema para proporcionar una base estable al grueso del diseño y del esfuerzo de implementación en la fase de Construcción. El flujo de trabajo de Implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes, es decir, ficheros de código fuente, *scripts*, ficheros de código binario, ejecutables y similares (Pressman, 2005).

4.1.1 Diagrama de componentes

El diagrama de componentes concibe un conjunto de elementos tales como, componentes, paquetes, subsistemas de implementación y sus relaciones. La idea principal de emplear este diagrama es para mostrar la estructura de nivel elevado del modelo de implementación enfatizado en los subsistemas de implementación y sus dependencias. Muestra como el sistema está dividido en componentes y las dependencias entre ellos. Ayuda a los desarrolladores a visualizar el camino de la implementación, permiten modelar la disposición física o topología de un sistema. Se utiliza para modelar la vista estática de un sistema, la organización y las dependencias entre un conjunto de clases (Jacobson, 2000).

En la Figura 11 se muestra el diagrama de componentes del sistema Proveedor de ficheros de media:

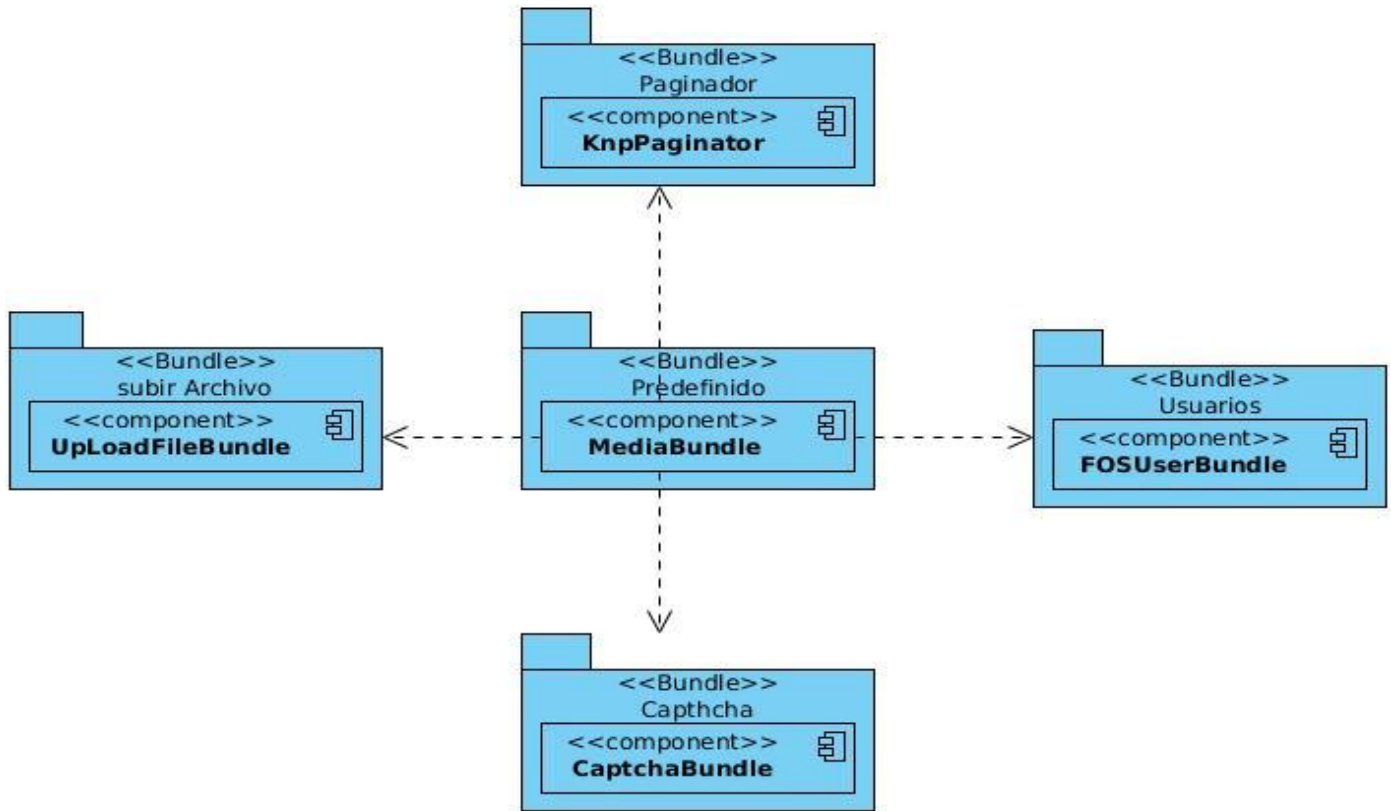


Figura 11: Diagrama de componentes

Descripción de los componentes

El diagrama representa un conjunto de componentes y elementos que garantizan el correcto funcionamiento del sistema. Se precisan los paquetes que interactúan con la estructura y configuración de la aplicación. A continuación se detallan cada una de las partes del diagrama expuesto en la Figura 11:

- ✓ **MediaBundle:** Agrupa los componentes que determinan una estructura organizativa del paquete. Se encuentran los paquetes (*Controller, Resources, Form y Entity*).
- ✓ **CaptchaBundle:** Posee un conjunto de funcionalidades con el objetivo de agregar código captcha a la aplicación. Es usado por el Componente MediaBundle el cual posee los formularios donde es utilizado este Bundle.
- ✓ **FOSUserBundle:** Posee un conjunto de funcionalidades para la gestión de usuarios. Es usado por el componente MediaBundle para crear, registrar y autenticar usuarios.
- ✓ **KnnpaginatorBundle:** Se utilizó para hacer la paginación de las plantillas que necesitaban mostrar varias páginas.
- ✓ **UploadFileBundle:** Almacena los archivos de enrutamiento y seguridad del paquete de Media.

- ✓ **Entity:** Son clases que almacenan un conjunto de datos relacional que permiten el trabajo orientado a objeto favoreciendo al manejo de la información de los archivos de media.
- ✓ **Form:** Contiene los formularios que son generados para la creación de los metadatos en clases, por ejemplo "VideoType.php".

4.2 Estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. La estandarización del código es muy importante para cualquier proyecto de desarrollo, principalmente si este involucra muchos desarrolladores, Lo que asegura que la solución sea de alta calidad, que contenga una cantidad baja de errores y sea fácil de mantener. Emplear técnicas de codificación concretas y efectuar buenas prácticas de programación con el objetivo de generar un código óptimo, es de vital importancia para la calidad del *software* y para obtener un buen rendimiento y mantenimiento del mismo.

El estándar de codificación fue definido por la dirección del proyecto del Sistema de Gestión Documental (SGD) el cual lleva por título "Estándares de codificación para PHP" que se encuentra registrado en el Expediente de Proyecto del Proveedor de ficheros de media.

En la Figura 12 se evidencia el estándar de código a utilizar y la estructura que debe llevar:

```

1  <?php
2  function ejemploDeLowerCamelCase($parametroEntrada){
3      →if($condicion)
4          →:→$variableAsignada = $valorCalculado + $parametroEntrada;
5      →else
6          →:→$variableAsignada = $valorCalculado + $parametroEntrada;
7      →return $variableAsignada;
8  }
9  ?>
```

Figura 12: Ejemplo del estándar de código en *lowerCamelCase*

4.2.1 Funciones y Métodos

- ✓ Los nombres de funciones deben empezar siempre con una letra minúscula. Cuando un nombre de función consiste en más de una palabra, la primera letra de cada nueva palabra debe estar en mayúsculas. Esto es llamado comúnmente como formato "*camelCase*".
- ✓ Por norma general, se recomienda la elocuencia. Los nombres de función deben ser lo suficientemente elocuentes como para describir su propósito y comportamiento.

4.2.2 Variables

- ✓ Al igual que los nombres de funciones los nombres de variables deben empezar siempre con una letra en minúscula y seguir la convención "*camelCaps*".

- ✓ Por norma general, se recomienda la elocuencia. Las variables deberían ser siempre tan elocuentes como prácticas para describir los datos que el desarrollador pretende almacenar en ellas. Variables escuetas como " \$i " y " \$n " están desaconsejadas, salvo para el contexto de los bucles más pequeños. Si un bucle contiene más de 20 líneas de código, las variables de índice deberían tener nombres más descriptivos.

4.2.3 Constantes

- ✓ Todas las letras pertenecientes al nombre de una constante deben aparecer en mayúsculas.
- ✓ Las palabras dentro del nombre de una constante deben separarse por barras bajas (_).
- ✓ Las constantes pueden contener tanto caracteres alfanuméricos como barras bajas (_).
- ✓ Los números están permitidos.

