

Universidad de las Ciencias Informáticas

FACULTAD 6



**Título: Módulo de consulta y recuperación de materiales
audiovisuales.**

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

Autor(es): Miriam Pérez Pérez.

Tutor(es): Yoandri Quintana Rondón

Junio, 2012



“Dadas las circunstancias adecuadas, sin más base que los sueños, la determinación y la libertad de intentarlo, personas muy corrientes hacen constantemente cosas extraordinarias.”

Dee Ward Hock

Fundador de la asociación de tarjetas de crédito VISA.

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Miriam Pérez Pérez

Ing. Yoandri Quintan Rondón

DATOS DE CONTACTO

Diplomante

Nombre y apellidos: Miriam Pérez Pérez

Sexo: F **Grupo:** 6505 **Correo electrónico:** mpperez@estudiantes.uci.cu

Tutor

Nombre y apellidos: Yoandri Quintana Rondón

Sexo: M **Institución:** Universidad de las Ciencias Informáticas (UCI)

Dirección de la institución: Carretera a San Antonio de los Baños, Km. 2 ½, Reparto: Torrens, Municipio: Boyeros, Provincia: Ciudad de La Habana.

Cargo del trabajador: Profesor

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas

Año de graduación: 2009 **Correo electrónico:** yqrondon@uci.cu

Institución donde se graduó: UCI

Oponente

Nombre y apellidos: Yesleny Becerra Torreira

Sexo: F **Institución:** UCI

Cargo del trabajador: Profesora

Título de la especialidad de graduado: Ingeniera en Ciencias Informáticas

Año de graduación: 2009 **Correo electrónico:** ybecerra@uci.cu

Institución donde se graduó: UCI

AGRADECIMIENTOS

A mi mamita por ser tan maravillosa , por su apoyo incondicional , y por enseñarme que el camino a veces es duro pero vale la pena.

A mi papito por ser tan estricto conmigo, lo cual me motivó a esforzarme cada día más para lograr mi objetivo.

A mi tío Michel por creer siempre que yo podía alcanzar la meta que me propusiera en la vida.

A mis abuelitos por ser tan lindos y tan buenos conmigo, por ayudarme en los momentos que más los necesité y darme esos consejos tan buenos que solo ellos saben dar.

A toda mi familia, por la sencilla razón de existir y estar siempre presente.

A mi pipo, que siempre me ha guiado por el camino de la superación y el estudio haciendo de mí una mejor persona y creándome el interés y la motivación para ser hoy quien soy.

A Luis y a Ernesto...gracias por ser conmigo como mis padres, los quiero y los valoro mucho por ser tan especiales.

A mi Yoandri, mi tutor, por ser tan atento y preocupado conmigo, por ser más que mi tutor mi amigo, por apoyarme cuando los momentos eran en verdad difíciles para mí.

A mis súper amigas Ana Margarita, Judith, y Anisbel, más que mis amigas mis hermanas, por demostrarme que aún se puede creer en el valor de la amistad y que buenos amigos valen más que cualquier tesoro en el mundo.

A Walfrido porque a pesar del poco tiempo que llevamos juntos me ha apoyado cuando lo he necesitado y con sus acciones me ha demostrado que tenemos que pasar la vida al lado de personas que te demuestran día a día que eres especial, gracias por los momentos tan especiales que he pasado junto a ti.

A Dainelys que ha sido mi amiga, desde que entre en primer año, gracias por ser tan especial y por demostrarme que todavía existen bichitos raros que creen en el valor de la amistad y su mami que es una de las personas más maravillosas que he conocido y que le voy a

agradecer por siempre que haya sido como una madre para mí.

A mis dos amigos varones que han sido los hermanos que nunca tuve...Elieser y Alejandro...demás está decirles que los quiero muchísimo.

A Karen, Mariemy, Eniley, Tailen, Katia, Indira, Lesdy, Sandor, Viñolo, Reinel, Ernesto, Ulises por todo su apoyo, por los buenos momentos que no se olvidan, demás está decirles que tienen un lugar especial en mi corazón.

A Daniel Sampedro, a Yoendri, Sisley y a Arian, por su ayuda cuando los necesité, por ser amigos cuando más me hizo falta.

A Yainery por haberse ganado en tan poquito tiempo un lugarcito en mi corazón.

A las muchachitas del apartamento por la convivencia más buena que haya podido tener, y

porque son además personas maravillosas, en especial a Eliana y a Annaliet.

A todos los profesores que han contribuido con mi formación a lo largo de estos años, pero no puedo dejar de mencionar a Juan Amador, Geovany y Zorilín por su gran apoyo cuando más los necesité.

A todos los amigos que conocí desde primer año hasta aquí, que hicieron que mis días en la universidad fueran grandiosos.

Y por último y no menos importante quiero agradecerle a mi abuelita Miriam, porque a pesar de que no pueda estar hoy aquí conmigo, ella ha sido mi inspiración para llegar donde estoy ya que esa fue la última cosa que me pidió y para mí es un orgullo haberlo logrado.

A todos...muchísimas gracias!!!

DEDICATORIA

A unas personitas especiales que siempre me han apoyado, que me han hecho ser lo que soy... Mis Padres.

A mi tío Michel, por la confianza que depositó en mí siempre.

A una mujer maravillosa que a pesar que no pueda estar aquí, sé que desde el cielo va a estar muy feliz... Mi Abuelita.

RESUMEN

Los sistemas de consulta y recuperación de materiales audiovisuales en los últimos años han pasado a formar parte importante de la sociedad, de manera más específica en aquellas personas que trabajan en entidades como televisoras y videotecas. Este tipo aplicaciones, a partir de la demanda que existe sobre las medias (audio, video, imagen) han evolucionado con el propósito de encontrar la información que el usuario necesita. Teniendo en cuenta esas necesidades se han creado software con el objetivo de localizar y visualizar materiales audiovisuales, los cuales no satisfacen las necesidades actuales de Cuba y la universidad, pues son propietarios y de alto costo en el mercado.

En el presente trabajo se propone desarrollar una aplicación que permita mejorar la consulta y recuperación de materiales audiovisuales, la misma debe responder a las necesidades del proyecto Sistema de Captura y Catalogación de Medias (SCCM). La propuesta para la implementación se basa en el uso de herramientas y tecnologías libres, garantizando la soberanía tecnológica que propone el país. Además se obtienen los artefactos generados en el proceso de desarrollo del software.

PALABRAS CLAVE

Audiovisuales, Consulta, Recuperación.

TABLA DE CONTENIDOS

DECLARACIÓN DE AUTORÍA II

DATOS DE CONTACTO III

AGRADECIMIENTOS IV

DEDICATORIA VI

RESUMEN VII

TABLA DE CONTENIDOS VIII

INDICE DE FIGURAS XII

INDICE DE TABLAS XIII

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 5

 1.1 Introducción 5

 1.2 Términos asociados al problema 5

 1.2.1 Materiales audiovisuales 5

 1.2.2 Sistema de Recuperación de Información 6

 1.2.3 Recuperación de materiales audiovisuales 6

 1.2.4 Solicitud de Materiales Audiovisuales 6

 1.3 Proceso de consulta y recuperación de materiales audiovisuales 7

 1.3.1 Descripción General 7

 1.3.2 Descripción actual del dominio del problema 7

 1.3.3 Situación Problemática 8

 1.4 Análisis de otras soluciones existentes 8

 1.4.1 YouTube 8

 1.4.2 DailyMotion 9

 1.4.3 Metacafe 10

 1.4.4 Mediabox 10

 1.4.5 TdtmTarsys, de la empresa Tedial 11

1.4.6 Videoma	11
1.5 Conclusiones Parciales.....	12
CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS PARA LA SOLUCIÓN	13
2.1 Introducción	13
2.2 Metodología de desarrollo de software	13
2.2.1 RUP (Proceso Unificado de Desarrollo o Rational Unified Process)	14
2.3 Lenguaje Unificado de Modelado	16
2.4 Herramientas CASE	18
2.4.1 Visual Paradigm.....	18
2.5 Tendencias y tecnologías actuales.....	19
2.5.1 Modelo Cliente-Servidor	19
2.6 Lenguaje de Programación	20
2.6.1 Lenguaje del Lado del Cliente	21
2.6.1.1 HTML (Lenguaje de Marcado de Hipertexto)	21
2.6.1.2 JAVASCRIPT	21
2.6.1.3 CSS (Cascading Style Sheets, Hojas de estilo en Cascada)	22
2.6.1.4 AJAX (Asynchronous JavaScript And XML (JavaScript asíncrono y XML))	22
2.6.2 Lenguaje del Lado del Servidor	23
2.6.2.1 Personal Home Page (PHP)	23
2.7 Framework.....	24
2.7.1 Symfony.....	24
2.7.2 DOJO.....	25
2.8 Entorno de Desarrollo Integrado (IDE).....	26
2.8.1 Netbeans	26
2.9 Sistemas Gestores de Bases de Datos (SGBD)	27
2.9.1 Postgresql.....	27
2.10 Conclusiones Parciales	28
CAPÍTULO 3: SOLUCIÓN PROPUESTA	29
3.1 Introducción	29

3.2 Patrones Arquitectónicos	29
3.2.1 Modelo-Vista-Controlador	29
3.2.2 Patrones del Diseño	30
3.2.2.1 Experto.....	30
3.2.2.2 Bajo Acoplamiento	30
3.2.2.3 Controlador	31
3.2.2.4 Decorador	31
3.2.2.5 Observador	31
3.3 Diagramas de clases del diseño	31
3.3.1 Diagrama de clases del diseño para el caso de uso “Buscar Material”	32
3.3.2 Diagrama de clases del diseño para el caso de uso “Reproducir Material”	32
3.3.3 Diagrama de clases del diseño para el caso de uso “Gestionar Solicitud”	34
3.4 Diseño de la Base de Datos.	35
3.5 Diagrama de Despliegue	37
3.6 Diagrama de Implementación.....	37
3.7 Estándar de codificación	38
3.8 Conclusiones Parciales	39
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.	40
4.1 Introducción	40
4.2 Pruebas	40
4.3 Casos de Prueba	41
4.3.1 Diseño de caso de prueba para el caso de uso Buscar materiales:	41
4.3.2 Diseño de caso de prueba para el caso de uso Reproducir Materiales.....	42
4.3.3 Diseño de caso de prueba para el caso de uso Gestionar Solicitud.....	42
4.3.4 Caso de prueba para el caso de uso Atender Solicitud	43
4.3.5 Caso de prueba para el caso de uso Descargar Solicitud	43
4.4 Resultados	44
4.5 Conclusiones Parciales	44
CONCLUSIONES GENERALES.....	46

RECOMENDACIONES.....	47
BIBLIOGRAFÍA.....	48
GLOSARIO DE TÉRMINOS.....	52

INDICE DE FIGURAS

Figura 1: Logo de You tube (8)	9
Figura 2: Logo de DailyMOtion (10).....	9
Figura 3: Logo de metacafe (11).....	10
Figura 4: Logo de Mediabox (12).....	10
Figura 5: Logo de tedial	11
Figura 6: Logo de Videoma (14).....	11
Figura 7: Flujo de trabajo de la metodología RUP.	15
Figura 8: Modelo Arquitectónico Cliente-Servidor. (19)	20
Figura 9: Tecnologías agrupadas bajo el concepto de AJAX. (27)	22
Figura 10: Diagrama de clases del diseño para el caso de uso “Buscar Material”	32
Figura 11: Diagrama de clases del diseño para el caso de uso reproducir material.....	33
Figura 12: Diagrama de clases del Diseño para el caso de uso Gestionar Solicitud.....	34
Figura 13: Diagrama de clase del diseño del caso de uso Atender Solicitud.....	35
Figura 14: Modelo de datos.	36
Figura 15: Diagrama de despliegue	37
Figura 16: Diagrama de componente	38
Figura 17:Codigo Php	39

INDICE DE TABLAS

Tabla 1: Caso de prueba para el caso de uso Buscar material	41
Tabla 2: Caso de prueba para el caso de uso Reproducir materiales	42
Tabla 3: Caso de prueba para el caso de uso Gestionar solicitud.	43
Tabla 4: Caso de prueba para el caso de uso Atender solicitud.....	43
Tabla 5: Caso de prueba para el caso de uso Descargar Solicitud.....	44

INTRODUCCIÓN

Desde sus inicios el hombre sintió la necesidad de comunicarse e intercambiar información, primero lo hizo mediante señas, luego a través de la voz y posteriormente con la escritura. Con el devenir de los años, el aumento de las poblaciones y el desarrollo de las tecnologías, surge la necesidad de medios de comunicación masivos, dando paso al nacimiento de la radio, el cine y la televisión.

El surgimiento de la televisión tuvo gran impacto social, las personas podían observar imágenes y al mismo tiempo escuchar sonidos. Surgen así las televisoras, empresas dedicadas a recepción, producción, almacenamiento y transmisión de materiales audiovisuales.

El video y el audio poseen gran poder para transmitir información a otras personas, ya sea educativa, comercial o simplemente para el entretenimiento. Hace 20 años era muy difícil lograr difundir información en diferentes formatos y de lograrse se hacía con muy mala calidad. Esto estaba dado por el poco ancho de banda con que contaban las redes en sus inicios y el gran tamaño que tienen los archivos de audio y video. Actualmente la sociedad utiliza la tecnología como un instrumento ventajoso en la vida diaria.

Con el transcurso de los años el almacenamiento fue aumentando considerablemente y a las empresas que trabajaban con materiales audiovisuales se les dificultaba el acceso a los mismos. A partir de esa razón se identificó la necesidad de implementar sistemas que brinden la posibilidad de localizar los materiales.

Los sistemas de consulta y recuperación de multimedia han tenido un gran auge en aquellas personas que trabajan en entidades vinculadas a los materiales audiovisuales. Este tipo de aplicaciones, a partir de la demanda que existe sobre las medias, han evolucionado con el propósito de encontrar la información que el usuario necesita. Teniendo en cuenta esas necesidades se han creado varios software con el objetivo de localizar y solicitar materiales audiovisuales. Un ejemplo de estos software son MediaBox, TdtmTarsys de la empresa Tedral y Videoma, los cuales no satisfacen las necesidades actuales de Cuba y la universidad, pues son propietarios y de alto costo en el mercado.

