



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad #8

ANÁLISIS Y DISEÑO DEL MÓDULO TEMAS DE LA COLECCIÓN MULTISABER

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

Autora:

Yaritza Galán Garcés

Tutora:

Ana Margarita Rojas Raberón

Ciudad de la Habana, julio del 2008

*El inteligente no es aquel que lo sabe todo,
sino aquel que sabe utilizar lo poco que sabe.*

Sebastian Cohen Saavedra

DECLARACIÓN DE AUTORÍA

Declaro ser autora 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 Julio del 2008.

Yaritza Galán Garcés

Ana Margarita Rojas Raberón

AGRADECIMIENTOS

Le agradezco a mi mamá, a mi papá y a mis hermanos por todo lo que aportaron para que yo estuviera aquí, las cosas que han hecho, por estar siempre presente en los momentos que más los he necesitado.

También les agradezco a todos aquellos amigos que han aportado un granito de arena para que mis sueños se hicieran realidad.

RESUMEN

En el siguiente trabajo se expone el resultado del análisis y diseño realizado al módulo Temas de los nuevos productos de la Colección Multisaber; para que los niños de las escuelas venezolanas cuenten con una colección de software que contienen imágenes, animaciones, videos, sonidos y textos, que estos permiten una mejor interactividad con la tecnología multimedia, es decir que en este trabajo se propone un diseño que permite controlar el proceso de cómo interactuar con los contenidos de los temas de cada multimedia. Teniendo en cuenta que esta aplicación debe ejecutarse sobre software libre y además que sea multiplataforma, se determinó usar la metodología RUP/ApEM-I para el proceso de desarrollo del software y modelado de los diferentes artefactos del flujo de trabajo análisis y diseño en Visual Paradigm logrando un desarrollo validado en la calidad que esta herramienta impone.

ÍNDICE

INTRODUCCIÓN	1
1. FUNDAMENTACIÓN DEL TEMA, TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR.	3
Introducción	3
1.1 Antecedentes de la multimedia.	3
1.2 Software multimedia.	6
1.3 Funciones del software multimedia educativo.	6
1.4 Colección Multisaber.....	8
1.5 Software Libre.....	9
1.6 Intento de producción de software multimedia con la utilización del software libre.	10
1.7 Metodologías de desarrollo del software.	11
1.7.1 MSF: Microsoft Solution Framework.....	11
1.7.2 XP: Programación Extrema (Extreme Programming).....	12
1.7.3 RUP: Proceso Unificado Racional.	12
1.8 Lenguajes de modelado.....	17
1.8.1 UML: Lenguaje de Modelado Unificado.	17
1.8.2 OMMMA – L: Lenguaje para la modelación Orientada a Objetos de Aplicaciones Multimedia.	17
1.8.3 OCL: Object Constraint Language.....	18
1.8.4 ApEM-L: Lenguaje de Modelado de Aplicaciones Educativas Multimedias.	18
1.9 Herramientas de Modelado Visual.	19
1.9.1 Rational Rose.....	19
1.9.2 Visual Paradigm 2.3.	20
Conclusiones	22
2. CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA.	23
Introducción	23
2.1 Modelo del Dominio.	23
2.1.1 Requisitos Funcionales.	25
2.1.2 Requisitos No Funcionales.....	25

2.2 Descripción del sistema propuesto.	27
2.2.1 Vistas de Presentación por interfaz de comunicación con el usuario.	27
2.2.2 Actores del sistema.	27
2.2.3 Descripción Textual de las Vistas de Presentación.	28
2.2.4 Vista de Gestión del Modelo.	32
2.2.5 Diagrama de Estructura de Presentación por vista.	33
2.2.6 Diagrama de Estructura de Navegación.	36
2.2.7 Diagrama de Clases.	37
2.2.8 Diagrama de Interacción.	40
2.2.9 Diagrama de Despliegue.	45
Conclusiones	45
3. ESTUDIO DE FACTIBILIDAD.	47
Introducción	47
3.1 Estimación del Costo del Proyecto.	47
3.1.1 Puntos de Función.	49
3.2 Beneficios tangibles e intangibles.	52
3.2.1 Beneficios Tangibles.	52
3.2.2 Beneficios intangibles.	52
Conclusiones	53
CONCLUSIONES GENERALES	54
RECOMENDACIONES	55
REFERENCIAS BIBLIOGRÁFICAS	56
BIBLIOGRAFÍA	58
ANEXOS	60
Anexo I. Diagrama de colaboración de la vista Temas.	60
Anexo II. Diagrama de colaboración de la vista Títulos.	61
Anexo III. Diagrama de colaboración de la vista Contenidos.	62
GLOSARIO DE TÉRMINOS	63

ÍNDICE DE FIGURAS

Figure 1. RUP en dos dimensiones.....	15
Figure 2. Modelo Conceptual.	24
Figure 3. Representación de los Actores del Sistema.	28
Figure 4. Representación de la Vista de Gestión del Modelo.	32
Figure 5. Diagrama de Estructura de Presentación de la vista Temas.	34
Figure 6. Diagramas de Estructura de Presentación de la vista Título.....	35
Figure 7. Diagramas de Estructura de Presentación de la vista Contenidos.....	36
Figure 8. Diagrama de Estructura de Navegación.	37
Figure 9. Diagrama de Clases del diseño de la vista Temas.	38
Figure 10. Diagrama de Clases del diseño de la vista Título.	39
Figure 11. Diagrama de Clases del diseño de la vista Contenidos.	40
Figure 12. Estereotipos de las clases interfaz, controladoras y entidad.....	41
Figure 13. Diagrama de Secuencia de la vista Temas.	42
Figure 14. Diagrama de Secuencia de la vista Títulos.....	43
Figure 15. Diagrama de Secuencia de la vista Contenidos.	44
Figure 16. Diagrama de Despliegue de la aplicación.....	45

ÍNDICE DE TABLAS

Tabla 1. Requerimiento de Software.....	26
Tabla 2. Requerimientos de Hardware.	26
Tabla 3. Restricciones en el diseño y la implementación.	26
Tabla 4. Descripción Textual de la Vista de Presentación Temas.....	28
Tabla 5. Descripción Textual de la Vista de Presentación Títulos.....	30
Tabla 6. Consultas Externas.....	47
Tabla 7. Salidas Externas del módulo Temas.....	48
Tabla 8. Consultas Externas del módulo Temas.....	49
Tabla 9. Puntos de Función.....	49
Tabla 10. Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo. .	49
Tabla 11. Factores de Escala.	50

INTRODUCCIÓN

Con la llegada al poder del presidente Hugo Chávez, en la hermana nación de Venezuela comenzaron a surgir una serie de transformaciones a favor de la clase obrera, el campesinado, la juventud, los niños, o sea, para el bienestar de la sociedad. Se han producido cambios en casi todas las esferas, en la salud, la economía, en la política y en la educación.

De igual forma se desarrollan acciones para mejorar el nivel cultural de todos los venezolanos, por tal motivo se hace necesaria la creación de productos que contengan información sobre diversos temas, que sean atractivos, dinámicos, que presenten la información de una manera sencilla, que sean interactivos para que los niños interactúen con las computadoras. De aquí se quiere que los niños adquieran los conocimientos necesarios para su desarrollo: tanto de la naturaleza, de la historia, del ajedrez, del inglés, las matemáticas y además que contengan juegos para desarrollar las habilidades que puedan adquirir con estos.

En estos momentos se están iniciando unas transformaciones para elevar el nivel cultural, como es la introducción de las nuevas tecnologías a la educación que con ayuda de nuestro país, se esta realizando, la contextualización y la migración de la Colección Multisaber en la Universidad de las Ciencias Informáticas (UCI) pero se hace necesario que cada multimedia sean creadas en software libre o sea que sean capaz de correr en los sistemas operativos Windows, Linux, Macintosh.

Debido a la no existencia de documentación de productos anteriores los cuales se hayan creados con la tecnología multimedia y utilizado el software libre; en la Universidad de las Ciencias Informáticas se están creando las bases para la documentación del proceso análisis y diseño.

Hasta aquí se ha expuesto de una forma breve la situación problemática, la cual conduce a la siguiente interrogante.

¿Cómo lograr la documentación requerida en el Flujo de Trabajo de análisis y diseño del módulo Temas de los nuevos software de la Colección Multisaber?

Con el propósito de dar respuesta a la incógnita anteriormente expuesta, se realizará un profundo estudio a la documentación de la Colección Multisaber, y se concentrará la investigación en el módulo Temas en el proceso de documentación para la migración de los software educativos con tecnología multimedia a software libre.

Como objetivo general tenemos realizar el análisis y diseño del módulo Temas de la Colección Multisaber.

De este objetivo general se plantean los siguientes objetivos específicos.

- Realizar un estudio de la documentación existente de los productos anteriores de la colección Multisaber.
- Estudiar metodologías existentes para el desarrollo de software educativo utilizando la tecnología multimedia y seleccionar la más adecuada.
- Estudiar los lenguajes de modelados para el desarrollo de software educativo utilizando la tecnología multimedia.
- Elaborar la documentación de módulo Temas de los nuevos productos de la Colección Multisaber.

Se enfocarán todos los esfuerzos en cumplir los objetivos anteriormente definidos para plantear un modelado de solución informática ya que si se logra la documentación necesaria y suficiente para desarrollar este producto en software libre, podrá servir de guía para creación o migración de productos similares, teniendo en cuenta que no existe la documentación para la realización de una multimedia a software libre. Constituye entonces una novedad la creación de la documentación de un nuevo proceso: la migración del módulo Temas en las multimedia a software libre.

Con la documentación de este proceso y después la marcha a la creación de multimedias habrá un gran aporte ya que permitirá a los software multimedias podrán contar con la documentación necesaria para estudios o cambios posteriores para futuras creaciones.

1. FUNDAMENTACIÓN DEL TEMA, TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR.

Introducción

Cada día que pasa el avance tecnológico resulta ser una de las cosas más interesantes; se desarrolla en todas las esferas de la sociedad, siendo así el renglón principal de diferentes sectores sociales.

Una de las tecnologías más importantes ha sido la tecnología multimedia ya que esta se usa para el desarrollo de la educación, se está desarrollando cada día más minuciosamente el proceso de su creación. Y partiendo de cómo surge ésta, en este capítulo se expondrán las tecnologías más adecuadas para la creación del módulo Temas de la Colección Multisaber pues usar la correcta metodología, el correcto lenguaje de modelado y la adecuada herramienta de modelado, ayuda a desarrollar los componentes necesarios de una manera más rápida y eficiente para creación de software educativos con tecnología multimedia.

1.1 Antecedentes de la multimedia.

El invento del transistor, a partir de los años 50, posibilitó la revolución de la computadora, con la fabricación del chip, los circuitos eléctricos y las tarjetas electrónicas, los cuales propician unidades compactas de procesamiento y la integración del video. Posteriormente, una serie de accesorios y periféricos han sido desarrollados para que la computadora pueda manejar imagen, sonido, gráficas y videos, además del texto.

Por otro lado, la comunicación desarrolla, a partir de los 70s, en la educación, la instrucción, la capacitación y la publicidad, el concepto operativo de multimedia. Por tal concepto se entiende la integración de diversos medios (visuales y auditivos) para la elaboración y envío de mensajes por diversos canales, potencializando la efectividad de la comunicación, a través de la redundancia; pues, así, la comunicación resulta

más atractiva, afecta e impacta a más capacidades de recepción de la persona y aumenta la posibilidad de eliminar el ruido que puede impedir la recepción del mensaje.

En el ámbito de la computación el término multimedia es más nuevo y designa el uso de varios recursos o medios, como audio, video, animaciones, texto y gráficas en una computadora. Al integrar los datos que puede manejar la computadora, la multimedia ofrece posibilidades de creatividad mediante los sistemas de computación.

La multimedia se inicia en 1984. En ese año, Apple Computer lanzó la Macintosh, la primera computadora con amplias capacidades de reproducción de sonidos equivalentes a los de un buen radio AM. Esta característica, unida a que: su sistema operativo y programas se desarrollaron, en la forma que ahora se conocen como ambiente windows, propicios para el diseño gráfico y la edición, hicieron de la Macintosh la primera posibilidad de lo que se conoce como Multimedia.

La tecnología multimedia toma auge en los video-juegos, a partir de 1992, cuando se integran: audio (música, sonido estéreo y voz), video, gráficas, animación y texto al mismo tiempo. [5]

Hoy en día los sistemas de autor (authoring systems) y el software de autor (authoring software), permiten desarrollar líneas de multimedia integrando 3 o más de los datos que son posibles de procesar actualmente por computadora: texto y números, gráficas, imágenes fijas, imágenes en movimiento y sonido y por el alto nivel de interactividad, tipo de navegación. Los software de autor permiten al "desarrollador de multimedia" generar los prototipos bajo la técnica llamada "fast prototype" (prototipo rápido), el método más eficiente de generar aplicaciones.

Se reconoce que los "authoring software" hacen más eficiente el proceso de producción de multimedia en la etapa de diseño, la segunda de las cuatro etapas que se reconocen para el desarrollo de la misma, porque allí es donde se digitaliza e integra la información. [11]

Mediante la interacción con la máquina, la multimedia tendrá una función semejante a la de los libros en el aprendizaje e información, tendrá su base en las imágenes interactivas y en la premisa de que "la gente adquiere sus conocimientos de manera más efectiva manejando la información de manera interactiva". [5]

Hoy en día, los cambios asegurados son una realidad y la tecnología multimedia es tan común que resulta impensable una computadora sin ellos. Los multimedios computarizados emplean los medios - la palabra, los recursos de audio, las imágenes fijas y las imágenes en movimiento- para tener una mayor interacción con el usuario quien ha pasado de ser considerado como alguien que esporádicamente empleaba una computadora (con el respectivo recelo e inseguridad) a ser quien la maneja como una herramienta más en su beneficio.

Las aplicaciones multimedia comprenden productos y servicios que van desde la computadora (y sus dispositivos "especiales" para las tareas multimedia, como bocinas, pantallas de alta definición, etc.) donde se puede leer desde un disco compacto hasta las comunicaciones virtuales que posibilita Internet, pasando por los servicios de vídeo interactivo en un televisor y las videoconferencias. [11]

Multimedia se ha convertido en la palabra talismán de los últimos años. En la actualidad multimedia puede significar muchas cosas, dependiendo del contexto en que nos encontremos y del tipo de especialista que lo defina.

Multimedia hoy suele significar la integración de dos o más medios de comunicación que pueden ser controlados o manipulados por el usuario vía ordenador. [6]

Se refiere normalmente a vídeo fijo o en movimiento, texto, gráficos, audio y animación controlados por un ordenador. Pero esta integración no es sencilla. Es la combinación de hardware, software y tecnologías de almacenamiento incorporadas para proporcionar un entorno multisensorial de información.

Es la capacidad de mostrar gráficos, video, sonido, texto y animaciones como forma de trabajo, e integrarlo todo en un mismo entorno llamativo para el usuario, que interactuara o no sobre el para obtener un resultado visible, audible o ambas cosas". Esto aplicado técnicamente en una computadora personal. Se le describe también como una "nueva plataforma que integra componentes para hacer ciertas tareas; combinando a las computadoras y los medios tradicionales en una nueva forma de expresión. [8]

De forma general Multimedia no es más que la integración de varios elementos (texto, sonido, video) en un solo medio, que posibilitará la interacción hombre- computador, facilitando así el manejo de información.

1.2 Software multimedia.

Conocida como una de las áreas de mayor importancia de la Informática, como ya se había dicho, permite combinar diversos medios como texto, sonido, vídeo y gráficas, en una sola aplicación, que, junto con la técnica del hipertexto, permite agregar interactividad; lo que hace que el usuario pueda navegar a través de la aplicación, a su libre elección, de acuerdo con sus intereses o necesidades de aprendizaje.

La multimedia evita las muchas dificultades que en países poco desarrollados se presentan al hacer uso de la red de redes, quien ofrece sistemas de enseñanza más sofisticados y con mayores posibilidades, no obstante debemos tener en cuenta que la solución mejor será siempre la que se adapte a las condiciones de cada lugar, modelo pedagógico, alcance tecnológico, etc., ningún modelo o tecnología será mejor o peor fuera del contexto en que se intente aplicar.

Ningún contenido multimedia, educativo o no, puede ser llevado con éxito a grandes escalas sin una previa organización del proceso de producción, más al tener en cuenta la amplia gama de recursos que intervienen y la necesidad de la sincronización de los esfuerzos.

