



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 4

**ANÁLISIS Y DISEÑO DE LA VERSIÓN 2.0 DEL MÓDULO MEDiateca DE
LA COLECCIÓN DE SOFTWARE EDUCATIVO “EL NAVEGANTE”.**

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

AUTORA

EDITH GÓMEZ ROQUE

TUTORES

ING. ANA MARÍA ÁLVAREZ VALDÉS

ING. RAFAEL JACOBO HIDALGO URBINO

LA HABANA, CUBA, ENERO 2011

“AÑO 53 DE LA REVOLUCIÓN”

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de junio del año 2011.

Edith Gómez Roque

Ing. Ana M. Álvarez Valdés

Ing. Rafael J. Hidalgo Urbino

Firma del Autor

Firma del Tutor

Firma del Tutor

"No basta dar pasos que puedan conducir hasta la meta; sino que cada paso sea una meta, sin dejar de ser un paso".

Johann Peter Eckerman

DEDICATORIA

A mota, por estar conmigo en cada paso que di a pesar de la distancia, por darme aliento en los momentos más difíciles y por todo el esfuerzo que has realizado en aras de mi bienestar. Por estar con los brazos abiertos siempre que los necesito, a ti que sabes comprender cuando necesito a una amiga. Porque tus ojos han sabido endurecer en los momentos que necesito una lección, así como tu fuerza, tu amor y tu sonrisa me ayudan a levantarme de mis caídas. Eres el más preciado tesoro que agradeceré eternamente, gracias por llenar mi vida, día a día, de tu cariño y amor.

A yeye también dedico este trabajo, a retribuirte una pequeña fracción de lo mucho que has hecho por mí y la confianza depositada.

A Ate por poner toda tu fe en mí, por apoyarme en todas mis decisiones, por tus consejos y tu cariño.

A toda mi familia, todos los que saben que son mis amigos, siéntanse parte de esta dedicatoria.

AGRADECIMIENTOS

A la Revolución, por ser artífice de este gran proyecto.

A mi familia, que siempre me ha apoyado y ha depositado su confianza en la materialización de este logro.

A foshito, por tu apoyo incondicional, por darme todo tu amor, por estar siempre a mi lado en los buenos y malos momentos.

Al nené Ariel porque gracias a ti y tu gran amistad he podido lograr este triunfo.

A todos mis compañeros y profesores de toda una vida de estudiante.

A todos, agradecida infinitamente, quiero que sientan que el objetivo logrado también es de ustedes y que la fuerza que me ayudó a conseguirlo fue su apoyo.

RESUMEN

La Universidad de las Ciencias Informáticas ha estado trabajando en base a la implementación de software educativos de alta calidad que constituyen hiperentornos educativos o de aprendizaje. Como parte del convenio integral de cooperación Cuba-Venezuela en la X mixta, se realizó un contrato para el desarrollo de una nueva versión de la colección “El Navegante”, existente en las escuelas secundarias cubanas, teniendo en cuenta las necesidades de Venezuela. Durante el curso 2009-2010 fue desarrollado el módulo Mediateca, perteneciente a dicha colección, sin embargo no se cumplió con las expectativas del Ministerio del Poder Popular para la Educación venezolano producto a que se realizó antes del levantamiento de requisitos con el cliente. En estas condiciones surgió la necesidad de efectuar el análisis y diseño de la segunda versión del módulo Mediateca, siendo esto el objetivo del presente trabajo. Para cumplir tal tarea se hizo uso de la metodología de desarrollo Proceso Unificado de Desarrollo por las múltiples ventajas que esta posee, además se utilizó UML como lenguaje de modelado y la herramienta Visual Paradigm como apoyo para realizar los diagramas y modelos pertinentes a dicho sistema. Fueron definidos requisitos funcionales y no funcionales, se elaboró el diagrama de casos de uso del sistema y se hizo una descripción textual de cada uno, los diagramas de clases del análisis y el del diseño también fueron creados, constituyendo este trabajo un punto de partida para la implementación del módulo Mediateca.

PALABRAS CLAVES: software educativo, Mediateca

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	5
CONCEPTOS RELACIONADOS AL PROCESO DE SOFTWARE EDUCATIVO.....	5
ANÁLISIS DE SISTEMAS SIMILARES EXISTENTES	8
TECNOLOGÍAS Y HERRAMIENTAS A UTILIZAR	9
CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA.....	19
MODELO DE DOMINIO	19
DESCRIPCIÓN DEL SISTEMA PROPUESTO	22
ESPECIFICACIÓN DE LOS REQUISITOS	22
MODELO DE CASOS DE USO DEL SISTEMA.....	27
PATRONES DE CASOS DE USO	29
DESCRIPCIÓN DE CASOS DE USO DEL SISTEMA	31
CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA.....	38
MODELO DE ANÁLISIS.....	38
MODELO DE DISEÑO	45
PATRONES UTILIZADOS PARA EL DISEÑO	57
MODELO DE DATOS.....	59
CONCLUSIONES GENERALES	61
RECOMENDACIONES	62
REFERENCIAS BIBLIOGRÁFICAS.....	63

INTRODUCCIÓN

Durante los últimos años las Tecnologías de la Información y las Comunicaciones (TIC) han venido transformando la sociedad, y en particular a los procesos educativos, los que han sufrido diversas transformaciones: cambios en la educación, en las relaciones interpersonales, en las formas de difundir y generar el conocimiento, cambios en los modos de intervención docente, desarrollo de nuevos espacios didácticos en línea, entre otros. Estos cambios han comenzado a generar grandes expectativas, controversias, demandas y desafíos hacia los sistemas de enseñanza y aprendizaje.

La tecnología puede generar temor a lo desconocido y ser juzgada como perjudicial o contraproducente para los estudiantes. Pero lo cierto es que internet ha introducido ya en gran parte de la población notables cambios en los hábitos y costumbres, debido a esto el debate sobre las TIC ya no puede mirarse solo como un recurso respecto al cual la escuela puede decidir si incluir o no, porque sus alumnos y docentes coexisten en este nuevo contexto cultural lo que supone un desarrollo de habilidades intelectuales de las cuales la escuela no puede desentenderse. La incorporación de las nuevas tecnologías al ámbito educativo promueve la creación de nuevos entornos didácticos que afectan de manera directa tanto a los estudiantes como al escenario donde se lleva a cabo el proceso de enseñanza-aprendizaje. (1)

Con el objetivo de beneficiar la calidad de las clases en las escuelas cubanas, y así desarrollar todavía más el campo educativo, se han venido insertando con el transcurso de los años una serie de recursos informáticos, tomando protagonismo el software educativo como una herramienta de gran utilidad tanto para alumnos como para profesores.

Cuba, en particular la Universidad de las Ciencias Informáticas, ha estado trabajando en base a la implementación de software educativos de alta calidad que constituyen hiperentornos educativos o de aprendizaje. El desarrollo de estos software le ha posibilitado la realización de contratos con empresas extranjeras y convenios de trabajo con empresas nacionales.

Como parte del convenio integral de cooperación Cuba-Venezuela en la X mixta, se realizó un contrato para el desarrollo del proyecto “Colecciones de Software Educativo Multisaber y El Navegante para Planteles de los Niveles Primaria y Secundaria del Sistema Educativo“, con el objetivo de dotar al sistema escolar de la República Bolivariana de Venezuela de 24 software educativos, como herramientas pedagógicas en el proceso de enseñanza - aprendizaje.

Durante el curso 2009-2010 se desarrollaron varios trabajos de diploma con el objetivo de eliminar las limitantes de la colección “El Navegante” existente en las escuelas cubanas, y obtener un software implementado sobre estándares educativos, en el que cada producto tenga el mismo funcionamiento, desarrollado con herramientas libres, multiplataforma¹ y contextualizado al entorno venezolano. La realización del módulo Mediateca fue el objetivo de uno de estos trabajos de diploma, cuyo fin era permitir la visualización de imágenes, así como la reproducción de sonidos, videos, animaciones y diaporamas de la colección o de un tema específico dentro de la misma. Además, Mediateca facilitaría la consulta de palabras de difícil comprensión a través del Glosario de términos y de datos de personalidades relacionadas con las temáticas abordadas en el software.

Dicho módulo se desarrolló parcialmente antes de realizar el levantamiento de requisitos con el cliente, lo que, aunque significó un paso de avance, dio lugar a que no se cumpliera con las expectativas del Ministerio del Poder Popular para la Educación (en lo adelante MPPE) de Venezuela. Esta versión 1.0 no cuenta con un registro de las trazas de los estudiantes en las distintas secciones a las que accede; no brinda la posibilidad de consultar medias por temas y no permite que se amplíen las imágenes seleccionadas en el módulo administrativo; tampoco superó la etapa de pruebas.

Este trabajo surge de la necesidad de dar solución a las problemáticas antes expuestas; por lo que se deriva de aquí el siguiente problema científico: ¿Cómo lograr que el módulo Mediateca de la colección El Navegante cumpla con las funcionalidades requeridas por el MPPE de Venezuela?

Se precisó como objeto de estudio el proceso de desarrollo de software educativo.

¹ Un programa o aplicación multiplataforma puede utilizarse sin inconvenientes en distintas plataformas de hardware y sistemas operativos.

Para ofrecer una solución favorable se ha trazado como objetivo general de este trabajo diseñar la segunda versión del módulo Mediateca de la colección El Navegante de modo que cumpla con las funcionalidades requeridas por el MPPE de Venezuela.

Se deriva por tanto que el campo de acción queda enmarcado en el proceso de análisis y diseño de la versión 2.0 del módulo Mediateca de la colección El Navegante.

Consecuentemente se definieron los siguientes objetivos específicos:

- Elaborar el marco teórico de la investigación.
- Realizar el análisis de la versión 2.0 del módulo Mediateca de la colección El Navegante.
- Realizar el diseño de la versión 2.0 del módulo Mediateca de la colección El Navegante.

Para cumplir de forma satisfactoria con los objetivos descritos y resolver la situación problemática planteada, se definieron las siguientes tareas de investigación:

1. Caracterización de aplicaciones similares existentes.
2. Caracterización de la metodología, lenguajes y herramienta de modelado a utilizar en el desarrollo del análisis y diseño de la segunda versión del módulo Mediateca.
3. Definición de las funcionalidades especificadas por el MPPE para el módulo Mediateca.
4. Realización del modelo de sistema correspondiente al módulo Mediateca.
5. Realización de los diagramas de clases del análisis de la segunda versión del módulo Mediateca.
6. Realización de los diagramas de clases del diseño de la segunda versión del módulo Mediateca.

Las tareas de investigación determinan la estructura de este trabajo, el cual se divide en tres capítulos que incluyen los resultados obtenidos en el trabajo educativo.

Capítulo 1: Fundamentación teórica. Comprende un análisis de los sistemas existentes vinculados con la investigación, y el estudio de las tecnologías y herramientas que serán necesarias dominar para un correcto análisis y diseño de la aplicación.

Capítulo 2: Descripción de la solución propuesta. Se identifican las necesidades del cliente. Se explican las funcionalidades del sistema. Se describe la solución propuesta, utilizando los casos de uso y el Diagrama de casos de uso del sistema.

Capítulo 3: Análisis y diseño del sistema. Se realiza el Diagrama de clases del análisis con el propósito de refinar y estructurar los requisitos obtenidos para facilitar la comprensión, preparación, modificación y mantenimiento de los mismos. A partir del análisis se confecciona el Diagrama de clases del diseño obteniéndose una visión más detallada del sistema.

CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

1.1 INTRODUCCIÓN

Este capítulo es el resultado de la búsqueda y el análisis de la información vinculada a los procesos de desarrollo de software educativo. Se abordan en él conceptos asociados como software educativo e hiperentorno de enseñanza–aprendizaje. También completa un análisis de sistemas similares existentes y se hace una descripción de las herramientas y tecnologías que serán necesarias dominar para un correcto análisis y diseño del software. Para este análisis se necesitó información que ha sido obtenida, consultada y recopilada de una amplia bibliografía.

1.2 CONCEPTOS RELACIONADOS AL PROCESO DE SOFTWARE EDUCATIVO

La necesidad de introducir la tecnología en los procesos docentes se ha hecho evidente, desempeñando un papel fundamental el uso del software educativo. A continuación se expone un breve repaso de las definiciones más comunes que se manejan entorno al desarrollo de este.

Software educativo

Se reconoce como software educativo a los programas didácticos que facilitan impartir el conocimiento a una audiencia, además de adaptarlo al ritmo de trabajo y las necesidades de cada educando, para lograr diversas alternativas de aprendizaje. (2)

Según Rodríguez Lamas, software educativo es una aplicación informática, que soportada sobre una bien definida estrategia pedagógica, apoya directamente el proceso de enseñanza–aprendizaje constituyendo un efectivo instrumento para el desarrollo educacional del hombre del próximo siglo. (3)

Los software educativos pueden tratar diferentes materias de formas muy diversas, y ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los alumnos y más o menos rico en posibilidades

de interacción. Están caracterizados por ser altamente interactivos a partir del empleo de recursos multimedia. (4)

Recursos multimedia

Los componentes multimedia son los diversos medios de comunicación digitalizados que tiene el hombre para transmitir un mensaje con imágenes de cualquier tipo, sonidos, películas de video, y hasta el mismo texto, y constituyen un elemento crucial en la calidad de los software educativos, lo que posibilita que esta herramienta de apoyo al proceso de enseñanza - aprendizaje sea más atractiva e interactiva.

Algunos sistemas trabajan con la filosofía de incrustación, que es cuando un recurso se integra físicamente en el nodo sin que exista ningún enlace para ello; la ventaja de la incrustación reside en la simplificación en la consulta y mantenimiento de recursos.

Otros sistemas orientan la integración de recursos multimedia hacia el enlace: cada imagen, sonido, texto o video son considerados como nodos individuales, que posteriormente se enlazan entre sí en un único nodo "maestro" (nodos compuestos). El método anterior tiene la ventaja de que cualquier cambio en uno de los nodos repercute en los nodos compuestos. La web utiliza esta filosofía ya que para que en una página aparezca una imagen se debe especificar el fichero de la misma, así los ficheros de un servidor pueden ser considerados como nodos.

La tecnología multimedia permite el acceso, a través de la computadora, a información de diversa naturaleza (una película, un sonido, una imagen, un texto), permitiendo la posibilidad de interactuar de manera dinámica con dichos recursos, superando la rigidez del sistema lineal que solo permitía transitar entre distintos menús. De este modo, habilita a aproximarse cada vez más a los modos del pensamiento humano, en virtud de la posibilidad de asociar, elegir y acceder a la información almacenada en la computadora. (5)

La interactividad es, en este caso, la que potencia su uso para la adquisición de saberes y el aprendizaje de estrategias que habilitan al estudiante en el conocimiento del sistema y consolidan sus propias habilidades.

Hipermedia

La tecnología hipermedia es ampliamente utilizada para la producción de aplicaciones informáticas con disímiles propósitos. Su empleo con fines docentes gana cada vez más espacio en los currículos de los diferentes niveles de enseñanza a nivel internacional y es ya una realidad su presencia en las escuelas cubanas.

Un sistema hipermedia es una estructura similar a la del hipertexto (organiza la información en forma de red, entendiendo esta como un modo de unión entre los elementos de información que se caracteriza porque no existe un orden único de recorrido entre ellos, no tiene principio ni fin), con la particularidad de que esta información es multimedia (textos, imágenes, secuencias de animaciones, sonidos, video). En esta red de información se distinguen tres elementos fundamentales: los nodos, que son los contenedores de la información; las anclas que indican las zonas de información que pueden estar relacionadas con otras; y los enlaces, que representan relaciones entre la información que contienen los nodos. (6)

En un hipermedia se pueden utilizar dos formas básicas para consultar la información que se encuentra almacenada en sus nodos: la navegación y la búsqueda. La navegación es considerada la forma primaria de acceso a la información en un hipermedia y consiste en recorrer los nodos a través de los enlaces, consultando la información que en él se almacena. En la búsqueda el usuario solicita al sistema el acceso directo a un determinado dato. (6)

Hiperentorno de enseñanza–aprendizaje

La colección El Navegante puede ser clasificada como un hiperentorno de enseñanza–aprendizaje, siendo este uno de los tipos de software educativo más difundido en el ámbito escolar contemporáneo, y se basa en la tecnología hipermedia.

Los hiperentornos de enseñanza–aprendizaje son los sistemas hipermedia que están concebidos para ser utilizados como medios de enseñanza–aprendizaje por profesores y estudiantes. Estos hiperentornos, que tienen en común la utilización de un ambiente informático con una intención didáctica, integran los agentes humanos (estudiantes, profesores) y los artificiales (recursos informáticos) y ofrecen condiciones de interacción local o a través de redes.

El concepto de hiperentorno de enseñanza–aprendizaje es, como plantea Pierre Tchounikine, lo suficientemente general para incluir tanto una plataforma de educación abierta o a distancia destinada a la difusión de recursos pedagógicos como un software para el aprendizaje de una determinada competencia de álgebra. (7)

Los hiperentornos de enseñanza–aprendizaje se distinguen por las amplias potencialidades tecnológicas que poseen: (6)

- La integración de varias formas de presentación de la información (textos, imágenes, sonidos, animaciones, videos).
- Las posibilidades de interactividad de los estudiantes con los contenidos que le permiten el control de su aprendizaje.
- Las herramientas de comunicación de que se disponen para el trabajo colaborativo.
- Las capacidades de almacenamiento de información sobre el trabajo realizado que permiten evaluar los resultados con carácter procesal.

1.3 ANÁLISIS DE SISTEMAS SIMILARES EXISTENTES

A medida que avanza la informatización de la sociedad cubana se introducen en las escuelas recursos informáticos con el propósito de beneficiar la calidad de las clases en las aulas. En este apartado se abordará acerca de dos colecciones con que cuenta la escuela cubana: “Multisaber” y “El Navegante”, tratando específicamente estas dos pues tienen integrado un módulo dedicado a la interacción del usuario con recursos multimediales.

La colección Multisaber es un grupo de multimedias educativas destinadas a la enseñanza primaria. Está compuesta por 32 software e incluye en su modelo pedagógico seis módulos: Temas, Ejercicios, Biblioteca, Profesor, Resultados y Juegos. El módulo Biblioteca es el encargado de visualizar recursos multimedia (imágenes, videos, glosario), no permitiendo adicionar, eliminar y modificar ninguno de ellos.

La versión de la colección “El Navegante” instalada en las escuelas secundarias cubanas está conformada por 10 productos y a su vez cada uno de ellos está compuesto por seis módulos básicos. Entre ellos existe un módulo “Mediateca”, encargado de la interacción del usuario con los recursos multimedia de la colección.