Cuba también se está abriendo paso al desarrollo de las nuevas tecnologías; poco a poco se han ido informatizando la salud, la educación y algunos sectores que son de vital importancia para el desarrollo del país. Gran parte de la información que existe en Cuba, se encuentra almacenada en materiales audiovisuales, pero no existe aún un sistema informático que permita a los usuarios acceder a los mismos para visualizarlos y solicitarlos.

La Universidad de las Ciencias Informáticas (UCI), como puntera en el desarrollo de software en Cuba, es una de las principales guías de informatización de la sociedad cubana. En ella se encuentra el centro de Geoinformática y Señales Digitales (GEYSED), en el mismo existen varios proyectos que se dedican al trabajo especializado con las medias, dentro de ellos el SCCM.

En este proyecto se efectuó un estudio de algunas instituciones que necesitan acceder a las medias para poder visualizarlas y solicitarlas (proceso que permite al usuario requerir un material X en un momento determinado), dentro de las cuales se encuentran el Instituto Cubano de Radio y Televisión (ICRT) y la Dirección de Televisión Universitaria (DTU) de la UCI, la cuales no cuentan con una aplicación que logre satisfacer esas necesidades. El SCCM ha identificado que en estas instituciones existe un gran volumen de almacenamiento de audiovisuales, sin embargo, el proceso para la localización, visualización y solicitud resulta complejo, pues no existe un mecanismo que permita recuperar un material y garantizar el control de la solicitud de los mismos.

Por lo anteriormente planteado se puede definir como **problema a resolver**: ¿Cómo mejorar el proceso de búsqueda y solicitud de materiales audiovisuales digitales en el SCCM?

Para darle solución al problema a resolver se tiene como **objeto de estudio**: Los procesos de consulta y recuperación de materiales audiovisuales. Las prestaciones informáticas de los procesos de consulta y recuperación de materiales audiovisuales del SCCM, recogería el **campo de acción**.

Se define como **idea a defender**: La implementación del módulo de consulta y recuperación de materiales audiovisuales garantizará que los usuarios de la institución donde se despliegue el sistema puedan acceder a esos materiales para poder visualizarlos y solicitarlos. Alcanzando como **objetivo general**: Desarrollar el módulo de consulta y recuperación de materiales audiovisuales.

Como guía para la realización de esta investigación se han propuesto las siguientes **tareas de la investigación**:

- Describir el estado del arte relacionado con los procesos de recuperación y solicitudes de materiales audiovisuales.
- Seleccionar las tecnologías a emplear en el desarrollo del módulo de consulta y recuperación de materiales audiovisuales para el proyecto SCCM.
- Realizar el diseño del módulo de consulta y recuperación de materiales audiovisuales para el proyecto SCCM.
- Implementar el módulo de consulta y recuperación de materiales audiovisuales para el proyecto SCCM.

- Realizar pruebas de caja negra al módulo de consulta y recuperación de materiales audiovisuales para el proyecto SCCM.

Para la realización de estas tareas se emplearon diferentes métodos de la investigación, tanto teóricos como empíricos

Métodos teóricos:

- **Analítico – sintético:** Este método se utilizó con el objetivo de llegar a conocer, mediante el análisis de los diferentes documentos encontrados a lo largo de la investigación, la esencia de los fenómenos relacionados con el proceso de consulta y recuperación de materiales audiovisuales.
- **Análisis histórico – lógico:** Este método permitió realizar un estudio de los sistemas que permiten solicitar y visualizar los materiales audiovisuales y así poder conocer sus antecedentes y evolución a lo largo de la historia.
- **Modelación:** Mediante este método se realizaron los modelos que especifican como se va a desarrollar la aplicación y que permitirán un mejor entendimiento para una posterior implementación.

Método empírico:

Observación: Este método permite investigar el fenómeno en su manifestación externa, sin llegar a la esencia del mismo. El mismo es utilizado con la finalidad de observar detalladamente los requisitos de consulta y recuperación de materiales audiovisuales identificados, con el propósito de desarrollar un sistema acorde con las necesidades.

Principales aportes.

Como aporte práctico de la investigación: Se desarrollará un módulo que permita que los usuarios puedan acceder a los materiales audiovisuales digitalizados para visualizarlos y solicitarlos, con el objetivo que sirva para los productos relacionados con medias que existan en la UCI.

A continuación se muestra la estructura de la investigación que consta de 4 capítulos en los cuales se abordará todo el contenido necesario para concluir la investigación:

Capítulo 1: “Fundamentación Teórica”

Este capítulo está dedicado a los conceptos esenciales relacionados con los procesos de solicitud y visualización de los materiales audiovisuales. Efectúa también un estudio de las soluciones existentes similares al módulo a desarrollar.

Capítulo 2: “Herramientas y Tecnologías para la solución”

Tendencias y tecnologías a utilizar para realizar la consulta y recuperación de materiales audiovisuales. Este capítulo describe las tecnologías, los lenguajes de programación y la metodología de desarrollo que se utilizan para el desarrollo del módulo consulta y recuperación, se explica la selección y uso de estas.

Capítulo 3: “Solución Propuesta”

En este capítulo se refleja a partir del análisis existente del módulo, una descripción del mismo. Se recoge la propuesta del diseño para la aplicación, el cual incluye diagramas web, así como la descripción de las clases. Además se especifican los patrones de arquitectura y diseño utilizados, conjuntamente se analiza el estándar de codificación. Mencionando el diseño de la base de datos e incluyendo el diagrama de clases persistente y el modelo de datos.

Capítulo 4: “Validación de la solución propuesta”

Este capítulo trata sobre las pruebas realizadas al sistema. Se describen los casos de prueba del sistema y se realizan las pruebas de caja negra sobre las interfaces.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Para comprender en qué consiste la presente investigación, es necesario conocer todo lo relacionado con el objeto de estudio de la misma, en este caso: el proceso de consulta y recuperación de materiales audiovisuales. En este capítulo se plantean conceptos y elementos teóricos que constituyen la base para el desarrollo de esta investigación. Además quedan plasmados, algunos de los aspectos teóricos fundamentales relacionados con el objeto de estudio y se analizarán otras aplicaciones existentes en el mundo que tienen entre sus funcionalidades el proceso de localización, visualización y solicitud de medias.

1.2 Términos asociados al problema

1.2.1 Materiales audiovisuales

“Los materiales audiovisuales son un conjunto de imágenes en movimiento, que a su vez estas imágenes son perceptibles por el oído y por la vista y se caracterizan por el hecho de que:

- su grabación, transmisión, percepción y comprensión requieren habitualmente un dispositivo tecnológico.
- el contenido visual y sonoro tiene una duración lineal.
- el objetivo es la comunicación de ese contenido, no la utilización de la tecnología con otros fines” (1).

Se hace necesario explicar el significado de **materiales audiovisuales** ya que a estos términos se les atribuyen definiciones muy diversas. Se puede plantear según estudios realizados que **audiovisual** se refiere “a las imágenes en movimiento y a los sonidos grabados, registrados en cinta magnética, disco o cualquier otro medio actualmente conocido o por inventar” (2).

”La definición de **materiales audiovisuales** procura abarcar el máximo de formas y formato...Las imágenes en movimiento constituyen la clásica forma de material audiovisual y la forma principal explícitamente mencionada en la Recomendación de la UNESCO de 1980... incluyen grabaciones sonoras” (2).

Se define entonces que un **material audiovisual** es uno de los términos que se utilizan para referirse de forma más generalizada a los **audiovisuales**, o sea, a las imágenes en movimiento, sonidos grabados o ambos a la vez.

1.2.2 Sistema de Recuperación de Información

“Proceso donde se accede a una información previamente almacenada, mediante herramientas informáticas que permiten establecer ecuaciones de búsqueda específicas. Dicha información ha debido de ser estructurada previamente a su almacenamiento” (3).

“Los sistemas de recuperación de información, permiten el almacenamiento, procesamiento, recuperación y difusión de la información, sea esta de tipo texto, imágenes, audio, vídeo o multimedia.” (4)

Los **sistemas de recuperación de información** se pueden definir como herramientas que permiten procesar las consultas realizadas por los usuarios para satisfacer las necesidades de información de los mismos y que están previamente almacenadas.

1.2.3 Recuperación de materiales audiovisuales

El proceso de búsqueda y recuperación de información ha sido analizado como un proceso de resolución de problemas en el que intervienen varios factores. Por una parte se tiene un usuario que busca información porque tiene un problema o necesidad informativa. Otro factor importante es el sistema de búsqueda, donde se incluyen las técnicas para la recuperación de la información, documentos, representaciones de los documentos y la organización de esas representaciones. El conjunto de resultados que se obtienen del sistema y que debe satisfacer la necesidad de información del usuario se incluye dentro de los elementos. Los elementos inicialmente planteados influyen en el resultado final de recuperación o búsqueda e interactúan entre ellos, logrando después que se pueda visualizar y solicitar dicha información (6).

La recuperación de los materiales audiovisuales implica el proceso inicialmente descrito, todo se basa en la necesidad del usuario de encontrar el material que desea.

1.2.4 Solicitud de Materiales Audiovisuales

“**Solicitud:** Pretender, pedir o buscar algo con diligencia y cuidado” (5).

Una **solicitud de un material audiovisual** es una petición u orden que se realiza luego de que un determinado usuario encuentra mediante la realización de la búsqueda, la media que desea, luego queda controlado que usuario y en qué momento realizó la solicitud, es decir, se almacenan los datos del usuario y los materiales solicitados.

1.3 Proceso de consulta y recuperación de materiales audiovisuales

1.3.1 Descripción General

La recuperación de información es un tema, que en la práctica está relacionado con muchas disciplinas, (las ciencias de la información, ciencias de la documentación, la informática, la lingüística, entre otras) las cuales de una u otra forma se enfocan en facilitar que el documento que un determinado usuario está buscando, le sea entregado y que este logre satisfacer la necesidad de información que el usuario presentó. La importancia de la recuperación de información es brindar la posibilidad de que la misma sea accesible, o sea, que no existan barreras que impidan su difusión y consumo, además debe permitir que las necesidades de información sean resueltas de la mejor manera posible (7).

Son muchos los enfoques que se han experimentado para abordar el objetivo esencial de la Recuperación de información (RI), que se refiere a la recuperación de todos los documentos relevantes y al mismo tiempo rechazar todos los documentos irrelevantes ante la formulación de una consulta por parte del usuario. Para esto existen los modelos de recuperación de la información, estos se encargan de una forma u otra de definir cuáles son las premisas que se tienen que tener en cuenta para determinar si un determinado documento es relevante o no a una necesidad de información. En otras palabras, un modelo determina cómo se realizará la comparación entre la consulta y los documentos para calcular una medida de similitud que permita determinar la relevancia (o no) y el ranking.

1.3.2 Descripción actual del dominio del problema

En la actualidad existe una creciente demanda de los usuarios para poder consultar, recuperar y solicitar información de los materiales audiovisuales, la UCI cuenta con la Facultad 6, dentro de la cual existen diferentes proyectos que se dedican específicamente al trabajo detallado con los materiales audiovisuales. El proyecto SCCM se encarga de gestionar materiales audiovisuales y posee diversas funcionalidades dentro de las que se puede mencionar la codificación, catalogación y búsquedas de los mismos. SCCM pretende desarrollar un software para la recuperación y catalogación de medias, que sea capaz de permitirles a los usuarios de cualquier institución acceder a los materiales

audiovisuales para visualizarlos y solicitarlos.

La aplicación debe brindar la posibilidad de hacer una búsqueda básica, por contenido o avanzada a preferencia del usuario para obtener el material audiovisual que desea. Una vez realizada la búsqueda se obtendrán un conjunto de resultados de acuerdo a la necesidad del mismo, y el sistema deberá brindar a los usuarios una vista previa de los resultados para que este verifique cual es la media que desea solicitar y visualizar. El módulo de consulta y recuperación de materiales audiovisuales debe ser capaz de reproducir las medias que el usuario haya seleccionado y también de solicitarlas.

1.3.3 Situación Problemática

El desarrollo del mundo actual y de las nuevas tecnologías ha provocado un gran problema de recuperación de información que surge de la creación no controlada de documentos, estos problemas también están presentes en los materiales audiovisuales. La demanda de los materiales audiovisuales ha incitado a varias compañías a dedicarse a la gestión de los mismos. La principal problemática consiste en que existen empresas que son privativas y cobran un alto precio por brindar este tipo de servicios.

Con el estudio realizado a algunas instituciones como Venezolana de Televisión (VTV), el ICRT, y la DTU de la universidad se evidenció que además de que el proceso de localización de la información es engorroso, no existe un sistema que permita visualizar y solicitar los materiales. El departamento de Señales Digitales de la Facultad 6 en la UCI se encuentra desarrollando un software con el objetivo de mejorar el proceso de localización, visualización y solicitud de los audiovisuales en la empresa donde sea desplegado el sistema.

1.4 Análisis de otras soluciones existentes

En la actualidad existen disímiles software que se encargan de la búsqueda y recuperación e materiales audiovisuales y que permiten además visualizarlos y solicitarlos. A continuación se describen las características de algunos de ellos, sus ventajas y algunos inconvenientes con el fin de optimizar la solución que se pretende realizar.

1.4.1 YouTube

YouTube mantiene una logística que permite localizar cualquier vídeo por medio de las etiquetas de metadato, títulos y descripciones que los usuarios asignan a sus vídeos, pero cuyo uso está restringido únicamente a los vídeos alojados en YouTube. Todos los vídeos deben ocupar un espacio en disco no

mayor a 2 GB y deben tener una duración menor o igual a 15 minutos, con una tolerancia extra de 59 segundos de grabación, es decir, una duración total de 15:59 segundos. La totalidad de los vídeos son convertidos a resoluciones de 320x240 y 480x360 píxeles, y a 30 fotogramas por segundo. A Los usuarios que tienen un buen historial de cumplimiento de **reglamento de la comunidad** del sitio se les ofrecerá la posibilidad de subir vídeos de duración ilimitada, lo que requiere la verificación de la cuenta, normalmente a través de un teléfono móvil.