Dentro del grupo de los materiales multimedia están los materiales multimedia educativos, que son los materiales multimedia que se utilizan con una finalidad educativa. [9]

Atendiendo a su estructura, los materiales didácticos multimedia se pueden clasificar en programas tutoriales, de ejercitación, simuladores, bases de datos, constructores, programas herramienta..., presentando diversas concepciones sobre el aprendizaje y permitiendo en algunos casos (programas abiertos, lenguajes de autor) la modificación de sus contenidos y la creación de nuevas actividades de aprendizaje por parte de los profesores y los estudiantes. [9]

1.3 Funciones del software multimedia educativo.

Los materiales multimedia educativos, como los materiales didácticos en general, pueden realizar múltiples funciones en los procesos de enseñanza y aprendizaje. Las

principales funciones que pueden realizar los recursos educativos multimedia son las siguientes: [6]

Informativa: Estos materiales, a través de sus actividades, presentan unos contenidos que proporcionan información estructurada de la realidad, a los estudiantes.

Instructiva Entrenadora: Todos los software educativos orientan y regulan el aprendizaje de los estudiantes ya que, explícita o implícitamente, promueven determinadas actuaciones de los mismos encaminadas a este fin. Además, mediante sus códigos simbólicos, estructuración de la información e interactividad condicionan los procesos de aprendizaje.

Motivadora: La interacción con el ordenador suele resultar por sí misma motivadora.

Estos programas incluyen además elementos para captar la atención de los alumnos, mantener su interés y focalizarlo hacia los aspectos más importantes.

Evaluadora: La posibilidad de "feed back" inmediato a las respuestas y acciones de los alumnos, hace adecuados a los programas para evaluarles. Esta evaluación puede ser:

- Implícita: el estudiante detecta sus errores, se evalúa a partir de las respuestas que le da el ordenador.
- Explícita: el programa presenta informes valorando la actuación del alumno.

Explorar Experimentar: Les presentan a los estudiantes interesantes entornos donde explorar, experimentar, investigar, buscar determinadas informaciones, cambiar los valores de las variables de un sistema, etc.

Innovadora: Los programas educativos pueden desempeñar esta función ya que utilizan una tecnología actual y, en general, suelen permitir muy diversas formas de uso. Esta versatilidad abre amplias posibilidades de experimentación didáctica e innovación educativa en el aula.

1.4 Colección Multisaber.

A partir de la tercera Revolución Educativa nuestro país se da la tarea de realizar transformaciones en la educación, donde estas transformaciones estuvieron inspirada en la “Batalla de Ideas”, se crearon 16 centros de desarrollo de software educativo en todo el país sobre la base de cada uno de los ISP del MINED, para elevar el componente educativo de la formación del hombre quiere decir la formación de valores y el desarrollo del sentimiento de independencia, se disminuyó la cantidad de alumnos por profesores para lograr una mayor atención a las diferencias individuales de los educandos. De esta forma se crean los “maestros integrales” (PGI) y forma surge la idea de la creación de una colección de software educativo para cada sistema de educación del país, estos fueron: Multisaber para la educación primaria, Navegante para educación secundaria y Futuro para la educación preuniversitaria.

Para la creación de esta misión estuvo a cargo la dirección política de la organización de la Unión de Jóvenes Comunistas (UJC), contó con la parte pedagógica y la asesoría técnica del MINED y por parte informático se conto con el Ministerio de la Informáticas y las Comunicaciones. Estas colecciones responden a la tipología de hiperentornos no es mas que una mezcla de diversas tecnologías de software (libros electrónicos, entrenadores, evaluadores, juegos, simuladores, bibliotecas virtuales, etc.). La cual se implementó por las siguientes instituciones: MINED, MES, Joven Club de Computación y Electrónica (JCCE).

A continuación se hará una breve descripción de las principales características de la Colección Multisaber, tiene un enfoque curricular y multidisciplinario por su relación con los contenidos de los programas de cada asignatura del currículo de estudio de la Educación Primaria. Cuenta también con un grupo de software que tributan a la formación de una cultura general integral. Está constituida por una concepción pedagógica que se ha dado a conocer como Hiperentornos de Aprendizaje en la que se integran armónicamente módulos como: Temas, Ejercicios, Juegos, Biblioteca, Registro y Maestro, entre otros. Contempla una interfaz estandarizada, que proporciona un ambiente de trabajo amigable e intuitivo con un alto nivel de interactividad para acceder a la información existente en el software. [5]

La colección posee diversos servicios informáticos como son: la búsqueda y extracción de información para realizar la impresión y en algunos casos para procesar la información en otras aplicaciones informáticas.

Ejemplo de esto es el Registro que va guardando la “traza” (desenvolvimiento del estudiante) de manera exhaustiva, cuyo análisis permite establecer estrategias de corrección y diagnóstico por parte del docente o tutor.

También algunos de los programas de la colección son configurables a partir de las bases de datos que contiene el propio software y otros tienen carácter abierto posibilitando la actualización de los ejercicios con nuevas bases de datos.

1.5 Software Libre.

Con la necesidad de la utilización del software libre para la creación de este producto se hace imprescindible el estudio de este.

Software libre (free software) es la dominación del software que brinda libertad a los usuarios sobre el producto adquirido; no confundirlo con software gratis, es decir Software libre es la libertad con que cuentan los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software de modo más efectivo, que este se ajuste al contexto y peticiones del usuario. Software libre se refiere a cuatro libertades de los usuarios del software.

- **Libertad 0:** La libertad de usar el programa, con cualquier propósito.
- **Libertad 1:** La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades. El acceso al código fuente es una condición previa para esto.
- **Libertad 2:** La libertad de distribuir copias, con lo que puedes ayudar a tu vecino.
- **Libertad 3:** La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto. [2]

1.6 Intento de producción de software multimedia con la utilización del software libre. [12]

En el sitio personal de Marko Sanchez, el pretende ofrecer un acercamiento a la utilización de software libre en la producción de multimedia donde da una serie de herramientas que brinda el software libre que con su unión posibilitarían la creación de multimedias. Por ejemplo:

A la hora de diseñar, el software libre ofrece el Gimp (Programa de manipulación de imágenes de GNU), con esta herramienta se pueden crear dibujos, retocar imágenes, editar capturas de pantalla y hacer los diseños a nivel profesional. Por otra parte, se dispone del Inkscape (es un editor de gráficos vectoriales de código abierto, con capacidades similares a Illustrator, Freehand, CorelDraw o Xara) pues, a veces es necesario crear diseños vectoriales para ilustraciones y dibujos.

Para la edición de textos, hoja de cálculo y programa de presentaciones se dispone de OpenOffice. Este conjunto de herramientas nos permite, del mismo modo que el Microsoft Office, manejar textos con imágenes, crear presentaciones, etc. Y si lo que se necesita es crear archivos PDF, se puede utilizar el PDFCreator.

En el caso de que se necesite crear sitios web se pueden utilizar tanto NVU (editor WYSIWYG (o sea visual), permite editar la página web tal y como si estuviera en el navegador) como Kompozer. Ambas herramientas tienen un sistema que permite ir observando los cambios que se realizan al ritmo de edición, también llamado, lo que ves es lo que obtienes (WYSIWYG por sus siglas en inglés).

Si se necesitara crear tutoriales se puede utilizar el Wink. Además, de que es una herramienta muy eficiente, exporta en archivos SWF sus presentaciones.

Para la realización de animaciones vectoriales SWF se puede utilizar el Synfig. Que permite crear películas animadas en dos dimensiones con calidad de películas de cine. Pero, si lo que se necesita es editar video, el Jahshaka permite una edición no lineal muy poderosa. Finalmente, si lo que se necesita es editar sonidos, con el Audacity se puede lograr.

Pero se llegó a la conclusión de que el proceso de creación de software con tecnología multimedia estaría casi completo, salvo porque, no existe en este momento una

herramienta como el Macromedia Director o como el Macromedia Flash en software libre. El Synfig permite hacer animaciones flash, pero no dispone de posibilidades de programación como el ActionScript.

1.7 Metodologías de desarrollo del software.

El problema que presentan los desarrolladores hoy en día a la hora de realizar una aplicación es la necesidad de saber cómo organizar las actividades para cada desarrollador por separado y para el equipo, definir qué artefactos deben ser creados y contar con una serie de criterios que permitan controlar y medir los productos que se obtienen, por lo tanto se necesita de una metodología capaz de dirigir estas actividades y así convertir los requisitos de los usuarios en un producto software. En el mundo existen distintas metodologías para dirigir las actividades vinculadas al proceso de desarrollo de software, entre estas metodologías se estudiaron las siguientes:

1.7.1 MSF: Microsoft Solution Framework.

Es una metodología que se considera tradicional, es de desarrollo orientado a objetos, flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

Esta metodología presenta las siguientes características: [13]

Adaptable: Es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.

Escalable: Puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.

Flexible: Es utilizada en el ambiente de desarrollo de cualquier cliente. Tecnología.

Agnóstica: Porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto,

Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

1.7.2 XP: Programación Extrema (Extreme Programming).

XP es una metodología para equipos de desarrollo de software pequeños o medianos, desventaja fundamental para la aplicación de esta metodología al software en cuestión. Es una metodología que se utiliza en organizaciones en donde los requisitos son vagos y variantes. Sus elementos claves son: poca documentación, simplicidad, análisis como una actividad constante, diseño evolutivo, integraciones y testeos diarios. Elimina la gran documentación del sistema como un todo y ataca mínimas partes del sistema que son rápidamente implementadas y testeadas. De esta forma el sistema va creciendo junto con el nuevo conocimiento del cliente y sus nuevas necesidades.

La metodología XP se basa en: [11]

- Pruebas Unitarias: Basadas en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, se pueda hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.
- Re-fabricación: Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

1.7.3 RUP: Proceso Unificado Racional.

“Los orígenes del Proceso Unificado de Desarrollo de Software (RUP) se remontan al modelo espiral original de Barry Boehm. Ken Hartman, uno de los contribuidores claves de RUP, colaboró con Boehm en la investigación. En 1995 Rational Software es comprada por una compañía sueca llamada Objectory AB. El Proceso Unificado de Desarrollo de Software (RUP) fue el resultado de una convergencia de Rational

Approach y Objectory, proceso desarrollado por el fundador de Objectory Ivar acobson. El primer resultado de esta fusión fue el Rational Objectory Process, la primera versión de RUP, fue puesta en el mercado en 1998, siendo el arquitecto en jefe Philippe Kruchten”. [14]

Los elementos del RUP son:

- **Actividades:** Son los procesos que se llegan a determinar en cada iteración.
- **Trabajadores:** Son las personas o entes involucrados en cada proceso.
- **Artefactos:** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

RUP presenta cuatro fases, estas son:

- **Fase de conceptualización (Concepción o Inicio):** Esta fase tiene por finalidad definir la visión, los objetivos y el alcance del proyecto, tanto desde el punto de vista funcional como del técnico, obteniéndose como uno de los principales resultados una lista de los casos de uso y una lista de los factores de riesgo del proyecto. El principal esfuerzo está radicado en el “Modelamiento del Negocio” y el “Análisis de Requerimientos”. Es la única fase que no necesariamente culmina con una versión ejecutable, si bien muchas veces se desarrollan las interfaces con el usuario, o se prueban algunos aspectos técnicos críticos (por ejemplo la factibilidad de conectarse a una determinada Base de Datos).
- **Fase de elaboración:** Esta fase tiene como principal finalidad completar el análisis de los casos de uso y definir la arquitectura del sistema. En esta etapa se busca eliminar los principales riesgos técnicos.
- **Fase de construcción:** Esta fase está compuesta por un ciclo de varias iteraciones, en las cuales se van incorporando sucesivamente los casos de uso, de acuerdo a los factores de riesgo del proyecto. Este enfoque permite por ejemplo contar en forma temprana con versiones del sistema que satisfacen los principales casos de uso. Los cambios en los requerimientos no se incorporan hasta el inicio de la próxima iteración.

- **Fase de transición:** Esta fase se inicia con una versión “beta” del sistema y culmina con el sistema en fase de producción.

Flujos de trabajo.

- **Modelamiento del negocio:** Describe cómo desarrollar una visión de la nueva organización, basado en esta visión se definen procesos, roles y responsabilidades de la organización, se busca un mejor entendimiento de la organización donde se va a implantar el producto.
- **Requerimientos:** Establece qué tiene que hacer exactamente el sistema, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Traduce los requisitos a una especificación que describe cómo implementar el sistema.
- **Implementación:** Se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. Además se deben hacer las pruebas de unidad: cada implementador es responsable de probar las unidades que produzca.
- **Prueba (Testeo):** Se encarga de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida.
- **Instalación o despliegue:** Produce con éxito distribuciones del producto y distribuirlo a los usuarios y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Gestión del proyecto:** Es el arte de lograr un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.
- **Administración de configuración y cambios:** Mantiene la integridad de todos los artefactos que se crean en el proceso, así como de mantener información del proceso evolutivo que han seguido.

- **Ambiente:** Da soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir.

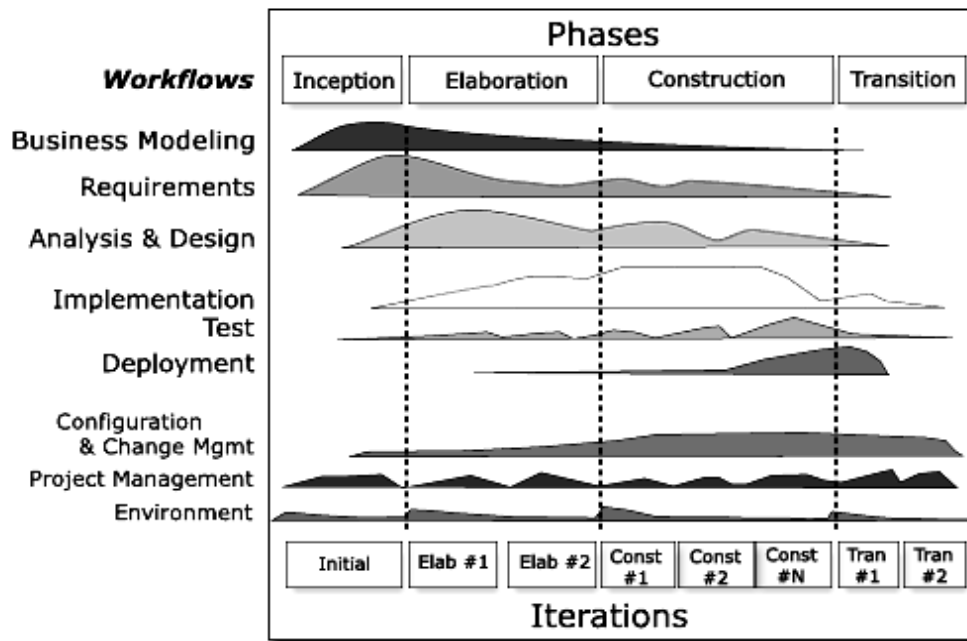


Figure 1. RUP en dos dimensiones.

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptadas al contexto y necesidades de cada organización. Reemplaza el tradicional esquema de desarrollo “en cascada”, donde cada etapa se iba desarrollando en forma secuencial (modelado del negocio, análisis de requerimientos, diseño, construcción, prueba e implementación) por un proceso iterativo e incremental.

El Proceso Unificado de Desarrollo de Software (RUP), es un proceso de desarrollo de software que junto con el Lenguaje Unificado de Modelado (UML) constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso.

- Dirigido por los casos de uso:

RUP define los casos de uso como el conjunto de acciones que debe realizar un sistema para dar un resultado de valor a un determinado usuario y los utiliza, tanto para especificar los requisitos funcionales del sistema, como para guiar todos los demás pasos de su desarrollo, dígase diseño, implementación y prueba.

- Estar centrado en la arquitectura:

La arquitectura es una vista del diseño completo con las características más importantes, dejando a un lado los detalles. Esta no solo incluye las necesidades de los usuarios, sino también otros aspectos técnicos como el hardware, sistema operativo, sistema de gestión de base de datos y protocolos de red con los que debe coexistir el sistema. En otras palabras, la arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.

