

Universidad de las Ciencias Informáticas



Facultad 8

***Análisis y Diseño del Módulo Mediateca
de la Colección Multisaber.***

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

Autor: Danaisy Gil Harrison.

Tutor: Lic. Liana Isabel Araujo Pérez.

Ciudad de La Habana, Julio 2008.

“Año 50 de la Revolución.”

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 año 2008.

Danaisy Gil Harrison

Firma del Autor

Lic. Liana Isabel Araujo Pérez

Firma del Tutor

*La clave del éxito depende sólo de lo que podemos hacer
de la mejor manera posible.*

Henry Wadsworth Longfellow.

Agradecimiento

Un especial agradecimiento a mis padres por estar ahí siempre que los necesito, por brindarme su apoyo de manera incondicional.

A toda mi familia que siempre se preocupó porque saliese adelante con la tesis.

A mis amistades por ayudarme siempre que podían y por soportar mi mal carácter.

A Yaritza por sacarme de tantos apuros.

A mi tutora Liana, por ayudarme siempre que podía, y cuando no también.

A Héctor que sin ser mi tutor me socorrió de muchas cosas como si lo fuese.

A Osdalme por explicarme cada detallito dudoso que tuviese.

A los profesores del MINED por su ayuda en todo lo pedagógico.

A Abdelasís por su apoyo, comprensión y por alegrarme los últimos días de universidad.

A todas esas personas que cuando me veían me preguntaban ¿y la tesis como va?

*A mi mamá y mi papá, es mi tesis, su mayor
sueño...*

Resumen

El presente trabajo aborda los principales aspectos que se desarrollaron con el fin de realizar el análisis y diseño del Módulo Mediateca de la nueva colección "Multisaber": software educativo para las escuelas primarias de la hermana República Bolivariana de Venezuela. En él se muestran todos los pasos que se siguieron para analizar y diseñar una herramienta que cumpliera con los requerimientos del cliente. A partir del mismo se obtuvo una documentación que servirá como material de estudio para futuras investigaciones y constituye una importante guía para la realización de proyectos similares. Con este documento queda cumplido el objetivo general de la investigación: "realizar la documentación en la Colección Multisaber, específicamente en el módulo Mediateca".

Palabras Claves: Análisis, Diseño, Sistema, Software Libre, Software Educativo.

Índice

Introducción	1
Capítulo 1	6
Fundamentación del tema.....	6
1.1. Introducción	6
1.2. Estado del arte.....	6
1.2.1. Software: hombre del futuro.....	6
1.2.1.1. Software Educativo.....	6
1.2.1.2. Utilización de Multimedia como Software Educativos.	10
1.2.2. Software Libre.....	10
1.2.2.1. ¿Qué es el Software Libre?.....	10
1.2.2.2. Historia del Software Libre	12
1.2.2.3. ¿Qué es la Fundación para el Software Libre?	14
1.2.2.4. ¿Qué es el proyecto GNU?	14
1.3. Conclusiones	15
Capítulo 2	16
Tendencias y tecnologías.....	16
2.1. Introducción.....	16
2.2. Tendencias y tecnologías actuales a considerar.....	16
2.2.1. Ingeniería de Software.....	16
2.2.2. Proceso de Ingeniería de Software.....	16
2.2.3. Metodologías.....	17
2.2.3.1. Proceso Unificado de Desarrollo de Software (RUP).....	20
2.2.4. Lenguajes de Modelados.....	24
2.2.4.1. Lenguaje para la modelación de Aplicaciones Educativas Multimedia ApEM-L	26
2.2.5. Herramientas.....	28
2.2.5.1. Herramientas CASE.....	29
2.3. Conclusiones	30
Capítulo 3	31
Descripción y Construcción de la solución propuesta.....	31
3.1. Introducción	31
3.2. Modelo Dominio.....	31
3.2. Descripción del Sistema propuesto.....	32
3.3.1. Requerimientos Funcionales.....	33
3.3.2. Requerimientos No Funcionales	34
3.4. Subsistemas del Módulo Mediateca.....	36
3.4.1. Vistas por Subsistemas.....	36
3.4.2. Justificación del actor.....	37
3.4.3. Descripción textual de las Vistas.....	38
3.5. Diagrama de Estructura de Presentación.....	67
3.6. Diagrama de Estructura de Navegación.....	74
3.7. Diagrama de Clases.....	75

3.8. Diagrama de Secuencia.....	82
3.9. Diagrama de Despliegue.....	90
3.10. Conclusiones.....	90
Capítulo 4.....	91
Estudio de factibilidad.....	91
4.1. Introducción.....	91
4.2. Planificación.....	91
4.3. Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.....	94
4.4. Beneficios tangibles e intangibles.....	96
4.5. Conclusiones.....	97
Conclusiones.....	98
Recomendaciones.....	99
Referencias Bibliográficas.....	100
Bibliografía.....	101
Glosario.....	102