FIGURA 1: IMAGEN DEL LOGO DE YOU TUBE (8)

A pesar de que todos los vídeos se muestran predeterminadamente en la resolución menor, los usuarios pueden reproducirlos en calidad alta ajustando sus preferencias. Al igual que el formato de HQ (alta calidad), el formato HD (alta definición) solamente se visualizará si existe la versión. Recientemente se añadieron las resoluciones 2k y 4k o Ultra alta definición (hasta 7,680 x 4,320 píxeles) cuya resolución es superior a la Alta definición y al sistema PAL (8).

1.4.2 DailyMotion

Dailymotion es un servicio de alojamiento de videos en Internet, fue puesto en marcha casi al mismo tiempo que el conocido servicio de Youtube.com en el 2005. Tiene varios tipos de usuario, que permiten distintos niveles de participación y posibilidades de la cuenta. Dentro de sus principales desventajas están que no permiten videos comerciales, la duración permitida es de 20 min y el peso permitido es de 2 GB (9). Dailymotion ofrece una aplicación que puede ser utilizada dentro de la red social Facebook.com, permite realizar búsquedas de videos dentro del portal de Facebook así como compartir videos con otros usuarios de dicha red. Dispone de una aplicación exclusivamente para dispositivos iPhone.



FIGURA 2: LOGO DE DAILYMOTION (10)

Este servicio también tiene disponible una versión para usuarios menores de edad aunque está condicionada a ciertos requerimientos técnicos resulta una propuesta interesante. No pasa por alto el tema de los derechos de autor, ha implementado un sistema para la detección de la violación de estos derechos aunque los videos subidos a su servidor difieran en cuanto al desarrollo de los mismos los cuales están más orientados a la emisión de videos profesionales y cortos, este punto no representa una desventaja en cuanto a la gran diversidad de videos que se pueden encontrar en Dailymotion (10).

1.4.3 Metacafe

Es un sitio web para compartir vídeos digitales en Internet. Ofrece videos generados por el usuario, videos de humor, videos de Internet y anuncios virales, así como deportes y noticias relacionadas. La aplicación de escritorio es capaz de descargar los videos de alta calidad a la computadora local mientras ésta está ociosa, está destinada principalmente a los usuarios que son “video adictos” y descargan muchos videos cada semana.

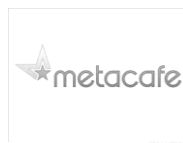


FIGURA 3: LOGO DE METACAFE (11)

Algunas de las otras características en el sitio incluyen juegos en flash y clips de sonido. Metacafe tiene una política anti-pornografía aun cuando allí existe una categoría de adultos. Sin embargo, permite el contenido con clasificación R según lo definido por el sistema de clasificación de la película. La duración de los vídeos es corta. Este sitio no permite videos comerciales. El tamaño de máximo de un video puede ser de 2 Gb (11).

1.4.4 Mediabox

MEDIABOX es un producto privativo sobre la plataforma de Windows que facilita la catalogación de materiales audiovisuales desde diferentes tipos de fuentes y diferentes formatos, así como la recuperación de los mismos. Trabaja en red y almacena los datos en un Servidor Central (12).



FIGURA 4: LOGO DE MEDIABOX (12)

Este producto permite la recuperación de vídeo desde sistemas robotizados de almacenamiento en cinta. Permite también la búsqueda por palabras claves y por similitud. Se puede catalogar mientras se codifica, o bien una vez finalizado el almacenamiento. El sistema crea metadatos de forma automática cada vez que se crea una plantilla de catalogación (12). La principal desventaja de este producto es su dependencia con los demás software privativos, otra de sus desventajas es que no permite la edición de una media audiovisual.

1.4.5 TdtmTarsys, de la empresa Tedral

TdtmTarsys es una herramienta orientada al usuario que asegura que cualquier contenido, donde quiera que esté almacenado, pueda ser consultado por los usuarios como un único archivo. Se dedica a la gestión de materiales audiovisuales. Permite la búsqueda de medias y la catalogación de materiales audiovisuales con información que ayude a su localización.



FIGURA 5: LOGO DE TEDIAL

También posibilita la descarga de estos materiales para su posible distribución y la edición de dichos materiales. Contiene un reproductor para la reproducción de dichas medias. Cada sistema de archivos Tarsys puede conectarse a otros con objeto de intercambiar media, metadatos y archivos. La principal desventaja de este producto es que es desarrollado en plataformas no libres y es muy costoso.

1.4.6 Videoma

Producto privativo desarrollado sobre la plataforma de Windows. Fue diseñado para trabajar en red a través de múltiples puestos, desde los que se pueden realizar consultas del material almacenado en un servidor central (13). Ofrece un amplio abanico de soluciones en la gestión de archivos. Videoma es un sistema modular con gran flexibilidad para agregar nuevas funcionalidades a la plataforma o para crear soluciones a medida (14).



FIGURA 6: LOGO DE VIDEOMA (14)

Contiene un módulo básico de gestión y recuperación de contenido que es el motor principal de la plataforma Videoma. Controla todos los procesos implicados con la administración, permisos de usuarios, ingestas, catalogación, edición, búsquedas de contenido y otros (14). Una de sus principales desventajas es que no permite la catalogación a los usuarios. Esta catalogación la realiza solo el personal autorizado trabajando directamente con el servidor de medias. Además no permite la edición de las medias audiovisuales sino que se comunica con editores externos como el Adobe Premiere. Esto trae consigo un alto costo además que la herramienta es privativa y su adquisición es muy costosa.

Al realizar el análisis de las soluciones existentes creadas en el mundo para la consulta y recuperación de materiales audiovisuales, se evidenció que las mismas no pueden ser empleadas por el departamento de Señales Digitales, pues están desarrolladas en plataformas privativas y son de alto costo. Además ninguna de ellas permite ser añadida como parte del proyecto SCCM.

1.5 Conclusiones Parciales

A través del análisis de los conceptos asociados se garantizó un mejor entendimiento del tema a tratar. Por otra parte, según el estudio de otras soluciones, se identificó que todas son desarrolladas con herramientas propietarias y de un alto costo en el mercado, pues la utilización de las mismas va en contra de la soberanía tecnológica por la que aboga el país. Con este estudio además se demuestra que es necesario el desarrollo del módulo de consulta y recuperación para mejorar este proceso en varias instituciones del país.

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS PARA LA SOLUCIÓN

2.1 Introducción

Con el progreso constante de la informática se han mejorado las herramientas y tecnologías para el desarrollo de software, constituyendo instrumentos fundamentales para lograr productos con calidad y desarrollados en tiempo récord. Este capítulo describe las herramientas y tecnologías que se van a utilizar para realizar el módulo consulta y recuperación de materiales audiovisuales, los lenguajes de programación y la metodología de desarrollo, además se explica su selección y uso.

2.2 Metodología de desarrollo de software

Una metodología de desarrollo de software no es más que una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software con la finalidad de garantizar la eficacia y la eficiencia en el proceso de generación de software. Los riesgos a afrontar y los controles a establecer varían en función de las diferentes etapas del ciclo de vida de desarrollo que defina cada metodología en particular. Pueden estar clasificadas de dos maneras, como metodologías ágiles o robustas. Una pregunta fundamental en el desarrollo de cualquier software es *¿qué metodología utilizar?* pues todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, no se obtiene el resultado deseado y el cliente no queda satisfecho.

Las metodologías pesadas se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son necesarias cuanto mayor es el proyecto que se pretende realizar, donde una gran organización es requerida. Una de las metodologías más conocidas y utilizadas es la Metodología RUP (Rational Unified Process), puede ser usada como robusta o ágil en dependencia del proyecto que se desee realizar. A continuación se expondrán algunas características de la misma.

2.2.1 RUP (Proceso Unificado de Desarrollo o Rational Unified Process)

Como metodología, RUP es un proceso que define claramente quien, cómo, cuándo y qué debe hacerse. Este aporta herramientas como los casos de uso, que definen los requisitos. Permite la ejecución iterativa del proyecto y del control de riesgos. RUP es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos (15).

Descripción general

- El RUP es un proceso de ingeniería de software.
- Utiliza el paradigma de orientación a objetos para su descripción.
- Es un marco de proceso configurable para satisfacer necesidades específicas.
- Implementa buenas prácticas de desarrollo de software.

Esta metodología divide en 4 fases el desarrollo, estas fases son:

- **Inicio:** El objetivo de esta fase es determinar la visión del proyecto.
- **Elaboración:** El objetivo de esta fase es determinar la arquitectura más óptima.
- **Construcción:** El objetivo de esta fase es obtener la capacidad operacional inicial (construir el software).
- **Transición:** El objetivo de esta fase es obtener el reléase del proyecto, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor menor hincapié en las distintas actividades.

RUP divide el proceso de desarrollo en los siguientes flujos de trabajo:

- **Modelamiento del negocio:** Este flujo se encarga de describir los procesos del negocio.
- **Requerimientos:** Define qué es lo que el sistema debe hacer.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas.

- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba:** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce release del producto y realiza actividades para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto.

Flujos de trabajo de RUP

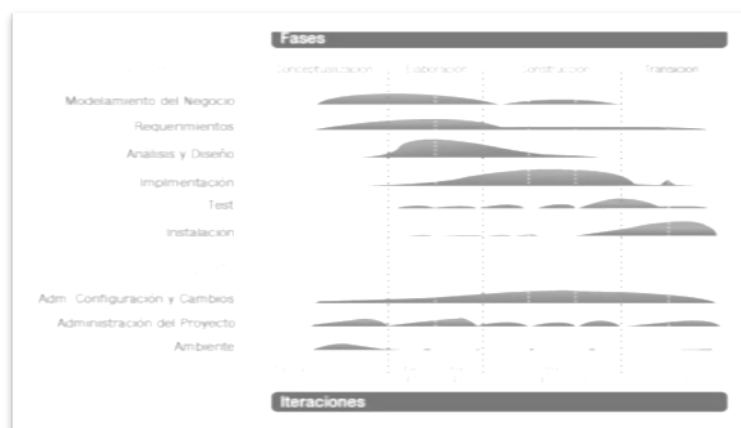


FIGURA 7: FLUJO DE TRABAJO DE LA METODOLOGÍA RUP.

Características de RUP

1. **Guiado por casos de uso:** ya que el desarrollo está dirigido a satisfacer las necesidades de los usuarios del sistema expresadas en casos de uso. Los casos de uso capturan requisitos funcionales y representan piezas de funcionalidad que brindan un resultado de valor al usuario.

- 2. Centrado en la Arquitectura:** porque comprende los aspectos estáticos y dinámicos más importantes del sistema. La arquitectura surge de las necesidades de la empresa, como las perciben los usuarios y los inversores y se refleja en los CU.
- 3. Iterativo e Incremental:** pues el trabajo se divide en piezas pequeñas o mini proyectos; cada uno proveyendo un subproducto incremental.

Después de realizar este análisis se pudo observar que RUP es muy utilizada para la documentación y realización de software orientados a objetos. El equipo de desarrollo cuenta con experiencia en el uso de esta metodología, aspecto a su favor pues no se requiere de tiempo para estudiar la misma. Propone varias iteraciones en el proceso de desarrollo del sistema posibilitando la corrección de errores y mejoras del software a medida que avanza el proyecto, lo que permite a los desarrolladores poder realizar un producto con gran calidad. Brinda una fuerte y abundante documentación, permitiendo un futuro entendimiento de cada componente del producto. Esto es fundamental para la aplicación pues con esta metodología el sistema se va desarrollando y documentando al mismo tiempo, esto es importante pues, si algún miembro del equipo no puede seguir con el trabajo el que ocupe su lugar pueda guiarse y conozca qué fue lo que se realizó hasta la fecha.

Esta metodología ayuda a construir un software de alta calidad, desarrollado en el tiempo planificado. Es necesario enfocarse en trabajar de forma organizada, donde se controle y documente todo lo relacionado con el proyecto en cuestión y puedan eliminarse los riesgos que podrían presentarse durante el desarrollo del mismo, lo cual no podría lograrse sin el empleo de una metodología que se adapte a las características propias del software que se esté desarrollando.

2.3 Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado (UML) surge con el objetivo de comunicar ideas a otros desarrolladores y para servir de apoyo en los procesos de análisis de un problema. UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente, de diseño (16).

UML es un lenguaje para visualizar, especificar, construir, y documentar los artefactos que se crean durante el proceso de desarrollo. Es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos, además es gráfico, puede ser usado en todas las fases de desarrollo de software y permite representar los sistemas con varios modelos parciales, lo que facilita su entendimiento y la comunicación.

La representación en UML de un software está formada por las 4+1 vistas o modelos parciales separados, relacionados entre sí, estas vistas son:

- Vista de casos de uso.
- Vista lógica.
- Vista de procesos.
- Vista de implementación.
- Vista de despliegue.

El UML es una notación estándar para el análisis y diseño de negocios y sistemas informáticos. Independiente de la arquitectura o el lenguaje que se vayan a seleccionar (o que hayan sido seleccionados) para la realización. Incorpora las mejores prácticas a nivel internacional. Cuenta con un amplio apoyo entre empresas e instituciones. Está abierto a nuevos avances, y no es propiedad de una sola compañía (17).

RUP utiliza para el proceso de desarrollo del sistema en cuestión como lenguaje de modelado a UML, por lo que se elige UML para modelar el análisis de la herramienta de personalización propuesta, ya que este constituye uno de los más utilizados para el análisis, implementación y documentación de sistemas orientados a objetos. Posee más características visuales que programáticas permitiendo a los analistas, diseñadores, programadores y especialistas de área a intercomunicarse con mayor facilidad durante el proceso de desarrollo.

Por las características expuestas anteriormente se decide utilizar el lenguaje UML durante el ciclo de vida del proyecto, pues brinda la posibilidad a los analistas de definir las características principales del sistema, describe las decisiones y todos los pasos necesarios representados gráficamente por modelos precisos y completos que proporcionan una mejor comprensión para los desarrolladores. Es un lenguaje que puede ser implementado en cualquier lenguaje orientado a objeto y permite tanto a los clientes como a los desarrolladores saber a ciencia cierta el alcance y la factibilidad que puede tener el producto final.