- Ser iterativo e incremental:

La alta complejidad de los sistemas actuales hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. Cada uno de estos mini-proyecto se les denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo. En cada iteración los desarrolladores seleccionan un grupo de casos de uso, los cuales se diseñan, implementan y prueban. La planificación de iteraciones hace que se reduzcan los riesgos de los costes de un solo incremento, no sacar al mercado un producto en el tiempo previsto, mantener la motivación del equipo pues puede ver avances claros a corto plazo y que el desarrollo pueda adaptarse a los cambios en los requisitos.

Se decidió utilizar esta metodología porque captura varias de las mejores prácticas en el desarrollo moderno de software, es una guía de cómo utilizar de manera efectiva el lenguaje de modelado, provee a cada miembro de un equipo un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo y crea y mantiene modelos, en lugar de enfocarse en la producción de una gran cantidad de papeles de documentación.

1.8 Lenguajes de modelado.

El lenguaje de modelado de objetos es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software orientado a objetos. Algunas organizaciones los usan en combinación con una metodología de desarrollo de software para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo un equipo de desarrolladores.

1.8.1 UML: Lenguaje de Modelado Unificado.

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

Ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. UML detalla los artefactos en el sistema y puede modelar además de los sistemas de software, sistemas de hardware y organizaciones del mundo real.

1.8.2 OMMMA – L: Lenguaje para la modelación Orientada a Objetos de Aplicaciones Multimedia.

OMMMA – L toma como base el lenguaje UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos, y basa su descripción en el patrón de arquitectura MVCMM. Tiene cuatro vistas fundamentales en su modelación: vista lógica, vista de presentación espacial, vista de comportamiento temporal predefinido y vista de control interactivo. Modifica los diagramas originales de UML de: clases, secuencia y estado. En la vista de presentación espacial añade un nuevo diagrama: el diagrama de presentación, para la representación espacial de los elementos visuales del software multimedia.

1.8.3 OCL: Object Constraint Language.

El lenguaje de especificación OCL es el lenguaje propuesto para especificar las restricciones semánticas del diagrama de clases. Se puede utilizar en cualquier contexto en que se use el diagrama de clases al que esté asociado. OCL es un lenguaje híbrido orientado a objetos-funcional, y es tipificado, porque usa el concepto de tipos de datos. Estos tipos de datos se encuentran jerarquizados a través de relaciones de inclusión.

El componente principal construido por este lenguaje es la expresión, que se manipula a través de operaciones que garantizan la transparencia referencial y la falta de efectos colaterales ya que toda expresión válida para el lenguaje OCL debe tener asociado un tipo de datos. Es decir las expresiones del lenguaje OCL se estructuran hasta llegar a su punto más alto, que es el de formar parte de una definición. En cada definición de OCL se describe cada restricción semántica del diagrama de clases, junto a otras propiedades también descritas con expresiones. Cada definición en OCL se aplica a lo que se llama un contexto, que representa el destinatario del diagrama de clases sobre quien se aplica esa definición.

1.8.4 ApEM-L: Lenguaje de Modelado de Aplicaciones Educativas Multimediales.

La tercera generación consiste en intentos creíbles de crear lenguaje unificado por la industria; para la creación del módulo se utilizará el lenguaje de modelado ApEM-L, este añade los elementos más significativos de extensiones anteriores como OMMMA – L (2001) y respeta lo establecido por el estándar OCL (2003), lo que produce los siguientes beneficios:

- Puede utilizar para su representación todas las herramientas CASE que existen actualmente para la modelación de UML.
- Es un lenguaje que utiliza el estándar internacional OCL, para la modelación de la programación Orientada a Objetos.
- No modifica la semántica del lenguaje base UML, sino que trabaja en estereotipos restrictivos, por lo que a su vez produce modificaciones

descriptivas y decorativas en la representación de los componentes del lenguaje base. [6]

ApEM-L incorpora una nueva vista: la vista de presentación, como aporte principal a la modelación de este tipo de aplicaciones con dos nuevos diagramas: el diagrama de estructura de presentación y el diagrama de estructura de navegación

Se escogió el lenguaje de modelado ApEM-L porque se expresa de manera tal que puede ser utilizado con cualquier proceso de desarrollo aunque se sugiere que sea utilizado por procesos de desarrollos iterativos e incrementales, que permitan la modelación de sistemas orientados a objetos. También utiliza los estereotipos restrictivos lo cual nos permite una mejor comprensión de lo que se está modelando.

1.9 Herramientas de Modelado Visual.

Modelado Visual es el proceso que permite representar el sistema software, permitiendo resaltar los detalles más importantes de la aplicación.

Para este modelado visual se utilizan las herramientas Case ya que son el mejor método para el análisis y soluciones de software, han venido a mejorar los aspectos claves en el desarrollo de los sistemas de información, han sido creadas para la automatización de procesos de análisis, diseño e implementación, brindando un sin número de componentes que hacen que los proyectos sean cada día más eficientes para los usuarios finales.

Desde que se crearon éstas herramientas (1984) hasta la actualidad, las CASE cuentan con una credibilidad y exactitud que tienen un reconocimiento universal, siendo usadas por cualquier analista y programador que busca un resultado óptimo y eficaz, para cada uno de sus procesos.

Se realizará un estudio de algunas herramientas CASE para la selección de la más adecuada para el modelado visual del módulo Temas.

1.9.1 Rational Rose.

Rational es actualmente conocida como una familia de software de IBM para el levantamiento de requerimientos, diseño, construcción, pruebas y administración de

proyectos en el proceso desarrollo de software y sus productos están centrados en la metodología del Proceso Racional Unificado o RUP (Rational Unified Process). Entre los productos más conocidos tenemos a: Rational Application Developer, Rational Software Architect (Posee la funcionalidad del producto antes conocido como Rational Rose) y Rational Portafolio Manager.

Con esta herramienta podemos utilizar cuatro tipos de modelos para realizar el diseño del sistema, utilizando una vista estática y otra dinámica de dichos modelos, uno lógico y otro físico. Permite crear y refinar, logrando así un modelo completo que representa el dominio del problema y el sistema de software. También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades.

Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML y proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa podemos obtener información sobre su diseño.

Para la utilización de esta herramienta resulta algo complicado debido a que propone el empleo del proceso de desarrollo unificado, el cual es altamente complejo e incorpora algunas funcionalidades a través de otras aplicaciones pero no las integra.

1.9.2 Visual Paradigm 2.3.

Visual Paradigm 2.3 es una herramienta CASE que da soporte a UML, esta ofrece un entorno donde se pueden crear diagramas que permiten ver, desde diferentes perspectivas, el sistema; por ejemplo: diagrama de componentes, de despliegue, de actividad, entre otros. El diseño es centrado en los casos de uso enfocado al negocio, esto hace que se genere un software de mayor calidad. Tiene disponibilidad en múltiples plataformas y en múltiples versiones.

Es una herramienta amigable para el usuario, puede ser usada en varios idiomas y cada componente utilizado en el diagrama que se esté creando, sugiere nuevos posibles componentes a utilizar, por lo que ya no es necesario localizarlos en la barra y así se crea fácilmente cualquier tipo de diagrama. Tiene un amplio número de estereotipos que proporciona la creación de diagramas de fácil entendimiento, además de que estos diagramas te los organiza automáticamente. Puedes obtener un

Diagrama de Clases del Diseño a partir de un Diagrama de Entidad Relación y viceversa.

Brinda la posibilidad de documentar todo el trabajo, en formato PDF o en forma de un sitio WEB, sin necesidad de utilizar herramientas externas. Una vez que los diagramas han sido confeccionados, es posible generar documentos que combinan texto e imágenes obtenidas a partir de los diagramas de manera muy sencilla, incluso Visual Paradigm genera toda la documentación de lo que se ha hecho hasta el momento, cumpliendo con estándares establecidos. Brinda también la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir del código. Entre los lenguajes conocidos por Visual Paradigm para la ingeniería inversa, se encuentran los siguientes:

- Java Source
- C++ Source
- PHP 5.0 Source
- XML

Visual Paradigm 2.3 se integra fácilmente con varios Ambientes de Desarrollo Integrados (IDEs), lo que permite pasar del modelado al código y viceversa, como son: Visual Studio, Eclipse y JBuilder.

Visual Paradigm 2.3 brinda la posibilidad de generar código ORM a partir de un Diagrama de Entidad Relación una Base de Datos Relacional, y el código necesario para acceder a esta base de datos utilizando Java, PHP, C# o Enterprise Object Framework. Para lograrlo, Visual Paradigm se conecta fácilmente a Servidores de Bases de Datos de varios vendedores, como por ejemplo: MySql y Oracle.

Apoya todo lo básico en cuanto a artefactos generados en las etapas de definición de requerimientos y de especificación de componentes. Tiene apoyo adicional en cuanto a generación de artefactos automáticamente. También tiene disponibilidad en múltiples plataformas, también soporta el ciclo de vida completo del desarrollo de software: análisis, diseño, implementación, pruebas y despliegue. Ayuda a una más rápida construcción de aplicaciones de calidad, mejores y más baratas. Permite dibujar todos los tipos de diagramas y genera documentación.

Conclusiones

En este capítulo se realizó un profundo estudio para la selección de las tecnologías a utilizar en el proceso de análisis y diseño del módulo Temas de las nuevas aplicaciones de la Colección Multisaber, como son la metodología, el lenguaje de modelado y la herramienta para modelar el problema.

Después de realizado todo el proceso de investigación se decidió que el análisis y diseño de la aplicación se desarrollará con la metodología RUP que es la más usada en estos momentos, el lenguaje de modelado ApEM – L porque brinda una mejor manera de modelar las multimedias educativas utilizando los prototipos restrictivos, y como herramienta CASE Visual Paradigm.

2. CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA.

Introducción

En el presente capítulo se realiza un breve estudio sobre el Modelo de Dominio y se hará una descripción del sistema propuesto. Se presenta el Modelo de Dominio, definiéndose los principales conceptos de la aplicación, una descripción de cada pantalla, la *Vista de Gestión del Modelo* y los diagramas de *Estructura de Presentación* y el de *Estructura de Navegación*, donde el Diagramas de Estructura de Navegación representa las clases que posibilitan la navegación entre las distintas clases que componen la aplicación y el Diagrama de Estructura de Presentación incorpora un conjunto de estereotipos que permitirán a los diseñadores gráficos decidir donde y como serán en términos visuales dichos elementos. Además se realizará un diagrama de clase por cada vista que establece el patrón arquitectónico MVC-E para la concepción del diseño de las aplicaciones educativas. Posteriormente se realizaran los diagramas de interacción que representan el intercambio de mensajes de los objetos para dar cumplimiento a sus responsabilidades. También en este capítulo se expondrá el diagrama de despliegue que describe la distribución física del sistema.

2.1 Modelo del Dominio.

Un modelo conceptual explica los conceptos más significativos en un dominio del problema, identificando los atributos y las asociaciones. Un modelo conceptual representa cosas del mundo real, no componentes del software. Por lo que se ve representado mediante un grupo de diagramas de estructura estática donde no se define ninguna operación. En estos diagramas se muestran conceptos (objetos) y asociaciones entre conceptos (relaciones). Es decir que el modelo conceptual es una herramienta de comunicación, con la cual se intenta comprender los conceptos importantes y sus relaciones. Es también útil para transmitir este conocimiento a otros.

Por el poco nivel de estructuración que presenta el entorno de la aplicación que se esta estudiando, se puede decir que no funciona como un negocio por lo que se

propone modelar el entorno como un dominio, ya que permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema. Esto ayuda a los usuarios a utilizar un vocabulario común para poder entender el contexto en que se sitúa el sistema.

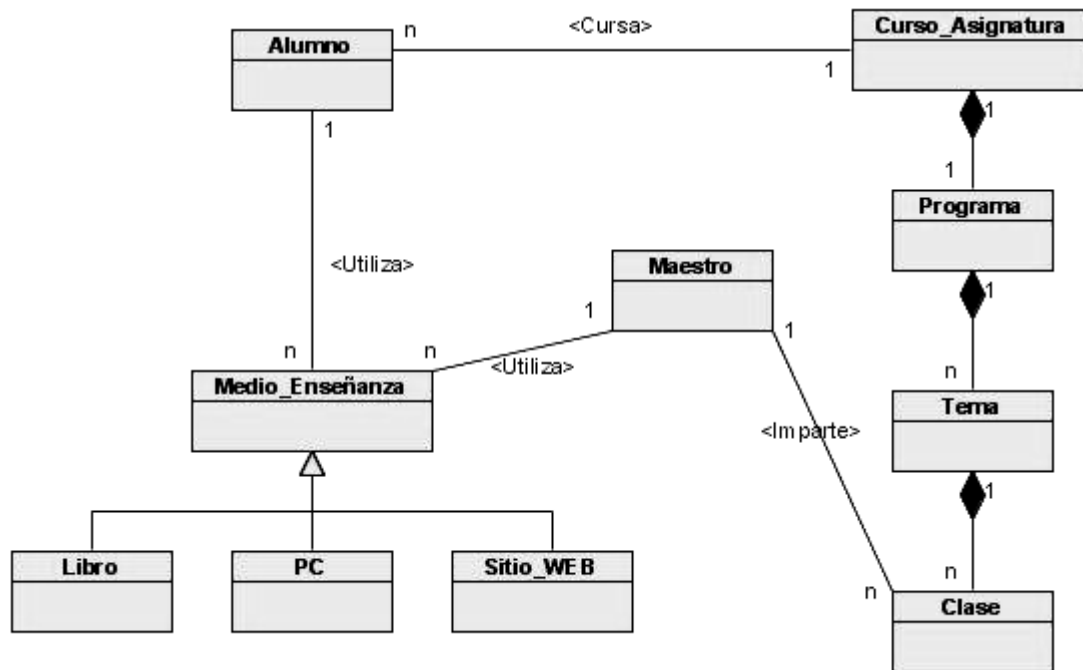


Figure 2. Modelo Conceptual.

- **Curso Asignatura:** Se le llama al curso de cada una de las disciplinas que se imparte por asignatura.
- **Medio de enseñanza:** Será todo objeto que sea utilizado en el proceso de enseñanza-aprendizaje de las asignaturas.
- **Clase:** Será el turno de educación física donde tiene lugar el proceso de enseñanza-aprendizaje de la natación.
- **Profesor:** Será toda aquella persona que imparte clases para enseñar.
- **Estudiante:** Será toda aquella persona que reciba clases para aprender.
- **Libro:** Aquel libro con contenido sobre las asignaturas y sea utilizado como medio de enseñanza.

- **Sitio web:** Será aquel sitio web con contenido sobre las asignaturas y sea utilizado como medio de enseñanza.
- **Programa:** Será la guía que contiene estructuradamente el contenido del curso.
- **Tema:** Se le denomina a la estructura básica del contenido del curso.

2.1.1 Requisitos Funcionales.

Como punto de partida para determinar que debe lograr el sistema, se han identificado los siguientes requerimientos funcionales que representan las condiciones y capacidades que el software debe cumplir.

RF1. Cargar presentación del módulo Tema.

RF1.1 Permitir el acceso a las imágenes que representa cada tema.

RF1.2 Dar acceso a los títulos.

RF2. Cargar presentación del Título.

RF2.1 Mostrar un área de texto.

RF2.2 Mostrar toldo.

RF2.3 Mostrar una imagen de fondo relacionada con el tema.

RF3. Permitir salir de cada pantalla.

RF4. Permitir ir a inicio.

RF5. Permitir hacer búsquedas.

RF6. Tener ayuda.

RF7. Tener música.

RF7.1 Conectar música.

RF7.2 Desconectar música.

2.1.2 Requisitos No Funcionales.

Lograr cumplir las expectativas del usuario, implica, además de lograr que el software funcione, que lo haga de la mejor manera posible; en base a los requisitos no funcionales que especifican a continuación.

Tabla 1. Requerimiento de Software.

Software	Requisito
Sistema Operativo	Linux: Ubuntu 7.0

Tabla 2. Requerimientos de Hardware.

Hardware	Requisito
Procesador	Pentium 233 MHz (<i>Recomendado:</i> Pentium 500MHz o mejor)
Soporte de video	Que admita resolución de la menos 800x600 y 24 bits
Memoria	64 MB RAM (<i>Recomendado:</i> 128 MB RAM o mejor)
Disco Duro	1 GB
Dispositivo	<ul style="list-style-type: none"> • De red de al menos 10 megabits • De audio
Lector de CD	

Tabla 3. Restricciones en el diseño y la implementación.

Lenguajes y Herramientas	Requisito
Lenguaje de Programación.	Java Script.
Herramienta de Desarrollo.	CMS Joomla!