Algunos de los inconvenientes que presentan dichos software son: no están implementados sobre estándares educativos, el funcionamiento de cada uno de sus productos no es el mismo, están desarrollados con herramientas propietarias, no son multiplataforma lo que limita su utilización en otros sistemas operativos, sus contenidos se encuentran contextualizados a Cuba haciendo imposible su uso en entornos educativos de otros países.

1.4 TECNOLOGÍAS Y HERRAMIENTAS A UTILIZAR

En este apartado se destacarán las cualidades más significativas que poseen las tecnologías y herramientas seleccionadas, como producto del estudio realizado en investigaciones anteriores por el equipo de desarrollo de la colección, para llevar a cabo la realización del módulo.

1.4.1 Estándar educativo SCORM

El estándar SCORM (*Sharable Content Object Reference Model*, o Modelo de Referencia para Objetos de Contenidos Intercambiables) es el estándar de e-learning² más utilizado a nivel mundial y es un modelo conceptual que describe cómo administrar, empaquetar y distribuir el material educativo en cualquier momento y en cualquier lugar. Asegura que este material sea: reutilizable, accesible, interoperable y durable: (8)

- Reusabilidad: Se enfoca en reutilizar lo que ya existe, disminuir los tiempos de producción y aumentar la calidad de los contenidos.
- Accesibilidad: Apunta a que los contenidos necesarios puedan ser accedidos en cualquier momento desde cualquier lugar y mediante los dispositivos disponibles.
- Interoperabilidad: Ofrece la capacidad de que una plataforma pueda exhibir contenidos independientemente de cómo fueron creados y de producir contenidos independientemente de la plataforma en la que serán incorporados.

² Simplificación de *Electronic Learning* o aprendizaje electrónico. Enseñanza a distancia caracterizada por una separación física entre profesorado y alumnado, entre los que predomina una comunicación de doble vía asíncrona donde se usa preferentemente Internet como medio de comunicación, de tal manera que el alumno es el centro de una formación independiente y flexible, al tener que gestionar su propio aprendizaje.

- Durabilidad: Es el principio destinado a impedir la caducidad tecnológica de los contenidos y de los estándares.

SCORM define dos principales tipos de objetos que pueden estar integrados a un contenido: Asset y SCO. Los Asset (Activos) son los objetos elementales que se pueden encontrar en un contenido y se trata de: texto, imágenes, sonidos, videos u otros datos que puedan ser brindados al usuario. Los SCO (*Shareable Content Object* u Objeto de Contenido Compartible) están compuestos por los mismos materiales que los assets pero a diferencia tienen la capacidad de comunicarse con la plataforma LMS³. (8)

Para el desarrollo de la aplicación se va a hacer uso de SCORM pues permite lograr una organización de los contenidos y en próximas versiones hacerlos reutilizables. En la UCI se está utilizando para describir parte del contenido que aparece en el Entorno Virtual de Aprendizaje. También, como parte del proyecto productivo en desarrollo Alfaomega, se espera obtener una plataforma al estilo LMS compatible con SCORM y una herramienta de autoría de objetos de aprendizaje.

1.4.2 Metodología de desarrollo de Software RUP

Para una buena realización de software se debe definir una metodología que estructure, planee y controle el proceso de desarrollo para garantizar la construcción del producto en el tiempo planificado, con el coste esperado y que reúna los parámetros de calidad exigidos por el cliente. Todo desarrollo de software es riesgoso y difícil de controlar, y si no se utiliza una metodología, lo que se obtiene son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

Para lograr un correcto desarrollo de software, estas metodologías establecen un conjunto de actividades que definen cómo se debe hacer el software, quién debe hacer cada actividad, cuándo hacerla y qué se debe hacer.

En la actualidad existen gran cantidad de metodologías orientadas al proceso de desarrollo de software, entre las más significativas se pueden mencionar el Proceso Unificado de Desarrollo (*Rational Unified Process* en inglés, habitualmente resumido como RUP) y Programación Extrema (*eXtreme Programming: XP*).

³ **LMS** (*Learning Management Systems* o Sistemas de Gestión de Aprendizaje): Se trata de una plataforma que permite hacer las funciones necesarias para gestionar cursos de formación a distancia (aunque pueden usarse como complemento en la enseñanza presencial).

Teniendo en cuenta las características, complejidad y envergadura del producto El Navegante se ha establecido el uso de la metodología RUP pues captura las mejores prácticas del conocimiento de líderes en ingeniería de software y proporciona a los equipos de desarrollo guías, estándares y recomendaciones para la construcción de software de alta calidad.

Constituye un proceso de software genérico que puede especializarse para modelar el desarrollo de una diversa variedad de sistemas, para diferentes áreas de aplicación, en distintos tipos de organizaciones y con desiguales niveles de aptitud. Es un marco de desarrollo de software dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. El cliente no debe necesariamente tomar un papel activo en el desarrollo del producto. (9)

Asociado al desarrollo se genera una abundante documentación que puede ayudar a entender y mantener el sistema. Es orientado a objetos.

Según “La guía completa del Proceso Unificado, escrita por sus creadores”, RUP divide en 4 fases el desarrollo del software. En la fase de **Inicio** se describe el negocio y se delimita el proyecto, describiendo el alcance con la identificación de los casos de uso del sistema. Durante la **Elaboración** se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. Mediante la fase de **Construcción** se logra obtener un producto listo para ser utilizado, el cual está documentado y cuenta con un manual de usuario. En la fase de **Transición** la versión final ya se encuentra lista para su instalación en condiciones reales del entorno de trabajo, durante esta fase se pueden reparar algunos errores del producto. (10)

En cada una de estas fases tienen lugar iteraciones de varios flujos de trabajo, los 6 primeros son conocidos como flujos de ingeniería y los 3 últimos de apoyo: (10)

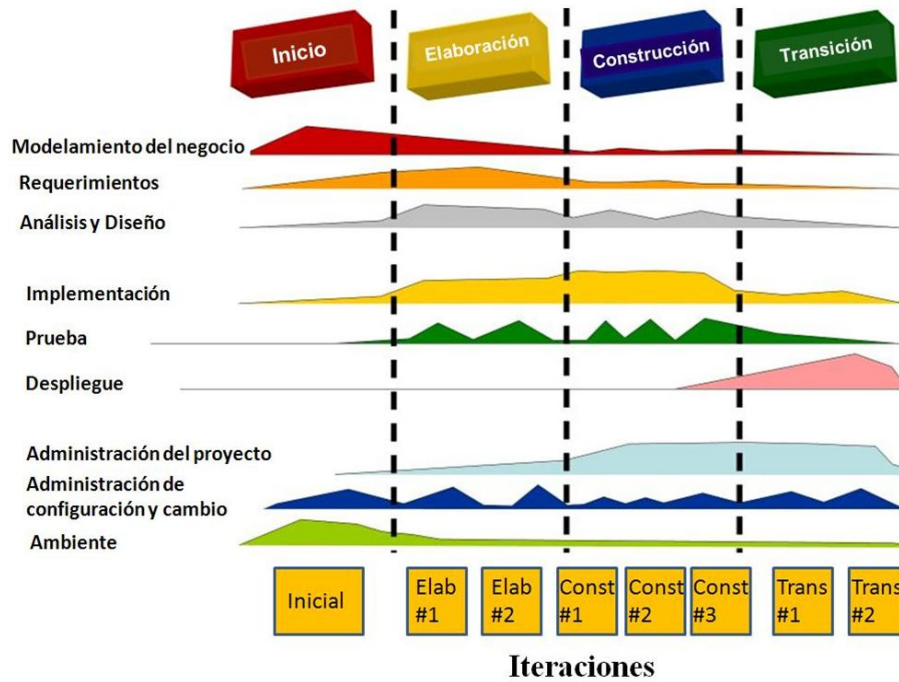


Figura 1: Fases y flujos de trabajos de RUP.

1.4.3 Lenguaje de modelado de sistemas de software

Para el desarrollo de este trabajo se utilizará el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés: *Unified Modeling Language*), pues es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. (11)

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a metodologías de desarrollo de software pero no especifica en sí mismo qué metodología o proceso usar.

Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida y dominios de aplicación. Mejora sustancialmente el desarrollo de los proyectos, pues facilita a integrantes de un equipo

multidisciplinario participar e intercomunicarse fácilmente y permite transformar los requisitos de un usuario en un sistema de software.

Hoy en día está considerado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo.

1.4.4 Herramienta CASE de Modelado con UML

CASE es una sigla, que corresponde a las iniciales de *Computer Aided Software Engineering* o en español a Ingeniería de Software Asistida por Computadoras. Estas herramientas influyen directamente en el aumento de la velocidad de los equipos de desarrollo, ahorrando horas de trabajo. Aumentan la productividad logrando así una reducción en el costo en términos de tiempo y de dinero. Las herramientas CASE de modelado con UML permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraerse del código fuente, en un nivel donde la arquitectura y el diseño son más fáciles de entender y modificar. (12)

Establecer una arquitectura racional y eficiente, acorde a las necesidades de los actores, es la razón fundamental para invertir tiempo y esfuerzos en una herramienta CASE que utiliza la notación UML.

Visual Paradigm para UML (6.4) permite la utilización de UML como lenguaje de modelado. Soporta el ciclo de vida completo de un software: Análisis, Diseño, Construcción, Pruebas y Despliegue. Permite modelar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Se integra con Netbeans, IDE⁴ seleccionado para el desarrollo del módulo. Es una herramienta multiplataforma que ayuda a construir productos de mejor calidad, de forma rápida y con poco coste, soporta aplicaciones web, es fácil de instalar y actualizar, es amigable, es gratuita y comercial. También proporciona abundantes tutoriales, demostraciones interactivas y proyectos de UML.

Estas razones hacen de Visual Paradigm la herramienta seleccionada para llevar a cabo el modelado visual de la nueva versión de Mediateca.

⁴ Integrated Development Environment o Entorno de desarrollo integrado: Software que su objetivo es el desarrollo de otro software. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Puede dedicarse en exclusiva a un lenguaje de programación específico o bien para varios.

1.4.5 Patrón arquitectónico Modelo Vista Controlador

Según su definición oficial en la IEEE⁵ Std 1471-2000 “La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”. (13)

La arquitectura no está atada a ninguna metodología de desarrollo, sino que es en sí una disciplina que evoluciona durante todo el ciclo de desarrollo de software. Abarca todo el proceso de desarrollo para dar distribución y ordena los elementos de un sistema informático.

Los patrones de arquitectura son modelos que proporcionan reglas y guías para organizar un sistema de software, permitiéndole a los programadores, diseñadores, ingenieros y analistas trabajar bajo una línea común que les posibilite la compatibilidad necesaria para lograr el objetivo deseado.

Teniendo en cuenta que la aplicación que se desea crear es interactiva con una interfaz humano-computadora flexible, se determinó que la mejor manera de estructurar el sistema es haciendo uso del patrón Modelo Vista Controlador o *Model View Controller*.

MVC (por sus siglas en inglés) es un patrón de diseño de arquitectura de software usado principalmente en aplicaciones complejas que manejan gran cantidad de datos, fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas.

Este modelo de arquitectura presenta varias ventajas: (14)

- Hay una clara separación entre los componentes de un programa; permitiendo implementarlos por separado.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- Facilita el mantenimiento en caso de errores.
- Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.

⁵ The Institute of Electrical and Electronics Engineers, más conocido por su sigla IEEE ('I-triple-E') es la sociedad técnica profesional más grande del mundo. El IEEE es el mayor distribuidor de información técnica en el mundo, produce casi el 30% de la literatura técnica publicada en electricidad, electrónica, e informática.

- La conexión entre el modelo y sus vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución. Si uno de los componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas. (14)

1.4.6 Lenguajes de Programación

Se hace necesario el uso de varios lenguajes de programación pues permiten crear programas mediante un conjunto de comandos, operadores y reglas de sintaxis que ponen a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

Los lenguajes del lado del cliente son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre-tratamiento. Para el desarrollo de la aplicación se usarán:

XHTML

eXtensible Hypertext Markup Language o Lenguaje de Marcado de Hipertexto Extensible, extiende HTML 4.0 combinando la sintaxis de HTML (diseñado para mostrar datos) con la de XML (diseñado para describir los datos).

XHTML, está orientado al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir en lo que a código se refiere. Entre estos requisitos básicos se puede mencionar, una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente, atributos de valores entrecomillados.

JavaScript

Es un lenguaje multiplataforma interpretado por la mayoría de los navegadores en sus últimas versiones por lo que no es necesario compilar los programas para ejecutarlos, es decir, los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Es pequeño y ligero, no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. Se trata de un lenguaje de tipo

script⁶ compacto, basado en objetos (no orientado a objeto por lo que no dispone de herencia) y guiado por eventos. Los programas JavaScript van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos. Es muy utilizado por su fiabilidad y seguridad ya que los scripts tienen capacidades limitadas. Es ejecutado en el cliente por lo que el servidor no es utilizado más de lo debido y como desventaja el código del script debe descargarse completamente para poder ejecutarlo.

CSS

Cascading Style Sheets u Hojas de Estilo en Cascada: Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar la presentación de sus contenidos y es imprescindible para crear páginas web complejas. CSS permite definir estilos propios en un archivo externo a las páginas; así, si en algún momento se quiere cambiar alguno de ellos, automáticamente se actualizarán todas las páginas vinculadas al sitio.

Los lenguajes del lado del servidor son aquellos lenguajes reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible. Para esta investigación se decidió utilizar:

PHP

Hypertext Preprocessor: Lenguaje muy utilizado para la generación de sitios web dinámicos, ejecutados en el servidor. No necesita ser compilado para ejecutarse. Soporta en cierta medida la orientación a objeto. Se caracteriza por ser un lenguaje muy rápido, multiplataforma, capaz de conectarse con la mayoría de los manejadores de base de datos y es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Como desventaja se puede mencionar que para la realización de proyectos complejos requiere un alto nivel de experiencia, y que por determinadas limitaciones del lenguaje existe un pequeño grupo de tareas que resultan muy difíciles de realizar en el mismo. (15)

⁶ Un lenguaje de script es un pequeño lenguaje de programación que su código puede ser insertado por ejemplo dentro de una página HTML.

1.4.7 Frameworks

Como su nombre lo indica, entregan una estructura, armazón o marco para desarrollar proyectos. Persiguen acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de implementación. Permite a los programadores pasar más tiempo identificando requerimientos que tratando los tediosos detalles de bajo nivel. La mayoría de los frameworks implementan uno o más patrones de diseño de software para asegurar la escalabilidad del producto. (16)

Es necesario entender que un framework es una abstracción de código común que provee funcionalidades genéricas que pueden ser utilizadas para desarrollar aplicaciones de manera rápida, fácil, modular y sencilla, ahorrando tiempo y esfuerzo. (17)

Symfony

Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. A su favor tiene que es fácil de instalar y configurar en sistemas Linux, funciona con todas las bases de datos comunes, se adapta con facilidad a las políticas y arquitecturas propias de cada empresa u organización. Se hará uso de su versión 1.4 la que automatiza la mayoría de elementos comunes de los proyectos web.

Symfony es el framework más documentado del mundo, cuenta con miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales. Aumenta exponencialmente la productividad y ayuda a mejorar la calidad de las aplicaciones web, aplicando todas las buenas prácticas y patrones de diseño que se han definido para la web. (18)

jQuery

Es una biblioteca JavaScript muy rápida y muy ligera que simplifica la manera de interactuar con los documentos HTML⁷ y el desarrollo del lado del cliente de las aplicaciones web. Las aplicaciones realizadas con jQuery pueden responder de forma mucho más rápida que las aplicaciones JavaScript tradicionales. (19)

⁷ HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web.

Esta potente biblioteca encabeza las listas de bibliotecas JavaScript más utilizadas gracias a su sencillez, su flexibilidad, su potencia y su rendimiento. Es un producto serio, estable, bien documentado, con un gran equipo de desarrolladores a cargo de su mejora y actualización, muy aceptado por los programadores y con un grado de penetración en el mercado muy amplio, lo que invita a su uso como una de las mejores opciones para el desarrollo. (20)

1.5 CONCLUSIONES DEL CAPÍTULO

Dada la necesidad de desarrollar una nueva versión del módulo Mediateca en este capítulo se realizó un amplio estudio de los temas teóricos que soportan la investigación. Se abordan las tecnologías y herramientas seleccionadas para llevar a cabo el desarrollo exitoso del nuevo subsistema. Se exponen las razones de la necesidad de utilizar RUP como metodología de desarrollo, además se presentan las principales características por las que se eligió UML como lenguaje de modelado y a Visual Paradigm para UML como herramienta CASE. La aplicación que se desarrollará será web y el patrón arquitectónico a utilizar es el MVC. Se utilizarán los lenguajes de desarrollo XHTML, CSS, JavaScript y PHP, además los frameworks jQuery y Symfony.

CAPÍTULO 2

CARACTERÍSTICAS DEL SISTEMA

2.1 INTRODUCCIÓN

En este capítulo se abordan las características que el sistema debe tener y se detallan los procesos esenciales que influyen en el objeto de estudio. Se aborda el funcionamiento general del subsistema mediante un modelo de dominio, retocando cada uno de los conceptos que lo conforman para que sea más entendible el contexto en que se ubica el sistema. Se hace una descripción de la propuesta del sistema y se enumeran los requisitos funcionales y los no funcionales especificados por el cliente, lo que permite hacer una concepción general del producto.

2.2 MODELO DE DOMINIO

El primer paso del modelado del negocio consiste en capturar los procesos de negocio de la organización bajo estudio. La definición del conjunto de procesos del negocio es una tarea crucial, ya que define los límites del proceso de modelado posterior. En la presente investigación no se realizará un modelo de negocio, pues no fue posible identificar ningún límite en el negocio que se desarrolla y existen procesos que no se encuentran bien definidos. Se hace necesario entonces describir el problema a través de un modelo de dominio, también conocido como modelo conceptual.

El modelo de dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real, no de los componentes de software. En este modelo no se muestra comportamiento. Las clases conceptuales pueden tener atributos pero no métodos. (21)

El modelo de dominio se describe mediante diagramas UML, específicamente mediante diagramas de clases conceptuales significativas en el dominio del problema, donde se muestran las clases del dominio y cómo se relacionan unas con otras a través de asociaciones. (10)

Su principal objetivo es ayudar a los implicados en el desarrollo del producto a utilizar un vocabulario común que posibilite una mejor comprensión entre ellos, para poder entender y describir las clases más importantes dentro del contexto donde se ubica el sistema. Además permite y facilita el levantamiento de requisitos así como definir los procesos y roles más significativos.

Análisis de los conceptos del dominio

El siguiente glosario o diccionario de datos se compone por los principales conceptos identificados en el contexto del problema y que se utilizarán en la confección del modelo de dominio. Tiene como objetivo principal permitir la comprensión del entorno y del problema.

Colección El Navegante: Colección de software educativo destinada a apoyar el proceso de enseñanza-aprendizaje de los niños en las escuelas secundarias venezolanas.