Índice de ilustraciones

<i>Ilustración 1: Representación de Ingeniería de Software</i>	16
<i>Ilustración 2: RUP en dos dimensiones</i>	22
<i>Ilustración 3: Vista Gestión del Modelo</i>	36
<i>Ilustración 4: Diagrama de Relación de los Usuarios</i>	38
<i>Ilustración 5: Diagrama de Estructura de Presentación de la Vista Presentación</i>	68
<i>Ilustración 6: Diagrama de Estructura de Presentación de la Vista_Galería_Categoría_Imágenes</i>	69
<i>Ilustración 7: Diagrama de Estructura de Presentación de la Vista_Galería_Selección_Categoría_Imágenes</i>	70
<i>Ilustración 8: Diagrama de Estructura de Presentación de la Vista_Galería_Categoría_Videos</i>	71
<i>Ilustración 9: Diagrama de Estructura de Presentación de la Vista_Galería_Selección_Categoría_Videos</i>	72
<i>Ilustración 10: Diagrama de Estructura de Presentación de la Vista_Glosario_Términos</i>	73
<i>Ilustración 11: Diagrama de Estructura de Presentación de la Vista_Interés</i>	74
<i>Ilustración 12: Diagrama de Estructura de Navegación del módulo Mediateca</i>	75
<i>Ilustración 13: Diagrama de Clases del Diseño de la Vista_Presentación</i>	76
<i>Ilustración 14: Diagrama de Clases del Diseño de la Vista_Categoría_Imágenes</i>	77
<i>Ilustración 15: Diagrama de Clases del Diseño de la Vista_Categoría_Selección_Imágenes</i>	78
<i>Ilustración 16: Diagrama de Clases del Diseño de la Vista_Categoría_Video</i>	79
<i>Ilustración 17: Diagrama de Clases del Diseño de la Vista_Categoría_Selección_Videos</i>	80
<i>Ilustración 18: Diagrama de Clases del Diseño de la Vista_Glosario_Términos</i>	81
<i>Ilustración 19: Diagrama de Clases del Diseño de la Vista_Interés</i>	82
<i>Ilustración 20: Diagrama de Secuencia Vista Presentación</i>	83
<i>Ilustración 21: Diagrama de Secuencia Vista_Categoría_Imagenes</i>	84
<i>Ilustración 22: Diagrama de Secuencia Vista_Selección_Categoría_Imágenes</i>	85
<i>Ilustración 23: Diagrama de Secuencia Vista_Categoría_Videos</i>	86
<i>Ilustración 24: Diagrama de Secuencia Vista_Selección_Categoría_Videos</i>	87
<i>Ilustración 25: Diagrama de Secuencia Vista_Glosario</i>	88
<i>Ilustración 26: Diagrama de Secuencia Vista_Interés</i>	89
<i>Ilustración 27: Diagrama de Despliegue Módulo Mediateca</i>	90

Índice de tablas

<i>Tabla 1: Productos de la colección Multisaber para Venezuela.</i>	2
<i>Tabla 2: Libertades del Software Libre</i>	<i>¡Error! Marcador no definido.</i>
<i>Tabla 3: Elementos de RUP</i>	21
<i>Tabla 4: Vistas y diagramas de ApEM – L</i>	28
<i>Tabla 5: Descripción Textual de la Vista de Presentación Módulo Mediateca</i>	38
<i>Tabla 6: Descripción Textual de la Vista Galería_Categoría_Imágenes</i>	45
<i>Tabla 7: Descripción Textual de la Vista Galería_Selección_Categoría_Imágenes</i>	50
<i>Tabla 8: Descripción Textual de la Vista Galería_Categoría_Video</i>	53
<i>Tabla 9: Descripción Textual de la Vista Galería_Selección_Categoría_Video</i>	57
<i>Tabla 10: Descripción Textual de la Vista Glosario de Términos</i>	60
<i>Tabla 11: Descripción Textual de la Vista Interés</i>	64
<i>Tabla 12: Consultas Externas</i>	91
<i>Tabla 13: Salidas Externas</i>	92
<i>Tabla 14: Total de Consultas Externas y Salidas Externas</i>	93
<i>Tabla 15: Puntos de Función</i>	93
<i>Tabla 16: Resumen del estudio de factibilidad</i>	97

Introducción

Antes de la tercera Revolución Educativa la Dirección del Ministerio de Educación (MINED) en Cuba decidió crear grupos de docentes con motivaciones informáticas para elaborar Software Educativos. Generalmente estos docentes eran especialistas de la Facultad de Matemática-Computación, Física, Lengua Española e Inglés de los Institutos Superiores Pedagógicos (ISP) a tiempo completo.

Lo que caracterizó a este periodo fue la elaboración de Software Educativos (SWE) de carácter puntual es decir dedicados a problemas específicos del proceso enseñanza-aprendizaje, como eran "solución de problemas ortográficos, de cálculos numéricos, cinemáticas, etcétera.", cuya elaboración no siempre estaba sustentada en bases teóricas bien fundamentadas y mucho menos se realizaba bajo preceptos que tuvieran que ver con la Ingeniería del Software; era un proceso amateur, espontáneo totalmente.

Otra característica en el aspecto informático lo constituyó la diversidad de herramientas y lenguaje que fueron empleados para estos, en su gran mayoría sobre el sistema MS-DOS (por ejemplo PASCAL, BASIC, C).

Se crearon de esta manera 16 centros de desarrollo de SWE a lo largo de todo el país sobre la base de cada uno de los ISP del MINED.