4.3 Interfaces principales del sistema



Figura 13: Interfaz Principal

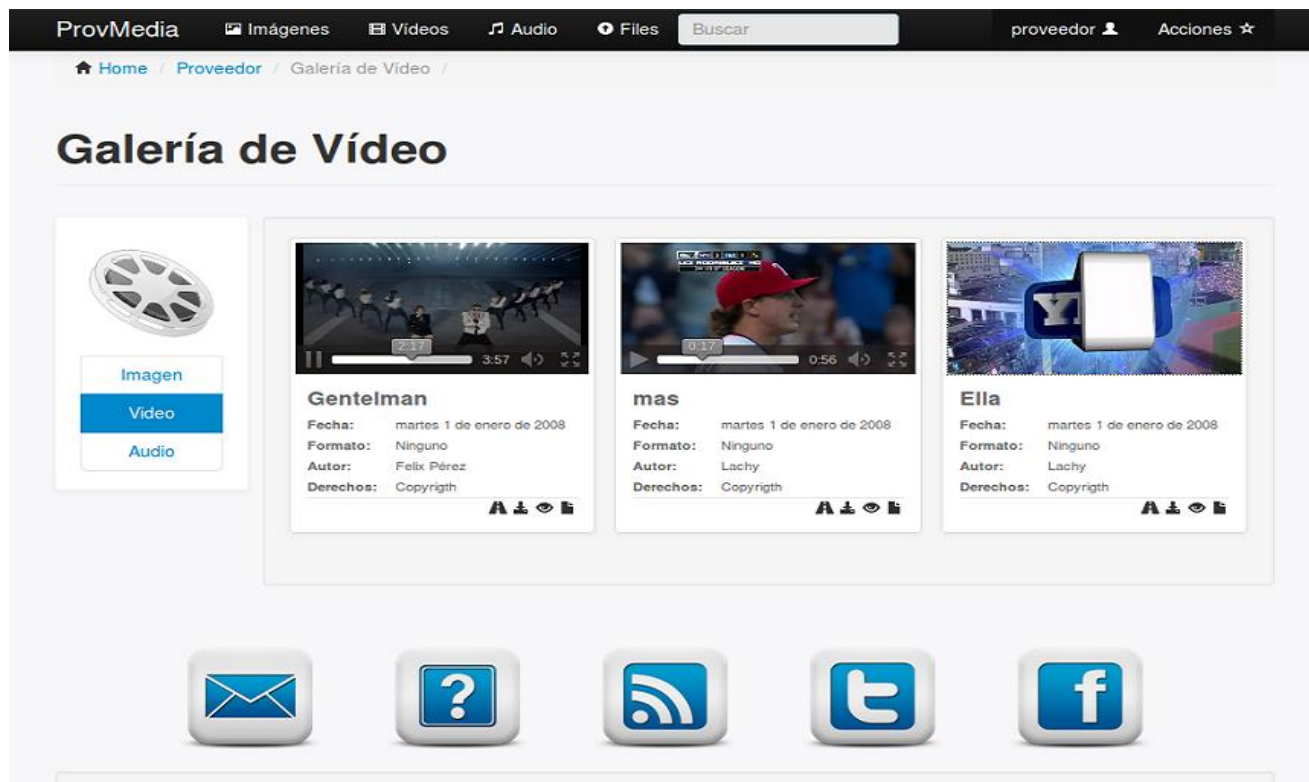


Figura 14: Interfaz de Galería de media



Figura 15: Interfaz Búsqueda avanzada

4.4 Pruebas de software

Uno de los elementos principales para certificar la calidad de una aplicación informática lo constituye el resultado de las pruebas que se le practican. Un control y una gestión de calidad bien implementados, especialmente durante el proceso de desarrollo del *software*, aumentan la calidad del sistema final, reducen los costos de avance y acortan el tiempo necesario para el mismo. Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto a las especificaciones principales del sistema (Pressman, 2005).

En el presente epígrafe se detallarán las pruebas funcionales y de fiabilidad realizadas a la aplicación.

4.4.1 Pruebas funcionales

El servicio de pruebas funcionales se centra en comprobar que los sistemas desarrollados funcionan acorde a las especificaciones funcionales y requisitos del cliente. Este servicio ayuda a su organización para detectar los posibles defectos derivados de errores en la fase de programación.

El método seleccionado para aplicar las pruebas es el de partición equivalente. Para la selección de este método se tuvo en cuenta que el mismo está orientado a los requisitos funcionales del *software* permitiendo que se creen juegos de datos para lograr probar a profundidad todas las funcionalidades del sistema. Como se planteó anteriormente, de este método se pueden derivar casos de pruebas que aseguren que se detecten errores de forma inmediata. Para el desarrollo de este método se realizó el diseño de casos de pruebas empleando la técnica de caja negra.

En la Tabla 12 se muestran el diseño de caso de prueba funcional a realizar a la interfaz del sistema específicamente al caso de uso **Autenticar usuario**. El resto de los casos de prueba se encuentran en el Anexo 7.

Diseño de Caso de Prueba basado en el CU Autenticar usuario

Descripción general: El caso de uso se inicia cuando se introduce el usuario y la contraseña para acceder a la aplicación, estos son verificados; el caso de uso finaliza mostrando la interfaz correspondiente según los permisos del usuario que se autentica.

Condiciones de ejecución: Para autenticarse en el sistema el usuario debe introducir los datos que requiere la aplicación.

Tabla 12: SC Autenticar usuario

Escenario	Descripción	Usuario	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Datos correctos.	Para autenticarse en el sistema el	V	V	Si los datos son introducidos correctamente y	El usuario llena todos los campos correctamente y

	usuario introduce su nombre de usuario y contraseña.			validados, el sistema mostrará la interfaz correspondiente al usuario.	da clic en Autenticar.
EC 1.2 Campos vacíos.	El usuario deja los campos vacíos.	I	I	Si alguno de los campos están vacíos el sistema muestra un mensaje de error, "Existen campos vacíos"	El usuario deja algún campo vacío y presiona el botón Aceptar.
EC 1.3 Datos introducidos incorrectamente.	El usuario introduce los datos incorrectamente.	V	I	Si el campo contraseña es inválido el sistema muestra un mensaje de error: "Contraseña incorrecta"	El usuario llena los campos pero introduce el dato de la contraseña incorrecto.
EC 1.4 Datos introducidos incorrectamente.	El usuario introduce los datos incorrectamente.	I	V	Si el campo usuario es inválido el sistema muestra un mensaje de error: "Usuario incorrecto"	El usuario llena los campos pero introduce el dato del usuario incorrecto.

Las celdas de la tabla contienen V, I. V indica válido, I indica inválido.

Resultados obtenidos

Durante el desarrollo del sistema se realizaron pruebas unitarias para ir comprobando el funcionamiento del mismo. Para evaluar la solución se realizaron varias iteraciones donde se probó el *software* íntegramente. Se realizaron las pruebas a los casos de uso y arrojaron errores que se incorporaron al documento de no conformidades. Se realizaron un grupo de iteraciones donde se detectaron veinticuatro no conformidades agrupadas en cuatro tipos de errores: mensajes del sistema escritos incorrectamente, mensajes del sistema que no correspondían con la acción ejecutada, campos no validados y funcionalidades con deficiencias.

En la Tabla 13 se evidencia los resultados de las tres iteraciones de pruebas funcionales realizadas al sistema:

Tabla 13: Resultado de las pruebas

Casos de Uso	Iteración	NC	Cerrada (S)	No procede
Realizar búsqueda avanzada.	1ra	12	11	1
Realizar búsqueda básica.	2da	8	6	3
Generar reportes de contenidos.	3ra	4	7	0

Como resultado de las pruebas se detectaron no conformidades que presentaba el sistema las cuales fueron corregidas a través de las iteraciones realizadas a la aplicación permitiendo obtener un producto libre de errores.

4.4.2 Pruebas de Seguridad

Las pruebas de seguridad certifican que los datos del sistema solamente sean accedidos por los actores definidos según los niveles de acceso que posean. Tienen como objetivo evaluar la confidencialidad, integridad y disponibilidad de los datos. Estas pruebas de seguridad garantizan (V\$V Quality, 2006):

- ✓ Que los usuarios estén restringidos a funciones específicas o su acceso está limitado únicamente a los datos a los que estén autorizados a acceder.
- ✓ Que solo aquellos usuarios autorizados a acceder al sistema son capaces de ejecutar las funciones del sistema.

La seguridad del sistema se encuentra basada en los distintos roles definidos para acceder a ella. Solo los usuarios que posean dichos roles pueden acceder a la aplicación y realizar las operaciones que para cada uno de ellos están definidas.

En la Tabla 14 se muestra la prueba de seguridad realizada al CU Autenticar usuario:

Tabla 14: Escenario para el CU Autenticar usuario

EC	Nombre de usuario	Contraseña	Respuesta del sistema	Resultado de la Prueba
EC 1.1: Autenticar Usuario	Usuario autenticado	xxxxxxxxxx	Al autenticarse en el sistema con el rol de usuario autenticado, el sistema debe permitir al usuario: <ul style="list-style-type: none"> ✓ Realizar búsqueda 	Satisfactorio. El sistema permite al usuario realizar las operaciones definidas.

			básica y avanzada. ✓ Visualizar archivo. ✓ Descargar archivo.	
Administrador	xxxxxxxxx	Al autenticarse en el sistema con el rol de administrador, el sistema debe permitir al usuario:	✓ Gestionar búsqueda. ✓ Gestionar usuario. ✓ Gestionar reporte. ✓ Gestionar permiso. ✓ Exportar JSON.	Satisfactorio. El sistema permite al usuario realizar todas las operaciones definidas.
Proveedor	xxxxxxxxx	Al autenticarse en el sistema con el rol de Proveedor, el sistema debe permitir al usuario:	✓ Gestionar media ✓ Gestionar metadato ✓ Subir archivos de media	El sistema permite al usuario realizar todas las operaciones definidas.

Resultados obtenidos de las pruebas de seguridad

Para probar la seguridad del sistema se aplicaron una serie de pruebas usando el software Acunetix WVS Reporter v8.0 el cual tras haberlo escaneado generó un reporte de las fallas de seguridad más importantes y comunes como: SQL injection, Cross Site Scripting, ataques por fuerza bruta y otros. En una primera iteración se detectaron un grupo de fallos entre los que se encuentran: envío de información sensible en texto plano y la no utilización del protocolo seguro https, enlaces rotos, formularios sin seguridad, los cuales son clasificados en alto, medio y bajo. De los fallos encontrados se detectaron dos riesgos de clasificación alto, diez de nivel medio y ocho de nivel bajo.