Requerimiento de apariencia o interfaz externa: El módulo debe tener un diseño sencillo, permitiendo la utilización de la aplicación sin necesidad de mucho conocimiento informático.

Requerimientos de Seguridad: Puede interactuar cualquier tipo de usuario al módulo.

Requerimientos de Usabilidad: Deberán tener al menos un conocimiento básico del uso de los cómputos, los usuarios que utilizarán el sistema, y también un conocimiento previo de los sistemas operativos visuales.

Requerimientos de Rendimiento.

- El tiempo de visualización de las medias no deberá exceder los 5s.
- El tiempo de ejecución de un elemento interactivo no deberá exceder los 5s.

Requerimientos de Portabilidad: El sistema debe ser multiplataforma.

Requerimientos de Ayuda y documentación en línea: El sistema contará con un fichero texto nombrado Léeme.txt que brindará las especificaciones de las necesidades de software antes de instalar el producto y los pasos a seguir para la instalación de este.

Requerimientos Políticos-Culturales: La aplicación responderá a las políticas por las que se rige el proyecto MULTISABER y que fueron dictadas por el Ministerio de Ciencia y Tecnología.

2.2 Descripción del sistema propuesto.

2.2.1 Vistas de Presentación por interfaz de comunicación con el usuario.

La interfaz de comunicación con el usuario es la forma en que los usuarios pueden comunicarse con el cómputo, y comprende todos los puntos de contacto entre el usuario y el equipo.

- Interfaces de Comunicación con el Usuario *Temas*.
 - Vista Temas.
 - Vista Títulos.
- Interfaces de Comunicación con el Usuario *Contenidos*.
 - Vista Contenidos.

2.2.2 Actores del sistema.

Usuario es la generalización de los actores: estudiantes o profesores. Realiza las actividades que son comunes a todos ellos.

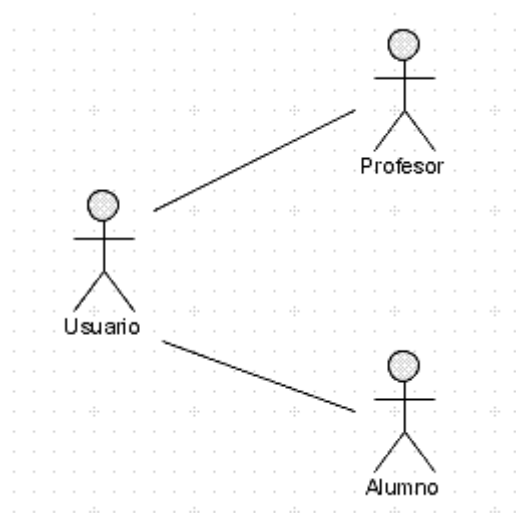


Figure 3. Representación de los Actores del Sistema.

2.2.3 Descripción Textual de las Vistas de Presentación.

Cada vista ha sido descrita textualmente en formato de la Vista de Presentación y se ha utilizado diagramas de estructura de navegación y el diagrama de estructura de presentación.

Tabla 4. Descripción Textual de la Vista de Presentación Temas.

Descripción Textual de la Vista de Presentación <i>Temas</i> .	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario de la aplicación, seleccionar entre los temas de la aplicación en su interfaz de comunicación con el usuario.
Objetivos pedagógicos	Permitir al usuario dominar las opciones de los temas dentro de la aplicación.
Resumen	La vista se inicia cuando el usuario carga el módulo, se muestran las imágenes que representan los temas y termina cuando el usuario selecciona alguna de estas.
Vistas asociadas	Vista "Títulos".
Referencias	RF1,RF3 ,RF4, RF5, RF6 y RF7

Tipo de Media	Nombre	Descripción	Estado
Imagen	-	Imágenes que representan a los temas donde se nota la relación con este.	Existente.
Video o Animación	-	Animación en donde la mascota se mueve abriendo los ojos y mostrando una expresión de alegría.	Existente.
	-	Animación en donde cada imagen se engrandece o cambia de estado mostrando el vínculo.	Existente.
Sonido	-	Canciones infantiles que se utiliza para simbolizar la selección.	Existente.
Texto	-	-	-
Reglas pedagógicas			
-			

Tabla 5. Descripción Textual de la Vista de Presentación Títulos.

Descripción Textual de la Vista de Presentación <i>Título</i>.	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario de la aplicación, seleccionar entre los títulos de la aplicación en su interfaz de comunicación con el usuario.
Objetivos pedagógicos	Permitir al usuario dominar las opciones de los títulos dentro de la aplicación.
Resumen	La vista se inicia cuando el usuario da clic en una de las imágenes (elemento interactivo) que representa cada tema, se muestra una interfaz de comunicación con el usuario, mostrando todos los títulos y termina cuando el usuario selecciona uno de estos.
Vistas asociadas	Vista " <i>Contenidos</i> ".
Referencias	RF1,RF3 ,RF4, RF5, RF6 y RF7
Precondiciones	-

Poscondiciones		Vistas de presentación visualizada de acuerdo a la elección del usuario.	
Curso Normal de los Eventos			
Acciones del Actor		Respuesta del Sistema	
<p>1. El usuario da clic en el elemento interactivo.</p> <p>5. El usuario selecciona uno de los títulos.</p>		<p>2. Se reproduce sonido de acuerdo con la interfaz de comunicación seleccionada.</p> <p>3. Se muestran todos los títulos del tema especificado.</p> <p>4. Se muestran las opciones de Salir (común) y, Sonido (común), Acerca de (común), Ayuda (común), Inicio (común), Búsqueda (común).</p> <p>6. Se muestra la Vista del contenido y se culmina la Vista.</p>	
		<p>3.1 Cada título será un elemento interactivo que dará acceso a los contenidos de cada uno.</p> <p>6.1 Se subraya el título seleccionado.</p>	
Cursos Alternos de los Eventos			
Acción		Curso Alterno	
-		-	
Prioridad		Crítica	
Mejoras		-	
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	-	-	-
Vídeo o Animación	-	-	-

Sonido	-	Canciones infantiles que se utiliza para simbolizar la selección.	Existente.
Texto	-	Textos que definen los Títulos.	Existente.
Reglas pedagógicas			
-			

2.2.4 Vista de Gestión del Modelo.

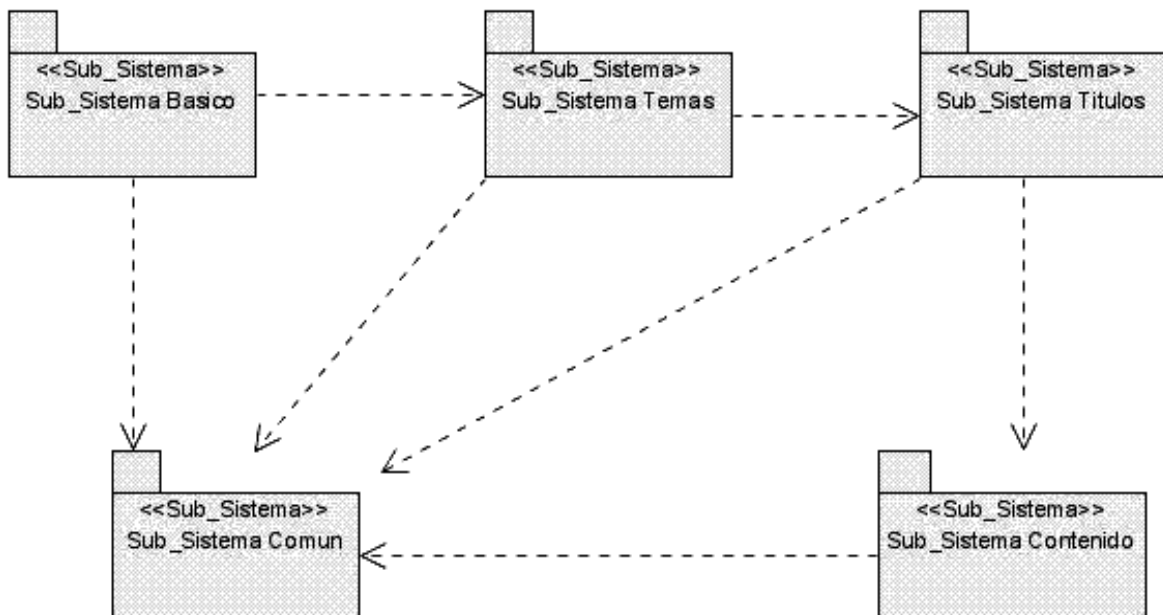


Figure 4. Representación de la Vista de Gestión del Modelo.

El subsistema **Básico** esta compuesto por todos los subsistemas en su conjunto.

El subsistema **Común** esta compuesto por varios elementos interactivos como son: Acerca de, Ayuda, Inicio, Imprimir, Buscar, Música y Salir. En el elemento Acerca de se pueden encontrar las vistas de presentación Patrocinadores, Licencia y Créditos. En Ayuda esta la vista de presentación Ayuda. Inicio es un elemento interactivo que lleva a la interfaz de comunicación con el usuario principal de la aplicación. Buscar ofrece un servicio de búsqueda compuesta por las vistas de presentación Búsqueda y Resultados. Música es un elemento interactivo para desconectar o conectar la

música. Salir para salir de la aplicación tiene como vista de presentación Confirmar. Imprimir tiene dos vistas Confirmar e Imprimir.

El subsistema **Temas** tiene como vista de presentación la vista Temas que aparece cuando se carga el módulo, está conformada por los elementos interactivos que conllevan a los títulos de cada tema.

El subsistema **Títulos** tiene la vista de presentación Títulos que aparece cuando se da clic en los temas.

El subsistema **Contenido** que esta compuesto por un texto y un proyector, el proyector puede mostrar una imagen o reproducir un video.

2.2.5 Diagrama de Estructura de Presentación por vista.

En el contexto de ApEM – L no se trabajará la distribución espacial, pues las herramientas CASE actuales que soportan el lenguaje no lo permiten tal y como pretendemos dibujarlo hoy en día. Se trabajará la estructura que tendrán esas interfaces de comunicación; este diagrama incorpora un conjunto de estereotipos que permitirán a los diseñadores gráficos decidir donde y como serán dichos elementos. Se realiza una clasificación definiendo dos nuevos tipos de clases: *la clase Estáticos* y *la clase Interacción*.

La clase *Estáticos* agrupará los componentes que solo tiene como función visualizar información, pero que no permiten interacción con el usuario.

La clase *Interacción* es todo lo contrario a la anterior, los cuales serán los elementos de la vista que permiten interacción del usuario con el sistema informático modelado.

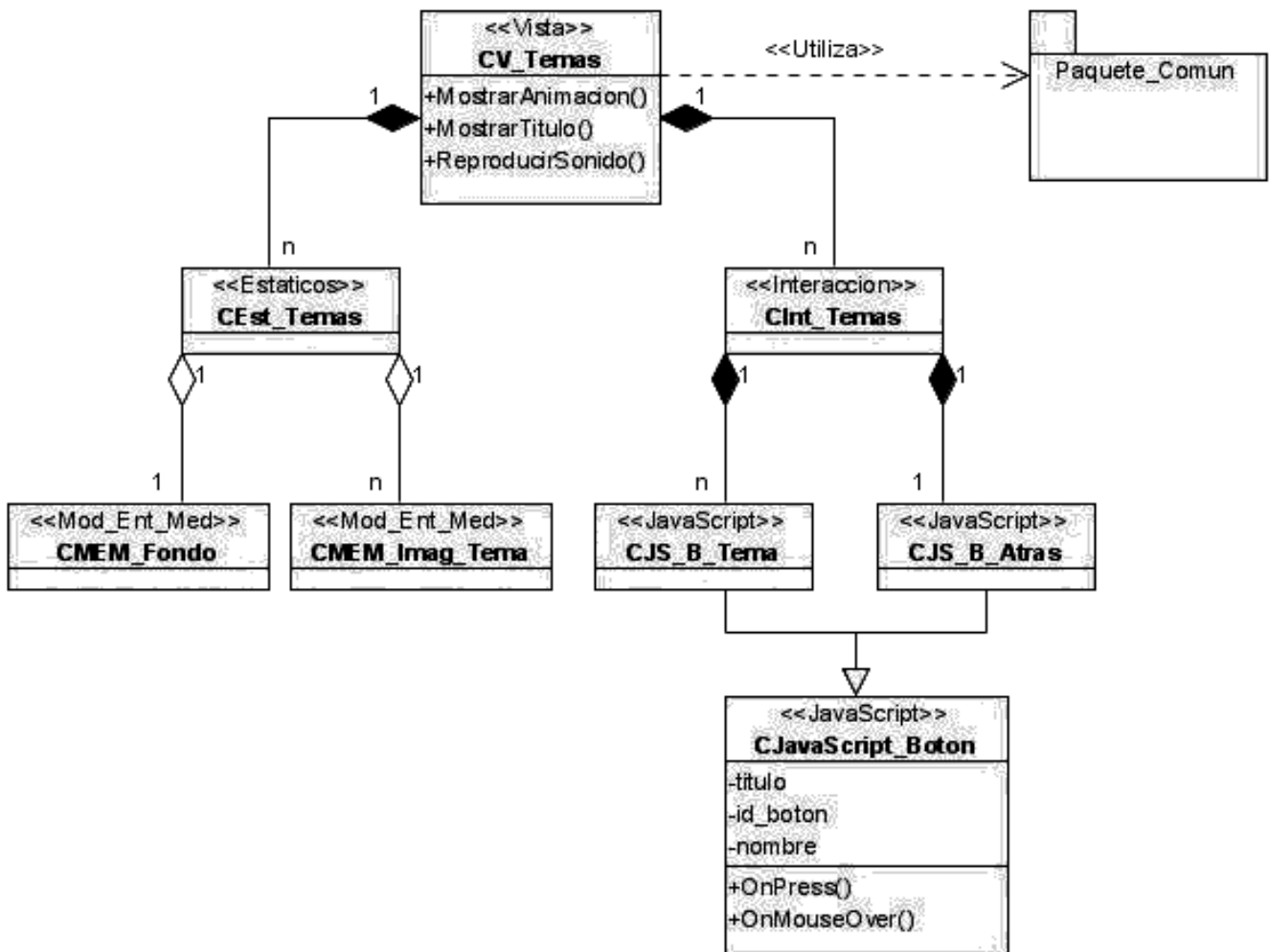


Figure 5. Diagrama de Estructura de Presentación de la vista Temas.

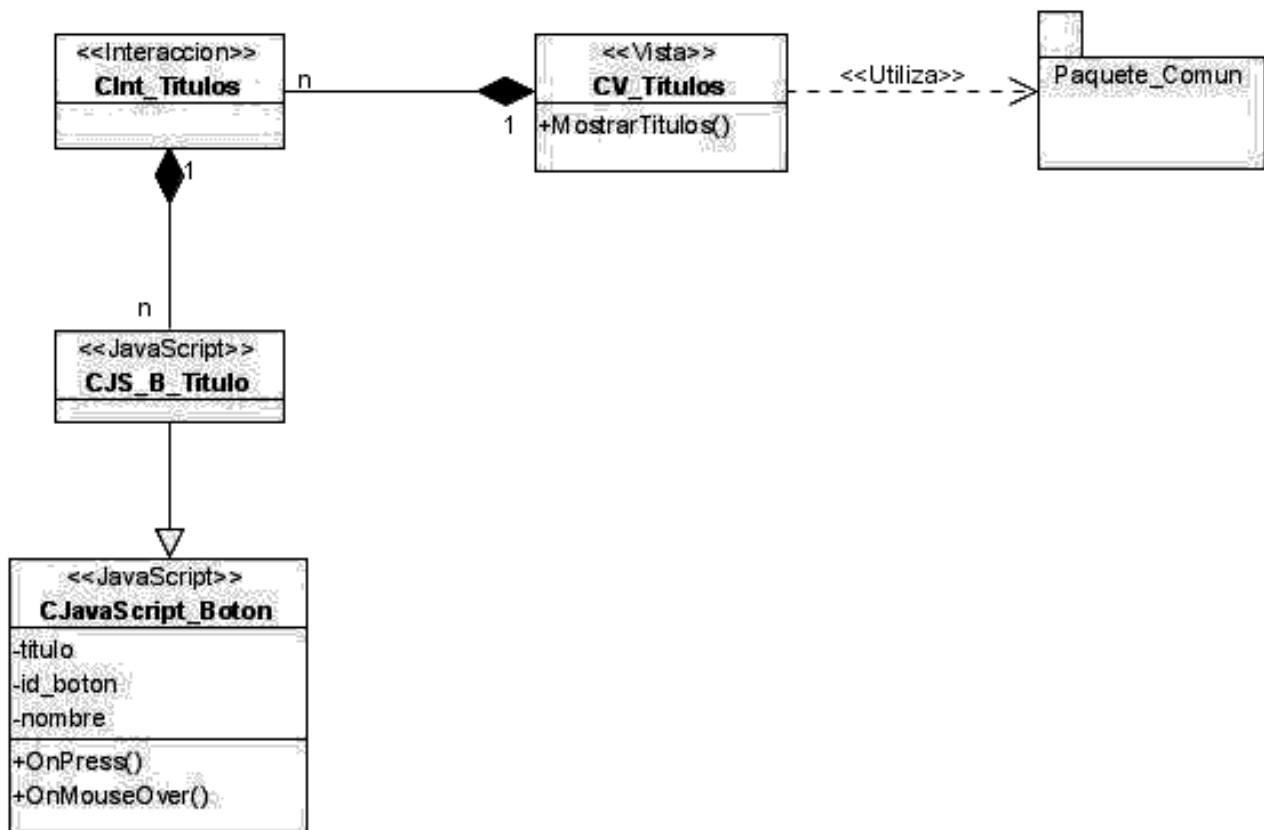


Figure 6. Diagramas de Estructura de Presentación de la vista Título.