A partir del Año 2000 y como resultado de la reciente disolución de la URSS, entre otros factores, nuestro país se dio a la tarea de realizar profundas transformaciones en la educación, las cuales estuvieron inspiradas en el marco de la conocida "Batalla de Ideas".

La Revolución Cubana tiene como base la búsqueda de un justo equilibrio entre los componentes instructivos y educativos del proceso enseñanza-aprendizaje en la escuela cubana, considerándose como fin primordial la elevación del componente educativo que tiene que ver con la formación del hombre, la formación de valores y esencialmente el desarrollo de un sentimiento de soberanía e independencia. Para el logro de estos objetivos "la piedra angular" lo constituyó la disminución sustancial de la cantidad de alumnos por profesores, para lograr una mayor atención a las diferencias individuales de los educandos y de esta forma reforzar el trabajo de los mismos. Este aspecto tiene una repercusión económica (como aumentan la formación de docentes lo que obliga a un mayor número de salario, y actualmente no se encuentra sustentado por la economía actual del país) que obliga a elevar el diapasón de materias que el profesor debía atender, es el fenómeno denominado "integralidad del maestro" (PGI). Resulta obvio que la integralidad, al menos en sus inicios, está en

contraposición con el dominio de los contenidos, de manera tal que para compensar este desequilibrio natural apareció la tecnología acompañada de modernas teorías de aprendizajes para dar solución a la contradicción planteada, de esta manera surgió la idea de la creación de unas colecciones de SWE para cada uno de los sistemas educacionales del país:

Multisaber	Educación Primaria
Navegante	Educación Secundaria
Futuro	Educación Preuniversitaria

La misión de estas colecciones estuvo a cargo de la dirección política de la Unión de Jóvenes Comunistas (UJC), bajo la asesoría técnica y pedagógica del MINED y del Ministerio de la Informática y las Comunicaciones (MIC) en figura de (SYS-COPEXTEL) por la parte informática. A tal efecto se elaboró un modelo didáctico para conformar la arquitectura del sistema. Este modelo respondía a la tipología de hiperentornos de aprendizaje que no son más que una mezcla sistémica de diversas tecnologías de SW (libros electrónicos, entrenadores, evaluadores, juegos, simuladores, bibliotecas virtuales) sustentados en tecnologías hipermedias. Nacionalmente se prepararon a un grupo de docentes en calidad de generadores de contenidos para los modelos mencionados. La implementación estuvo a cargo de las siguientes instituciones: MINED, MES, Joven Club de Computación y Electrónica (JCCE).

A partir del año 2001 con la llegada al poder del presidente Hugo Rafael Chávez Frías a la República de Venezuela, comenzaron a surgir una serie de transformaciones a favor de la clase obrera, el campesinado, la juventud, los niños, para el bienestar de la sociedad, comienzan así a fortalecerse las relaciones de amistad entre nuestro pueblo y la hermana nación, y se realizan una serie de contratos en las esferas mas importantes: salud, deporte, educación, entre otras.

En el sector de la educación el intercambio y la colaboración se extendió a la asistencia en métodos, programas y nuevas técnicas del proceso docente educativo para el desarrollo de la República Bolivariana de Venezuela, entre estos programas se encuentra Multisaber: colección de SWE para la Enseñanza Primaria que 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 (1). Es una colección que se encuentra actualmente en todas las escuelas primarias cubanas, es usada por los docentes, constituyendo una gran ayuda para lograr que el niño se apropie más amena y rápidamente de los contenidos que en el software se presentan. La implantación de estos software en Venezuela necesita de la contextualización de los contenidos presentes en las multimedias cubanas,

así como la reconstrucción de las mismas con la utilización exclusiva de software y recursos libres, respondiendo esto a la política en lo que a informática refiere de la República Bolivariana de Venezuela, contenida en el Decreto 3.390 del presidente Hugo Rafael Chávez, que determina la migración a Software Libre de todos los organismos estatales del país. La nueva colección estará formada por 14 productos (tabla 1) de los 32 presentes en la Colección Multisaber para los niños cubanos, tratando las asignaturas de Lengua Española, Matemática, Ciencias Naturales y Computación Básica.

Tabla 1: Productos de la Colección Multisaber para Venezuela.

No	Nombre del software	Asignaturas	Grados
1	El secreto de la lectura I	Lengua Española	1ro a 4to
2	El secreto de la lectura II	Lengua Española	5to y 6to
3	Nuestro idioma I	Lengua Española	1ro a 4to
4	Nuestro idioma II	Lengua Española	5to y 6to
5	Jugando con las palabras	Lengua Española	3ro a 6to
6	La feria de las Matemáticas	Matemática	1ro a 4to
7	El país de los números	Matemática	5to y 6to
8	Problemas matemáticos I	Matemática	1ro a 4to
9	Problemas matemáticos II	Matemática	5to y 6to
10	Las formas que nos rodean I	Matemática (Geometría)	1ro a 4to
11	Las formas que nos rodean II	Matemática (Geometría)	5to y 6to
12	Misterios de la naturaleza	Ciencias Naturales	5to y 6to
13	Amemos el medio ambiente	Ciencias Naturales	5to y 6to
14	El ratón y la ventana	Computación Básica	1ro a 6to

Cada uno de los productos de la colección presentará un módulo llamado Mediateca el cual cuenta habitualmente con los siguientes elementos:

- Glosario de términos
- Galería de imágenes
- Artículos de Interés
- Galería de videos

Glosario de términos: se incluyen palabras de difícil comprensión relacionadas con los temas abordados en el software y las palabras calientes que aparecen en los distintos módulos, las cuales ofrecen el significado de las mismas y en muchos casos ilustrados a través de imágenes. El glosario contribuye al desarrollo del vocabulario científico de los escolares, así como a crear una actitud consiente del uso del diccionario.