Se realizó una segunda iteración donde fueron corregidos todos estos fallos. Se concluye que el mecanismo de seguridad del sistema funciona correctamente y garantiza la integridad y disponibilidad de los recursos de media.

4.4.3 Pruebas de Rendimiento

Pruebas de carga: Mediante la ejecución de las pruebas de Carga es posible identificar la capacidad de recuperación de un sistema cuando es sometido a cargas variables, tanto de usuarios como de procesos. Al realizar las pruebas de carga se puede determinar el tiempo de respuesta de todas las transacciones críticas del sistema y encontrar cuellos de botella de la aplicación (V & V Quality, 2006).

Prueba de estrés: Mediante las pruebas de estrés es posible identificar la capacidad de respuesta de un sistema bajo condiciones de carga extrema, representadas por una alta concurrencia de Usuarios y/o Procesos. Una vez realizadas las pruebas de estrés se podrá conocer el punto de quiebre del aplicativo en términos de capacidad de respuesta, con lo cual será posible establecer acciones de optimización en diferentes niveles para asegurar una mejor capacidad de concurrencia de usuarios y/o procesos, que se verá reflejada en una óptima operación de negocio (V & V Quality, 2006).

Para comprobar la manera en que se comportó la aplicación en cuanto al uso de los recursos y su estabilidad en los momentos de carga máxima, se realizó la prueba de rendimiento con la herramienta JMeter. Se debe señalar que la simulación fue realizada en un servidor con las siguientes características: sistema operativo Ubuntu 12.04, tipo de procesador: Pentium 4, Core i3 3.1, 2 GB RAM.

La prueba brindó información importante sobre la carga y el estrés que puede soportar el sistema Proveedor de ficheros de media. Como resultado arrojó, que para 300 usuarios concurrentes en un intervalo de un segundo el sistema obtuvo como tiempo de respuesta mínimo 3.25 segundos y como máximo demoró 5.83 segundos procesando en general 4.3 peticiones por segundo con una media de 2949 respuestas satisfactorias. Se demuestra que la aplicación es estable para estos 300 usuarios pues se mantuvo prestando servicios todo el tiempo para un margen de error de 0.50 %, lo cual muestra que se mantiene en funcionamiento en situaciones donde concurren gran cantidad de usuarios trabajando sobre ella, accediendo y modificando información de manera simultánea. Los resultados obtenidos revelan la capacidad de respuesta y procesamiento del sistema. En la Figura 16 se muestran los resultados de las pruebas de rendimiento:

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento
/ProvMedia/web/app.php/login	360	1946	2001	2441	73	3112	0,00%	5,6/sec
/ProvMedia/web/app.php/	600	1764	1746	2086	105	2923	0,00%	8,0/sec
/ProvMedia/web/app.php/contactanos	120	2028	1997	2466	1071	2808	0,00%	13,7/sec
/ProvMedia/web/app.php/ayuda	120	1941	1919	2364	1343	2818	0,00%	12,1/sec
/ProvMedia/web/app.php/imagen/	240	4783	4854	5332	1875	6433	3,75%	3,1/sec
/ProvMedia/web/app.php/login_check	240	3874	3904	4374	2711	4984	0,00%	4,3/sec
/ProvMedia/web/app.php/user	240	1693	1711	2019	844	2384	0,00%	4,4/sec
/ProvMedia/web/app.php/alfrescofiles/	480	3512	3495	3901	2864	4348	0,00%	19,3/sec
/ProvMedia/web/app.php/alfrescofiles/22/show	120	3166	3147	3470	2673	3870	0,00%	10,9/sec
/ProvMedia/web/app.php/profile/change-password	120	3841	3818	4493	3043	4782	0,00%	9,9/sec
/ProvMedia/web/app.php/profile/edit	120	3895	3914	4216	2708	4673	0,00%	9,7/sec
/ProvMedia/web/app.php/logout	240	2999	3019	3376	1736	3895	0,00%	7,5/sec
/ProvMedia/web/app.php/busqueda	120	5598	5623	6061	4887	6833	0,00%	7,8/sec
/ProvMedia/web/app.php/imagen/27/show	120	3751	3800	4772	1923	5539	0,00%	10,4/sec
TOTAL	3240	2949	2977	4626	73	6833	0,28%	39,8/sec

Figura 16: Resultados de las pruebas de rendimiento

Conclusiones parciales

En el presente capítulo se generó el diagrama de componentes mostrando la estructura física de la aplicación. Se estableció el estándar de codificación utilizado para el desarrollo de la aplicación web lo que permitió lograr una mayor legibilidad y limpieza en el código. Se validó el sistema con la realización de las pruebas funcionales, de seguridad y de carga y estrés, garantizando la corrección de los errores identificados durante el ciclo de vida del producto. Como resultado de la implementación se obtuvo un sistema funcional que cumple con los requerimientos especificados.

Conclusiones Generales

- ✓ Con el estudio realizado del panorama nacional e internacional, teniendo en cuenta las tendencias en el desarrollo de los sistemas proveedores de media, se definió la necesidad de desarrollar una aplicación orientada a resolver la descentralización de los recursos de media, dígame imagen, audio y vídeo de los portales de los medios de prensa cubanos.
- ✓ Fueron seleccionadas las tecnologías, herramientas y la metodología que guiaron el proceso de desarrollo del sistema Proveedor de ficheros de media para los portales de los medios de prensa cubanos.
- ✓ Durante la fase de Análisis y Diseño se logró obtener una representación de la estructura general del sistema. Los artefactos y actividades generados, basados en la metodología Open Up, lograron una mejor obtención de los requerimientos para el posterior desarrollo de la implementación.
- ✓ Se desarrolló el sistema Proveedor de ficheros de media dando respuesta a las necesidades del cliente, permitiendo garantizar la centralización de los recursos de media en una aplicación que suministra los archivos de media a los portales de los medios de prensa cubanos.
- ✓ Se realizaron las pruebas funcionales, de seguridad y de rendimiento con las cuales se pudo comprobar el adecuado funcionamiento de la aplicación.

El presente trabajo proporcionó el desarrollo del sistema Proveedor de media para los portales de los medios de prensa cubanos cumpliendo con los requerimientos establecidos.

Recomendaciones

- ✓ Implementar otras funcionalidades que permitan la gestión de los recursos de media tales como: evitar la duplicación de contenidos en cuanto al fichero de media suministrado a al sistema Proveedor de media.

Referencias Bibliográficas

- Alvarez, Miguel Angel. 2009.** desarrolloweb.com. *Que es HTML 5.* [En línea] 2009. <http://www.desarrolloweb.com/articulos/que-es-html5.html>.
- , **2010.** Manual de Javascript I . *Introducción a Javascript.* [En línea] 2010. <http://www.desarrolloweb.com/manuales/20/>.
- Apache. 2012.** APACHE SOFTWARE FOUNDATION. *HTTP SERVER PROJECT.* [En línea] 2012. <http://httpd.apache.org>.
- Apache JMeter. 2013.** The Apache JMeter Foundation. *Apache JMeter.* [En línea] 2013. <http://jmeter.apache.org/>.
- Ciberaula. 2010.** Introducción, definición y evolución de PHP. *Ciberaula.* [En línea] 2010. <http://php.ciberaula.com/articulo/introduccion.php>.
- Cuervo, Victor. 2009.** Ayuda en la Web. *Que es Flickr?* [En línea] 2009. [Citado el: 20 de Enero de 2013.] <http://www.ayudaenlaweb.com/fotografia/que-es-flickr/>.
- Flumotion. 2012.** CRYSQL Ciudad Real Software Libre. *What is Flumotion?* [En línea] 2012. <http://www.flumotion.net/>.
- Galán, Bonifacio Martín y Davis Rodriguez Mateo Arellano Pardo MC, Hernández Pérez A, Nogales Flores JT. 2005.** Gestión de contenidos Web mediante herramientas de software libre. [En línea] 2005. IX Jornadas Españolas de Documentación FESABIDB.
- GenBeta. 2012.** GenBeta desarrollo y software. [En línea] 2012. [Citado el: 20 de Febrero de 2013.] <http://www.genbetadev.com/frameworks/bootstrap>.
- Hernández, Yulainne Alonso. 2012.** *Configuración de la Metodología OpenUP.* Centro Ideoinformática. Ciudad Habana : s.n., 2012.
- International Press Telecommunications Council. 2012.** IPTC. International Press Telecommunication Council. *IPTC STANDARDS UPDATE.* [En línea] 2012. [Citado el: 28 de Enero de 2013.] <http://www.iptc.org/site/Home/>.
- Jacobson, Ivar y Booch, James Rumbaugh Grady. 2000.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Addison Wesley, 2000.
- Jacobson, Rumbaugh Booch. 2000.** *El Proceso Unificado de Desarrollo de Software.* Madrid : s.n : s.n., 2000.
- JSON. 2011.** Introducción a JSON. *JSON.* [En línea] 2011. [Citado el: 10 de Febrero de 2013.] <http://www.json.org/json-es.html>.
- Lagoze, Carl. 2012.** Iniciativa de los Archivos Abiertos. [En línea] 2012. <http://www.openarchives.org/ore/0.9/datamodel>.

