

Universidad de las Ciencias Informáticas
FACULTAD 6



Título: Diseño e implementación del módulo de recuperación de materiales audio-visuales para el Sistema Captura y Catalogación de Medias.

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

Autor: Yoendry Fuentes Hernández

Tutor: Ing. Yoandri Quintana Rondón

Junio 2012



“Se puede adquirir conocimientos y conciencia a lo largo de toda la vida, pero jamás en ninguna otra época de su existencia una persona volverá a tener la pureza y el desinterés con que, siendo joven, se enfrenta a la vida.”

DEDICATORIA

Dedico mi tesis primeramente a nuestro Comandante Fidel Castro Ruz, por ser mi ídolo y por ser un líder en nuestra revolución.

Se la dedico a toda mi familia, muy en especial a mi mamá y a Dagoberto Prieto Gómez, que aunque no es mi padre de sangre, lo es de corazón y a él y a mi mamá les debo todo. A mi hermanita linda, que la quiero con la vida. También a mi papá, a mis abuelos, a mi novia que siempre me estuvo apoyando en los buenos y malos momentos y a mis tíos y primas.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al centro Geoinformática y Señales Digitales de 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 _____.

Yoendry Fuentes Hernández

Ing. Yoandri Quintana Rondón

RESUMEN

Con el desarrollo de las tecnologías de la información y las comunicaciones (TIC) y el surgimiento de internet, los materiales audiovisuales digitales aumentaron considerablemente. Muchas instituciones como las televisoras, productoras de cine y videotecas, utilizan las medias (audio, video, imagen) como entidad fundamental. Las compañías que se dedican a este tipo de funcionalidades requieren de mecanismos de búsqueda que posibiliten la recuperación precisa de dichos materiales.

El presente trabajo constituye la realización de la versión web del módulo de recuperación de materiales audio-visuales del Sistema Captura y Catalogación de Medias, dando cumplimiento a los requisitos funcionales y garantizando un despliegue fácil de la aplicación. En su elaboración se han utilizado tecnologías completamente libres que permiten dar respuestas a las exigencias del país y constituye una alternativa factible y aplicable en cualquier entidad que necesite recuperar materiales audiovisuales.

PALABRAS CLAVE

Audiovisuales, búsqueda, materiales, recuperación, web

CONTENIDO

<i>DEDICATORIA</i>	<i>II</i>
<i>DECLARACIÓN DE AUTORÍA</i>	<i>III</i>
<i>RESUMEN</i>	<i>IV</i>
<i>INTRODUCCIÓN</i>	<i>1</i>
<i>CAPÍTULO 1: INTRODUCCIÓN AL ESTUDIO DE LA RECUPERACIÓN DE MATERIALES AUDIOVISUALES. FUNDAMENTACIÓN TEÓRICA.</i>	<i>5</i>
1.1 <i>Introducción</i>	<i>5</i>
1.2 <i>Conceptos asociados al problema</i>	<i>5</i>
1.2.1 <i>Multimedia</i>	<i>5</i>
1.2.2 <i>Búsqueda y recuperación</i>	<i>5</i>
1.3 <i>Procesos básicos involucrados en la búsqueda y recuperación de materiales audiovisuales</i> ..	<i>8</i>
1.3.1 <i>Descripción general</i>	<i>8</i>
1.3.1.1 <i>Técnicas de recuperación de información</i>	<i>8</i>
1.3.1.2 <i>Herramientas generales de recuperación</i>	<i>9</i>
1.4 <i>Descripción actual del dominio del problema</i>	<i>10</i>
1.5 <i>Situación problemática</i>	<i>11</i>
1.6 <i>Análisis de otras soluciones existentes</i>	<i>12</i>
1.6.1 <i>Mediabox</i>	<i>12</i>
1.6.2 <i>Td^mTarsys, de la empresa Tedial</i>	<i>12</i>
1.6.3 <i>Videoma</i>	<i>13</i>

1.7	Conclusiones Parciales.....	13
CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS A UTILIZAR PARA REALIZAR LA RECUPERACIÓN DE MATERIALES AUDIOVISUALES.....		15
2.1	Introducción.....	15
2.2	Tendencias y tecnologías web	15
2.2.1	Estilo arquitectónico Cliente – Servidor.....	15
2.2.2	Metodología de desarrollo de software	16
2.2.3	Proceso Unificado de Desarrollo o Rational Unified Process (RUP)	17
2.2.4	Lenguaje de Modelado: UML.....	19
2.2.5	Herramientas CASE.....	19
2.2.5.1	Visual Paradigm	20
2.2.6	Lenguaje de Programación.....	20
2.2.6.1	Lenguajes del lado del cliente	21
2.2.6.2	Lenguajes del lado del servidor.....	23
2.2.7	Framework.....	23
2.2.8	Entorno de Desarrollo Integrado (IDE).....	25
2.2.9	Sistemas Gestores de Base de Datos	26
2.3	Conclusiones Parciales	27
CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN.		28
3.1	Introducción.....	28
3.2	Estilos Arquitectónicos	28
3.2.1	Patrones Arquitectónicos	29

3.2.1.1	Patrones de Diseños	30
3.3	Diagramas de Clases del Diseño.	31
3.3.1	Diagrama de clases del diseño para el caso de uso Buscar Material a Catalogar.....	31
3.3.2	Diagrama de clases del diseño para el caso de uso Buscar Material.....	33
3.4	Diseño de Base de Datos.....	35
3.5	Diagrama de Despliegue	36
3.6	Modelo de Implementación	37
3.7	Estándar de Codificación.....	39
3.8	Integración	41
3.9	Conclusiones Parciales	41
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.		42
4.1	Introducción.....	42
4.2	Pruebas de Caja Negra.....	42
4.3	Casos de Prueba.....	43
4.3.1	Casos de Prueba Para el Caso de Uso Buscar Materiales a Catalogar	43
4.3.2	Casos de Prueba Para el Caso de Uso Buscar Materiales	44
4.4	Resultado de las Pruebas	45
4.5	Conclusiones Parciales	45
CONCLUSIONES GENERALES.....		46
RECOMENDACIONES.....		47
TRABAJOS CITADOS.....		48
BIBLIOGRAFÍA.....		52

<i>Anexo 1. Requisitos funcionales</i>	<i>56</i>
<i>Anexo 2. Diagrama de Caso de Uso</i>	<i>57</i>
<i>Anexo 3. Caso de Uso Buscar Material.....</i>	<i>57</i>
<i>Anexo 4. Caso de Uso Buscar Material a Catalogar</i>	<i>66</i>
<i>GLOSARIO DE TÉRMINOS</i>	<i>68</i>

ÍNDICE DE TABLAS

Tabla 1: Casos de Prueba Para el Caso de Uso Buscar Materiales a Catalogar..... 43

Tabla 2: Casos de Prueba Para el Caso de Uso Buscar Materiales..... 44

Tabla 3: Descripción del caso de uso Buscar Material. 57

Tabla 4: Descripción del caso de uso Buscar Material a Catalogar..... 66

ÍNDICE DE FIGURAS

Figura 1: Fases del proceso de recuperación de información. 6

Figura 2: Mediabox (XTREAM 2010, 2010)..... 12

Figura 3: TdtmTarsys, empresa Tedral (NAB11, Tedral, 2011)..... 13

Figura 4: Videoma (ISID Gestión Inteligente de Activos Multimedia, 2009). 13

Figura 5: Estilo Arquitectónico Cliente-Servidor. 16

Figura 6: Fases y Flujos de Trabajos según RUP (Molpeceres, 2002)..... 19

Figura 7: Modelo-Vista-Controlador (Potencier, 2009) 29

Figura 8: Diagrama de clases del diseño para el caso de uso Buscar Material a Catalogar..... 32

Figura 9: Diagrama de clases del diseño para el caso de uso Buscar Material. 34

Figura 10: Modelo de datos. 35

Figura 11: Modelo de despliegue. 37

Figura 12: Modelo de implementación. 38

Figura 13: Ejemplo de código PHP. 40

Figura 14: Ejemplo de código JavaScript. 40

Figura 15: Diagrama de Caso de Uso..... 57

INTRODUCCIÓN

En los últimos 50 años las relaciones humanas están mediadas principalmente por procesos de almacenamiento, procesamiento y transmisión de datos de forma electrónica. La evolución imparable de las Tecnologías de la Información y las Comunicaciones (TIC) ha propiciado un cambio fundamental en la forma de gestionar la información.

Con el desarrollo de internet¹, el incremento de la información digital accesible a la sociedad fue creciendo considerablemente y la necesidad de recuperar esa información se hizo cada vez mayor. Debido a esto se han creado un gran número de programas que su función principal es permitir localizar y recuperar la información almacenada en internet a través de palabras claves y conceptos asociados. Esto es facilitado también por los motores de búsqueda que se encuentran en sitios web, la mayoría reconocidos mundialmente.

El ejemplo que más se conoce es *Google*, pero existen otros como *AltaVista* y *Yahoo*. El propósito fundamental de estos es ofrecer la forma más rápida y sencilla de encontrar información en la Web. Esta información es muy diversa y variada, pero los programas audiovisuales ocupan un lugar muy importante dentro de ella.

Hoy en día las medias son recursos esenciales para enriquecer culturalmente la humanidad. La circulación de estos contribuye a reforzar los procesos comunicativos en la sociedad, facilitando una mayor comprensión e interpretación de ideas, además ayuda a difundir la información de modo más rápido y a una mayor cantidad de personas. Es por eso que en las últimas décadas ha ocurrido un crecimiento notable de este tipo de información.

Las medias a la hora de almacenarlas van acompañadas de una descripción textual que caracteriza a cada una. En general esta descripción se realiza mediante el uso de metadatos u otras características que el usuario defina. También puede estar relacionada con la información verbal que contienen, en caso de tener. Lo anteriormente relacionado da la posibilidad de crear un sistema de recuperación basado en la descripción, que facilita la búsqueda de estos elementos.

¹ Red informática de comunicación internacional que permite el intercambio de todo tipo de información entre sus usuarios.

Cuba también se ha visto inmersa en este desarrollo imparable de la información. Tanto es así que se han ido informatizando todos los centros educacionales, científicos y de trabajo de forma tal que casi toda la información se ha logrado almacenar de forma digital. Esto ha provocado la necesidad de crear sistemas de recuperación para esta información.

La Universidad de las Ciencias Informáticas se incluye dentro del proceso de informatización del país y promueve toda labor social y cultural acerca de las TIC. En ella se encuentra el centro de Geoinformática y Señales Digitales (GEYSED), el cual cuenta con un departamento que se especializa en el procesamiento y transmisión de las señales digitales. Dicho departamento cuenta con un grupo de proyectos, entre ellos está el Sistema de Captura y Catalogación de Medias (SCCM). Este proyecto se encarga de estudiar la forma para gestionar los audiovisuales así como su almacenamiento, recuperación y catalogación.

En el mundo existen algunas herramientas especializadas que se encargan de realizar estos procesos. El gran inconveniente está dado a que la mayoría de estos programas son propietarios y presentan un alto valor de adquisición, además de que atan a los usuarios a licencias muy costosas. Debido a esto el proyecto SCCM se dio la tarea de crear un *software*² con esas características, utilizando para ello tecnologías libres.

Para hacer todo el flujo de actividades que el *software* quiere automatizar, necesita realizar una fuerte recuperación de los audiovisuales digitales almacenados. Actualmente se cuenta con una versión de escritorio que brinda la posibilidad de recuperar los materiales audiovisuales, pero los usuarios quedan inconformes debido a que los criterios de búsqueda son muy pobres. Dicha aplicación presenta otro inconveniente a la hora del despliegue, puesto que es necesario instalarla en todas las computadoras de la entidad que la necesite utilizar, además estas computadoras tienen que soportar y almacenar toda la aplicación.

De esta manera se identifica como **problema a resolver**: ¿Cómo garantizar la recuperación de los materiales audiovisuales en el SCCM? El **objeto de estudio** de esta investigación es el flujo de actividades en el proceso de recuperación de información.

Para el desarrollo de la investigación se hizo necesario estudiar los procesos de recuperación de la información audiovisual del SCCM, lo cual constituyó el **campo de acción**, planteando como **idea a**

²Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

defender que la implementación del módulo de recuperación de materiales audiovisuales digitales, aplicando correctos procedimientos de desarrollo de *software*, garantizará la búsqueda de los audiovisuales en el sistema.

Esto trae consigo que el **objetivo general** sea realizar el diseño e implementación del módulo de recuperación de materiales audio-visuales para el SCCM.

Para que esta investigación cumpla con el objetivo propuesto se han trazado las siguientes tareas:

- ✓ Describir los procesos de recuperación básica, recuperación avanzada y recuperación por tesauros, a partir del estudio del estado del arte, para materiales audio-visuales.
- ✓ Identificar y describir las actividades de recuperación de materiales audio-visuales para el Sistema Captura y Catalogación de Medias.
- ✓ Seleccionar las herramientas y tecnologías a utilizar en la implementación del Módulo de recuperación de materiales audio-visuales para el Sistema Captura y Catalogación de Medias.
- ✓ Realizar el diseño del Módulo de recuperación de materiales audio-visuales para el Sistema Captura y Catalogación de Medias.
- ✓ Implementar el Módulo de recuperación de materiales audio-visuales para el Sistema Captura y Catalogación de Medias.
- ✓ Realizar pruebas de caja negra al Módulo de recuperación de materiales audio-visuales para el Sistema Captura y Catalogación de Medias.

Para un mejor entendimiento y realización de este trabajo de diploma se utilizaron los siguientes métodos científicos:

Métodos Teóricos

Analítico – sintético: Permitió realizar un análisis bien definido de las actividades involucradas en la recuperación de materiales audiovisuales digitales en el SCCM, ayudando a elaborar una síntesis de estos procesos de recuperación.

Análisis Histórico – lógico: Permitió realizar un estudio profundo de cómo han ido evolucionando los sistemas de recuperación en el mundo así como las principales características que presentan. Esto permitió organizar la información de forma lógica para así lograr un mejor entendimiento de esta investigación.

Modelación: Se utilizó para modelar el módulo de recuperación de materiales audio-visuales para el Sistema Captura y Catalogación de Medias.

Esta investigación ha sido estructurada de la siguiente manera:

Capítulo1: Introducción al estudio de la recuperación de materiales audiovisuales. Fundamentación teórica. Este capítulo abarca los conceptos esenciales relacionados con la recuperación de materiales audiovisuales. Se explican con más profundidad aspectos relacionados con la situación problemática planteada. Se describen los tipos de búsquedas que se emplearán en el sistema, así como algunas de las herramientas existentes en el mundo que realizan la recuperación de medias.

Capítulo2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales. En este capítulo se describen las tecnologías, los lenguajes de programación y metodologías de desarrollo que se escogieron, además se explicará detalladamente el porqué de su selección.

Capítulo3: Construcción de la solución. En este capítulo se describen los diagramas de clases y principios del diseño, la arquitectura del módulo, el estándar de codificación, los elementos sobre la base de datos a utilizar, así como el diagrama de despliegue y el modelo de implementación.

Capítulo4: Validación de la Solución Propuesta. En este capítulo se realizan las pruebas de caja negra a la solución creada en el capítulo anterior.

CAPÍTULO 1: INTRODUCCIÓN AL ESTUDIO DE LA RECUPERACIÓN DE MATERIALES AUDIOVISUALES. FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción

En este capítulo se abarcan los conceptos esenciales relacionados con la recuperación de materiales audiovisuales. Se explica con más profundidad aspectos relacionados con la situación problemática planteada. Se describen los tipos de búsquedas que se emplearán en el sistema, así como algunas de las herramientas existentes en el mundo que realizan la recuperación de medias.

1.2 Conceptos asociados al problema

Con la evolución imparable de las TIC y el desarrollo de los medios informáticos han surgido diferentes conceptos que son muy importantes y que su dominio les puede servir para lograr un mejor entendimiento del problema planteado. A continuación se describen los principales conceptos asociados al problema planteado.

1.2.1 Multimedia

Una multimedia no es más que la utilización en conjunto y al mismo tiempo de diferentes medios como imágenes, textos y sonidos, en la transmisión de una determinada información (DRAE, 2001). El término multimedia también es conocido como media o material audiovisual. Un material audiovisual se caracteriza por ser interactivo y dinámico. Estos pueden servir para especificar un tema y explicarlos por medio de imágenes y sonidos. La grabación, transmisión, percepción y comprensión de una media requiere habitualmente de un dispositivo tecnológico.

1.2.2 Búsqueda y recuperación

Los procesos de búsquedas y recuperación son el conjunto de tareas mediante las cuales el usuario localiza y accede a la información que necesita para resolver su problema (Tramullas, 2000). Estos abarcan los procesos encaminados a identificar, seleccionar y acceder a la información útil. La recuperación de un material audiovisual es la selección de la información del mismo, depositada en un medio de almacenamiento en respuesta a consultas realizada por usuarios (Herrera, 2006). Los sistemas de recuperación generalmente utilizan bases de datos y procesan las consultas realizadas por los usuarios, posibilitándole acceder a la información deseada en un tiempo apropiado.