Galería de imágenes y Galería de videos: se recogen las imágenes y videos respectivamente utilizados en los distintos módulos, clasificados por temáticas y con una breve explicación sobre lo que muestran.

Artículos de interés: se presentan curiosidades, artículos periodísticos, efemérides de importancia relacionadas con la materia del software, bibliografías de autores, informaciones adicionales relacionadas a los contenidos del software en cuestión donde se encuentran.

Debido a la falta de documentación de la colección Multisaber, que puede implicar un mayor esfuerzo a la hora de realizar cambios en el proyecto se tiene como **problema a resolver** ¿Cómo lograr la documentación requerida en el flujo de trabajo de análisis y diseño del módulo Mediateca de la nueva colección de multimedias que se construye para la República Bolivariana de Venezuela, basada en la Colección Multisaber cubana?

. El **objeto de estudio** es el proceso de documentación de software educativo, específicamente en el módulo Mediateca de la Colección Multisaber como **campo de acción**, teniendo como **ideas a defender** que si se logra tener la documentación de este producto para software libre, servirá de guía para la creación o migración de productos similares. Como **objetivo general** se plantea realizar el análisis y diseño del módulo Mediateca de la Colección Multisaber y como **objetivos específicos**:

- Realizar un estudio de la descripción del módulo “Mediateca” de cada una de las 14 multimedias con las que contará la colección a reconstruir.

- Realizar un estudio de las metodologías y de los lenguajes de modelación existentes para el desarrollo de multimedia.
- Elaborar la documentación de módulo “Mediateca” de la nueva Colección Multisaber.
- Realizar el estudio de factibilidad de la construcción del módulo “Mediateca”

Aportes prácticos esperados del trabajo

- El resultado de este trabajo formará parte de la documentación de la nueva Colección Multisaber y servirá de apoyo para la realización de otros software similares.

Estructura del Documento

Capítulo 1: “Fundamentación del tema”, aborda de manera clara y precisa la descripción de la situación actual en cuanto al Software Educativo, con más profundidad, en los que utilizan tecnologías multimedia.

Capítulo 2: “Tendencias y Tecnología” se expone la situación a nivel internacional de las tecnologías apropiadas para el desarrollo del sistema; también se presentan las metodologías y los lenguajes de modelados para el desarrollo de software escogiéndose la más eficiente.

Capítulo 3: “Descripción y construcción de la solución propuesta” se hace alusión a la descripción y construcción de la propuesta de solución del sistema, exponiendo en este, una explicación detallada de todas las vistas del sistema, y los diagramas que permitirán un mayor entendimiento del mismo.

Capítulo 4: “Estudio de factibilidad”, se realiza un estudio con el objetivo de saber cuanto esfuerzo de manera real se necesita para realizar el sistema y cuanto costará la realización del mismo.

Capítulo 1

Fundamentación del tema.

1.1. Introducción

El presente capítulo es referido a la situación actual que presenta el tema en cuestión. Se hace alusión a todo lo concerniente con el término Software Educativo en la actualidad, y más específicamente a los que utilizan la tecnología multimedia. Se hace una minuciosa descripción del objeto de estudio.

1.2. Estado del arte.

1.2.1. Software: hombre del futuro.

Se denomina software, programática, equipamiento lógico o soporte lógico al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware). Esto incluye aplicaciones informáticas tales como un procesador de textos y software de sistema como un sistema operativo, facilitando la interacción con los componentes físicos y el resto de aplicaciones.[1]

Probablemente la definición más formal de software es la atribuida a la IEEE: «la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo». Bajo esta definición, el concepto de software va más allá de los programas de cómputo en sus distintas formas: código fuente, binario o ejecutable, además de su documentación: es decir, todo lo intangible.

El término «software» fue usado por primera vez en este sentido por John W. Tukey en 1957. En las ciencias de la computación y la ingeniería de software, el software es toda la información procesada por los sistemas informáticos: programas y datos. La teoría que forma la base de la mayor parte del software moderno fue propuesta por vez primera por Alan Turing en su ensayo de 1936, los números computables, con una aplicación al problema de decisión.[2]

1.2.1.1. Software Educativo.

Software educativo es el software destinando a la enseñanza y el autoaprendizaje y además permite el desarrollo de ciertas habilidades cognitivas. Como existen profundas diferencias entre las filosofías pedagógicas, también existe una amplia gama de enfoques para la creación de software educativo

atendiendo a los diferentes tipos de interacción que debería existir entre los actores del proceso de enseñanza-aprendizaje: educador, aprendiz, conocimiento, computadora. Como software educativo tenemos desde programas orientados al aprendizaje hasta sistemas operativos completos destinados a la educación, como por ejemplo las distribuciones Linux orientadas a la enseñanza.[3]

Software educativo es cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar. Un concepto más restringido de software educativo es definido como aquel material de aprendizaje especialmente diseñado para ser utilizado con una computadora en los procesos de enseñar y aprender. (2)

Según Rodríguez Lamas (2000), 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.