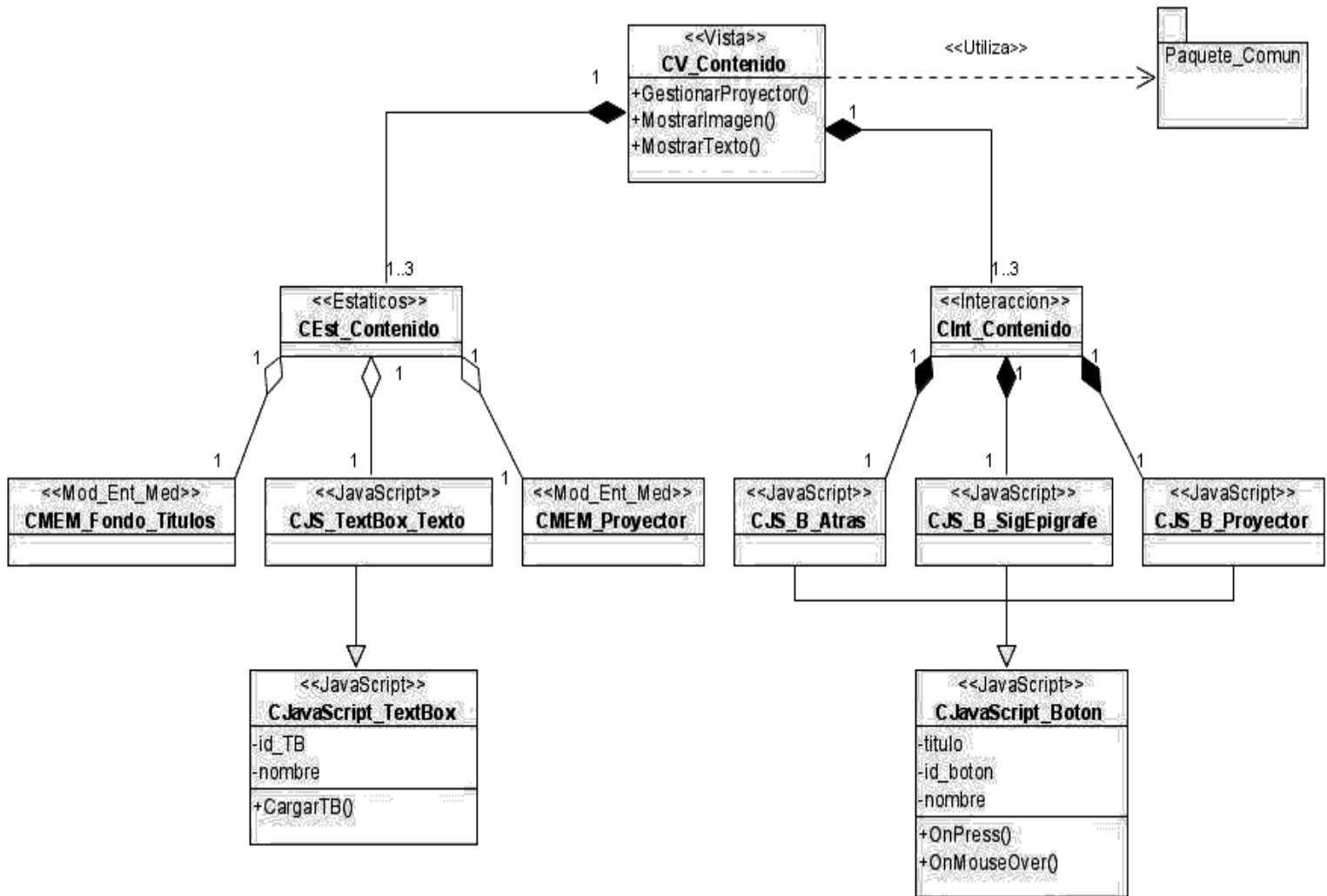


Figure 7. Diagramas de Estructura de Presentación de la vista Contenidos.

2.2.6 Diagrama de Estructura de Navegación.

Debido a la poca eficiencia del lenguaje base UML para modelar los sistemas multimedia se ha definido el *Diagrama de Estructura de Navegación* que representa las clases que posibilitan la navegación entre las distintas clases que componen la aplicación. Para el desarrollo de este diagrama se han definido varias clases, como son: *clase menú*, *clase índice*, *clase consulta* y *clase botón*. Además de las ya definidas *modelo-entidad-media texto* y *modelo-entidad-media imagen*.

La *clase menú* es el elemento donde se puede llegar a otras diferentes clases vistas con las cuales se conecta este menú, pues contiene una lista de las opciones de movimiento siguiente, pero los valores de esas opciones no tiene que llevar

necesariamente a una clase vista directamente, sino que puede ser a través de otro tipo de clase de navegación.

La *clase consulta* permite el enlace con una clase vista a y través de un valor directo.

La *clase índice* denota el valor de direccionamiento hacia una clase vista a partir de una opción determinada.

La *clase botón* denota la existencia de un elemento interactivo del tipo botón para producir un camino en la navegación hacia una clase vista, y utilizando como intermediarias a clases de tipo consulta o índice.

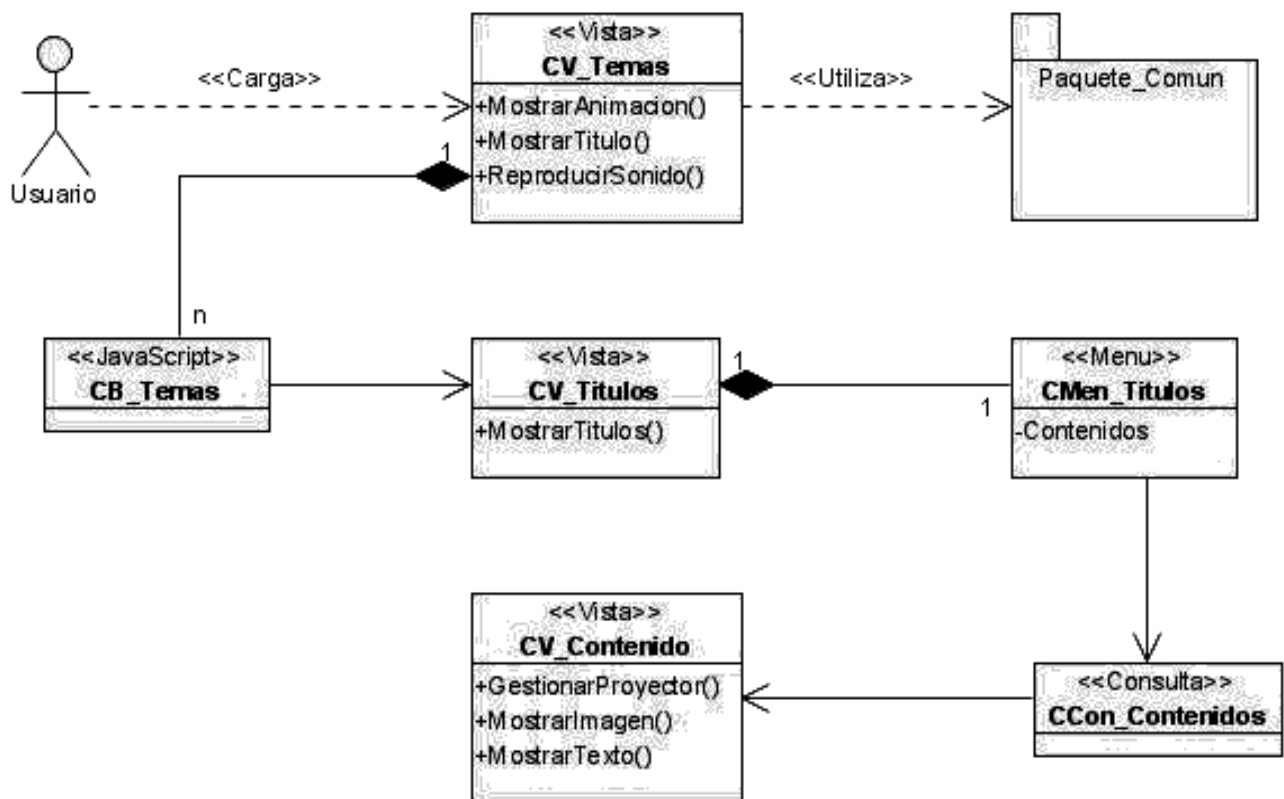


Figure 8. Diagrama de Estructura de Navegación.

2.2.7 Diagrama de Clases.

El diagrama de clases establece el patrón arquitectónico MVC-E para la concepción del diseño de las aplicaciones educativas. En el diagrama se utiliza la semántica y los estereotipos restrictivos y descriptivos para las clases asociadas a las tecnologías multimedia e hipertexto. También organiza en secciones para la representación

lógica de los distintos tipos de clases, incorporando las clases abstractas del modelo conceptual.

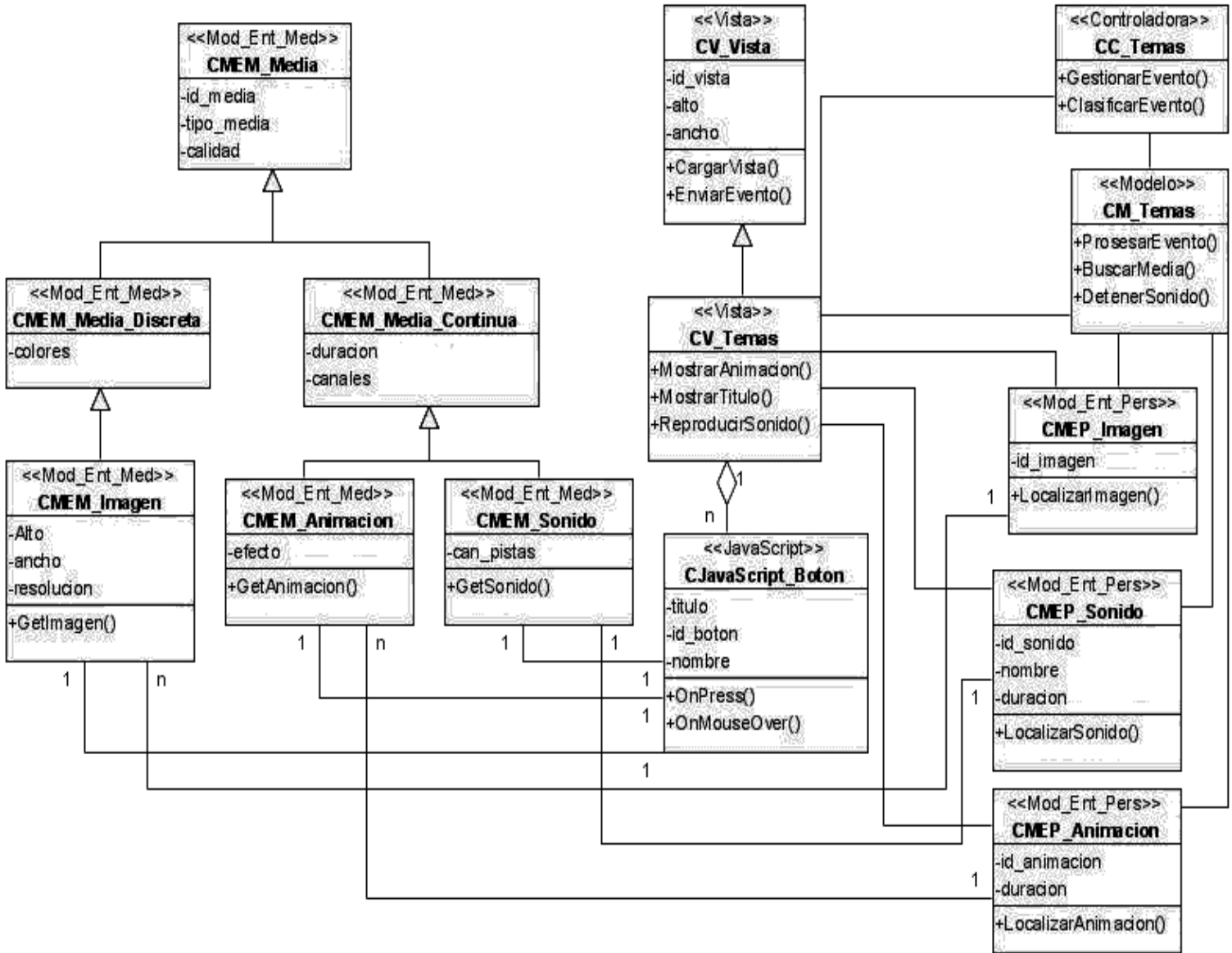


Figure 9. Diagrama de Clases del diseño de la vista Temas.

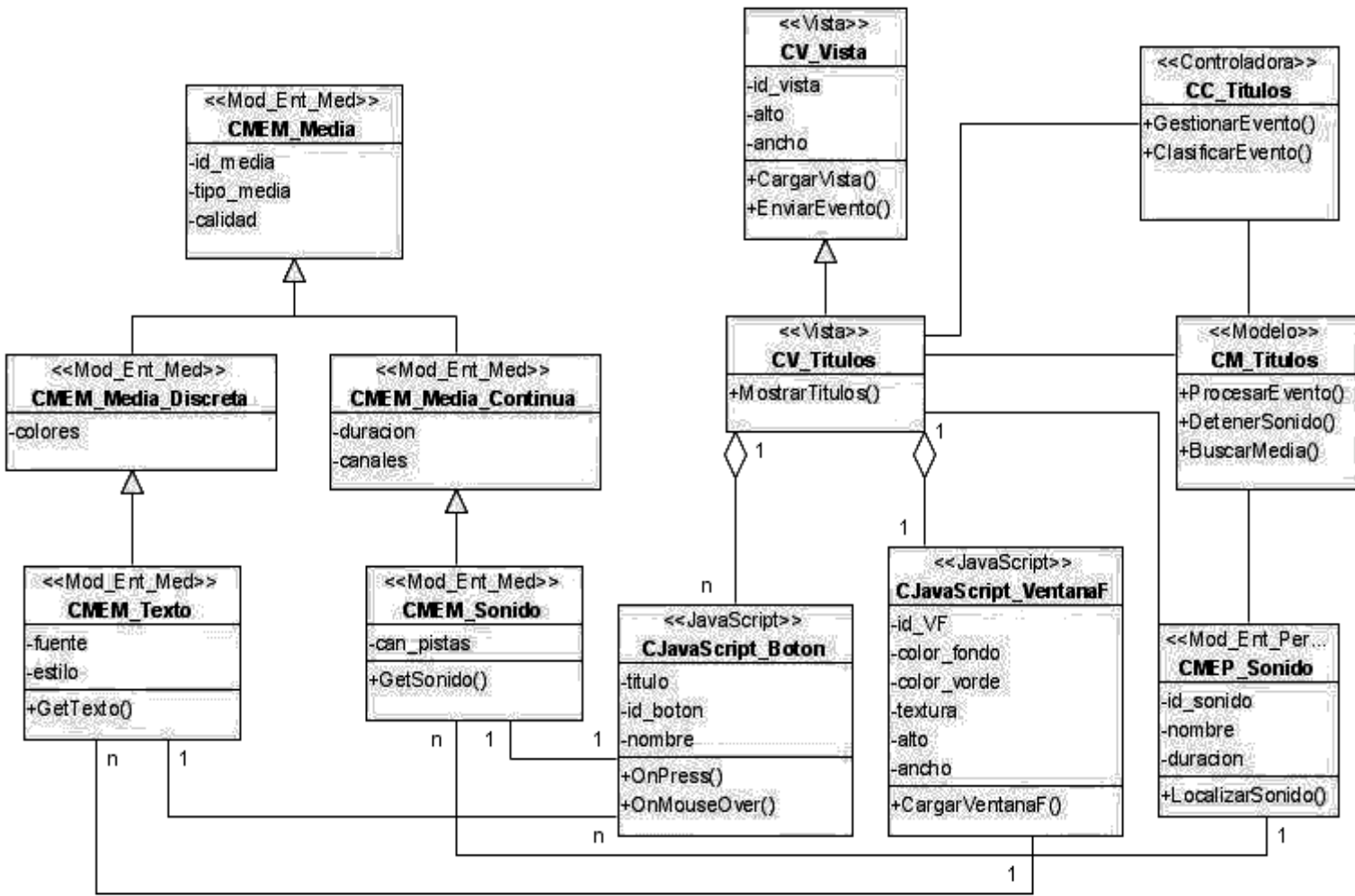


Figure 10. Diagrama de Clases del diseño de la vista Titulo.