Producto multimedia: Producto que combina diversos tipos de medias.

Módulo: Se le denomina al conjunto de elementos fundamentales que componen el producto.

Temas: Módulo en el que se representan los contenidos de las asignaturas o temas a los que responde el producto.

Ejercicios: Módulo que permite ejercitar y comprobar las habilidades adquiridas por los estudiantes.

Mediateca: Módulo que agrupa las medias presentes en la colección.

Juegos: Módulo que contiene los juegos que el estudiante tiene a su disposición para profundizar en algún contenido indicado por el profesor.

Resultados: Módulo que visualiza la traza del estudiante durante la interacción de los mismos con los restantes módulos de la aplicación.

Maestro(a): Módulo que permite al profesor configurar opciones del producto.

Media: Se utiliza para referirse a los textos, imágenes, sonidos, videos, animaciones o diaporamas.

Imagen: Representación gráfica de una expresión.

Video: Es la tecnología de la captación, grabación, procesamiento, almacenamiento, transmisión y reconstrucción por medios electrónicos digitales o analógicos de una secuencia de imágenes que representan escenas en movimiento.

Diaporama: Sucesión de imágenes o de documentos conectados por efectos y en los cuales es posible reproducir sonidos.

Animación: La animación es una secuencia lineal de imágenes, ya sean dibujos o fotos reales, la gran parte de las veces incluye audio, voces o música de fondo con la finalidad de transmitir un mensaje.

Texto: Es una composición de signos codificados en un sistema de escritura (como un alfabeto) que forma una unidad de sentido.

Sonido: Media que brinda información sonora.

Imágenes: Sección del módulo Mediateca que agrupa las imágenes por temas disponibles en el software.

Videos: Sección del módulo Mediateca que agrupa los videos por temas disponibles en el software.

Diaporamas: Sección del módulo Mediateca que agrupa los diaporamas por temas disponibles en el software.

Personalidades: Sección del módulo Mediateca que contiene las personalidades relacionadas con los temas abordados en el software.

Glosario: Sección del módulo Mediateca que contiene términos de difícil comprensión relacionados con los temas abordados en el software.

Diagrama del modelo de dominio

A continuación se muestra cómo interactúan los conceptos manipulados en el sistema:

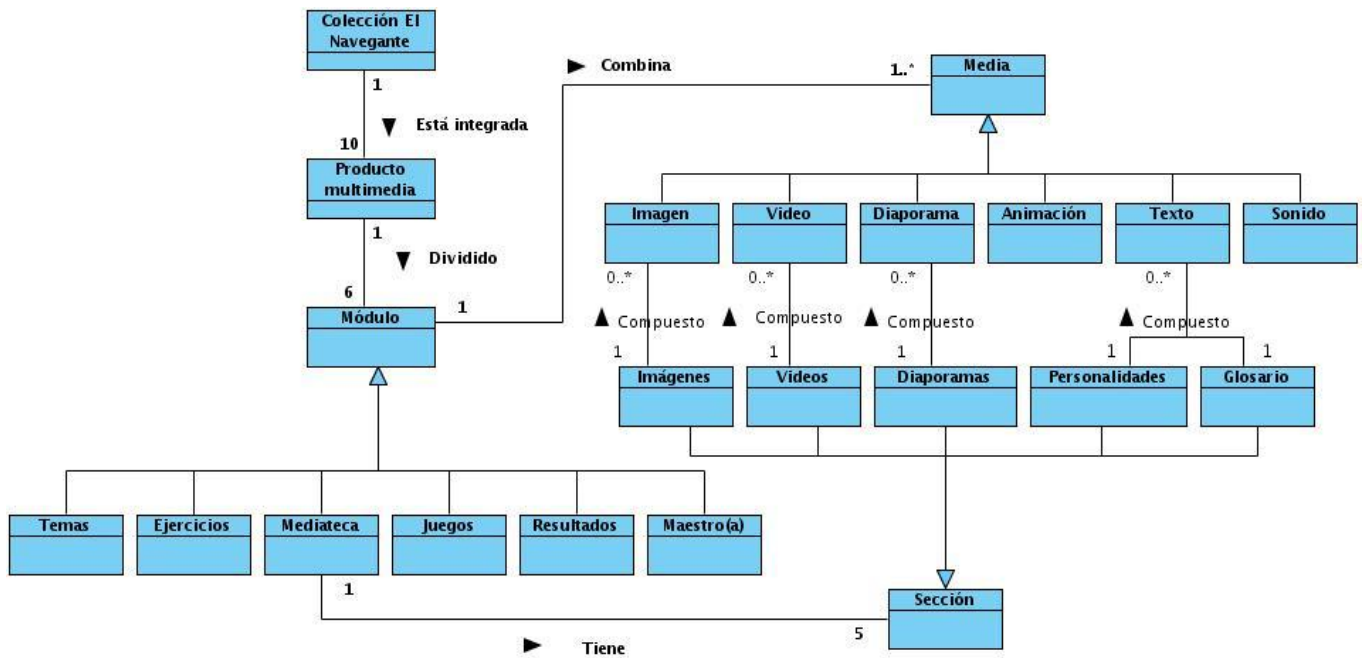


Figura 2: Diagrama de clases del dominio.

2.3 DESCRIPCIÓN DEL SISTEMA PROPUESTO

A partir del estudio realizado a la primera versión del módulo Mediateca de la colección El Navegante se identificó la necesidad de añadir nuevas funcionalidades a las aplicaciones que componen dicho módulo: la de gestión y la de visualización de las medias. Se hace necesario entonces, analizar y diseñar una nueva versión del módulo Mediateca con nuevas funcionalidades, teniéndose en cuenta la realización del modelo del sistema, las descripciones de los casos de uso, así como los diagramas de clases del análisis y los diagramas de clases del diseño, artefactos que permitirán obtener una visión amplia del sistema y constituirán la base para la realización de los flujos de implementación y pruebas de la aplicación.

2.4 ESPECIFICACIÓN DE LOS REQUISITOS

Una especificación de requisitos de software es una descripción completa del comportamiento del sistema a desarrollar. Una vez en este punto, se hace necesario comenzar a definir qué es lo que debe hacer el sistema, siendo necesario pasar a la captura de los requerimientos que este debe cumplir. Según Roger

S. Pressman para que un esfuerzo de desarrollo de software tenga éxito, es esencial comprender perfectamente los requisitos del software.

Los requerimientos son las características y cualidades que el sistema debe tener y se agrupan en funcionales y no funcionales. Los requerimientos funcionales definen las acciones que el sistema será capaz de realizar, son capacidades o condiciones que debe cumplir. Los requerimientos no funcionales describen las características del sistema como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad y estándares. Una vez definidos e identificados los mismos, se puede transitar al modelamiento del sistema. (9)(23)

Listado de los Requisitos Funcionales (RF)

RF1. Mostrar la pantalla principal del módulo Mediateca.

RF1.1: Mostrar las imágenes que representan metafóricamente a cada una de las secciones que integran la Mediateca.

RF2. Mostrar cada una de las secciones que integran la Mediateca: Videos, Imágenes, Sonidos, Glosario, Diaporamas, Personalidades y, Animaciones.

RF2.1: Mostrar los elementos de Videos, Imágenes, Sonidos, Diaporamas y Animaciones agrupados por temas.

RF3. Consultar un tema específico.

RF4. Reproducir videos, sonidos, animaciones y diaporamas.

RF4.1: Poner pausa.

RF4.2: Reproducir.

RF4.3: Rebobinar.

RF4.4: Adelantar.

RF4.5: Regular el volumen: aumentar, disminuir o silenciar.

RF5. Ver imagen seleccionada.

RF6. Ampliar a pantalla completa la media visualizada.

RF7. Ver la fuente de donde proviene la media visualizada.

RF8. Permitir la navegabilidad entre las pantallas internas de sección a través de las opciones de “atrás”, “siguiente”, “primera” y “última”.

RF9. Permitir en las secciones Glosario y Personalidades, buscar un elemento a través de la selección de una de las letras mostradas en la pantalla, o introduciendo por el teclado el nombre del elemento deseado.

RF9.1: Mostrar el contenido (texto, imagen) del elemento seleccionado por el usuario en una nueva ventana.

RF10. Mostrar el número de la página donde se encuentra el usuario y el total de páginas existentes en la sección o tema visible.

RF11. Mostrar el total de elementos y el elemento visualizado.

RF12. Permitir ir a la interfaz principal de la Mediateca desde cualquier pantalla donde se encuentre el usuario.

RF13. Gestionar videos.

RF13.1: Adicionar video.

RF13.2: Modificar datos del video.

RF13.3: Eliminar video.

RF14. Gestionar imágenes.

RF14.1: Adicionar imagen.

RF14.2: Modificar datos de la imagen.

RF14.3: Eliminar imagen.

RF15. Gestionar sonidos.

RF15.1: Adicionar sonido.

RF15.2: Modificar datos del sonido.

RF15.3: Eliminar sonido.

RF16. Gestionar personalidades.

RF16.1: Adicionar personalidad.

RF16.2: Modificar datos de la personalidad.

RF16.3: Eliminar personalidad.

RF17. Gestionar diaporamas.

RF17.1: Adicionar diaporama.

RF17.2: Modificar datos del diaporama.

RF17.3: Eliminar diaporama.

RF18. Gestionar animaciones.

RF18.1: Adicionar animación.

RF18.2: Modificar datos de la animación.

RF18.3: Eliminar datos de la animación.

RF19. Gestionar glosario.

RF19.1: Adicionar término al glosario.

RF19.2: Modificar término del glosario.

RF19.3: Eliminar término del glosario.

RF20. Registrar la traza de los estudiantes en la Galería de imágenes.

RF21. Registrar la traza de los estudiantes en la Galería de videos.

RF22. Registrar la traza de los estudiantes en el Glosario de términos.

Listado de los Requisitos no Funcionales (RNF)

RNF1. Requisitos de software.

- ✓ Computadora Personal con navegador Mozilla Firefox 3.5 o superior.

RNF2. Requisitos de hardware.

- ✓ Procesador Pentium 500 MHz o superior.
- ✓ Al menos 256 MB de RAM.
- ✓ Al menos 10 GB de espacio en disco duro.
- ✓ Dispositivos de audio.
- ✓ Soporte de video que admita resolución de al menos 800x600 y 24 bits.
- ✓ Dispositivo de red de al menos 10 Mbits.
- ✓ El sistema interactuará con una impresora que permita imprimir los diferentes contenidos como respuesta a las funcionalidades del sistema.

RNF3. Requisitos de restricciones en el diseño y en la implementación.

- ✓ Framework del lado del cliente: jQuery 1.4.
- ✓ Framework del lado del servidor: Symphony 1.4.3.

- ✓ Lenguaje de programación del lado del cliente: JavaScript.
- ✓ Lenguaje de programación del lado del servidor: PHP 5.2.
- ✓ Servidor web: Apache 2.0.
- ✓ IDE de desarrollo: NetBeans 6.9.
- ✓ Servidor de base de datos: PostgreSQL 8.4.
- ✓ Patrón arquitectónico: Modelo Vista Controlador.

RNF4. Requisitos de apariencia e interfaz externa.

- ✓ Las interfaces se desarrollarán teniendo en cuenta elementos de las aplicaciones con tecnología multimedia diseñadas para escritorio y las aplicaciones web.
- ✓ Las interfaces de usuario estarán acorde a las edades a las que van dirigidos cada uno de los 10 productos que integran la colección El Navegante.
- ✓ El sistema proporcionará claridad y correcta organización de la información, permitiendo la interpretación correcta e inequívoca de esta.
- ✓ El diseño de la interfaz gráfica deberá garantizar la distinción visual entre los elementos del sistema.

RNF5. Requisitos de soporte.

- ✓ Se realizará transferencia tecnológica de la colección a los clientes.
- ✓ Se impartirán clases a los profesores venezolanos para explicar el funcionamiento y utilidad del producto.

RNF6. Requisitos de portabilidad.

- ✓ El sistema podrá ser utilizado bajo cualquier sistema operativo que soporte Mozilla Firefox.

RNF7. Requisitos de usabilidad.

- ✓ Facilidad de desarrollo con software libre debido a la amplia gama de herramientas disponibles para el desarrollo de aplicaciones web.
- ✓ La utilización del software por parte de los estudiantes de los liceos u otros usuarios sin avanzados conocimientos de computación, no requiere de previa preparación, debido al diseño sencillo y estándar que tiene.

- ✓ Cada producto cuenta con una ayuda que se ubicará como un servicio en el mismo, cubriendo cada una de las pantallas o conceptos del software y será escrita en lenguaje español lo más sencilla y asequible posible para facilitar su entendimiento por parte de los usuarios.

RNF8. Requisitos Legales.

- ✓ Al desplegarse el producto, todas las medias que hayan sido utilizadas deben tener el permiso legal de sus autores y su aprobación para hacer uso de ellas. Además deben respetarse las licencias y condiciones legales bajo las que fueron adquiridas.

RNF9. Requisitos de rendimiento.

Uso local

- ✓ Tiempo de respuesta de una página: no mayor a 30 segundos.

Uso en la red

- ✓ Velocidad de respuesta máxima: 5 minutos.

2.5 MODELO DE CASOS DE USO DEL SISTEMA

El modelo de casos de uso describe las funcionalidades propuestas del nuevo sistema. Permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos. Es la integración del diagrama de casos de uso del sistema y la especificación de los casos de uso y actores relacionados en el mismo.

Un actor es una persona, sistema o dispositivo que interactúa con el sistema, iniciando, recibiendo los resultados o participando en alguna de las acciones de un caso de uso. Por lo general representa un rol. (22)

Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto. Los casos de uso están diseñados para cumplir los deseos del usuario cuando utiliza el sistema, por este motivo captura todos los requisitos funcionales del sistema. (23)

Definición de los actores

En la siguiente tabla se muestra la justificación de los actores que intervienen en el sistema:

Actores	Justificación
Administrador	Persona que se encarga de la adición, modificación y eliminación de los recursos multimediales.
Usuario	Persona que interactúa con el sistema como profesor, estudiante o invitado.

Diagrama de casos de uso del sistema

Como parte de la propuesta de sistema se define la realización de dos aplicaciones que integrarán el módulo Mediateca, una que permitirá la gestión de medias de la colección y otra que las visualizará. Se muestra a continuación los diagramas de casos de uso del sistema evidenciando cómo interactúan los actores con los casos de uso.



Figura 3: Diagrama de casos de uso del sistema de la aplicación de administración.

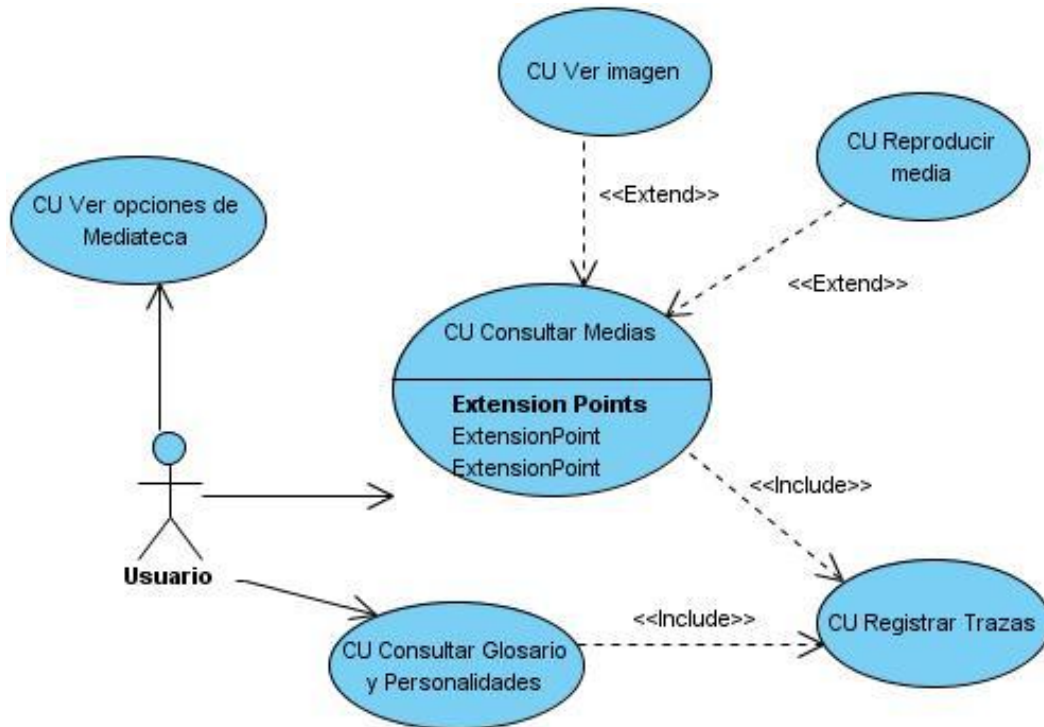


Figura 4: Diagrama de casos de uso del sistema de la aplicación de visualización.

2.6 PATRONES DE CASOS DE USO

Los patrones de casos de uso son utilizados generalmente como plantillas que describen cómo deberían ser estructurados y organizados los casos de uso. (24)

Para el modelado de la aplicación se utilizaron los siguientes:

Patrón CRUD: Usado en los casos donde se quiere realizar altas, bajas, cambios y consultas a alguna entidad del sistema. Su nombre es un acrónimo de las palabras en inglés *Create*, *Read*, *Update* y *Delete*. La variante “CRUD Completo” consiste en un único caso de uso para administrar la información de una entidad del sistema. Se evidencia en los casos de uso de gestión de medias.

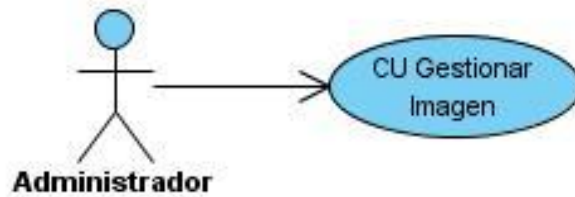


Figura 5: CRUD Completo

Patrón Concordancia: Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado. Reuso es la variante de este patrón que se evidencia en el modelado. Consta de 3 casos de uso. El primero llamado subsecuencia común, modela una secuencia de acciones que aparecerán en múltiples casos de uso en el modelo. Los otros casos de uso comparten la subsecuencia común de acciones, de manera que deben existir al menos dos de ellos. A continuación se muestra un ejemplo de su utilización:

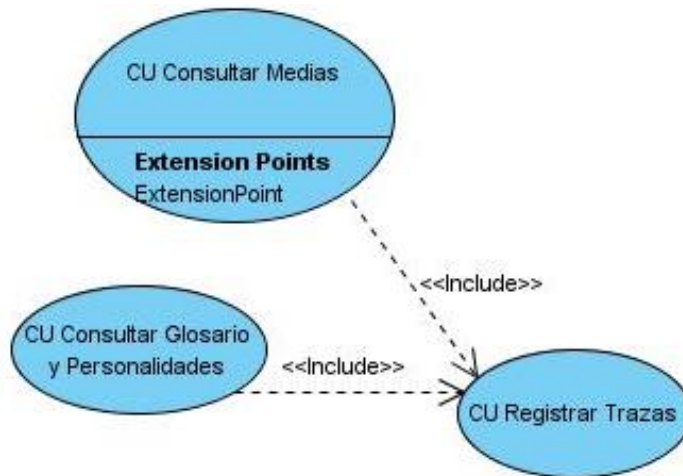


Figura 6: Concordancia (Modelo de Reuso)

2.7 DESCRIPCIÓN DE CASOS DE USO DEL SISTEMA

Mediante las descripciones de los casos de uso se detalla paso a paso la secuencia de eventos que los actores realizan para completar un proceso a través del sistema. A continuación se muestra una descripción de forma resumida de los casos de uso presentes en el Diagrama de casos de uso del sistema. Se situará el propósito general de cada caso de uso, el actor que da inicio al mismo y un pequeño resumen de su funcionamiento. En el **¡Error! No se encuentra el origen de la referencia.** se encuentran las descripciones de todos los casos de uso (CU) de forma ampliada.