- Lee, Tim Berners. 2009.** Linked Data. *RDF*. [En línea] 2009. [Citado el: 6 de Febrero de 2013.] <http://www.w3.org/DesignIssues/LinkedData.html>.
- León, Eduardo. 2010.** Scribd. *Tutorial Visual Paradigm for UML*. [En línea] 2010. <http://es.scribd.com/doc/36636137/Tutorial-Visual-Paradigm>.
- Maestros del web. 2004.** Maestros de la Web. *Estándares Web*. [En línea] 2004. [Citado el: 25 de Enero de 2013.] <http://www.maestrosdelweb.com/editorial/estandaresweb/>.
- . 2012.** Symfony Framework PHP orientado a objetos. *Introducción a Symfony 2*. [En línea] 2012. [Citado el: 20 de Enero de 2013.] <http://www.maestrosdelweb.com/editorial/curso-symfony2-introduccion-instalacion/>.
- Martinez, Rafael. 2010.** PosgreSQL-es. *Sobre PostgreSQL*. [En línea] 2010. http://www.postgresql-es.org/sobre_postgresql.
- NetBeans. 2012.** NetBeans. *Bienvenido a NetBeans y www.netbeans.org*. [En línea] 2012. http://netbeans.org/index_es.html.
- Pérez, Yeter Caraballo. 2007.** La gestión de contenidos en portales Web. [En línea] 2007. [Citado el: 15 de Enero de 2013.] http://bvs.sld.cu/revistas/aci/vol15_3_07/aci07307.htm.
- PICASA. 2012.** PICASA. *Características*. [En línea] 2012. [Citado el: 21 de Enero de 2013.] <http://informacionsobrepicazza.blogspot.com/2012/04/caracteristicas.html>.
- Potencier, Fabien. 2012.** Symfony Oficial Site. *Symfony ciene cuando PHP framework*. [En línea] 2012. <http://symfony.com/>.
- Pressman, Roger. 2002.** Ingeniería de Software, un enfoque práctico. [En línea] 2002.
- . 2005.** *Ingeniería de Software. Un enfoque práctico*. s.l. : España : s.n, 2005. 9701054733.
- RSS. 2012.** RSS Explicado. *¿Qué es RSS?* [En línea] 2012. [Citado el: 4 de Febrero de 2013.] <http://www.rss.nom.es/>.
- Sierra, Manuel. 2013.** APRENDERAPROGRAMAR.COM. *Que es CSS*. [En línea] 2013. http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=492.
- 2009.** Social Media Defined. *flickr Defined*. [En línea] 2009. [Citado el: 20 de Enero de 2013.] <http://www.socialmediadefined.com/2009/01/06/flickr-defined/>.
- Social Media Defined. 2009.** Social Media Defined. *flickr Defined*. [En línea] 2009. <http://www.socialmediadefined.com/2009/01/06/flickr-defined/>.
- TeVeo. 2012.** TeVeo servicio de audio y vídeo bajo demanda. [En línea] 2012. [Citado el: 15 de Enero de 2013.] <http://teveo.icrt.cu/>.
- Tramullas, Jesús. 2005.** Herramientas de software libre para la gestión de contenidos. *Hipertext.net*. [En línea] 2005. [Citado el: 20 de Enero de 2013.] <http://www.hipertext.net>.

- Troncy, Raphael. 2007.** NewsML and the Semantic Web. [En línea] 2007. [Citado el: 28 de Enero de 2013.] <http://newsml.cwi.nl>.
- V & V Quality. 2006.** V & V Quality. *PRUEBA DE ESTRÉS*. [En línea] 2006. <http://www.vyvquality.com/w1/index.php/servicios/pruebas-de-rendimiento/pruebas-de-estres.html>.
- . 2006.** V & V Quality. *PRUEBAS DE CARGA*. [En línea] 2006. <http://www.vyvquality.com/w1/index.php/servicios/pruebas-de-rendimiento/pruebas-de-carga.html>.
- V\$V Quality. 2006.** V&V Quality. *PRUEBAS DE SEGURIDAD*. [En línea] 2006. <http://www.vyvquality.com/w1/index.php/servicios/pruebas-de-seguridad.html>.
- Vimeo. 2009.** Alojamiento y gestión de material multimedia. *Principales Características*. [En línea] 2009. [Citado el: 20 de Enero de 2013.] http://www.edukanda.es/mediatecaweb/data/zip/972/vimeo/vimeo_02.htm.
- YouTube. 2013.** YouTube. *Qué es YouTube*. [En línea] 2013. [Citado el: 15 de Enero de 2013.] <http://www.queesyoutube.com>.

Bibliografía

A García Jiménez - Anales de documentación, 2004 - revistas.um.es

Javier Eguiluz, Desarrollo web ágil con Symfony2, 15 de octubre del 2012. Publicaciones **easybook** versión **4.8-DEV**.

Roger S. Pressman, Ingeniería de software 5 edición Un enfoque práctico.

PotencierFabien, ZanonottoFrancois. Symfony la Guia definitiva.

Hernández León, Rolando Alfredo, Coello González Sayda. El proceso de la investigación científica, 2011.

Ideoinformática, Centro. Configuración de la metodología OpenUP. Ciudad de La Habana. Cuba: s.n., 2012.

Jacobson, Ivar.El proceso unificado de desarrollo de software. Madrid: Addison Wesley, 2000.

Peña, Ing. Yadira Machado. Planificación, Diseño y Ejecución de Pruebas Funcionales. Habana:

Calisoft, marzo-2011.

PHP, 2013. [Disponible en:<http://www.php.net/manual/es/intro-what-is.php>.]

Glosario de términos

API: API (del inglés *Application Programming Interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción. Son usadas generalmente en las bibliotecas.

SOAP: Es un protocolo estándar que define cómo dos objetos en diferentes procesos, se comunican por medio del intercambio de datos.

XML-RPC: protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes SOAP.

REST: Transferencia de Estado Representacional, técnica de arquitectura *software* para sistemas hipermedia distribuidos, para interfaz web simple que utiliza **XML** y **XML-RPC**.

Streaming: Consiste en una tecnología utilizada para permitir la visualización y la audición de un archivo mientras se está descargando, a través de la construcción de un buffer por parte del cliente, una vez que este se ha conectado al servidor, el buffer del cliente se va llenando de la información descargada y se va reproduciendo en el ordenador. El sistema se encuentra sincronizado, tal que, una vez terminada la reproducción del contenido del archivo, finaliza la descarga (siempre y cuando no existan interrupciones en el envío del archivo).

Modelo del dominio: Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los elementos que existen o los eventos que suceden en el entorno en el que trabaja el sistema.

Anexos

Anexo 1: Entrevista

1. ¿Quién realiza directamente la solicitud del proyecto?
2. ¿Hacia quién está dirigida la solución?
3. ¿Qué problemas resolverá la solución de ser exitosa?
4. ¿Qué problemas específicos debería resolver la solución?
5. ¿Qué funcionalidades debe brindar este sistema para satisfacer con las necesidades existentes dentro del proyecto?
6. ¿Qué tecnologías son necesarias para integrar el sistema?
7. ¿Qué tecnologías se utilizan en otros proyectos relacionados con el sistema que se desea implantar?
8. ¿Por qué se eligió la tecnología, que tiene en particular?
9. Explique las herramientas que resuelvan problemas similares que conozca.
10. ¿Dónde están las tecnologías con las que hay que integrar y el acceso (dirección IP, usuario y contraseña) con permisos para poder desarrollar las funciones requeridas?
11. ¿Dónde está la Documentación que tiene sobre las tecnologías a integrarse?
12. ¿Qué elementos serán necesarios gestionarse?
13. ¿Cuántos usuarios podrán acceder a la aplicación?
14. ¿Qué estándares es necesario utilizar para la comunicación y la representación de los datos en el sistema?
15. ¿Qué artefactos y que metodología requiere el proyecto que se le entreguen?

Anexo 2: Especificación de Requisitos de Software

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF1	Autenticar usuario.	Para autenticarse en el sistema los usuarios introducen su nombre de usuario y contraseña, si los datos son introducidos correctamente, el sistema mostrará la interfaz correspondiente al usuario.	Baja	Alta
Prototipo				
				
Campos	Tipos de Datos		Reglas o Restricciones	
1.Usuario 2.Contraseña	Texto		El usuario debe introducir su identificador y su contraseña.	
Observaciones	En caso de introducir los datos incorrectamente se muestra un mensaje: "Debe introducir los datos correctamente" indicando que tipo de campo presentó problemas.			

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 8	Auditar cuenta de usuario.	El usuario administrador una vez autenticado en el sistema accede al vínculo Usuarios.	Media	Media

		Seguido aparece un listado de los usuarios registrados, después accede al vínculo Trazas de las Acciones disponibles sobre el usuario. El sistema muestra un reporte de incidencias del usuario seleccionado.		
Prototipo				
Campos	Tipos de Datos	Reglas o Restricciones		
	No procede	Que la cuenta de usuario se encuentre registrada en el sistema.		
Observaciones	En caso de que no existan incidencias se lanzará un mensaje: "El usuario "" no tiene trazas".			

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF 50	Generar reporte de contenidos.	El administrador una vez autenticado en el sistema procede a posicionarse en la sección de Administración, luego selecciona el vínculo de Generar reportes, donde se genera un reporte estadístico de las incidencias de los usuarios con el sistema es decir, muestra las media más seleccionadas, las más populares y las más gustadas.	Media	Media
Prototipo				

Campos	Tipos de Datos	Reglas o Restricciones
	No procede	Deben de haberse realizado operaciones de los usuarios dentro del sistema.
Observaciones	En caso de no mostrarse ningún reporte se muestra un mensaje: "No se realizó ninguna operación en el sistema por parte de los usuarios registrados".	