2.4 Herramientas CASE

Las herramientas (**C**omputer **A**ided **S**oftware **E**ngineering¹) CASE son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Estas herramientas ayudan en el ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. La mejor razón para la creación de estas herramientas fue el incremento en la velocidad de desarrollo de los sistemas. Por esto, las compañías pudieron desarrollar sistemas sin encarar el problema de tener cambios en las necesidades del negocio, antes de finalizar el proceso de desarrollo. (18)

2.4.1 Visual Paradigm

Es una herramienta colaborativa, en otras palabras, soporta múltiples usuarios trabajando sobre el mismo proyecto, constituye una herramienta profesional para el modelado UML que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Posibilita el modelado de base de datos, requerimientos, proceso de negocio, permite realizar todo tipo de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. A continuación se mencionan algunas ventajas que ofrece para el proyecto SCCM:

- Multiplataforma (Incluyendo GNU/Linux).
- Diseño centrado en casos de uso y enfocado al negocio, permitiendo generar un diseño de software de gran calidad.
- Permite realizar ingeniería directa e inversa.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo facilitando la comunicación.
- Permite el control de versiones.
- Modelo y código permanecen sincronizados en todo el ciclo de desarrollo.
- Permite integrarse con diferentes IDE.

¹Significado de CASE en español: Ingeniería de Software Asistida por Ordenador

Visual Paradigm es la herramienta seleccionada, pues además de emplear UML para el modelado, es la herramienta por excelencia para ser utilizada en un ambiente de software libre. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual. Es muy sencillo de usar, fácil de instalar y actualizar. Otra de las ventajas que posee es que se adecua al entorno de desarrollo permitiéndoles a los diseñadores, analistas y a todo aquel que trabaje con el mismo, una mayor organización y claridad en el trabajo. Esta herramienta es bastante común y muy utilizada a nivel mundial por la variedad de funciones que permite realizar, lo que posibilita su utilización para la evolución del proyecto SCCM.

2.5 Tendencias y tecnologías actuales

En la actualidad cuando se abordan temas sobre tendencias y tecnologías actuales a utilizar para el desarrollo de algún software o algún componente de software, es importante resaltar las relacionadas con el entorno web. Esto se debe a que dicho entorno brinda un grupo de ventajas considerables en cuanto a la disminución de recursos y seguridad informática.

Se puede destacar también que el entorno web permite tener numerosos clientes independientes de las tecnologías. El capital humano necesario para el mantenimiento constante del software disminuye, ya que no es necesaria la instalación del mismo en distintas estaciones, solo se instala en una computadora determinada y se accede a este mediante un navegador web. Debido a esto, la información quedaría almacenada en un solo ordenador y la administración sería de forma centralizada.

Por esas características en los últimos años ha ocurrido un auge considerable en el desarrollo de sitios web para la gestión de información. Todos estos aspectos posibilitaron que el proyecto SCCM optara por el desarrollo del producto utilizando tecnologías web.

2.5.1 Modelo Cliente-Servidor

En el mundo de TCP/IP las comunicaciones entre computadoras se rigen básicamente por lo que se llama modelo Cliente-Servidor, éste es un modelo que intenta proveer usabilidad, flexibilidad, interoperabilidad y escalabilidad en las comunicaciones. El término Cliente-Servidor fue usado por primera vez en 1980 para referirse a PC's en red.

Desde el punto de vista funcional, se puede definir la computación Cliente-Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. En el modelo cliente servidor, el cliente envía un

mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio) (Ver Figura 8).

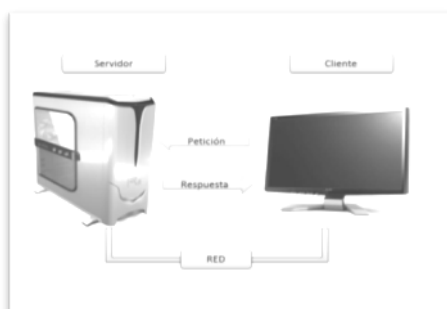


FIGURA 8: MODELO ARQUITECTÓNICO CLIENTE-SERVIDOR. (19)

Las características básicas de la arquitectura Cliente-Servidor:

- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
 - El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.
 - Un servidor ofrece servicio a variados clientes de forma concurrente.
 - Si ocurrieran modificaciones en el servidor estas afectarían en ninguna o poca escala al cliente.
- (19)

Para el desarrollo del sistema una de las ventajas fundamentales que brinda esta tecnología es la de ser una infraestructura versátil, modular y basada en mensajes que mejora la portabilidad, la interoperabilidad y la escalabilidad de los sistemas. Además con el uso de este modelo aumenta la interactividad y se facilita el mantenimiento de las aplicaciones, por estas razones es que se decidió utilizarlo en el proyecto SCCM.

2.6 Lenguaje de Programación

Un lenguaje de programación es un conjunto de sintaxis y reglas semánticas que definen los programas del computador (20). Es una técnica estándar de comunicación para entregarle instrucciones al computador. Un lenguaje le da la capacidad al programador de especificarle al

computador, qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano.

Un programa escrito en un lenguaje de programación necesita pasar por un proceso de compilación, interpretación o intermedio, es decir, ser traducido al lenguaje de máquina para que pueda ser ejecutado por el ordenador.

Los lenguajes de programación Web han ido surgiendo según las necesidades de las plataformas, intentando facilitar el trabajo a los desarrolladores de aplicaciones. Se clasifican en lenguajes del lado del cliente y lenguajes del lado servidor. (21)

2.6.1 Lenguaje del Lado del Cliente

Los lenguajes del lado del cliente son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre-tratamiento. Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio.

2.6.1.1 HTML (HyperText Markup Language o Lenguaje de Marcado de Hipertexto)

El HTML no es más que una aplicación del SGML (Standard Generalized Markup Language), un sistema para definir tipos de documentos estructurados y lenguajes de marcas para representar esos mismos documentos. El HTML es el lenguaje de marcas de texto utilizado en la WWW (World Wide Web). Es muy sencillo y permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido) (22). Este lenguaje está constituido de elementos que el navegador interpreta y los despliega en la pantalla de acuerdo a su objetivo.

2.6.1.2 JAVASCRIPT

Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario (23).

JavaScript es una de las múltiples aplicaciones que han surgido para extender las capacidades del Lenguaje HTML, es un lenguaje script orientado a documento. Este lenguaje permite variar

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS PARA LA SOLUCIÓN

dinámicamente el contenido del documento, modificar el comportamiento normal del navegador, validar formularios y realizar pequeños trucos visuales (24).

2.6.1.3 CSS (Cascading Style Sheets, Hojas de estilo en Cascada)

CSS son las siglas de Cascading Style Sheets, en español Hojas de estilo en Cascada. CSS, es una tecnología que permite crear páginas web de una manera más exacta. CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML, como por ejemplo: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos y posición de cada elemento dentro de la página (25). CSS se utiliza además para dar estilo a documentos HTML y XML, separando el contenido de la presentación (26).

2.6.1.4 AJAX (Asynchronous JavaScript And XML (JavaScript asíncrono y XML))

El término AJAX es un acrónimo de **Asynchronous JavaScript + XML**, que se puede traducir como "JavaScript asíncrono + XML".

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

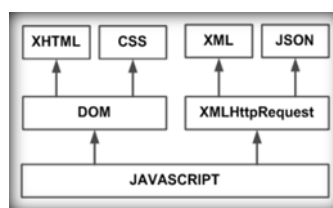


FIGURA 9: TECNOLOGÍAS AGRUPADAS BAJO EL CONCEPTO DE AJAX. (27)

AJAX no es una tecnología, sino la unión de varias tecnologías. La principal ventaja de las aplicaciones realizadas con estas tecnologías es que se actualizan de forma dinámica, sirviéndose casi siempre del lenguaje XML, y no mediante el 'refresco' de páginas enteras que se muestran al usuario cada vez que éste realiza alguna acción. A continuación se muestran algunas características de AJAX que son una ventaja para el sistema:

- Las aplicaciones son más interactivas, responden a las interacciones del usuario más rápidamente, al estilo aplicaciones de escritorio.
- Estas aplicaciones tienen un aspecto (look and feel) muy similar a las aplicaciones de escritorio tradicionales sin depender de plugins o características específicas de los navegadores.
- Se reduce el tamaño de la información intercambiada.
- Muchas micro-peticiones, pero el flujo de datos global es inferior.
- Se libera de procesamiento a la parte servidora (se realiza en la parte cliente)
- AJAX actualiza porciones de la página en vez de la página completa. (27)

2.6.2 Lenguaje del Lado del Servidor

Lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él (28). Un lenguaje del lado del servidor es independiente del navegador utilizado, no necesitará plug-in especiales para visualizar correctamente cualquier página. Además, el código de los script puede ocultarse al terminal cliente, que sólo verá el código html terminado. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente.

2.6.2.1 Personal Home Page (PHP)

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Se caracteriza por ser un lenguaje muy rápido, es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI² separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor.

Las ventajas que presenta este lenguaje para el desarrollo del módulo son:

- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, , Oracle, MS SQL Server y PostgreSQL que es la utilizada en el proyecto.

²Common Gateway Interface, CGI: es un protocolo genérico que permite extender las capacidades de HTTP.

- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

2.7 Framework

Un framework es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta. (29)

Este no es más que una abstracción de código común que provee funcionalidades genéricas que pueden ser utilizadas para desarrollar aplicaciones de manera rápida, fácil, modular y sencilla, ahorrando tiempo y esfuerzo. Entonces, un framework es concreto y también “incompleto”. Concreto porque es, desde un punto de vista simple, un conjunto de componentes; incompleto, porque por sí mismos no pueden ser utilizados, ya que guían a la solución de problemas de programación recurrentes, pero, por lo general, no son la solución específica completa. Además ofrece una guía arquitectónica partiendo el diseño en clases abstractas y definiendo sus responsabilidades y sus colaboraciones. Un desarrollador personaliza el marco de trabajo para una aplicación particular mediante herencia y composición de instancias de las clases.

2.7.1 Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se deberá inventar la rueda cada vez que se crea una nueva aplicación web.

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS PARA LA SOLUCIÓN

Symfony está desarrollado completamente con PHP 5, es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. Añade una nueva capa por encima de PHP y proporciona herramientas que simplifican el desarrollo de las aplicaciones web complejas.

La capa de internacionalización que incluye symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos. Los formularios incluyen validación automatizada y relleno automático de datos. La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor. La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario. Las interacciones con Ajax son muy fáciles de implementar mediante los helpers que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código. (30)

Symfony emplea el patrón modelo-vista-controladora (MVC) con el objetivo de organizar las distintas partes que componen una aplicación web, separándola en tres capas principalmente: el modelo, la vista y la controladora. El modelo es el encargado del dominio de toda la información del sitio, de darle significado a dicha información, acceder a los datos de la aplicación, además de la comunicación con la base de datos y la ejecución de consultas. La vista es la que se encarga de la interfaz que se le muestra al usuario, presentando los datos del modelo en un formato adecuado para la interacción, además decide cómo mostrar los datos suministrados por la controladora. La controladora es la que decide qué se va a mostrar y en qué momento, además es el que dirige todo el proceso de gestión y responde a las acciones de los usuarios.

Por las razones expuestas anteriormente se considera symfony la mejor opción para el desarrollo del proyecto ya que la calidad que brinda al código fuente y la cantidad de documentación disponible es favorable para desarrollar el mismo. El número de sus colaboradores ha crecido rápidamente y su labor va desde proponer parches y mejoras hasta detectar errores de la documentación.

2.7.2 DOJO

Dojo es un framework JavaScript publicado como software de código abierto. Este conjunto de herramientas JavaScript proporciona muchos componentes para desarrollar aplicaciones ricas de Internet. Se puede utilizar DojoToolkit para desarrollar aplicaciones web dinámicas. Dojo brinda a los desarrolladores de páginas web un grupo de funcionalidades importantes. Dichas funcionalidades, de

usarse correctamente les da embellecimiento y dinamismo a las páginas que programan. Permite mejorar el diseño y la usabilidad de la página web además de ser compatible en la mayoría de los navegadores.

Entre algunas de sus características está la manipulación de DOM (Document Object Model), animaciones y contiene Apis para el trabajo con Ajax. Dojo contiene un grupo de componentes reutilizables (widgets), permite manejo de eventos y soporte grafico para vectores en segunda y tercera dimensión. Dojo ya ha inventado las ruedas para los programadores y estos sólo tienen que utilizar la API de Dojo en su aplicación. (31)

2.8 Entorno de Desarrollo Integrado (IDE)

Un entorno de desarrollo integrado (Integrated Development Environment o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos.

2.8.1 Netbeans

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Tiene soporte para crear interfaces gráficas de forma visual, crear aplicaciones para móviles, desarrollar aplicaciones web y además estas funcionalidades son ampliables mediante instalación de paquetes adicionales. Existe además un número importante de módulos para extender el NetBeans IDE (32).

Presenta una fácil integración con symfony y zenFramework que agiliza la elaboración de proyectos. Elimina el uso de los comandos de consola que se necesitaban para instalar symfony, esto se debe a que presenta una interfaz visual para desarrollar dicha instalación. Posee un fuerte editor de código fuente, sobre todo el de PHP, ya que provee todas las ventajas que presenta PHP 5.3, además de ser robusto y ágil.

NetBeans es gratuito y de código abierto. Presenta todas las herramientas para no solo elaborar aplicaciones en entorno web sino también de escritorio, móviles y empresas. Incorpora diferentes lenguajes como C/C++, Java, Ruby y sus variantes tales como JRuby y Ruby onRails. Todas esas características posibilitaron que el proyecto SCCM escogiera a NetBeans7.0 como IDE de desarrollo.

2.9 Sistemas Gestores de Bases de Datos (SGBD)

Un sistema gestor de bases de datos (SGBD) o DBMA³ es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos (33). En estos sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos.

Hoy en día existen numerosos SGBD que tienen características propias, no obstante, todos deben poseer los siguientes aspectos: abstracción de la información, la independencia de los datos, redundancia mínima, consistencia en los datos, seguridad, integridad, respaldo y recuperación, tiempo de respuesta y control de concurrencia.

