

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 6



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

**Módulos de gestión, almacenamiento y publicación de archivos
multimedia para VideoWeb 2.0**

AUTORES: Elaine Lorente Rodríguez

Manuel A. González López

TUTOR: Ing. Iván Betancourt Rodríguez

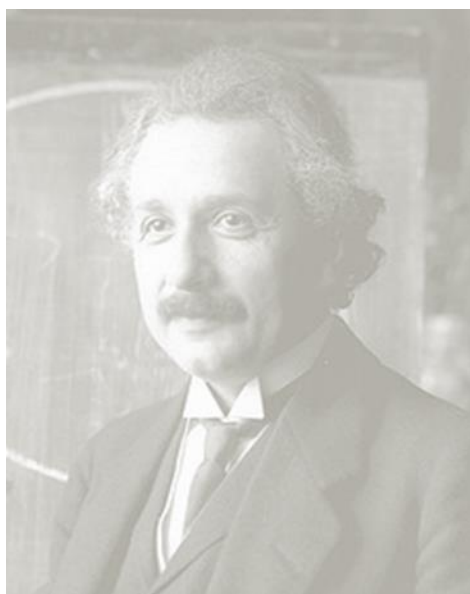
CO-TUTOR: Ing. Vilmavis La Rosa Sordo

11 Junio 2013

“Año 55 de la Revolución”

“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”

Albert Einstein



DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2013.

Autor:

Elaine Lorente Rodríguez

Autor:

Manuel A. González López

Co-Tutor:

Ing. Vilmavis La Rosa Sordo

Tutor:

Ing. Iván Betancourt Rodríguez

Generales del diplomante

Nombre y Apellidos: Elaine Lorente Rodríguez

Sexo: Femenino

Grupo: 6504

Correo electrónico: elorente@estudiantes.uci.cu

Nombre y Apellidos: Manuel Alejandro González López

Sexo: Masculino

Grupo: 6504

Correo electrónico: mglez@estudiantes.uci.cu

Generales del tutor

Nombre y apellidos: Iván Betancourt Rodríguez

Centro de trabajo: Universidad de las Ciencias Informáticas.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2011

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Correo electrónico: ibetancourt@uci.cu

AGRADECIMIENTOS

A mi mamá Lazara y a mi papá Noé por apoyarme en todos los momentos de mi vida. Lo amo mucho.

A mi hermana Elina, que la quiero con la vida, eres mi guía y ejemplo. Te quiero

A mis tías Maira, Teresa y Loyda, a mis primos Alejandro, Suleidy, Diana Liz, Joe Luis, Car Luis, que de una manera u otra me han apoyado mucho. Los quiero.

*A mi abuelita que no se encuentra presente pero sé que estaría muy orgullosa de mi si me estuviera viendo.
Nunca te olvidare.*

A mi sobrino que viene en camino, que también forma parte ya de mi vida.

A mis amigos que me han apoyado y han contribuido a que hoy este aquí entre ellos, Mislaidys, Yarisel, Yaima, Yeilen, Renier J, José A, Yaimy, Alejandro. Los quiero y siempre los recordare. Elaine

Primero que todo a mi familia por estar siempre preocupada y atenta por mí en estos 5 años, a mi mamá y abuela que siempre han esperado este momento. A mi tía Leidis que ha sido como otra madre para mí, así como Lesnier un hermano en todos sentidos. A mi tía Lili que siempre me ha apoyado y aconsejado en mi desarrollo y a Cristian el flaco que espero que siga los pasos nuestros para formarse como profesional. A mi tía Liurkis y Yanelis que aun en la distancia me han apoyado bastante.

A Leonardo, Odalis y la tropa de Holguín que estos últimos años han sido como otra familia para mí gracias por todo.

Muy especial a mi compañera de viaje en esta vida Yaimy que la mayor parte de mi carrera me ha acompañado aún en los momentos más difíciles, gracias por todo el cariño y amor brindado, te quiero mucho.

A todo el piquete del apto en la recta final, que con el paso de los años nos hemos convertido en una familia como hermanos, a Luis Manuel el "político pelotero", Falcón el "cracker del apto", A todo el grupo que se formó desde 1er año que se ha mantenido estrechamente vinculado. A David que ha sido como un hermano para mí durante la carrera

*Al tutor Joán que nos ha apoyado en todo momento y ha estado al tanto de todo nuestro desarrollo, a Uilmavis que del mismo modo ha sido un guía muy importante para nuestro resultado.
Manuel Alejandro*

RESUMEN

La plataforma VideoWeb permite la gestión de contenidos en la web así como la transmisión de archivos multimedia, con la utilización de tecnología *streaming*. Dicha plataforma es una aplicación web capaz de realizar la gestión y publicación de archivos multimedia. Este sistema está inmerso en un proceso de renovación de la versión del CMS utilizado en el desarrollo, este proceso se deriva en una reimplementación de los módulos con que cuenta actualmente el sistema. Este trabajo tiene como objetivo reimplementar los módulos de gestión, publicación y almacenamiento para que sean 100% funcionales con la versión 7.0 de Drupal. Para dar solución a los objetivos planteados, se detallan herramientas y tecnologías a utilizar, se hace un análisis de las soluciones similares en Cuba y el mundo que aportan nuevas ideas a la investigación e influyen sobre la implementación, además se generaron los artefactos que podrán guiar el soporte del producto en futuras actualizaciones de la plataforma.

Palabras clave: almacenamiento, aplicación web, gestión, módulo, publicación, *streaming*.

Índice de Contenido

Introducción	1
CAPÍTULO 1: Fundamentos Teóricos	6
1.1 Principales conceptos asociados al dominio del problema	6
1.1.1 Archivo multimedia	6
1.1.2 Almacenamiento de archivos multimedia.....	7
1.1.3 Gestión de archivos multimedia.....	7
1.1.4 Publicación de archivos Multimedia.....	8
1.2 Situación problemática	8
1.3 Soluciones existentes.....	9
1.3.1 Solución internacional	9
1.3.2 Solución nacional	10
1.4 Herramientas y tecnologías.....	11
1.4.1 Metodología de desarrollo	11
1.4.2 Lenguaje de modelado 2.0	15
1.4.3 Herramienta para modelado 8.0	16
1.4.4 Gestor de Bases de Dato 9.1	17
1.4.5 Lenguaje de programación 5.3.10	17
1.4.6 Entorno de desarrollo integrado (IDE) 7.3	18
1.4.7 Sistemas gestión de contenidos (CMS) Drupal 7.22.....	18
1.5 Conclusiones	18
CAPÍTULO 2: Características de la solución.....	20
2.1 Dominio conceptual.....	20
2.1.1 Descripción del dominio	20
2.1.2 Diagrama de clases del Modelo de Dominio.....	21

ÍNDICE DE CONTENIDO

2.1.3 Descripción de las clases del modelo de dominio.....	21
2.2 Requisitos	22
2.2.1 Requisitos Funcionales.	22
2.2.2 Requisitos no Funcionales	30
2.3 Descripción de los actores	32
2.3.1 Casos de uso del sistema	32
2.3.2 Diagrama de casos de uso.....	33
2.3.3 Descripción de Casos de Uso del Sistema	33
2.4 Conclusiones	42
CAPÍTULO 3: Análisis y Diseño	43
3.1 Modelo de análisis	43
3.1.1 Diagramas de Clases de Análisis	43
3.1.2 Diagramas de Colaboración	44
3.2 Arquitectura.....	45
3.2.1 Patrón Arquitectónico	46
3.2.1.1 Patrón Modelo-Vista-Controlador	46
3.3 Patrones de Diseño.....	47
3.3.1 Patrones GRASP	47
3.3.2 Patrones GOF	48
3.4 Modelo de Diseño	49
3.4.1 Diagramas de clases del Diseño	49
3.4.2 Diagrama de Despliegue	50
3.4.3 Modelo Datos	51
3.5 Conclusiones	53
CAPÍTULO 4: Validación de la solución propuesta.....	54
4.1 Modelo de implementación.....	54
4.1.1 Diagrama de componente	54

ÍNDICE DE CONTENIDO

4.2 Pruebas del sistema.....	55
4.2.1 Métodos de Prueba	55
4.2.2 Pruebas de Caja Negra	55
4.2.3 Aplicación de las pruebas de Caja Negra.	56
4.2.3.1 Caso de prueba para el caso de uso Gestionar archivos multimedia.....	57
4.2.4 Descripción de variable.....	59
4.2.5 Secciones a revisar.....	60
4.3 Resultados de pruebas	62
4.4 Conclusiones	62
Conclusiones generales.....	64
Recomendaciones	65
Referencias bibliográficas	74
Glosario de términos.....	75

Índice de Tablas y Figuras

Figura 1 Disciplina, fases, iteraciones del RUP 13

Figura 2 Diagrama del modelo de dominio..... 21

Figura 3 Diagrama de casos de uso del sistema..... 33

Figura 4 DCA caso de uso. Gestionar archivos multimedia..... 44

Figura 5 DCA caso de uso. Gestionar publicación de archivo multimedia..... 44

Figura 6 DC caso de uso. Gestionar archivos multimedia: sección adicionar archivo multimedia..... 44

Figura 7 DC caso de uso. Gestionar archivos multimedia: sección editar archivo multimedia..... 45

Figura 8 DC caso de uso. Gestionar archivos multimedia: sección eliminar archivo multimedia..... 45

Figura 9 Patrón Modelo-Vista-Controlador..... 46

Figura 10 DCD Gestionar archivos multimedia..... 50

Figura 11 DCD Gestionar publicación de archivo multimedia..... 50

Figura 12 Diagrama de despliegue..... 51

Figura 13 Diagrama Entidad-Relación..... 52

Figura 14 Diagramas de componente del módulo archivo multimedia..... 55

Tabla 1 Actores del sistema..... 32

Tabla 2 Descripción Caso de Uso Gestionar archivos multimedia..... 34

Tabla 3 Descripción Caso de Uso Gestionar publicación de archivos multimedia..... 38

Tabla 4 Casos de pruebas. CU Gestionar archivos multimedia..... 57

Tabla 5 Descripción de las variables para el CU Gestionar archivos multimedia..... 59

Tabla 6 SC 1 Adicionar AM..... 60

Tabla 7 SC 2 Eliminar AM..... 60

Tabla 8 SC 3 Editar AM 61

Introducción

La comunicación humana surgió en el momento en que las primeras generaciones iniciaron su lucha por la supervivencia y en respuesta a sus instintos se vieron obligados a transmitir a quienes les rodeaban, sus impresiones, sentimientos y emociones (1). Es por ello que la necesidad de vías para establecer comunicación, dió paso a una evolución que hasta la actualidad continúa en desarrollo. Este desarrollo fue bastante influenciado con el surgimiento de la revolución electrónica iniciada en la década de los 70 que constituye el punto de partida para el progreso creciente de la era Digital. Gracias a las investigaciones desarrolladas se ha logrado la convergencia de la informática y las telecomunicaciones, posibilitando la interconexión entre redes.

El surgimiento de esta revolución tecnológica ha abarcado casi todas las esferas sociales, con el objetivo de satisfacer los gustos y necesidades de las personas. Entre los avances fundamentales se encuentra la web que fue creada en 1989 por Tim Berners-Lee en el CERN¹ (Ginebra). Su objetivo inicial era compartir información entre grandes grupos de investigación localizados en diversos lugares del mundo. La web facilita el acceso a los distintos recursos que ofrece Internet a través de una interfaz común basada en el hipertexto. Haciéndose enormemente popular con la aparición de programas dotados de interfaces gráficas amigables que permiten a cualquier persona acceder a los recursos de la red, conocidos como navegadores.

La reproducción de archivos de audio y video en Internet es un proceso complejo, que requiere una óptima tecnología de transmisión que pueda dar respuesta a todas las peticiones de los usuarios eficientemente. Hasta el año 1995 la reproducción era un proceso lento que requería la descarga del archivo multimedia, fue entonces cuando surgió la tecnología *streaming* con el lanzamiento de RealAudio. Este último es un formato de audio comprimido creado por la compañía *RealNetworks*².

Cuba ha puesto todo su empeño en el desarrollo de tecnologías y las telecomunicaciones, dedicando parte de su desarrollo en el campo de la gestión, almacenamiento y publicación de los archivos multimedia a través de la red. Decidiendo formar escuelas como la CUJAE (Instituto Superior Politécnico José Antonio Echeverría), politécnicos entre otras, que potencien el desarrollo de la informática y las comunicaciones. Es por ello que en el 2002 se crea la Universidad de las Ciencias

¹ La Organización Europea para la Investigación Nuclear (nombre oficial), comúnmente conocida por la sigla CERN.

² *RealNetworks* es un proveedor de software para Internet y servicios. <http://www.realnetworks.com>

Informáticas (UCI), centro de estudios universitarios donde ingresan estudiantes de todo el país que vinculan los procesos Investigación más Desarrollo. Siendo una universidad que cuya misión es producir software y servicios informáticos, además tiene como propósito preparar ingenieros informáticos integrales y a su vez generar soluciones informáticas para Cuba y otros países del mundo. También respalda la Misión Médica Cubana en Venezuela garantizando el adecuado funcionamiento de los sistemas informáticos (2). Uno de estos centros de desarrollo con los que cuenta, es Geoinformática y Señales Digitales (GEYSED) de la Facultad 6, el cual cuenta con el proyecto de Catalogación y Publicación de Medias. Dicho proyecto tiene entre sus productos la plataforma VideoWeb, que está desarrollada con el Sistema Gestor de Contenido o *Content Management System* (CMS) Drupal 6. Esta permite la gestión de contenidos y la transmisión de archivos multimedia por la red, con la utilización de la tecnología *streaming*. Este producto se ve constantemente sometido a mejoras e incorporación de nuevos servicios de acuerdo a las necesidades de los clientes y basado principalmente, en el desarrollo de nuevas tecnologías, lo cual evidencia la necesidad de incorporar nuevas funcionalidades. El proceso de mejoras de la plataforma también se ve influenciado debido a que Drupal ha evolucionado, en estos momentos la última versión estable es la 7.22, y la comunidad de desarrollo sigue aportando y debatiendo las nuevas funciones y tecnologías a usar. De este modo las versiones anteriores van quedando obsoletas sin soporte ni actualizaciones de módulos³ y herramientas.

Un CMS debe permitir la gestión y administración de los contenidos que se desean distribuir a través de la red (3). Si se mantiene la plataforma VideoWeb con la versión actual de este CMS (6.0) estaría en desventaja con el resto de las aplicaciones similares en el mundo, y se desaprovecharían facilidades y mejoras técnicas que proporcionan las versiones superiores. Es por ello que se hace presente la necesidad de migrar a la versión 7 de Drupal que garantizaría estabilidad ante cualquier vulnerabilidad o fallo de seguridad.

El cambio que viene dado de una versión de Drupal a otra es drástico, debido a que en la versión 6 se hace más molesto y complicado el proceso de acceder a la información almacenada en base de datos, sin embargo en Drupal 7 las consultas y el acceso a datos es mucho menos complejo. Este progreso viene dado por el uso de bibliotecas que facilita el acceso a los datos.

Drupal almacena los contenidos de una página web en nodos. Estos constituyen por tanto los

³ Un módulo consiste en un conjunto de archivos con código PHP, que utiliza la arquitectura y las APIs de Drupal para incorporar nuevas características funcionales al sitio web.

elementos básicos de información en un sitio web hecho con este CMS. En cuanto a los avances de la versión 7, el cambio más importante radica en el hecho de que los nodos son ahora considerados entidades. Una entidad es un elemento genérico, al que se le pueden añadir campos. Los nodos, usuarios, taxonomía, archivos y otros elementos son definidos como entidades, de los cuales se gestionan los campos asociados y su presentación. Producto de los nuevos cambios de una versión a otra es que el proceso de desarrollo de los nuevos módulos requiere un rediseño de la lógica orientado los cambios en su arquitectura para así lograr un sistema más personalizable y menos abstracto. (3)

A partir de la situación problemática planteada anteriormente se define como **problema a resolver**: ¿Cómo facilitar los procesos administrativos de gestión, publicación y almacenamiento de archivos multimedia en la plataforma VideoWeb 2.0?

Por tal motivo queda definido como **objeto de estudio**: el proceso de gestión, publicación y almacenamiento de contenidos en la web. Enmarcado en el **campo de acción**: los procesos de gestión, publicación y almacenamiento de archivos multimedia en la plataforma VideoWeb 2.0.

Se define como **objetivo general**: desarrollar los módulos de gestión, almacenamiento y publicación de archivos multimedia para la plataforma VideoWeb 2.0.

Se presenta como **idea a defender** que el desarrollo de los módulos de gestión, publicación y almacenamiento de archivos multimedia en la plataforma de VideoWeb 2.0 facilitará los procesos administrativos de gestión.

Para dar cumplimiento al objetivo trazado se definen las siguientes **tareas de la investigación**:

- 1– Caracterizar los procesos relacionados con la gestión, almacenamiento y publicación de archivos multimedia en plataformas web.
- 2– Caracterizar en el ámbito nacional e internacional plataformas web que realicen gestión y publicación de archivos multimedia.
- 3– Caracterizar las herramientas y tecnologías a utilizar en la construcción de la solución.
- 4– Realizar el análisis y diseño de los módulos a implementar.
- 5– Implementar las funcionalidades relacionadas con la gestión, almacenamiento y publicación de archivos multimedia para la plataforma VideoWeb.
- 6– Diseñar y aplicar pruebas al sistema.

Para el desarrollo completo del trabajo y su total entendimiento se hizo necesario emplear los

métodos de investigación científica tales como los teóricos y empíricos los cuales se describen a continuación:

Métodos Teóricos (4):

Analítico-Sintético: mediante este método se analizaron los principales conceptos y estándares relacionados con el proceso de publicación de archivos multimedia en plataformas web, con el objetivo de sentar las bases teóricas de la investigación, así como herramientas y tecnologías para el desarrollo de plataformas.