Anexo 3: Especificación de caso de uso

CU 1: Autenticar usuario

Objetivo	Autenticar el usuario en el sistema según los permisos que le hayan sido asignados.
Actores	Usuario Anónimo, Usuario Administración, Usuario Autenticado.
Resumen	El caso de uso se inicia cuando se introduce el usuario y la contraseña para acceder en la aplicación, estos son verificados; el caso de uso finaliza mostrando la interfaz correspondiente según los permisos del usuario que se autentica.

Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	El usuario debe existir en la Base de Datos y en el servidor LDAP y debe estar habilitado en el sistema.	
Poscondiciones	Se autentica el usuario anónimo, administración y autenticado.	
Flujo de eventos		
	Actor	Sistema
1	1. El caso de uso se inicia cuando el actor escribe su nombre de usuario (X) y contraseña (Y). Debe presionar el botón "Aceptar" (Z) para entrar en el sistema.	1.1 El sistema verifica que exista conexión con la base de datos.
2		1.2. El sistema verifica que el usuario exista en la Base de Datos y coincida su contraseña.
3		1.3 El sistema verifica los permisos que tiene asignados y en dependencia de los permisos le permite acceder a la aplicación mostrando la interfaz correspondiente.
Flujos alternos		
	Actor	Sistema
1		1.1 Si no se puede establecer conexión con la Base de Datos, el sistema muestra el siguiente mensaje: "No se puede establecer conexión con los datos", y culmina el caso de uso.
2		1.2 Si el usuario no existe en la Base de Datos del sistema, se muestra un mensaje indicando que los datos de autenticación no son correctos y culmina el caso de uso.
3		1.3 Si no coincide la contraseña del

		usuario se establece conexión con el servidor LDAP, se comparan las contraseñas del usuario y si coinciden las mismas se actualiza la contraseña almacenada en la Base de Datos del sistema. Ir a la acción 1.3 del Flujo normal de eventos.
4		1.4 Si no se puede establecer conexión con el servidor LDAP, el sistema muestra el siguiente mensaje: “No se puede establecer conexión con el servidor LDAP”, y culmina el caso de uso.
Flujos alternos		
	Acción del Actor	Respuesta del Sistema
		1.1 Si los datos de autenticación introducidos por el actor no son válidos, el sistema muestra el siguiente mensaje: “Usuario o contraseña incorrectos”. Culmina el caso de uso.

CU 2: Gestionar usuario

Objetivo	Realizar la Gestión de los usuarios registrados en el sistema.
Actores	Usuario Administración.

Resumen	El caso de uso se inicia cuando el Usuario Administrador realiza las acciones de insertar, modificar o eliminar los datos de los usuarios existentes en el sistema. Culmina el caso de uso al realizar una de las mismas.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	El usuario debe estar autenticado y poseer permisos para la gestión de usuarios.	
Poscondiciones	Se insertan, modifican o eliminan los datos de usuario en dependencia de las acciones del actor.	
Flujo de eventos		
	Actor	Sistema
1	<p>1. El caso de uso se inicia cuando el actor selecciona una de las opciones en el menú general para la gestión de usuarios, puede seleccionar:</p> <ul style="list-style-type: none"> ✓ Insertar Usuario, ir a la sección "Insertar Usuario" ✓ Modificar Usuario, ir a la sección "Modificar Usuario" ✓ Eliminar Usuario, ir a la sección "Eliminar Usuario" 	
	Sección "Insertar Usuario"	
	Acción del Actor	Respuesta del Sistema
		<p>1.1 El sistema verifica que exista conexión con la Base de Datos.</p> <p>1.2 El sistema muestra la interfaz de gestión de usuarios.</p>
2	2. El actor realiza la búsqueda del usuario que desea insertar al sistema dentro de los usuarios de LDAP	
3	3. El actor selecciona la opción "Insertar"(A) del usuario que desea agregar en el sistema	3.1 El sistema carga los datos del usuario seleccionado en la interfaz de

	de la lista de usuarios de LDAP.	<p>inserción de usuario.</p> <ul style="list-style-type: none"> ✓ Usuario (B) ✓ Nombre (C) ✓ Apellidos (D)
4	<p>4. El actor introduce el resto de los datos necesarios para la creación de un usuario en el sistema:</p> <ul style="list-style-type: none"> ✓ Identificación (obligatorio) (E) ✓ Rol (obligatorio) (G) ✓ Habilitado (obligatorio) (H) 	4.1 El sistema valida cuando se han introducido todos los datos necesarios para la creación del usuario y habilita el botón "Guardar" (I).
5	5. El actor pulsa el botón "Guardar" (I)	5.1 El sistema muestra un mensaje de notificación de la acción. Culmina el caso de uso.
Flujos alternos		
	Acción del Actor	Respuesta del Sistema
1		1.2. Si no se puede establecer conexión con la Base de Datos, el sistema muestra el siguiente mensaje: "No se puede establecer conexión con los datos." Culmina el caso de uso.
Sección: "Modificar usuario"		
	Acción del Actor	Respuesta del Sistema
1		<p>1.1 El sistema verifica que exista conexión con la Base de Datos.</p> <p>1.2 El sistema muestra la interfaz de gestión de usuarios.</p>
2	2. El actor selecciona la opción "Modificar" (J) del usuario que desea modificar dentro de la lista de usuarios del sistema.	2.1 El sistema carga los datos del usuario seleccionado en la interfaz correspondiente permitiendo editarlos. (K).
3	3. El actor modifica los datos de usuario.	3.1 El sistema habilita el botón "Guardar" (I) en caso de que se hayan efectuado cambios en los datos del usuario y los

		datos necesarios del usuario estén completos.
4	El actor pulsa el botón “Guardar” (I).	4.1 El sistema valida los datos y muestra un mensaje de notificación. Culmina el caso de uso.
Flujos alternos		
	Acción del Actor	Respuesta del Sistema
1		1.1 Si no se puede establecer conexión con la Base de Datos, el sistema muestra el siguiente mensaje: “No se puede establecer conexión con los datos.” Culmina el caso de uso.
Sección: “Eliminar usuario”		
	Acción del Actor	Respuesta del Sistema
1		1.1 El sistema verifica que exista conexión con la Base de Datos. 1.2 El sistema muestra la interfaz de gestión de usuarios.
2	2. El actor selecciona la opción “Eliminar” (L) del usuario que desea eliminar dentro de la lista de usuarios del sistema.	2.1 El sistema verifica que el usuario seleccionado no coincida con el que está ejecutando la acción.
3	3. El actor confirma la acción.	3.1 El sistema elimina el usuario. Culmina el caso de uso.
Flujos alternos		
	Acción del Actor	Respuesta del Sistema
		En caso que no se desee realizar la acción se accede al botón Cancelar y no se ejecuta.