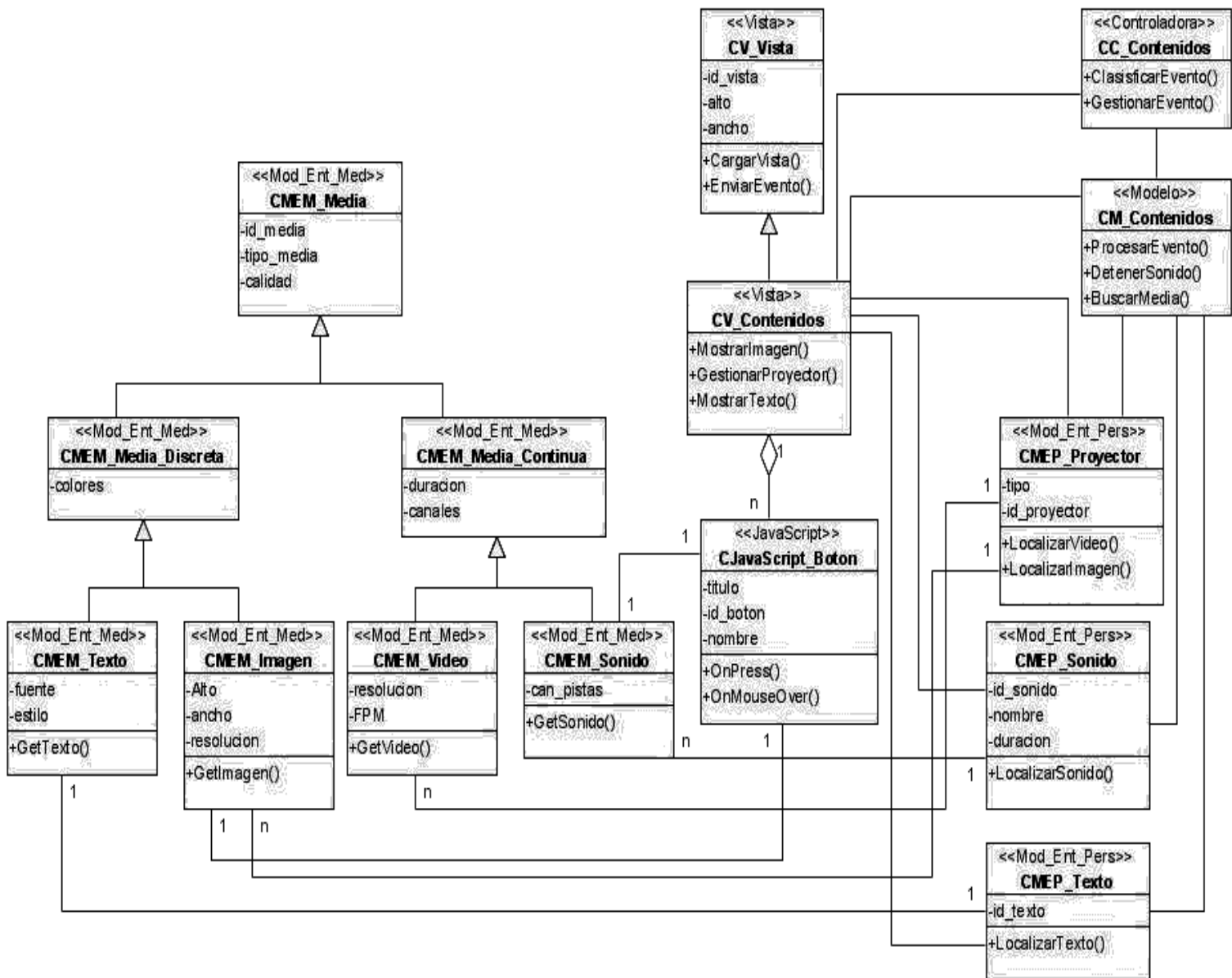


Figure 11. Diagrama de Clases del diseño de la vista Contenidos.

2.2.8 Diagrama de Interacción.

Los diagramas de interacción representan la manera en la que los objetos de la aplicación intercambian mensajes para darle cumplimiento a sus responsabilidades, en ApEM-L al diagrama de secuencia se le ha añadido un estereotipo descriptivo para denotar el tiempo como variable para este tipo de aplicaciones.

Las clases que se utilizan para modelar este tipo de diagramas se estereotipan de la siguiente manera:

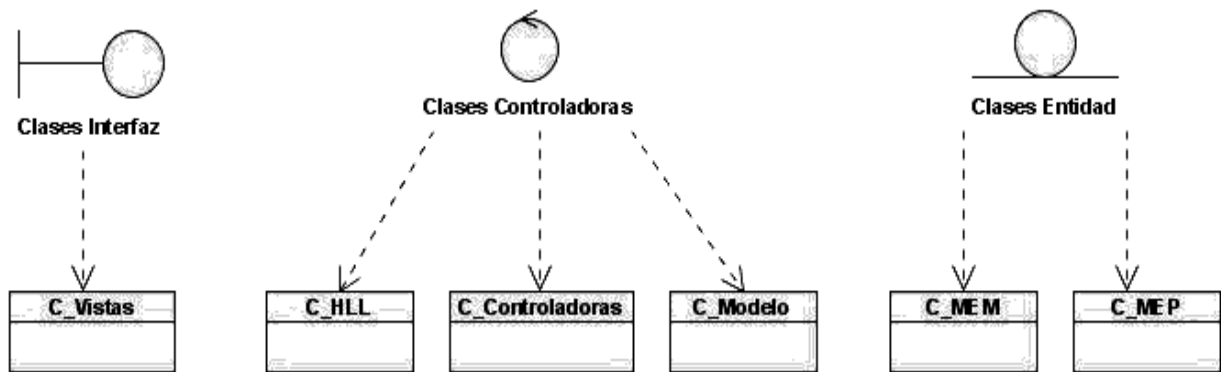


Figure 12. Estereotipos de las clases interfaz, controladoras y entidad.

Las clases *Interfaz* serían las clases *Vistas*, que recibirán las peticiones del usuario al sistema y mostrarán los mensajes de salida o respuestas.

Las clases *Controladoras* estarían compuestas por las clases *Controladoras* que estas gestionarán las peticiones y las muestras de las respuestas y también por la clases *Modelos* que contendrán la lógica del negocio para el procesamiento de la información, y por último las clases *HLL* que son las seleccionadas para la programación del software.

Las clases *Entidad* serían las clases *Modelo Entidad* tanto, las que son para el procesamiento de la información *persistente*, como las correspondientes a la representación del árbol jerárquico de *medias*.

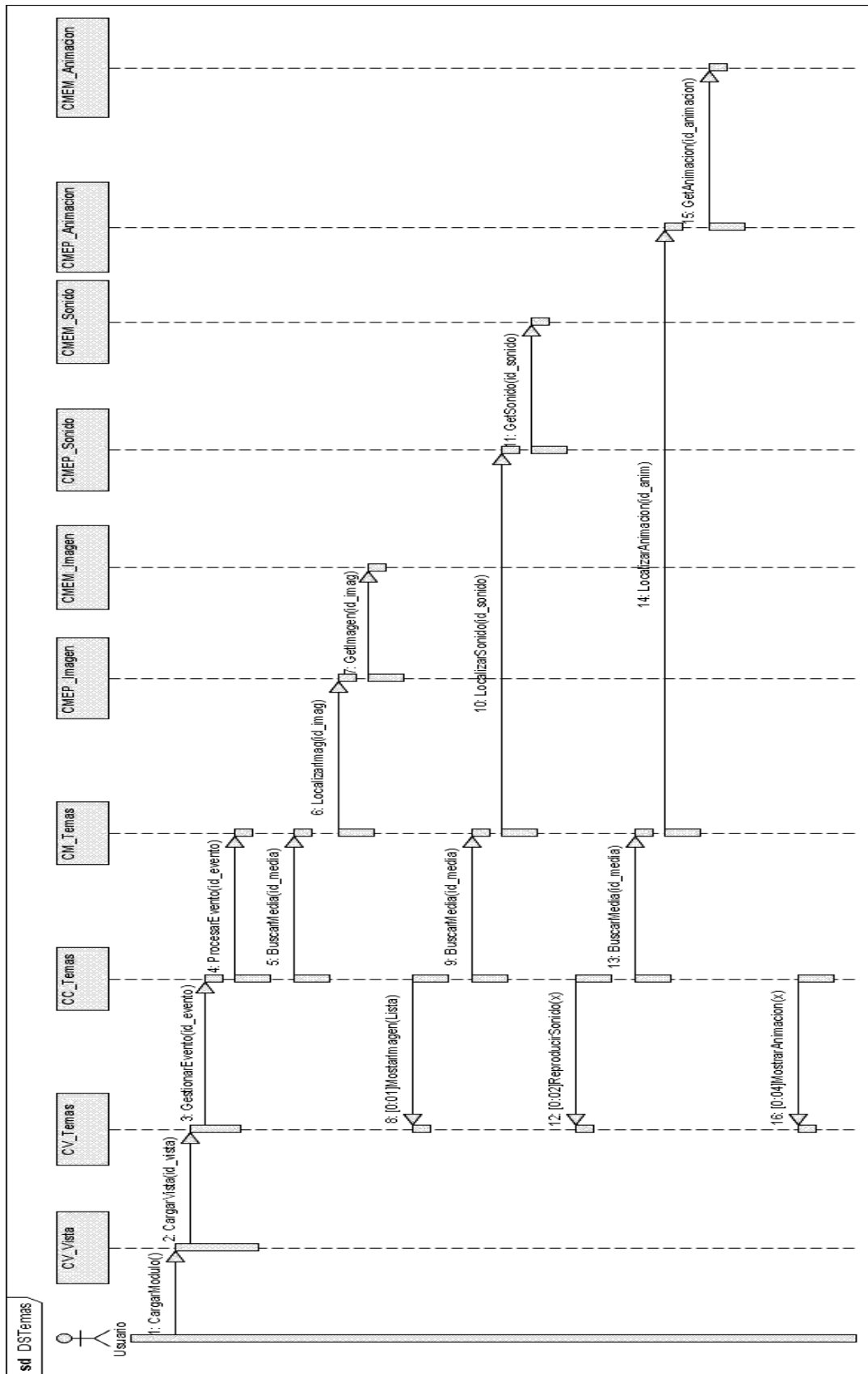


Figure 13. Diagrama de Secuencia de la vista Temas.

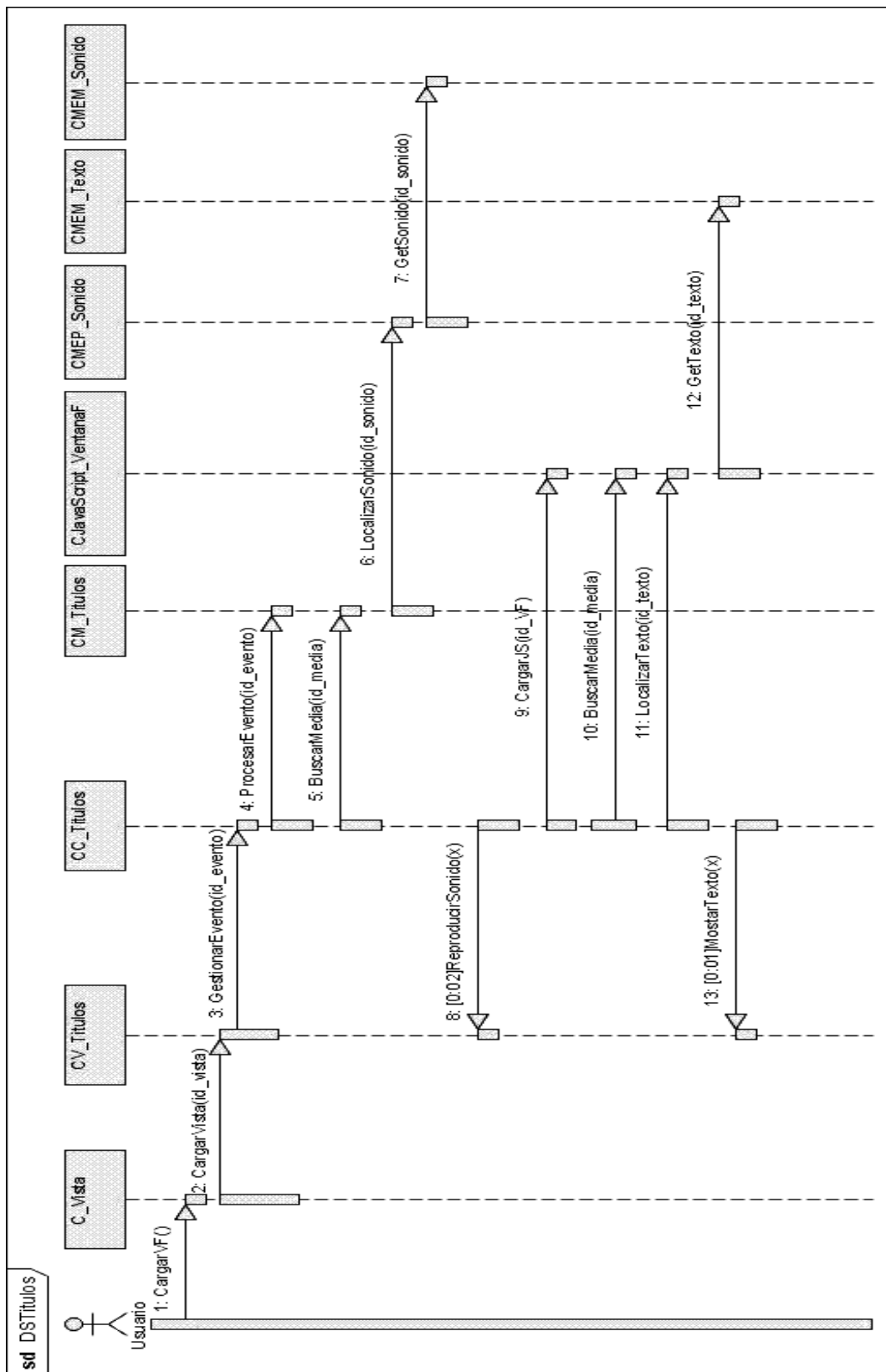


Figure 14. Diagrama de Secuencia de la vista Títulos.