Histórico-Lógico: se utilizó en el análisis de la base teórica, el estudio de las tecnologías a emplear y la descripción de los procesos relacionados con la gestión, publicación y almacenamiento de archivos multimedia para plataformas web.

Inductivo-Deductivo: el proceso de inducción permitió generalizar las características presentes en la plataforma VideoWeb 1.0 utilizadas, para deducir conclusiones asociadas a las necesidades de la investigación.

Modelación: se empleó este método en la elaboración de modelos para obtener una visualización simplificada de los procesos. Se hace visible en el trabajo al modelar la representación del patrón arquitectónico Modelo-Vista-Controlador.

Métodos Empíricos (4):

Observación: permitió un control sistemático durante todo el proceso de la investigación, para corregir los errores que se presenten. Además de dar cumplimiento a las tareas trazadas en el cronograma permitiendo un control sobre las mismas.

Para un mejor entendimiento del trabajo de diploma se pone a continuación un resumen de lo que se abordará en cada uno de los capítulos:

Capítulo 1: en este capítulo se hace referencia a los principales conceptos asociados al dominio del problema. Además se realiza un análisis de las soluciones que utilicen los procesos de almacenamiento, gestión y publicación de archivos multimedia en el ámbito nacional e internacional. También se analiza la metodología de desarrollo de software y las herramientas que se van a utilizar para dar soporte al desarrollo de la aplicación, haciendo énfasis en las ventajas y funcionalidades que brindan las mismas.

Capítulo 2: en el presente capítulo se realiza el diagrama de Modelo del Domino, lo cual ayuda mucho definir los procesos y roles más trascendentales para la gestión de la plataforma. También se especifican los requisitos funcionales y no funcionales con los que debe cumplir el sistema para su

correcto funcionamiento y se describen los casos de uso del sistema con sus flujos.

Capítulo 3: en el capítulo 3 se especifican los diagramas significativos de la aplicación como son los de Análisis y Diseño, junto al Modelo de datos y el diagrama de Despliegue. Además se describen los patrones arquitectónicos utilizados en la construcción del software.

Capítulo 4: en el capítulo 4 se refleja el flujo de trabajo de implementación y pruebas. Estas pruebas se utilizan para verificar cuales son las funcionalidades que debe de cumplir el sistema, de manera que quede libre de no conformidades. También se modela el diagrama de Componentes y describen los Casos de pruebas.

CAPÍTULO 1: Fundamentos Teóricos

En el presente capítulo se hace referencia a los principales conceptos asociados al dominio del problema, permitiendo un mayor entendimiento sobre el tema a tratar. También se hace un acercamiento general a las funciones de otras soluciones existentes nacionales e internacionales. Se define la metodología de desarrollo de software y las características fundamentales que posibilitaron su elección. A su vez, se analizaron las herramientas y tecnologías que serán empleadas en la construcción de la solución propuesta.

1.1 Principales conceptos asociados al dominio del problema

Para dar solución al problema planteado es necesario analizar algunos conceptos asociados al dominio del problema, permitiendo una mejor comprensión. Entre los procesos claves, están los de almacenamiento, gestión y publicación de archivos multimedia que son la guía fundamental para dar solución al problema planteado anteriormente.

1.1.1 Archivo multimedia

Un archivo es un espacio que se reserva en el dispositivo de memoria de un computador para almacenar porciones de información que tienen la misma estructura y que pueden manejarse mediante una instrucción única. (5)

Archivo es un conjunto de información que se almacena en una computadora y puede ser identificado por su ruta completa. Los archivos informáticos se llaman así porque son los equivalentes digitales de los archivos en tarjetas, papel o microfichas del entorno de oficina tradicional. Los archivos informáticos facilitan una manera de organizar los recursos usados para almacenar permanentemente información dentro de una computadora. (6)

El vocablo multimedia refiere a que utiliza conjunta y simultáneamente diversos medios, como imágenes, sonidos y texto, en la transmisión de una información. (5)

Multimedia es cualquier combinación de texto, arte gráfico, sonido, animación y video que llega al usuario por computadora u otros medios electrónicos. Es un centro de información tan poderoso, expresivo y natural que logra que se capte en forma mucho más efectiva la información que se recibe, estimulando increíblemente los sentidos, haciendo que el usuario esté mucho más alerta y receptivo. Todo esto es porque permite interactuar con los sonidos, las imágenes, los colores y la acción. (7)

Tomando en cuenta las definiciones dadas por los autores anteriores se puede concluir que un

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

archivo multimedia es un segmento de información almacenado en un dispositivo estructuradamente, a través de diferentes medios como imágenes, audio o video entre otros.

1.1.2 Almacenamiento de archivos multimedia

El almacenamiento, es la acción y efecto de almacenar (reunir o guardar cosas, registrar información). (5)

El proceso de almacenar consiste en guardar algún objeto o elemento específico con el fin de poder recurrir a él en el caso que sea necesario, por lo que será conveniente por razones de eficacia establecer una vía de menos longitud para el acceso a los materiales de alta rotación. En este sentido el proceso de almacenar archivos multimedia se realiza minimizando la circulación interna. (8)

Un mecanismo similar a lo planteado anteriormente es utilizado por YouTube⁴ con relación a los videos más vistos, es precisamente en la arquitectura que guarda en memoria de alta respuesta (*caché*⁵), los datos del sistema que con más frecuencia se solicitan o se muestran. Cuando se detecta que un sector comienza a ser menos requerido por los usuarios u otras partes del sistema, los datos se descargan de la memoria, y dejan paso a otras funciones (9). Básicamente este proceso consiste en gestionar de forma óptima la acción de almacenar un archivo de multimedia para en cualquier momento acceder al mismo garantizando el rendimiento del servidor.

1.1.3 Gestión de archivos multimedia

En internet existen muchísimos sitios especializados en la gestión de videos, donde de forma sencilla se puede entrar y ver los videos publicados, los cuales se encuentran agrupados por temas, ya sean deporte, cultura, educación entre otros. Muchos de estos sitios permiten que los propios usuarios puedan subir sus propios videos. La única razón por la que no se pueden subir videos a estos portales es que no hayan sido creados por uno, los mismos no están autorizados a subirse si no es con el derecho autor, además tampoco se autorizan a subir videos que traten temas delicados, ya sean de política, racismo u otros. (10)

Al subir videos en sitios como YouTube, sus sistemas se encargan de procesar cada video, para ofrecer una mejor experiencia en la web. Los usuarios podrán elegir si lo desean ver en HD o en resoluciones más sencillas. También detectan la conexión de internet y el tipo de dispositivo del usuario, con el objetivo de reproducir el video en la calidad que es mejor para el espectador. (10)

⁴ Sitio web en el cual los usuarios pueden subir y compartir vídeos.

⁵ La caché es una memoria pequeña y rápida usada por la unidad central de procesamiento (CPU), la cual almacena copias de datos ubicados en la memoria principal que se utilizan con más frecuencia.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

Actualmente existen muchas soluciones informáticas que implementan una correcta gestión de los contenidos multimedia. Del mismo modo con el surgimiento de la web 2.0 y el concepto de alojamiento en la nube, se han creado proyectos que realizan una gestión automatizada de los contenidos que el usuario sube a la nube o Internet para ser almacenado y difundido. Mejorando así la seguridad y la accesibilidad desde cualquier plataforma web.

1.1.4 Publicación de archivos Multimedia

El proceso de publicar consiste en: Acción de hacer pública una información o una noticia, dándola a conocer a muchas personas (5). El proceso de publicación de medias en la web consiste en, una vez que el video o material audiovisual a publicar se encuentra codificado o comprimido para ser distribuido sobre una red, con un determinado ancho de banda, se sube dicho video al servidor y es publicado. (11)

La publicación de archivos multimedia en la plataforma VideoWeb consiste en publicar los materiales audiovisuales en la web de un modo sencillo usando las ventajas de Drupal. En este caso se utilizan las vistas de Drupal para mostrar los tipos de contenido “publicación”. Al publicar un archivo multimedia se debe tener en cuenta los fotogramas clave para representar el material audiovisual así como los vínculos necesarios para reproducir o descargar el material.

1.2 Situación problemática

En el Centro GEYSED de la Facultad 6 de la UCI existe el proyecto de Catalogación y Publicación de Medias, el cual tiene entre sus productos la plataforma VideoWeb desarrollada con el CMS Drupal 6. Continuamente este producto está en desarrollo y brinda soporte a los clientes, es por ello que se debe tener en cuenta que el mismo va evolucionando a nuevas versiones. En estos momentos la última versión estable es la 7.22, y a su vez la comunidad de desarrollo va detrás de esta, aportando y debatiendo las nuevas funciones y tecnologías a usar, de este modo las pasadas se van quedando atrás sin soporte y actualizaciones de módulos y herramientas.

Por este motivo fundamental si se mantiene la plataforma VideoWeb con esta versión, este producto estaría en desventaja con el resto de las aplicaciones similares en el mundo, ya que esta versión pronto dejará de tener soporte de la comunidad de desarrollo y se desaprovecharían facilidades y mejoras técnicas que proporcionan las versiones superiores. El cambio que viene dado de una versión a otra es drástico debido a que Drupal 6 no incluye capa de abstracción de la base de datos, por lo que se hace tedioso y complicado acceder a la información almacenada, Drupal 7 es más sencillo en ese aspecto, las consultas y el acceso a datos es menos complejo y mejora

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

considerablemente el rendimiento, esto se realiza utilizando la biblioteca PDO⁶ de PHP⁷ para comunicarse con la base de datos.

Un aspecto muy importante en cuanto a los avances de la versión 7 es respecto a los nodos, el cambio más importante radica en el hecho de que los nodos son ahora considerados entidades. Una entidad es un elemento genérico, al que se le pueden añadir campos. De este modo los tipos de contenido (o nodos), los usuarios, los términos de taxonomía y otros elementos son definidos como entidades, de los cuales se puede gestionar sus campos asociados así como su presentación.

Producto de los nuevos cambios de una versión a otra es que el proceso de desarrollo de los nuevos módulos requiere un rediseño de la lógica enfocado a los cambios en su arquitectura. Esto se debe principalmente a que el código que constituye el núcleo de Drupal está formado por un conjunto de bibliotecas que permiten gestionar los procesos de arranque del sistema. Estas bibliotecas ofrecen además un conjunto de servicios que permiten integrar las funcionalidades adicionales de los módulos, servicios como conexión y administración de la base de datos, gestión de procesos de correo, tratamiento de imágenes, internacionalización, soporte para la codificación y un potente entorno de integración de utilidades. A su vez en esta nueva versión se realizan cambios en muchas funciones *hook*⁸, basados principalmente en hacer un sistema menos abstracto y más personalizable al dividir las funciones, las cuales tenían múltiples propósitos, en funciones independientes.

1.3 Soluciones existentes

En la actualidad existen numerosos sitios web dedicados a la publicación y difusión de videos en Internet, cada uno en particular tiene su propio sistema de gestión, almacenamiento y publicación de archivos multimedia. Teniendo en cuenta el objetivo de este trabajo, se hará un análisis de las soluciones existentes en la actualidad más prestigiosas, tanto nacionales como internacionales.

1.3.1 Solución internacional

YouTube es una plataforma de publicación de videos *online* fundada en febrero de 2005 que permite a los usuarios de todo el mundo subir, descargar videos y visualizarlos. Cuando un usuario sube un video en YouTube le agrega al mismo su título, descripción o resumen, calidad visual y duración. Estos datos son publicados en dependencia de la categoría a la que pertenecen (autos, vehículos,

⁶ PHP Data Objects (o PDO) es una extensión que provee una capa de abstracción de acceso a datos para PHP 5.

⁷ PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

⁸ Funciones de comunicación entre los módulos y el núcleo de Drupal.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

comedia, educación, entretenimiento, deporte, noticias, etc.), y pueden ser visualizados en la interfaz del sitio mediante un reproductor *flash* embebido en el navegador. (11)

La arquitectura de la plataforma está conformada en su totalidad por herramientas de software libre, utilizando Apache como servidor web para la distribución de las páginas. *Lighttpd* como servidor para distribuir los videos sobre protocolo HTTP, el que solamente funciona bien si tiene una constante carga de peticiones. Como sistema operativo de los servidores utiliza la distribución *SuSe* de Linux, como lenguaje de programación utiliza *Python*, el que permite un desarrollo y despliegue rápido. Como gestor de base de datos utiliza *MySQL*, la misma se encuentra particionada en fragmentos con los usuarios distribuidos entre estas diferentes fracciones. Además utiliza una Red de Distribución de Contenidos (RDC) para los videos más populares, esta red replica el contenido en múltiples lugares lo que aumenta la posibilidad de que el contenido se encuentre más cerca del usuario. (11)

Otro dato importante es que la arquitectura del sistema guarda en memoria de alta respuesta (cache) los datos que con más frecuencia se solicitan o se muestran. Cuando se detecta que un sector comienza a ser menos requerido por los usuarios u otras partes del sistema, los datos se descargan de la memoria, y dejan paso a otras funciones. (9)

1.3.2 Solución nacional

Inter-nos es una plataforma para la publicación y transmisión de audio y video por la web utilizando la tecnología *streaming*. Fue creada en la UCI usando como lenguaje de programación ASP, como servidor web Microsoft Internet Information Services (IIS), gestor de base de datos Microsoft SQL Server 2000 y como servidor *streaming* Windows Media Server. Para la reproducción de audio y video en demanda, como emisoras de radio y canales de televisión en vivo, el cliente utiliza como reproductor el que le sea más factible para consumir la transmisión.

Su principal objetivo es apoyar el proceso docente educativo en la universidad, publicando tele clases, materiales educativos todo con el fin de contribuir a una mejor formación del estudiante. Del mismo modo apoya la cultura y recreación, publicando películas, series televisivas, documentales, programas de radio, festivales de artistas aficionados, además de la transmisión de radio y televisión en vivo.

Esta plataforma almacena información asociada a los materiales publicados pero no proporciona a los usuarios servicio de búsqueda de materiales, lo cual representa una desventaja. Las modificaciones al sitio se realizan directamente al código, por lo que suele ser un proceso complicado, además no cuenta con un sistema de gestión de contenidos. Los procesos de publicación, gestión de materiales y noticias resultan un poco engorrosos, estos son realizados por personal de soporte técnico

calificado ya que la plataforma no cuenta con un sistema de gestión de contenidos que facilite estos procesos, los cuales en algunos casos se deben realizar de forma manual, copiando los archivos de video a publicar para una carpeta destinada para este fin en el servidor e interactuando directamente con el código de las páginas web en las que se desea mostrar alguna información. (11)

Luego de analizar las soluciones existentes que utilizan procesos de publicación de medias en la web se llega a la conclusión de que una solución de este tipo, que permita publicar, administrar materiales y distribuirlos en un entorno web es muy útil. La plataforma analizada en el ámbito internacional permite la visualización de contenidos ya publicados y admiten a los usuarios subir y publicar sus propios videos, pero estos no muestran su código, haciéndolo privado. En el caso de Inter-nos, los procesos de gestión de contenidos no están totalmente automatizados, además está desarrollado utilizando tecnologías y herramientas propietarias, lo cual dificulta su uso y comercialización. Es por esto que surge la necesidad de desarrollar una plataforma, que incluya los aspectos positivos y corrija los aspectos negativos de las soluciones analizadas, basada en software libre, que pueda ser personalizada y comercializada en dependencia de los gustos y necesidades de los clientes.

1.4 Herramientas y tecnologías

Después de realizado el análisis de los conceptos asociados al tema y de las soluciones existentes nacionales e internacionales, se caracterizan las herramientas y tecnologías que se van a utilizar para dar soporte a lo planteado anteriormente. Comenzando con la metodología de desarrollo de software.

1.4.1 Metodología de desarrollo

Las Metodologías del Desarrollo de Software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto de software.

El equipo de desarrollo de la plataforma VideoWeb tiene establecido en la arquitectura del proyecto a RUP⁹ como metodología de desarrollo de software a utilizar. Además el equipo cuenta con experiencia y conocimiento en esta metodología. Al mismo tiempo la documentación y los diagramas generados por RUP permiten almacenar toda la base de conocimientos necesaria para continuar con el desarrollo en futuro aun cuando sea otro equipo de trabajo.

1.4.1.1 RUP

El proceso de desarrollo RUP aplica varias de las mejores prácticas en el desarrollo moderno de

⁹ *Rational Unified Process*

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

software en una forma que se adapta a un amplio rango de proyectos y organizaciones. Provee a cada miembro del equipo, un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas. Esta metodología permite que todos los integrantes de un equipo de trabajo, conozcan y compartan el proceso de desarrollo, una base de conocimientos y los distintos modelos de cómo desarrollar el software utilizando un lenguaje modelado común: Lenguaje de Modelado Unificado (UML).

RUP es un proceso de desarrollo de software que provee un enfoque estructurado para realizar tareas y responsabilidades en una organización de desarrollo. Su principal objetivo es asegurar la producción de software de alta calidad, que cumpla las necesidades de sus usuarios finales, que sea realizado en las fechas acordadas y con el presupuesto disponible.

Incorpora muchas de las conocidas como “buenas prácticas” en el desarrollo de software moderno, las cuales se deben tener presentes en el desarrollo de aplicaciones empresariales para garantizar el éxito del proyecto, tales como: Desarrollo iterativo, Gestión de Requerimientos, Arquitectura basada en componentes, Modelo Visual, Verificación de la calidad en forma continua y control de cambios.
(12)

Como RUP es un proceso, en su modelación define como sus principales elementos (12):

- Trabajadores (“quién”): define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- Actividades (“cómo”): es una tarea que tiene un propósito claro, siendo realizada por un trabajador para manipular elementos.
- Artefactos (“qué”): productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- Flujo de actividades (“cuándo”): secuencia de actividades realizadas por trabajadores y que producen un resultado de valor observable.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales también llamados disciplinas, las 6 primeras son conocidas como flujos de ingeniería y los tres últimos como flujos de apoyo o soporte. A continuación se muestra la imagen que contiene las iteraciones, fases y las disciplinas de RUP.