Anexo 4: Diagrama de Clase de Análisis

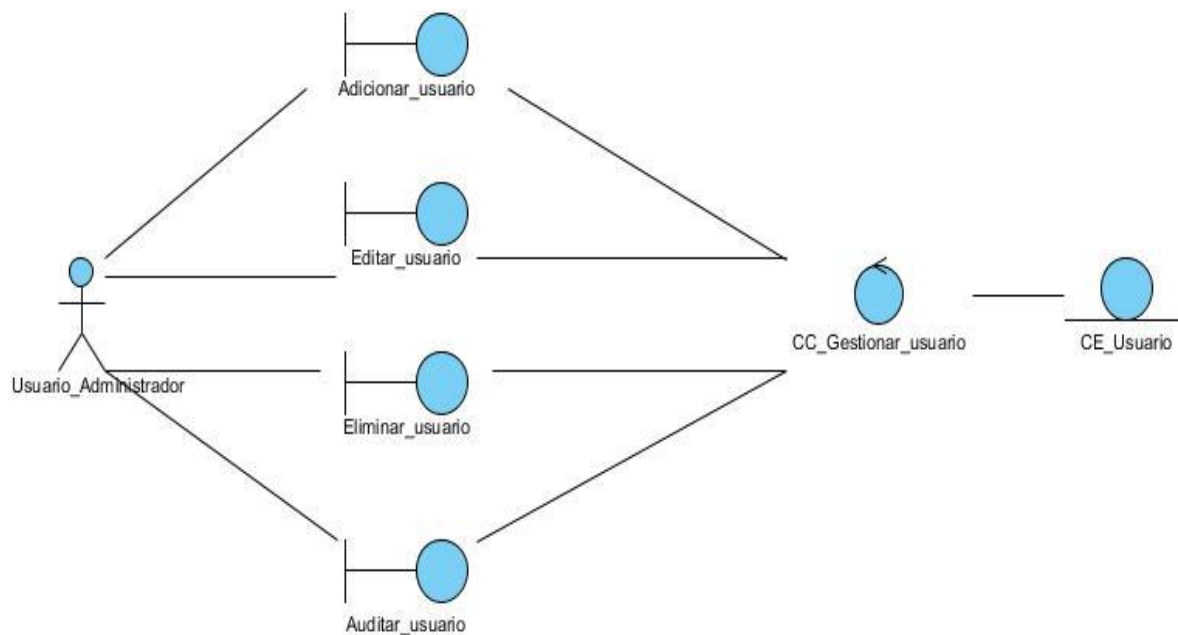


Diagrama de Clases del Análisis Gestionar Usuario

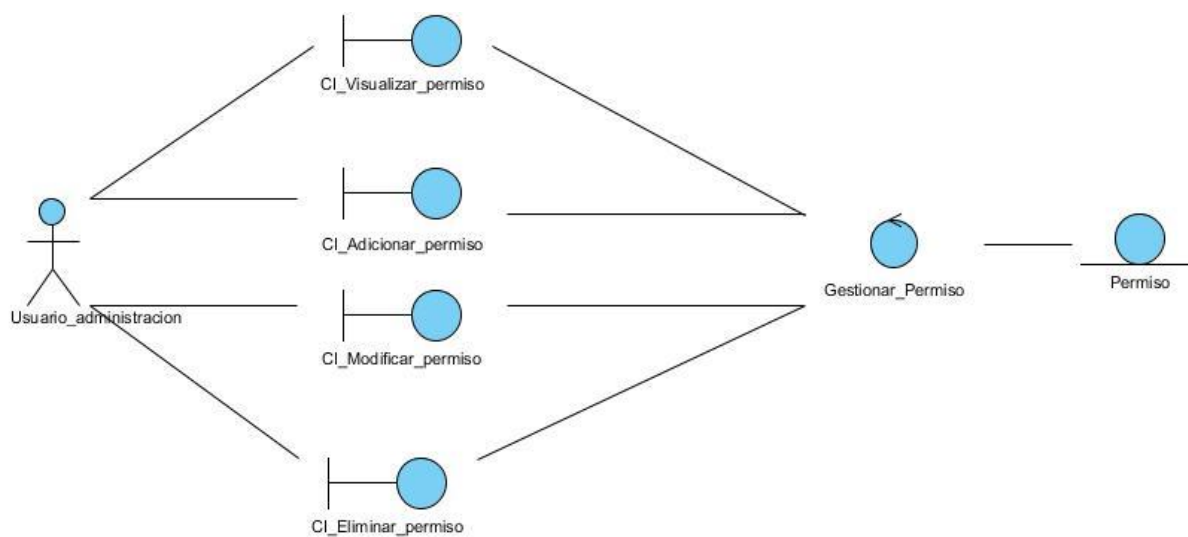


Diagrama de Clases del Análisis Gestionar Permiso

Anexo 5: Diagramas de clases del Diseño

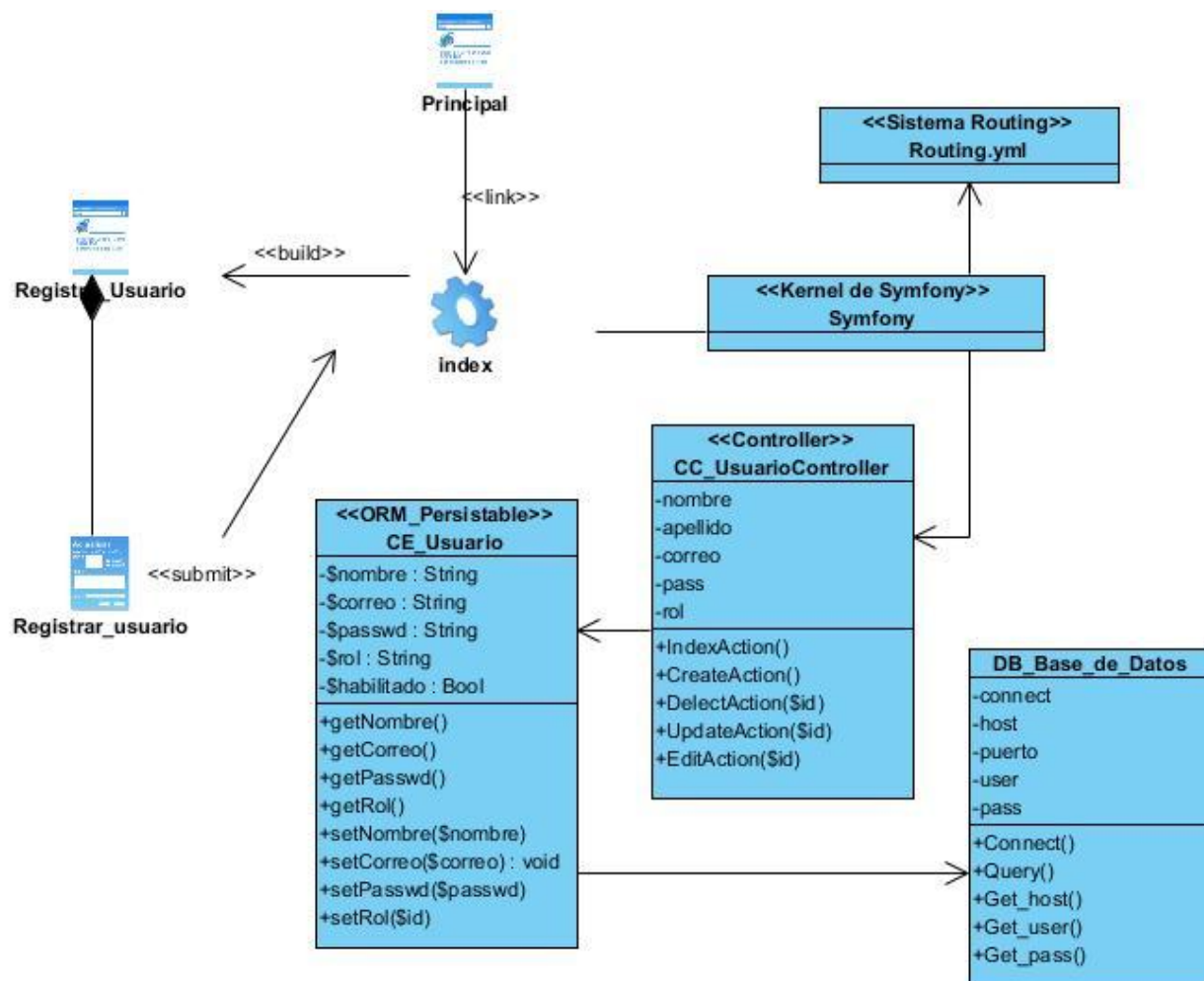


Diagrama de clases de estereotipos web Registrar usuario

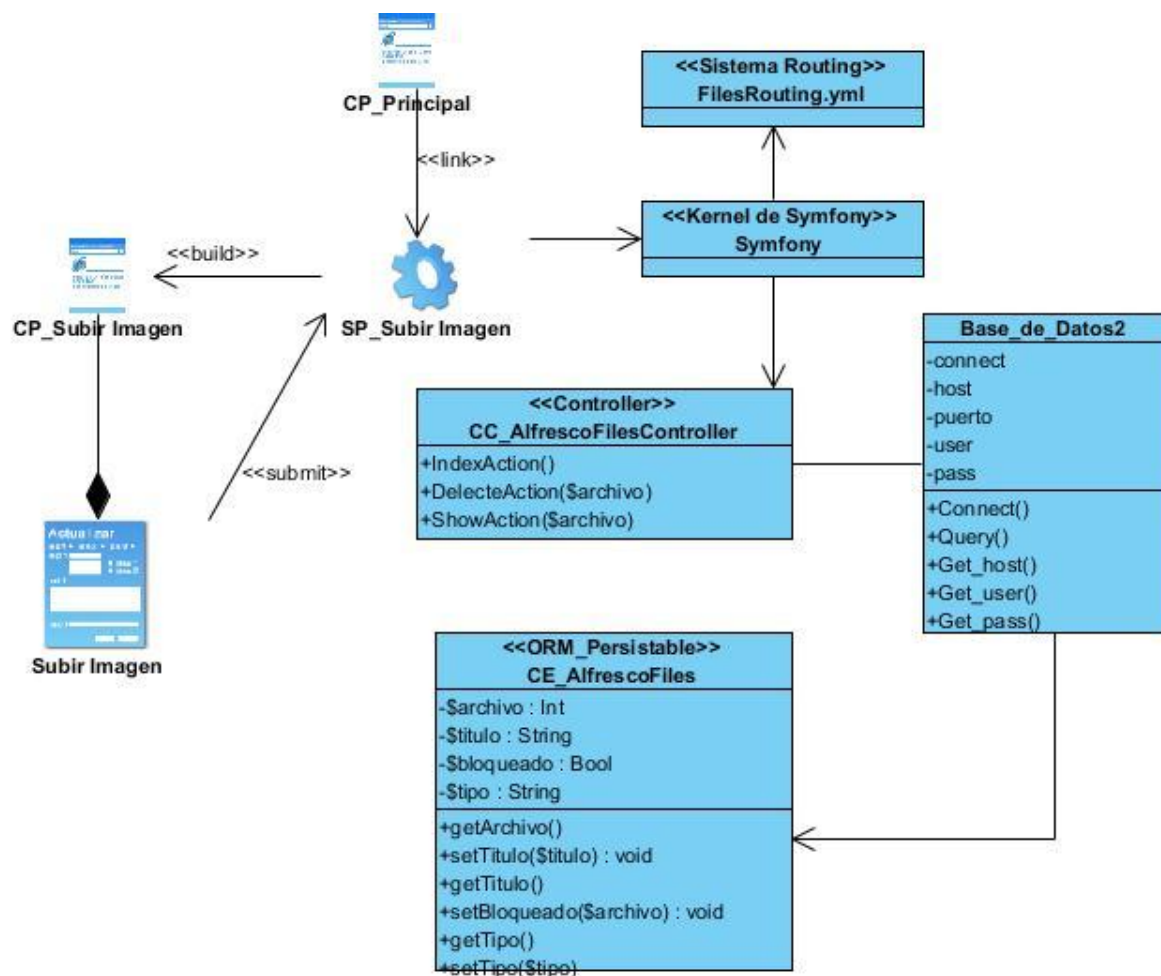


Diagrama de clases de estereotipos web Subir Imagen

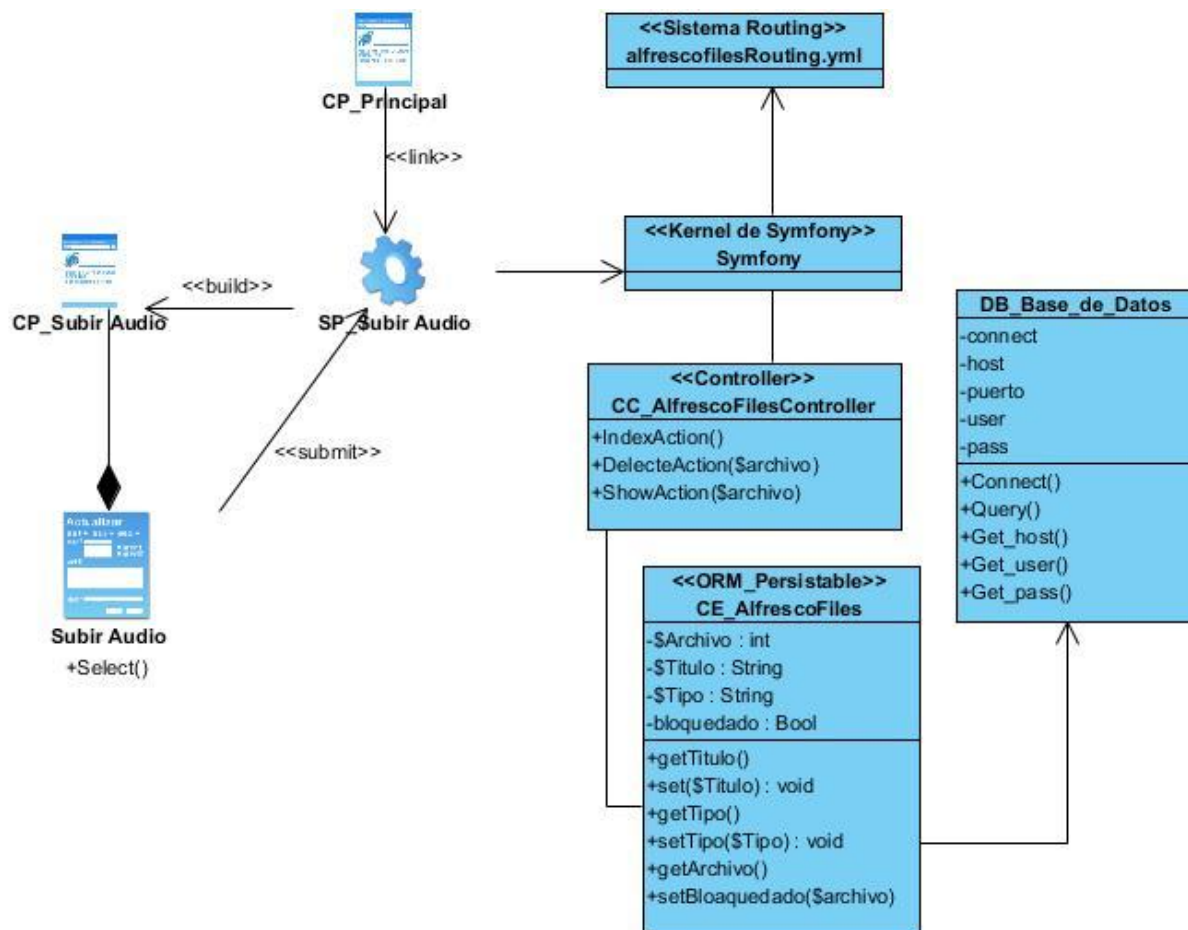


Diagrama de clases de estereotipos web Subir Audio

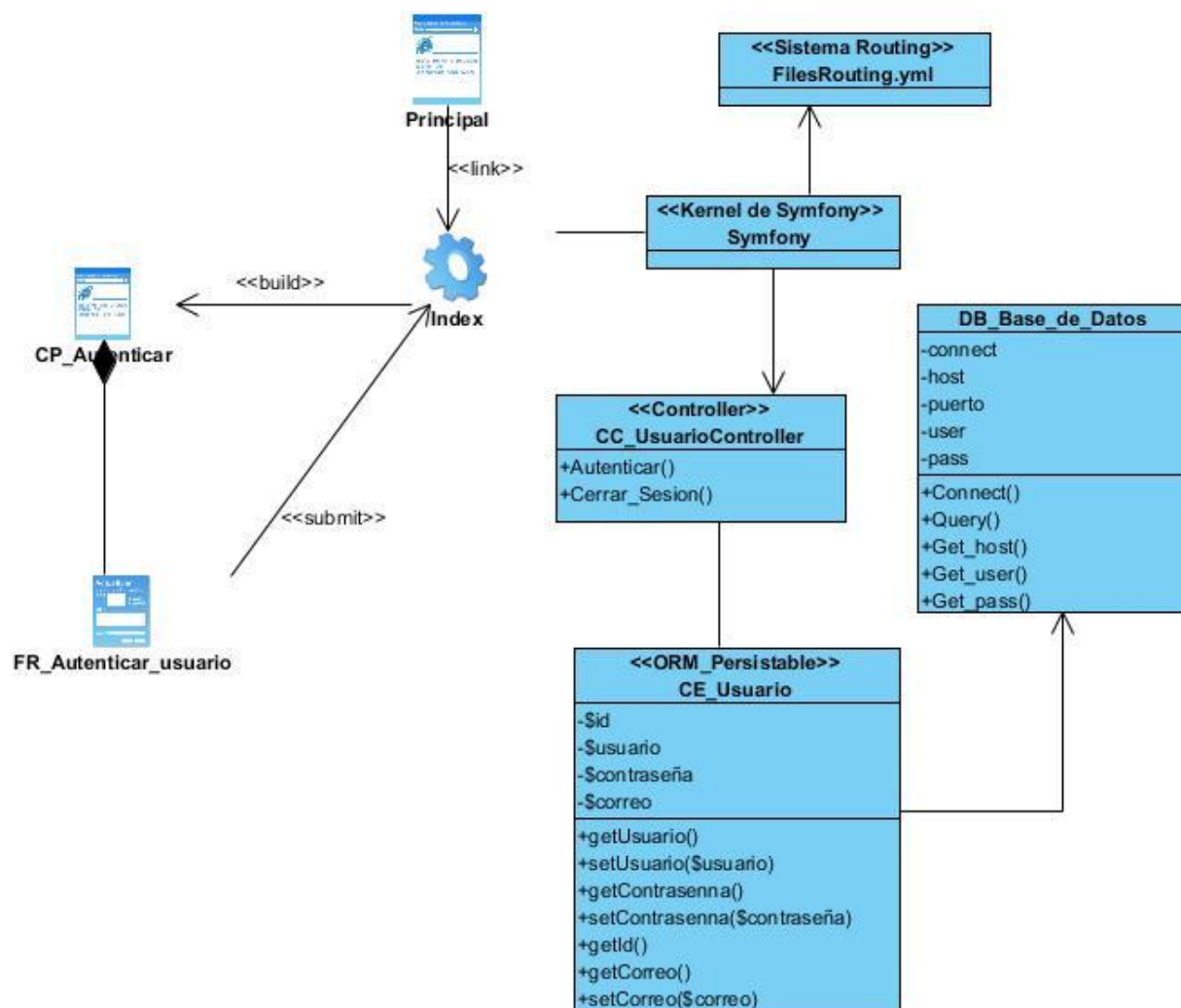


Diagrama de clases de estereotipos web Autenticar Usuario

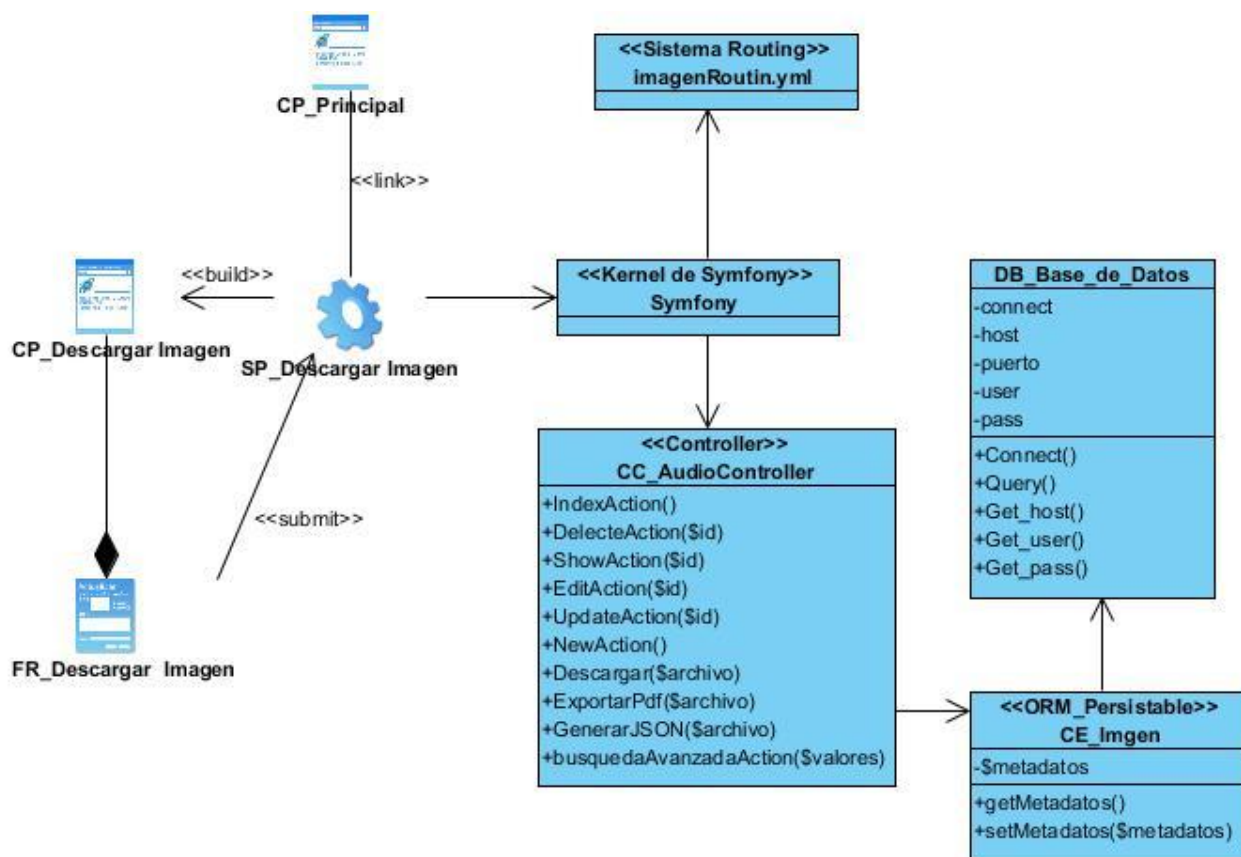


Diagrama de clases de estereotipos web Descargar Imagen

Anexo 6: Diagramas de Interacción

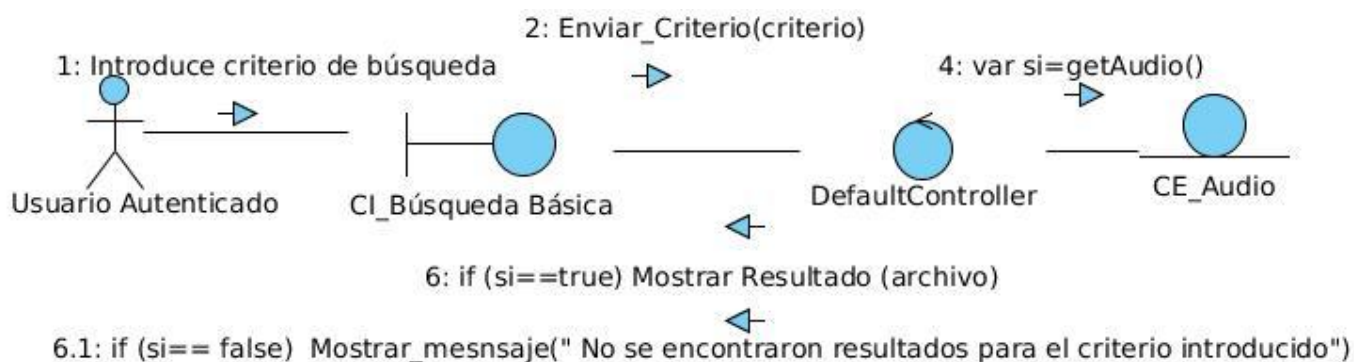


Diagrama de Colaboración de Búsqueda Avanzada



Diagrama de Colaboración Registrar Usuario

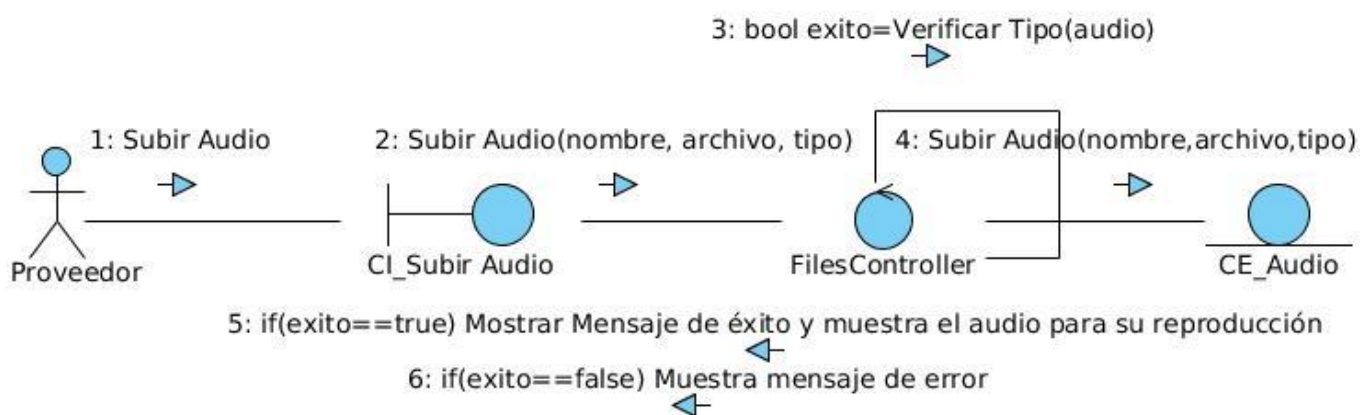


Diagrama de Colaboración Subir Audio

Anexo 7: Pruebas Funcionales

Descripción general: El caso de uso se inicia cuando el usuario introduce en el campo de texto el criterio de búsqueda. Finaliza así el CU.

Condiciones de ejecución: Que el usuario introduzca el criterio de búsqueda.

SC Búsqueda simple de contenido

Escenario	Descripción	Variable 1: nombre	Respuesta del sistema	Flujo central
EC 1.1 Datos correctos	El usuario introduce en el campo de texto el criterio de búsqueda	V	Se muestra el resultado.	El usuario introduce el criterio de búsqueda y presiona el botón "Buscar" o Enter.