CU-1	Ver opciones de Mediateca
Actor	Usuario
Propósito	Ver las opciones de consulta que permite el módulo Mediateca.
Resumen	El caso de uso inicia cuando el actor accede al módulo Mediateca. El sistema muestra las imágenes que representan a cada una de las secciones en las que se divide la Mediateca, y finaliza así el caso de uso.

Tabla 1: Descripción resumida del CU Ver opciones de Mediateca.

CU-2	Consultar glosario y personalidades
Actor	Usuario
Propósito	Buscar y mostrar una breve descripción de las personalidades o de los términos disponibles en el glosario relacionados con los temas del software.
Resumen	El caso de uso se inicia cuando el actor accede a la sección Glosario o a la sección Personalidades y especifica el identificador que desea consultar. El

	sistema muestra una breve descripción del identificador consultado y finaliza así el caso de uso.
--	---

Tabla 2: Descripción resumida del CU Consultar glosario y personalidades.

CU-3	Consultar medias
Actor	Usuario
Propósito	Listar las medias por temas disponibles en el software y permitir la consulta de los mismos.
Resumen	El caso de uso inicia cuando el actor accede a una de las Galerías de medias disponibles en el módulo Mediateca. El sistema muestra de forma organizada los temas que contienen esas medias. El usuario selecciona una de estos temas y se muestra el listado de medias disponibles en la misma finalizando así el caso de uso.

Tabla 3: Descripción resumida del CU Consultar medias.

CU-4	Ver imagen
Actor	Usuario
Propósito	Mostrar la imagen seleccionada por el usuario.
Resumen	El caso de uso inicia cuando el usuario selecciona la opción que le permite ver una imagen. El sistema muestra la imagen correspondiente finalizando así el caso de uso.

Tabla 4: Descripción resumida del CU Ver imagen.

CU-5	Reproducir media
Actor	Usuario
Propósito	Reproducir una media.
Resumen	El caso de uso se inicia cuando el actor selecciona una media o un artículo con media asociada. El sistema reproduce la media finalizando así el caso de uso.

Tabla 5: Descripción resumida del CU Reproducir media.

CU-6	Registrar trazas de Mediateca
Actor	Usuario
Propósito	Registrar los datos del resultado de la interacción de los estudiantes con el módulo Mediateca.
Resumen	El caso de uso inicia cuando el estudiante accede a una de las secciones del módulo Mediateca. El sistema registra los datos de la interacción del estudiante con la sección seleccionada finalizando así el caso de uso.

Tabla 6: Descripción resumida del CU Registrar trazas de Mediateca.

CU-7	Gestionar video
Actor	Administrador
Propósito	Adicionar, modificar y eliminar video.
Resumen	El caso de uso se inicia cuando el administrador selecciona Videos. El sistema muestra el listado de videos existentes y permite adicionar un nuevo video, modificar los datos del video seleccionado y eliminarlo. El administrador selecciona una de estas opciones y finaliza el caso de uso.

Tabla 7: Descripción resumida del CU Gestionar video.

CU-8	Gestionar imagen
Actor	Administrador
Propósito	Adicionar, modificar y eliminar imagen.
Resumen	El caso de uso se inicia cuando el administrador selecciona Imágenes. El sistema muestra el listado de las imágenes existentes y permite adicionar una nueva imagen, modificar los datos de la imagen seleccionada y eliminarla. El administrador selecciona una de estas opciones y finaliza el caso de uso.

Tabla 8: Descripción resumida del CU Gestionar imagen.

CU-9	Gestionar sonido
Actor	Administrador
Propósito	Adicionar, modificar y eliminar sonido.
Resumen	El caso de uso se inicia cuando el administrador selecciona Sonidos. El sistema muestra el listado de sonidos existentes y permite adicionar un nuevo sonido, modificar los datos del sonido seleccionado y eliminarlo. El administrador selecciona una de estas opciones y finaliza el caso de uso.

Tabla 9: Descripción resumida del CU Gestionar sonido.

CU-10	Gestionar personalidad
Actor	Administrador
Propósito	Adicionar, modificar y eliminar Personalidad.
Resumen	El caso de uso se inicia cuando el administrador selecciona Personalidades. El sistema muestra el listado de las personalidades existentes y permite adicionar una nueva personalidad, modificar los datos de la personalidad seleccionada y eliminarla. El administrador selecciona una de estas opciones y finaliza el caso de uso.

Tabla 10: Descripción resumida del CU Gestionar personalidad.

CU-11	Gestionar diaporama
Actor	Administrador
Propósito	Adicionar, modificar y eliminar diaporama.
Resumen	El caso de uso se inicia cuando el administrador selecciona Diaporamas. El sistema muestra el listado de diaporamas existentes y permite adicionar un nuevo diaporama, modificar los datos del diaporama seleccionado y eliminarlo. El administrador selecciona una de estas opciones y finaliza el caso de uso.

Tabla 11: Descripción resumida del CU Gestionar diaporama.

CU-12	Gestionar animación
Actor	Administrador
Propósito	Adicionar, modificar y eliminar animación.
Resumen	El caso de uso se inicia cuando el administrador selecciona Animaciones. El sistema muestra el listado de las animaciones existentes y permite adicionar una nueva animación, modificar los datos de la animación seleccionada y eliminarla. El administrador selecciona una de estas opciones y finaliza el caso de uso.

Tabla 12: Descripción resumida del CU Gestionar animación.

CU-13	Gestionar glosario
Actor	Administrador
Propósito	Adicionar, modificar y eliminar término del glosario.
Resumen	El caso de uso se inicia cuando el administrador selecciona Glosario. El sistema muestra el listado de términos existentes en el glosario y permite adicionar un nuevo término al glosario, modificar los datos de un término seleccionado y eliminarlo. El administrador selecciona una de estas opciones y finaliza así el caso de uso.

Tabla 13: Descripción resumida del CU Gestionar glosario.

2.8 CONCLUSIONES DEL CAPÍTULO

En el capítulo desarrollado se argumentan los motivos de la realización del Modelo de dominio para lograr un entendimiento preciso de lo requerido. Se comenzó a profundizar en el desarrollo de la propuesta de solución, obteniéndose una lista de las funcionalidades que debe tener el sistema, las mismas fueron representadas mediante un diagrama de casos de uso, y finalmente fueron descritas todas las acciones que realizan los actores y el sistema en general. A partir de aquí se puede comenzar a construir el sistema, cumpliendo con todos los requerimientos y las funcionalidades que se consideraron en este capítulo.

CAPÍTULO 3

ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 INTRODUCCIÓN

A partir del diagrama de casos de uso, obtenido como parte del flujo de trabajo Requerimientos, se puede continuar con el flujo que continúa el ciclo de vida de RUP: Análisis y Diseño. De esta forma se obtienen los diagramas de clases de análisis correspondientes a cada caso de uso, refinándose este análisis a través de diagramas de clases del diseño, con la finalidad de adaptarlo al ambiente de implementación.

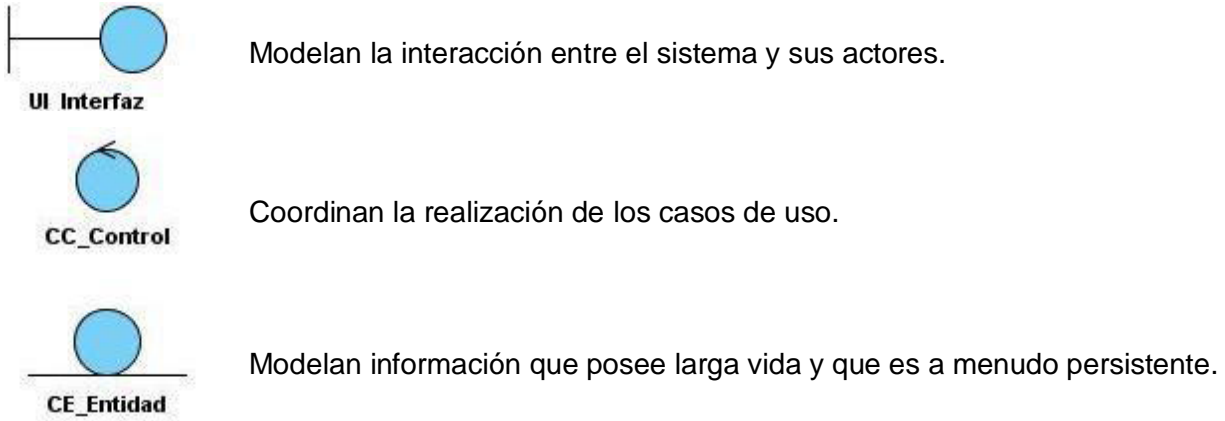
3.2 MODELO DE ANÁLISIS

El lenguaje que se utiliza en el análisis se basa en un modelo de objetos conceptual llamado modelo de análisis. Este ayuda a redefinir los requisitos y permite razonar sobre aspectos internos del sistema, además ofrece un mayor poder expresivo y formalización. Es una primera aproximación al diseño. (25)

Aunque en el modelo del análisis hay un refinamiento de los requisitos, no se toma en cuenta el lenguaje de programación a usar en la construcción, el objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución. Se centra en describir qué debe hacer el sistema y no en cómo lo va a hacer. (25)

Diagramas de clases del análisis

Un diagrama de clases del análisis constituye una vista estática de las clases que conforman el modelo del análisis y las asociaciones entre las mismas. En la construcción de los diagramas de clases del análisis se tienen que identificar las clases que describirán la realización de los casos de uso y las relaciones entre ellas. Se utilizan tres tipos de clases y los siguientes estereotipos son los que las identifican:



A continuación se muestran los diagramas de clases de análisis correspondiente a cada caso de uso del sistema:

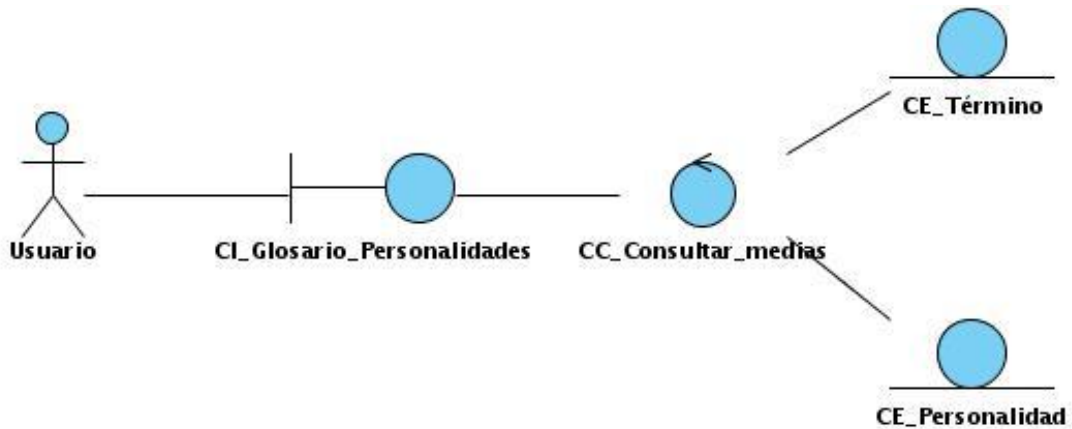


Figura 7: Diagrama de clases de análisis del CU Consultar glosario y personalidades.

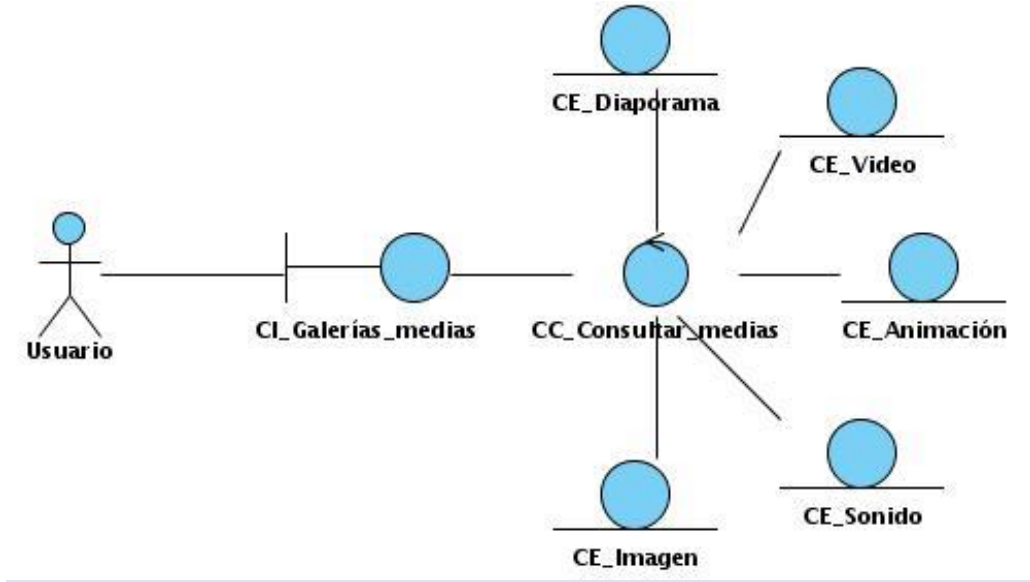


Figura 8: Diagrama de clases de análisis del CU Consultar medios.

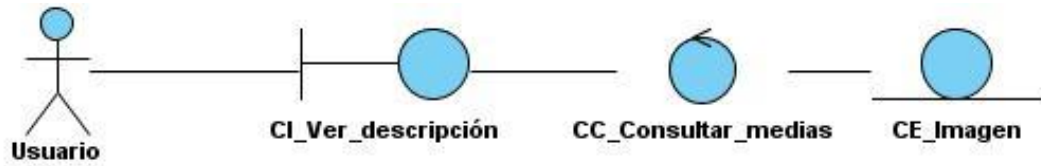


Figura 9: Diagrama de clases de análisis del CU Ver imagen.

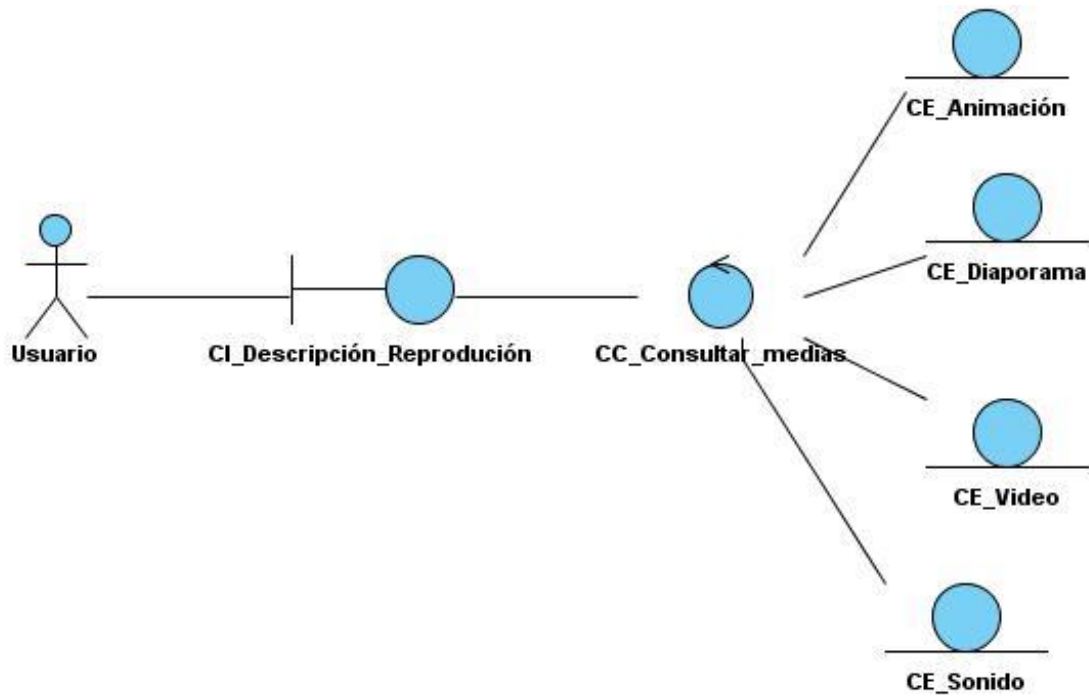


Figura 10: Diagrama de clases de análisis del CU Reproducir media.

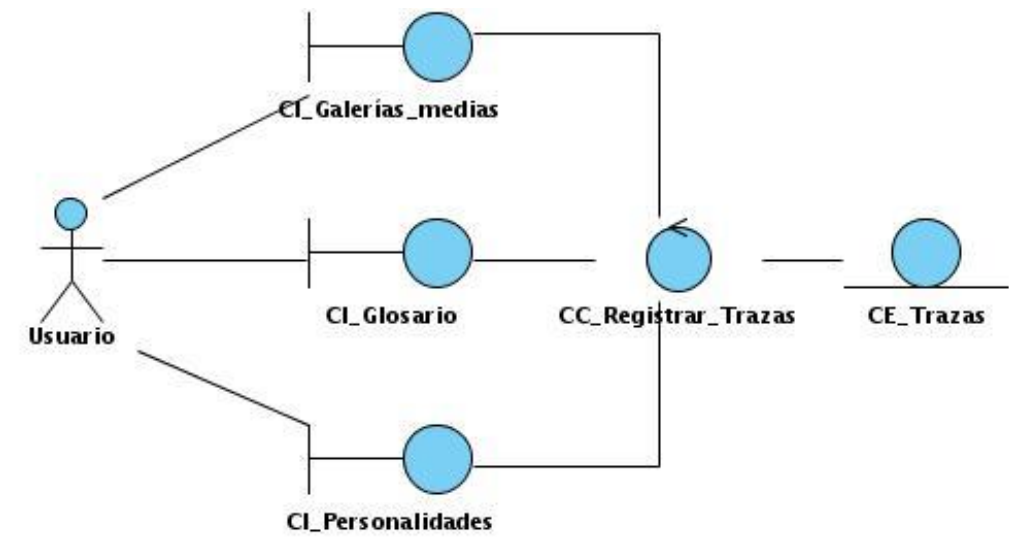


Figura 11: Diagrama de clases de análisis del CU Registrar trazas de Mediateca.