Capítulo 1: Introducción al estudio de la recuperación de materiales audiovisuales.

Fundamentación teórica.

Los procesos de recuperación están compuestos principalmente por las siguientes fases (Figura 1):

- ✓ Necesidad de la información que desea el usuario: Define la información de acuerdo a su necesidad de conocimiento.
- ✓ Definición de la necesidad: Selección de las fuentes de información de acuerdo a su necesidad.
- ✓ Selección de recursos: Se define donde realizar dicha recuperación.
- ✓ Consulta de recuperación de la información: El usuario escoge un criterio de búsqueda asociado con la información que desea recuperar.
- ✓ Evaluación de los resultados: El sistema evalúa el criterio de búsqueda escogido por el usuario y realiza una consulta obteniendo los resultados.
- ✓ Presentación al usuario: El sistema le muestra todos los resultados encontrados al usuario.

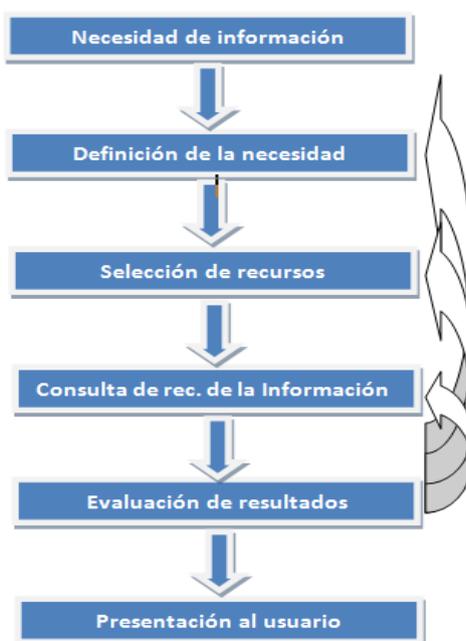


Figura 1: Fases del proceso de recuperación de información.

Existen otros conceptos asociados a la búsqueda y recuperación de materiales audiovisuales, estos serían la búsqueda básica, por contenido, avanzada, aplicando tesauros y combinada.

Búsqueda básica:

Cada material audiovisual tiene asociado un grupo de metadatos o etiquetas. Estos metadatos son datos muy bien estructurados que contienen información relevante de cada media en específico (Pérez, 2009).

Se define entonces como búsqueda básica toda aquella en la cual el usuario busca uno o varios valores en la base de dato y recupera todas las medias que contengan dichos valores. Estos valores son los relacionados con los metadatos de cada media.

Búsqueda por contenido:

Se define como búsqueda por contenido aquellas donde el usuario busca una media según la categoría a la cual pertenece. Las categorías están definidas según el tipo de material que sea; como es el caso de noticieros, deportes, educativos, entre otros.

Búsqueda avanzada:

La búsqueda avanzada está muy relacionada con los sistemas de recuperación basados en texto los cuales se encargan de buscar un conjunto de palabras en bases de datos (Pérez, 2009). Esta permite encontrar un grupo de términos aplicando diferentes criterios de búsquedas y permite aplicar filtros avanzados de fecha.

Búsqueda combinada:

La búsqueda combinada sería la relación de todas las búsquedas anteriores pero actuando al mismo tiempo, o sea, unir todos los criterios de la búsqueda básica, la avanzada, por clasificación y por tesauros y recuperar todas las medias relacionadas con dichos criterios.

Búsqueda por Tesauros:

La palabra tesoro viene derivado del neo latín que significa tesoro, se refiere al listado de palabras o términos empleados para representar conceptos. Los tesauros son vocabularios que presentan numerosos términos relacionados entre sí. Dichas relaciones ayudan a encontrar el vocablo que el usuario desea buscar y realizan una recuperación más amplia de la información.

La norma ISO 2788/TC 46 define tesoro desde dos puntos de vista:

- ✓ Según su función un tesoro es un instrumento de control de la terminología que se utiliza mediante la transposición del lenguaje natural a un lenguaje más estricto como es el documental.
- ✓ Según su estructura es un vocabulario controlado y dinámico de términos con relaciones semánticas entre ellos y que se aplican a campos temáticos particulares del conocimiento. (EcuRed, 2006).

1.3 Procesos básicos involucrados en la búsqueda y recuperación de materiales audiovisuales

1.3.1 Descripción general

A medida que las TIC fueron evolucionando, la información digital fue aumentando considerablemente. Cuando las computadoras fueron capaces de indexar grandes volúmenes de textos, se hizo necesario elaborar diversas estrategias para recuperar esa información. Para ello surgieron algunas técnicas que posibilitaron realizar la búsqueda y recuperación de forma eficiente.

1.3.1.1 Técnicas de recuperación de información

Hoy en día se aplican algunas técnicas que son muy importantes a la hora de realizar exitosamente la búsqueda y recuperación de una determinada información. Entre ellas están:

- **Sistemas de recuperación de lógica difusa:** Esta técnica permite realizar consultas con frases de forma tal que se eliminan los caracteres extraños que pueden aparecer en cualquier documento tradicional.
- **Técnicas de ponderación de términos:** Esta técnica se utiliza para darle un orden de prioridad a la búsqueda que desea realizar. Los documentos buscados dependen del valor obtenido en la ponderación. El valor depende de los términos que existan en el documento y la frecuencia con que estos se repiten. De tal forma, el documento más apropiado de búsqueda sería el que contenga todos los términos de búsqueda y además el que más valor tenga repetido.
- **Técnica de agrupación:** Este es un modelo probabilístico que le asigna valores a los documentos agrupándolos por un orden de importancia. Estos valores son asignados según la frecuencia con que los documentos son recuperados teniendo en cuenta el criterio de búsqueda utilizado.
- **Técnicas de retroalimentación por relevancia:** Consiste en que una vez que son identificados los documentos previamente recuperados como relevantes o irrelevantes por el usuario, se debe utilizar la información que proporcionan para adaptar la pregunta, de forma que se recuperen más documentos relevantes y menos irrelevantes.

Capítulo 1: Introducción al estudio de la recuperación de materiales audiovisuales.

Fundamentación teórica.

- **Técnicas de derivados:** Esta técnica lo que pretende es quitar las posibles confusiones semánticas que pueden aparecer en la búsqueda de cualquier información. Es por eso que elimina los prefijos y sufijos que pueden presentar las palabras.

1.3.1.2 Herramientas generales de recuperación

Existen numerosas herramientas generales que se usan para la recuperación de información en internet. Esta es la red de comunicación internacional por la cual circula una gran cantidad de información. Cuenta con un gran número de herramientas encargadas de recuperar esa información como son las revistas electrónicas. Estas no son más que los sitios web que los editores utilizan para acceder a los artículos incluidos en cada volumen de una publicación periódica. Las revistas electrónicas son publicaciones que aparecen en internet en soporte digital.

Internet también para la recuperación de información cuenta con los buscadores, estos, también denominados como motores de búsqueda, son sistemas basados en el uso de un *robot*³ o *software* que recorre la red automáticamente para localizar documentos, indexarlos e introducirlos en una base de datos, que puede ser consultada utilizando programas de consulta. Como ejemplos de esos buscadores son *Google*⁴, *Hotboot*⁵, *lycos*⁶, *Altavista*⁷, *Bing*⁸, *Yahoo*⁹ y otros más. Otra de las herramientas son los directorios, los cuales son una agrupación de archivos de datos, atendiendo a su contenido, a su propósito o a cualquier criterio que decida el usuario. Como ejemplo de directorio está *Internet Public Library*¹⁰.

En internet existen otras herramientas de recuperación como son los metabuscadores, los buscadores selectivos y los agentes inteligentes. Los metabuscadores son buscadores que no se limitan a recuperar la información en una única base de datos, realizan la búsqueda en diversas

³Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas.

⁴<http://www.google.com>

⁵<http://www.hotbot.com>

⁶<http://www.lycos.com>

⁷<http://www.altavista.com>

⁸<http://www.bing.com/>

⁹<http://www.yahoo.com/>

¹⁰<http://www.ipl.org>

bases de datos. Como ejemplos de estos son *Kartoo*¹¹ y *Vivísimo*¹². Los buscadores selectivos utilizan una base de datos especializada en una materia en específico, ejemplos de estos son *Electric Library*¹³, *Teoma*¹⁴ y otros más. Los agentes inteligentes son herramientas que recuperan la información de forma automática, solo necesitan un criterio de búsqueda y el lugar donde van a realizar la recuperación. Ejemplos de estos son *Webferret*¹⁵ y *WebSeeker 5*¹⁶.

1.4 Descripción actual del dominio del problema

Actualmente la mayoría de los programas creados por el departamento de Señales Digitales de la facultad 6 en la Universidad de las Ciencias Informáticas trabajan con audiovisuales. SCCM quiere desarrollar un *software* con un grupo de funcionalidades que garanticen la gestión, edición, catalogación y préstamo de las medias. Para que estos procesos funcionen correctamente necesitan un módulo de recuperación de materiales audiovisuales que permita buscar y recuperar medias en una base de datos.

La búsqueda y recuperación de las medias en la base de datos se quiere que sea de cuatro formas, la búsqueda básica, por contenido, avanzada y por tesauros. La búsqueda básica se debe hacer de forma tal que cuando el usuario consulta en la base de datos por algún metadato específico de la media, por ejemplo, título de la media, título del programa, duración de la media, entre otros; recupere todas las medias que contengan el valor especificado por el usuario.

La búsqueda por contenido se quiere hacer de forma tal que le posibilite al usuario realizar la recuperación de las medias teniendo en cuenta la categoría a la cual pertenece. Las categorías están dadas según el tipo de material que sea.

La búsqueda avanzada se quiere hacer definiendo filtros avanzados para el caso de la fecha y de los datos generales. En el caso de la fecha, que el usuario pueda recuperar todas las medias en un intervalo determinado u omitiendo algún parámetro de la fecha establecida como puede ser el día, el mes o el año. En el caso de los datos generales, que el usuario pueda definir el texto, el ámbito y el nivel de precisión de la búsqueda.

¹¹<http://www.kartoo.com>

¹²<http://www.vivisimo.com>

¹³<http://www.elibrary.com>

¹⁴<http://www.teoma.com>

¹⁵<http://www.ferretsoft.com>

¹⁶<http://www.bluesquirrel.com>

Se define texto a la frase que se desea buscar. El ámbito es la forma en que el usuario desee buscar dicho texto. Este ámbito podrá ser de cuatro variantes. La primera sería con alguna de las palabras, refiriéndose a que las medias recuperadas deberán tener en los datos generales al menos una de las palabras especificadas en el texto que el usuario definió. La segunda sería con todas las palabras, refiriéndose a que los materiales recuperados deberán tener en los datos generales, todas las palabras especificadas en el texto antes mencionado. La tercera variante es sin las palabras, refiriéndose a que las medias que se recuperen no deben tener ninguna de las palabras del texto en los datos generales. Por último, la cuarta variantes sería con la frase exacta refiriéndose a que los materiales recuperados deberán contener en los datos generales, la frase especificada en el texto, tal y como está definida. En el nivel de precisión el usuario podrá escoger si desea que el texto es sensible o no a los errores ortográficos.

La búsqueda empleando tesauros debe permitir realizar la recuperación de materiales audiovisuales cuyos metadatos están relacionados de una manera especial con uno o varios términos del tesoro. El usuario en caso de que esté inseguro a la hora de realizar la búsqueda, tendrá la posibilidad de buscar uno o varios términos en diferentes tesauros, brindándoles conceptos más generales definidos por las relaciones que presentan dichos vocablos en los tesauros.

1.5 Situación problemática

Con el gran aumento y uso de los materiales audiovisuales en el mundo, ha surgido la necesidad de hacer cada vez más eficientes las herramientas de gestión de estos. La búsqueda y recuperación de estos materiales la han implementado varias compañías como es el caso de Tedral, empresa española que se dedica a la gestión completa de materiales audiovisuales. El principal problema radica en que los productos de dicha empresa son privativos, lo que implica un alto costo de adquisición.

El departamento de Señales Digitales de la facultad 6 en la UCI está desarrollando un software con el objetivo de darle solución a dicha problemática. Dicho departamento cuenta con una versión elaborada con tecnologías de escritorio. Esta versión tiene un módulo de búsqueda y recuperación de medias que no satisface los requerimientos funcionales. Los criterios de búsqueda que brinda son muy pobres y los usuarios quedan inconformes, además los obliga a tener un conocimiento avanzado en el negocio del sistema para poder recuperar los materiales audiovisuales. Otro inconveniente es a la hora del despliegue ya que hay que instalarlo en cada una de las computadoras donde se vaya a utilizar y estas

computadoras tienen que cumplir con requisitos mínimos indispensables para que la aplicación funcione.

El software se está elaborando sobre tecnologías web completamente libres. De esta forma se facilita el despliegue de la aplicación. La herramienta no permite la recuperación de materiales audiovisuales, trayendo consigo que a los usuarios de la aplicación les resulte engorroso la búsqueda de las medias. Es por eso que se hace necesaria la implementación de un módulo de recuperación para agilizar los procesos que se quieren automatizar.

1.6 Análisis de otras soluciones existentes

En la actualidad existen diversos software que se encargan de la gestión de materiales audiovisuales y que desarrollan una fuerte recuperación de los mismos. El único inconveniente es que son propietarios y el país no puede acceder a ellos debido a los altos costos de adquisición. A continuación se pueden apreciar algunas de estas herramientas así como sus principales características.

1.6.1 Mediabox

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 (XTREAM 2010 , 2010). La principal desventaja de este producto es su dependencia con otros software privativos y que no permite la edición de una media audiovisual.



Figura 2: Mediabox (XTREAM 2010, 2010).

1.6.2 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. 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.



Figura 3: TdtmTarsys, empresa Tediol (NAB11, Tediol, 2011).

1.6.3 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 (Baby González, Ronald; Almeida Maldonado, Enrique, 2009).

Contiene un módulo básico de gestión y recuperación de contenido que es el motor principal de la plataforma Videoma. 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. Tampoco permite la edición de las medias audiovisuales sino que se comunica con editores externos como el Adobe Premiere.



Figura 4: Videoma (ISID Gestión Inteligente de Activos Multimedia, 2009).

1.7 Conclusiones Parciales

En el presente capítulo se ha descrito la situación problemática, los procesos básicos involucrados en la búsqueda y recuperación de materiales audiovisuales y los tipos de búsqueda establecidos en el proyecto Sistema de Captura y Catalogación de Medias. Se puede decir que los sistemas de gestión de materiales audiovisuales para que sean robustos, necesitan una fuerte recuperación de medias. La evolución de estos sistemas ha sido directamente proporcional con el desarrollo de internet y la digitalización, puesto que cada día se acumulan más información audiovisual y la complejidad de su

Capítulo 1: Introducción al estudio de la recuperación de materiales audiovisuales.

Fundamentación teórica.

gestión se hace cada vez mayor. El surgimiento de las técnicas para recuperar información les posibilitan a estos sistemas agilizar los procesos de búsqueda. Existen diversos software que realizan la gestión de materiales audiovisuales pero no son aconsejables puesto que son soluciones privativas y presentan un alto nivel de adquisición.

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS A UTILIZAR PARA REALIZAR LA RECUPERACIÓN DE MATERIALES AUDIOVISUALES.

2.1 Introducción

En este capítulo se describen las tecnologías, los lenguajes de programación y metodología de desarrollo escogida, además se explica detalladamente el porqué de su selección.

2.2 Tendencias y tecnologías web

En la actualidad cuando se abordan temas sobre tendencias y tecnologías a utilizar para el desarrollo de algún *software* o algún componente de recuperación, no se puede dejar de mencionar las relacionadas con el entorno *web*. Hay que 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 éste mediante un navegador *web*. Debido a esto, la información quedaría almacenada en un solo ordenador y la administración sería de forma centralizada.

2.2.1 Estilo arquitectónico Cliente – Servidor

El esquema cliente-servidor "es un modelo de computación en el que el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones relacionadas se divide entre dos o más procesos que cooperan entre sí " (Ronda Amador, et al., 2002) (Datapro Client/Server Analyst, 1994). Dicho modelo permite el acceso transparente a las aplicaciones, datos o cualquier otro recurso en múltiples plataformas. Está compuesto principalmente por los clientes que serían los consumidores de servicios, los servidores que serían los proveedores de servicios y la infraestructura de comunicaciones.

Los clientes interactúan con los usuarios generalmente a través de interfaces compuestas de diferentes elementos que pueden ser formularios, tablas, gráficas, entre otros. Son los encargados de pedirle información a los servidores a través de un grupo de procesos auxiliares, que son responsables de hacer el pedido para satisfacer las demandas del cliente, además de garantizar la seguridad y protección.

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

Los servidores son los encargados de satisfacer la demanda de los clientes devolviendo los resultados a sus peticiones. Están conectados con los consumidores a través de redes, para proveer múltiples servicios. Además de proporcionar este tipo de recursos, pueden dar acceso a otras redes, actuando como servidores de comunicaciones que conectan a otros servidores.