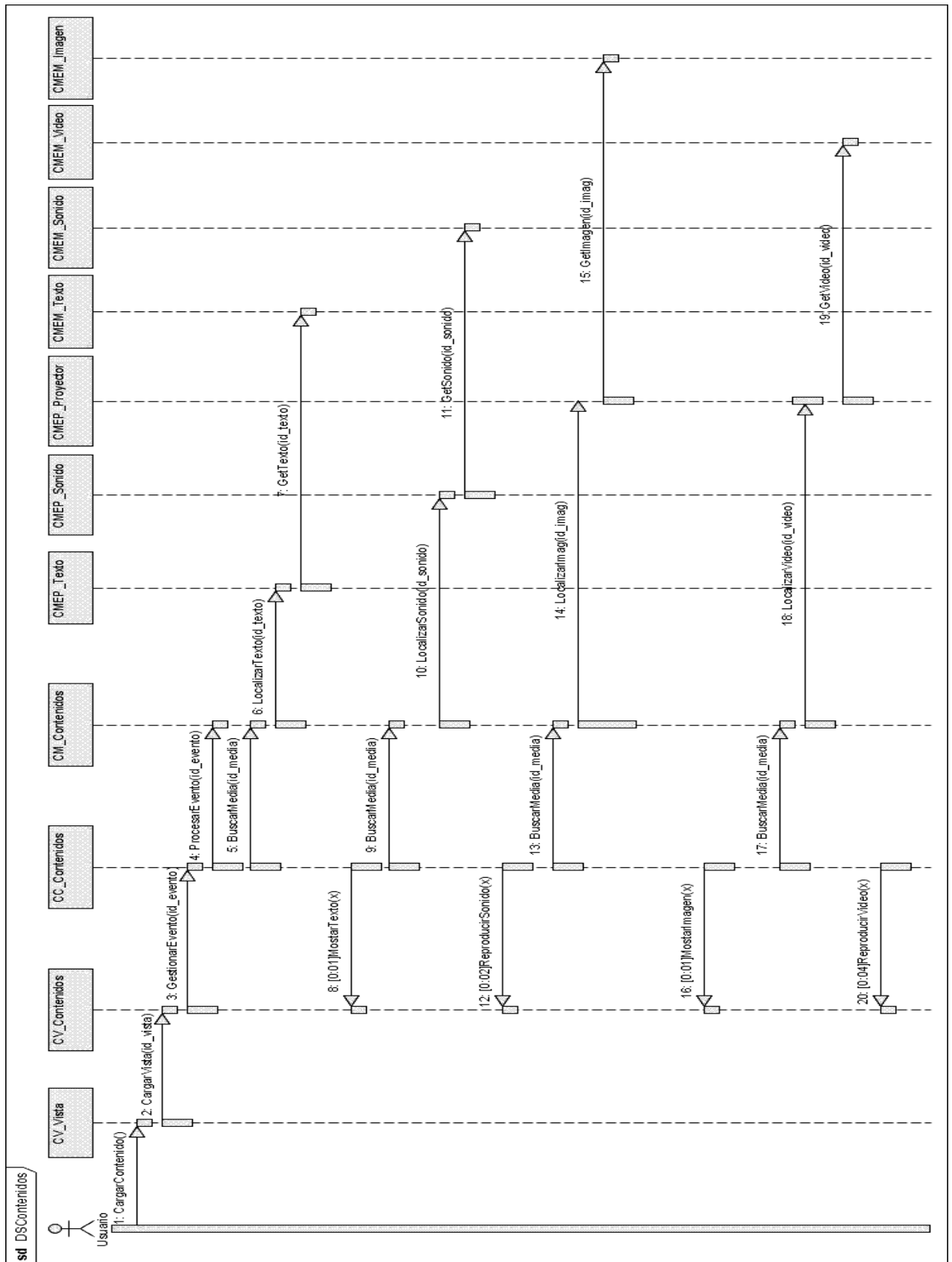


Figure 15. Diagrama de Secuencia de la vista Contenidos.

2.2.9 Diagrama de Despliegue.

El diagrama de despliegue describe la distribución física del sistema. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware.

En el diagrama de despliegue que se presenta, se especifica la existencia por separado de los servidores web y los servidores de bases de datos, que serán definidos en ubicación y en número por la plataforma de la educación venezolana. Presenta además, el nodo correspondiente a las PC Usuario, correspondiente a las máquinas desde las cuales accederán los estudiantes y profesores de las escuelas primarias venezolanas. El dispositivo conectado a la PC Usuario es decir la Impresora responde a las necesidades de estos equipos para el desarrollo exitoso del software educativo.

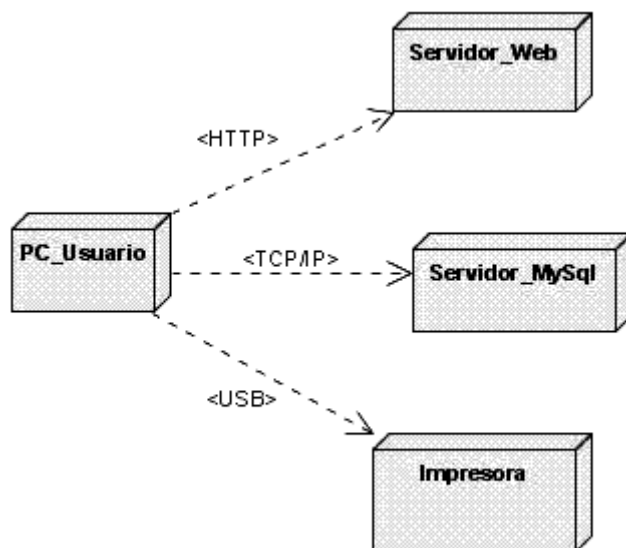


Figure 16. Diagrama de Despliegue de la aplicación.

Conclusiones

El capítulo muestra las características que debe presentar el software para cumplir los requerimientos especificados por el cliente. Para describir el contexto en que se desenvuelve se realiza el modelo del dominio. Se definieron los requisitos funcionales y no funcionales que debe cumplir la aplicación, se elaboró la vista de gestión del modelo que mejora la comprensión del sistema. Se hace una descripción textual por cada vista presentación y se presentó el análisis y diseño del sistema, los diagramas

que fueron necesarios realizar con la ayuda del visual Paradigm para obtener una mejor visión de la solución del problema presentado.

3. ESTUDIO DE FACTIBILIDAD.

Introducción

Uno de los aspectos más importantes para realizar un software es conocer los beneficios que este aportaría, pues se tomarían decisiones en cuanto si su implementación sería factible o no. Con el estudio de factibilidad se determinan parámetros que ayudan a la planificación de la realización de la aplicación multimedia como son el personal necesario para el desarrollo de la aplicación multimedia y también el tiempo que esta requiere según tamaño y complejidad.

En este capítulo se realizará el estudio para determinar si es viable o no desarrollar el software multimedia que se propone como solución al problema planteado. Para ello se utiliza el método COCOMO II (Modelo Constructivo de Costes) que en estos momentos es el preferido para la estimación del esfuerzo cuando no se tiene información histórica a la cual recurrir.

3.1 Estimación del Costo del Proyecto.

El método COCOMO II consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar o la cantidad de líneas de código (SLOC, Source Lines Of Code) estimados para un proyecto. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo (cost drivers) que influyen en el esfuerzo requerido para el desarrollo del software.

Tabla 6. Consultas Externas.

Nombre de la salida externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (Simple, Media y Compleja)
Buscar datos.	1	1	Simple

Tabla 7. Salidas Externas del módulo Temas.

Nombre de la salida externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (Simple, Media y Compleja)
Mostrar las imágenes que representan los temas.	1	1	Simple
Mostrar animación de las imágenes	1	1	Simple
Mostrar la vista de los títulos.	1	1	Simple
Mostrar los títulos de cada tema.	1	1	Simple
Mostrar contenido de cada titulo.	1	1	Simple
Mostrar el texto de cada contenido	1	1	Simple
Mostrar el proyector de cada contenido	1	1	Simple
Mostrar el video o la imagen de cada contenido	1	1	Simple
Permitir al usuario ir a inicio.	1	1	Simple
Permitir el acceso al resto de los módulos en todo momento.	1	1	Simple
Permitir al usuario volver a la pantalla de selección de los temas.	1	1	Simple
Permitir salir en cualquier momento.	1	1	Simple

Tabla 8. Consultas Externas del módulo Temas.

Elementos de Información	Simples		Medios		Complejos		Subtotal de Puntos de Función
	Cuenta	Peso	Cuenta	Peso	Cuenta	Peso	
Consultas externas	1	4	0	4	0	7	4
Salidas externas	12	4	0	5	0	7	48
Subtotal (UFP)							52

3.1.1 Puntos de Función.

Tabla 9. Puntos de Función.

Características	Valor
Puntos de función desajustados	52
Lenguaje	Java Script 2.0
Instrucciones fuentes por puntos de función	48
Instrucciones fuentes	2496

Resultados:

Puntos de función del proyecto: 52

Total de instrucciones fuentes: 2496

Miles de instrucciones fuentes (*MF*): 2.5 (aproximadamente)

Tabla 10. Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

Multiplicadores de esfuerzo	Valor
RCPX (RELY, DATA, CLPX, DOCU)	1.00

RUSE	1.00
PDIF (TIME, STOR, PVOL)	1.00
PERS (ACAP,PCAP, PCON)	1.00
PREX (APEX, PLEX, LTEX)	1.00
FCIL (TOOL, SITE)	1.00
SCED	1.00

Tabla 11. Factores de Escala.

Factores de Escala	Valor
PREC	3.72
FLEX	3.04
RESL	5.65
TEAM	2.19
PMAT	4.68

Multiplicadores de esfuerzo:

7

$$EM = \prod_{i=1}^7 E_{mi} = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED = 1.000$$

i=1

Factores de escala:

5

$$SF = \sum_{i=1}^5 S_{Fi} = PREC + FLEX + RESL + TEAM + PMAT = 19.28$$

i=1

Valores calibrados:

$$A = 2.94; B = 0.91; C = 3.67; D = 0.24$$

$$E = B + 0.01 * SF$$

$$F = D + 0.2 * (E - B)$$

$$E = 0.91 + 0.01 * 19.28$$

$$F = 0.24 + 0.2 * (1.1 - 0.91)$$

$$E = 1.1028 \approx 1.1$$

$$F = 0.278 \approx 0.3$$

Esfuerzo:

7

$$PM = A * (2.5)^E * \prod_{i=1}^n Em_i$$

$$PM = 2.94 * (2.5)^{1.1} * 1.000 = 8.055 \approx 8.1$$

$$PM \approx 8 \text{ Hombres mes.}$$

Cálculo del tiempo:

$$TDEV = C * (PM)^F$$

$$TDEV = 3.67 * (8)^{0.28} = 6.56$$

$$TDEV \approx 6.6 \text{ meses}$$

Cálculo de la cantidad de hombres:

$$CH = PM / TDEV$$

$$CH = 8 / 6.6 = 1.212$$

CH \approx 1 hombres

Al ser 3 la cantidad real de hombres ha desarrollar la aplicación al reajustar el tiempo de desarrollo según la cantidad de hombres, resultó un tiempo de **2.7 meses**.

Salario promedio:

Para determinar el salario promedio se tiene en cuenta que los desarrolladores del sistema pueden ser ingenieros recién graduados además de estudiantes de tercer año pertenecientes a la UCI (Universidad de las Ciencias Informáticas) por lo que se toma como salario promedio: **\$212.0**

Costo:

$$\text{CHM} = 3 * \text{Salario Promedio}$$

$$\text{Costo} = \text{CHM} * \text{PM}$$

$$\text{CHM} = 3 * \$212$$

$$\text{Costo} = \$636 * 8$$

$$\text{CHM} = \$636$$

$$\text{Costo} = \$5088$$

3.2 Beneficios tangibles e intangibles.

3.2.1 Beneficios Tangibles.

Con el desarrollo del proceso que se propone: Análisis y diseño del módulo Temas de la Colección Multisaber, traerá consigo numerosos beneficios tangibles se señalan los siguientes:

- Se contará con la documentación del módulo para la creación de aplicaciones interactiva, dinámico y de interfaz agradable que posibilitará a los alumnos perfeccionar su aprendizaje.
- Se podrá integrar la tecnología multimedia a la educación ya que se contará con los medios necesarios.

3.2.2 Beneficios intangibles.

Este módulo se convierte en unos de los elementos más importantes para la composición de los nuevos productos de la Colección Multisaber, que a su vez estos productos se convierten en aplicaciones de gran utilidad para todos los alumnos de las escuelas primarias de Venezuela, puesto que la misma permite el estudio de diferentes temas de una forma amigable, permitiendo a su vez, un aumento de la calidad de la enseñanza y un

incremento en el nivel cultural de estos alumnos. Aumenta la calidad de la presentación de los contenidos a los alumnos, lo que logrará una mayor motivación de estos.

Conclusiones

En este capítulo se han analizado muchos aspectos que intervienen de una u otra manera a la hora de realizar este y otros proyectos con tecnología multimedia. Como resultado de este análisis se han puntualizado claramente una serie de datos muy importantes como el costo total que requiere la realización del módulo Temas de los nuevos productos de la Colección Multisaber resultando este de \$ 5151, otro dato significativo que se ha detallado es el tiempo de desarrollo de dicho producto siendo de 2.7 meses de duración, así como el esfuerzo requerido que es de 8 Hombres mes, sin dejar de mencionar los recursos materiales y humanos necesarios, y los beneficios que se obtienen con el software ya sean tangibles o intangibles.

CONCLUSIONES GENERALES

El presente trabajo de tesis demostró la necesidad de la documentación del análisis y diseño del módulo Temas de los nuevos productos de la Colección Multisaber a software libre. Por los beneficios que este trae para las escuelas primarias de la educación venezolana, se realizó un estudio de la documentación existente de los productos anteriores de la colección Multisaber, se hizo un estudio de las tecnologías existentes para el desarrollo de multimedias, se seleccionan RUP como metodología para el desarrollo del software y Apem – L como lenguaje de modelado, la cual conllevó a la documentación del proceso, para que las creaciones o migraciones futuras de productos similares cuenten con una documentación que le servirá de guía para el proceso de desarrollo.

RECOMENDACIONES

- Realizar la documentación de los restantes flujos de trabajo que propone la metodología utilizada (RUP), llegando a la implementación de las funcionalidades que han propuesto para el Módulo Temas.
- Seguir investigando las necesidades de las escuelas venezolanas, para incorporar nuevas funcionalidades al módulo, así como determinar nuevas formas de presentar los contenidos.
- Utilizar la documentación del proceso análisis y diseño del módulo Temas para la creación o migración de productos similares.

REFERENCIAS BIBLIOGRÁFICAS

- [1] **2008**. Colección Multisaber. Disponible en:
<http://www.insted.rimed.cu/multisaber.asp>
- [2] **2008**. Las 4 libertades del “Software Libre”. Disponible en:
http://www.aniz.es/wordpress/?page_id=103
- [3] **Caballero, I. 2008?** *Una Herramienta CASE para ADOO: Visual Paradigm.* (Clase de ingeniería del software 3 de la Escuela Superior de Informática de Ciudad Real). Disponible en:
http://alarcos.infcr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf
- [4] **Ciudad, A. F. R. 2007.** *ApEM – L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI.* Tesis para optar por el título de Máster en Informática Aplicada.
- [5] **Díaz, C.C. 1994,** LA TECNOLOGIA MULTIMEDIA: Una nueva Tecnología de Comunicación e Información. Características, concepciones y aplicaciones. Disponible en: <http://iteso.mx/~carlosc/pagina/documentos/multidef.htm#inicio>.
- [6] **Graells, P. M. 2004.** CLASIFICACION DE LOS MATERIALES DIDÁCTICOS MULTIMEDIA. Disponible en: <http://dewey.uab.es/pmarques/funcion.htm>
- [7] **GUERRERO, L. A.** *Análisis y Diseño Orientado a Objetos. Modelo conceptual.* Disponible en: <http://www.dcc.uchile.cl/~luguerre/cc40b/clase4.html>.
- [8] **Mora, A.J.H.** *Historia de multimedia.* [pagina web] Disponible en:
<http://www.monografias.com/trabajos7/mult/mult2.shtml>
- [9] Multimedia, Disponible en: <http://es.wikipedia.org/wiki/Multimedia>
- [10] **Salinas, D.J.M. 1996,** MULTIMEDIA EN LOS PROCESOS DE ENSEÑANZA-APRENDIZAJE: ELEMENTOS DE DISCUSION. Disponible en:
<http://www.uib.es/depart/gte/multimedia.html>.
- [11] **SANCHEZ, M. A. M. 2007.** Metodologías de Desarrollo de Software Disponible en:

<http://64.233.167.104/search?q=cache:lzgHcDIV3CEJ:www.informatizate.net/articulos/pdfs>

[12] **Sanchez, M.2007**. *¿Se puede producir multimedia con software libre?*(Sitio personal de Marko Sanchez). Disponible en:

<http://markosanchez.com/%c2%bfse-puede-producir-multimedia-con-software-libre/>

[13] **Turner, M. 2006**. Microsoft Solutions Framework Essentials, Microsoft Press.