2.9.1 Postgresql

PostgreSQL es un servidor de bases de datos relacionales orientadas a objetos, de software libre bajo licencia BSD⁴. Al ser un proyecto Open Source, el desarrollo no se lo adjudica a una sola persona sino más bien a toda una comunidad de desarrolladores, los mismos se denominan PGDG (PostgreSQL Global Development Group). Una de las características que comparte con otros motores de bases de datos es el hecho de ser multiplataforma. El mismo posee casi 30 años de desarrollo (34).

La utilización de este SGBD en el proyecto SCCM obedece al propósito de la elaboración de un sistema con gran robustez y alto nivel de escalabilidad. Trabajar con este gestor es factible ya que es muy rápido, tiene buenas utilidades de administración, es confiable y fácil de usar tanto para volúmenes de datos grandes como pequeños, sin límites en los tamaños de registros. Consecuentemente tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. PostgreSQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad, lo hacen altamente conveniente para acceder a bases de datos en internet ya que tiene mejor control de acceso a usuarios. Al mismo tiempo, los lenguajes que se proponen para el desarrollo de la plataforma logran una rica interacción con este gestor lo que resulta

³DataBase Management System

⁴Licencia de software otorgada principalmente para los sistemas (*Berkeley Software Distribution*) BSD. Es una licencia de software libre, tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público, permite el uso del código fuente en software no libre.

compatible y de mucho beneficio a la hora de diseñar de forma rápida y eficaz aplicaciones dirigidas a bases de datos.

2.10 Conclusiones Parciales

- La caracterización de las tecnologías y herramientas propuestas para la elaboración de la aplicación permitió conocer las ventajas que ofrecen las mismas para el desarrollo del módulo web de consulta y recuperación de materiales audiovisuales.
- De esta manera se está en condiciones de realizar el desarrollo del módulo, utilizando herramientas adecuadas para dar cumplimiento a los requisitos establecidos en el sistema.
- La selección se realizó teniendo en cuenta que las tecnologías escogidas fueran bajo licencias libres, para seguir con el principio de soberanía tecnológica por la que aboga el país.

CAPÍTULO 3: SOLUCIÓN PROPUESTA

3.1 Introducción

En este capítulo se recoge el diseño del módulo de consulta y recuperación de materiales audiovisuales que se propone, el cual incluye diagramas web, así como la descripción de las clases. Además se especifican los patrones de diseño utilizados y los estándares de codificación manejados en la aplicación. También se hace alusión a la base de datos incluyendo el diagrama de clases persistente y el modelo de datos.

3.2 Patrones Arquitectónicos.

Los patrones arquitectónicos definen la estructura general del software, indican las relaciones entre los subsistemas y los componentes y definen las reglas para especificar las relaciones entre los elementos (Clases, paquetes, subsistemas) de la arquitectura (35).

La arquitectura es el conjunto de decisiones significativas sobre la organización de un sistema software. La misma no está relacionada sólo con la estructura y el comportamiento, sino también con el uso, la funcionalidad, el rendimiento, la flexibilidad, la reutilización, la facilidad de comprensión, las restricciones y compromisos económicos y tecnológicos y con la estética (36).

Un estilo arquitectónico es una transformación impuesta al diseño de todo un sistema. Algunos de estos estilos arquitectónicos son: Arquitectura centrada en datos, Arquitectura de flujo de trabajo, Arquitectura de llamada y retorno, Arquitectura orientada a objetos (35).

Para modelar la arquitectura del sistema se utilizó el patrón arquitectónico Modelo-Vista- Controlador (MVC) que forma parte del estilo arquitectónico de llamada de retorno.

3.2.1 Modelo-Vista-Controlador

Un patrón utilizado para la realización de este sistema es el Modelo-Vista-Controlador, es un patrón que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Este patrón provee simplicidad en el mantenimiento del sistema.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, e-mail). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (37)

Con la utilización de esta arquitectura se garantiza que el modelo, la vista y la controladora se desarrollen de forma independiente, permitiendo realizar modificaciones en cada una de estas partes sin afectar las demás. También se puede mencionar que esta arquitectura permite la especialización del personal en un componente único, potencializando el trabajo en equipo y eliminando el exceso de responsabilidades a un mismo desarrollador, favoreciendo así al proyecto en rapidez y calidad.

3.2.2 Patrones del Diseño

Un patrón de diseño describe una estructura de diseño que resuelve un problema de diseño en particular. Estos patrones se aplican a un elemento específico, relaciones entre componentes o los mecanismos para efectuar la comunicación de componente a componente. (35)

Para dar solución al problema planteado se hizo necesaria la utilización de patrones de diseños los cuales se clasifican **GRASP** (General Responsibility Assignment Software Patterns) y **GOF** (Gang-of-Four).

Los patrones de asignación de responsabilidades GRASP, permiten asignar correctamente responsabilidades a cada una de las clases (38), de este grupo de patrones se utilizaron los siguientes:

3.2.2.1 Experto

Lo que plantea este patrón es asignar una responsabilidad al experto en información, es decir, la clase que tiene la información necesaria para cumplir con la responsabilidad. El problema que resuelve el patrón experto está referido al principio más básico mediante el cual las responsabilidades son asignadas en el diseño orientado a objetos. Un ejemplo donde se presencia la utilización de este patrón es en las clases que constituyen componentes como Estadísticas, la cual tiene la información referente a las solicitudes y que por lo tanto las funcionalidades para manipularlos (Modificarlos o eliminarlos) deben quedar implementados en esta clase.

3.2.2.2 Bajo Acoplamiento

Se utilizó para establecer un uso mínimo de clases entrelazadas de modo que si se hacen modificaciones solamente afecten a la menor cantidad de clases. Esto se evidencia en la estructura de componentes que se ha empleado, como el reproductor, el componente de búsqueda, y las solicitudes, los cuales pueden ser reutilizados de manera independiente sin que sus funcionalidades se vean afectadas por estar separados del resto de la estructura del módulo.

3.2.2.3 Controlador

Este patrón se utilizó pues propone que se debe asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente el sistema global. En la implementación del módulo se puede ver que la clase controladora **PrestamoActions** se encarga de encausar las funcionalidades realizadas en el sistema, por ejemplo, si se requieren datos existentes en la base de datos esta clase ejecuta las funcionalidades correspondientes en el modelo para obtenerlos.

Además de los patrones anteriores se hace uso de los patrones GOF (Gang Of Four), los cuales se dividen en varias categorías entre las que se encuentran: creacionales, estructurales y de comportamiento y dentro de estas los patrones que se manifiestan en la solución:

3.2.2.4 Decorador

En el módulo de consulta y recuperación de materiales audiovisuales se encuentra un archivo llamado layout.php que contiene el layout de la página principal del proyecto. Este archivo, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este comportamiento es una implementación del patrón de diseño llamado "decorador".

3.2.2.5 Observador

Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan automáticamente todos los objetos (39). Los observadores necesitan saber cuándo se produce un cambio en el objeto observado. La solución que se aplicaría usando el patrón Observador, trata de que el objeto observador que en este caso es la clase (Observer_Pattern_Implementation) se añadan a una lista de objetos, y el objeto observado (SCCM_Search_Component) notificará el cambio a todos los objetos de esta lista cuando se produzca el cambio.

3.3 Diagramas de clases del diseño

Los diagramas de clases del diseño constituyen un elemento esencial en la concepción de la aplicación que se propone, ya que servirán de guía a los desarrolladores para la construcción de una aproximación del sistema que se desea implementar, contribuyendo así a una mayor calidad del

producto final (40). Además describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación.

A continuación se muestran los diagramas de clases del diseño correspondientes a cada caso de uso.

3.3.1 Diagrama de clases del diseño para el caso de uso “Buscar Material”

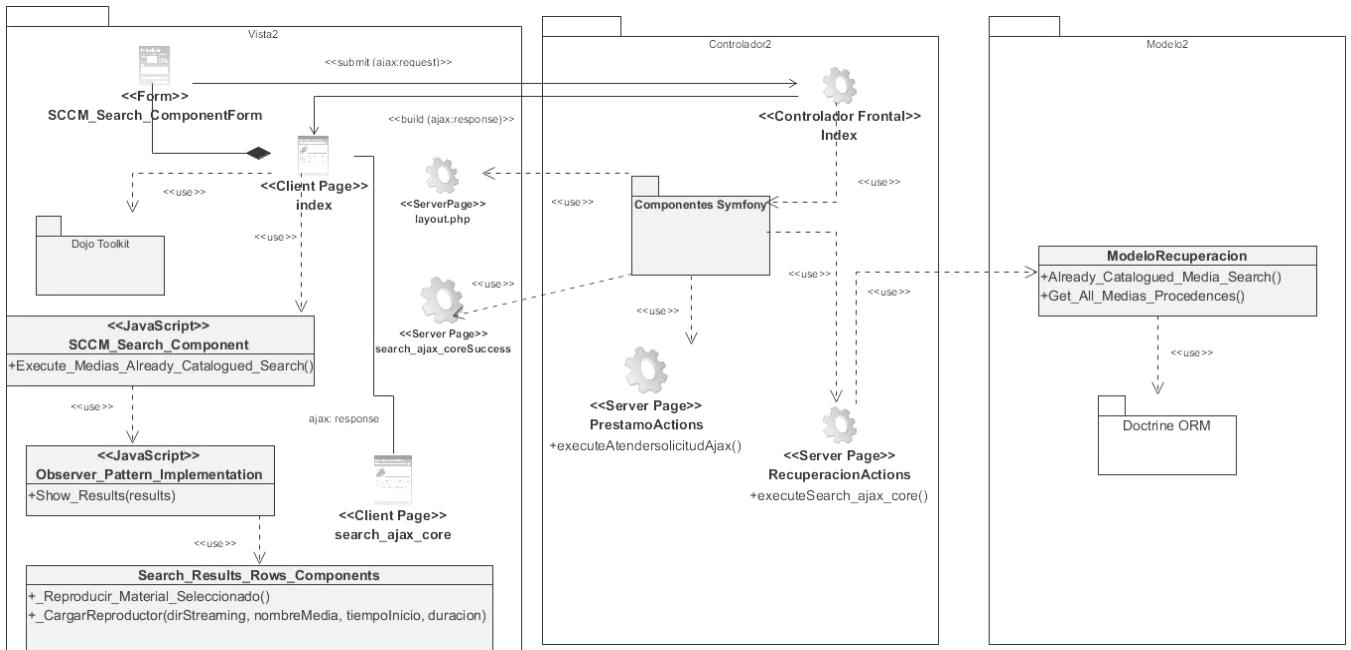


FIGURA 10: DIAGRAMA DE CLASES DEL DISEÑO PARA EL CASO DE USO “BUSCAR MATERIAL”

El caso de uso **Buscar Materiales** a solicitar describe el proceso de respuesta del sistema ante la petición del usuario de buscar una o varias medias a solicitar, estas medias son las que ya han sido catalogadas. Las clases principales son **SCCM_Search_Component** con el método **Execute_Medias_Already_Catalogued_Search ()**, y el formulario **SCCM_Search_ComponentForm** que ofrece campos como: título, texto o palabras claves para la búsqueda. Se utilizan también la clase controladora **RecuperacionActions** y modelo Recuperación que se apoya para este tipo de búsqueda en el método **Already_Catalogued_Medias_Search ()**.

3.3.2 Diagrama de clases del diseño para el caso de uso “Reproducir Material”

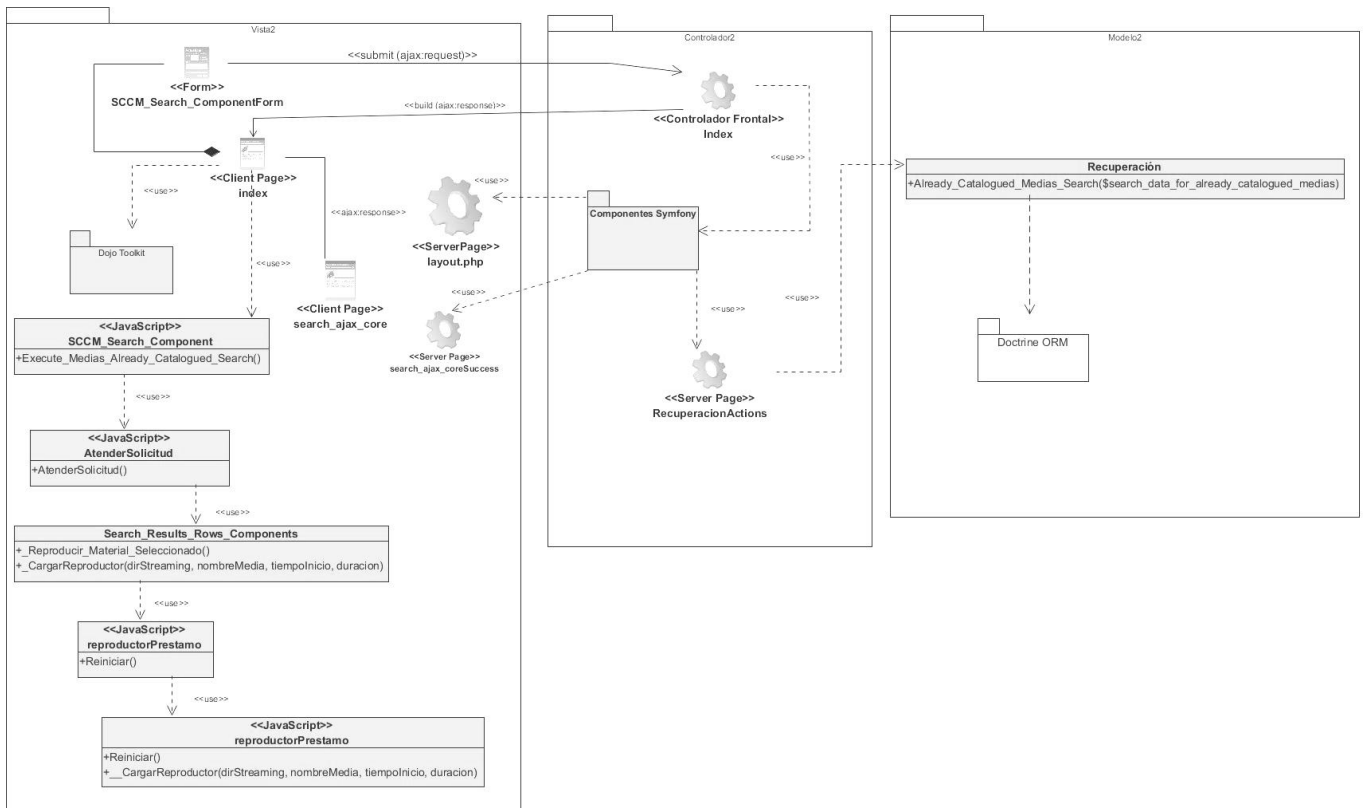


FIGURA 11: DIAGRAMA DE CLASES DEL DISEÑO PARA EL CASO DE USO REPRODUCIR MATERIAL

Para implementar el caso de uso Reproducir Material se diseñó el diagrama de clases que se muestra en la imagen anterior. La clase **SCCM_Search_Component** es la que inicializa el flujo de reproducción de una media en el sistema. Esta clase posee un método llamado **_publish_all_data_for_medias_already_catalogued_search ()** que retorna un listado de las medias encontradas durante la búsqueda y lo muestra en el formulario **SCCM_Search_ComponentForm**. Cuando el usuario selecciona la opción de reproducir una de las medias presentes en la lista y ordena la acción, el controlador **RecuperacionActions**, se encarga de dirigir la solicitud de la dirección streaming de la media hacia la clase del modelo **Recuperación** que se apoya en la funcionalidad **Already_Catalogued_Medias_Search ()**, para realizar las consultas pertinentes a la base de datos. El controlador envía los datos de la media a la clase de la vista **search_Ajax_CoreSuccess** desde donde son tomados por **_index**. Desde ese punto, el archivo javascript **Search_Results_Row_Component** inicializará el reproductor con el método **__Cargar_Reproductor ()**, pasándole la información necesaria a **reproductor_Prestamo**, que conforma el componente encargado de la visualización y el control de la reproducción del material audiovisual.