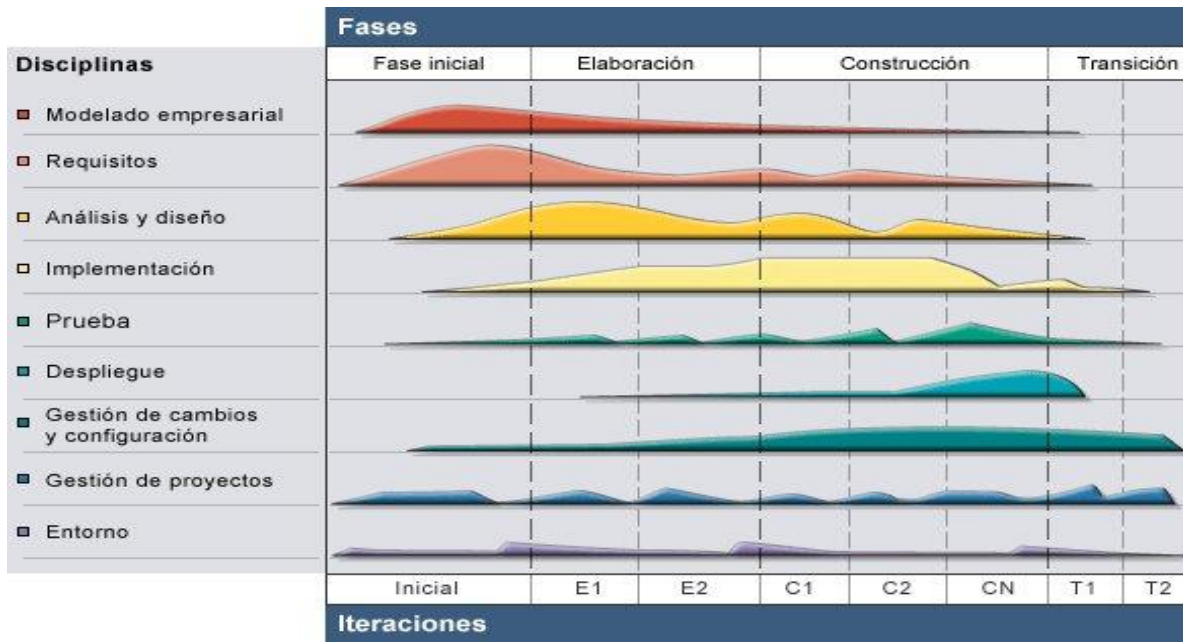


Figura 1 Disciplina, fases, iteraciones del RUP¹⁰

Disciplinas de RUP (12):

- Modelo del Negocio: describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- Requerimiento: define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- Análisis y Diseño: describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- Implementación: define cómo se organizan las clases, objetos en componentes, cuales nodos se utilizarán, la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- Prueba (Testeo): busca los defectos a lo largo del ciclo de vida del software.
- Despliegue: produce la liberación de una versión del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

¹⁰ Obtenido de la ayuda de RUP.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

- Gestión de proyecto: involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- Gestión de configuración y cambios: describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- Entorno: contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementarlos en una organización.

El proceso de desarrollo está dividido en fases, cada una de las cuales tiene objetivos específicos y un conjunto de artefactos definidos que deben alcanzarse. La duración de cada fase depende de las habilidades del equipo y de la complejidad del producto a generar.

Estas fases definidas son: (12)

- Concepción o Inicio: tiene por finalidad definir la visión, los objetivos y el alcance del proyecto, tanto desde el punto de vista funcional como del técnico. Obteniéndose como uno de los principales resultados una lista de los casos de uso y una lista de los factores de riesgo del proyecto.
- Elaboración: tiene como principal finalidad completar el análisis de los casos de uso y definir la arquitectura del sistema, además se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- Construcción: está compuesta por un ciclo de varias iteraciones, en las cuales se van incorporando sucesivamente los casos de uso, de acuerdo a los factores de riesgo del proyecto. Este enfoque permite por ejemplo contar en forma temprana con versiones del sistema que satisfacen los principales casos de uso. Los cambios en los requerimientos no se incorporan hasta el inicio de la próxima iteración.
- Transición: se inicia con una versión “beta” del sistema y culmina con el sistema en fase de producción.

RUP presenta 3 características que constituyen la esencia de todo el proceso de desarrollo (12):

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

- Dirigido por los casos de uso: los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- Centrado en la arquitectura: la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción. Estos conforman los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.
- Ciclo de vida iterativo: una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

RUP brinda una fuerte y abundante documentación, permitiendo un futuro entendimiento de cada componente del producto. Esto es fundamental para la plataforma VideoWeb debido a que si se le quiere incorporar un conjunto de mejoras y no sean los mismos desarrolladores, pues ya se tiene todo bien documentado. Unido a esto, la concepción de la plataforma VideoWeb como producto hace necesaria una buena documentación, favoreciendo el soporte, mantenimiento y producción de nuevas versiones. La experiencia del equipo de desarrollo en esta metodología constituye un factor fundamental en la selección de la misma. (13)

1.4.2 Lenguaje de modelado 2.0

La comunidad del software precisa de una forma para comunicar sus modelos, no solo entre los miembros de un proyecto, sino a todas las personas involucradas en él, y con el paso del tiempo, a los desarrolladores de futuras generaciones. Es por ello que se necesita un lenguaje no solo para comunicarse con otros, sino para proporcionar un marco en el que desarrolladores individuales puedan pensar y analizar las soluciones. Al mismo tiempo estos no pueden retener toda esta información durante meses o años, por tal motivo deben registrarlo sobre papel o electrónicamente. (12)

Para lograr la realización y diseño de los principales artefactos en el desarrollo del proyecto con RUP, se utiliza el UML, que es el lenguaje gráfico de modelaje orientado a objetos estándar de la industria para especificar, visualizar, construir y documentar los elementos de los sistemas de software. Esta

combinación es la más utilizada para realizar el análisis, implementación y documentación de los sistemas, razones por las cuales fue seleccionado (14). Este lenguaje proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto los aspectos conceptuales, (procesos del negocio y funciones del sistema), como los aspectos concretos, (las clases escritas en un lenguaje de programación específico, esquemas de base de datos y componentes de software reutilizables). Lo que simplifica el proceso complejo de análisis y diseño de software, facilitando un plano para la construcción.

1.4.3 Herramienta para modelado 8.0

Se propone para el modelado del sistema la herramienta CASE (Ingeniería de Software Asistida por Computadora) Visual Paradigm, que es una herramienta para el desarrollo de aplicaciones utilizando el lenguaje de modelado UML, su uso es de excelencia en un ambiente de software libre, elemento que influyo notablemente en su selección. Al mismo tiempo es ideal para ingenieros de software, analistas y arquitectos de sistemas, pues brinda confiabilidad y estabilidad en el desarrollo orientado a objetos. También incorpora el soporte para trabajo en equipo, permitiendo que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. (15)

Ventajas

- Permite integrarse con diferentes IDE¹¹.
- Generación de documentación en formatos HTML¹² y PDF.
- Disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, Vista o W8), GNU/Linux, Mac OS X, Solaris o Java.
- Generación de código e ingeniería inversa: brinda la posibilidad de generar código a partir de los diagramas, para las plataformas como .Net, Java y PHP, así como obtener los diagramas a partir del código.
- Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.

Para el desarrollo de la plataforma VideoWeb 2.0 es fundamental que pueda ejecutarse en GNU/Linux ya que se desea un software implementado completamente con herramientas libres.

¹¹ *Integrated Development Environment*

¹² Hace referencia al lenguaje de marcado para la elaboración de páginas web.

Además el equipo de desarrollo está familiarizado con la herramienta. (13)

1.4.4 Gestor de Bases de Dato 9.1

Para manejar todos los datos de la plataforma se va a utilizar PostgreSQL el cual es un sistema gestor de bases de datos objeto-relacional, distribuidos bajo licencia BSD¹³ y con su código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos¹⁴ en vez de multihilos¹⁵ para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'. Es altamente confiable en cuanto a estabilidad se refiere. Este sistema permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. (16)

1.4.5 Lenguaje de programación 5.3.10

PHP es un lenguaje de script interpretado en el lado del servidor, gratuito e independiente de plataforma, rápido, con una gran biblioteca de funciones y mucha documentación. Es usado frecuentemente para la generación dinámica de páginas web. Además tiene soporte para el gestor de bases de datos: PostgreSQL (17).

Ventajas de utilizar PHP:

- Multiplataforma
- Manejo de excepciones
- Permite técnicas de programación orientada a objetos.
- Amplia documentación en su página oficial PHP.
- Es libre.

Como se ha planteado anteriormente la plataforma VideoWeb está implementado bajo el CMS Drupal, el cual está programado en PHP, por tal motivo los módulos desarrollados deben estar bajo este mismo lenguaje para lograr una correcta compatibilidad con el resto de los elementos desarrollados. La experiencia del equipo de desarrollo posibilitará agilizar la preparación en la construcción de los módulos para el funcionamiento de la plataforma VideoWeb 2.0.

¹³ Es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution).

¹⁴ Multiproceso es tradicionalmente conocido como el uso de múltiples procesos concurrentes en un sistema en lugar de un único proceso en un instante determinado.

¹⁵ Es la capacidad en hardware que tienen algunos CPU para ejecutar eficientemente múltiples hilos de ejecución.

1.4.6 Entorno de desarrollo integrado (IDE) 7.3

Como IDE se selecciona NetBeans 7.3 el cual provee una fácil integración con el lenguaje de programación analizado anteriormente. Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, etc. Es un producto libre, gratuito, sin restricciones de uso. Un proyecto de código abierto de gran éxito, con una amplia base de usuarios, una comunidad en constante crecimiento y con cerca de 100 socios en todo el mundo. *Sun Microsystems*¹⁶ fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos. (18)

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo java que contiene clases escritas para interactuar con las APIs de NetBeans y un archivo especial (*manifest file*) que lo identifica como módulo. (18)

1.4.7 Sistemas gestión de contenidos (CMS) Drupal 7.22

Drupal es un sistema de gestión de contenido modular y muy configurable. Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Se destaca por la calidad de su código y de las páginas generadas.

El diseño es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar muchos tipos diferentes de sitio web (3). Como se ha planteado anteriormente la plataforma VideoWeb 1.0 está desarrollada bajo el CMS Drupal 6, en el cual el equipo de desarrollo está capacitado. Los mismos continúan con este gestor pero esta vez con su versión 7.22 que es la última hasta el momento.

1.5 Conclusiones

En el presente capítulo se abordaron una serie de conceptos asociados al problema, específicamente en los módulos de almacenamiento, gestión y publicación de archivos multimedia en plataformas web. De esta forma se logra un mayor entendimiento del mismo para dar inicio a la siguiente fase del problema.

¹⁶ Sun Microsystems fue una empresa informática que se dedicaba a vender ordenadores, componentes informáticos, software y servicios informáticos. www.sun.com

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

Con el estudio realizado de las soluciones nacionales e internacionales existentes en el mundo, se demuestra que estas no satisfacen el problema planteado, por lo que se necesita implementar un nuevo sistema que cumpla con las características trazadas por institución.

Una vez vistas las características, ventajas y desventajas de las herramientas y tecnologías a utilizar el equipo de desarrollo conjunto con el arquitecto definieron a: RUP como metodología de desarrollo, Visual Paradigm 8.0 como herramienta CASE, UML 2.0 como lenguaje de modelado, NetBeans 7.3 como Entorno de Desarrollo Integrado, PHP 5.3.10 como lenguaje de programación, Drupal 7.22 como gestor de contenidos y PostgreSQL 9.1 como gestor de base de datos.

CAPÍTULO 2: Características de la solución

En el presente capítulo se describe el entorno mediante un modelo conceptual para comprender los conceptos presentes en el contexto donde trabaja la plataforma y se realiza el levantamiento de los requisitos funcionales con los que deberá cumplir la aplicación. También se presenta el diagrama de casos de uso del sistema y se describen los mismos.

2.1 Dominio conceptual

Debido a que la plataforma VideoWeb puede ser un producto personalizado o comercializado a cualquier cliente o entidad interesada y teniendo en cuenta que la problemática esencial radica en la reimplementación de los módulos, resulta difícil encontrar procesos de negocio bien estructurados que permitan realizar un modelado completo de él, por lo que se decidió realizar un modelo conceptual.

Un modelo conceptual tiene como principal objetivo lograr identificar y describir todos los conceptos presentes en el dominio del problema, lo cual ayuda a definir los procesos y roles más trascendentales para el sistema a desarrollar. Esto ayuda a los usuarios, desarrolladores y clientes a manejar un vocabulario común. (12)

2.1.1 Descripción del dominio

En la plataforma VideoWeb 2.0 existe la sección usuario **invitado**, que permite a este visualizar las publicaciones y reproducir los archivos multimedia existentes. Una vez que el usuario se autentica puede tomar diferentes roles, entre los que se encuentran **gestor de contenido** o **autor**, que estos son los encargados de gestionar los archivos multimedia y las publicaciones en la plataforma VideoWeb 2.0.

El proceso de **publicación de media** comienza cuando un **autor**, gestiona una publicación que puede ser un archivo de (**audio o video**) el cual se almacena en el **servidor de medias**. Este archivo debe pertenecer a una tipología de archivo multimedia a partir de la cual se definen los **metadatos** asociados a él, los que se almacenan en la **base de datos**.

El **gestor de contenido** gestiona los **archivos multimedia** que puede ser de dos tipos (**audio o video**), y los mismos se almacenan en el **servidor de medias**. Los archivos multimedia se distribuyen sobre el servidor *streaming* y se vincula con las publicaciones de multimedia, que pueden ser visualizadas por los usuarios invitados.

A continuación se realiza el modelado de la plataforma VideoWeb 2.0.

2.1.2 Diagrama de clases del Modelo de Dominio

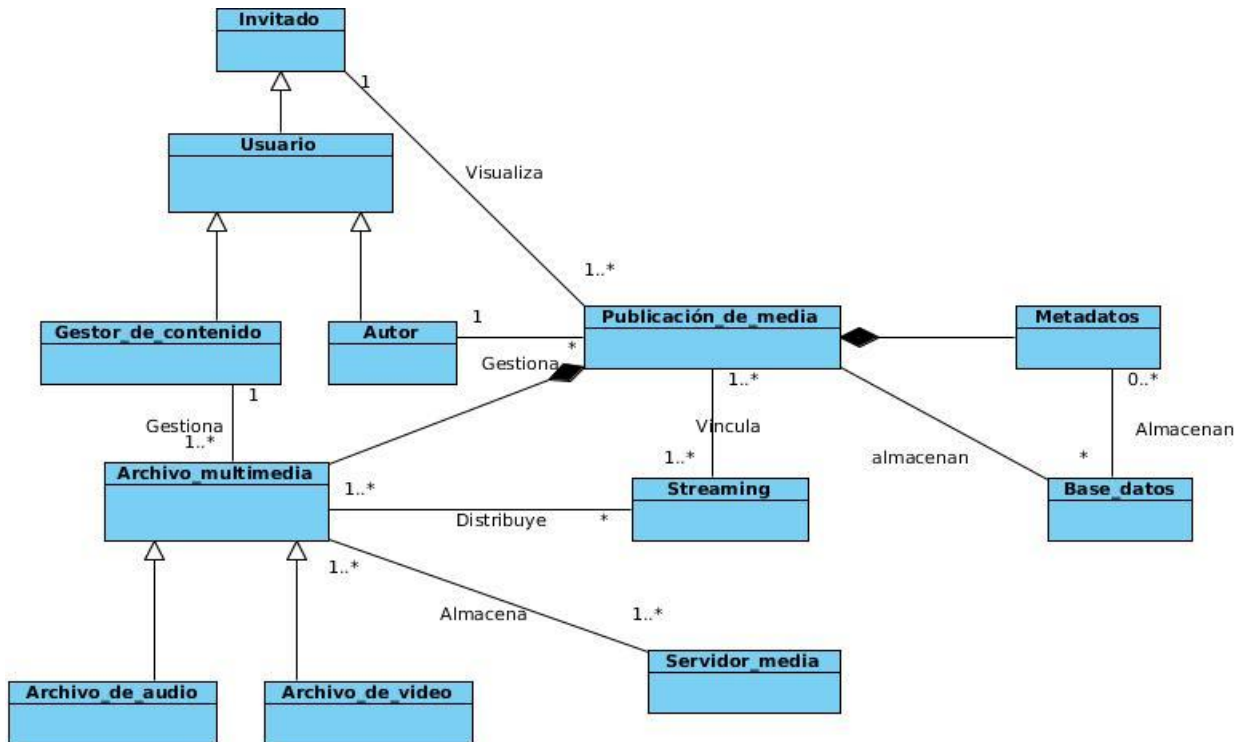


Figura 2 Diagrama del modelo de dominio.

2.1.3 Descripción de las clases del modelo de dominio

- Invitado: usuario no autenticado que tiene permisos para visualizar las publicaciones y reproducir los archivos multimedia.
- Usuario: usuario autenticado que tiene permisos para acceder a las funciones del sistema según su rol.
- Gestor de contenido: usuario autenticado que tiene permisos para gestionar los archivos multimedia.
- Autor: usuario autenticado que tiene permisos para gestionar las publicaciones.
- Archivo multimedia: archivo multimedia de audio o video que será publicado en la plataforma para que pueda ser reproducido por los usuarios.