[14] **Wikipedia, 2007** Proceso Unificado de Rational. Disponible en:

http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational

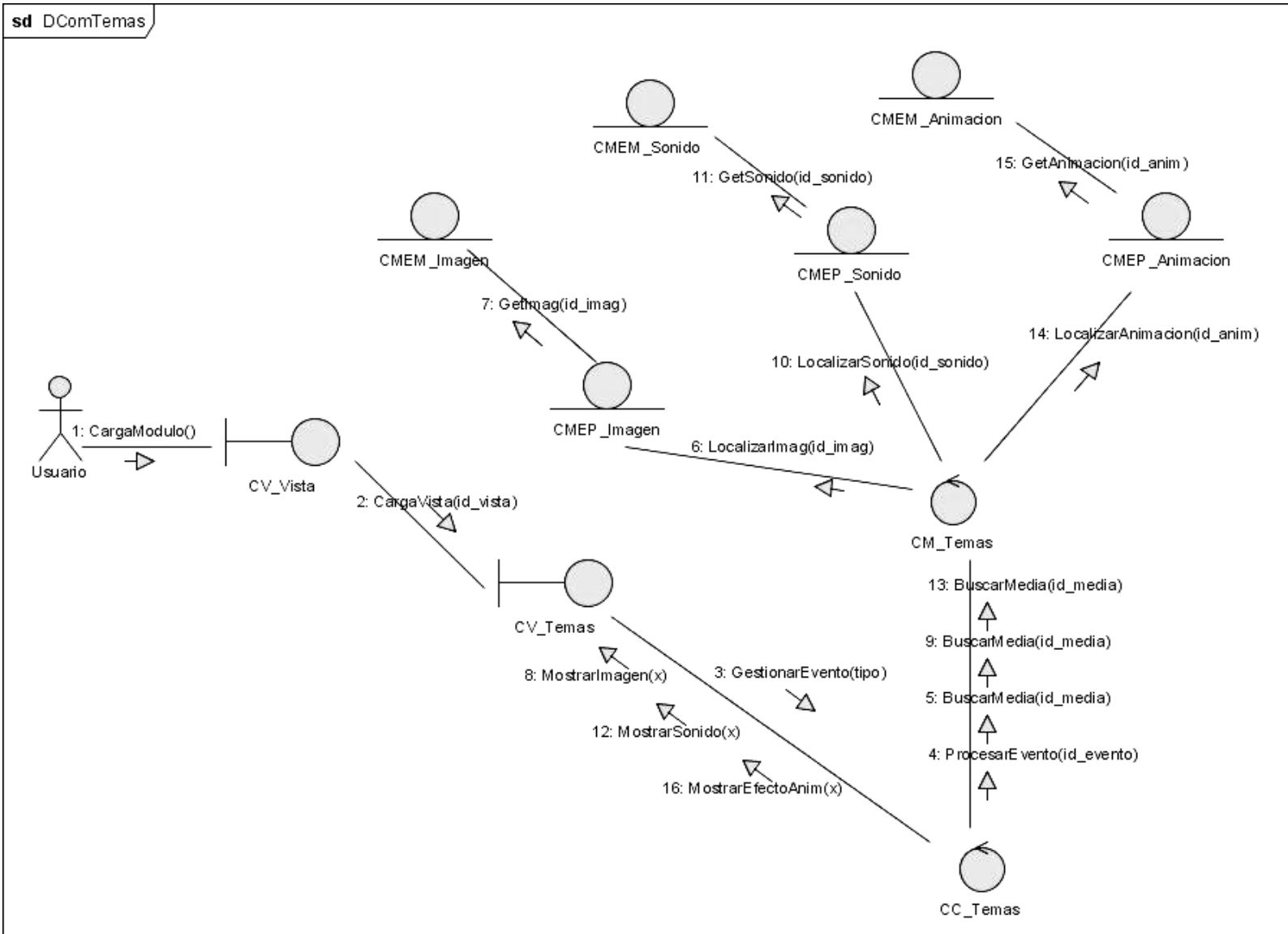
BIBLIOGRAFÍA

- [1] **2008.** Colección Multisaber. Disponible en:
<http://www.insted.rimed.cu/multisaber.asp>
- [2] **2008.** Las 4 libertades del “Software Libre”. Disponible en:
http://www.aniz.es/wordpress/?page_id=103
- [3] **Caballero, I. 2008?** *Una Herramienta CASE para ADOO: Visual Paradigm. (Clase de ingeniería del software 3 de la Escuela Superior de Informática de Ciudad Real).* Disponible en:
http://alarcos.infcr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf
- [4] **Ciudad, A. F. R. 2007.** *ApEM – L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI.* Tesis para optar por el título de Máster en Informática Aplicada.
- [5] **Díaz, C.C. 1994,** LA TECNOLOGIA MULTIMEDIA: Una nueva Tecnología de Comunicación e Información. Características, concepciones y aplicaciones. Disponible en: <http://iteso.mx/~carlosc/pagina/documentos/multidef.htm#inicio>.
- [6] **Gracia, R.E. 2007.** Multimedia educativa: entre fantasía y realidad [artículo]; Disponible en :
http://www.sappiens.net/html/ejemplos/economia1/sappiens/comunidades/ejemplo_seconomia1nsf/unids/Multimedia%20educativa_%20entre%20fantas%EDa%20y%20realidad/87786C549EC27BA941256FAF0060448E2d8e.html?opendocument
- [7] **GRAELLS, D. P. M., 2007.** Multimedia educativo: clasificación, funciones, ventajas, e inconvenientes. Disponible en:
<http://dewey.uab.es/pmarques/funcion.htm>
- [8] **Graells, P. M. 2004.** CLASIFICACION DE LOS MATERIALES DIDÁCTICOS MULTIMEDIA. Disponible en: <http://dewey.uab.es/pmarques/funcion.htm>
- [9] **GUERRERO, L. A.** *Análisis y Diseño Orientado a Objetos. Modelo conceptual.* Disponible en: <http://www.dcc.uchile.cl/~luguerre/cc40b/clase4.html>.

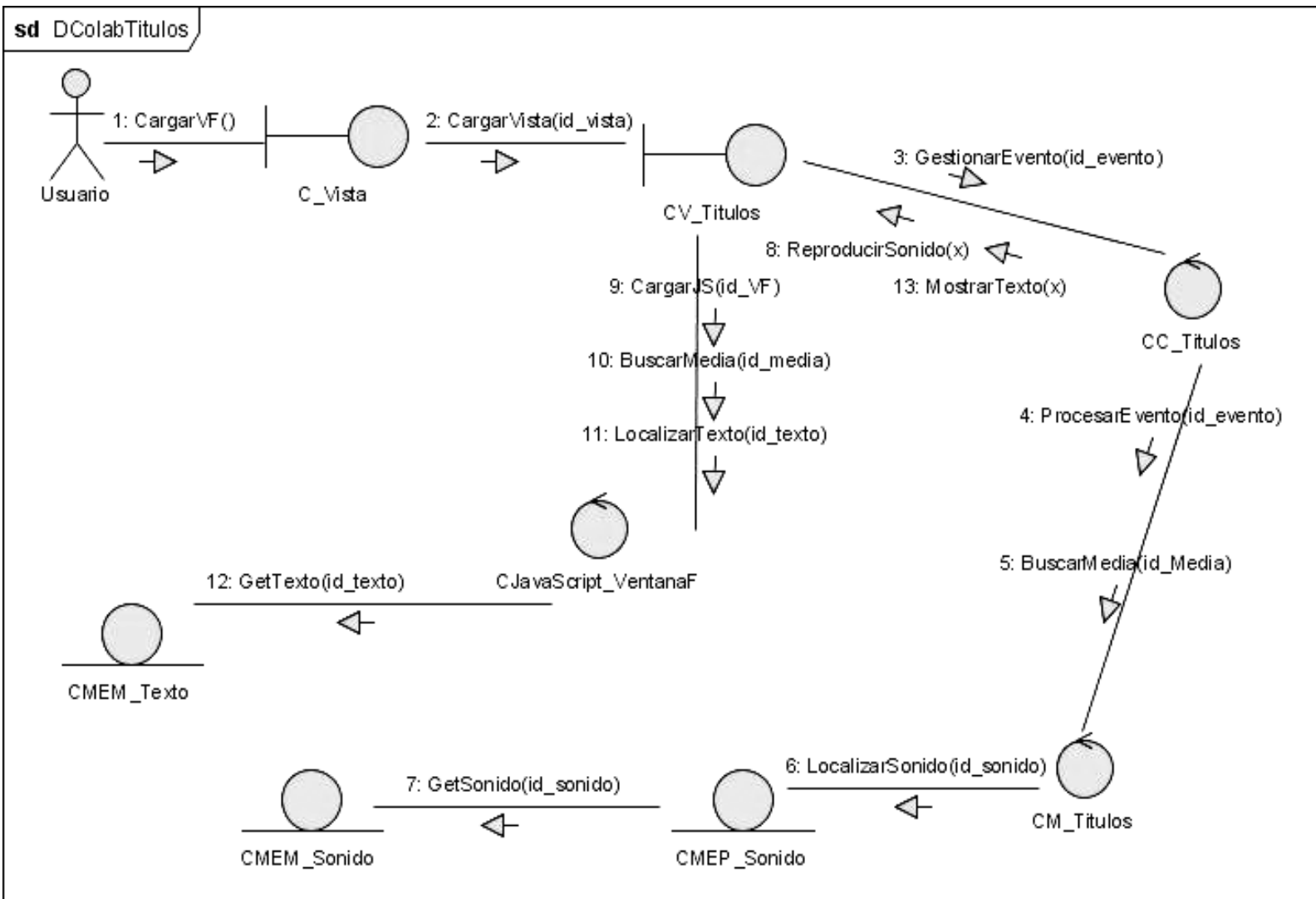
- [10] **JACOBSON, I.; G. BOOCH, et al, 1977.** El proceso Unificado del Software. La Habana Cuba.
- [11] **Mora, A.J.H.** *Historia de multimedia.* [pagina web] Disponible en: <http://www.monografias.com/trabajos7/mult/mult2.shtml>
- [12] Multimedia, Disponible en: <http://es.wikipedia.org/wiki/Multimedia>
- [13] **PRESSMAN, R. S., 1977.** Ingenieria del Software. La Habana, Cuba.
- [14] **Salinas, D.J.M. 1996,** MULTIMEDIA EN LOS PROCESOS DE ENSEÑANZA-APRENDIZAJE: ELEMENTOS DE DISCUSION. Disponible en: <http://www.uib.es/depart/gte/multimedia.html>.
- [15] **SANCHEZ, M. A. M. 2007.** Metodologías de Desarrollo de Software Disponible en: <http://64.233.167.104/search?q=cache:lzgHcDIV3CEJ:www.informatizate.net/articulos/pdfs>
- [16] **Sanchez, M.2007.** *¿Se puede producir multimedia con software libre?(Sitio personal de Marko Sanchez).* Disponible en: <http://markosanchez.com/%c2%bfse-puede-producir-multimedia-con-software-libre/>
- [17] **Turner, M. 2006.** Microsoft Solutions Framework Essentials, Microsoft Press.
- [18] **Wikipedia, 2007** Proceso Unificado de Rational. Disponible en: http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational
- [19] **WIKIPEDIA. , 2007.** Modelo Vista Controlador.

ANEXOS

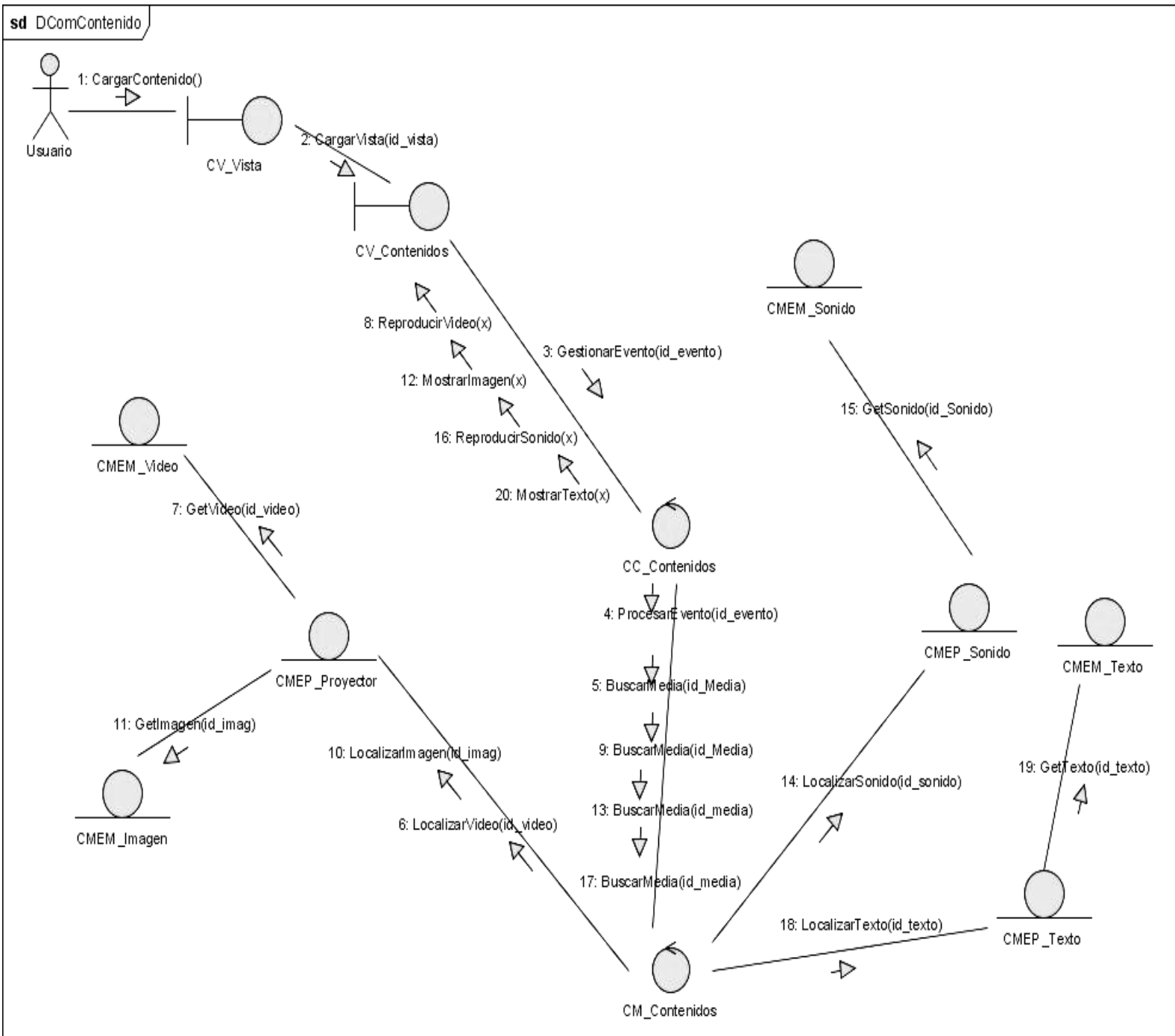
Anexo I. Diagrama de colaboración de la vista Temas.



Anexo II. Diagrama de colaboración de la vista Títulos.



Anexo III. Diagrama de colaboración de la vista Contenidos.



GLOSARIO DE TÉRMINOS

Actor del Sistema: Rol o función que asume una persona, sistema o entidad que interactúa con el sistema que estamos construyendo de la misma forma. Tiene la propiedad de ser externo al sistema.

Herramienta CASE: (Computer Aided Software Engineering) Ingeniería de Software Asistida por Ordenador. Diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Interfaz de Comunicación con el Usuario: Forma en que los usuarios pueden comunicarse con una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.

Multimedia: Es todo aquello que utiliza conjunta y simultáneamente diversos medios de comunicación en la presentación de la información, como imágenes, animación, videos, sonido y texto.

Colección Multisaber: Conjunto de software multimedias educativas para la enseñanza primaria.

Software Libre: Es la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Estereotipo: Artefactos que se utilizan para el modelado de un software.