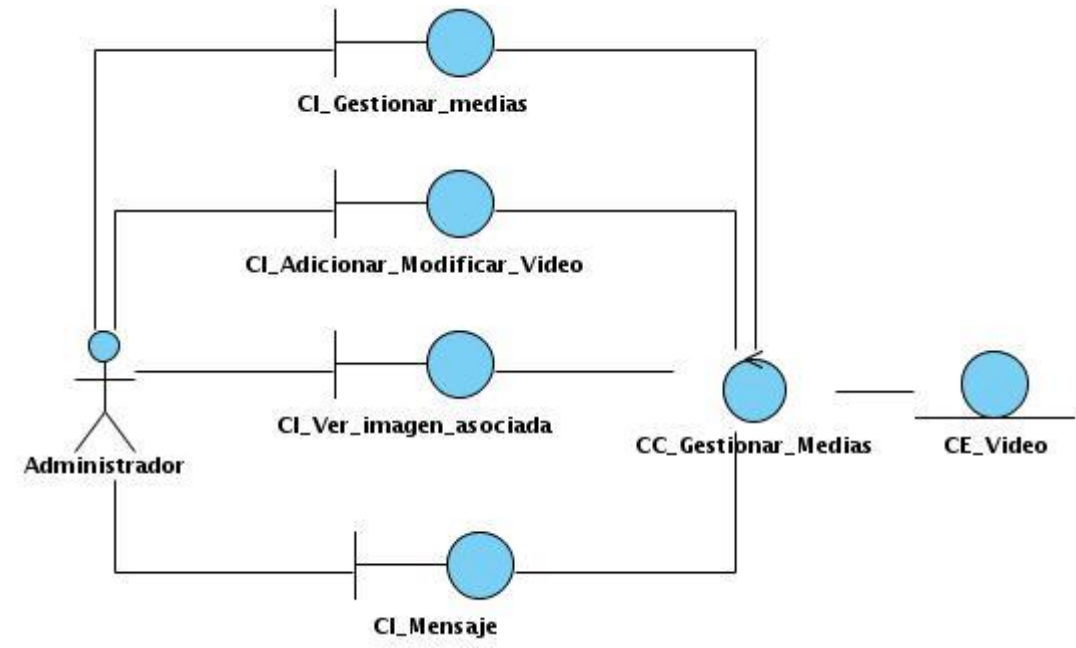


Figura 12: Diagrama de clases de análisis del CU Gestionar video.

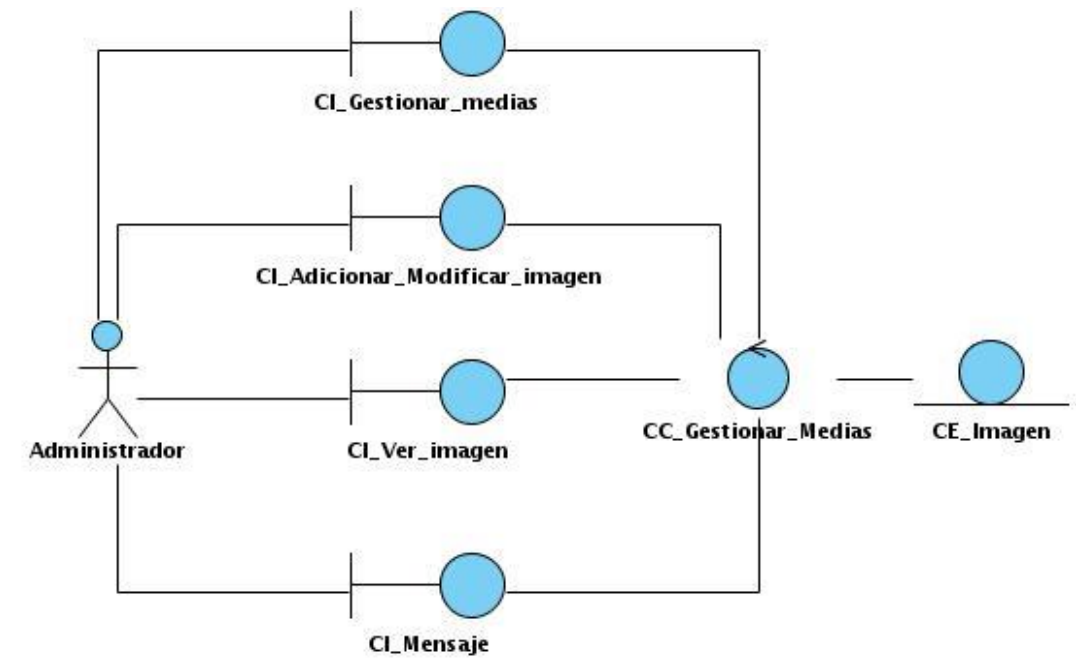


Figura 13: Diagrama de clases de análisis del CU Gestionar imagen.

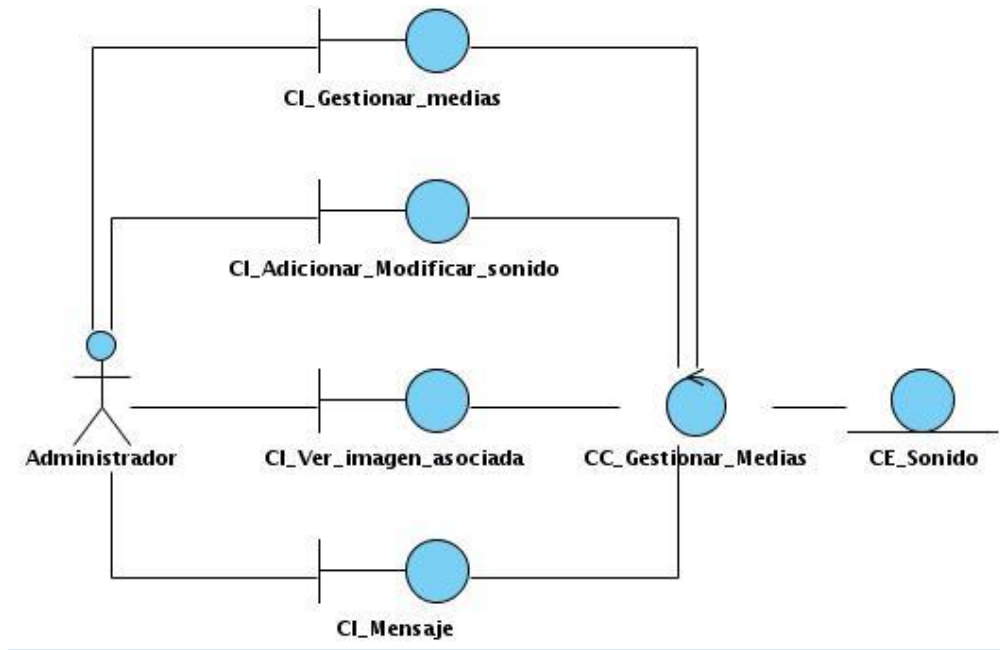


Figura 14: Diagrama de clases de análisis del CU Gestionar sonido.

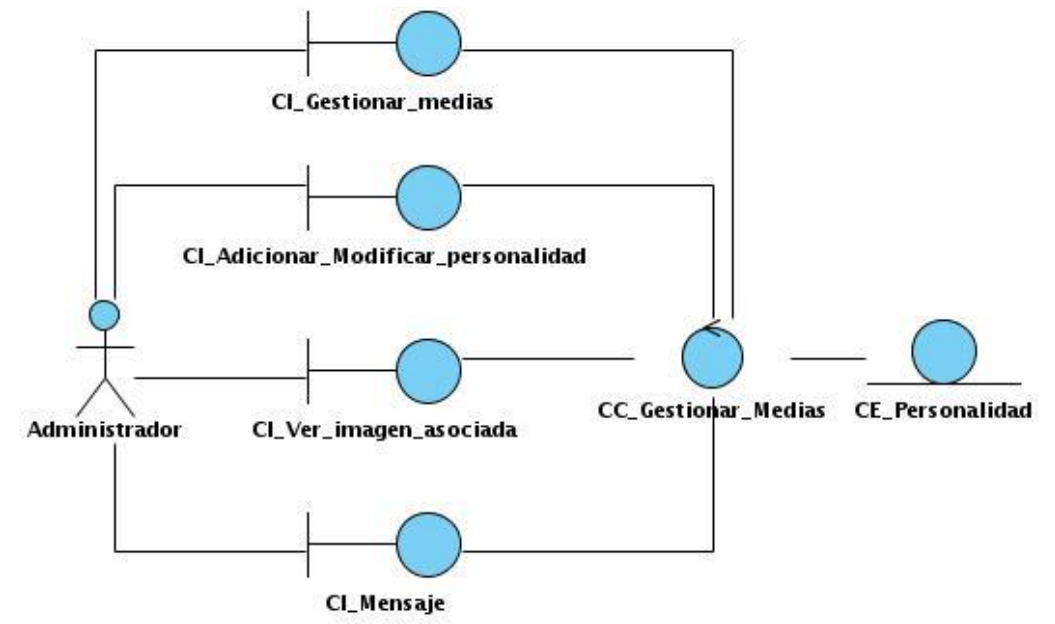


Figura 15: Diagrama de clases de análisis del CU Gestionar Personalidad.

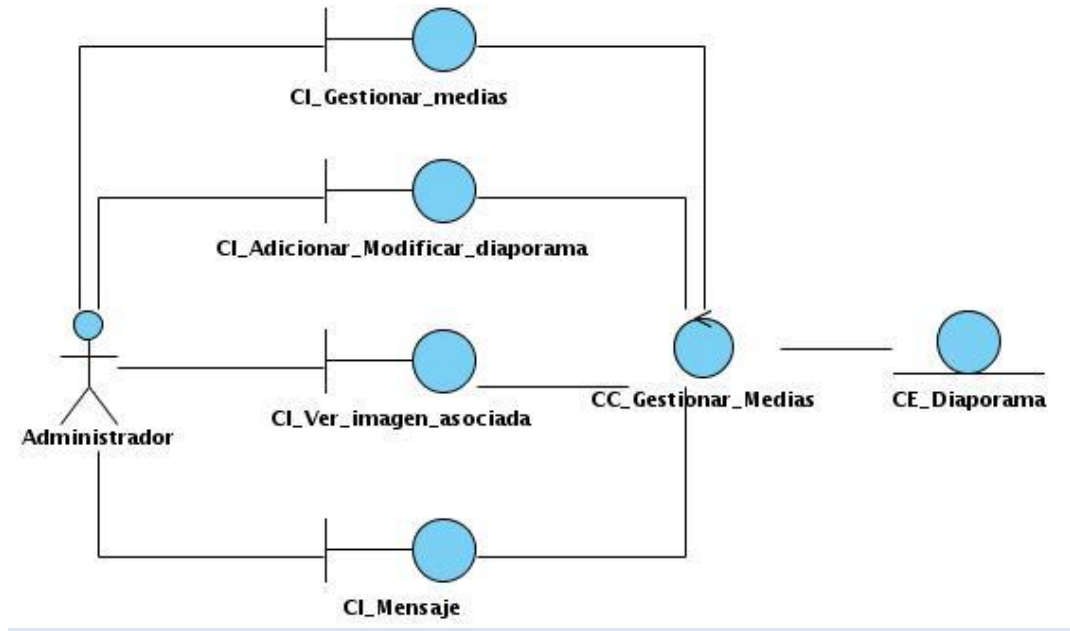


Figura 16 Diagrama de clases de análisis del CU Gestionar diaporama.

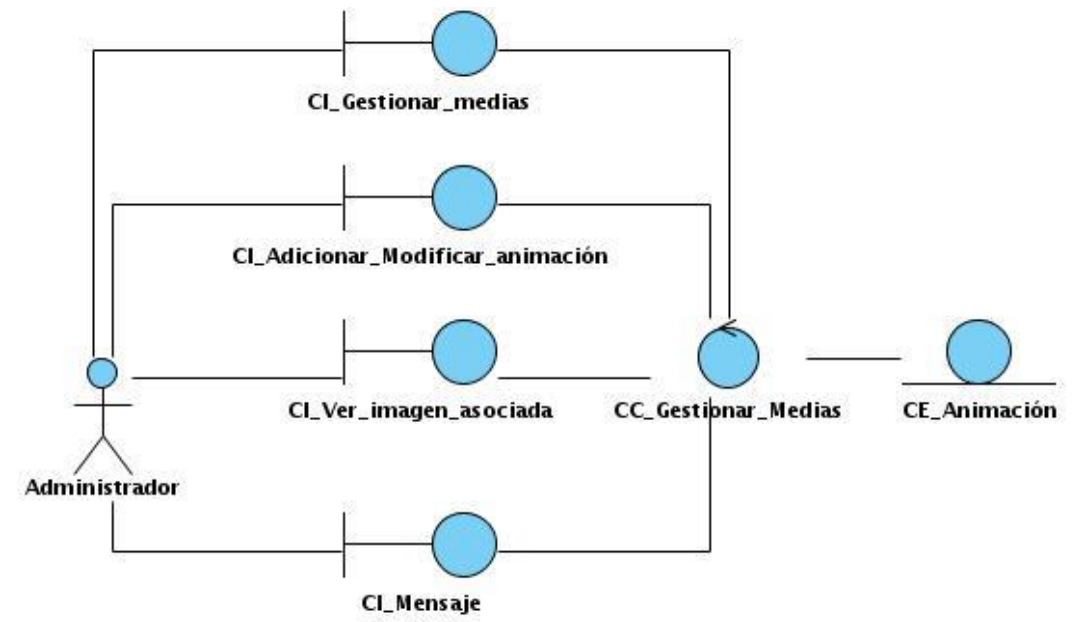


Figura 17: Diagrama de clases de análisis del CU Gestionar animación.

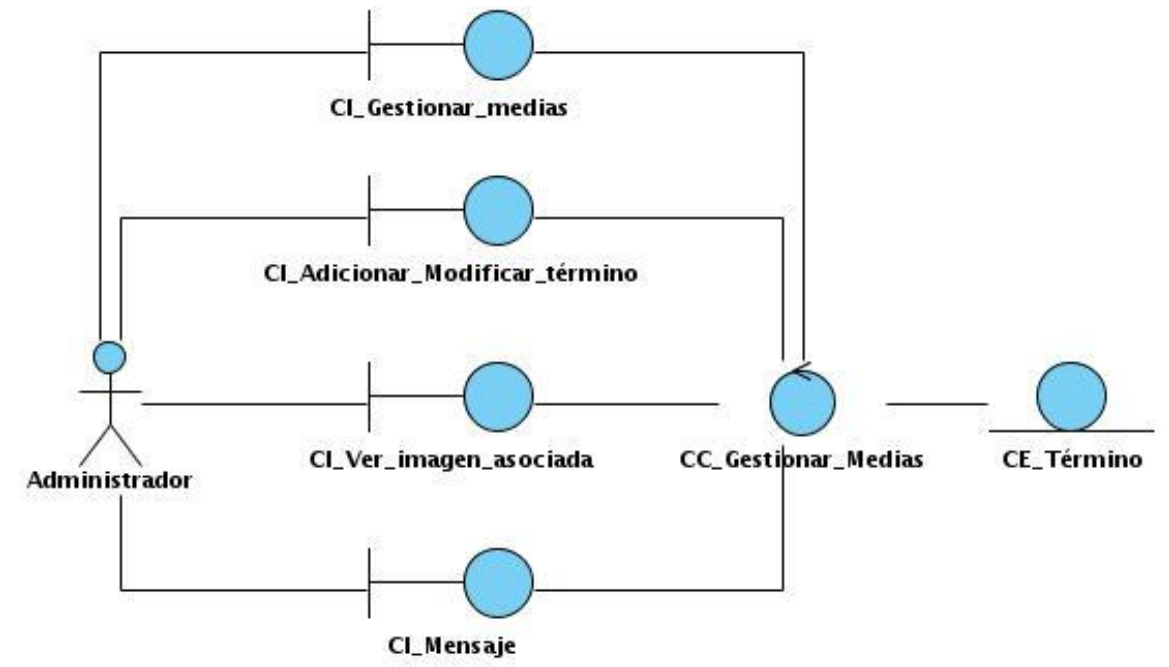


Figura 18: Diagrama de clases de análisis del CU Gestionar glosario.

3.3 MODELO DE DISEÑO

El diseño consiste en el refinamiento del modelo de análisis para crear especificaciones adicionales que lo enriquecen con detalles próximos a la implementación. (26)

El diseño define la estructura estática del sistema en forma de subsistemas, clases e interfaces, define los casos de uso reflejados como colaboraciones entre los subsistemas, clases e interfaces. Debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. Contribuye a una arquitectura estable y sólida y a crear un plano del modelo de implementación. (27)

En el presente epígrafe se plasman los resultados de la etapa de diseño del sistema.

Diagramas de clases del diseño

Unos de los artefactos más importantes a obtener en el flujo de diseño son los diagramas de clases de diseño, donde se exponen las clases que intervienen en las realizaciones de los casos de uso del sistema.

En este tipo de diagrama se representa un nivel de detalle más alto que en los diagramas de clases del análisis, relacionándose con el lenguaje de programación del cual se hará uso en la implementación del sistema.

Para modelar las aplicaciones web no se usa la forma tradicional de modelar las clases, se usan varios mecanismos de extensión que fueron definidos para extender su semántica en aquellas situaciones en las que no es posible capturar todas las características de una determinada arquitectura o dominio en particular. (28)

En el diagrama de clases para las aplicaciones web se modelan las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de estas, una vez que estén en el navegador del cliente; estos son los elementos que se necesitan para que el desarrollador los implemente.

En la siguiente tabla se muestran las terminologías utilizadas en los Diagramas de clases del diseño:




Clases	Estereotipos	Función
Server Page [SP] (en español Página Servidora)		Representa la página Web que tiene código que se ejecuta en el servidor. Su principal función es construir la página cliente.
Client Page [CP] (en español Página Cliente)		Representa una página Web, con formato HTML. Son interpretadas por el navegador, su función es visualizar, interactuar y mostrar lo que el usuario necesita.
Form (en español Formulario)		Colección de elementos de entrada que son parte de una página cliente. Sus atributos son los elementos de entrada del formulario. Su principal función es enviar los datos entrados por el usuario a la página servidora.