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

- Archivo de audio: archivo de audio en el formato definido soportado por el servidor *streaming*.
- Archivo de video: archivo de video en el formato definido y optimizado para ser distribuido por la red mediante el servidor *streaming*.
- Publicación de media: entidad conformada por un archivo multimedia y los datos asociados a este previamente almacenado en el servidor de medias para su posterior publicación.
- *Streaming*: servidor encargado de distribuir los archivos multimedia a través de la red.
- Servidor media: servidor de almacenamiento donde se van a guardar físicamente los archivos multimedia con el formato requerido para ser publicados en la plataforma.
- Metadatos: conjunto de datos asociados a los archivos multimedia.
- Base datos: entidad que se almacenan de manera estructurada y con la menor redundancia posible los datos referentes a los artículos de contenido, los contenidos audiovisuales y la información relativa de los usuarios en la plataforma de VideoWeb.

2.2 Requisitos

Los requisitos expresan una condición o capacidad que debe ser conformada por el sistema. Algo que debe hacer o una cualidad que el sistema debe poseer. Estos tienen dos clasificaciones, funcionales y no funcionales. (12)

2.2.1 Requisitos Funcionales.

Los requisitos funcionales de un sistema describen lo que debe hacer, estos representan acciones específicas por el cliente.

RF1 Adicionar punto de almacenamiento.

Descripción: el sistema debe permitir crear punto de almacenamiento. Un punto de almacenamiento es un directorio físico donde se almacenarán los ficheros que se gestionan en el sistema.

Entradas:

Nombre del punto de almacenamiento (obligatorio): es el que identifica donde se colocará el archivo multimedia, ejemplo: almacén de películas, almacén de videos.

Ruta de almacenamiento (obligatorio): determinar la ruta física del punto de almacenamiento donde serán almacenadas las medias correspondientes.

Descripción (opcional): se hace una breve descripción del punto de almacenamiento por

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

ejemplo: “Almacén donde se van a guardar los videos que van a ser utilizados por el sitio VideoWeb de la UCI”.

Salida:

Muestra un mensaje “Se adicionó correctamente el almacenamiento”.

RF2 Editar datos del punto de almacenamiento creado.

Descripción: el sistema debe permitir editar los datos de cualquier punto de almacenamiento creado.

Entradas:

Modificar datos asociados al punto de almacenamiento.

Salidas:

Muestra un mensaje “Se modificó correctamente el almacenamiento”.

RF3 Eliminar punto de almacenamiento.

Descripción: una vez adicionado el punto de almacenamiento el sistema debe permitir eliminarlo.

Entradas: identificador del punto almacenamiento.

Salidas:

Muestra un mensaje “Se eliminó correctamente el almacenamiento”.

RF4 Listar los puntos de almacenamiento.

Descripción: el sistema debe permitir mostrar los puntos de almacenamiento que existen.

Entradas:

Listar (opcional): se marca la opción listar punto de almacenamiento.

Salidas:

Lista de puntos: muestra una lista con todos los puntos de almacenamiento existentes en la plataforma.

RF5 Adicionar punto de publicación de *streaming*.

Descripción: el sistema debe permitir crear puntos de publicación de *streaming* para publicar los materiales contenidos en los puntos de almacenamiento.

Entradas:

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

Punto de almacenamiento (obligatorio): lugar en el servidor de almacenamiento donde se copiará el archivo multimedia.

IP (opcional): dirección IP desde donde se va a realizar el proceso de *streaming*.

Puerto (obligatorio): puerto desde donde se va a realizar el proceso de *streaming*.

Protocolo (obligatorio): protocolo por el cual se va a realizar el proceso de *streaming*.

Salidas:

Muestra un mensaje “Se adicionó correctamente el punto de almacenamiento *streaming*”.

RF6 Editar datos del punto de publicación de *streaming* creado.

Descripción: el sistema debe permitir editar los datos del punto de publicación de *streaming*.

Entradas:

Modificar los datos asociados al punto de publicación de *streaming*.

Salidas:

Muestra un mensaje “Se modificó correctamente el punto de almacenamiento *streaming*”.

RF7 Eliminar punto de publicación de *streaming*.

Descripción: una vez adicionados los punto de publicación *streaming* el sistema debe permitir eliminarlos.

Entradas: identificador del punto de *streaming* a eliminar.

Salidas:

Muestra un mensaje “Se eliminó correctamente el punto de almacenamiento *streaming*”.

RF8 Listar los puntos de publicación de *streaming*.

Descripción: el sistema debe permitir mostrar los puntos de publicación de *streaming* que existan en el sistema.

Entradas:

Listar (opcional): se marca la opción listar punto de publicación *streaming*.

Salidas:

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

Lista de puntos: muestra una lista con los puntos de publicación *streaming* existentes en la plataforma.

RF9 Configurar parámetro de archivo multimedia.

Descripción: el sistema debe permitir al administrador determinar las extensiones permitidas para el archivo multimedia.

Entradas:

Extensiones permitidas (obligatorio): extensiones que pueden ser del tipo ogg, ogv, mp4, avi, mkv, mpeg separadas por coma.

Salidas:

Muestra un mensaje “Ha sido guardada la configuración”.

RF10 Adicionar archivo multimedia.

Descripción: el sistema debe permitir a los usuarios con permisos, almacenar el archivo multimedia en el servidor de medias y crear las referencias pertinentes en la base de datos con todos los datos asociados a estos.

Entradas:

Título (obligatorio): nombre del archivo multimedia.

Descripción (obligatorio): una breve descripción del archivo para tener conocimiento del mismo.

Almacenamiento (obligatorio): lugar en el servidor de almacenamiento donde se copiará el archivo multimedia.

Archivo (obligatorio): se selecciona el archivo multimedia desde su ubicación física en la PC terminal para subirlo al servidor de almacenamiento.

Tipología (opcional): selección de la tipología a la que pertenecerá el archivo multimedia que se quiere adicionar al sistema.

Salidas:

Muestra un mensaje “El archivo multimedia (*el nombre del archivo*) ha sido adicionado”.

RF11 Editar datos asociados al archivo multimedia creado.

Descripción: el sistema debe permitir a los usuarios con permisos modificar los datos al archivo

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

multimedia.

Entradas:

Puede modificarse los datos de un archivo multimedia ya adicionado con anterioridad.

Salidas:

Muestra un mensaje “El archivo multimedia (*el nombre del archivo*) ha sido modificado”.

RF12 Eliminar archivo multimedia.

Descripción: el sistema debe permitir a los usuarios con permisos eliminar el archivo multimedia y los datos asociados.

Entradas: Identificador asociado al archivo multimedia.

Salidas:

Muestra un mensaje “El archivo multimedia (*el nombre del archivo*) ha sido eliminado”.

RF13 Buscar archivo multimedia.

Descripción: el sistema debe permitir a los usuarios autenticados realizar búsquedas relacionadas con el archivo multimedia en el servidor de medias.

Entradas:

Filtrar búsqueda (opcional): palabra o cadena bajo la cual el usuario espera encontrar un archivo multimedia en el sistema.

Salidas:

Listado de resultado: lista de archivos multimedia relacionados al criterio de búsqueda utilizado por el usuario.

RF14 Listar archivos multimedia

Descripción: el sistema debe permitir mostrar la lista de los archivos multimedia que existan en el sistema.

Entradas:

Listar (opcional): se marca la opción listar archivos multimedia.

Salidas:

Listado de resultado: muestra una lista con todos los archivos existentes en la plataforma.

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

RF15 Adicionar publicación de archivo multimedia.

Descripción: El sistema debe permitir a los usuarios con permisos, crear publicación de archivo multimedia a partir del archivo multimedia previamente almacenado en el servidor de medias, permitiendo predefinir el tiempo de publicación acotado por un inicio y fin.

Entradas:

Nombre (obligatorio): título de la publicación del archivo multimedia.

Punto de publicación (obligatorio): selección del punto donde estará el archivo multimedia a transmitir.

Tipología (obligatorio): se selecciona la tipología para realizar la búsqueda.

Archivo multimedia (obligatorio): selección del archivo multimedia que se va a publicar.

Descripción (obligatorio): breve descripción del archivo multimedia que se va a publicar, puede contener el objetivo de la publicación o una simple descripción del contenido audiovisual que se quiere publicar.

Salidas:

Muestra un mensaje “La publicación ha sido adicionada correctamente”.

RF16 Editar datos de la publicación de archivo multimedia.

Descripción: el sistema debe permitir a los usuarios con permisos modificar los datos de la publicación del archivo multimedia.

Entradas:

Puede modificarse los datos de una publicación ya creada con anterioridad.

Salidas:

Muestra un mensaje “La publicación ha sido modificada correctamente”.

RF17 Eliminar publicación de archivo multimedia.

Descripción: una vez adicionada al menos una de las publicaciones de archivo multimedia el sistema debe permitir a los usuarios con permisos eliminarlas.

Entradas: identificador de la publicación de archivo multimedia.

Salidas:

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

Muestra un mensaje “La publicación ha sido eliminada correctamente”.

RF18 Publicar archivo multimedia

Descripción: el sistema debe permitir a los usuarios con permisos activar las publicaciones de archivos multimedia previamente elaboradas, creando un enlace simbólico al archivo en el punto de publicación.

Entradas:

Activación (opcional): de las publicaciones elaboradas con anterioridad se marca de no publicada a publicada.

Salidas:

Se muestra el archivo multimedia en estado “publicado”.

RF19 Terminar publicación de archivo multimedia.

Descripción: el sistema debe permitir a los usuarios con permisos desactivar las publicaciones de archivos multimedia activas, eliminando el enlace simbólico del punto de publicación.

Entradas:

Activación (opcional): de las publicaciones elaboradas con anterioridad se marca de publicada a no publicada y así termina la publicación del archivo.

Salidas:

Se muestra el archivo multimedia en estado “no publicado”.

RF20 Buscar publicación de archivo multimedia.

Descripción: el sistema debe permitir buscar publicaciones de archivo multimedia existentes en la plataforma.

Entradas:

Título (opcional): palabra o cadena bajo la cual un usuario espera encontrar una publicación de archivo multimedia.

Salidas:

Listado de resultado: listado de publicaciones del archivo multimedia relacionado con las búsquedas introducidas por el usuario inicialmente.

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

RF21 Visualizar publicación de archivo multimedia.

Descripción: el sistema debe permitir visualizar las publicaciones de archivo multimedia.

Entradas:

Estado de la publicación (opcional): se marca el estado de la publicación a “página principal”.

Salidas:

Se muestra la publicación en la página principal.

RF22 Adicionar señal en vivo.

Descripción: el sistema debe permitir a los usuarios registrados crear una señal en vivo.

Entradas:

Nombre (obligatorio): título de la señal en vivo.

Url (obligatorio): dirección donde se encuentra la señal en vivo.

Autor (obligatorio): nombre del autor que tiene el archivo.

Texto de descripción (obligatorio): breve descripción del archivo que pertenece a la señal en vivo.

Salidas:

Muestra un mensaje “La señal en vivo fue adicionada de forma satisfactoria”.

RF23 Editar señal en vivo creada.

Descripción: el sistema debe permitir a los usuarios registrados modificar los datos asociados a las señales de audio o video en vivo.

Entradas:

Una vez adicionado la señal en vivo, el usuario puede modificar los datos pertenecientes a la misma.

Salidas:

Muestra un mensaje “La señal en vivo fue modificada de forma satisfactoria”.

RF24 Eliminar señal en vivo.

Descripción: el sistema debe permitir a los usuarios registrados eliminar señales de audio o video en

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

vivo.

Entradas: identificador de la señal en vivo.

Salidas:

Muestra un mensaje “La señal en vivo fue eliminada de forma satisfactoria”.

RF25 Filtrar señal en vivo.

Descripción: el sistema debe permitir mostrar las señales en vivo registradas en el sistema.

Entradas:

El titulo contiene (obligatorio): se introduce una palabra clave del título una vez adicionado la señal en vivo.

Salidas: muestra una lista de las señales en vivo según los parámetros entrados.

RF26 Codificar video.

Descripción: el sistema debe enviar una petición al gestor de procesos de media para realizar la codificación de video de un fragmento de material a publicar.

Entradas:

Dirección del video (obligatoria): dirección donde se guarda de forma obligatoria la ubicación física del video.

Salidas:

Mensaje de notificación informando que el video ha sido codificado con éxito.

RF27 Extraer fotograma.

Descripción: el sistema debe permitir extraer de un una media los fotogramas claves de la misma.

Entradas:

Dirección del video (opcional): dirección donde se guarda de forma opcional la ubicación de los fotogramas extraídos.

Salidas:

Mensaje de notificación informando que la extracción de fotogramas ha concluido.

2.2.2 Requisitos no Funcionales

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad. (12)

RNF1 Soporte.

- Los módulos desarrollados contarán con una ayuda donde el usuario podrá suplir las dudas que se le puedan presentar durante la utilización de los mismos. Además el usuario deberá recibir un adiestramiento previo en la utilización del sistema.

RNF2 Usabilidad

- El sistema debe contar con una interfaz amigable, interactiva e intuitiva para el cliente.
- El sistema debe mostrar la información de forma lógica y correctamente estructurada.

RNF3 Software

- Para la reproducción de los archivos multimedia se recomienda utilizar el reproductor VLC, como reproductor externo y como *plugin* embebido en el navegador, por ser multiplataforma.

RNF4 Hardware

El sistema para que pueda funcionar correctamente deberá contar su hardware con las siguientes características:

Máquina Cliente:

Procesador Pentium 4 o superior.

- Mínimo 256 Mb de RAM.
- Mínimo 100 Mb de capacidad del disco duro.

Máquinas servidoras:

- Procesador Intel Xeon 2.4 GHz o superior.
- Mínimo 2 Giga de RAM.
- Mínimo 500 GB de capacidad de disco duro.

RNF5 Restricciones de diseño

- Para la implementación de los módulos se tuvo en cuenta las restricciones del diseño, como CMS, Drupal 7.22, como lenguaje de programación, PHP 5.3, a PostgreSQL 9.1

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

como gestor de base de datos, como servidor *streaming* Flumotion 0.10 y como biblioteca de JavaScript a jQuery 1.4.4.

2.3 Descripción de los actores

Una vez definidos los requisitos funcionales y no funcionales del sistema se muestran los actores identificados. Los actores representan personas o hardware externo que interactúa con el sistema. Cada actor asume un conjunto coherente de papeles cuando interactúa con el mismo. (12)

Tabla 1 Actores del sistema

Actores	Justificación
Configurador	Usuario con permiso para determinar los parámetros de conversor de los archivos multimedia y para establecer la configuración de la comunicación con los puntos de almacenamiento y publicación de <i>streaming</i> .
Gestor de Contenido	Usuario autenticado que tiene permisos para gestionar los archivos multimedia.
Autor	Usuario autenticado que tiene permisos para gestionar las publicaciones de archivo multimedia.
Editor	Usuario autenticado que tiene permisos para realizar la publicación y terminar la publicación.
Usuario	Cualquier persona que trate de acceder al sistema y que pueda autenticarse.

2.3.1 Casos de uso del sistema

Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto (12). Los 27

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

requisitos funcionales identificados se agrupan en 15 casos de usos y tienen interacción con los actores descritos anteriormente. A continuación se muestra el diagrama de CU¹⁷ del sistema.

2.3.2 Diagrama de casos de uso

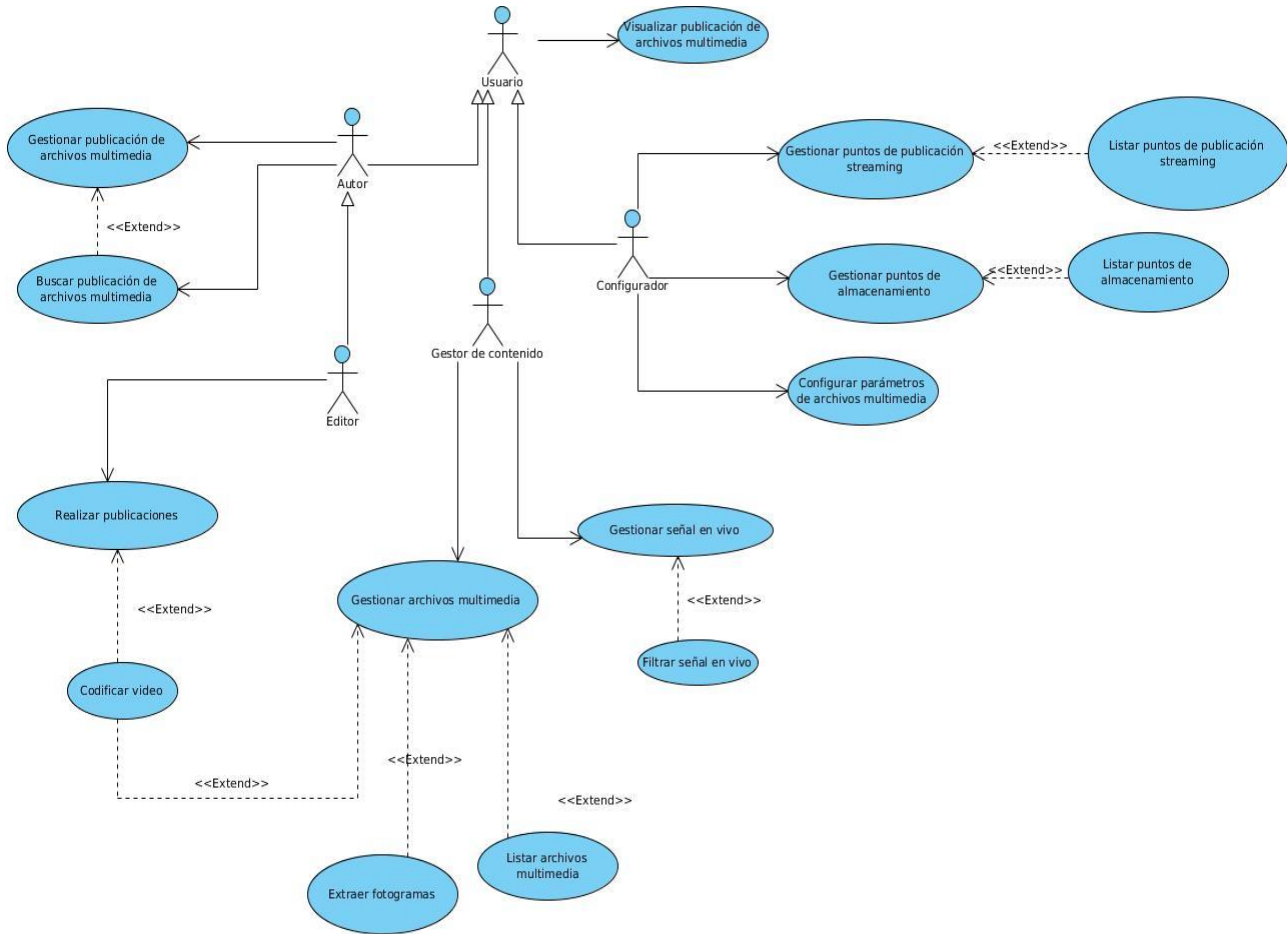


Figura 3 Diagrama de casos de uso del sistema.