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, la cual proporciona los mecanismos básicos de direccionamiento y transporte (Ronda Amador, et al., 2002). Hay que destacar también que el cliente y el servidor pueden actuar como la misma entidad, realizando actividades independientes. Los servidores pueden brindar servicios a múltiples clientes concurrentemente y pueden estar en distintas plataformas. Los cambios que se realicen ya sea en la plataforma cliente o en la servidora, son transparentes al usuario final.

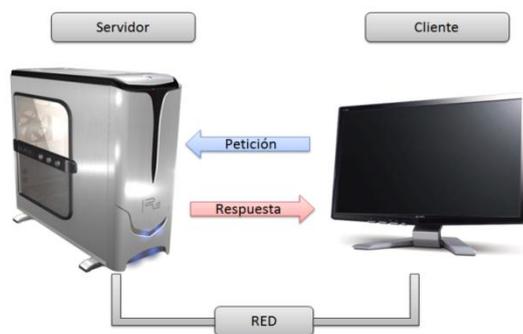


Figura 5: Estilo Arquitectónico Cliente-Servidor.

2.2.2 Metodología de desarrollo de software

El desarrollo de un *software*, por sencillo que sea, es una tarea difícil. Como resultado de esto fue que surgieron las metodologías de desarrollo de *software*. Estas no son más que un conjunto de pasos y procedimientos que se deben tener en cuenta a la hora de desarrollar *software*. Son aplicables a la elaboración de cualquier tipo de proyecto, aunque toman mayor sentido si éste tiene mayor envergadura.

Las metodologías se caracterizan por plantear un grupo de pasos sobre la elaboración de un *software* con el fin de que al final, el resultado sea predecible y eficiente. Su objetivo principal es elevar la calidad del producto final. Existen diferentes metodologías aplicadas a diferentes

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

tendencias de desarrollo de *software*, tal es el caso de Proceso Unificado de Desarrollo o *Rational Unified Process* (RUP).

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

El Proceso Unificado de Desarrollo (RUP), es el proceso de desarrollo de *software* más utilizado para la creación de proyectos a largo plazo. Éste, en conjunto con el Lenguaje Unificado de Modelado (UML), se ha convertido en un estándar a nivel internacional debido a la gran cantidad de empresas e instituciones que lo utilizan. Puede ser adaptado a una gran variedad de sistemas, sin importar el tamaño ni el nivel de aptitud.

RUP presenta tres características fundamentales, dirigidas por casos de uso: da la posibilidad al usuario de pensar en términos importantes para él y no solo en funciones que debería tener la solución final; centrado en la arquitectura: está relacionada con la toma de decisiones que indican cómo debe ser construido el sistema y en qué orden, además debe considerar la calidad, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible en todo el proceso de desarrollo; y es iterativo e incremental: divide el trabajo en partes más pequeñas o mini proyectos, de esta forma permite que el equilibrio entre casos de uso y arquitectura se vaya logrando durante cada mini proyecto en todo el proceso de desarrollo, cada mini proyecto se puede ver como una iteración del cual se obtiene un incremento que produce un crecimiento en el producto (Letelier, 2008).

RUP presenta cuatro fases de desarrollo de *software*:

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*).

Transmisión: El objetivo de esta fase es obtener el *release* del proyecto.

Hay que destacar también que RUP hace uso de arquitecturas basadas en componentes, enfocándose en el desarrollo temprano de una arquitectura base ejecutable. Dicha arquitectura debe ser flexible, fácil de modificar, comprensible intuitivamente y promueve una efectiva reutilización de código. Además apoya el desarrollo basado en componentes.

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

RUP describe cómo obtener, organizar y documentar los requerimientos funcionales y las restricciones. También permite documentar las decisiones y comunicar requerimientos del negocio. Muestra cómo hacer el modelamiento visual de la estructura, además del comportamiento de la arquitectura y los componentes. Los bloques de construcción permiten ocultar detalles, comunicación entre los desarrolladores y analizar la consistencia entre componentes, así como el diseño e implementación (Guerrero, 1999). Todo el modelamiento visual que desarrolla es sobre la base de UML.

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

Modelamiento del negocio: Describe 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 utilizará el equipo de trabajo del proyecto.

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

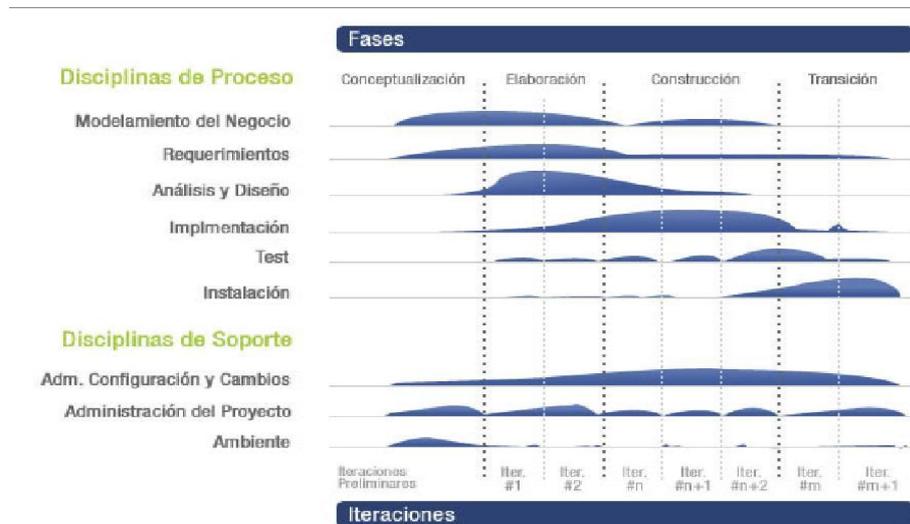


Figura 6: Fases y Flujos de Trabajos según RUP (Molpeceres, 2002).

Por todas estas características mencionadas y debido al alcance del *software* que se desea realizar, se seleccionó RUP debido a que aporta todos los elementos necesarios para elaborar proyectos grandes que requieran mucha documentación.

2.2.4 Lenguaje de Modelado: UML

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 popular para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente, de diseño (Orallo, 2010).

Dicho lenguaje mejora los tiempos totales de desarrollo permitiendo modelar sistemas utilizando conceptos orientado a objetos. Posibilita establecer conceptos y artefactos ejecutables y encamina el desarrollo escalable en sistemas complejos. También brinda un mejor soporte a la planeación y al control de proyectos permitiendo una alta reutilización y minimización de costos.

2.2.5 Herramientas CASE

Las herramientas *CASE* son aplicaciones informáticas con el objetivo de representar objetos de datos de negocio, sus relaciones, y proporcionan un mejor entendimiento acerca de cómo fluye el flujo de datos en el negocio. Estas herramientas tienen como objetivo, facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones, simplificar el mantenimiento de los programas,

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

mejorar y estandarizar la documentación, facilitar la reutilización de componentes de *software* y permitir un desarrollo y refinamiento visual de las aplicaciones utilizando gráficos.

2.2.5.1 Visual Paradigm

Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado. Soporta el ciclo de vida completo del desarrollo de *software*. Permite dibujar todo tipo de diagramas de clase, código inverso, generar código desde diagramas y generar documentación. Esta herramienta provee el modelado de procesos de negocio, además que permite generar un mapeo de objetos relacionales para diferentes lenguajes de programación.

Presenta unas características gráficas muy cómodas que facilitan la realización de los diagramas de modelado que siguen el estándar de UML que son: diagramas de clase, casos de uso, comunicación, secuencia, estado, actividad, componentes, entre otros. Otra característica importante que presenta es la integración con IDE's de desarrollo como son *NetBeans*, *JDeveloper*, *Eclipse* y *JBuilder*. Permite la ingeniería inversa de *JAVA*, *.NET*, *XML* e *Hibernate*. Además presenta una licencia gratuita y comercial (Altamirano, 2009).

Visual Paradigm es un producto de excelente calidad que soporta aplicaciones *web*, es fácil de instalar y actualizar, presenta varios idiomas, tiene un entorno de creación de diagramas para UML 2.1, disponibilidad en múltiples plataformas y de múltiples versiones para cada necesidad. Facilita la comunicación entre el equipo de desarrollo ya que hace uso de un lenguaje estándar común.

Por todas estas características antes mencionadas es que se escoge como herramienta CASE a *Visual Paradigm* para el trabajo con la ingeniería de *software*. Esta herramienta es muy potente para propósitos ingenieriles y posee versiones multiplataforma. Además que es completamente compatible con RUP y genera la documentación del *software* de forma eficiente y con alta calidad.

2.2.6 Lenguaje de Programación

Los lenguajes de programación se crearon para que las personas pudieran darles instrucciones a los equipos, describiendo un conjunto de acciones que estos deben cumplir. Dichas instrucciones están compuestas de un grupo de reglas gramaticales, términos y símbolos que le permiten al

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

desarrollador comunicarse con el equipo y crear programas, además pueden ser utilizados para controlar el funcionamiento de los ordenadores.

En la actualidad existen diversos lenguajes para desarrollar en la *web* y se encuentran divididos principalmente en dos grupos, los lenguajes del lado del cliente y los lenguajes del lado del servidor. Los lenguajes del lado del cliente se utilizan principalmente en la validación de los datos y en la estética de la interfaz. Los lenguajes del lado del servidor son los encargados de comunicarse de forma segura y eficiente con el servidor, enviándole de forma dinámica los datos recogidos en la interfaz, y a su vez, la capacidad de respuesta por parte del servidor.

2.2.6.1 Lenguajes del lado del cliente

➤ JavaScript:

El lenguaje *JavaScript* es un lenguaje interpretado que no requiere de compilación. No es un lenguaje orientado a objetos, por lo que no presenta la herencia y el polimorfismo. Este lenguaje se caracteriza principalmente por ser fiable y seguro, el código se ejecuta del lado del cliente y es descifrado por el navegador.

Permite definir efectos especiales en las páginas y mantiene una alta interacción con el usuario. Es un lenguaje sencillo y pensado para desarrollar páginas con estética y rapidez. También es utilizado para crear contenidos dinámicos, elementos con animaciones, cambiar colores, crear menús iterativos, entre otros. Permite la programación desde pequeños *scripts* hasta programas más grandes con funciones y estructuras de datos complejas. Permite acceder a todos los elementos de la página *web* y realizar una fuerte validación de los datos de entrada de los clientes.

➤ AJAX:

Se utiliza *AJAX* como método para la comunicación con la base de datos. *AJAX* es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "*JavaScript* asíncrono + *XML*". Es la unión de varias tecnologías *web* que tienen como objetivo crear aplicaciones interactivas. Esta técnica se ejecuta en el lado del cliente de forma asíncrona con el servidor, posibilitando que la página *web* no se recargue nuevamente cuando se quiera hacer un cambio en el entorno, permitiendo así que las páginas sean mucho más eficientes, rápidas e interactivas.

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

➤ **Dojo Toolkit 1.6:**

Se utiliza la librería *Dojo Toolkit 1.6* escrita en el lenguaje *JavaScript* que le brinda a los desarrolladores de páginas *web* un grupo de funcionalidades importantes. Dichas funcionalidades posibilitan el embellecimiento y dinamismo a las páginas, además mejoran el diseño y la usabilidad y son compatibles en la mayoría de los navegadores.

Entre algunas de sus características está la manipulación de *DOM (Document Object Model)*, animaciones y contiene *API's* para el trabajo con *Ajax*. *Dojo* presenta un grupo de componentes reutilizables (*widgets*), posibilita el manejo de eventos y soporte gráfico para vectores en segunda y tercera dimensión. Posibilita a los desarrolladores crear sus propios *widgets* e incluso modificar los existentes.

➤ **HTML 5:**

El *HyperText Markup Language (HTML)* es el lenguaje que predomina en la creación de la mayoría de las páginas *web*. Dicho lenguaje describe la estructura y el contenido en forma de texto. Los documentos *HTML* son hipertextos en los cuales aparecen enlaces a otros documentos. Presenta un estándar de código bien conocido por la mayoría de las empresas relacionadas con internet. Esto posibilita que se visualice de forma similar en muchos de los navegadores sin importar el sistema operativo que esté instalado en la computadora.

Permite crear lenguajes de codificación descriptivos. Define una estructura de documentos jerárquica, con elementos y componentes interconectados. Proporciona una especificación formal completa del documento. No tiene un conjunto implícito de convenciones de señalización. Soporta, por tanto, un conjunto flexible de juegos de etiquetas. Los documentos generados por él son legibles (Martín, 2001).

➤ **Cascading Style Sheets (CSS):**

Las Hojas de Estilo en Cascada (*Cascading Style Sheets*), es un mecanismo simple que describe cómo se va a mostrar o a imprimir un documento en la pantalla, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos (W3C (MIT, ERCIM, Keio), 2008).

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

CSS se utiliza para dar estilo a documentos *HTML*, separando el contenido de la presentación. CSS permite a los desarrolladores *web* controlar la estética y el formato de múltiples páginas *Web* al mismo tiempo.

2.2.6.2 Lenguajes del lado del servidor

➤ PHP Hypertext Pre-processor(PHP 5):

Personal Home Page (PHP 5) es un lenguaje de programación del lado del servidor utilizado en aplicaciones *web*. Está ampliamente difundido gracias a su código abierto y no causa costos extras para *software* del servidor. Se utiliza principalmente para la elaboración de sitios *web* grandes que tengan una fuerte administración y consultas a base de datos. Esto es posible debido a que es un lenguaje fácil y eficiente.

A diferencia de *JavaScript*, *PHP 5* se ejecuta en el servidor y el resultado es enviado al navegador generalmente en forma de *HTML*. Permite acceder a los recursos que tenga el servidor. La mayoría de los Sistemas de Administración de Contenidos (CMS) de código abierto están desarrollados con este lenguaje. Presenta una sintaxis flexible y es sencillo de utilizar, además brinda la posibilidad de conectarse a la mayoría de los gestores de base de datos. Incluye la implementación de muchas funciones que agilizan su interpretación y no presenta tipos de datos. Es un lenguaje libre por lo que representa una alternativa de acceso.

2.2.7 Framework

Un *framework* es una estructura *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 (Gutiérrez., 2008).

Se podría decir que un *framework* no es más que un esquema de una aplicación que comprende una organización estructural y enfocada a la reutilización a gran escala de sus clases con el objetivo de acelerar el proceso de desarrollo de *software*. Tiene como objetivos principales: realizar el proceso desarrollo de *software* de forma rápida y eficiente, utilizar buenas prácticas de desarrollo; como es el caso del uso de patrones y reutilizar el código existente.

✓ Symfony 1.4.6

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

Symfony 1.4.6 es un *framework PHP* que facilita el desarrollo de las aplicaciones *web*. Se encarga de todos los aspectos comunes y aburridos de dichas aplicaciones, dejando que el programador se dedique a aportar valor desarrollando las características únicas de cada proyecto (Eguiluz, 2011).

Es un *framework* muy documentado, productivo, elegante y muy estable. Fácil de instalar en diferentes sistemas operativos, funciona con todos los gestores de base de datos comunes, ampliamente flexible y extensible mediante un completo sistema de *plugins*, traducido a diversos idiomas, publicado bajo licencia de *software* libre *GNU Lesser Public General License (LGPL)* y compatible con *PHP 5*.

Además sigue las mejores prácticas de desarrollo y patrones de diseño para la *web*. Contiene un código legible, fácil de entender, incluye comentarios y permite un mantenimiento sencillo. Permite el uso de plantillas en la capa de presentación, validación de datos y generación dinámica de formularios, es muy potente contra las inyecciones *SQL*, contiene un eficiente sistema de caché, paginación automática, facilita la integración con *Ajax* mediante *helpers*, los cuales encapsulan código *JavaScript* en una sola línea.

Dicho *framework* emplea el patrón arquitectónico modelo-vista-controlador (MVC), con el objetivo de organizar las distintas partes que componen una aplicación *web*, separándola en tres capas principales: el modelo, la vista y la controladora. La versión de *symfony* que se propone elimina *Propel* e incorpora *Doctrine* para realizar el mapeo de objetos a las base de datos (*ORM*) (*object-relational mapping*). Esto es muy importante puesto a que *Doctrine* tiene una amplia documentación, es muy configurable, permite generar automáticamente el modelo de clases basándose en el modelo relacional de tablas, además posibilita crear por el mismo usuario el mapeo de la base de datos y las relaciones entre sus tablas. Facilita la realización de búsquedas por cualquier campo de una tabla de la base de datos y posibilita la realización de consultas en un lenguaje orientado a objeto llamado *DQL*.

Por todas estas características que presenta *Symfony* es que se escoge como *framework PHP* para el desarrollo del proyecto SCCM. *Symfony* ya ha sido probado en numerosos proyectos

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

reales como es el caso de *Yahoo*, *Answers*¹⁷, *Delicious*¹⁸, entre otros. Además que presenta una amplia documentación.