Finalmente, los software educativos se pueden considerar como el conjunto de recursos informáticos diseñados con la intención de ser utilizados en el contexto del proceso de enseñanza–aprendizaje.

Están caracterizados por ser altamente interactivos, a partir del empleo de recursos multimedia, como videos, sonidos, fotografías, diccionarios especializados, explicaciones de experimentados profesores, ejercicios y juegos instructivos que apoyan las funciones de evaluación y diagnóstico.

Los software educativos pueden tratar las diferentes materias (Matemática, Idiomas, Geografía, Dibujo), de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos) 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; pero todos comparten las siguientes características:

- Permite la interactividad con los estudiantes, retroalimentándolos y evaluando lo aprendido.
- Facilita las representaciones animadas.
- Incide en el desarrollo de las habilidades a través de la ejercitación.
- Permite simular procesos complejos.
- Reduce el tiempo de que se dispone para impartir gran cantidad de conocimientos facilitando un trabajo diferenciado, introduciendo al estudiante en el trabajo con los medios computarizados.
- Facilita el trabajo independiente y a la vez un tratamiento individual de las diferencias.
- Permite al usuario (estudiante) introducirse en las técnicas más avanzadas.

El uso del software educativo en el proceso de enseñanza - aprendizaje puede ser:

- Por parte del alumno.

Se evidencia cuando el estudiante opera directamente el software educativo, pero en este caso es de vital importancia la acción dirigida por el profesor.

- Por parte del profesor.

Se manifiesta cuando el profesor opera directamente con el software y el estudiante actúa como receptor del sistema de información. La generalidad plantea que este no es el caso más productivo para el aprendizaje.

El uso del software por parte del docente proporciona numerosas ventajas, entre ellas:

- Enriquece el campo de la Pedagogía al incorporar la tecnología de punta que revoluciona los métodos de enseñanza - aprendizaje.
- Constituyen una nueva, atractiva, dinámica y rica fuente de conocimientos.
- Pueden adaptar el software a las características y necesidades de su grupo teniendo en cuenta el diagnóstico en el proceso de enseñanza - aprendizaje.
- Permiten elevar la calidad del proceso docente - educativo.
- Permiten controlar las tareas docentes de forma individual o colectiva.
- Muestran la interdisciplinariedad de las asignaturas.
- Marca las posibilidades para una nueva clase más desarrolladora.

Los software educativos a pesar de tener unos rasgos esenciales básicos y una estructura general común se presentan con unas características muy diversas: unos aparentan ser un laboratorio o una biblioteca, otros se limitan a ofrecer una función instrumental del tipo máquina de escribir o calculadora, otros se presentan como un juego o como un libro, bastantes tienen vocación de examen, unos pocos se creen expertos y la mayoría participan en mayor o menor medida de algunas de estas peculiaridades.

Para poner orden a esta disparidad se elaboraron múltiples tipologías que los clasifican a partir de diferentes criterios.

Por ejemplo, hasta el año 2003, según los polos en los cuales se ha movido la educación, existían dos tipos de software educativos:

1.- Algorítmicos donde predomina el aprendizaje vía transmisión del conocimiento, pues el rol del alumno es asimilar el máximo de lo que se le transmite.

Se pueden clasificar según función educativa en:

Sistemas Tutoriales

Sistema basado en el diálogo con el estudiante, adecuado para presentar información objetiva, tiene en cuenta las características del alumno, siguiendo una estrategia pedagógica para la transmisión de conocimientos.

Sistemas Entrenadores

Se parte de que los estudiantes cuentan con los conceptos y destrezas que van a practicar, por lo que su propósito es contribuir al desarrollo de una determinada habilidad, intelectual, manual o motora, profundizando en las dos fases finales del aprendizaje: aplicación y retroalimentación.

Libros Electrónicos

Su objetivo es presentar información al estudiante a partir del uso de textos, gráficos, animaciones, videos, etcétera., pero con un nivel de interactividad y motivación que le facilite las acciones que realiza.

2.- Heurísticos donde el estudiante descubre el conocimiento interactuando con el ambiente de aprendizaje que le permita llegar a él.

Se pueden clasificar según su función educativa en:

Simuladores

Su objetivo es apoyar el proceso de enseñanza – aprendizaje, semejando la realidad de forma entretenida.

Juegos Educativos

Su objetivo es llegar a situaciones excitantes y entretenidas, sin dejar en ocasiones de simular la realidad.

Sistemas Expertos

Programa de conocimientos intensivo que resuelve problemas que normalmente requieren de la pericia humana. Ejecuta muchas funciones secundarias de manera análoga a un experto, por ejemplo, preguntar aspectos importantes y explicar razonamientos.

Sistemas Tutoriales Inteligentes de enseñanza

Despiertan mayor interés y motivación, puesto que pueden detectar errores, clasificarlos, y explicar por qué se producen, favoreciendo así el proceso de retroalimentación del estudiante.