Tabla 14: Terminologías utilizadas en los diagramas de clases del diseño.

Mediateca estará compuesta por dos aplicaciones, el backend⁸ y el frontend⁹. Cada una cuenta con un único punto de entrada que es el controlador frontal, el cual recibe todas las peticiones y mediante él Symfony delega a las actions la acción a ejecutar. Las actions consultan el modelo de datos, llevan a cabo la lógica de negocio y preparan los datos; y Symfony muestra las vistas correspondientes a la acción. En el backend las páginas servidoras construyen las páginas clientes enviadas a los usuarios y en el frontend la vista se construye en las páginas clientes mediante peticiones asincrónicas al servidor. Las páginas clientes hacen link al controlador frontal, enviando como parámetro la acción a ejecutar.

Se usaron paquetes para distribuir y mostrar las responsabilidades de las clases en el patrón Modelo Vista Controlador. El subsistema “ComponentesSymfony” contiene todos los componentes que brinda Symfony para su uso y el subsistema “Doctrine” representa todos los componentes que utiliza el ORM Doctrine el cual es utilizado para el acceso a los datos.

Se realizaron los Diagramas de clases del diseño correspondientes a cada uno de los casos de uso del sistema, los atributos y métodos de las clases fueron ocultados para una mejor visualización de dichos diagramas.

⁸ Parte que procesa la entrada de datos que se efectuó desde el *front-end* es decir, son los procesos que utiliza el administrador del sitio con sus respectivos sistemas para resolver las peticiones de los usuarios.

⁹ El front-end en diseño de software y desarrollo web hace referencia a la parte que interactúa con los usuarios.

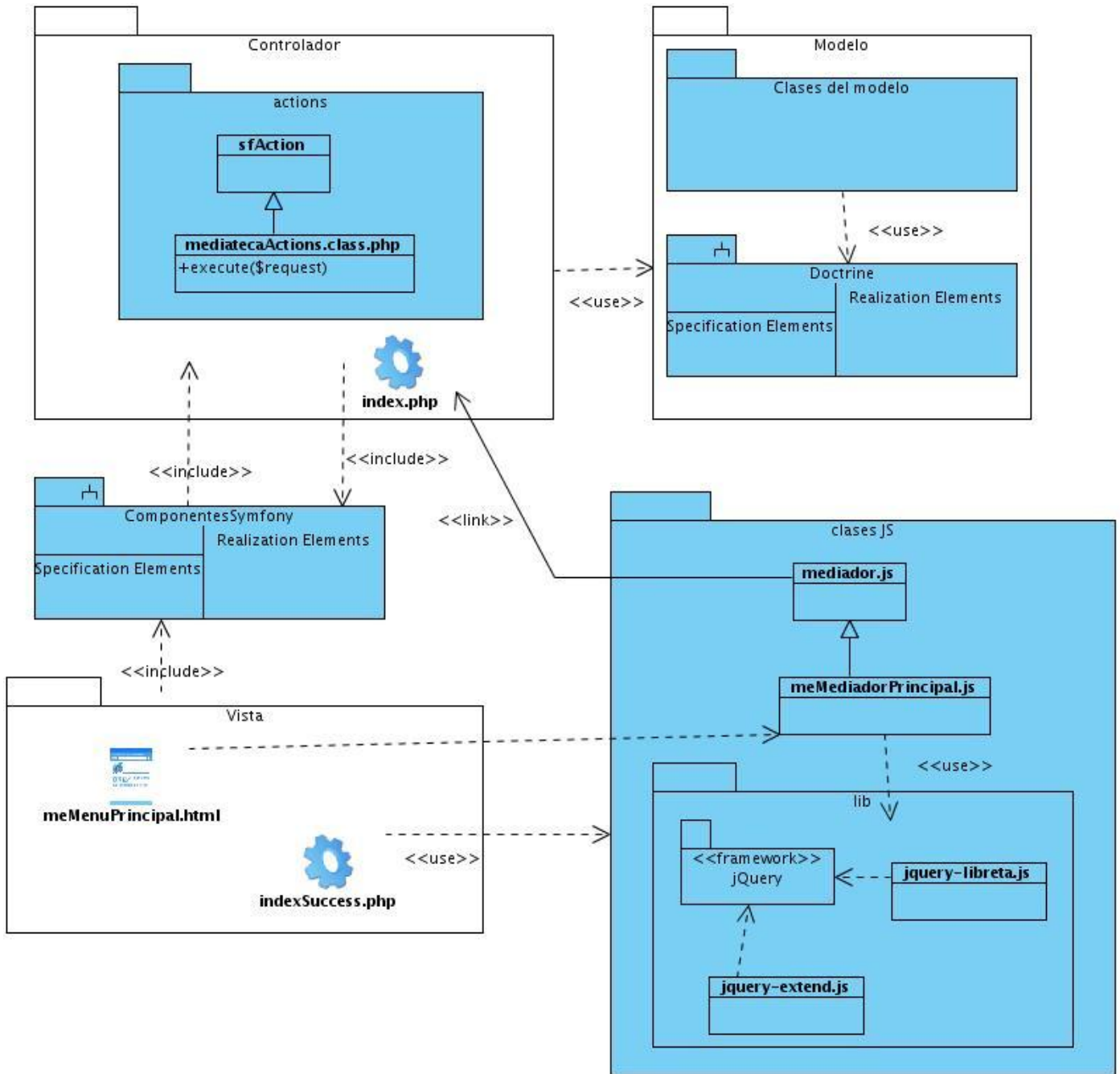


Figura 19: Diagrama de clases del diseño del CU Ver opciones de Mediateca.

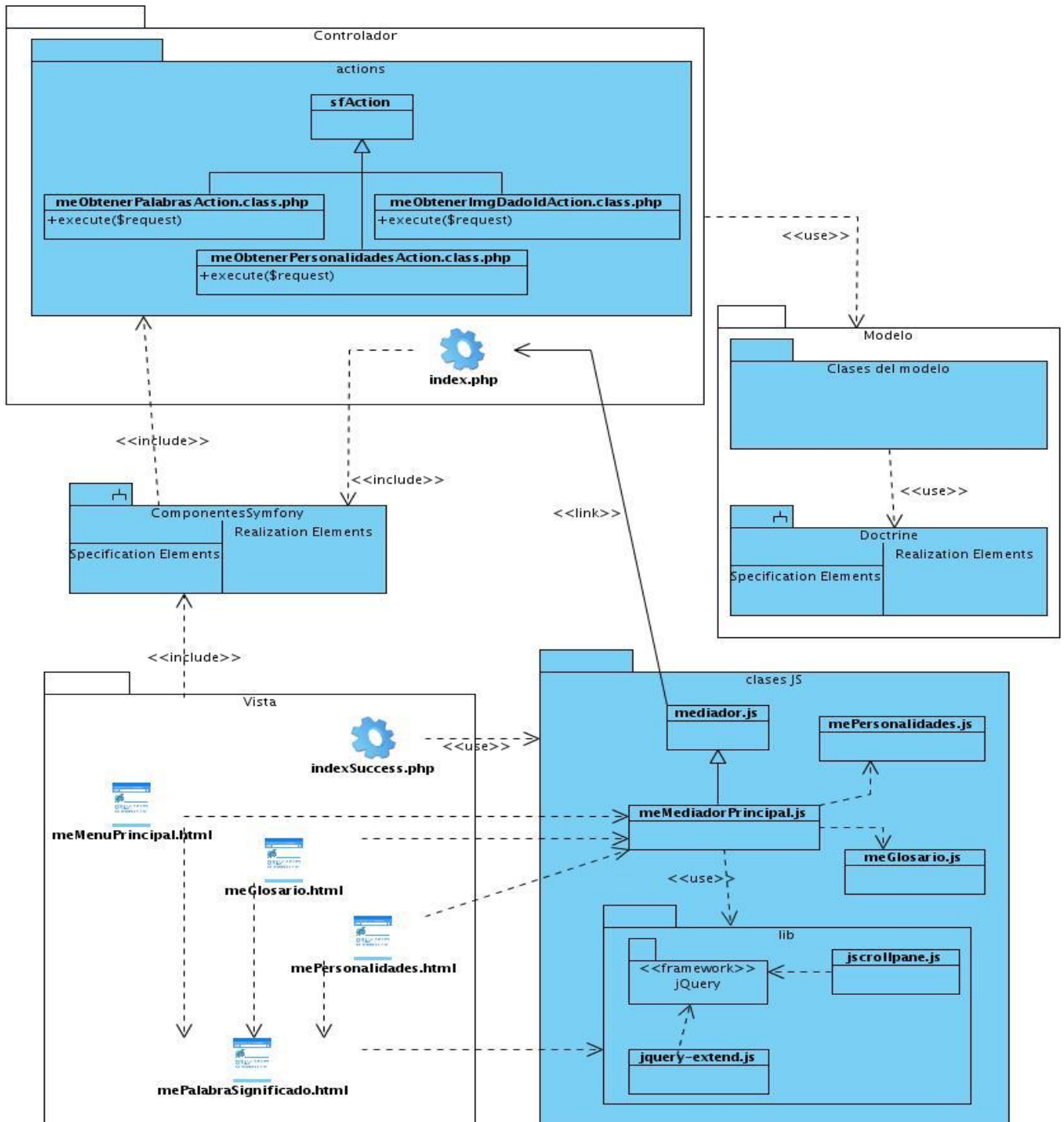


Figura 20: Diagrama de clases del diseño del CU Consultar glosario y personalidades.

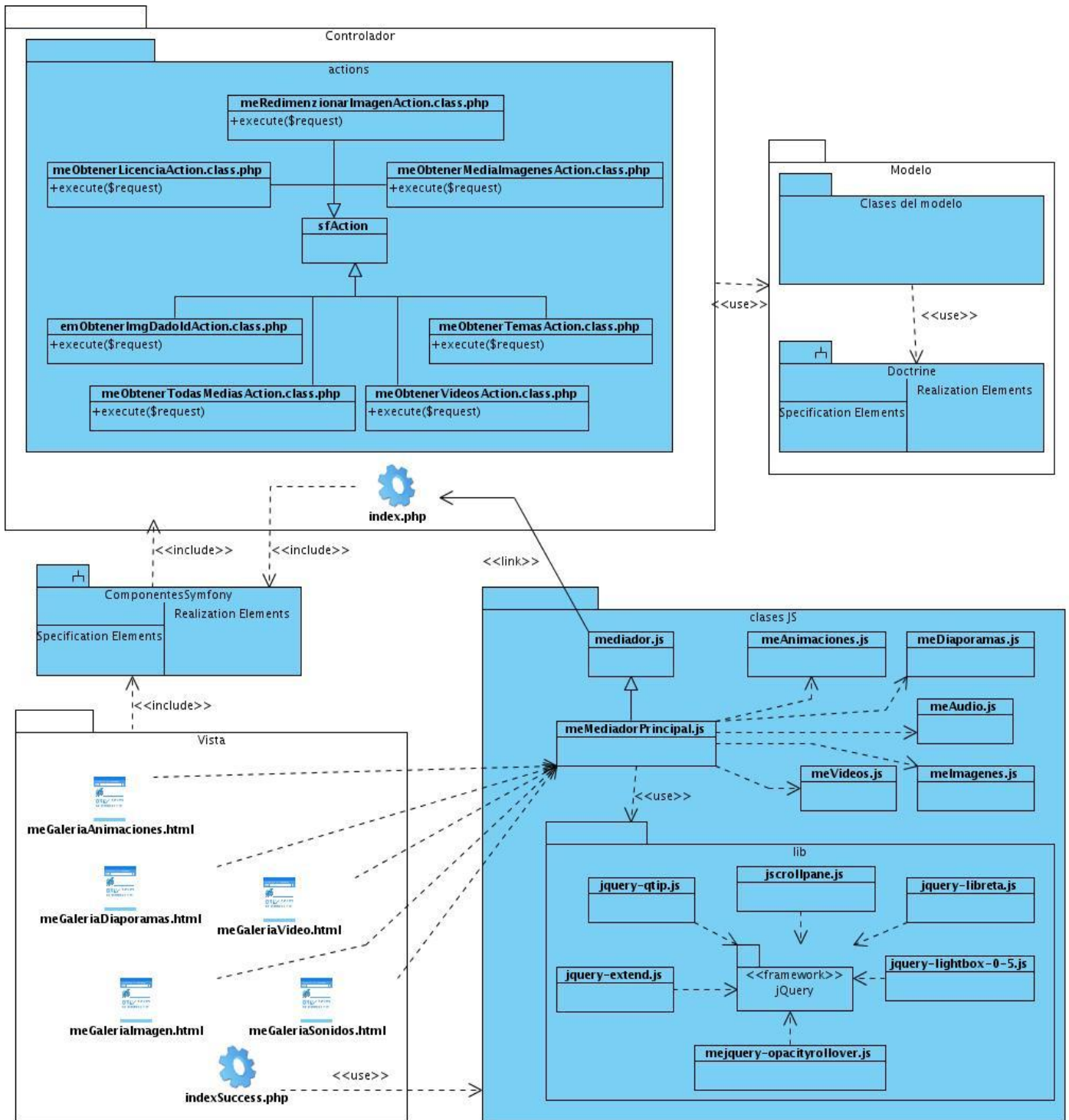


Figura 21: Diagrama de clases del diseño del CU Consultar medias.

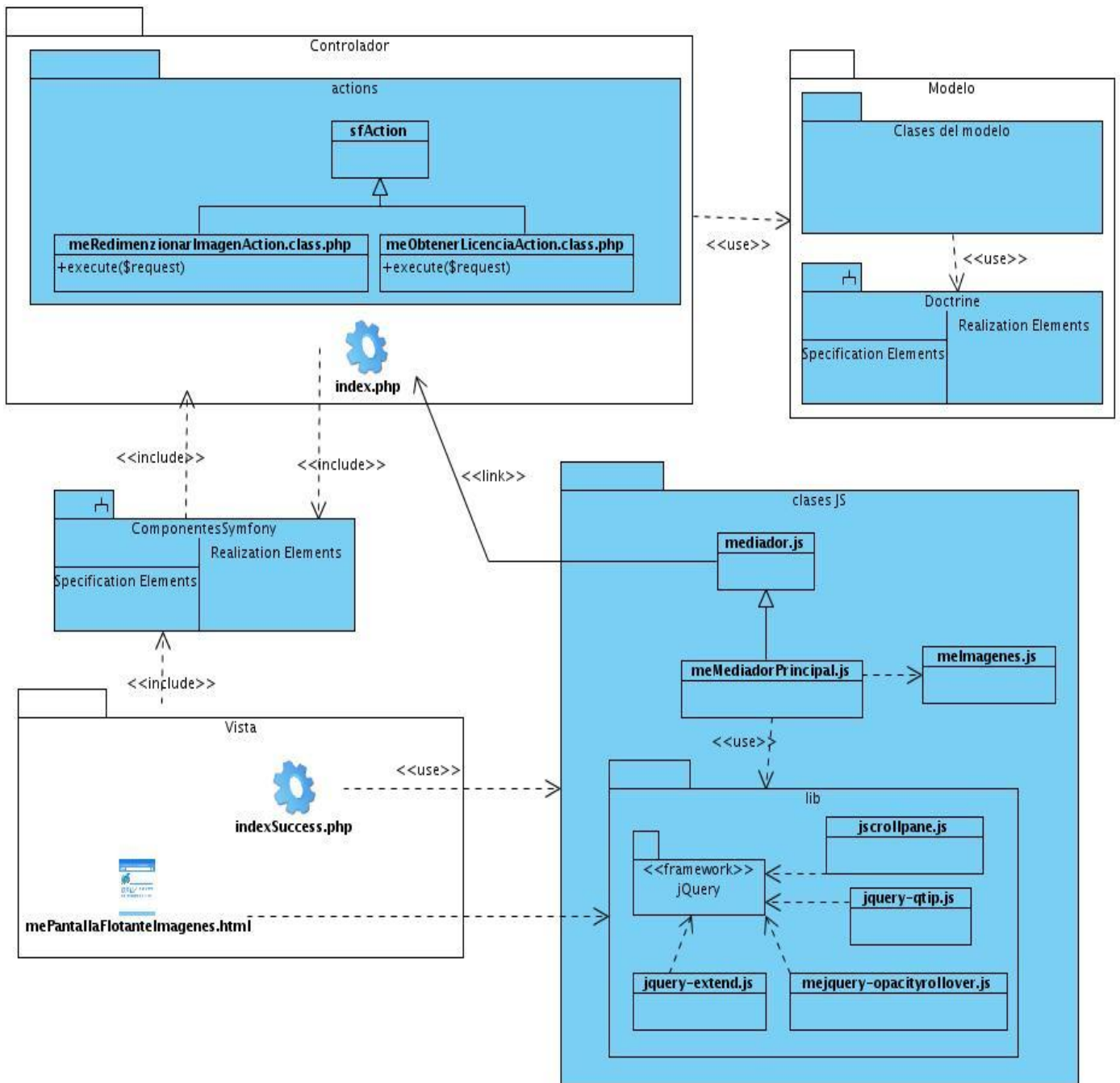


Figura 22: Diagrama de clases del diseño del CU Ver imagen.