2.2.8 Entorno de Desarrollo Integrado (IDE)

Un Entorno de Desarrollo Integrado (IDE) es una aplicación o una parte de ella que integra diferentes funciones de edición, compilación, interpretación, depuración del código además de una interfaz de edición en un entorno común. Un IDE de desarrollo puede ser específico para un lenguaje de programación o puede incorporar más de uno.

✓ NetBeans 7.1

NetBeans 7.1 es un IDE de desarrollo que se adapta a un gran número de lenguajes de programación. Funciona en los sistemas operativos compatibles con la máquina virtual de *Java*. Contiene un grupo de características importantes como es el caso de la integración con *PHP*.

Su editor integra lenguajes como *JavaScript*, *HTML* y *CSS*. Posee un sistema de escaneo de todos los proyectos que se crean o se cargan, para así reconocer funcionalidades, clases, y proporcionar un completamiento de código eficiente, que acelera la velocidad en la elaboración de aplicaciones.

Presenta una fácil integración con *Symfony* que agiliza la elaboración de proyectos. Posee una interfaz visual para la instalación de dicho *framework*. Tiene un fuerte editor de código fuente que provee todas las ventajas que presenta *PHP*, además de ser robusto y ágil.

Ofrece líneas de comandos para la depuración posibilitando inspeccionar el código *HTML* sin tener que cambiar a un navegador. Permite una integración completa en términos de administración de *PostgreSQL*. También posee una eficiente y ágil integración con sistemas de control de versiones. Es gratuito y de código abierto.

Todas esas características posibilitaron que el proyecto SCCM escogiera a *NetBeans* como IDE de desarrollo para la elaboración del módulo de recuperación.

¹⁷ <http://es.answers.yahoo.com/>

¹⁸ <http://www.delicious.com/>

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

2.2.9 Sistemas Gestores de Base de Datos

Un sistema de gestión de bases de datos (SGBD; en inglés, *Database Management System: DBMS*) es un conjunto de programas que permite a los usuarios crear y mantener una base de datos. Si bien, no es imprescindible contar con un SGBD para implementar una base de datos, este *software* de uso general facilita el proceso de definir, construir y manipular bases de datos para diversas aplicaciones (Pereyra, 2005).

Estos sistemas tienen como objetivo eliminar la inconsistencia de los datos, compartir dichos datos entre distintas aplicaciones con diferentes necesidades, ahorrar espacio al no existir tanta redundancia, mejorar la seguridad, crear entornos de alta disponibilidad y mejorar servicios de copia de seguridad y recuperación de fallos.

✓ PostgreSQL 9.0

PostgreSQL 9.0 es un avanzado sistema de bases de datos relacionales basado en *Open Source*. Esto quiere decir que el código fuente del programa está disponible a cualquier persona libre de cargos directos, permitiendo a cualquiera colaborar con el desarrollo del proyecto o modificar el sistema para ajustarlo a sus necesidades. *PostgreSQL 9.0* está bajo licencia *BSD* (Denzer, 2002).

Dicho gestor implementa el estándar SQL92/SQL99, incorpora la estructura de datos de arreglos, además de soportar diversos tipos de datos e incluso deja definir uno si es necesario. Permite la definición de disparadores y funciones propias, soporta el uso de vistas, reglas e índices. Incluye herencia entre tablas y permite la gestión de diferentes usuarios así como los permisos asignados a cada uno de ellos.

Hay que destacar que *PostgreSQL 9.0* es estable, de gran flexibilidad, eficiente y de alto rendimiento. Permite migrar bases de datos que estén hechas en otras herramientas para que lo utilicen como gestor de estas. Utiliza una arquitectura cliente servidor y usa multiprocesos. Esto es muy importante porque de afectarse algún proceso determinado, los restantes no se afectan.

Estas fueron las características que posibilitaron que el proyecto SCCM se inclinara por escoger *PostgreSQL* como gestor de base de datos. Además por estar basado en un sistema de código abierto, representa una alternativa muy importante en desarrollo de *software*.

Capítulo 2: Tendencias y tecnologías a utilizar para realizar la recuperación de materiales audiovisuales.

2.3 Conclusiones Parciales

Con el desarrollo imparable de las TIC, la *web* se ha convertido en el entorno que muchos usuarios y desarrolladores prefieren. Han surgido numerosas técnicas y herramientas que posibilitan su desarrollo y es de vital importancia para la producción, la correcta selección de estas. En el presente capítulo se escogieron las tecnologías y herramientas que se utilizaron para la elaboración del módulo de recuperación de materiales audio-visuales. Hay que destacar que estas tecnologías son actuales y las más usadas en el desarrollo *web*, que con su correcto uso se podrán alcanzar *software* que cumplan con la calidad requerida. Además con el desarrollo del *software* guiado por *RUP*, utilizando como *IDE* de desarrollo *NetBeans 7.1*, vinculado con el *framework* para *PHP Symfony 1.4.6*, y también utilizando *Dojo 1.6* como librería *JavaScript* en conjunto con *HTML 5* y *CSS*, se generarán diferentes documentos y artefactos que permitirán un entendimiento del módulo a desarrollar.

CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN.

3.1 Introducción

La implementación del módulo de recuperación de materiales audio-visuales para el Sistema Captura y Catalogación de Medias, está basada en un grupo de requisitos funcionales realizados con anterioridad por el proyecto SCCM. Dichos requisitos y sus respectivas descripciones se podrán consultar en los anexos del 1 al 4. En este capítulo se tratará la solución propuesta donde se describen los diagramas de clases y principios del diseño, la arquitectura del módulo, así como elementos sobre la base de datos a utilizar.

3.2 Estilos Arquitectónicos

Los estilos arquitectónicos determinan la forma en que van a estar constituidos los componentes, así como la forma en que se comunicarán. También definen las restricciones que se imponen en la comunicación entre ellos.

Se define estilo arquitectónico a una familia de sistemas de *software* en términos de un patrón de organización estructural, que define un vocabulario de componentes y tipos de conectores y un conjunto de restricciones de cómo pueden ser combinadas. Para muchos estilos puede existir uno o más modelos semánticos que especifiquen cómo determinar las propiedades generales del sistema partiendo de las propiedades de sus partes (Camacho, y otros, 2004). Tiene como objetivo principal establecer una estructura para todos los componentes que realizan una función requerida por el sistema y determina las relaciones entre ellos, rigiéndose por las restricciones que definen cómo se integran dichos componentes para formar el sistema, así como las reglas para su construcción (Camacho, y otros, 2004)(Pressman, 2010).

En el capítulo 2 se seleccionó *Symfony* como el *framework* de *PHP* a utilizar en la elaboración del módulo de recuperación. Dicho *framework* está desarrollado bajo las restricciones del estilo arquitectónico **Llamada y Retorno**, el cual plantea que el sistema se constituye de un programa principal que tiene el control del sistema y varios subprogramas que se comunican con éste mediante el uso de llamadas (Camacho, y otros, 2004). Este estilo permite que un diseñador de *software* obtenga una estructura del sistema que resulta relativamente fácil de modificar (Pressman, 2010).

3.2.1 Patrones Arquitectónicos

Los patrones arquitectónicos expresan el esquema de organización estructural fundamental para sistemas de *software*. Provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. Propone que son plantillas para arquitecturas de *software* concretas, que especifican las propiedades estructurales de una aplicación y tienen un impacto en la arquitectura de subsistemas (Camacho, y otros, 2004).

Un patrón arquitectónico es una solución probada a un problema determinado. Permite la reutilización de soluciones a problemas comunes. Además son esqueletos genéricos que los diseñadores adaptan a su aplicación. *Symfony* está basado en un patrón clásico del diseño *web* conocido como **Modelo Vista Controlador (MVC)**.

MVC plantea la separación del *software* en tres elementos fundamentales: el modelo, la vista y el controlador. El modelo es la representación de la información que maneja la aplicación, lo constituyen los datos puros y la lógica de los propios datos que puestos en el contexto del sistema proveen de información al usuario y en algunos casos a la propia aplicación. La vista es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación *web*, la “Vista” sería una página *HTML* o *PHP* con contenido dinámico sobre la cual el usuario puede realizar sus operaciones. El controlador es la parte encargada de manejar y responder las solicitudes del usuario, procesando toda la información necesaria y modificando el modelo en caso de ser necesario (Vega Ortiz, et al., 2010).

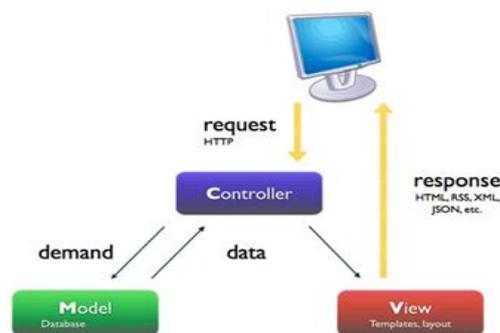


Figura 7: Modelo-Vista-Controlador (Potencier, 2009)

3.2.1.1 Patrones de Diseños

Los patrones de diseño proveen esquemas para refinar los subsistemas o componentes de un sistema de *software*, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular (Camacho, y otros, 2004).

Symfony sigue las mejores prácticas de patrones de diseño para la *web*. A continuación se mencionarán algunos de los patrones que dicho *framework* implementa (Vega Ortiz, et al., 2010):

Patrón creador: En las clases controladoras se encuentran las acciones definidas para las operaciones lógicas del negocio en cuestión y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, esto evidencia que la clase *Actions* es "creadora" de dichas entidades. En el caso del módulo de recuperación se puede ver como desde la clase *RecuperacionActions* se crean instancias de las clases *ModeloRecuperacion* y *Tesauo*.

Patrón controlador: Todas las peticiones *web*, generalmente, son manejadas por un solo controlador frontal, que es el punto de entrada único de toda la aplicación para un entorno determinado. El controlador es encargado de recibir la petición del usuario y en función de ella, envía la solicitud a las distintas clases para que sea procesada. En la implementación del módulo se puede observar como la clase *RecuperacionActions* se encarga de controlar todas las solicitudes que se introduzcan en el sistema.

Patrón estructural decorator (envoltorio): Añade funcionalidad a una clase, dinámicamente. El archivo *layout.php*, 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 el contenido de la plantilla.

Patrón Experto: Es uno de los patrones de diseño tenido en cuenta, plantea asignarle la responsabilidad a la clase que tiene la información necesaria para cumplir con dicha responsabilidad. Un ejemplo donde se presencia la utilización de este patrón es en las clases que constituyen componentes como *Thesaurus_Search_Component*, es aquí donde se realiza la búsqueda empleando tesauros, por tanto todas las funcionalidades referentes a esa búsqueda están implementadas ahí.

3.3 Diagramas de Clases del Diseño.

Los diagramas de clases del diseño describen como quedarían estructurados los sistemas, mostrando las clases con sus atributos, funciones y las relaciones entre ellas. Muestran gráficamente las especificaciones de las clases de *software* y de las interfaces en una aplicación (Visconti, y otros, 2010). Las clases del diseño se van a agrupar de acuerdo a los casos de uso del sistema, siguiendo el estilo arquitectónico Modelo Vista Controlador. A continuación se muestran los diagramas de clases del diseño referente a cada uno de los casos de uso.

3.3.1 Diagrama de clases del diseño para el caso de uso Buscar Material a Catalogar

El caso de uso buscar material a Catalogar comienza cuando el documentalista desea buscar una o varias medias en el servidor. El peso fundamental recae sobre las clases:

- ✓ **SCCM_Search_Component:** Dicha clase sirve de núcleo para el componente de búsqueda. Presenta el atributo *required_data_for_inner_components* el cual agrupa los datos referente a como se va a visualizar el componente. Esta información es recogida en la base de datos y pasada al componente para que este se muestre de forma dinámica. También posee el método *Execute_Medias_Not_Catalogued_Yet_Search ()* el cual se encarga de iniciar el proceso de búsqueda.
- ✓ **Media_To_Catalog_Search_Component:** Recoge los criterios de búsqueda del documentalista a través del método *Get_Data_For_Search ()*.
- ✓ **Result_Grid_Component:** Es la encargada de visualizar los resultados de la búsqueda.
- ✓ **SCCM_Search_ComponentForm:** Presenta los campos pertinentes para determinar la fecha en que fue insertada la media al servidor.
- ✓ **RecuperacionActions:** Esta es la clase controladora que con el método *executeSearch_ajax_core ()*, se encarga de realizar la petición de búsqueda al modelo.
- ✓ **ModeloRecuperacion:** Se encarga de realizar las consultas a la base de datos para obtener el resultado según los filtros seleccionados por el documentalista.

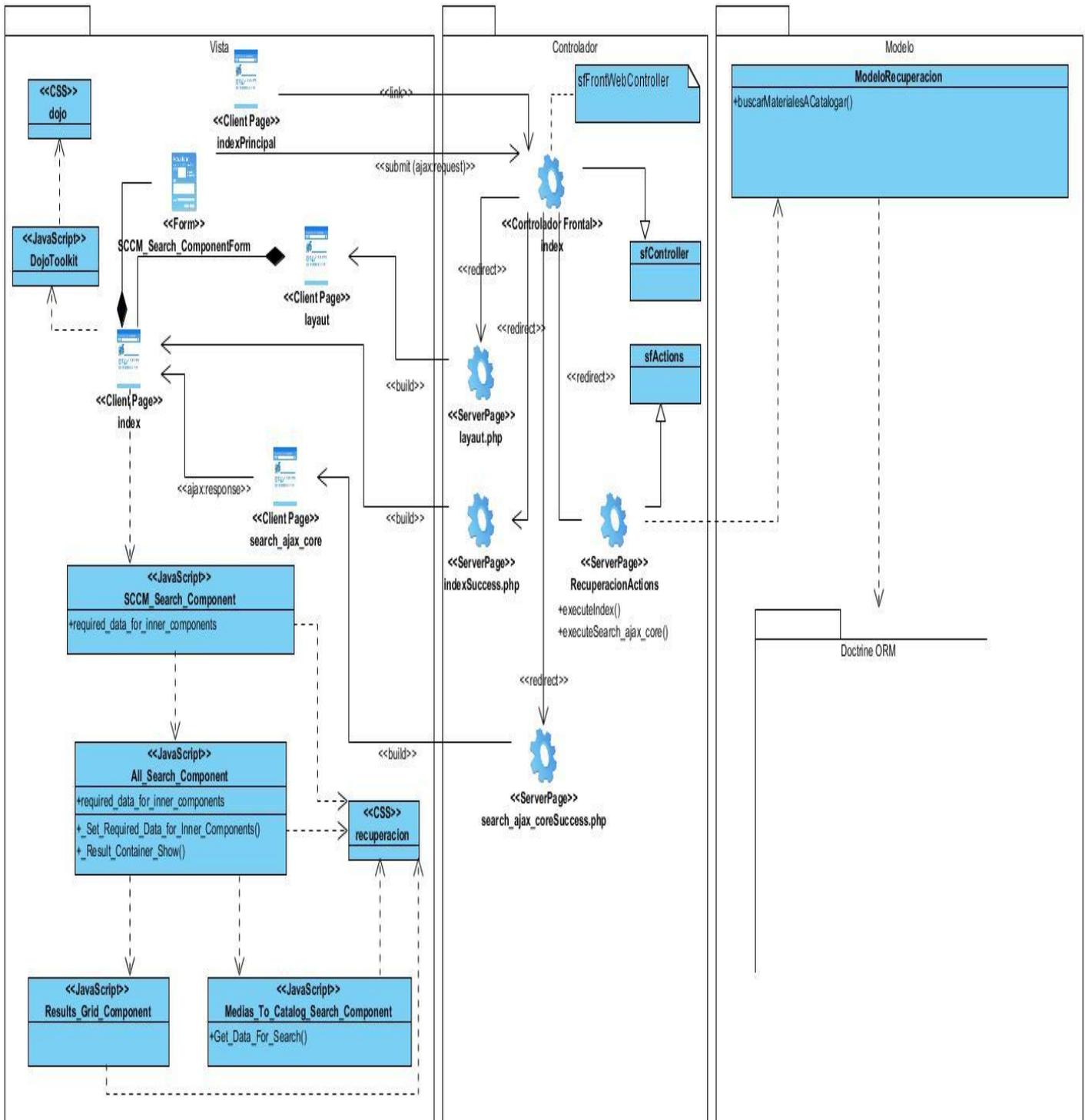


Figura 8: Diagrama de clases del diseño para el caso de uso Buscar Material a Catalogar.