3.3.3 Diagrama de clases del diseño para el caso de uso “Gestionar Solicitud”

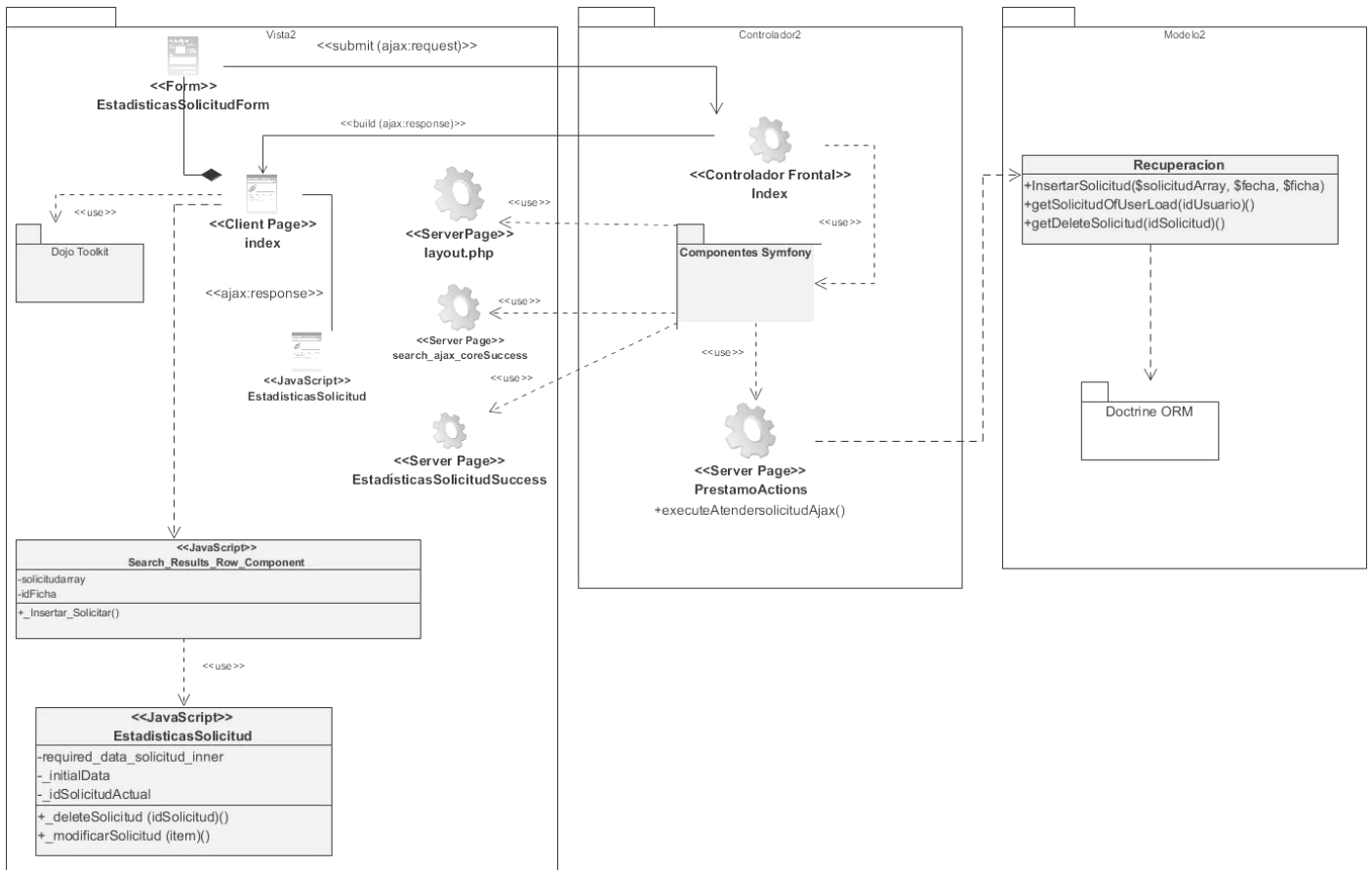


FIGURA 12: DIAGRAMA DE CLASES DEL DISEÑO PARA EL CASO DE USO GESTIONAR SOLICITUD

Las clases principales que intervienen en este caso de uso y que no han sido mencionadas anteriormente son **EstadisticasSolicitud**, que se encarga de gestionar la lista de las solicitudes creadas, con métodos como **delete_Solicitud ()** y **_modificar_ Solicitud ()**, y el formulario **EstadisticasSolicitudForm** el cual muestra dicha lista y sirve de intermediario entre el sistema y el usuario para que se puedan insertar, modificar o eliminar los elementos presentes en la misma. Los métodos **InsertarSolicitud ()**, **getSolicitudOfUser ()** y **getDeleteSolicitud ()** en la clase **Recuperación** del modelo se utilizan para obtener algunos datos de la solicitud que fue creada.

3.3.4 Diagrama de clases del diseño para el caso de uso “Atender Solicitud”

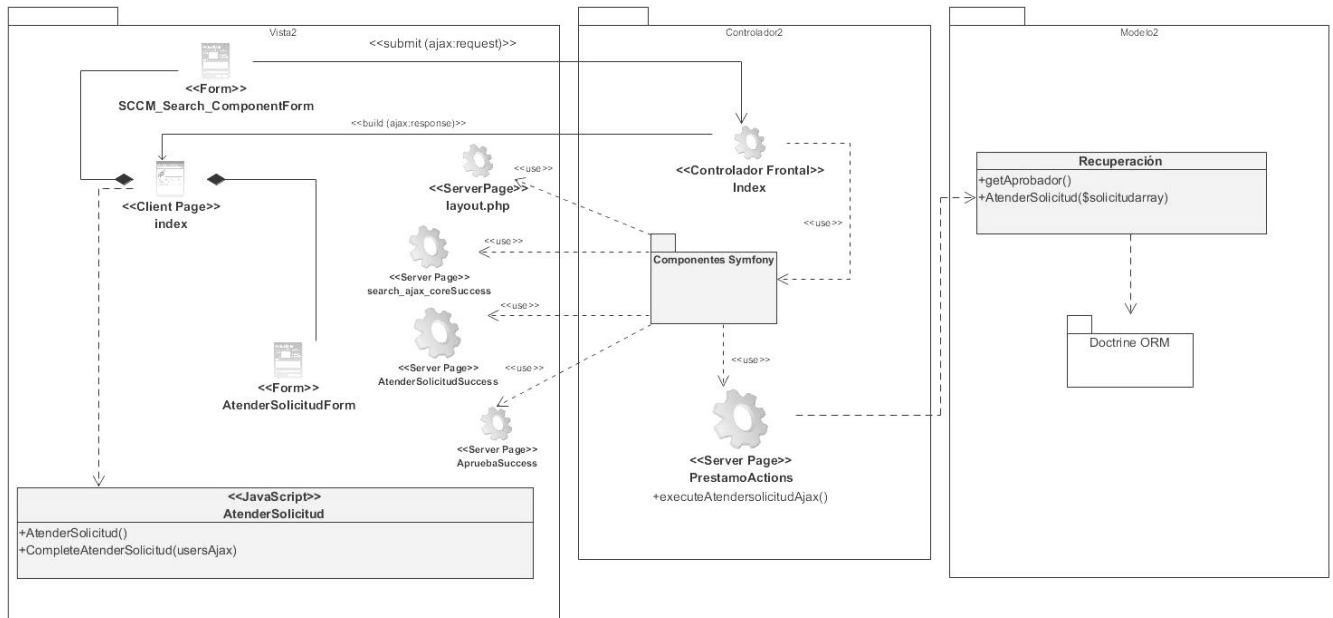


FIGURA 13: DIAGRAMA DE CLASE DEL DISEÑO DEL CASO DE USO ATENDER SOLICITUD

El caso de uso **Atender Solicitud** en la vista cuenta con el javascript **AtenderSolicitud**, estructurado para manipular la visualización de las solicitudes existentes, permite a través de los métodos **AtenderSolicitud()** y **CompletarAtenderSolicitud()** y si la solicitud fue aprobada y el usuario posee permisos descargar el material. La clase **SCCM_Search_Component** permite obtener los resultados de la búsqueda.

3.4 Diseño de la Base de Datos.

Un modelo de datos es un conjunto de conceptos que permiten describir los datos, las relaciones que existen entre ellos, la semántica y las restricciones de consistencia. Elaborar un correcto modelo de datos es imprescindible para manejar y obtener correctamente los datos. A continuación se muestra el modelo de datos del módulo consulta y recuperación de materiales audiovisuales.

En la imagen (Figura # 14) que se muestra a continuación aparece una porción del modelo de datos, que representa físicamente la base de datos. Esta fue diseñada en el proyecto SCCM para almacenar y gestionar la información referente a todo el sistema. En las mismas se reflejan las tablas

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS PARA LA SOLUCIÓN

relacionadas con el proceso de consulta y recuperación de materiales audiovisuales que se implementa en el módulo con este propósito.

Entre las tablas existentes en la base de datos, destinadas a manipular la información de la recuperación de los materiales audiovisuales, existen cinco que conforman la espina dorsal de este proceso: **DMedia**, **DFicha**, **DTipología**, **DCliente**, **DSolicitud**. Con DMedia se gestionan los datos propios de los archivos de video como la fecha en que fue insertada en el servidor de medias, de dónde proviene y su dirección streaming. DFicha en su estructura guarda la duración del material catalogado (ya sea una media completa o un subclip) tiempo de inicio, un resumen y, además, establece una relación con DTipología que caracteriza a la media descrita partiendo de tipos de contenido especificados por los trabajadores de la institución donde se despliegue la aplicación entre los cuales podría estar video clip o película, entre otros. DClientes la encargada de almacenar la información relacionada con la solicitud que realiza el usuario como son el carnet de identidad, el nombre, la entidad a la pertenece y los apellidos. DSolicitud recoge los datos pertenecientes a la solicitud que realizó el usuario, como la fecha, usuario y la naturaleza.

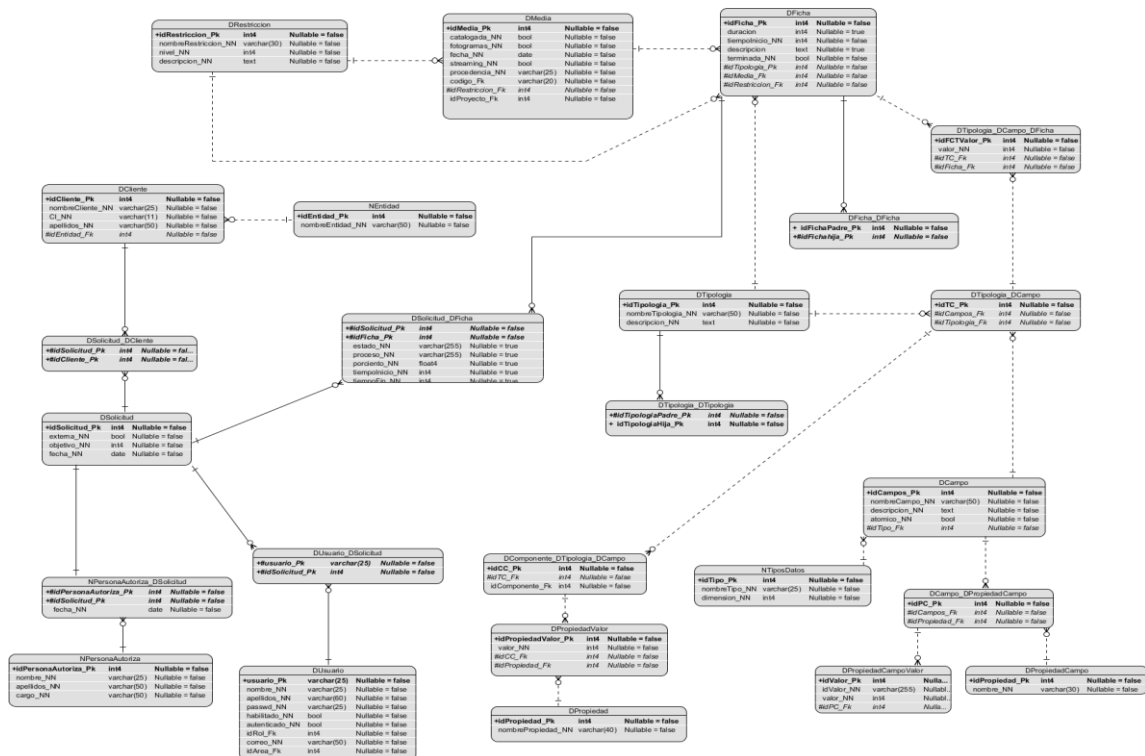


FIGURA 14: MODELO DE DATOS.