2.3.3 Descripción de Casos de Uso del Sistema

Una vez que se han encontrado los actores y los casos de uso, se describe con detalle el flujo normal y el flujo alternativo de cada uno. Describiendo cómo interactúan los actores con el sistema. Debido a la extensión de las descripciones de caso de uso solo se refleja los más significativos. Las restantes se encuentran en el documento de anexos titulado "Descripción de los casos de uso".

¹⁷ Caso de Uso

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

Tabla 2 Descripción Caso de Uso Gestionar archivos multimedia.

Caso de Uso	Gestionar archivos multimedia.	
Objetivo	Permitir adicionar, editar, eliminar y buscar archivo multimedia.	
Actores	Gestor de contenido: (Inicia).	
Resumen	El caso de uso inicia cuando el Gestor de contenido selecciona adicionar, modificar, eliminar y buscar un archivo multimedia, el caso de uso termina cuando se adiciona, modifica, elimina o busca los archivos multimedia.	
Complejidad	Media.	
Referencia	RF10, RF11, RF12 y RF13.	
Prioridad	Alta.	
Precondiciones	El usuario está autenticado y tiene los permisos para editar datos asociados al archivo multimedia, adicionar, eliminar o buscar.	
Flujo de eventos		
Actor		
Sistema		
El gestor de contenido selecciona una de las siguientes opciones. Ver Prototipo de Interfaz 1.		
<ul style="list-style-type: none"> • Adicionar archivo multimedia (A). Ver sección. “Adicionar archivo multimedia”. • Editar archivo multimedia (B). Ver sección. “Editar archivo multimedia “. • Eliminar archivo multimedia (C). Ver sección. “Eliminar archivo multimedia”. • Buscar archivo multimedia (D). Ver sección. “Buscar archivo multimedia”. 		
Sección 1. Adicionar archivo multimedia		
Actor		
Sistema		
	1. El sistema muestra la interfaz para insertar los parámetros requeridos para un nuevo archivo multimedia en el sistema.	
2. El gestor de contenido inserta los	3. El sistema verifica que no existan campos	

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

<p>parámetros de almacenamiento. Ver Prototipo de Interfaz 2.</p> <ul style="list-style-type: none"> • Título (obligatorio) (A). • Ayuda (obligatorio) (B). • Almacenamiento (obligatorio) (C). • Archivo (Dirección de origen del archivo) (obligatorio) (D). • Peso (opcional) (E). • Tipología (opcional) (F). <p>Selecciona la opción “Guardar” (G).</p>	<p>obligatorios vacíos.</p>
	<p>4. El sistema valida que los datos introducidos estén correctos.</p>
	<p>5. El sistema inserta los datos en la base de dato y muestra un mensaje de información. Termina el CU.</p>
<p>Flujos alternos</p>	
<p>3.1. El gestor de contenido selecciona la opción “Cancelar” (H). Ver Prototipo de Interfaz 2. Termina el CU</p>	
	<p>3.1. El sistema muestra un mensaje de error indicando que hay campos vacíos. Regresa al paso 3 del flujo normal “Gestionar archivos multimedia”.</p>
	<p>5.1. El sistema muestra un mensaje de error indicando que hay datos incorrectos. Regresa al paso 3 del flujo básico “Gestionar archivos multimedia”.</p>
<p>Prototipo de Interfaz 1 Ver anexos 4</p>	

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

Prototipo de Interfaz 2 Ver anexos 5	
Sección 2. Editar archivo multimedia	
Actor	Sistema
	1. El sistema busca por el identificador el archivo multimedia seleccionado y muestra una interfaz cargando los datos permitiendo que sean modificados.
2. El gestor de contenido modifica los parámetros deseados. Ver Prototipo de Interfaz 2. <ul style="list-style-type: none"> • Título (obligatorio) (A). • Ayuda (obligatorio) (B). • Almacenamiento (obligatorio) (C). • Archivo (Dirección de origen del archivo) (obligatorio) (D). • Peso (opcional) (E). • Tipología (opcional) (F). Selecciona la opción "Guardar" (G).	3. El sistema verifica que no existan campos obligatorios vacíos.
	4. El sistema valida que los datos introducidos estén correctos.
	5. Inserta los datos en la base de datos. Termina el CU.
Flujos alternos	
3.1 El gestor de contenido selecciona la opción "Cancelar" (H). Ver Prototipo de Interfaz 2. Termina el CU	

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

	3.1. El sistema muestra un mensaje de error indicando que hay campos vacíos. Regresa al paso 3 del flujo básico “Gestionar archivos multimedia”.
	5.1. El sistema muestra un mensaje de error indicando que hay datos incorrectos. Regresa al paso 3 del flujo básico “Gestionar archivos multimedia”.
Sección 3. Eliminar archivo multimedia	
Actor	Sistema
	1. El sistema muestra un mensaje de confirmación para eliminar el archivo multimedia. Ver Prototipo de Interfaz 3
2. El gestor de contenido selecciona la opción “Eliminar” (I).	3. El sistema elimina el archivo multimedia de la base de datos del sistema. Termina el CU.
Flujos alternos	
3.1. El gestor de contenido selecciona la opción “Cancelar” (J). Ver Prototipo de Interfaz 3. Finaliza el CU	
Prototipo de Interfaz 3	
Ver anexos 6	
Sección 4. Buscar archivo multimedia	
Actor	Sistema
1. El gestor de contenido selecciona la opción “Gestionar archivos multimedia”	2. El sistema muestra la lista de archivos multimedia. Termina el CU.

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

Flujos alternos	
<p>3. El gestor de contenido inserta cualquiera de los siguientes filtros de búsqueda:</p> <ul style="list-style-type: none"> • Poner Nombre (A). • Buscar (B). Ver Prototipo de Interfaz 4 	<p>4. El sistema busca en la base datos del sistema el archivo multimedia que cumple con los filtros insertados.</p>
	<p>5. El sistema muestra la lista de archivos multimedia.</p> <p>5.1. Si la lista es vacía no se muestran archivo multimedia.</p>
<p>Prototipo de Interfaz 4</p> <p>Ver anexos 7</p>	

Tabla 3 Descripción Caso de Uso Gestionar publicación de archivos multimedia.

Caso de Uso	Gestionar publicación de archivos multimedia.
Objetivo	Permitir adicionar, editar y eliminar publicación de archivo multimedia.
Actores	Autor: (Inicia).
Resumen	El caso de uso inicia cuando el autor selecciona adicionar, modificar o eliminar publicación de archivo multimedia, el caso de uso termina cuando se adiciona, modifica o elimina la publicación archivo multimedia.
Complejidad	Media
Referencia	RF15, RF16 y RF17.
Prioridad	Alta
Precondiciones	El usuario está autenticado y tiene los permisos para editar datos asociados a la publicación de archivo multimedia, adicionar o eliminar.
Flujo de eventos	
Actor	Sistema
Selecciona una de las opciones siguientes (Ver prototipo de Interfaz 1):	

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

<ul style="list-style-type: none"> • Adicionar publicación de archivo multimedia (A). Ver sección “Adicionar publicación de archivo multimedia”. • Editar publicación de archivo multimedia (B). Ver sección “Editar publicación de archivo multimedia”. • Eliminar publicación de archivo multimedia (C). Ver sección “Eliminar publicación de archivo multimedia”. 	
Sección 1. Adicionar publicación de archivo multimedia.	
Actor	Sistema
	<p>1. El sistema muestra la interfaz para insertar los parámetros requeridos para una nueva publicación de archivo multimedia en el sistema</p>
<p>2. El autor inserta los parámetros de almacenamiento. Ver prototipo de Interfaz 2.</p> <ul style="list-style-type: none"> • Nombre (obligatorio) (A). • Punto de publicación (obligatorio) (B). • Archivo multimedia (obligatorio) (C). • Descripción (opcional) (F). • Sección (opcional) (D). • Información de la revisión (opcional) (G). • Opciones de comentarios (opcional) (H). • Opciones de la ruta URL (opcional) (I). • Opciones de programación (opcional) (J). • Opciones de la publicación (opcional) (K). <p>Selecciona la opción “Guardar” (N)</p>	<p>3. Verifica que no existan campos obligatorios vacíos.</p>
	<p>4. Valida que los datos introducidos estén correctos.</p>

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

	5. Inserta los datos en la base de datos.
	6. Muestra un mensaje notificando que los datos fueron guardados. Termina el CU.
Flujos alternos	
3.1. El autor selecciona la opción “Cancelar” (L)	3.2. Cancela la operación, cierra la interfaz para insertar publicación de archivo multimedia. Finaliza el CU
	4.1 Muestra un mensaje de error indicando que hay campos vacíos. Regresa al paso 3 del flujo básico “Gestionar publicación de archivo multimedia”.
	5.1 Muestra un mensaje de error indicando que hay datos incorrectos. Regresa al paso 3 del flujo básico “Gestionar publicación de archivo multimedia”.
<p>Prototipo de Interfaz 1 Ver anexos 8</p> <p>Prototipo de Interfaz 2 Ver anexos 9</p>	
Sección 2. Editar publicación archivo multimedia	
Actor	Sistema
	1. El sistema busca los datos de la publicación seleccionada y los muestra en un formulario.
2. El autor modifica los datos de la publicación y presiona el botón “Guardar	3. El sistema valida los datos entrados por el autor, en caso de ser correctos actualiza los

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

publicación” (J). Ver Prototipo de Interfaz 3	datos de la publicación de archivo multimedia y muestra un mensaje indicando que los datos de la publicación han sido actualizados de forma satisfactoria. Termina el CU.
Flujos alternos	
Actor	Sistema
3.1 El autor presiona el vínculo “Cancelar” (K). Ver Prototipo de Interfaz 3. Termina el CU	
	4.1. Si se dejó algún campo requerido en blanco el sistema muestra un mensaje de error indicando que se deben llenar los campos obligatorios y pasa al paso 3 del flujo normal de los eventos.
Prototipo de Interfaz 3 Ver anexos 9	
Sección 3. Eliminar publicación de archivo multimedia	
Actor	Sistema
	1. El sistema muestra un formulario con un mensaje de confirmación y las opciones “Eliminar” (L) y “Cancelar” (M). Ver Prototipo de Interfaz 2.
2. El autor selecciona la opción “Eliminar” (L). Ver Prototipo de Interfaz 4.	3. El sistema verifica el estado de la publicación de archivo multimedia, en caso de no estar publicada la elimina de la base de datos y muestra un mensaje indicando que se eliminó la publicación. Termina el CU.
Flujos alternos	
3.1 El autor selecciona la opción “Cancelar”.	

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

(M). Ver Prototipo de Interfaz 4. Termina el CU.	
	4.1. Si la publicación de archivo multimedia se encuentra publicada el sistema deshabilita el archivo multimedia asociado del servidor de <i>streaming</i> , elimina la publicación de la base de datos y muestra un mensaje indicando que se eliminó la publicación. Termina el CU
Prototipo de Interfaz 4 Ver anexos 10	

2.4 Conclusiones

El modelo dominio realizado permitió conocer todos los términos y conceptos presentes en el entorno, los cuales fueron descritos y especificados. Se logró identificar los roles más significativos y los requisitos con lo que debe cumplir el sistema.

La conformación del diagrama de CUS permitió agrupar los requisitos funcionales en CU. Al mismo tiempo se pudo obtener una visión de cómo sería las interacciones de los roles identificados, con las funcionalidades que brindará el sistema. Además la descripción detallada del flujo normal eventos y el flujo alterno de cada CU permitió comprender en detalles las acciones ejercidas por el actor y la respuesta que brinda el sistema.

CAPÍTULO 3: Análisis y Diseño

En el presente capítulo se comienza a construir la fase de análisis y diseño, dando paso a la elaboración de los diagramas más significativos de cada etapa. Además se define la arquitectura y los patrones de diseño que se utilizan en cada uno de los módulos que se implementarán, también se realiza la modelación de diagrama Entidad-Relación y el de despliegue.

3.1 Modelo de análisis

Un modelo de análisis se describe utilizando el lenguaje de los desarrolladores y puede introducir un mayor formalismo, también es utilizado para razonar sobre los funcionamientos internos del sistema. Ofreciendo una especificación más precisa de los requisitos, de modo que facilita su comprensión, preparación, modificación y mantenimiento en general. (12)

Las **clases de interfaz** modelan las partes del sistema que dependen de sus actores, lo cual implica que clasifican y reúnen los requisitos en los límites del sistema. Por tanto, un cambio en la interfaz de usuario o en un interfaz de comunicaciones queda normalmente aislado en una o más clases de interfaz. (19)

Las **clases de entidad** reflejan la información de un modo que beneficia a los desarrolladores al diseñar e implementar el sistema, incluyendo su soporte de persistencia. (19)

Las **clases de control** representan coordinación, secuencia, transacciones, control de otros objetos y se usan con frecuencia para encapsular el control de los casos de uso. (19)

A continuación se muestra los diagramas de clases del análisis y los de colaboración.

3.1.1 Diagramas de Clases de Análisis

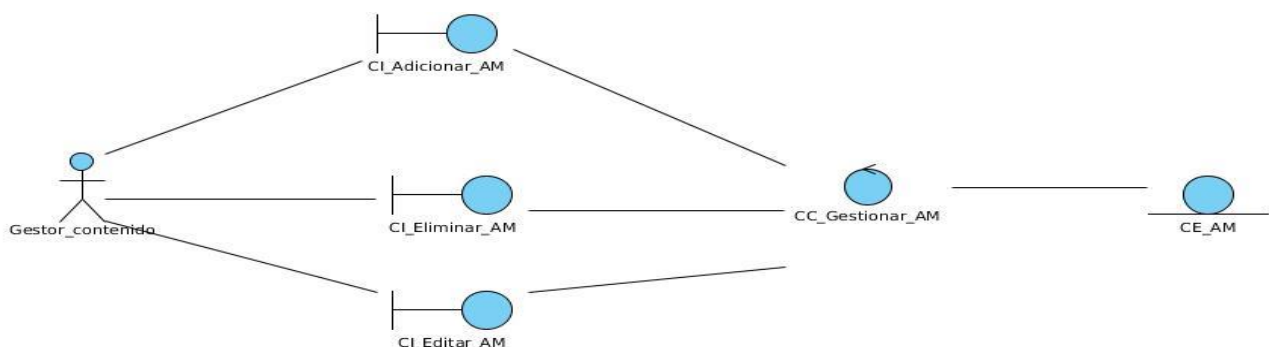


Figura 4 DCA¹⁸ caso de uso. Gestionar archivos multimedia.

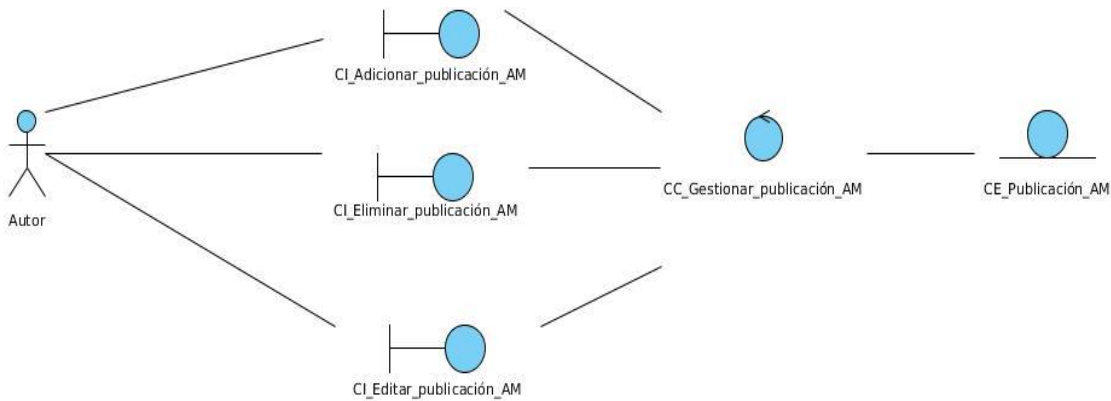


Figura 5 DCA caso de uso. Gestionar publicación de archivo multimedia.