3.3.2 Diagrama de clases del diseño para el caso de uso Buscar Material

El diagrama de clases del diseño para el caso de uso Buscar Material es muy similar al anterior (Figura 9). La diferencia está dada a que en este caso se realiza la búsqueda a las medias que ya fueron catalogadas. Las clases fundamentales son:

- ✓ **SCCM_Search_Component:** En este caso es el método *Execute_Medias_Already_Catalogued_Search ()* quien se encargaría de iniciar el proceso de búsqueda.
- ✓ **SCCM_Search_ComponentForm:** En este caso dicho formulario ofrece campos en dependencia de la búsqueda que desea realizar el usuario.
- ✓ Las clases **Basic_Search_Component**, **Advanced_Search_Component**, **Clasificacion_Serach_Component** y **Thesaurus_Search_Component** son las encargadas de recoger los criterios de recuperación referentes a la búsqueda básica, avanzada, por clasificación y utilizando tesauros.
- ✓ Se utilizan también la clase del controlador **RecuperacionActions** y del modelo **ModeloRecuperacion** que se apoya para estos tipos de búsqueda en los métodos *Already_Catalogued_Medias_Search ()*, *Basic_Search ()*, *Advanced_Search ()*, *Clasification_Search ()* y *Thesaurus_Search ()*.

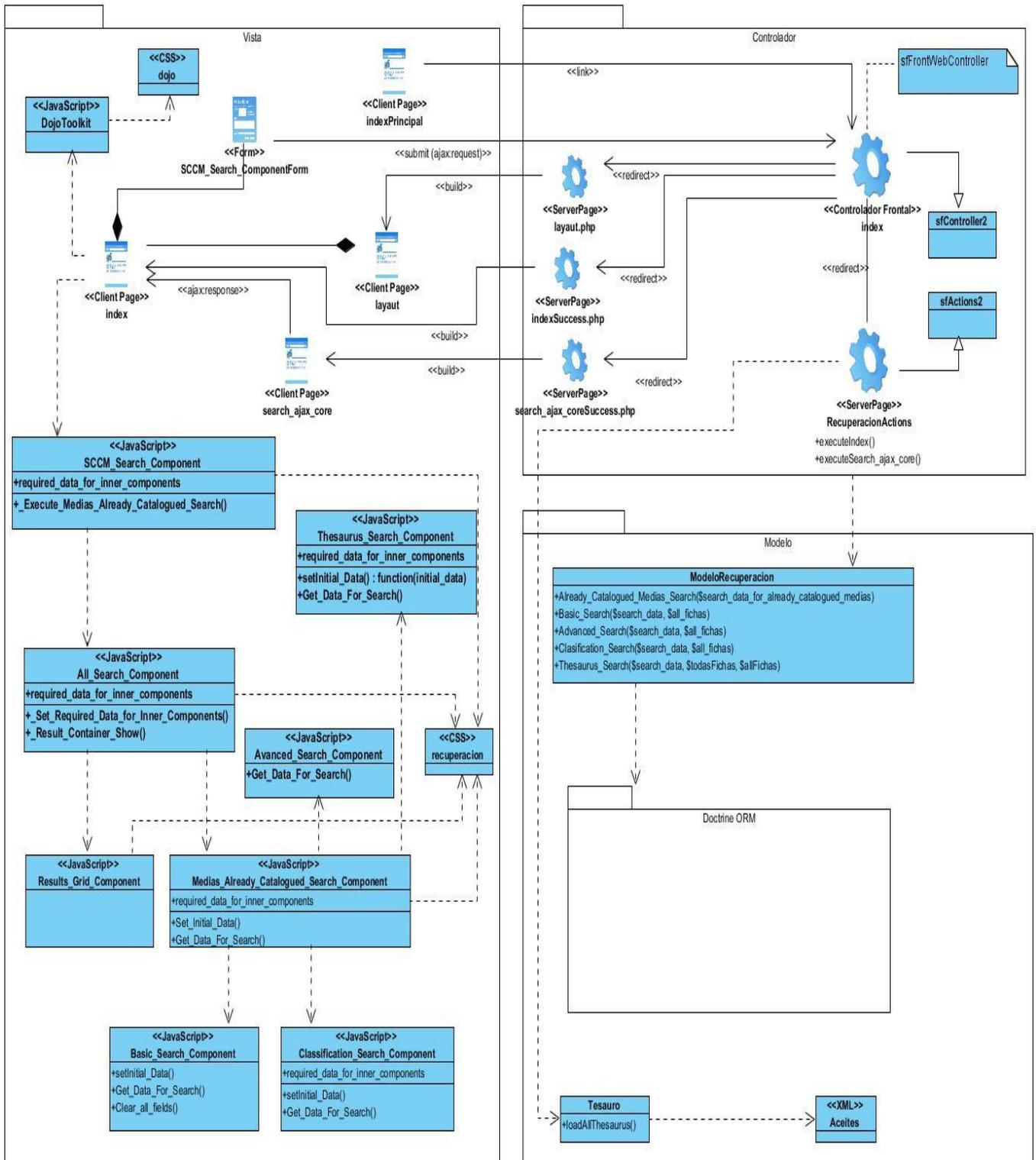


Figura 9: Diagrama de clases del diseño para el caso de uso Buscar Material.

3.4 Diseño de Base de Datos

El correcto funcionamiento de la aplicación también depende de la correcta elaboración y diseño de la base de datos. Un buen diseño de esta permite tener acceso a la información de forma eficiente y brinda una mayor seguridad de la información almacenada.

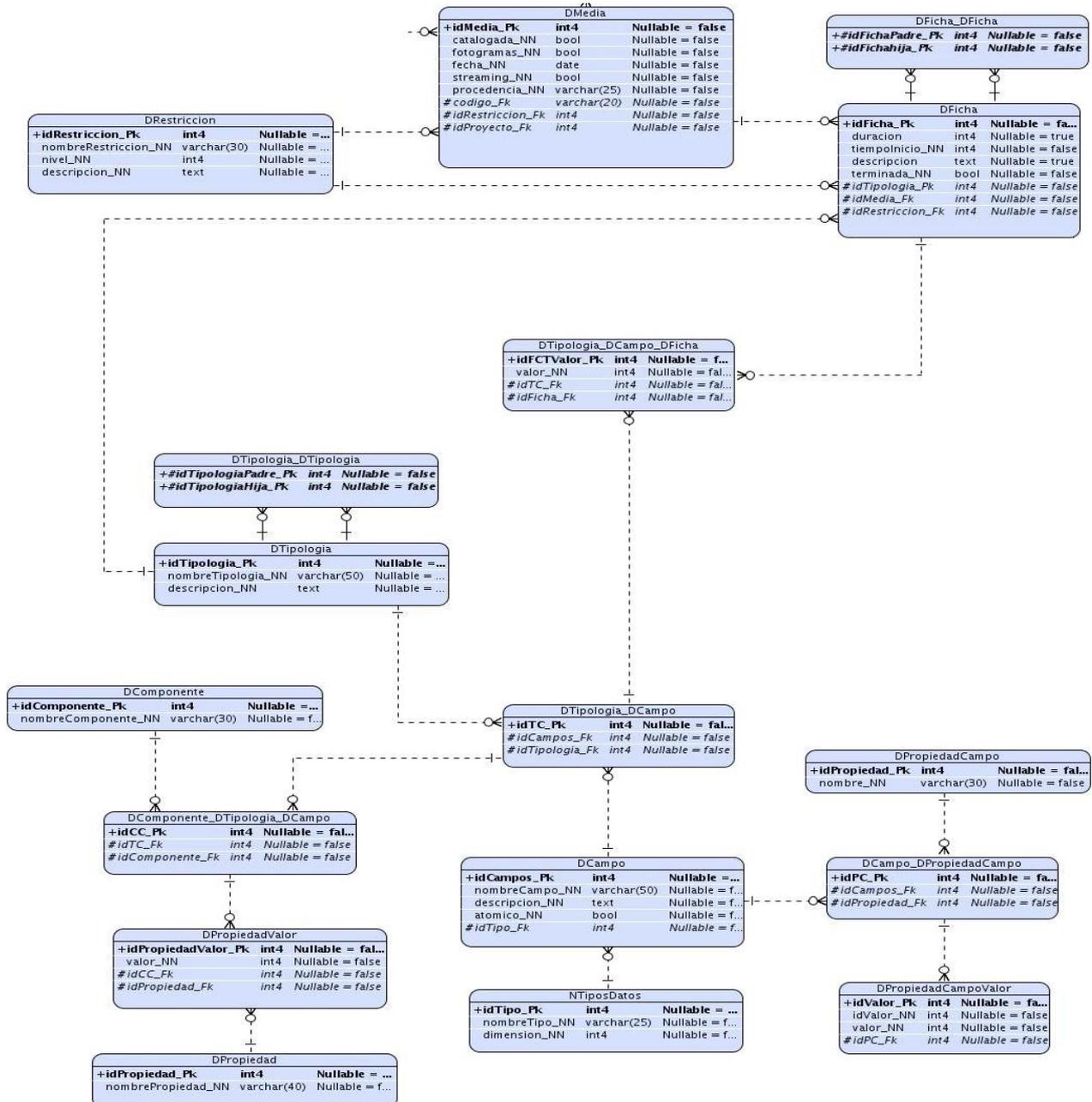


Figura 10: Modelo de datos.

En la imagen se muestra una parte del modelo de datos de la base de datos del proyecto SCCM. Esta fue diseñada con el objetivo de almacenar y gestionar toda la información referente al flujo de actividades desarrollada por el *software*. Se muestran las tablas con las cuales el módulo de recuperación de medias interactúa.

En dicha interacción existen tres tablas que son de vital importancia en el funcionamiento del módulo. Ellas son *DMedia*, *DFicha* y *DTipología*. En *DMedia* se consulta la información referente a la búsqueda de materiales a catalogar, puesto que es donde se almacena lo relacionado con la procedencia y la fecha en que fue insertado el material audiovisual. En *DFicha* es donde se recupera la información referente a la búsqueda básica, avanzada y utilizando tesauros. Esto se debe a que es en esta tabla donde se guarda la descripción y el título de las medias catalogadas. La tabla *DTipología* establece una relación con *DFicha* que caracteriza al material audiovisual en dependencia del contenido de este, posibilitando así realizar la recuperación de las medias atendiendo a la clasificación de estas, en dependencia de su contenido audiovisual.

3.5 Diagrama de Despliegue

El modelo de despliegue se realiza con el objetivo de mostrar las relaciones que existen entre los componentes *hardware* y *software* de la aplicación. Está compuesto por un conjunto de nodos unidos por conexiones de comunicación. En la siguiente imagen se visualiza el diagrama de despliegue referente al módulo de recuperación de materiales audiovisuales para el proyecto SCCM.

El nodo PC Cliente representa las computadoras que permitirán la interacción de los usuarios con la aplicación cliente del sistema, brindando las funcionalidades de recuperación de contenido multimedia de acuerdo a filtros seleccionados. Este se conecta con los nodos donde se encuentran el *Servidor Web* con el protocolo *HTTP*. El *Servidor Web* a su vez se comunica con el *Servidor de Base de Datos* mediante el protocolo *TCP-IP*.

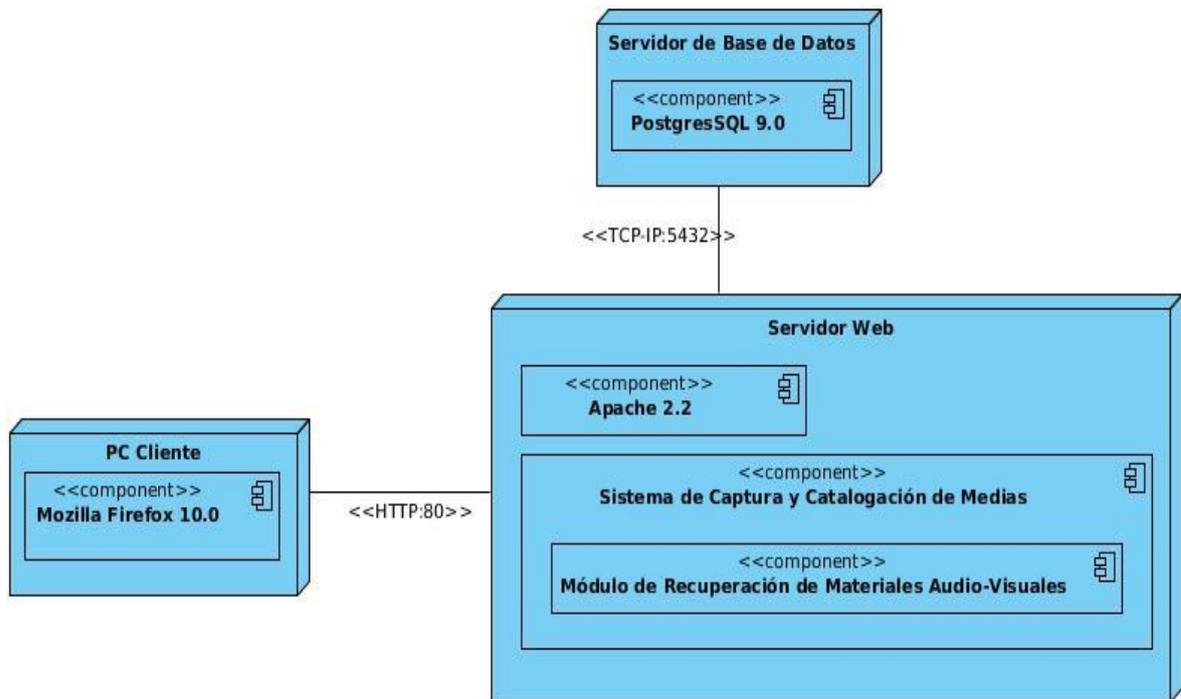


Figura 11: Modelo de despliegue.

3.6 Modelo de Implementación

El modelo de implementación está constituido por un conjunto de componentes y subsistemas que constituyen la constitución física de la implementación de un sistema. Son utilizados para estructurar el modelo de implementación en términos de módulos y mostrar las relaciones entre dichos componentes.

La siguiente imagen muestra la distribución e interacción entre los componentes que conforman el módulo *web* de recuperación de materiales audiovisuales, teniendo en cuenta el diseño y la arquitectura establecida para el mismo.

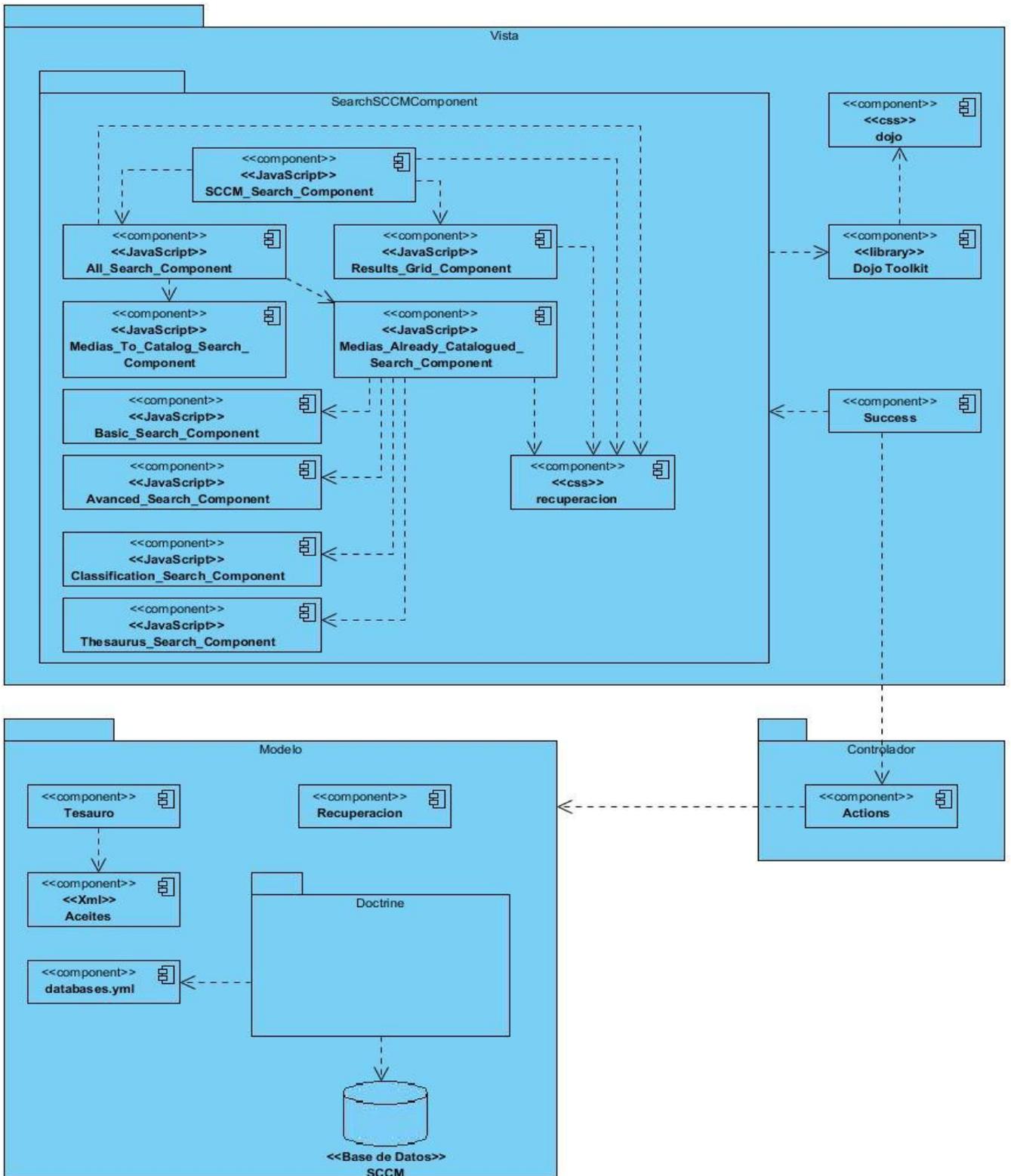


Figura 12: Modelo de implementación.