3.5 Diagrama de Despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Estos representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP (41). En la (Figura # 15) se muestra el diagrama de despliegue del módulo consulta y recuperación de materiales audiovisuales.

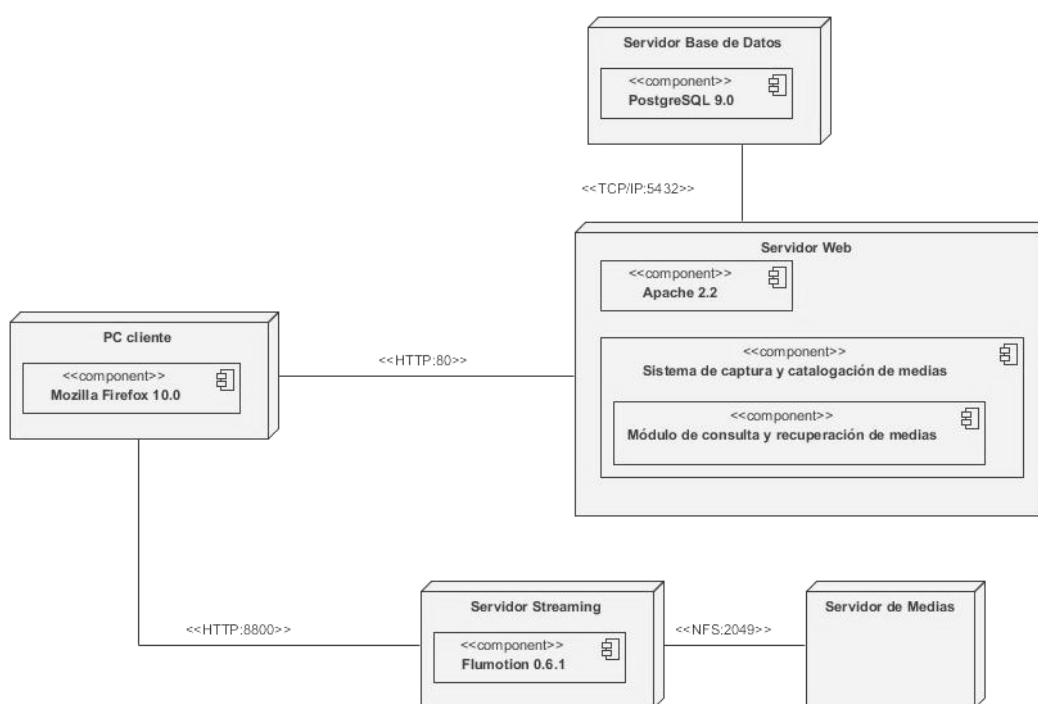


FIGURA 15: DIAGRAMA DE DESPLIEGUE

En la imagen anterior se muestra el diagrama de despliegue propuesto para el módulo web de Consulta y Recuperación de materiales audiovisuales el cual cuenta con cinco nodos que responden a la distribución física del sistema.

3.6 Diagrama de Implementación

Se derivan de los diagramas de proceso y módulos de la metodología de Booch, aunque presentan algunas modificaciones. Los diagramas de implementación muestran los aspectos físicos del sistema. Incluyen la estructura del código fuente y la implementación, en tiempo de implementación (42).

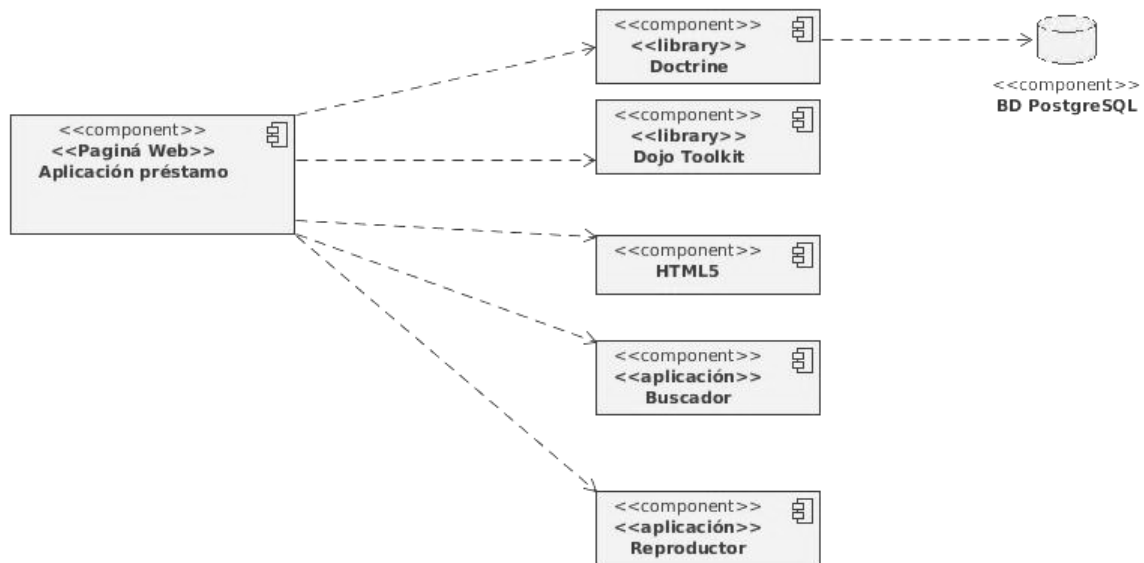


FIGURA 16: DIAGRAMA DE COMPONENTE

3.7 Estándar de codificación

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento. Además, si se aplica de forma continuada un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas, y, posteriormente, se efectúan revisiones del código de rutinas, caben muchas posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y de mantener.

Los nombres de clases pueden contener sólo caracteres alfanuméricos. Los números están permitidos en los nombres de clase, pero desaconsejados en la mayoría de casos. Las barras bajas (_) están permitidas solo como separador de ruta (el archivo " Zend/Db/Table.php " debe apuntar al nombre de clase " Zend_Db_Table "). Si el nombre de una clase está compuesto por más de una palabra, la primera letra de cada palabra debe aparecer en mayúsculas. Ejemplo no permitido: "Zend_PDF". Ejemplo correcto: " Zend_Pdf".

Los nombres de funciones deben comenzar con mayúscula. Los guiones bajos (_) solo están permitidos para el caso en que el nombre de la función tenga más de una palabras. Después del guion bajo (_) debe comenzar con mayúscula solo la primera letra. Los números están permitidos en los nombres de función pero no se aconseja en la mayoría de los casos. Por norma general, se

recomienda la elocuencia. Los nombres de función deben ser lo suficientemente elocuentes como para describir su propósito y comportamiento.

Los nombres de variables deben empezar siempre con una letra en minúscula. Para las variables de instancia que son declaradas con el modificador "private" o "protected", el primer carácter de la variable debe ser una única barra baja (_). Las variables declaradas como "public" no pueden empezar nunca por barra baja. Los guiones bajos (_) solo están permitidos para el caso en que el nombre de la variable tenga más de una palabra y después del guion se mantiene en minúscula.

```
public function Get_Frame($nb_media) {  
  
    $frames = glob("../sm/".$nb_media."/*.jpg" , GLOB_BRACE);  
    $nombres = array();  
    foreach ($frames as $frame) {  
        $url = explode("/", $frame);  
        $nombre = array_pop($url);  
        $nombres[] = $nombre;  
    }  
  
    return $nombres;  
}
```

FIGURA 17: CODIGO PHP

3.8 Conclusiones Parciales

- El uso de patrones permitió elaborar diagramas de clases de diseño fiables para la implementación de los diferentes casos de uso del módulo web de consulta y recuperación de materiales audiovisuales.
- La elaboración de los diagramas de clases del diseño, junto a otros artefactos como el modelo de datos, el diagrama de despliegue, el modelo de implementación y el estándar de codificación, sirvió de guía para la implementación del módulo y ha formado la base para que otros desarrolladores mejoren sus funcionalidades y le den mantenimiento al software.
- Haber realizado los flujos de trabajo establecidos en la metodología seleccionada permitió el desarrollo del sistema en el tiempo establecido.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

4.1 Introducción

Este capítulo trata sobre las pruebas realizadas al sistema. Se describen los casos de prueba del sistema y se realizan las pruebas de caja negra sobre las interfaces.

4.2 Pruebas

La prueba es un proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene alta probabilidad de mostrar un error no descubierto. 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, a pesar de esto una prueba no puede asegurar la ausencia de errores; solo puede demostrar que existen defectos en el software (43).

Cualquier producto de ingeniería se puede probar de dos formas (43):

- **Pruebas de caja blanca:**

En este tipo de prueba se realiza un examen minucioso de los detalles procedimentales, comprobando los caminos lógicos del programa, los bucles y condiciones, y examinando el estado del programa en varios puntos.

- **Pruebas de caja negra:**

Estas pruebas permiten pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta. Se centran en los requisitos funcionales del software y permiten encontrar errores de tipo de funciones incorrectas o ausentes, de interfaz, en la estructura de datos, de rendimiento y de inicialización y terminación.

Al módulo de consulta y recuperación de materiales audiovisuales se le realiza pruebas de caja negra pues estas se centran en lo que se espera del mismo, es decir, intentan encontrar casos en que este no atiende a su especificación. Por ello se denominan pruebas funcionales y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo por dentro.

El método de prueba de caja negra utilizado fue el de partición equivalente:

“La partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Se dirige a la definición

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar (43).”

Según RUP estas pruebas verifican el comportamiento de la unidad observable externamente. Cuando el número de entradas y salidas es grande, se dividen estas en clases de equivalencias. “Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (43).”

4.3 Casos de Prueba

Un caso de prueba es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Los casos de pruebas se pueden derivar de los casos de uso del sistema o de la realización de estos en el modelo de diseño, permitiendo así validar los requerimientos funcionales del sistema (36).

4.3.1 Diseño de caso de prueba para el caso de uso Buscar materiales:

Caso de prueba					
Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central	Resultados Esperados	Resultados Obtenidos
SC 1: Verificar sección de Búsqueda de materiales catalogados.	EC 1.1: Se obtienen los resultados satisfactoriamente.	El usuario inserta valores que correspondan con las medias catalogadas almacenadas en el sistema y se muestran las mismas como resultado de la búsqueda.	Módulo Préstamo/Sección de Búsqueda de materiales catalogados.	El sistema debe mostrar la tabla de resultados con las medias catalogadas existentes en el archivo digital que cumplan con los filtros seleccionados de la aplicación.	El sistema muestra la tabla de resultados correspondientes a la búsqueda.
	EC 1.2: No se obtienen resultados de la búsqueda.	El usuario inserta valores que no posean las medias catalogadas almacenadas en el sistema no se muestran resultados de la búsqueda.	Módulo Préstamo/Sección de Búsqueda de materiales catalogados.	El sistema debe mostrar en la lista de resultados un mensaje indicando que no hay medias que cumplan con los filtros insertados.	El sistema no muestra ningún resultado

TABLA 1: CASO DE PRUEBA PARA EL CASO DE USO BUSCAR MATERIAL

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

4.3.2 Diseño de caso de prueba para el caso de uso Reproducir Materiales

Caso de prueba					
Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central	Resultados Esperados	Resultados Obtenidos
SC 1: Reproducir material.	EC 1.1: Visualizar correctamente.	El usuario elige una de las medias en el resultado de la búsqueda y el sistema la reproduce.	Módulo Préstamo/Efectuar una búsqueda/Seleccionar la media catalogada/Hacer clic sobre el botón reproducir	El sistema debe mostrar en un cuadro de imagen la reproducción en curso.	Se muestra la reproducción del material.
	EC 1.2: Controlar la reproducción.	El usuario manipula la reproducción en curso utilizando los controles del reproductor.	Durante la reproducción de una media.	El reproductor debe responder correctamente ante cada acción realizada por el usuario.	El reproductor responde correctamente.

TABLA 2: CASO DE PRUEBA PARA EL CASO DE USO REPRODUCIR MATERIALES

4.3.3 Diseño de caso de prueba para el caso de uso Gestionar Solicitud

Caso de prueba					
Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central	Resultados Esperados	Resultados Obtenidos
SC 1: Gestionar Solicitud.	EC 1.1: Crear la solicitud correctamente.	El usuario selecciona el material que desea solicitar, correspondientes y el sistema lo agrega a la las estadísticas del usuario	Módulo Préstamo/Luego de obtener el material deseado, introduces los datos/Botón Aceptar	El sistema debe mostrar un mensaje que indique que se añadió correctamente la solicitud y añadirse a la lista de estadísticas.	El sistema muestra la solicitud realizada en las estadísticas del usuario.
	EC 1.2: No se puede crear la solicitud correctamente	El usuario selecciona el material que desea solicitar, pero no llena los datos de la solicitud y al presionar el botón Aceptar, se muestra un mensaje de error.	Módulo Préstamo/Luego de obtener el material deseado, introduces los datos/Botón Aceptar	El sistema debe mostrar un mensaje indicando que no se puede crear la solicitud.	El sistema muestra el mensaje esperado.
	EC 1.3: Eliminar Solicitud.	El usuario elimina una solicitud creada, de la lista y el sistema elimina sus datos.	Módulo Préstamo/Lista de Estadísticas Solicitudes/Botón		El sistema debe quitar de la lista de solicitudes creadas la que

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

		Eliminar Solicitud.	fue eliminada por el usuario.	
EC 1.4: Modificar Solicitud	El usuario selecciona la solicitud que desea modificar y aparece de nuevo la solicitud con los datos para que modifique los que desee.	Módulo Préstamo/Lista de Estadísticas /Botón Modificar Solicitud.	El sistema muestra la interfaz para modificar la solicitud con los datos correspondiente s a esa solicitud.	El sistema muestra la interfaz para modificar la solicitud con los datos correspondientes a esa solicitud.