EC 1.2 Campos Vacíos	El usuario deja los campos vacíos.	I	El sistema no realiza nada.	El usuario deja algún campo vacío y presiona el botón Aceptar o presiona el botón Enter.
EC 1.3 Datos introducidos incorrectamente	El usuario introduce un criterio de búsqueda.	V	Se muestra un mensaje: "No hay resultados para este criterio"	El usuario introduce el criterio de búsqueda y presiona el botón "Buscar" o Enter.

Descripción general: El CU se inicia cuando un usuario Proveedor desea importar un contenido al sistema, el mismo muestra la interfaz correspondiente con la opción Subir contenido. Finaliza así el CU.

Condiciones de ejecución: Que el Proveedor introduzca los datos que requiere la aplicación para importar el contenido en el sistema.

SC Subir contenido

Escenario	Descripción	Variable 1: nombre	Respuesta del sistema	Flujo central
EC 1.1 Datos correctos	El usuario Proveedor selecciona el fichero y lo añade de manera correcta al sistema.	V	El sistema inserta los datos obtenidos del fichero.	El usuario llena todos los campos de manera correcta y da clic en Subir.
EC 1.2 Campos Vacíos	El usuario Proveedor deja campos sin llenar.	I	El sistema muestra un mensaje que dice: "Por favor, rellene este campo".	El usuario deja algún campo vacío y le da clic en Subir.

Descripción general: El CU se inicia cuando un usuario Proveedor desea exportar la estructura del sistema, el mismo muestra la interfaz correspondiente con la opción Crear Estructura. Finaliza así el CU.

Condiciones de ejecución: Que el administrador selecciona la opción que requiere la aplicación para exportar la estructura en el sistema.