3.7 Estándar de Codificación

Los estándares de código resultan importantes en cualquier proyecto de desarrollo, pero son especialmente importantes cuando muchos desarrolladores trabajan en el mismo proyecto. Ayudan a asegurar que el código tenga una alta calidad, menos errores, y pueda ser mantenido fácilmente. En el desarrollo del módulo de recuperación de materiales audiovisuales se establecieron los siguientes estándares:

General: Para archivos que contengan únicamente código *PHP*, la etiqueta de cierre ("`?>`") no está permitida. No es requerida por *PHP*, y omitirla evita la inyección de espacios en blanco en la respuesta.

Clases: Los nombres de clases pueden contener sólo caracteres alfanuméricos. 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`".

Clases Abstractas: En general, las clases abstractas siguen las mismas convenciones que las clases, con una regla adicional: Los nombres de las clases abstractas deben acabar con el término, "`Abstract`", y ese término no debe ser precedida por un guión bajo. Ejemplo no válido: `Zend_Controller_Plugin_Abstract`. Serían nombres válidos: `Zend_Controller_PluginAbstract`.

Interfaces: En general, las clases abstractas siguen las mismas convenciones que las clases, con una regla adicional: Los nombres de las interfaces opcionalmente pueden acabar con el término, "`Interface`", pero término no debe ser precedida por un guión bajo. Ejemplo considerado un nombre no válido: `Zend_Controller_Plugin_Interface`. Serían nombres válidos: `Zend_Controller_PluginInterface`.

Funciones y Métodos: 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.

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

Constantes: Las constantes pueden contener tanto caracteres alfanuméricos como barras bajas (_). Los números están permitidos. Todas las letras pertenecientes al nombre de una constante deben aparecer en mayúsculas. Las palabras dentro del nombre de una constante deben separarse por barras bajas (_). Por ejemplo, *EMBED_SUPPRESS_EMBED_EXCEPTION* está permitido, pero *EMBED_SUPPRESSEMBEDEXCEPTION* no. Las constantes deben ser definidas como miembros de clase con el modificador "*const*". Definir constantes en el alcance global con la función "*define*" está permitido pero no recomendado.

```
private function Basic_Search($search_data, $all_fichas) {
    $filtered_fichas = array();
    foreach ($all_fichas as $ficha) {
        if (empty($search_data->titulo) || strstr($ficha->getTituloNn(), $search_data->titulo))
            if (empty($search_data->palabra_clave) || $this->Ficha_Posee_Palabra_Clave($ficha, $search_data->palabra_clave))
                if (empty($search_data->procedencia) || $ficha->getDmedia()->getDprocedencia()->getNombreNn() == $search_data->procedencia)
                    if (empty($search_data->descripcion) || strstr($ficha->getDescripcion(), $search_data->descripcion)) {
                        $filtered_fichas[] = $ficha;
                    }
            }
    }
    return $filtered_fichas;
}
```

Figura 13: Ejemplo de código PHP.

```
__Set_Proper_Dimensions_For_Inner_Components: function()
{
    dojo.style(this.Main_Search_Tab_Container_AP.domNode, "width", this.width + "px");
    dojo.style(this.Main_Search_Tab_Container_AP.domNode, "height", this.height + "px");

    this.point_medias_already_catalogued_inner_component.setWidth(this.width - 3);
    this.point_medias_already_catalogued_inner_component.setHeight((this.height - 35));

    this.point_results_grid.setHeight(this.height - 57);
},
```

Figura 14: Ejemplo de código JavaScript.

3.8 Integración

Actualmente el módulo *web* de recuperación de medias se encuentra integrado con los módulos de catalogación y préstamo. Para ello presenta el componente visual *SCCM_Search_Component*, el cual es llamado desde el *HTML* de la interfaz y tiene todas las funcionalidades que facilitan la construcción dinámica de los formularios, en dependencia de los datos almacenados en la base de datos.

Dicho componente se incluye en la página cliente donde se quiere mostrar, se le pasan los atributos referentes a los datos de la base de datos y se le especifica, en forma de parámetros, la forma en que se va a visualizar. La forma de visualización puede ser completa o para los materiales catalogados. La primera se utiliza en el módulo de catalogación y permite realizar la búsqueda sobre todos los materiales almacenados. La segunda se utiliza en el módulo de préstamo y permite realizar la búsqueda solamente de aquellas medias que ya fueron catalogadas. Los resultados para el primer caso son mostrados en un *grid* que brinda la posibilidad de catalogar el material seleccionado. Los resultados para el segundo caso no son visualizados, simplemente se devuelven para que el módulo de préstamo los pueda utilizar.

3.9 Conclusiones Parciales

Con el correcto uso de los estilos y patrones arquitectónicos, se lograron elaborar los diagramas de clases del diseño, el modelo de datos, el diagrama de despliegue y el modelo de implementación. Estos artefactos sirvieron de guía a la implementación del módulo web para la recuperación de medias, y servirán como punto de partida a próximos desarrolladores que deseen implementar nuevas versiones. En este capítulo se describió el estándar de codificación utilizado. Se realizaron los flujos de trabajo establecidos en la metodología seleccionada, permitiendo el desarrollo del módulo en el tiempo establecido.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

4.1 Introducción

Las pruebas del *software* son un elemento crítico para garantizar la calidad de la aplicación desarrollada. Representan una revisión final de las especificaciones, del diseño y de la codificación de dicha aplicación. Tienen como objetivo general validar un programa, independientemente de las características y el entorno donde se desarrolla, además permite documentar los errores en el sistema y posibilita probar la correcta implementación de los requisitos que el *software* debe cumplir.

En la actualidad existen varios tipos de prueba que su función fundamental es garantizar la calidad del *software* desarrollado y detectar los errores que este puede presentar. Un ejemplo sería las pruebas de caja negra, las cuáles se encargan de estudiar el comportamiento de la aplicación, teniendo en cuenta los datos de entrada y los de salida, desconociéndose el funcionamiento interno del programa.

Estas pruebas se centran en lo que se espera de la aplicación, es decir, intentan encontrar casos en que el módulo no funcione correctamente. Es por eso que se denominan pruebas funcionales, y el probador se limita a suministrarle datos de entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

4.2 Pruebas de Caja Negra

Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. Muchos autores consideran que estas pruebas permiten encontrar (EcuRed, 2010):

- ✓ Funciones incorrectas o ausentes.
- ✓ Errores de interfaz.
- ✓ Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- ✓ Errores de rendimiento.
- ✓ Errores de inicialización y terminación.

En resumen, las pruebas de caja negra permiten encontrar errores como funciones incorrectas o ausentes, en la interfaz, en la estructura de datos, de rendimiento y de inicialización y terminación.

4.3 Casos de Prueba

La realización de las pruebas de caja negra al módulo de recuperación de materiales audio-visuales para el Sistema Captura y Catalogación de Medias, requirió el diseño de los casos de prueba en correspondencia con las descripciones de casos de uso. Los casos de prueba permiten comprobar todos los flujos de información del sistema y validar que este cumple con los requisitos del cliente. A continuación se presentan los casos de prueba utilizados para los casos de uso más importantes del módulo:

4.3.1 Casos de Prueba Para el Caso de Uso Buscar Materiales a Catalogar

La descripción de las variables y la matriz de datos se encuentran en los anexos 5 y 6 respectivamente.

Tabla 1: Casos de Prueba Para el Caso de Uso Buscar Materiales a Catalogar.

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 a catalogar.	EC 1.1: Obtener resultados satisfactoriamente.	El usuario inserta filtros en la interfaz de búsqueda que correspondan con los que poseen medias sin catalogar en el sistema y se muestran estas en los resultados de la búsqueda.	Módulo Catalogación/ Sección de Búsqueda de materiales a catalogar.	El sistema debe mostrar la tabla de resultados con las medias que no estén catalogadas en el archivo digital, cuyos datos cumplan con los filtros seleccionados.	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 almacenadas sin catalogar en	Módulo Catalogación/ Sección de Búsqueda de materiales a	El sistema debe mostrar un mensaje en la lista de resultados indicando que	El sistema muestra el mensaje de "0 resultados" en la lista resultante de

		los filtros de la aplicación	catalogar.	no hay medias con esos valores.	la búsqueda.
--	--	------------------------------	------------	---------------------------------	--------------

4.3.2 Casos de Prueba Para el Caso de Uso Buscar Materiales

La descripción de las variables y la matriz de datos se encuentran en los anexos 7 y 8 respectivamente.

Tabla 2: Casos 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: Obtener 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 Catalogación/ 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 exceptuando los resultados de la búsqueda por clasificación.
	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 Catalogación/ 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 muestra el mensaje de "0 resultados" en la lista resultante de la búsqueda.

4.4 Resultado de las Pruebas

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 por clasificación de materiales catalogados, las funcionalidades del módulo cumplen con todos los requisitos que se definieron al inicio del proyecto.

La no conformidad referente a la búsqueda por clasificación de los materiales catalogados fue resuelta después de haberse detectado y actualmente funciona como es necesario, devolviendo los resultados correctos cada vez que es utilizada en el sistema.

4.5 Conclusiones Parciales

Las pruebas realizadas a la aplicación, en particular al módulo de recuperación de materiales audiovisuales, lograron detectar y documentar a tiempo los errores que existían para poder corregirlos. Además validaron que la implementación satisface todos los requisitos identificados. Al finalizar con la fase de prueba se obtuvo como resultado una aplicación que funciona correctamente, cumpliéndose así la totalidad de los objetivos planteados para el módulo *web* de búsqueda y recuperación de materiales audiovisuales. Dicha aplicación está lista para ser desplegada.

CONCLUSIONES GENERALES

Una vez culminado el trabajo se puede concluir que:

- ✓ La caracterización de los procesos relacionados con la búsqueda y recuperación de materiales audiovisuales, en el proyecto Sistema de Captura y Catalogación de Medias, posibilitó comprender el flujo de información y las formas de recuperación para poder implementar con mayor claridad las funcionalidades correspondientes.
- ✓ El estudio de las herramientas y tecnologías correctas, permitieron elaborar una solución informática que erradicó las dificultades encontradas en la versión anterior del módulo.
- ✓ La ingeniería realizada en la creación de la solución permitió documentar los artefactos generados en los flujos de trabajo de la metodología seleccionada, posibilitando la futura utilización de los mismos en próximas versiones.
- ✓ Se ha obtenido como producto principal de este trabajo el módulo *web* de recuperación de materiales audiovisuales para el proyecto Sistema de Captura y Catalogación de Medias, que cumple con las funcionalidades establecidas en el proyecto, entre las que se encuentran: buscar materiales catalogados y buscar materiales a catalogar.
- ✓ El módulo fue construido con tecnologías completamente libres, cumpliendo con la política de soberanía tecnológica por la que aboga nuestro país.
- ✓ El módulo fue revisado por calidad y está listo para ser utilizado en el proyecto SCCM.

RECOMENDACIONES

Al concluir el desarrollo de este trabajo se recomienda:

- ✓ Continuar desarrollando la investigación para perfeccionar y aumentar los criterios de búsqueda brindados a los usuarios.
- ✓ Implementar un editor de tesauros que posibilite el enriquecimiento de estos y así mejorar la búsqueda aplicando estos tesauros.

TRABAJOS CITADOS

Altamirano, Ing. Alfonso Valdez. 2009. *Comparativo de Entornos de Desarrollo Integrados (IDE's)*. 2009.

Ávila, Sergio José Villaneda. 2005. Mentores.net Introducción a Microsoft Solutions Framework. *Mentores.net Introducción a Microsoft Solutions Framework*. [En línea] 2005. [Citado el: 8 de 1 de 2011.] http://www.mentores.net/articulos/intro_microsoft_sol_frame.htm.

Baby González, Ronald; Almeida Maldonado, Enrique. 2009. *Sistema de Catalogación de Medias*. Ciudad de La Habana : s.n., 2009.

Broncano, Rubén García. 2006. Recuperacion y organizacion de la informacion. Universidad carlos III de Madrid . *Recuperacion y organizacion de la informacion. Universidad carlos III de Madrid* . [En línea] 09 de 04 de 2006. [Citado el: 22 de 10 de 2010.] <http://modelosrecuperacion.tripod.com/vectorial.html>.

Castañeda, Gerardo Ojeda -. 1992. Ministerios de Educación y Ciencia de España. *Ministerios de Educación y Ciencia de España*. [En línea] 1992. [Citado el: 17 de 11 de 2010.] <http://ares.cnice.mec.es/informes/13/contenido/22.htm#2.1...>

Datapro Client/Server Analyst. 1994. *client server computing: emerging trends, solutions and strategies*. 1994.

Denzer, Patricio. 2002. *PostgreSQL*. 2002.

DRAE. 2001. REAL ACADEMIA ESPAÑOLA. *REAL ACADEMIA ESPAÑOLA*. [En línea] 2001. [Citado el: 15 de 11 de 2010.] <http://rae.es/rae.html>.

EcuRed. 2010. EcuRed. *EcuRed*. [En línea] 2010. [Citado el: 10 de 5 de 2012.] http://www.ecured.cu/index.php/Pruebas_de_caja_negra.

EcuRed. 2006. EcuRed. *EcuRed*. [En línea] 2006. [Citado el: 25 de 9 de 2011.] <http://www.ecured.cu/index.php/Tesoros>.

Eguiluz, Javier. 2011. symfony.es. *symfony.es*. [En línea] 2011. [Citado el: 20 de 1 de 2011.] <http://www.symfony.es/que-es-symfony/>.

- FECYT. 2004.** Tecnociencia . *Tecnociencia* . [En línea] Octubre de 2004. [Citado el: 25 de 11 de 2010.] http://www.tecnociencia.es/e-revistas/especiales/revistas/pdf/e-revistas_informe.pdf.
- Flower, Martin y Wessley, Addison. 1997.** *Analysis Patterns:Reusable Object Models*. s.l. . 1997.
- Guerrero, Luis A. 1999.** *Universidad de Chile. Departamento de Ciencias de La Comunicación. Taller de UML*. Chile : s.n., 1999.
- Gutiérrez., Javier J. 2008.** *¿Qué es un framework web?* 2008.
- Hernández Reyes., Carlos Enrique y Leyva Durán., Julio Alberto. 2008.** *Desarrollo de la Arquitectura del Sistema Automatizado de Control de Gestión de Indicadores de Refinación, SACGIR*. 2008.
- Herrera, Antonio Gabriel López. 2006.** *Modelos de Sistemas de Recuperación de Información Documental Basados en Información Lingüística Difusa*. Granada : s.n., 2006.
- ISID Gestión Inteligente de Activos Multimedia. 2009.** ISID. *IS/D*. [En línea] 2009. [Citado el: 13 de 2 de 2012.] http://www.isid.es/spanish/product/producto_v00.htm.
- ISID Gestión Inteligente de Activos Multimedia. 2009. 2009.** ISID . *IS/D* . [En línea] 2009. [Citado el: 26 de 11 de 2010.] http://www.isid.es/Spanish/product/producto_v00.htm.
- Joskowicz, Ing. José. 2008.** *Reglas y Prácticas en eXtreme Programming*. España : s.n., 2008.
- Latinos.us. 2008** . [En línea] 2008 . [Citado el: 26 de 11 de 2010.] <http://www.referencia.us/terminos/personal/definicion-de-directorio-en-base-a-la-informatica/>.
- Letelier, Patricio. 2008.** Portal Desarrollo de Software. *Portal Desarrollo de Software*. [En línea] 2008. [Citado el: 5 de 1 de 2011.] <https://pid.dsic.upv.es/C1/Material/default.aspx>.
- M. Fowler, K. Beck, J. Brant, W. Opdyke, and D. Roberts,A. Wesley. 1999.** *Refactoring: Improving the Desing of Exist Code*. s.l. . 1999.
- Martín, Pedro Rufo. 2001.** ASPtutor. *ASPtutor*. [En línea] 2001. [Citado el: 12 de 1 de 2011.] <http://www.asptutor.com>.
- Mireles, Gabriel Alberto García. 2004.** *Introducción a los casos de uso*. Sonora : s.n., 2004.