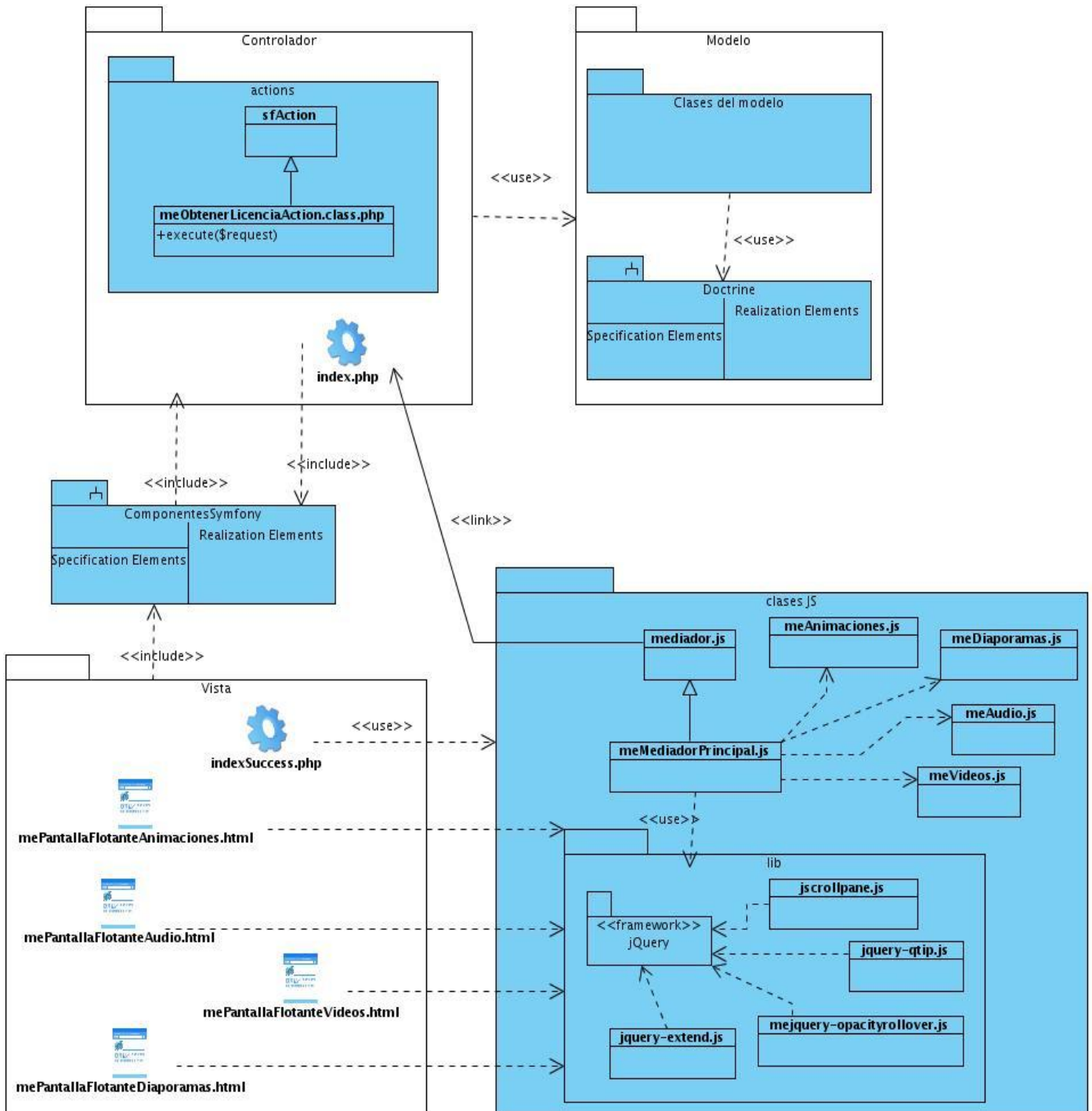


Figura 23: Diagrama de clases del diseño del CU Reproducir media.

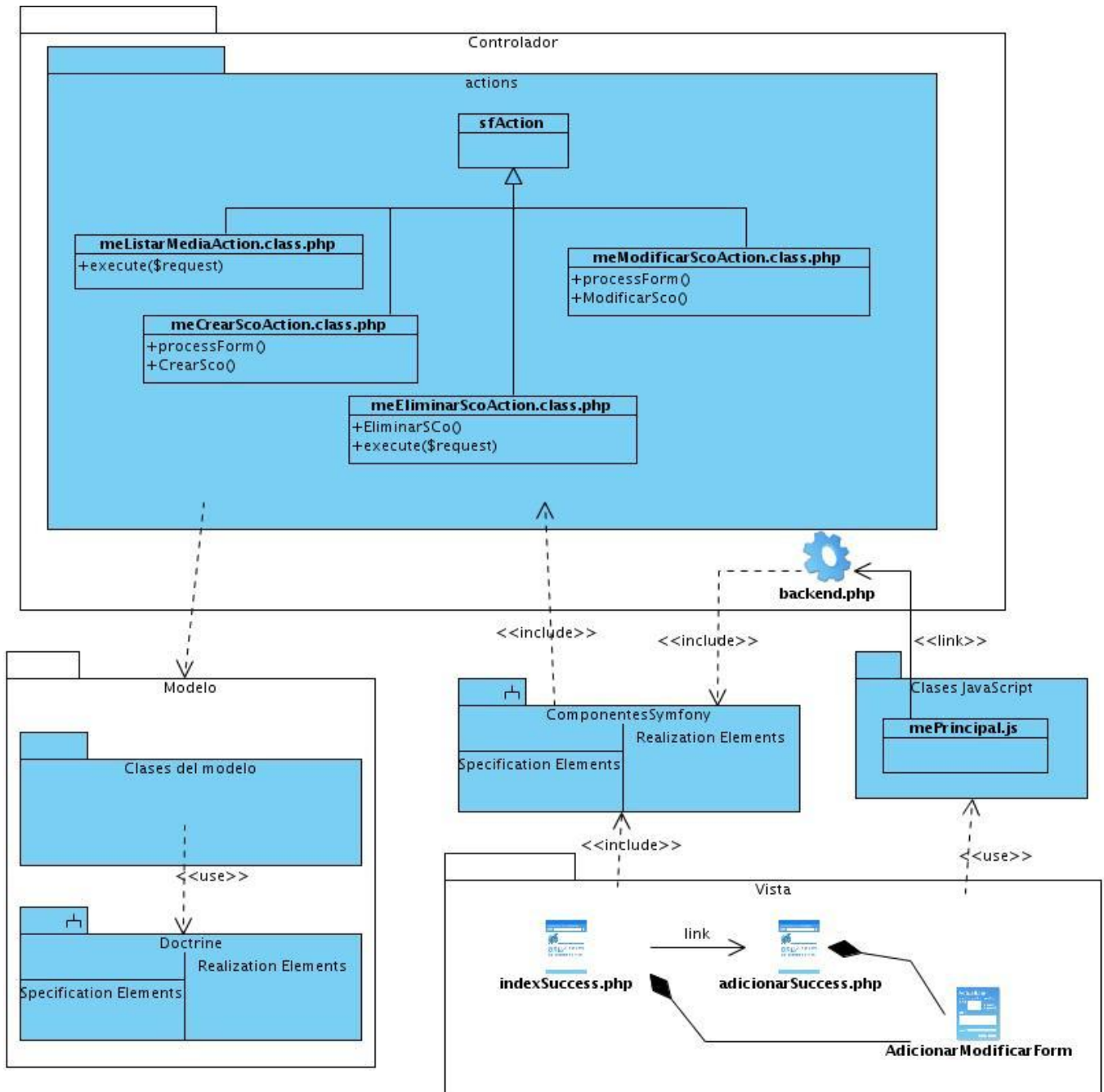


Figura 24: Diagrama de clases del diseño para los CU Gestionar animación, Gestionar video, Gestionar imagen, Gestionar diaporama y Gestionar sonido.

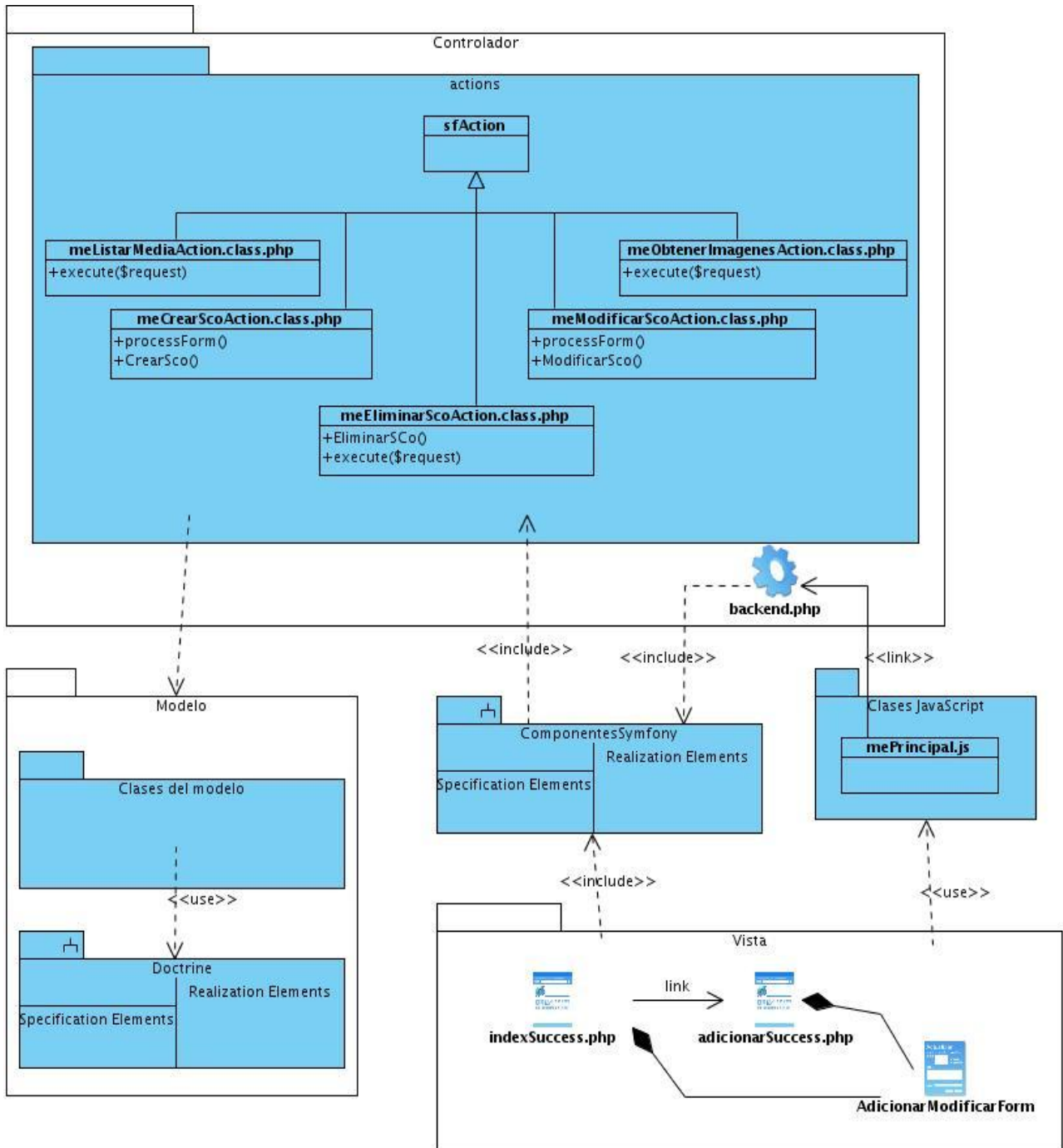


Figura 25: Diagrama de clases del diseño del CU Gestionar personalidad.

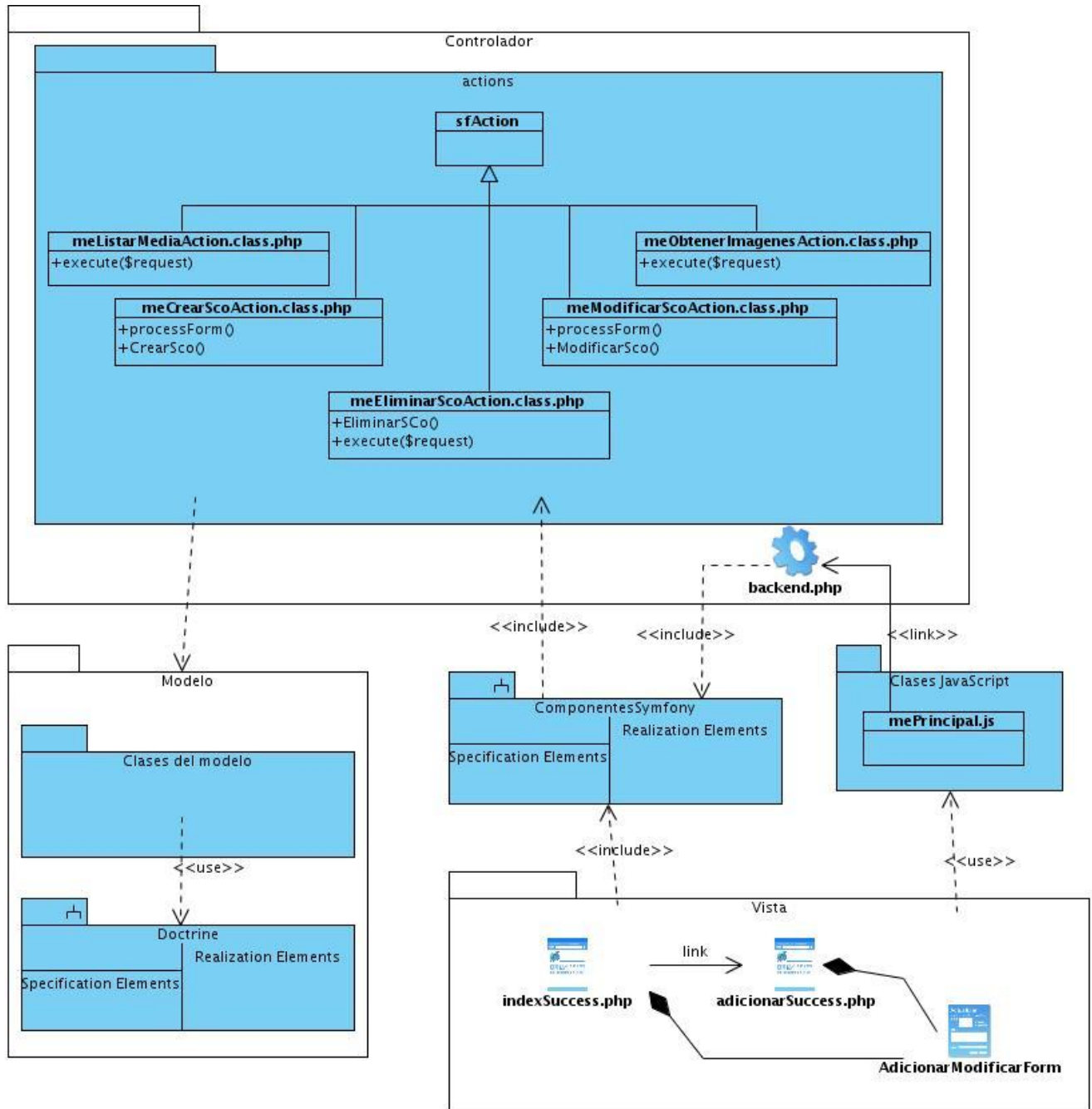


Figura 26: Diagrama de clases del diseño del CU Gestionar glosario.

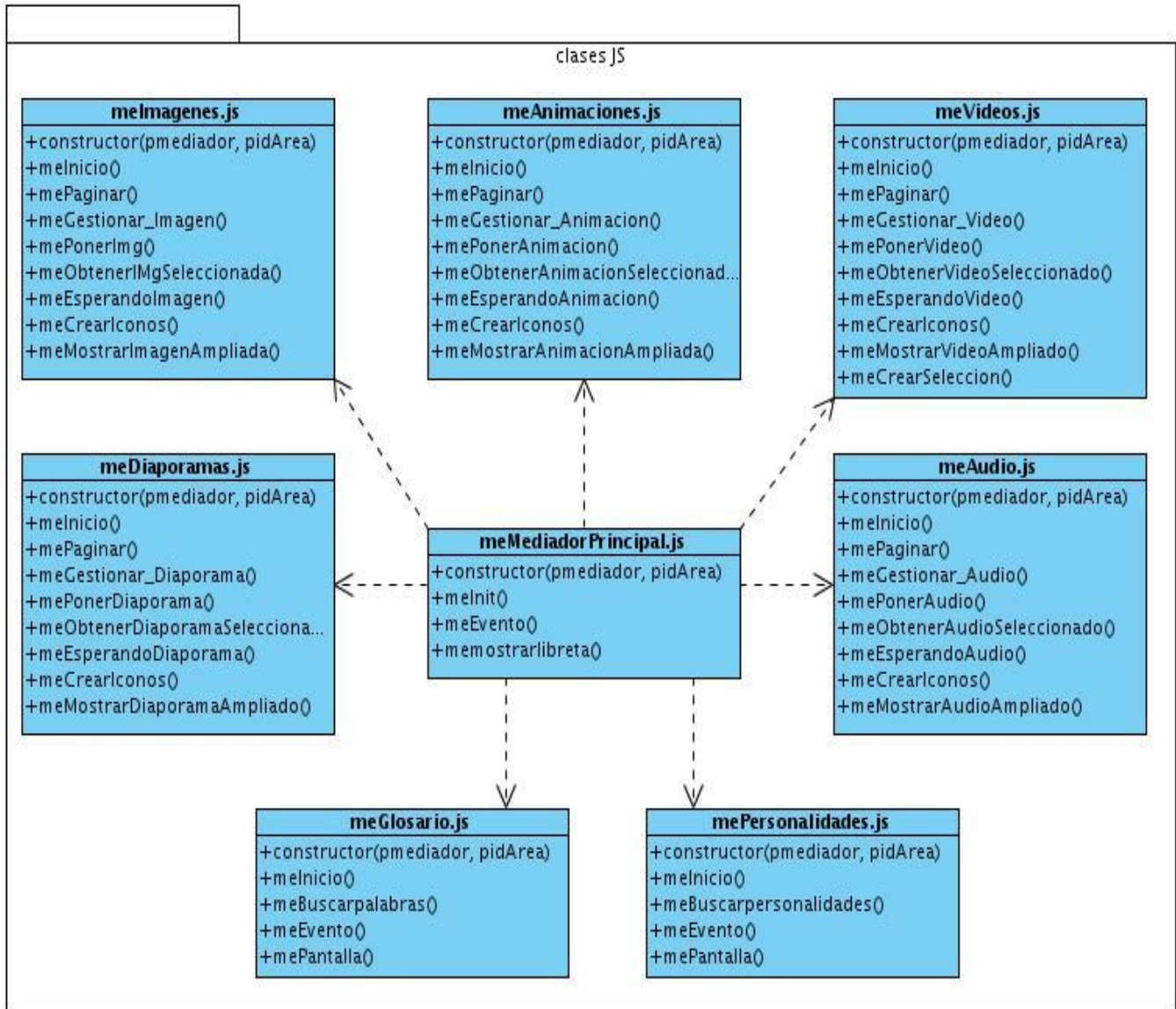


Figura 27: Diagrama de clases JavaScript del Frontend.