SC Exportar metadatos

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Datos correctos	El usuario selecciona la opción exportar metadatos del contenido.	El sistema exporta los metadatos del contenido seleccionado.	El usuario selecciona la opción exportar metadatos.

Descripción general: El CU se inicia cuando un usuario desea visualizar una imagen en el sistema, el mismo muestra la interfaz correspondiente con la opción Visualizar imagen. Finaliza así el CU.

Condiciones de ejecución: Que el administrador realice las operaciones que requiere la aplicación para mostrar la imagen en el sistema.

SC Visualizar Imagen

Escenario	Descripción	Variable 1: nombre	Respuesta del sistema	Flujo central
EC 1.1 Datos correctos	El usuario selecciona en la sección Imagen la opción mostrar imagen en el sistema.	V	El sistema muestra la imagen de manera correcta.	El usuario da clic en el botón mostrar imagen.

Descripción general: El CU se inicia cuando un administrador desea reproducir (vídeo/audio) en el sistema, el mismo muestra la interfaz correspondiente con la opción Reproducir (vídeo/audio). Finaliza así el CU.

Condiciones de ejecución: Que el administrador realice las operaciones que requiere la aplicación para reproducir el (audio/vídeo) en el sistema.

SC Visualizar Audio/Vídeo

Escenario	Descripción	Variable 1: nombre	Respuesta del sistema	Flujo central
EC 1.1 Datos correctos	El usuario selecciona en la galería (Vídeo/Audio) la opción mostrar (vídeo/audio).	V	El sistema muestra el (vídeo/audio) de manera correcta.	El usuario da clic en el botón mostrar (vídeo/audio).