- Molina, María Pinto. 2009.** http://www.mariapinto.es/e-coms/recu_infor.htm# .
http://www.mariapinto.es/e-coms/recu_infor.htm#. [En línea] 2009. [Citado el: 20 de 11 de 2010.]
- Molpeceres, Alberto. 2002.** *Procesos de desarrollo: RUP, XP, FDD*. Alemania : s.n., 2002.
- Murugarren, Joaquin Gracia. 2004.** WebEstilo. *WebEstilo*. [En línea] 3 de 10 de 2004. [Citado el: 16 de 1 de 2011.] <http://www.webestilo.com/php/php00.phtml>.
- NAB11, Tedral . 2011.** Panorama audiovisual.com. *Panorama audiovisual.com*. [En línea] 13 de 4 de 2011. <http://www.panoramaaudiovisual.com/es/2011/04/13/tedial-presenta-en-nab-sus-ultimos-avances-en-gestion-de-contenidos/>.
- Orallo, Enrique Hernández. 2010 .** El Lenguaje Unificado de Modelado (UML). *El Lenguaje Unificado de Modelado (UML)*. [En línea] 2010 . [Citado el: 8 de 1 de 2011.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
- Pereyra, Beatriz. 2005.** *Cátedra de Introducción a la Computación. BASES DE DATOS* . 2005.
- Pérez, Javier Eguíluz. 2008.** Introducción a AJAX. *Introducción a AJAX*. [En línea] 2008. [Citado el: 14 de 1 de 2011.] http://www.librosweb.es/ajax/pdf/introduccion_ajax.pdf.
- Pérez, Sonia Collada. 2009.** *Sistema de Indexación y Búsqueda de Documentos Audiovisuales*. Leganés : s.n., 2009.
- Perissé, Marcelo Claudio. 2001.** *PROYECTO INFORMATICO: Una metodología simplificada*. Buenos Aires : s.n., 2001.
- Potencier, Fabien. 2009.** *El tutorial Jobeet*. 2009.
- Potencier, Fabien y Zaninotto, François. 2009.** *Symfony la guía definitiva Doctrine*. 2009.
- Rodríguez Vázquez, Solangel y Serrano Rosales, Carlos Luis. 2008.** *Desarrollo de Biblioteca de Métodos Numéricos (BMN), referente a Sistemas de Ecuaciones Lineales, Sistemas de Gran Dimensión y Poco Densos e Integración Numérica*. la Habana : s.n., 2008.
- Ronda Amador, Ing. Yoelys, Cobas Santos, Ing. Kadir y Marrero Viñas., Ing. Daymel. 2002.** *El modelo cliente-servidor de tres capas para el trabajo en redes de información*. 2002.

Rozic, Sergio Ezequiel. 2004. Base de Datos . *Base de Datos* . [En línea] 2004. [Citado el: 22 de 11 de 2010.] <http://bibliodoc.uci.cu/pdf/reg03438.pdf>.

Sanchez, María A. Mendoza. 2004. Metodologías De Desarrollo De Software. *Metodologías De Desarrollo De Software*. [En línea] 2004. [Citado el: 8 de 1 de 2011.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Tedial, NAB11. 2011. Panorama audiovisual.com. *Panorama audiovisual.com*. [En línea] 13 de 4 de 2011. [Citado el: 13 de 1 de 2012.] <http://www.panoramaaudiovisual.com/es/2011/04/13/tedial-presenta-en-nab-sus-ultimos-avances-en-gestion-de-contenidos/>.

Tramullas, Jesús. 2000. tramullas.com. *tramullas.com*. [En línea] 2000. [Citado el: 15 de 11 de 2010.] <http://tramullas.com/documatica/3-1.html>..

UCU. 1971. Universidad de Concepción del Uruguay. *Universidad de Concepción del Uruguay*. [En línea] 1971. [Citado el: 26 de 11 de 2010.] http://www.ucu.edu.ar/index.php?option=com_content&view=article&id=149&Itemid=86.

Vega Ortiz, Yurisbel, Figueroa Hidalgo, Daira y Martell Fernández, Vladimir. 2010. *Propuesta de diseño para proyectos informáticos que utilizan Symfony como Framework*. 2010.

Visconti, Marcello y Astudillo, Hernán. 2010. *Fundamentos de Ingeniería de Software*. Chile : s.n., 2010.

W3C (MIT, ERCIM, Keio). 2008. W3C World Wide Web consortium. *W3C World Wide Web consortium*. [En línea] 2008. [Citado el: 13 de 1 de 2011.] <http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>.

XTREAM 2010 . 2010. Xstream, Sistema de Información Global . *Xstream, Sistema de Información Global* . [En línea] 2010. [Citado el: 26 de 11 de 2010.] <http://www.xtreamsig.com/>.

BIBLIOGRAFÍA

Altamirano, Ing. Alfonso Valdez. 2009. *Comparativo de Entornos de Desarrollo Integrados (IDE's)*. 2009.

Ávila, Sergio José Villaneda. 2005. Mentores.net Introducción a Microsoft Solutions Framework. *Mentores.net Introducción a Microsoft Solutions Framework*. [En línea] 2005. [Citado el: 8 de 1 de 2011.] http://www.mentores.net/articulos/intro_microsoft_sol_frame.htm.

Baby González, Ronald; Almeida Maldonado, Enrique. 2009. *Sistema de Catalogación de Medias*. Ciudad de La Habana : s.n., 2009.

Broncano, Rubén García. 2006. Recuperacion y organizacion de la informacion. Universidad carlos III de Madrid . *Recuperacion y organizacion de la informacion. Universidad carlos III de Madrid* . [En línea] 09 de 04 de 2006. [Citado el: 22 de 10 de 2010.] <http://modelosrecuperacion.tripod.com/vectorial.html>.

Castañeda, Gerardo Ojeda -. 1992. Ministerios de Educación y Ciencia de España. *Ministerios de Educación y Ciencia de España*. [En línea] 1992. [Citado el: 17 de 11 de 2010.] <http://ares.cnice.mec.es/informes/13/contenido/22.htm#2.1...>

Datapro Client/Server Analyst. 1994. *client server computing: emerging trends, solutions and strategies*. 1994.

Denzer, Patricio. 2002. *PostgreSQL*. 2002.

DRAE. 2001. REAL ACADEMIA ESPAÑOLA. *REAL ACADEMIA ESPAÑOLA*. [En línea] 2001. [Citado el: 15 de 11 de 2010.] <http://rae.es/rae.html>.

EcuRed. 2010. EcuRed. *EcuRed*. [En línea] 2010. [Citado el: 10 de 5 de 2012.] http://www.ecured.cu/index.php/Pruebas_de_caja_negra.

EcuRed. 2006. EcuRed. *EcuRed*. [En línea] 2006. [Citado el: 25 de 9 de 2011.] <http://www.ecured.cu/index.php/Tesauros>.

Eguiluz, Javier. 2011. symfony.es. *symfony.es*. [En línea] 2011. [Citado el: 20 de 1 de 2011.] <http://www.symfony.es/que-es-symfony/>.

- FECYT. 2004.** Tecnociencia . *Tecnociencia* . [En línea] Octubre de 2004. [Citado el: 25 de 11 de 2010.] http://www.tecnociencia.es/e-revistas/especiales/revistas/pdf/e-revistas_informe.pdf.
- Flower, Martin y Wessley, Addison. 1997.** *Analysis Patterns:Reusable Object Models*. s.l. . 1997.
- Guerrero, Luis A. 1999.** *Universidad de Chile. Departamento de Ciencias de La Comunicación. Taller de UML*. Chile : s.n., 1999.
- Gutiérrez., Javier J. 2008.** *¿Qué es un framework web?* 2008.
- Hernández Reyes., Carlos Enrique y Leyva Durán., Julio Alberto. 2008.** *Desarrollo de la Arquitectura del Sistema Automatizado de Control de Gestión de Indicadores de Refinación, SACGIR*. 2008.
- Herrera, Antonio Gabriel López. 2006.** *Modelos de Sistemas de Recuperación de Información Documental Basados en Información Lingüística Difusa*. Granada : s.n., 2006.
- ISID Gestión Inteligente de Activos Multimedia. 2009.** ISID. *ISID*. [En línea] 2009. [Citado el: 13 de 2 de 2012.] http://www.isid.es/spanish/product/producto_v00.htm.
- ISID Gestión Inteligente de Activos Multimedia. 2009.** ISID . *ISID* . [En línea] 2009. [Citado el: 26 de 11 de 2010.] http://www.isid.es/Spanish/product/producto_v00.htm.
- Joskowicz, Ing. José. 2008.** *Reglas y Prácticas en eXtreme Programming*. España : s.n., 2008.
- Latinos.us. 2008** . [En línea] 2008 . [Citado el: 26 de 11 de 2010.] <http://www.referencia.us/terminos/personal/definicion-de-directorio-en-base-a-la-informatica/>.
- Letelier, Patricio. 2008.** Portal Desarrollo de Software. *Portal Desarrollo de Software*. [En línea] 2008. [Citado el: 5 de 1 de 2011.] <https://pid.dsic.upv.es/C1/Material/default.aspx>.
- M. Fowler, K. Beck, J. Brant, W. Opdyke, and D. Roberts,A. Wesley. 1999.** *Refactoring: Improving the Desing of Exist Code*. s.l. . 1999.
- Martín, Pedro Rufo. 2001.** ASPtutor. *ASPtutor*. [En línea] 2001. [Citado el: 12 de 1 de 2011.] <http://www.asptutor.com>.
- Mireles, Gabriel Alberto García. 2004.** *Introducción a los casos de uso*. Sonora : s.n., 2004.

- Molina, María Pinto. 2009.** http://www.mariapinto.es/e-coms/recu_infor.htm# .
http://www.mariapinto.es/e-coms/recu_infor.htm# . [En línea] 2009. [Citado el: 20 de 11 de 2010.]
- Molpeceres, Alberto. 2002.** *Procesos de desarrollo: RUP, XP, FDD.* Alemania : s.n., 2002.
- Murugarren, Joaquin Gracia. 2004.** WebEstilo. *WebEstilo.* [En línea] 3 de 10 de 2004. [Citado el: 16 de 1 de 2011.] <http://www.webestilo.com/php/php00.phtml>.
- NAB11, Tedral . 2011.** Panorama audiovisual.com. *Panorama audiovisual.com.* [En línea] 13 de 4 de 2011. <http://www.panoramaaudiovisual.com/es/2011/04/13/tedial-presenta-en-nab-sus-ultimos-avances-en-gestion-de-contenidos/>.
- Orallo, Enrique Hernández. 2010 .** El Lenguaje Unificado de Modelado (UML). *El Lenguaje Unificado de Modelado (UML).* [En línea] 2010 . [Citado el: 8 de 1 de 2011.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
- Pereyra, Beatriz. 2005.** *Cátedra de Introducción a la Computación. BASES DE DATOS .* 2005.
- Pérez, Javier Eguíluz. 2008.** Introducción a AJAX. *Introducción a AJAX.* [En línea] 2008. [Citado el: 14 de 1 de 2011.] http://www.librosweb.es/ajax/pdf/introduccion_ajax.pdf.
- Pérez, Sonia Collada. 2009.** *Sistema de Indexación y Búsqueda de Documentos Audiovisuales.* Leganés : s.n., 2009.
- Perissé, Marcelo Claudio. 2001.** *PROYECTO INFORMATICO: Una metodología simplificada.* Buenos Aires : s.n., 2001.
- Potencier, Fabien. 2009.** *El tutorial Jobeet.* 2009.
- Potencier, Fabien y Zaninotto, François. 2009.** *Symfony la guía definitiva Doctrine.* 2009.
- Rodríguez Vázquez, Solangel y Serrano Rosales, Carlos Luis. 2008.** *Desarrollo de Biblioteca de Métodos Numéricos (BMN), referente a Sistemas de Ecuaciones Lineales, Sistemas de Gran Dimensión y Poco Densos e Integración Numérica.* la Habana : s.n., 2008.
- Ronda Amador, Ing. Yoelys, Cobas Santos, Ing. Kadir y Marrero Viñas., Ing. Daymel. 2002.** *El modelo cliente-servidor de tres capas para el trabajo en redes de información.* 2002.

- Rozic, Sergio Ezequiel. 2004.** Base de Datos . *Base de Datos* . [En línea] 2004. [Citado el: 22 de 11 de 2010.] <http://bibliodoc.uci.cu/pdf/reg03438.pdf>.
- Sanchez, María A. Mendoza. 2004.** Metodologías De Desarrollo De Software. *Metodologías De Desarrollo De Software*. [En línea] 2004. [Citado el: 8 de 1 de 2011.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
- Tedial, NAB11. 2011.** Panorama audiovisual.com. *Panorama audiovisual.com*. [En línea] 13 de 4 de 2011. [Citado el: 13 de 1 de 2012.] <http://www.panoramaaudiovisual.com/es/2011/04/13/tedial-presenta-en-nab-sus-ultimos-avances-en-gestion-de-contenidos/>.
- Tramullas, Jesús. 2000.** tramullas.com. *tramullas.com*. [En línea] 2000. [Citado el: 15 de 11 de 2010.] <http://tramullas.com/documatica/3-1.html>.
- UCU. 1971.** Universidad de Concepción del Uruguay. *Universidad de Concepción del Uruguay*. [En línea] 1971. [Citado el: 26 de 11 de 2010.] http://www.ucu.edu.ar/index.php?option=com_content&view=article&id=149&Itemid=86.
- Vega Ortiz, Yurisbel, Figueroa Hidalgo, Daira y Martell Fernández, Vladimir. 2010.** *Propuesta de diseño para proyectos informáticos que utilizan Symfony como Framework*. 2010.
- Visconti, Marcello y Astudillo, Hernán. 2010.** *Fundamentos de Ingeniería de Software*. Chile : s.n., 2010.
- W3C (MIT, ERCIM, Keio). 2008.** W3C World Wide Web consortium. *W3C World Wide Web consortium*. [En línea] 2008. [Citado el: 13 de 1 de 2011.] <http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>.
- XTREAM 2010 . 2010.** Xtreem, Sistema de Información Global . *Xtreem, Sistema de Información Global* . [En línea] 2010. [Citado el: 26 de 11 de 2010.] <http://www.xtreamsig.com/>.

ANEXOS

Anexo 1. Requisitos funcionales

1.1. Realizar búsquedas de material audiovisual.

1.1.1. Realizar búsqueda por clasificación.

Descripción: El sistema permitirá la realización de búsqueda de videos que están almacenados en el servidor. Se permitirá establecer filtros referentes a la clasificación del material y a los atributos específicos de la clasificación.

1.1.2. Realizar búsqueda avanzada.

Descripción: El sistema permitirá realizar búsquedas definiendo filtros avanzados relativos a la fecha o a la descripción del material, lo que posibilitará la búsqueda más específica de materiales atendiendo a las fechas almacenadas y a las descripciones asociadas a los mismos.

1.1.3. Realizar búsqueda básica.

Descripción: El sistema permitirá la realización de búsqueda de videos que están almacenados en el servidor. Se permitirá establecer filtros básicos en correspondencia con los datos que han sido asociados en la catalogación de materiales audiovisuales digitalizados.

1.1.4. Realizar búsqueda por tesauros.

Descripción: El sistema permitirá la realización de búsqueda de videos que están almacenados en el servidor. Se permitirá realizar búsquedas de términos relacionados con algún tesauro para que la búsqueda sea más general.

1.1.5. Realizar búsqueda combinada.

Descripción: El sistema permitirá la realización de búsqueda de videos que están almacenados en el servidor. Permitirá combinar las búsquedas anteriormente expuestas.

1.1.6. Realizar búsqueda de materiales a catalogar.

Descripción: El sistema permitirá la realización de búsqueda de videos que están almacenados en el servidor que no hayan sido catalogados.

1.1.7. Limpiar filtros de búsqueda.

Descripción: El sistema deberá permitir limpiar todos los filtros de búsqueda insertados por el usuario con el objetivo de introducir nuevos para obtener otros resultados.

Anexo 2. Diagrama de Caso de Uso

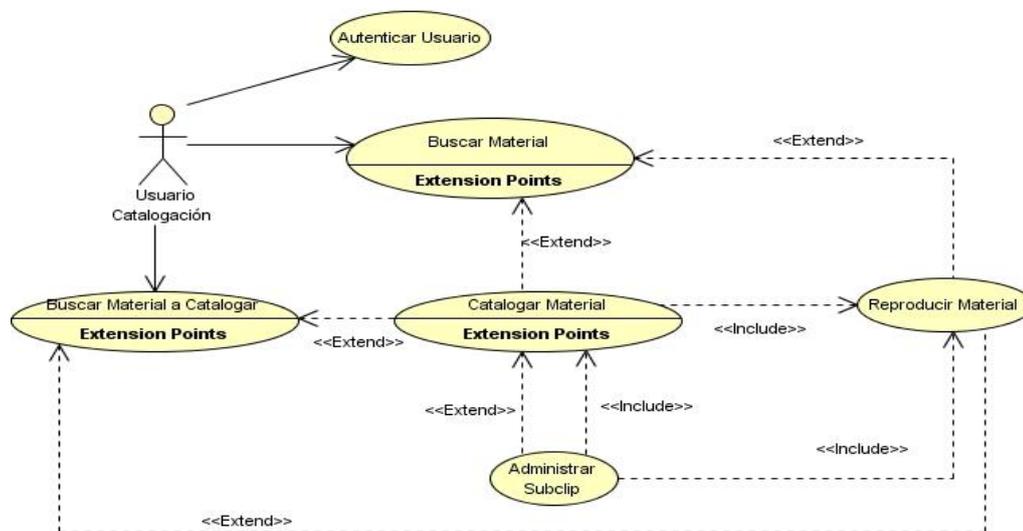


Figura 15: Diagrama de Caso de Uso

Anexo 3. Caso de Uso Buscar Material

Tabla 3: Descripción del caso de uso Buscar Material.