TABLA 3: CASO DE PRUEBA PARA EL CASO DE USO GESTIONAR SOLICITUD.

4.3.4 Caso de prueba para el caso de uso Atender Solicitud

Caso de prueba					
Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central	Resultados Esperados	Resultados Obtenidos
SC 1: Atender Solicitud	EC 1.1: Se pueden obtener las solicitudes creadas saber su estado y descargar el material seleccionado.	El usuario busca la solicitud que desea si tiene permiso puede descargar el material.	Módulo Préstamo/Sección Atender Solicitud	El sistema debe mostrar la tabla de resultados con las solicitudes existentes que cumplan con los filtros de búsqueda seleccionados.	El sistema muestra la tabla de resultados correspondientes a la búsqueda
	EC 1.2: No se puede atender la solicitud	El usuario busca la solicitud que desea si tiene permiso puede descargar el material.	Módulo Préstamo/Sección Atender Solicitud	El sistema debe mostrar en la lista de resultados un mensaje indicando que no hay medias que cumplan con los filtros insertados.	El sistema muestra el mensaje de "0 resultados" en la lista resultante de la búsqueda.

TABLA 4: CASO DE PRUEBA PARA EL CASO DE USO ATENDER SOLICITUD

4.3.5 Caso de prueba para el caso de uso Descargar Solicitud

Caso de prueba					
Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central	Resultados Esperados	Resultados Obtenidos
SC 1: Descargar Material.	EC 1.1: El usuario selecciona la opción de descargar material	El usuario si tiene permiso y la solicitud fue aprobada puede descargar el	Módulo Préstamo/Sección Descargar Material	El sistema debe permitir descargar el material que desea el usuario	El sistema permite descargar el material que desea el usuario.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

		material.		y mostrar un mensaje.	
	EC 1.2: El usuario no puede descargar el material seleccionado	El usuario tiene permiso para descargar el material, pero la solicitud no fue aceptada.	Módulo Préstamo/Sección Descargar Material	El sistema debe mostrar un mensaje describiendo que no se puede descargar el material.	El sistema no permite descargar el material.

TABLA 5: CASO DE PRUEBA PARA EL CASO DE USO DESCARGAR SOLICITUD

4.4 Resultados

Las pruebas de caja negra fueron realizadas de manera estricta a partir de los casos de prueba elaborados. En la culminación de este proceso se obtuvo como resultado que, exceptuando la búsqueda de materiales catalogados, las funcionalidades del módulo responden como se definió al inicio del proyecto. La no conformidad referente a la búsqueda de materiales catalogados fue resuelta posterior a su hallazgo y actualmente la búsqueda por clasificación funciona como es necesario, devolviendo los resultados correctos cada vez que es utilizada en el sistema. Para cada una de estas pruebas se tuvo en cuenta las diferentes situaciones que podían existir en las interacciones con el usuario, con el fin de evaluar el comportamiento de la funcionalidad en cada situación, obteniéndose como resultado la igualdad con la especificación de los requisitos, lo que demuestra la correcta implementación de los mismos.

4.5 Conclusiones Parciales

La realización de este capítulo permitió conocer la estructura física del sistema, validar los requerimientos identificados en anteriores capítulos, mediante las pruebas funcionales, obteniéndose resultados satisfactorios que validan la solución del problema presentado. Se puede añadir que:

- Realizar las pruebas de caja negra permitió detectar, documentar y solucionar los errores existentes en el sistema implementado.
- Se validó que el software creado satisface los requisitos identificados para el módulo consulta y recuperación de materiales audiovisuales.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

- Concluir el período de pruebas permitió obtener una aplicación que responde correctamente, cumpliendo con los objetivos planteados para el módulo web de consulta y recuperación de materiales audiovisuales del proyecto SCCM.

CONCLUSIONES GENERALES

Se puede concluir que:

- La caracterización de los procesos relacionados con la consulta y recuperación de materiales audiovisuales, permitió comprender e implementar con mayor precisión las funcionalidades correspondientes.
- El trabajo de ingeniería realizado durante el desarrollo de la aplicación permitió documentar detalladamente los artefactos generados en los flujos de trabajo de la metodología seleccionada, con el objetivo de que los mismos sean utilizables en versiones futuras.
- El producto obtenido cumple con las funcionalidades establecidas, mejorando la accesibilidad a los videos en las instituciones donde sea desplegado.
- El uso de tecnologías y herramientas libres para el desarrollo del módulo permitió cumplir con la política de soberanía tecnológica por la que aboga el país.

RECOMENDACIONES

Al concluir el presente trabajo, se recomienda:

- Realizar el estudio del proceso para permitir al usuario gestionar la manera de visualizar el resultado de la búsqueda, ya sea agrupados por tipologías o todos los existentes.
- Formalizar el estudio de los procesos que se pueden efectuar cuando los usuarios realicen una solicitud para que los mismos tengan la posibilidad de escoger varios materiales audiovisuales.

BIBLIOGRAFÍA

1. Fuentes y recursos de Información.

[Citado el: 6 de 6 de 2012.]

<http://www.concla.net/Tutorial%20Fuentes/fuentes%20y%20recursos/Audiovisuales.html>.

2. UNESCO.

[Citado el: 6 de 6 de 2012.] <http://www.unesco.org/webworld/publications/philos/philos3.htm>.

3. Molina, Maria Pinto. BÚSQUEDA Y RECUPERACIÓN DE INFORMACIÓN.

http://www.mariapinto.es/e-coms/recu_infor.htm.

4. Antonio Gabriel Lopez Herrera. S

[Citado el: 6 de 6 de 2012.] <http://recuperaciondinformacion.blogspot.com/.CI2S Thematic Public Websites>.

[Citado el: 6 de 6 de 2012.] http://sci2s.ugr.es/publications/ficheros/tesis_difinitiva.pdf.

5. Real Academia de la Lengua Española.

[Citado el: 8 de 6 de 2012.] http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=solicitud.

6. Rincon del Vago.

[Citado el: 6 de 6 de 2012.] <http://recuperaciondinformacion.blogspot.com/>.

7. Salle, Universidad de la. ARCHIVES IN CLOUD.

8. Silvia Contardi. EcuRed.

[Citado el: 7 de 6 de 2012.] <http://www.ecured.cu/index.php/YouTube>.

9. CRISTIANV.QUISPE. Slideshare.

[Citado el: 7 de 6 de 2012.] <http://www.slideshare.net/promo2011iem/herramientas-web-20quispe-10554308>.

10. Perfilesweb.

[Citado el: 7 de 6 de 2012.] <http://perfilesweb.com/dailymotion/>.

11. Recursosfree.

[Citado el: 7 de 6 de 2012.] <http://www.recursosfree.com/alojamiento-hosting/videos/?s=A&p=1>.

12. Xtream,2012. Xtream, Sistema de Información Global.

[Citado el: 7 de 6 de 2012.] <http://www.xtreamsig.com/>.

13. Baby González, Ronald y Almeida Maldonado, Enrique.

Sistema de Catalogación de Medias. Ciudad Habana : s.n., 2009.

14. Multimedia, ISID Gestión Inteligente de Activos. ISID.

[Citado el: 26 de 11 de 2012.] http://www.isid.es/Spanish/product/producto_v00.htm.

15. Pressman, Roger Putnam y Myers. *Five core metrics*. s.l. : Dorset House, 2003.

16. Orallo, Enrique Hernández. EL Lenguaje Unificado de Modelado UML.

[Citado el: 7 de 6 de 2012.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.

17. Milestone.

[Citado el: 14 de 1 de 2012.]

http://www.milestone.com.mx/articulos/el_lenguaje_unificado_de_modelado.htm.

18. Instituto Nacional de Estadística e Informática de Peru. *Instituto Nacional de Estadística e Informática de Peru*.

[Citado el: 14 de 1 de 2012.] <http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>.

19. Universidad de las Américas Puebla. Colección de Tesis Digitales. *Colección de Tesis Digitales*.

[Citado el: 15 de 1 de 2012.]

http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf.

20. Lenguajes de programación. *Lenguajes de programación*.

[Citado el: 17 de 1 de 2012.] <http://www.mitecnologico.com/Main/DefinicionDeProgramacion>.

21. Lenguaje del lado del Cliente. *Lenguaje del lado del Cliente*.

[Citado el: 17 de 1 de 2012.]

http://www.google.com.cu/url?sa=t&rct=j&q=lenguaje+del+lado+del+cliente&source=web&cd=8&ved=0CFsQFjAH&url=http%3A%2F%2Fwww.ecured.cu%2Findex.php%3Ftitle%3DEspecial%3APdfprint%26page%3DLenguaje_de_Programaci%25C3%25B3n_Web&ei=xoYVT5jvO4nh0QH9maSRAw&usg=AFQ.

22. Álvaro Martínez Echevarría. MANUAL PRÁCTICO DE HTML. *MANUAL PRÁCTICO DE HTML*.

[Citado el: 22 de 1 de 2012.] <http://www-app.etsit.upm.es/~alvaro/manual/manual.html>.

23. Miguel Angel Alvarez. Desarrollo Web. *Desarrollo Web*.

[Citado el: 26 de 1 de 2012.] <http://www.desarrolloweb.com/articulos/25.php>.

24. Biblioteca Digital. *Biblioteca Digital*.

[Citado el: 26 de 1 de 2012.] <http://www.biblioteca-digital.net.ve/wordpress/2010/07/29/curso-de-javascript-inicial/>.

25. Libros Web. *Libros Web*.

<http://www.librosweb.es/css/capitulo1.html>.

26. Guía Breve de CSS. *Guía Breve de CSS*.

<http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>.

27. Lic. Yoandry Pacheco Aguila. AJAX un nuevo acercamiento a las aplicaciones Web. *AJAX un nuevo acercamiento a las aplicaciones Web*.

<http://www.monografias.com/trabajos43/ajax/ajax2.shtml>.

28. Lenguajes del lado del Cliente. *Lenguajes del lado del Cliente*.

[Citado el: 17 de 1 de 2012.]

http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.

29. Javier J. Gutiérrez. . CSSBlog.es. *CSSBlog.es*.

[Citado el: 6 de 2 de 2012.] <http://www.cssblog.es/guias/Framework.pdf>.

30. Fabien Potencier, François Zaninotto. Libros Web. *Libros Web*.

[Citado el: 7 de 2 de 2012.]

http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html.

31. Rose India. *Rose India*.

[Citado el: 31 de 1 de 2012.]

http://translate.googleusercontent.com/translate_c?hl=es&prev=/search%3Fq%3Ddojo%2Btutorial%26hl%3Des%26sa%3DX%26biw%3D1366%26bih%3D610%26prmd%3Dimvns&rurl=translate.google.com.cu&sl=en&u=http://www.roseindia.net/dojo/dojo-tips.shtml&usg=ALkJrhgtAg1fq9NSu.

32. Netbeans. *Netbeans*.

[Citado el: 8 de 2 de 2012.] http://netbeans.org/index_es.html.

33. CAVSI.

[Citado el: 9 de 2 de 2012.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.

34. Daniel M. Maldonado. El código.

[Citado el: 9 de 2 de 2012.] <http://www.elcodigok.com.ar/2008/07/%C2%BFque-es-postgresql/>.

35. Xavier Ferré Grau, María Isabel Sánchez Segura. Scribd.

[Citado el: 18 de 5 de 2012.] <http://es.scribd.com/doc/51139064/95/IV-4-4-Diagrama-de-Clases-de-Diseno>.

36. Pressman, Roger. *Ingeniería de Software. Un enfoque práctico*. España : Mc Graw Hill, 2005. 9701054733.

37. Jacobson, Booch, Rumbaugh. *El Proceso Unificado de Desarrollo de Software*. Madrid : Addison Wesley, 2000.

38. Potencier Fabien, Zaninotto François. *Symfony la guía definitiva*. 2008.

39. Larman, Craig. *UML Y PATRONES. Introducción al análisis y diseño orientado a objetos*. México : PRENTICE HALL, 1999. 970-17-0261-1.

40. Joaquin Gracia. *Ingeniero Software*.

[Citado el: 20 de 5 de 2012.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.

41. CARLOS REYNALDO. *Análisis y Diseño*.

http://www.google.com.cu/url?sa=t&rct=j&q=diagrama+de+despliegue+definicion&source=web&cd=6&ved=0CGYQFjAF&url=http%3A%2F%2Fvirtual.usalesiana.edu.bo%2Fweb%2Fpractica%2Farchiv%2Fdesplieg1.ppt&ei=HE-6T5_dGM7ogAeRhKzhCg&usg=AFQjCNFYcsTRWyxRUu5Jw5ENJZ7jd3M94Q.

42. *Diagramas de implementación*.

[Citado el: 21 de 5 de 2012.]

http://gidis.ing.unlpam.edu.ar/personas/glafuente/uml/diagramas_de_implementacion.html.

43. Pressman, Roger. *Ingeniería de Software un enfoque práctico*. s.l. : Dorset House, 2006.

GLOSARIO DE TÉRMINOS

TCP/IP : Son las siglas de Protocolo de Control de Transmisión/Protocolo de Internet (en inglés Transmission Control Protocol/Internet Protocol), un sistema de protocolos que hacen posibles servicios Telnet, FTP, E-mail, y otros entre ordenadores que no pertenecen a la misma red.

CASE: Computer Aided Software Engineering. Bajo el término de Ingeniería de Software Asistida por Ordenador se incluyen una serie de herramientas, lenguajes y técnicas de programación que permiten la generación de aplicaciones de manera semiautomática. Las herramientas CASE liberan al programador de parte de su trabajo y aumentan la calidad del programa a la vez que disminuyen sus posibles errores.

Common Gateway Interface, CGI: Protocolo genérico que permite extender las capacidades de HTTP.

BSD: Licencia de software otorgada principalmente para los sistemas (Berkeley Software Distribution) BSD. Es una licencia de software libre, tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público, permite el uso del código fuente en software no libre.