3.1.2 Diagramas de Colaboración

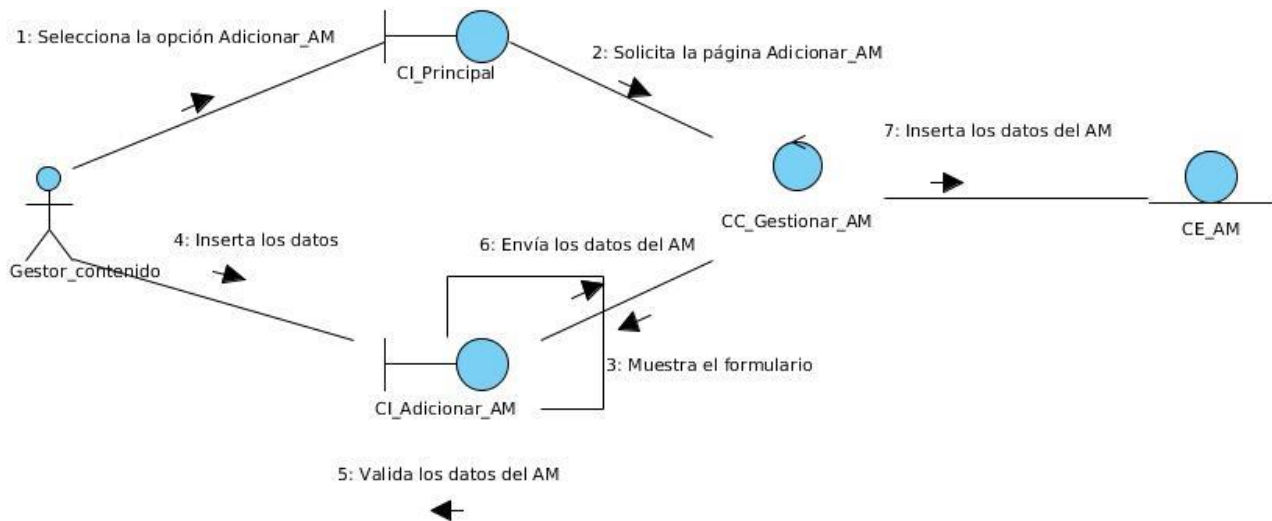


Figura 6 DC¹⁹ caso de uso. Gestionar archivos multimedia: sección adicionar archivo multimedia

¹⁸ Diagrama de Clases del Análisis

¹⁹ Diagrama de Colaboración

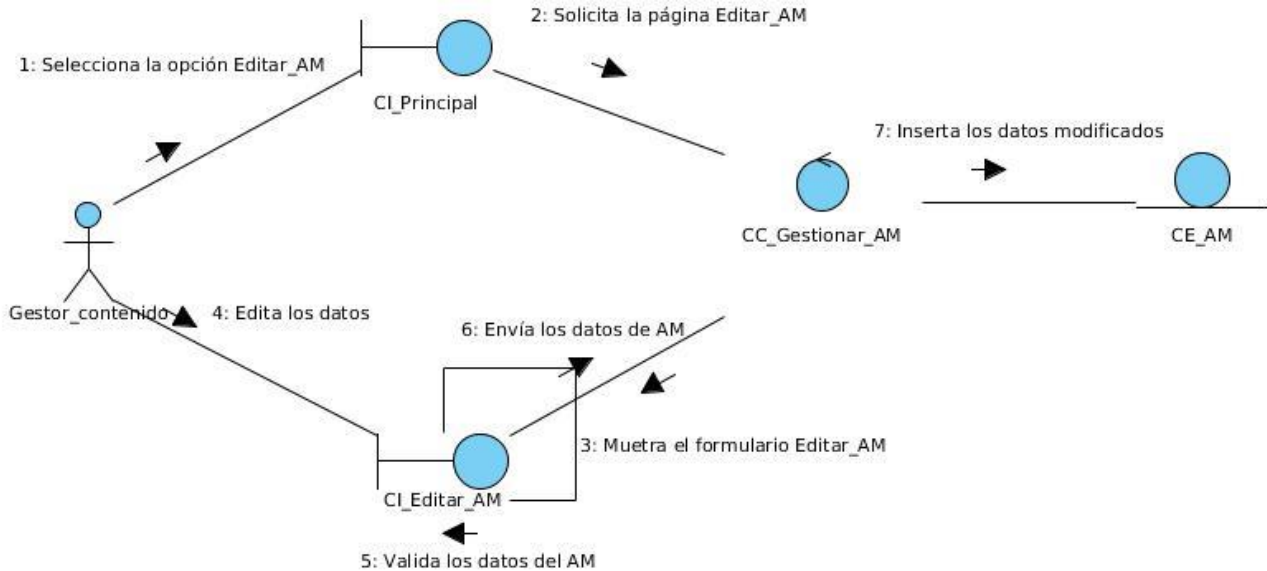


Figura 7 DC caso de uso. Gestionar archivos multimedia: sección editar archivo multimedia.

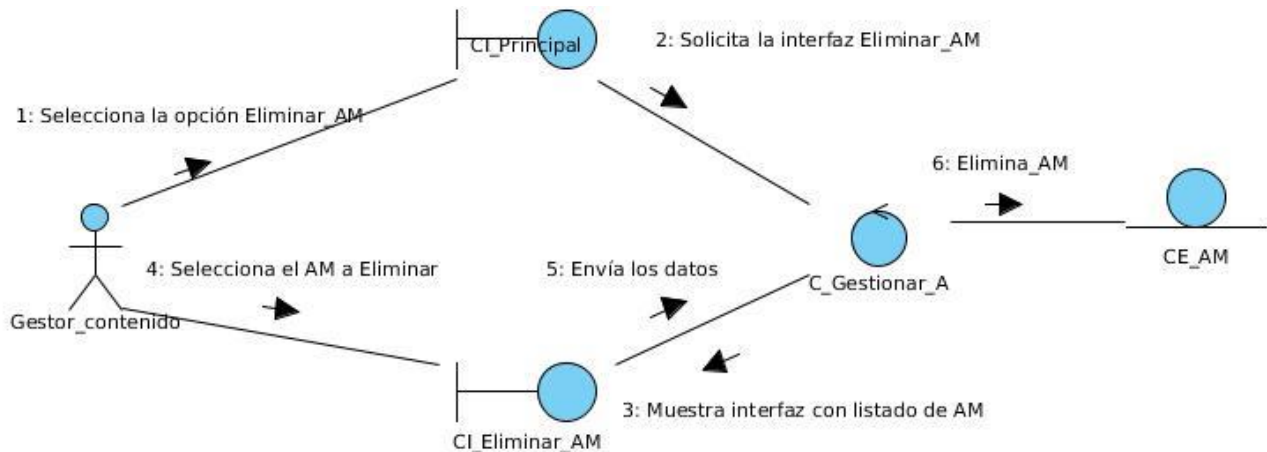


Figura 8 DC caso de uso. Gestionar archivos multimedia: sección eliminar archivo multimedia.

3.2 Arquitectura

La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán. También son los principios que orientan su diseño y evolución (20). Esta agrupa un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a todo el conjunto de desarrolladores compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación.

3.2.1 Patrón Arquitectónico

Los patrones arquitectónicos son patrones de diseño de software que ofrecen soluciones a problemas. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor (21). Para el desarrollo de las funcionalidades a implementar en el módulo se seleccionó el patrón Modelo Vista Controlador (MVC) como se muestra a continuación.

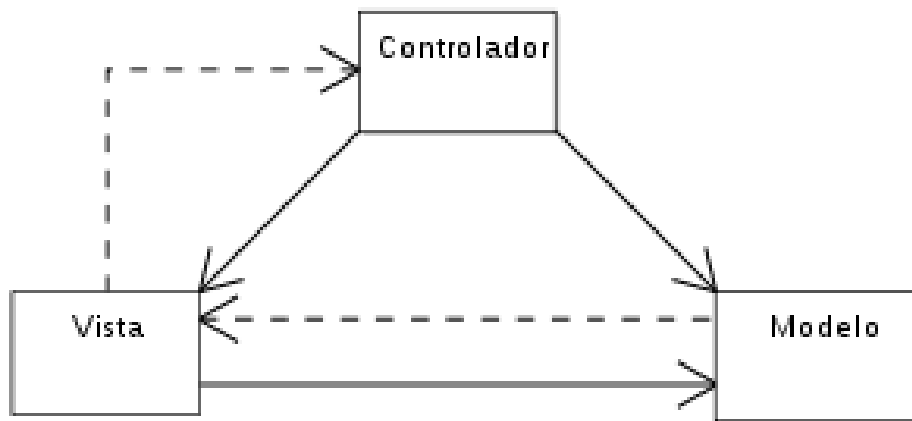


Figura 9 Patrón Modelo-Vista-Controlador.

3.2.1.1 Patrón Modelo-Vista-Controlador

MVC es un patrón de arquitectura de software usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de los conceptos para que el desarrollo esté estructurado, facilitando la programación en diferentes capas de forma paralela e independiente.

Para el desarrollo de la aplicación se decide hacer uso del patrón arquitectónico MVC, este es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz del usuario y es uno de los más utilizado en aplicaciones Web, ya que facilita la funcionalidad y escalabilidad del sistema, de forma simple. (22)

Modelo: es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros que puestos en un contexto del sistema proveen de información al usuario o a la

aplicación misma. (22)

Vista: es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación web la "Vista" es la página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones. (22)

Controlador: es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el modelo en caso de ser necesario. (22)

El funcionamiento de los módulos se basa en diferentes capas donde cada una atiende funcionalidades específicas, están definidas por las vistas, los modelos y las controladoras, que se relacionan según las restricciones de la arquitectura. Esta implementación permite gestionar la lógica del negocio, que luego es utilizada por los *hook* definidos por el núcleo de Drupal para obtener las funcionalidades necesarias en la plataforma. Además provee las clases vistas que posibilitan una interfaz de comunicación para dichos *hook*. Ambas clases se relacionan con las entidades que necesitan para la manipulación de los datos que son gestionados por la plataforma.

3.3 Patrones de Diseño

3.3.1 Patrones GRASP²⁰

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. En el desarrollo de la aplicación se utilizaron una serie de patrones GRAP, específicamente en los módulos de almacenamiento, gestión y publicación.

Experto: encargado de asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Es la asignación de responsabilidades a cada clase, facilitando que los sistemas sean más fáciles de entender, mantener y ampliar. Presentando la oportunidad de reutilizar los componentes en futuras aplicaciones (23). El patrón experto se encuentra reflejado en las clases `Archivo_multimedia.vista` y `Archivo_multimedia.modelo`, también en las clases `Publicación_de_archivo_multimedia.vista` y `Publicación_de_archivo_multimedia.modelo`, además en las clases `Punto_de_almacenamiento.vista` y `Punto_de_almacenamiento.modelo`.

²⁰ GRASP es un acrónimo que significa "General Responsibility Assignment Software Patterns", patrones generales de software para asignar responsabilidades.

Creador: se refiere a asignar a las clases la responsabilidad de crear una instancia de otra. Soportando mayor claridad y encapsulamiento (23). El patrón creador se encuentra reflejado en las clases `Archivo_multimedia.controlador`, `Publicación_de_archivo_multimedia.controlador` y `Punto_de_almacenamiento.controlador`.

Bajo acoplamiento: es asignar una responsabilidad para mantener el bajo acoplamiento. Mantener las clases lo menos ligadas posibles, es la medida de la fuerza con que una clase está conectada a otras. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, acrecentando la oportunidad de una mayor productividad (23). Este patrón se ve reflejado en todas las clases del diagrama.

Alta cohesión: se refiere a asignar una responsabilidad de modo que la cohesión siga siendo alta. Es la medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una clase con alta cohesión es fácil darle mantenimiento, entenderla y reutilizarla (23). Este patrón se ve reflejado en todas las clases del diagrama.

Controlador: se encarga de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Es un objeto de interfaz no destinado al usuario que se encarga de manejar un evento del sistema. Permitiendo un mayor potencial de los componentes reutilizables (23). El patrón controlador se encuentra reflejado en las clases `Archivo_multimedia.controlador`, `Publicación_de_archivo_multimedia.controlador` y `Punto_de_almacenamiento.controlador`.

3.3.2 Patrones GOF²¹

Singleton (Instancia única): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia (24). El uso de dicho patrón se evidencia en todos los módulos que posee Drupal y en los que se creen nuevos.

Bridge (Puente): desacopla una abstracción de su implementación (24). El uso de este patrón se evidencia en la capa de abstracción de bases de datos de Drupal. Los módulos necesitan ser escritos de forma que sean independientes del sistema.

Command (Orden): encapsula una operación en un objeto, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma (24). Este patrón se usa para reducir el número de funciones que son necesarias para la aplicación.

²¹ Gang of Four

3.4 Modelo de Diseño

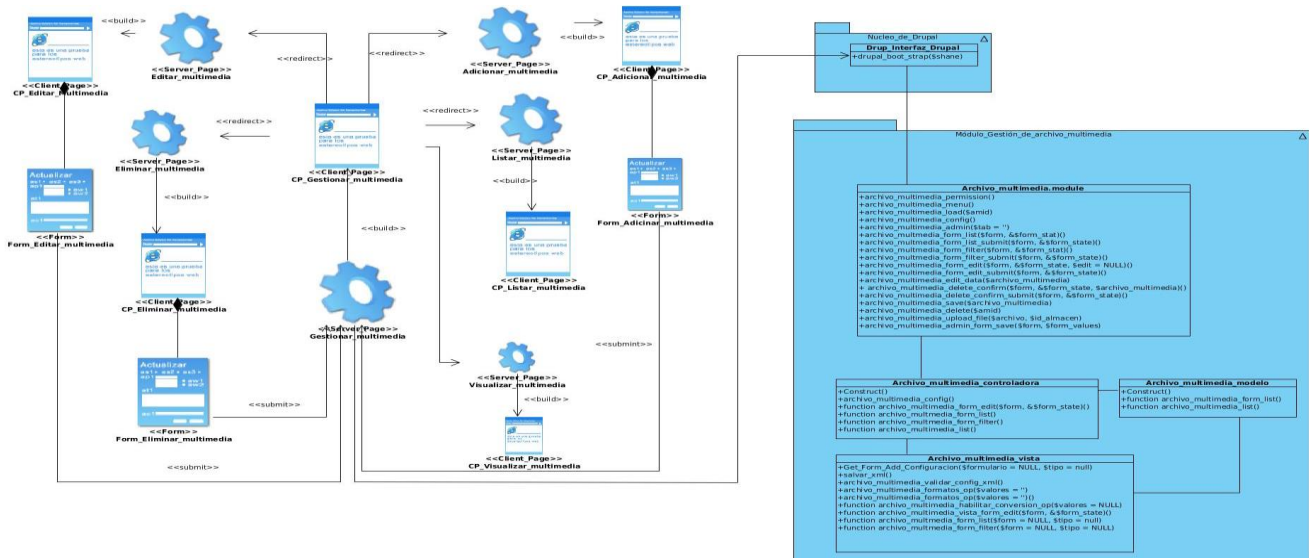
El modelo de diseño, es un modelo de objetos que describe la realización física de los casos de uso centrándose, principalmente en el impacto que tienen los requisitos funcionales y no funcionales. Además, este modelo sirve de abstracción a la implementación del sistema y es, de ese modo, utilizado como entrada fundamental de las actividades de implementación. A continuación se describen los prototipos de web, que se utilizan en el Diagrama clases de diseño. (12)

Server Page: representa una página web que tiene *scripts*²² que son ejecutados por el servidor. Estos *scripts* interactúan con recursos del servidor como bases de datos, lógica de negocio, sistemas externos y se encarga de construir (*build*) o generar el resultado HTML. (25)

Client Page: es una página web con formato HTML y una mezcla de datos, presentación e incluso lógica. Las páginas clientes son representadas por los navegadores clientes, y pueden contener *scripts* que son interpretados por el navegador. (25)

Form: es una colección de campos de entrada que forman parte de una página cliente. Los formularios envían sus datos al código servidor para ser procesados los pedidos (*submit*). (25)

3.4.1 Diagramas de clases del Diseño



²² Script es un guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades. Son ejecutados por un intérprete de línea de comandos y usualmente son archivos de texto.

Figura 10 DCD²³ Gestionar archivos multimedia

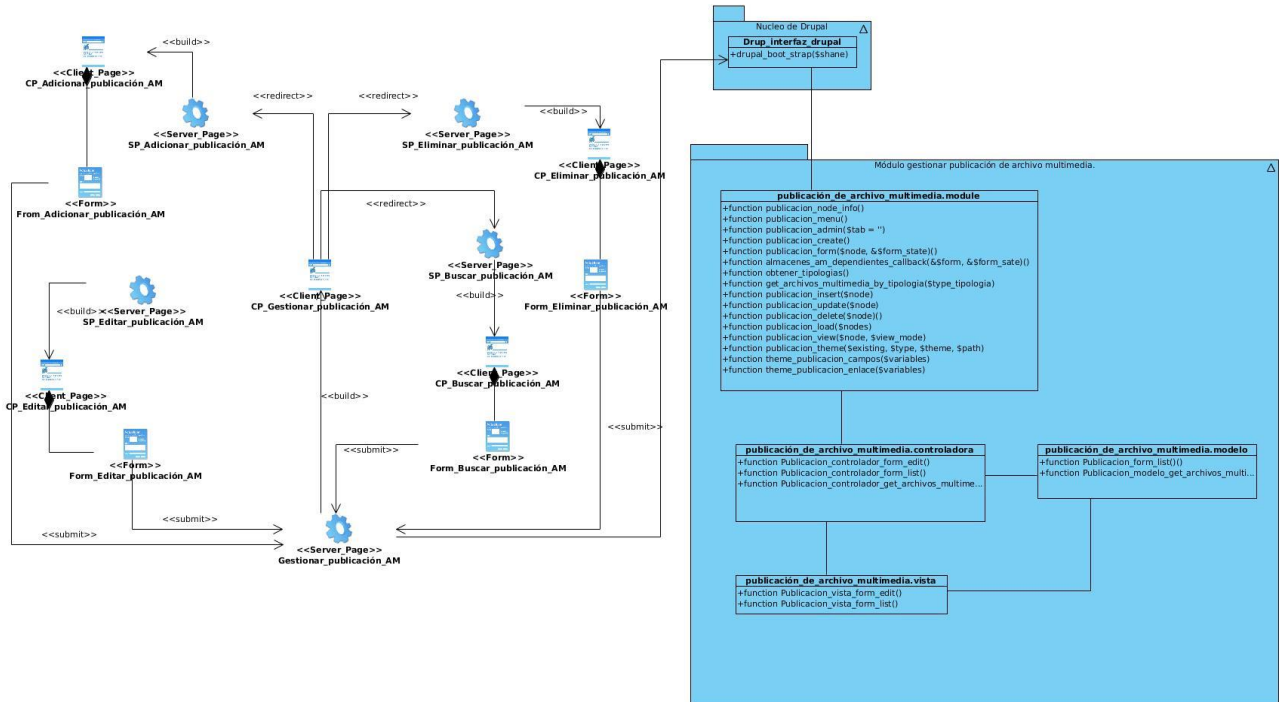


Figura 11 DCD Gestionar publicación de archivo multimedia.