Caso de Uso:	Buscar Material	
Actores:	Usuario de Catalogación	
Resumen:	El caso de uso se inicia cuando se selecciona la opción de búsqueda de materiales. El sistema permite realizar cuatro tipos de búsqueda: por clasificación, avanzada, básica y por tesauros, el resultado de la búsqueda se muestra teniendo en cuenta los filtros introducidos. El caso de uso culmina cuando se muestran los resultados obtenidos a partir de los filtros introducidos por el usuario.	
Precondiciones:	El usuario debe estar autenticado y poseer permisos para la realización de búsqueda de materiales.	
Referencias	RF2 [1]	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso de inicia cuando el usuario activa el área correspondiente para introducir opcionalmente los filtros de búsqueda de		

<p>acuerdo al tipo de búsqueda que desea realizar, las que pueden ser:</p> <ul style="list-style-type: none"> a) Búsqueda por Clasificación, ir a la sección “Búsqueda por clasificación” b) Búsqueda Avanzada, ir a la sección “Búsqueda avanzada” c) Búsqueda Básica, ir a la sección “Búsqueda básica” d) Búsqueda por tesauros, ir a la sección “Búsqueda por tesauros” 	
<p>Sección “Búsqueda por clasificación”</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
	<p>1.1 El sistema activa el área de Búsqueda por clasificación (A), permitiendo introducir los filtros del tipo de búsqueda seleccionado.</p> <p>1.2 El sistema llena de forma dinámica el campo seleccionable de tipologías con todas las tipologías existentes en la base de datos.</p>
<p>2. El usuario escoge la tipología (Opcional) (B).</p>	<p>2.1 El sistema muestra dinámicamente una tabla para introducir los valores de los atributos correspondientes a la tipología seleccionada.</p>
<p>3. El usuario introduce los valores de los atributos correspondientes a la clasificación seleccionada (todos, ninguno o alguno) (Opcional) (C).</p>	
<p>4. El usuario puede realizar una de las siguientes acciones:</p> <ul style="list-style-type: none"> a) Activar el área de búsqueda avanzada (D) para introducir más filtros de búsqueda, ir a la sección “Búsqueda avanzada” b) Activar el área de búsqueda básica (E) para 	

<p>introducir más filtros de búsqueda, ir a la sección “Búsqueda básica”</p> <p>c) Activar el área de búsqueda por tesauros (H) para introducir más filtros de búsqueda, ir a la sección “Búsqueda por tesauros”</p> <p>d) Presionar botón Buscar (F), ir a la sección “Búsqueda combinada”</p> <p>e) Presionar botón “Limpiar Campos” (G), ir a la sección “Limpiar Búsqueda”.</p>	
Flujo alterno	
Acción del Actor	Respuesta del Sistema
	<p>3.1 De existir algún campo de fecha en la tipología seleccionada y si no introduce la fecha con el formato adecuado el sistema muestra el mensaje: “El valor especificado no es válido.”. El sistema da la posibilidad de corregirlo y continúa el caso de uso.</p>
Sección “Búsqueda avanzada”	
Acción del Actor	Respuesta del Sistema
	<p>1.1 El sistema activa el área de Búsqueda avanzada, permitiendo introducir los filtros de búsqueda avanzada.</p> <p>1.2 El sistema verifica si el usuario ha introducido filtros en el área de búsqueda por clasificación, en caso positivo si la clasificación que ha introducido allí tiene atributos de fecha, llena el campo seleccionable de tipos de fecha con las estas además de las fechas generales definidas que poseen todos los materiales: de inserción y de catalogación.</p>
2. El usuario introduce los datos correspondientes	2.1 El sistema verifica que los datos

<p>a los filtros de búsqueda avanzada:</p> <ul style="list-style-type: none"> • Filtro de fecha (opcional) (A) <ul style="list-style-type: none"> ○ Tipo de fecha (obligatorio) (B) ○ Por Intervalo (opcional) (C) <ul style="list-style-type: none"> ▪ Fecha Inicio (obligatorio si selecciona Por intervalo) (D) ▪ Fecha Fin (obligatorio si selecciona intervalo) (E) ○ Con omisión (opcional) (F) <ul style="list-style-type: none"> ▪ Fecha (obligatorio si selecciona con omisión) (G) ▪ Atributo de la fecha a omitir (al menos uno si selecciona con omisión) (H) • Filtros avanzados de datos generales (opcional) (I) <ul style="list-style-type: none"> ○ Texto (opcional) (J) ○ Ámbito (opcional) (K) ○ Respetar ortografía (opcional) (L) 	<p>introducidos estén correctos.</p>
<p>3. El usuario puede realizar las siguientes acciones:</p> <ol style="list-style-type: none"> a) Activar el área de búsqueda por clasificación (M) para introducir los filtros correspondientes, ir a la sección “Búsqueda por clasificación” b) Activar el área de búsqueda básica (N) para introducir los filtros correspondientes, ir a la sección “Búsqueda básica” c) Activar el área de búsqueda por tesauros (Q) para introducir los filtros correspondientes, ir a la sección “Búsqueda por tesauros” 	

<p>d) Presionar botón “Buscar” (O), ir a la sección “Búsqueda combinada”</p> <p>e) Presionar botón “Limpiar Campos” (P), ir a la sección “Limpiar Búsqueda”.</p>	
<p>Flujos Alternos</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
	<p>2.1 Si se seleccionó el filtro por omisión en fechas y no seleccionó la fecha el sistema muestra el mensaje: “Este valor es necesario”; el sistema da la posibilidad de seleccionar la fecha y continúa el caso de uso.</p>
	<p>2.1 Si se seleccionó el filtro intervalo y no seleccionó las fechas de inicio y fin el sistema muestra el mensaje: “Este valor es necesario”; el sistema da la posibilidad de seleccionar la fecha y continúa el caso de uso.</p>
<p>Sección “Búsqueda básica”</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
	<p>1.1 El sistema activa el área de Búsqueda básica, donde podrán ser introducidos los filtros de búsqueda básica.</p>
<p>2. El usuario introduce los filtros correspondientes a la búsqueda básica:</p> <ul style="list-style-type: none"> • Título del material (opcional) (A) • Palabras claves (opcional) (B) • Procedencia (opcional) (C) • Descripción (opcional) (D) 	<p>2.1. El sistema verifica que los datos introducidos estén correctos.</p>
<p>3. El usuario puede realizar las siguientes acciones:</p> <p>a) Activar el área de búsqueda por</p>	

<p>clasificación (E) para introducir los filtros correspondientes, ir a la sección “Búsqueda por clasificación”</p> <p>b) Activar el área de búsqueda avanzada (F) para introducir los filtros correspondientes, ir a la sección “Búsqueda avanzada”</p> <p>c) Activar el área de búsqueda por tesauros (I) para introducir los filtros correspondientes, ir a la sección “Búsqueda por tesauros”</p> <p>d) Presionar botón “Buscar” (G), ir a la sección “Búsqueda combinada”</p> <p>e) Presionar botón “Limpiar Campos” (H), ir a la sección “Limpiar Búsqueda”.</p>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Sección “Búsqueda por tesauros”	
Acción del Actor	Respuesta del Sistema
	1.1 El sistema activa el área de Búsqueda básica, donde podrán ser introducidos los filtros de búsqueda por tesauros.
<p>2. El usuario introduce los filtros correspondientes a la búsqueda por tesauros:</p> <ul style="list-style-type: none"> • Escribe el término que desea buscar (opcional) (A) • Presiona el botón Th (opcional), ir a la sección “Buscar término en tesauro” (B) 	2.1 Si el usuario escribe el término el sistema espera la próxima acción del usuario.
<p>3. El usuario adiciona el término con sus relaciones:</p> <ul style="list-style-type: none"> • Escoge una o varias relaciones del término en el tesauro (opcional) (C) 	3.1 El sistema adiciona el término con sus relaciones a la tabla de términos a buscar si el término que desea buscar (A) no está vacío.

<ul style="list-style-type: none"> • Presiona el botón Adicionar (D) (opcional), para adicionar el término con sus relaciones a la tabla de términos a buscar (E) 	
<p>4. El usuario puede realizar las siguientes acciones:</p> <ol style="list-style-type: none"> Activar el área de búsqueda por clasificación (F) para introducir los filtros correspondientes, ir a la sección “Búsqueda por clasificación” Activar el área de búsqueda avanzada (G) para introducir los filtros correspondientes, ir a la sección “Búsqueda avanzada” Activar el área de búsqueda básica (H) para introducir los filtros correspondientes, ir a la sección “Búsqueda básica” Presionar el botón eliminar fila de la tabla para eliminar un término que no desee buscar (K). Presionar botón “Buscar” (I), ir a la sección “Búsqueda combinada” <p>5. Presionar botón “Limpiar Campos” (J), ir a la sección “Limpiar Búsqueda”.</p>	<p>4.1 Si escoge la opción i) el sistema muestra una ventana para asegurar de que desea eliminar el término de la fila.</p>
<p>6. El usuario puede realizar las siguientes acciones:</p> <ol style="list-style-type: none"> Presionar botón “Aceptar” (L), para eliminar la fila. Presionar botón “Cancelar” (M), para no eliminar la fila. 	
<p>Sección “Buscar término en tesauro”</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>

	1.1 El sistema activa el área de Buscar término en tesauro, donde podrán ser introducidos los filtros de búsqueda por tesauros.
2. El usuario escoge el tesauro en el cual desea buscar el término (A).	2.1 El sistema espera la próxima acción del usuario.
3. El usuario escoge la relación (B) como quiere ver el árbol y presiona el botón ver árbol (C).	3.1 El sistema muestra el árbol en dependencia de la relación que seleccione (D).
4. El usuario escoge o escribe un término (E) y selecciona una de las siguientes opciones: a) Buscar exacto (F) b) Buscar similar (G)	4.1 El sistema muestra el árbol de términos asociados a la búsqueda que se realizó en dependencia del término seleccionado (G).
5. El usuario escoge un término (G) y selecciona una de las siguientes opciones: a) Aceptar (H) b) Cancelar (I)	5.1 Si escoge la opción Aceptar agrega el término en el filtro correspondiente en la sección Búsqueda por tesauros. 5.2 Si da cancelar no lo agrega.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.1 Si no selecciona el tesauro no mostrará el árbol de búsqueda. 4.1 Si no selecciona o escribe el término a buscar no se mostrará el árbol de búsqueda referente a ese término.
Sección “Búsqueda combinada”	
Acción del Actor	Respuesta del Sistema
	1.1 El sistema verifica que exista conexión con la Base de Datos. 1.2 El sistema combina todos los filtros de búsqueda introducidos y muestra la lista de los materiales que cumplen con los mismos (A) y culmina el caso de uso.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.1. Si no se puede establecer conexión con la Base de Datos, el sistema muestra el siguiente mensaje: “No se puede establecer conexión con los datos” y culmina el caso de uso.
	1.2. Si no existen materiales que cumplan con los filtros introducidos por el usuario la lista de resultados no mostrará ningún elemento indicando que no existen coincidencias con los filtros introducidos y culmina el caso de uso.
	1.3. Una vez obtenidos los materiales, el usuario puede seleccionar uno de los resultados de la búsqueda y seleccionar la opción de catalogarlo (B), ir al Caso de Uso Catalogar Material y culmina el caso de uso.
Sección “Limpiar búsqueda”	
Acción del Actor	Respuesta del Sistema
	1.1. Se reinician todos los valores introducidos en los filtros de búsqueda de todos los tipos de búsqueda. El sistema da la posibilidad de volver a insertar valores en los filtros. Culmina el caso de uso.
Pos condiciones	Se obtiene el listado de los materiales de acuerdo a los filtros establecidos por el usuario.

Anexo 4. Caso de Uso Buscar Material a Catalogar

Tabla 4: Descripción del caso de uso Buscar Material a Catalogar.

Caso de Uso:	Buscar Material a Catalogar	
Actores:	Usuario de Catalogación	
Resumen:	El caso de uso se inicia cuando el actor selecciona la opción de buscar los materiales a catalogar, se introducen los filtros necesarios para la búsqueda, el mismo culmina cuando el sistema muestra un listado de los materiales que no han sido catalogados y cumplen con los filtros establecidos.	
Precondiciones:	El actor debe estar autenticado y poseer permiso de catalogación.	
Referencias	RF2 [1]	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el usuario activa la opción de “Búsqueda de materiales a catalogar” (A).	1.1. El sistema muestra la interfaz correspondiente a la búsqueda de materiales a catalogar.	
2. El usuario del sistema introduce los filtros por los que desea buscar, los que pueden ser: <ul style="list-style-type: none"> • Procedencia (obligatorio) (B) • Intervalo de fecha de inserción (opcional) (C) • Fecha inicial (obligatorio si selecciona intervalo de fecha de inserción) (D) • Fecha final (obligatorio si selecciona intervalo de fecha de inserción) (E) 	2.1 El sistema verifica los valores introducidos.	
3. El usuario puede realizar una de las siguientes acciones: <ol style="list-style-type: none"> a) Presionar el botón “Buscar” (F), continua este Caso de Uso. b) Presionar el botón “Limpiar Campos” (G), ir al Caso de Uso Buscar Material, 	3.1. El sistema verifica que exista conexión con la Base de Datos. 3.2. El sistema muestra la lista de los materiales (H) que no han sido catalogados y que cumplen con los filtros introducidos en la búsqueda.	

sección "Limpiar búsqueda"	
4. Una vez obtenidos los resultados, el usuario puede seleccionar uno de los resultados de la búsqueda y seleccionar la opción de catalogarlo (I), ir al Caso de Uso Catalogar Material.	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.1. Si no se puede establecer conexión con la Base de Datos, el sistema muestra el siguiente mensaje: "No se puede establecer conexión con los datos." Culmina el caso de uso.
	3.2. Si no existen materiales que cumplan con los filtros introducidos por el usuario la lista de resultados no mostrará ningún elemento indicando que no existen coincidencias con los filtros introducidos y culmina el caso de uso.
Pos condiciones	Se obtienen los materiales no catalogados de acuerdo a los filtros especificados por el usuario.

GLOSARIO DE TÉRMINOS

Internet: Red informática de comunicación internacional que permite el intercambio de todo tipo de información entre sus usuarios.

Web: Es un vocablo inglés que significa red, telaraña o malla. El concepto se utiliza en el ámbito tecnológico para nombrar a una red informática y, en general, a internet.

Medias: Película, imagen o cualquier otro material audio visual que requiere de un uso especial de equipamiento para visualizarlo.

Software: Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

Robot: Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas.

Buscador: Sistema basado en el uso de un software que recorre la red automáticamente para localizar documentos, indexarlos e introducirlos en una base de datos, que puede ser consultada utilizando programas de consulta.

Google: Es un buscador. Se encuentra en la dirección: <http://www.google.com>.

Hotboot: Es un buscador. Se encuentra en la dirección: <http://www.hotbot.com>.

Lycos: Es un buscador. Se encuentra en la dirección: <http://www.lycos.com>.

Altavista: Es un buscador. Se encuentra en la dirección: <http://www.altavista.com>.

Bing: Es un buscador. Se encuentra en la dirección: <http://www.bing.com>.

Yahoo: Es un buscador. Se encuentra en la dirección: <http://www.yahoo.com>.

Directorio: Agrupación de archivos de datos, atendiendo a su contenido, a su propósito o a cualquier criterio que decida el usuario.

Internet Public Library: Es un directorio. Se encuentra en la dirección: <http://www.ipl.org>.

Metabuscaadores: Buscadores que no se limitan a recuperar la información en una única base de datos, realizan la búsqueda en diversas bases de datos

Kartoo: Es un metabuscador. Se encuentra en la dirección: <http://www.kartoo.com>.

Vivísimo: Es un metabuscador. Se encuentra en la dirección: <http://www.vivisimo.com>.

Buscadores selectivos: Utilizan una base de datos especializada en una materia en específico.

Electric Library: Es un buscador selectivo. Se encuentra en la dirección: <http://www.elibrary.com>.

Teoma: Es un buscador selectivo. Se encuentra en la dirección: <http://www.teoma.com>.

Agentes Inteligentes: Herramientas que recuperan la información de forma automática, solo necesitan un criterio de búsqueda y el lugar donde van a realizar la recuperación.

Webferret: Es un agente inteligente. Se encuentra en la dirección: <http://www.ferretsoft.com>.

WebSeeker 5: Es un agente inteligente. Se encuentra en la dirección: <http://www.bluesquirrel.com>.

Answers: Sitio web. Se encuentra en la dirección: <http://es.answers.yahoo.com>.

Delicious: Sitio web. Se encuentra en la dirección: <http://www.delicious.com>.