A partir del 2004 surge una nueva tendencia, que es la de integrar en un mismo producto, todas o algunas de estas tipologías de software educativo. A este nuevo modelo de software se le ha denominado **HIPERENTORNO EDUCATIVO o HIPERENTORNO DE APRENDIZAJE**, lo cual no es más que un sistema informático basado en tecnología hipertexto que contiene una mezcla de elementos representativos de diversas tipologías de software educativo.[4]

1.2.1.2. Utilización de Multimedia como software educativo.

Multimedia (Lat. Multi + Medium) es un término que se aplica a cualquier objeto que usa simultáneamente diferentes formas de contenido informativo como texto, sonido, imágenes, animación y video para informar o entretener al usuario. También se puede calificar como multimedia a los medios electrónicos (u otros medios) que permiten almacenar y presentar contenido multimedia. Multimedia es similar al empleo tradicional de medios mixtos en las artes plásticas, pero con un alcance más amplio. Se habla de multimedia interactiva cuando el usuario tiene cierto control sobre la presentación del contenido, como qué desea ver y cuándo desea verlo.[5]

1.2.2. Software Libre.

1.2.2.1. ¿Qué es el Software Libre?

«Software Libre» se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

La libertad de usar el programa, con cualquier propósito (libertad 0).

La libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.

La libertad de distribuir copias, con lo que puede ayudar a otros (libertad 2).

La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (libertad 3). De igual forma que la libertad 1 el acceso al código fuente es un requisito previo.[6]

Ciertos teóricos usan este cuarto punto (libertad 3) para justificar parcialmente las limitaciones impuestas por la licencia GNU GPL frente a otras licencias de software libre (ver Licencias GPL). Sin embargo el sentido original es más libre, abierto y menos restrictivo que el que le otorga la propia situación de incompatibilidad, que podría ser resuelta en la próxima versión 3.0 de la licencia GNU GPL (en desarrollo), causa en estos momentos graves perjuicios a la comunidad de programadores de software libre, que muchas veces no pueden reutilizar o mezclar códigos de dos licencias distintas, pese a que las libertades teóricamente lo deberían permitir.

En el sitio web oficial de OSI está la lista completa de las licencias de software libre actualmente aprobadas y tenidas como tales.

El término software no libre se emplea para referirse al software distribuido bajo una licencia de software más restrictiva que no garantiza estas cuatro libertades. Las leyes de la propiedad intelectual reservan la mayoría de los derechos de modificación, duplicación y redistribución para el dueño del *copyright*; el software dispuesto bajo una licencia de software libre rescinde específicamente la mayoría de estos derechos reservados.

La definición de Software Libre no contempla el asunto del precio; un eslogan frecuentemente usado es "libre como en libertad, no como en cerveza gratis" o en inglés "*Free as in freedom, not as in free beer*" (aludiendo a la ambigüedad del término inglés "*free*"), y es habitual ver a la venta CDs (Discos Compactos) de software libre como distribuciones Linux. Sin embargo, en esta situación, el comprador del CD tiene el derecho de copiarlo y redistribuirlo. El software gratis puede incluir restricciones que no se adaptan a la definición de software libre —por ejemplo, puede no incluir el código fuente, puede prohibir explícitamente a los distribuidores recibir una compensación a cambio.

Para evitar la confusión, algunas personas utilizan los términos "libre" (*Libre software*) y "gratis" (*Gratis software*) para evitar la ambigüedad de la palabra inglesa "free". Sin embargo, estos términos alternativos son usados únicamente dentro del movimiento del software libre, aunque están extendiéndose lentamente hacia el resto del mundo. Otros defienden el uso del término *open source software* (software de código abierto, también llamado de fuentes abiertas). La principal diferencia entre los términos "open source" y "free software" es que éste último tiene en cuenta los aspectos éticos y filosóficos de la libertad, mientras que el "open source" se basa únicamente en los aspectos técnicos.

En un intento por aunar los mencionados términos que se refieren a conceptos semejantes, se está extendiendo el uso de la palabra "FLOSS" con el significado de "Free - Libre - Open Source Software" e, indirectamente, también a la comunidad que lo produce y apoya.[7]

Software libre (en inglés *free software*) es la denominación del software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la *Free Software Foundation*, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (para la segunda y última libertad mencionada, el acceso al código fuente es un requisito previo).[6]

El software libre suele estar disponible gratuitamente, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo no hay que asociar software libre a "software gratuito" (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en algunas ocasiones el código fuente; no obstante, este tipo de software *no es libre* en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con "software de dominio público". Éste último es aquél que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquel cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado, tras un plazo contado desde la muerte de éste, habitualmente 70 años. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es dominio público.[7]

1.2.2.2. Historia del Software Libre

Entre los años 60 y 70 del Siglo XX, el software no era considerado un producto sino un añadido que los vendedores de los grandes computadores de la época (los *mainframes*) aportaban a sus clientes para que éstos pudieran usarlos. En dicha cultura, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros. Este comportamiento era

particularmente habitual en algunos de los mayores grupos de usuarios de la época, como DECUS (grupo de usuarios de computadoras DEC). A finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de licencia.