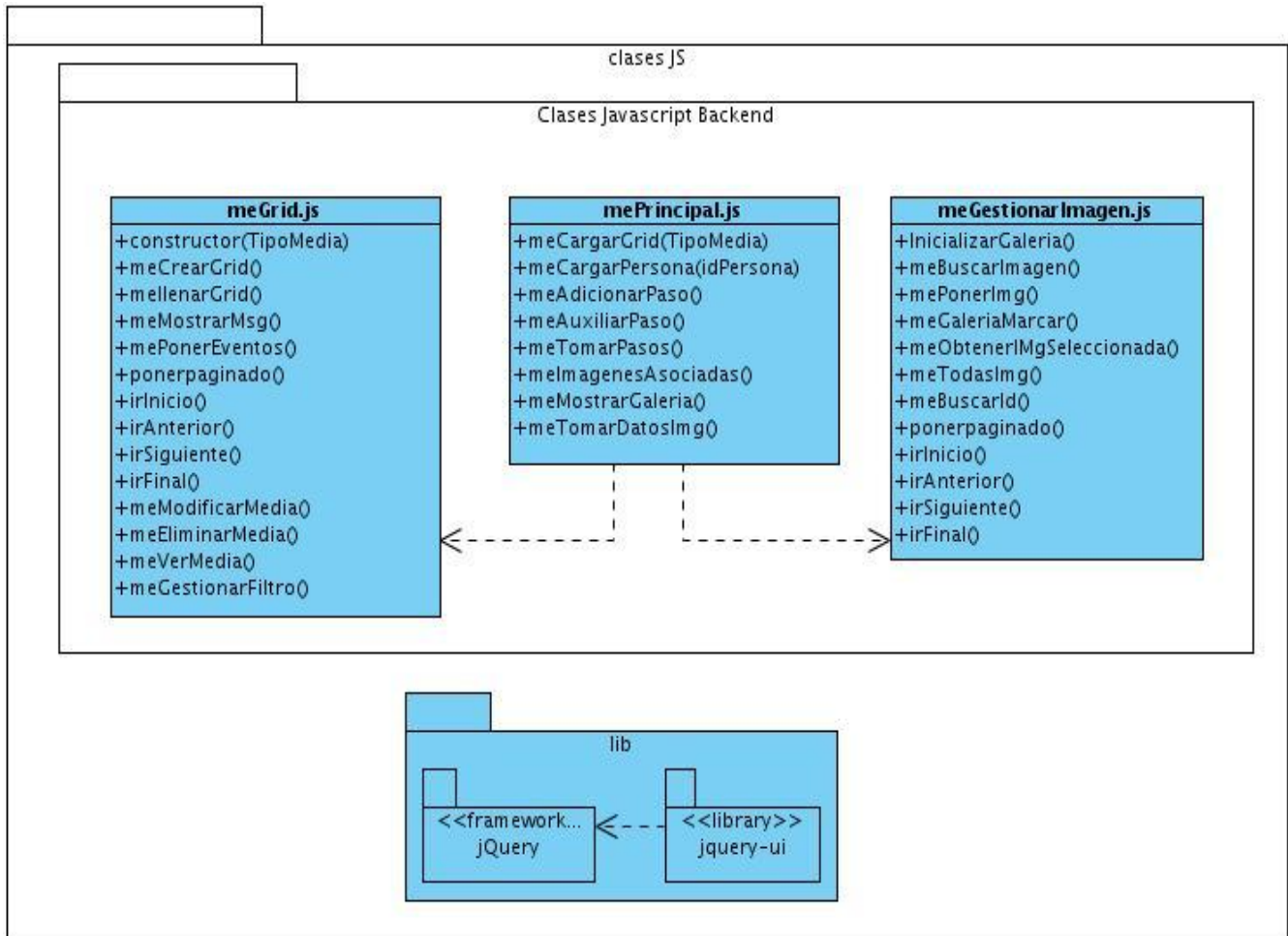


Figura 28: Diagrama de clases JavaScript del Backend.

3.4 PATRONES UTILIZADOS PARA EL DISEÑO

“Una arquitectura orientada a objetos bien estructurada está llena de patrones. La calidad de un sistema orientado a objetos se mide por la atención que los diseñadores han prestado a las colaboraciones entre sus objetos”. Grady Booch (29)

Identificar patrones y mecanismos de diseño en un mundo lleno de problemas a resolver es un paso significativo hacia una solución. Facilita una manera de aprovechar lo hecho y evitar que se cometan errores que comúnmente retrasan los proyectos. Un patrón de diseño es una solución a un problema de

diseño que es efectiva pues resolvió el problema satisfactoriamente en ocasiones anteriores y reusable porque se puede aplicar a diferentes problemas de diseño en distintas circunstancias.

Teniendo en cuenta que la aplicación que se desea crear es interactiva con una interfaz humano-computadora flexible se hizo uso del patrón de diseño **Modelo Vista Controlador**. Se definieron las clases dominio (Modelo) para que no tengan acoplamiento ni visibilidad directa respecto a las clases ventana (Vista) y para que los datos de la aplicación se conserven en las clases de dominio, no en las de ventana. Además se definieron las clases manejadores (Controlador) para que procesen las peticiones al sistema y redireccionen al Modelo y a la Vista tanto el procesamiento como la visualización de resultados respectivamente.

Con el MVC se logra la presencia de otros patrones como:

Bajo acoplamiento: Existe un mínimo de dependencia entre el modelo, la vista y el controlador (un cambio realizado en un componente no implica cambios en otro).

Alta cohesión: Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio).

Por otra parte cuando muchos objetos interactúan con otros, se puede formar una estructura demasiado compleja y en un caso extremo cada objeto puede conocer a todos los demás. Para evitar esto se utilizó el patrón **Mediator** (ver Figura 30) encapsulando el comportamiento de todo un conjunto de objetos en uno solo (meMediadorPrincipal.js), este mediador base conoce todos los objetos y cómo manipular su estado cuando un evento dado es reportado. De este modo se consigue que los objetos se puedan comunicar con otros sin tener que incluir referencias directas de código en sus clases propiciando el bajo acoplamiento.

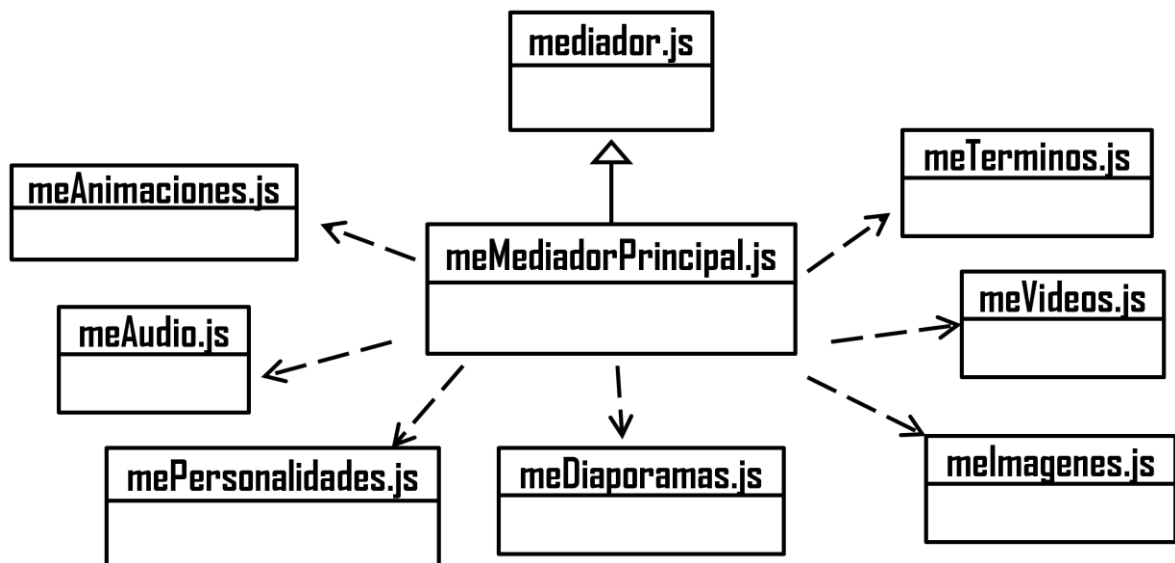


Figura 29: Uso del patrón Mediator.

3.5 MODELO DE DATOS

Un modelo de datos es una colección de herramientas conceptuales para describir los datos y las relaciones que existen entre ellos. Los modelos de datos se dividen en tres grupos:

- Modelos lógicos basados en objetos: Con este modelo se representan los datos de tal forma como se captan en el mundo real, tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente.
- Modelos lógicos basados en registro: A diferencia de los modelos de datos basados en objetos, se usan para especificar la estructura lógica global de la base de datos y para proporcionar una descripción a nivel más alto de la implementación.
- Modelos físicos de datos: Se usan para describir a los datos en el nivel más bajo, aunque existen muy pocos modelos de este tipo, básicamente capturan aspectos de la implementación de los sistemas de base de datos.

En la Figura 31 se muestra el modelo físico de datos correspondiente al módulo Mediateca obtenido como parte del trabajo del diseñador de base de datos de la colección. Para una mayor comprensión del mismo

remitirse al ¡Error! No se encuentra el origen de la referencia. donde se puede consultar la descripción de cada tabla.

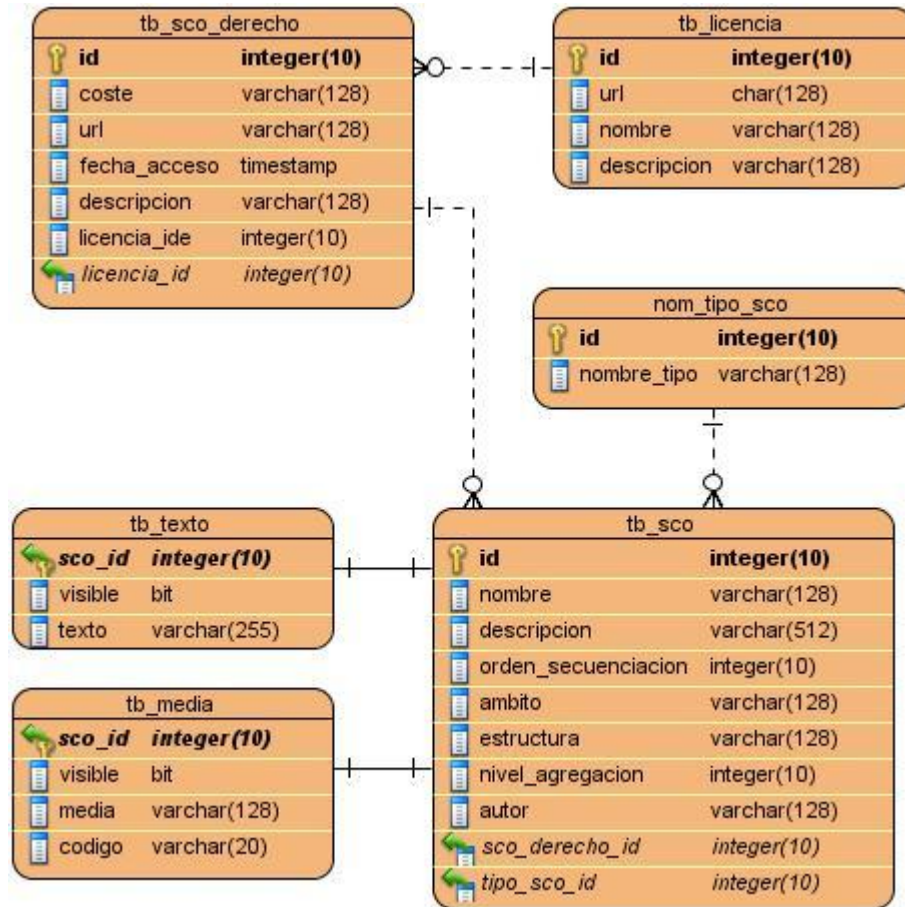


Figura 30: Modelo físico de datos del módulo Mediateca.

3.6 CONCLUSIONES

En este capítulo se ha diseñado el sistema a través de una serie de modelos que serán entregados a los implementadores, facilitándoles las funcionalidades a implementar con el fin de obtener una aplicación que contenga los requisitos que desea el cliente. Para lograr un mejor diseño se utilizaron algunos patrones que ayudan a la optimización del mismo.

CONCLUSIONES GENERALES

Una vez finalizada la investigación y realizadas todas las tareas propuestas, se puede afirmar que se cumplió el objetivo planteado pues:

- Se realizó un estudio de las aplicaciones existentes, similares a Mediateca.
- Se realizó un Modelo del dominio logrando combinar los principales conceptos existentes en la investigación.
- Se obtuvieron las funcionalidades que debía cumplir el sistema mediante los requisitos funcionales y no funcionales del mismo.
- Se especificaron los casos de uso del sistema.
- Se realizó el análisis y diseño de Mediateca con el fin de lograr una mejor comprensión para la futura implementación del módulo.
- Se utilizaron patrones de diseño para optimizar la flexibilidad del diseño realizado.

Como resultado final se obtuvo el modelo de análisis y diseño de una futura herramienta que permitirá la visualización de medias, facilitará la consulta de palabras de difícil comprensión y de datos de personalidades y además posibilitará la adición, modificación y eliminación de las medias relacionadas con las temáticas abordadas en el software.

RECOMENDACIONES

En la realización del trabajo fue posible determinar algunas recomendaciones necesarias a la dirección del proyecto:

- A partir de los artefactos generados seguir con la implementación de las funcionalidades propuestas en las próximas fases del ciclo de desarrollo.
- Integrar el subsistema a los demás subsistemas que componen la colección “El Navegante”.

REFERENCIAS BIBLIOGRÁFICAS

1. I.E.S. Dr. Fleming. *Las Nuevas Tecnologías en la Educación*. [En línea] [Citado el: 3 de Diciembre de 2010.] <http://nntt.informatica-fleming.com/coordina/?p=23>.
2. **Ramos Pérez L, Domínguez Lovaina J, Gavilondo Mariño X, Fresno Chávez C.** *¿Software educativo, hipermedia o entorno educativo?* Ciudad de La Habana, Cuba : Centro Nacional de Información de Ciencias Médicas-Infomed, 2008. Disponible en: http://bvs.sld.cu/revistas/aci/vol18_4_08/aci61008.htm.
3. *La Informática educativa en el contexto actual.* **Rodríguez Lamas, Raúl.** s.l. : Eductec. Revista Electrónica de Tecnología Educativa, 2000. Disponible en <http://www.uib.es/depart/gte/edutec-e/Revelec13/RIamas.html>.
4. **Núñez Rojas, Nemecio.** Concepción del Software educativo. *La Webquest, el aula virtual y el desarrollo de competencias para la investigación en los estudiantes del I Ciclo de educación – USAT.* 2010. Disponible en: www.eumed.net/libros/2010a/669/ .
5. **Rosario, Jimmy.** *La Tecnología de la Información y la Comunicación (TIC). Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual.* 2005. Disponible en el ARCHIVO del Observatorio para la CiberSociedad en <http://www.cibersociedad.net/archivo/articulo.php?art=218>.
6. **MSc. Rodríguez, Mario del Toro.** Modelos de diseño didáctico de hiperentornos de enseñanza-aprendizaje. *Resumen de la Tesis en opción al grado de Doctor en Ciencias Pedagógicas.* Ciudad de La Habana, República de Cuba : Instituto Superior Pedagógico "Enrique José Varona", 2006.
7. **Dr. C. Barreto Gelles, Iván, Dr. C. del Toro Rodríguez, Mario y MSc. Riquenes Tarragó, Aniuska.** *TECNOLOGÍA EDUCATIVA: DOS MODELOS PARA LA ACCIÓN DEL MAESTRO.* La Habana, Cuba : Ministerio de Educación Cubano, 2009.
8. e-ABC. [En línea] [Citado el: 3 de Noviembre de 2010.] <http://www.e-abclearning.com/queesscorn>.
9. **Hernández Valverde, Pedro.** *El Proceso Unificado de Rational (RUP) y su relación con las técnicas y métodos de la ingeniería de usabilidad del software.* 2005.
10. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** . *El Proceso Unificado de Desarrollo, La guía completa del Proceso Unificado, escrita por sus creadores.* Madrid : Addison Wesley, 2000.

11. Lenguaje Unificado de Modelado. *EIDV Soluciones Tecnológicas*. [En línea] [Citado el: 7 de Diciembre de 2010.] <http://www.eidv.com.ar/articulos/tecnologia-de-la-informacion/lenguaje-unificado-de-modelado/>.
12. **Landeros Gómez, Ruth Priscila y Del Juncal Huerta, Jorge Luis**. Herramientas Case. s.l. : Universidad Veracruzana, 2007.
13. IEEE Standards Association. [En línea] [Citado el: 15 de Enero de 2011.] http://standards.ieee.org/reading/ieee/std_public/description/se/1471-2000_desc.html.
14. **López, Alejandro Rivera**. *Sistema asistente para la generación de horarios de cursos*. Cholula, Puebla, México : Escuela de Ingeniería y Ciencias, Departamento de Computación, Electrónica, Física e Innovación., 16 de enero de 2008.
15. Los diferentes lenguajes de programación para la web . *New Web Star*. [En línea] [Citado el: 16 de Diciembre de 2010.] <http://www.newwebstar.com/ebooks/133193-los-diferentes-lenguajes-de-programaciun-para-la.html>.
16. Que son los FrameWorks. *SOA Agenda*. [En línea] [Citado el: 10 de Diciembre de 2010.] <http://www.soaagenda.com/journal/.../que-son-los-frameworks/>.
17. **Tavárez, David**. Maestros del Web. [En línea] 16 de Diciembre de 2010. <http://www.maestrosdelweb.com/editorial/comparacion-frameworks-javascript/>.
18. **Potencier, F. y Zaninotto, F**. LibrosWeb.es. [En línea] [Citado el: 15 de Diciembre de 2010.] http://librosweb.es/symfony_1_2/capitulo1/symfony_en_pocas_palabras.html.
19. **Alvarez, Miguel Angel**. *Manual de jQuery*. 2009. Disponible en: www.biblioteca-digital.net.ve/wordpress/wp-content/.../manual-jquery.pdf.
20. **Sánchez Suárez, Jose Manuel**. <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jQueryIntro>. [En línea] [Citado el: 10 de Diciembre de 2010.]
21. **Pressman, Roger S**. *Ingeniería de software un enfoque práctico*. Cuarta edición. 2002.
22. **Hurtado Gil, Sandra Victoria**. *Representación de la arquitectura de software usando UML*. s.l. : Universidad Icesi-I2T, 2008.
23. **Schmuller, J**. *Aprendiendo UML en 24 horas* . s.l. : Prentice-Hall, 2001.
24. **Chesini, Ulises**. *Selección e implementación de prácticas de Diseño Centrado en el Usuario en un Proceso RUP*. s.l. : Facultad de Informática, UNLP, 2006.

25. **Lic Espinoza, Robles.** El flujo de análisis. [En línea] [Citado el: 3 de Febrero de 2011.]
<http://www.slideshare.net/juliopari/13-clase-flujo-de-analisis>.
26. **ARIZACA, LIC. ELISA.** *Patrones de Diseño*. s.l. : Universidad Salesiana de Bolivia, 2008.
27. **Martínez, Alejandro y Martínez, Raúl.** *Guía a Rational Unified Process*. España : Escuela Politécnica Superior de Albacete – Universidad de Castilla la Mancha, 2011.
28. Milestone. *Uso de UML en aplicaciones Web: páginas y relaciones*. [En línea] [Citado el: 2011 de Febrero de 19.]
http://www.milestone.com.mx/articulos/uso_de_uml_en_aplicaciones_web.htm.
29. **Booch, Grady.** *Análisis y diseño orientado a objetos con aplicaciones*. s.l. : 2ª ed., Addison-Wesley, cop. 1996 .