3.4.2 Diagrama de Despliegue

La metodología RUP, plantea que el diagrama de despliegue tiene como objetivo capturar la configuración de los elementos de proceso y las conexiones, en el sistema. Además, permite comprender el entorno de ejecución física del sistema y para comprender las cuestiones de distribución. Por tanto, constituye el medio principal para conocer el ambiente donde se desarrolla el sistema. A continuación se muestra el diagrama de despliegue con lo que está conformado el sistema. (26)

²³ Diagrama de Clases del Diseño

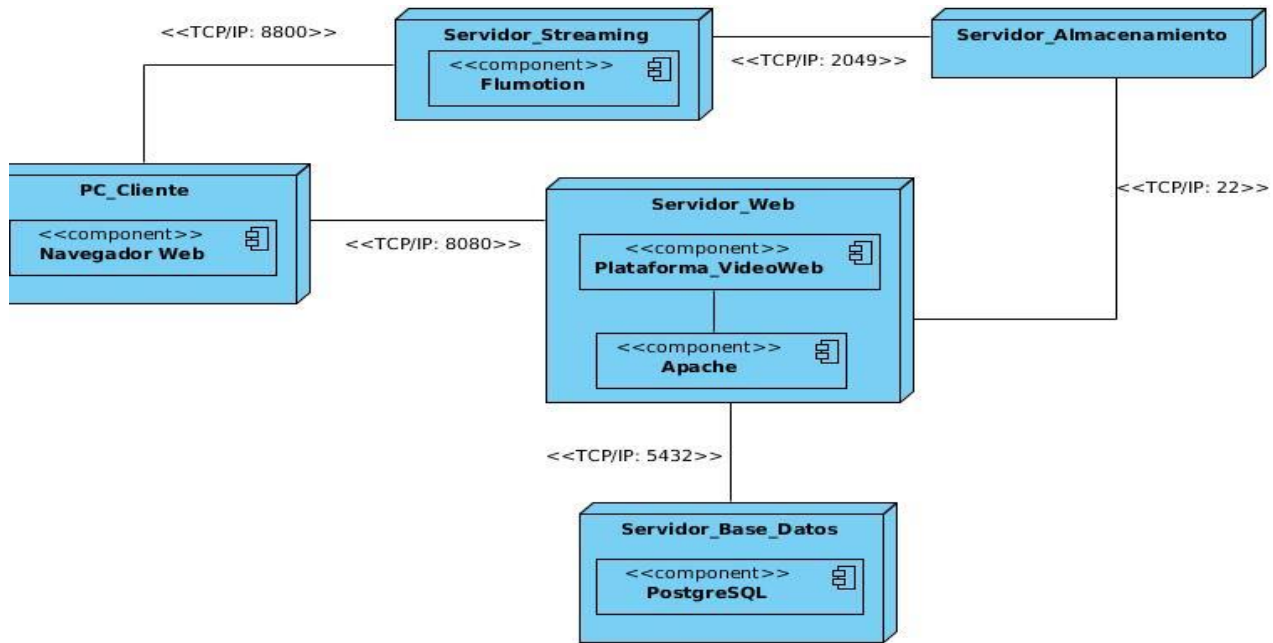


Figura 12 Diagrama de despliegue.

Descripción de los Nodos:

- PC Cliente: nodo cliente en el cual el actor accederá a la aplicación mediante un navegador web.
- Servidor Web: nodo donde se encontrará instalado el servidor web Apache que alojará los módulos pertenecientes a la plataforma, en este caso almacenamiento, gestión y publicación de archivos multimedia.
- Servidor de Base datos: nodo donde se encontrará la base de datos del componente.
- Servidor Almacenamiento: nodo donde se almacenan los archivos multimedia.
- Servidor *Streaming*: nodo encargado de realizar los procesos de *streaming*.

3.4.3 Modelo Datos

Un modelo de base de datos es un tipo de modelo de datos que determina la estructura lógica de una base de datos y de manera fundamental determina el modo de almacenar, organizar y manipular los datos. (27)

Para dar solución a los requisitos funcionales definidos en los módulos de publicación, almacenamiento y gestión de archivos multimedia en la plataforma VideoWeb 2.0, se agregan varias tablas en la base de datos, que van a contener los atributos necesarios para su correcto

funcionamiento. A continuación se muestran las tablas, las relaciones que existen entre ellas y con otras ya existentes, quedando así estructurada.

La siguiente Figura muestra el Diagrama-Entidad-Relación de los módulos.

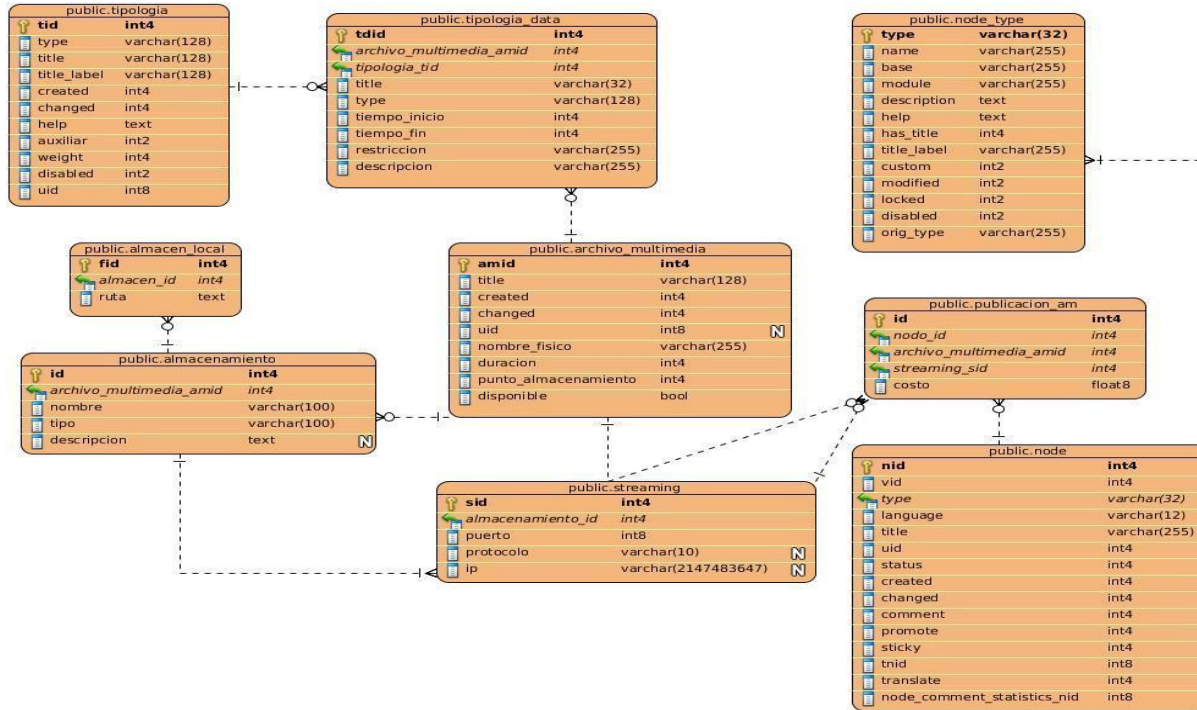


Figura 13 Diagrama Entidad-Relación.

A continuación se describen las tablas generadas por los módulos de almacenamiento, publicación y gestión de archivo multimedia. La tabla **public.archivo_multimedia** contiene la información necesaria de los archivos multimedia.

- **title**: define el título del archivo multimedia.
- **created**: fecha de creación.
- **changed**: fecha de modificado.
- **uid**: representa el id del usuario que lo agrega al sistema.
- **Nombre_físico**: nombre físico del archivo multimedia.
- **duración**: duración del archivo multimedia.
- **punto_almacenamiento**: id del almacenamiento donde estará alojado.
- **punto_almacen_fotograma**: id del almacenamiento donde guarda el fotograma

CAPÍTULO 3: ANÁLISIS Y DISEÑO

- disponible: define si está disponible o no para ser publicado.
- cataloged: define si el archivo multimedia ha sido catalogado o no.
- id_conversión: guarda el id del flujo de conversión y el id de la notificación de la conversión.

La tabla **public.almacenamiento** contendrá la información necesaria de los almacenamientos.

- id: del almacenamiento.
- nombre: define el nombre del almacenamiento.
- tipo: define el tipo de almacén.
- descripción: define la descripción del almacenamiento.

La tabla **public.publicacion_am** contendrá la información necesaria de las publicaciones.

- nodo_id: representa el id de la publicación.
- archivo_multimedia_amid: representa el id del archivo multimedia asociado a la publicación.
- *streaming_sid*: representa el id del almacén utilizado para la publicación.
- costo: define el costo de la publicación del archivo multimedia.

3.5 Conclusiones

En el presente capítulo se modelaron los diagramas de clase del análisis y los del diseño, permitiendo una mejor abstracción y entendimiento de los módulos con los que va a contar la aplicación.

Cada uno de ellos responde al estilo arquitectónico MVC, manteniendo las responsabilidades definidas por el patrón, además se logra flexibilizar el código adaptándolo a los patrones de diseño.

También se modeló el diagrama de despliegue y el modelo entidad-relación que ayuda a comprender la distribución física del sistema.

CAPÍTULO 4: Validación de la solución propuesta

El presente capítulo se refleja el flujo de trabajo de implementación y pruebas, de acuerdo con las características de RUP. Se desarrolla el diagrama de componente que no es más que la visión física del sistema. Además se exponen las técnicas de prueba usadas para validar las entradas y salidas de la solución, para detectar y documentar los errores del sistema, y probar que los requisitos fueron implementados de acuerdo con lo planteado.

4.1 Modelo de implementación

El modelo de implementación está constituido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño. Este artefacto describe como se implementan los componentes, congregándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre estos. (28)

4.1.1 Diagrama de componente

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema. Representa las clases del diseño en componentes de código fuente, mostrando las relaciones entre estos.

El diagrama de componentes que se muestra a continuación, modela la distribución física de los componentes. El resto de los módulos se puede encontrar en el anexo.

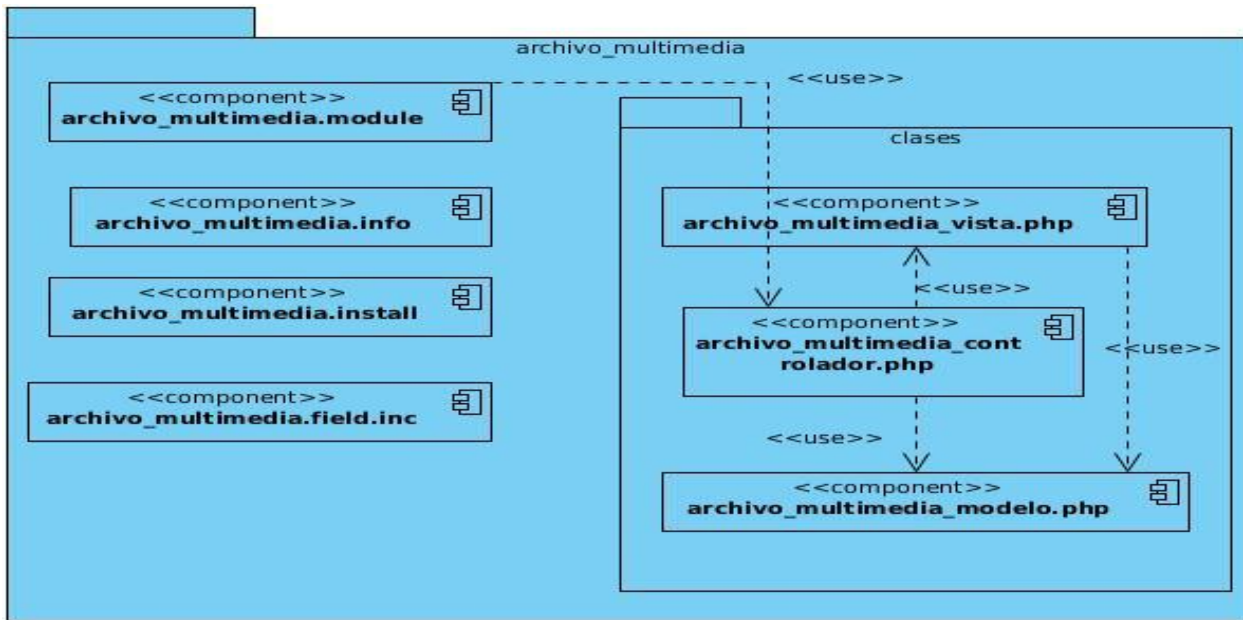


Figura 14 Diagramas de componente del módulo archivo multimedia

4.2 Pruebas del sistema

Las pruebas del sistema buscan discrepancias entre el programa y sus objetivos o requisitos, enfocándose en los errores hechos durante la fase de implementación. El éxito de una prueba depende de descubrir un error no detectado hasta entonces y su objetivo es diseñar casos de prueba que, sistemáticamente, saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y de esfuerzo (21).

4.2.1 Métodos de Prueba

Existen dos métodos de prueba fundamentales que se le aplican a un producto software con el objetivo de que los módulos una vez que han sido implementados queden libres de errores. Estas pruebas son de caja blanca y de caja negra. La de caja blanca se basa en la lógica interna del programa utilizando técnicas de diseño de casos de prueba y la de caja negra se basa en los requisitos del software utilizando técnicas de diseño de casos de prueba. (29)

Para probar el correcto funcionamiento de los módulos implementados, se realizan las pruebas de caja negra, por lo que deben estar muy bien definidos los parámetros de entradas y salidas.

4.2.2 Pruebas de Caja Negra

El sistema de pruebas de caja negra no considera la codificación dentro de sus parámetros a evaluar, es decir, que no están basadas en el conocimiento del diseño interno del programa. Estas pruebas se

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

centran principalmente en los requisitos funcionales de software.

La prueba de caja negra no es una alternativa a las técnicas de prueba de la caja blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la caja blanca. Muchos autores consideran que estas pruebas permiten encontrar: (19)

- Funciones incorrecta o ausente.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.

Es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, esta suele ser la última etapa de pruebas al software antes de proceder a su entrega a los usuarios.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

Técnica del Análisis de Valores Límites: esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de caja negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software. (29)

Esta prueba consiste en introducirle un conjunto de valores equivalentes para las pruebas:

Si un parámetro de entrada debe estar comprendido en un cierto rango hay 3 clases de equivalencia: por debajo, en y por encima del rango.

Si una entrada requiere un valor de entre los de un conjunto, aparecen 2 clases de equivalencia: en el conjunto o fuera de él.

Si una entrada es booleana, hay 2 clases: si o no. (30)

4.2.3 Aplicación de las pruebas de Caja Negra.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

4.2.3.1 Caso de prueba para el caso de uso Gestionar archivos multimedia.

Tabla 4 Casos de pruebas. CU Gestionar archivos multimedia.

Casos de Prueba					
Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central	Resultados Esperados	Resultados Obtenidos
SC 1. Gestionar archivos multimedia	EC 1.1. Adicionar archivo multimedia satisfactoriamente.	El usuario inserta en el sistema los valores correspondientes al AM ²⁴ según los campos a llenar.	VideoWeb/ Estructura/ Archivo Multimedia/ Adicionar archivo multimedia.	El sistema debe mostrar un mensaje de notificación indicando <i>“El archivo multimedia (el nombre del archivo) ha sido adicionado”</i> .	Muestra el mensaje correspondiente al proceso de adiconamiento.
	EC 1.2. Adicionar archivo multimedia falla.	El usuario inserta los datos del archivo multimedia en la interfaz usuario y el mismo no cumple con el formato de video que debe	VideoWeb/ Estructura/ Archivo Multimedia/ Adicionar archivo multimedia.	Una vez enviados los datos, el sistema debe mostrar un error indicando <i>“Formato de</i>	Muestra el mensaje correspondiente al proceso de falla.

²⁴ Archivo Multimedia

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

		pertenecer el archivo.		archivo multimedia no soportado”	
EC 2.1.	Eliminar archivo multimedia satisfactoriamente.	El usuario selecciona el AM anteriormente adicionado para eliminarlo de la plataforma y presiona el botón “Aceptar”.	VideoWeb/ Estructura/ Archivo Multimedia/ Eliminar.	El sistema elimina el AM y muestra un mensaje informando datos eliminado correctamente.	Muestra el mensaje correspondiente al proceso de eliminación.
EC 2.2.	Eliminar archivo multimedia falla.	El usuario selecciona el archivo que desea eliminar.	VideoWeb/ Estructura/ Archivo Multimedia/ Eliminar.	El sistema muestra un mensaje de error indicando que ocurrió un error en el proceso de eliminación.	Muestra el mensaje correspondiente al proceso.
EC 3.1.	Editar archivo multimedia satisfactoriamente.	El usuario selecciona de la lista de los AM, el archivo que desea modificar y cambia	VideoWeb/ Estructura/ Archivo Multimedia/ Editar.	El sistema valida los campos y muestra un mensaje	Muestra el mensaje correspondiente al proceso de editar el

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

	amente.	en los campos los parámetros correspondientes.		indicando, datos modificados satisfactoriam ente.	archivo.
	EC 3.2. Editar archivo multimedia falla.	El usuario modifica los datos del archivo multimedia y los envía.	VideoWeb/ Estructura/ Archivo Multimedia/ Editar.	El sistema muestra un error indicando que hubo un error en los datos enviados.	Muestra el mensaje correspondient e al proceso.