Allá por el 1971, cuando la informática todavía no había sufrido su gran desarrollo, las personas que hacían uso de ella, en ámbitos universitarios y empresariales, creaban y compartían el software sin ningún tipo de restricciones.

Con la llegada de los años 80 la situación empezó a cambiar. Las computadoras más modernas comenzaban a utilizar sistemas operativos privativos, forzando a los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones a dicho software.

En caso de que algún usuario o programador encontrase algún error en la aplicación, lo único que podía hacer era darlo a conocer a la empresa desarrolladora para que esta lo solucionara. Aunque el programador estuviese capacitado para solucionar el problema y lo deseara hacer sin pedir nada a cambio, el contrato le impedía que mejorase el software.

El mismo Richard Stallman cuenta que por aquellos años, en el laboratorio habían recibido una impresora donada por una empresa externa. El dispositivo, era utilizado en red por todos los trabajadores, parecía no funcionar a la perfección dado que cada cierto tiempo el papel se atascaba. Como agravante, no se generaba ningún aviso que se enviase por red e informase a los usuarios de la situación.

La pérdida de tiempo era constante, ya que en ocasiones, los trabajadores enviaban por red sus trabajos a imprimir y al ir a buscarlos se encontraban la impresora atascada y una cola enorme de trabajos pendientes. Richard Stallman decidió arreglar el problema, e implementar el envío de un aviso por red cuando la impresora se bloqueara. Para ello necesitaba tener acceso al código fuente de los controladores de la impresora. Pidió a la empresa propietaria de la impresora lo que necesitaba, comentando, sin pedir nada a cambio, que era lo que pretendía realizar. La empresa se negó a entregarle el código fuente.

En ese preciso instante, Richard Stallman se vio en una encrucijada, debía elegir entre aceptar el nuevo software privativo firmando acuerdos de no revelación y acabar desarrollando más software privativo con licencias restrictivas, que a su vez deberían ser más adelante aceptadas por sus propios colegas.

Con este antecedente, en 1984 Richard Stallman comenzó a trabajar en el proyecto GNU, y un año más tarde fundó la Free Software Foundation (FSF). Stallman introdujo una definición para *free*

software y el concepto de "*copyleft*", el cual desarrolló para dar a los usuarios libertad y para restringir las posibilidades de apropiación del software [7]

1.2.2.3. ¿Qué es la Fundación para el Software Libre?

La Fundación para el Software Libre (FSF) es la principal organización que patrocina el proyecto GNU. La FSF recibe muy pocos fondos de corporaciones o fundaciones, realmente dependemos de personas como usted para defender la misión de la FSF y así preservar, proteger y promover los derechos de los usuarios. La libertad de usar, estudiar, copiar, modificar y redistribuir software. Como puede ver, la continuidad de nuestro trabajo depende principalmente de este apoyo. Por favor considere hacer una donación, ser un miembro asociado de la FSF, comprar el libro *Free Software, Free Society*, o impulsar a su empresa para que sea un patrocinador corporativo (Corporate Patron) de la FSF.

La FSF apoya la libertad de expresión, prensa y asociación en Internet, el derecho a usar software criptográfico en comunicaciones privadas, y el derecho a escribir software sin los impedimentos del monopolio. También puede leer más sobre estas ediciones al suscribirse a la revista independiente del Software Libre.

1.2.2.4. ¿Qué es el proyecto GNU?

El proyecto GNU se inició en 1984 para desarrollar un completo sistema operativo tipo Unix, bajo la filosofía del software libre: el sistema GNU. Las variantes del sistema operativo GNU que utilizan el núcleo llamado Linux, son utilizadas ampliamente en la actualidad; aunque a menudo estos sistemas se refieren como «Linux», deben ser llamados sistemas GNU/Linux.

GNU es un acrónimo recursivo para «GNU No es Unix» y se pronuncia fonéticamente en español.

GNU GPL (Licencia Pública General) es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Existen varias licencias "hermanas" de la GPL, como la licencia de documentación libre GNU (GFDL) que cubre los artículos de la Wikipedia, la Open Audio License, para trabajos musicales, etcétera, y otras menos restrictivas, como la LGPL, o la LGPL (Lesser General Public License o Library General Public License), que permiten el enlace dinámico de aplicaciones libres a aplicaciones no libres.

1.3. Conclusiones

Hay algo que sí es un hecho y que se debe considerar: la idea de que las tecnologías de la información y la comunicación están aquí para quedarse, y con ellas el avance de los últimos años en la manera de ver el lente abstracto de la "enseñanza-aprendizaje".

Es por eso que al concluir este capítulo y hacer varias reflexiones sobre todo lo que en el se expone, quedan argumentadas claramente las bases que sustentan este trabajo, al ser precisamente bases tecnológicas e informáticas, apreciándolas desde el punto de vista que esta es una solución óptima cuando de enseñar se habla, siempre y cuando sea utilizada en una forma adecuada. Se evidencia además la gran utilidad que tiene el conocimiento que a lo largo de estos años ha acumulado la Informática Educativa, para poder enfrentar como es debido, un software en el ámbito pedagógico y educacional.

Capítulo 2

Tendencias y tecnologías.