4.2.4 Descripción de variable.

Tabla 5 Descripción de las variables para el CU Gestionar archivos multimedia.

Nro.	Nombre de Campo	Clasificación	Valor Nulo	Descripción
1	Título	Campo de texto.	No	Admite cadenas desde 1 a 255 caracteres. Que puede ser cualquier combinación de letras, números, guiones bajos.
2	Ayuda	Campo de texto.	No	Admite cualquier combinación de letras y números.
3	Tipología	Campo de selección.	Si	Se selecciona una de las tipologías a la que

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

				pertenece el archivo.
4	Peso	Campo de texto.	Si	Cualquier combinación de números enteros.
5	Id archivo multimedia	Campo oculto	No	Número entero que debe estar en correspondencia con el identificador de archivo multimedia existente en la base de datos.

4.2.5 Secciones a revisar.

SC 1 Adicionar archivo multimedia.

Tabla 6 SC 1 Adicionar AM

Id del escenario	Título	Ayuda	Respuesta del Sistema	Resultado de la Prueba
EC 1.1. Adicionar archivo multimedia satisfactoriamente.	V	V	Se añade el archivo multimedia y se muestra un mensaje que fue satisfactorio.	Satisfactorio.
EC 1.2. Adicionar archivo multimedia falla.	V	I	Se muestra un mensaje de error indicando que el archivo no se adicionó pues los datos no son correctos.	Satisfactorio
	I	V		
	I	I		

SC 2 Eliminar archivo multimedia.

Tabla 7 SC 2 Eliminar AM.

Id del escenario	Id archivo multimedia	Respuesta del Sistema	Resultado de la Prueba

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

EC 2.1. Eliminar archivo multimedia satisfactoriamente.	V	Se elimina el archivo multimedia correctamente y se muestra un mensaje informando que se la operación fue satisfactoria.	Satisfactorio.
EC 2.2. Eliminar archivo multimedia falla.	I	Se muestra un mensaje indicando que el archivo multimedia no se pudo eliminar de forma satisfactoria	Satisfactorio

SC 3 Editar archivo multimedia.

Tabla 8 SC 3 Editar AM

Id del escenario	Título	Ayuda	Tipología	Peso	Respuesta del Sistema	Resultado de la Prueba
EC 3.1. Editar archivo multimedia satisfactoriamente.	V	V	V	V	Se edita el archivo multimedia correctamente y se muestra un mensaje informando que la operación fue satisfactoria.	Satisfactorio
EC 3.2. Editar archivo multimedia falla.	V	I	V	V	Se muestra un mensaje de error indicando que el archivo no se adicionó pues los datos no son correctos.	Satisfactorio
	I	V	V	I		
	I	I	I	V		

En la documento **Caso de pruebas** se encuentran los restantes casos de prueba aplicados al

módulo.

4.3 Resultados de pruebas

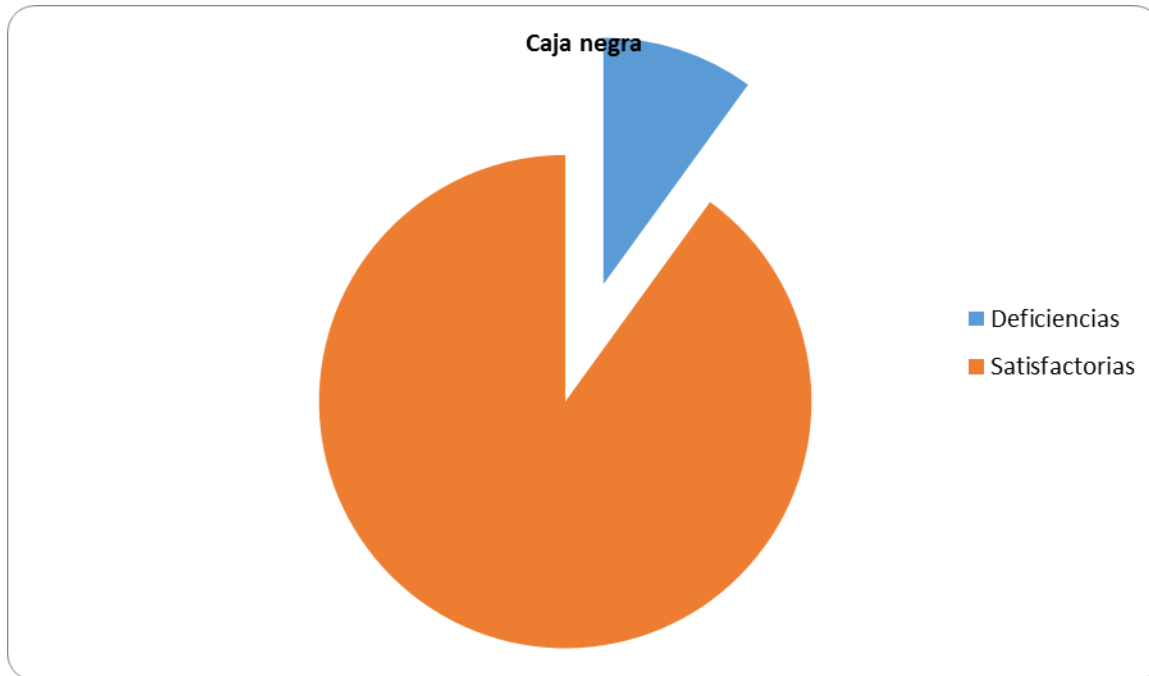


Figura 15 Resultados de las pruebas de caja negra.

Las pruebas de caja negra se desarrollaron de manera estricta a los casos de prueba expuestos anteriormente, entrando los juegos de datos que cumple con la partición de equivalencia que utiliza la misma. Una vez concluidas se obtuvo un resultado 90% satisfactorio y 10% de deficiencia como se ve en el gráfico anterior. El resultado de cada prueba coincide con la especificación de requisito lo que demuestra la correcta implementación de los mismos.

La no conformidad encontrada en los archivos multimedia fue resuelta de manera inmediata y actualmente funciona correctamente, estableciéndose que una vez que el usuario quiera subir un archivo multimedia debe registrarse por los formatos establecidos en la aplicación, al mismo tiempo se detectó que la opción de eliminación del archivo no cumplía con el requisito expuesto anteriormente. Se hicieron 4 iteraciones y el último arrojo resultados 100% satisfactorios dando por terminada la fase de pruebas para este caso de uso.

4.4 Conclusiones

Con la elaboración del diagrama de componentes se tuvo una visión más clara de la distribución física del sistema, permitiendo que otros desarrolladores conozcan las relaciones que existen entre

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

los componentes.

Después de haberle realizado las pruebas al sistema permitió revelar los errores existentes en la implementación de la misma, entregándolo a los usuarios finales libre de no conformidades.

Conclusiones generales

Una vez culminado el trabajo de diploma permitió no solo desarrollar los módulos de almacenamiento, gestión y publicación de archivo multimedia para la plataforma VideoWeb 2.0, sino que permitió arribar a las siguientes conclusiones:

- Las tareas investigativas que se trazaron contribuyeron a la organización de la investigación de manera fácil.
- Con la implementación de los módulos de almacenamiento, publicación y gestión de archivo multimedia, se obtuvo mejoras en la plataforma, facilitando los procesos administrativos del cliente.
- Se utilizaron herramientas y tecnologías de software libre en la construcción del producto, que posibilitan la comercialización del mismo.
- Los artefactos generados por RUP, permiten tener un control y una documentación detallada del sistema, dándole la posibilidad a los futuros desarrolladores entender y mejorar el producto.
- Las pruebas desarrolladas al sistema, arrojaron resultados 100 % satisfactorio. Dándole culminación a los módulos implementados, quedando el cliente a gusto con la entrega.

Recomendaciones

Una vez concluido el desarrollo de los Módulos de gestión, almacenamiento y publicación de archivos multimedia para VideoWeb 2.0 y luego de haber cumplido totalmente con los objetivos trazados al inicio de la investigación, se presentan las siguientes recomendaciones.

- Hacer un nuevo módulo que se integre con el de archivo multimedia, que permita que los usuarios puedan compartir sus propios videos en la plataforma como lo hace YouTube, teniendo en cuenta el derecho autor.
- El módulo de pasarela de pago se debe integrar con el de archivo multimedia para que cumpla con el negocio del sistema, ya que cada video debe tener un precio. Teniendo en cuenta que los usuarios que suban los videos también pueden recibir bonificaciones de pago según la calidad del video.

Bibliografía

1. **ESTEVA-GRILLET, R.** [En línea] El dibujo en Venezuela: Estudio y antología de textos. Fundarte. Alcaldía de Caracas. http://www.fen.ufg.br/fen_revista/revista4_1/dibujo.html.
2. UCI. [En línea] www.uci.cu.
3. Drupal. [En línea] 11 de 4 de 2005. <http://drupal.org.es/drupal>.
4. **Ortiz Frida, García María del Pilar.** Editorial Limosa. Metodología de la Investigación. [En línea] México(2005).
5. Real Academia Española. [En línea] www.rea.es.
6. **Lópe, Josmar.** Concepto de Archivos y Datos. [En línea] <http://boards5.melodysoft.com/GM005N/concepto-de-archivos-y-datos-4.html>.
7. **SÁNCHEZ, MYRIAM RUÍZ y MARÍA ISABEL LEAL RUGAMA.** *Multimedia*. 2008.
8. **Casanovas, August.** 2005.
9. **YouTube.** Software ¿Como funciona YouTube? [En línea] 2012. <http://www.highscalability.com>.
10. **Felipe, Luis.** Consejos para la gestión de videos en WordPress. [En línea] 11 de marzo de 2013. <http://mvkoen.com/consejos-para-los-videos-en-wordpress/>.
11. **Abreu, Angel Dayán Marín.** *Desarrollo de la Plataforma VideoWeb*. La Habana : s.n., 2009.
12. **Booch, Grady; Jacobson, Ivar; Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000.
13. **Rodríguez, Frank Torres.** *Desarrollo de la Arquitectura de Software* . La habana : s.n., 2009.
14. **Jiménez, Yasiel González.** *Arquitectura del servidor de streaming distribuido ALLFRY'S*. La habana : s.n., 2011.
15. **VP.** Visual Paradigm . *About Visual Paradigm* . [En línea] 2011. [Citado el: 5 de Diciembre de 2012.] <http://www.visual-paradigm.com/aboutus>.
16. **PostgresSQL.** Portal español sobre PostgreSQL. [En línea] 2010. [Citado el: 13 de Diciembre de 2012.] http://www.postgresql.org.es/sobre_postgresql#caracteristicas.
17. **Gutmans, Andi.** *PHP5 Power Programming*. s.l. : Prentice Hall, 2005.

18. **NetBeans IDE**. Pagina Oficial. [En línea] 2012. [Citado el: 15 de Diciembre de 2012.] <http://www.netbeans.org/about/index.html>.
19. **Pressman, Roger S.** *Software Engineering, a practitioner's approach*. . 2011. ISBN 9780071267823.
20. **IEEE Std, 1471-2000**. 2010.
21. **Pressman, Roger.** *Ingeniería de Software. Un enfoque práctico*. España : s.n., 2005. 9701054733.
22. **Cabrera, Daymel Machado.** *Implementación del módulo Distribución de productos*. Habana : s.n., 2010.
23. **Benitez, Aliuska Rivero.** *Módulo de resumen de video integrado a la Plataforma VideoWeb*. La habana : s.n., 2012.
24. **Pérez Mariñán, P.** Patrones de Diseño . [En línea] ww.ecured.cu.
25. **Pressman, Roger S.** *Ingeniería del Software*. 2007.
26. **Morreira, Dianeyis Santiago.** *Desarrollo del Módulo Web de Monitorización y Administración del Sistema de Video Vigilancia* . La Habana : s.n., 2011.
27. **Zhuge, H.** The Web Resource Space Model. Web Information Systems Engineering and Internet Technologies Book Series. [En línea]
28. **Centro Nacional de Tecnologías de Información.** [En línea] GNU FDL. http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
29. Prueba funcional o Prueba de Caja Negra. [En línea] <http://testingfuncional.wordpress.com/2011/03/12/%C2%BFtesting-funcional-o-pruebas-de-caja-negra/>.
30. **D. Rodríguez, G. López.** *Information Engineering Research Group* .
31. **Gil, Frank.** *Experto en Drupal, Nivel Avanzado*. 2012.
32. **Larman, Graig.** *UML y Patrones* .
33. **Isaías Carrillo Pérez y Pérez González, Rodrigo.** *Metodología de desarrollo del software*.
34. **Lockhart, Thomas.** *Tutorial de PostgreSQL*. 2009.
35. **EVA.** Ingeniería de Software I. Conferencia 10. [En línea] 2011. [Citado el: 21 de 05 de 2012.]
36. **Cornejo, José Enrique González.** *¿Qué es UML?* 2004.

37. Bahit, Eugenia. *POO y MVC en PHP*. 2011.
38. Martínez, Juan Alberto. *Los IDE de desarrollo*. 2010.
39. Real Academia Española, RAE. *Diccionario de la real academia*. [En línea] [Citado el: 15 de Diciembre de 2012.] <http://www.rae.es/drae/>.
40. Gestión de la Información. Ecured. [En línea] 2012.
http://www.ecured.cu/index.php/Gesti%C3%B3n_de_la_informaci%C3%B3n_en_las_tecnolog%C3%ADas.
41. Martin, Sergio Adrián]. *PostgreSQL: Una poderosa base de datos libre*. 2011.
42. Bejerano, R. F. Gestión del Conocimiento. Conceptos, aplicaciones y experiencias. [En línea]
http://www.ecured.cu/index.php/Gesti%C3%B3n_de_la_informaci%C3%B3n_en_las_tecnolog%C3%ADas.

Referencias bibliográficas

1. **ESTEVA-GRILLET, R.** [En línea] El dibujo en Venezuela: Estudio y antología de textos. Fundarte. Alcaldía de Caracas. http://www.fen.ufg.br/fen_revista/revista4_1/dibujo.html.
2. UCI. [En línea] www.uci.cu.
3. Drupal. [En línea] 11 de 4 de 2005. <http://drupal.org.es/drupal>.
4. **Ortiz Frida, García María del Pilar.** Editorial Limosa. Metodología de la Investigación. [En línea] México(2005).
5. Real Academia Española. [En línea] www.rea.es.
6. **Lópe, Josmar.** Concepto de Archivos y Datos. [En línea] <http://boards5.melodysoft.com/GM005N/concepto-de-archivos-y-datos-4.html>.
7. **SÁNCHEZ, MYRIAM RUÍZ y MARÍA ISABEL LEAL RUGAMA.** *Multimedia*. 2008.
8. **Casanovas, August.** 2005.
9. **YouTube.** Software ¿Como funciona YouTube? [En línea] 2012. <http://www.highscalability.com>.
10. **Felipe, Luis.** Consejos para la gestión de videos en WordPress. [En línea] 11 de marzo de 2013. <http://mvkoen.com/consejos-para-los-videos-en-wordpress/>.
11. **Abreu, Angel Dayán Marín.** *Desarrollo de la Plataforma VideoWeb*. La Habana : s.n., 2009.
12. **Booch, Grady; Jacobson, Ivar; Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000.
13. **Rodríguez, Frank Torres.** *Desarrollo de la Arquitectura de Software* . La habana : s.n., 2009.
14. **Jiménez, Yasiel González.** *Arquitectura del servidor de streaming distribuido ALLFRY'S*. La habana : s.n., 2011.
15. **VP.** Visual Paradigm . *About Visual Paradigm* . [En línea] 2011. [Citado el: 5 de Diciembre de 2012.] <http://www.visual-paradigm.com/aboutus>.
16. **PostgreSQL.** Portal español sobre PostgreSQL. [En línea] 2010. [Citado el: 13 de Diciembre de 2012.] http://www.postgresql.org.es/sobre_postgresql#caracteristicas.

REFERENCIAS BIBLIOGRÁFICAS

17. **Gutmans, Andi.** *PHP5 Power Programming*. s.l. : Prentice Hall, 2005.
18. **NetBeans IDE.** Pagina Oficial. [En línea] 2012. [Citado el: 15 de Diciembre de 2012.] <http://www.netbeans.org/about/index.html>.
19. **Pressman, Roger S.** *Software Engineering, a practitioner's approach*. . 2011. ISBN 9780071267823.
20. **IEEE Std, 1471-2000.** 2010.
21. **Pressman, Roger.** *Ingeniería de Software. Un enfoque práctico*. España : s.n., 2005. 9701054733.
22. **Cabrera, Daymel Machado.** *Implementación del módulo Distribución de productos*. Habana : s.n., 2010.
23. **Benitez, Aliuska Rivero.** *Módulo de resumen de video integrado a la Plataforma VideoWeb*. La habana : s.n., 2012.
24. **Pérez Mariñán, P.** Patrones de Diseño . [En línea] ww.ecured.cu.
25. **Pressman, Roger S.** *Ingeniería del Software*. 2007.
26. **Morreira, Dianeyis Santiago.** *Desarrollo del Módulo Web de Monitorización y Administración del Sistema de Video Vigilancia* . La Habana : s.n., 2011.
27. **Zhuge, H.** The Web Resource Space Model. Web Information Systems Engineering and Internet Technologies Book Series. [En línea]
28. **Centro Nacional de Tecnologías de Información.** [En línea] GNU FDL. http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
29. Prueba funcional o Prueba de Caja Negra. [En línea] <http://testingfuncional.wordpress.com/2011/03/12/%C2%BFtesting-funcional-o-pruebas-de-caja-negra/>.