2.1. Introducción.

En el presente capítulo se realiza un estudio de cómo se encuentran a nivel internacional las tecnologías que pueden ser apropiadas para llevar a cabo el sistema que se pretende desarrollar. Además se analiza la metodología a utilizar para el análisis y diseño del sistema, teniendo en cuenta las facilidades que puede aportar al trabajo. Por último se exponen las propuestas de herramientas a utilizar para el desarrollo de la aplicación, teniendo en cuenta el resultado de los análisis anteriores.

2.2. Tendencias y tecnologías actuales a considerar.

2.2.1. Ingeniería de Software.

La Ingeniería de Software es una tecnología multicapa en la que, según Pressman, se pueden identificar: los métodos (indican cómo construir técnicamente el software), el proceso (es el fundamento de la Ingeniería de Software, es la unión que mantiene juntas las capas de la tecnología) y las herramientas (soporte automático o semiautomático para el proceso y los métodos).[1]



Ilustración 1: Representación de Ingeniería de Software

2.2.2. Proceso de Ingeniería de Software.

El proceso de Ingeniería de Software se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad".

El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo".[8]

2.2.3. Metodologías.

La rama de la metodología, dentro de la ingeniería de software, se encarga de elaborar estrategias de desarrollo de software centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente.

En un proyecto de desarrollo de software la metodología define Quién debe hacer, Qué, Cuándo y Cómo debe hacerlo. La metodología constituye la columna vertebral del proceso de desarrollo de software. No existe una metodología universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable. [9]

Entre las metodologías más destacadas en el desarrollo de software multimedia podemos nombrar:

- **SCRUM:** Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se han utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Entre sus principales características se pueden encontrar: el desarrollo de software se realiza mediante iteraciones, denominadas sprint, con una duración de 30 días, el resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (3)
- **Crystal Methodologies. (Metodologías Cristal):** Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros). (4)
- **Dynamic Systems Development Method. (DSDM) (Método de Desarrollo de Sistemas Dinámicos):** Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: un

proceso iterativo e incremental; el trabajo en conjunto del equipo de desarrollo y el usuario. Propone cinco fases: estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación.(5)

· **Adaptive Software Development (ASD) (Desarrollo de Software Adaptativos):** Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo. (6)

· **Feature -Driven Development (FDD) (Desarrollo de Características Impulsivas):** Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad. (7)

· **Lean Development (LD) (Desarrollo Inclinado):** Definida por Bob Charette's a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios. (8)

Extreme Programming (XP) (Programación Extrema): XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Hypertext Design Model (HDM) (Modelo de Diseño de Hipertexto): fue creada con el objetivo de crear un modelo que sea de utilidad para realizar el diseño de una aplicación de hipertexto (o hipermedia).

Más que describir la estructura interna de una aplicación que utilice hipermedia, crea un modelo que sea de utilidad para realizar el diseño de una aplicación a partir de la estructura de datos siendo muy útil para el análisis de aplicaciones ya existentes (con el fin de detectar errores en la estructura de navegación) y comparaciones entre ellas. Es un modelo más bien para evaluar aplicaciones (evaluación orientada al diseño).

Este modelo no permite que unas entidades sean componentes de otras entidades (estas tienen complejas estructuras internas), aunque los autores no lo recomiendan, diferentes unidades pueden compartir el mismo cuerpo (según los autores, esto conlleva a una desorientación para el usuario, aunque por otra parte, fomentaría la reusabilidad). Seguir el modelo HDM es extremadamente complicado cuando el número de entidades involucradas crece.

Relationship Management Methodology (RMM) (Metodología de Administración de Relaciones): fue la primera metodología para el diseño de multimedia. Se trata, probablemente, del único método para hipermedia que parece cubrir todo el ciclo de desarrollo, desde el estudio de factibilidad hasta la evaluación del sistema, aunque sólo propone actividades y productos concretos para las fases de análisis y de diseño.

La clase de aplicaciones para la cual RMM es más adecuada, corresponde a las que presentan una estructura regular para un dominio de interés, en donde hay clases de objetos, relaciones definibles entre éstas clases, y múltiples instancias de objetos dentro de cada clase. Ejemplo de software que satisfacen estos requerimientos son los catálogos de productos, las aplicaciones hipermediales frontales para bases de datos tradicionales o aplicaciones legadas. Considerando que muchas de ellas poseen datos volátiles que requieren actualizaciones frecuentes, se hace necesario disponer de medios que permitan automatizar y agilizar los desarrollos iniciales y los subsecuentes procesos de actualización.

El análisis se realiza por medio de un diagrama entidad-relación en el que sólo se permiten relaciones con cardinalidades $1 \rightarrow 1$ o $1 \rightarrow N$ y tampoco es posible establecer relaciones reflexivas.

Con respecto al diseño, una de las características más relevantes de este método es que se hace tanto de forma ascendente como descendente, ofreciendo una interesante manera de realizar una verificación.

Para el resto de las fases: estudio de factibilidad, implementación, pruebas y evaluación, no se propone ningún tipo de guía.

RMM constituye una metodología tentadora para el desarrollo del proceso por el desglose de las fases de la producción y la incorporación de diagramas para el diseño de la presentación, el comportamiento dinámico y la estructura de la navegación. No obstante, su uso óptimo se basa en las aplicaciones de catálogo de productos y aplicaciones hipermediales frontales para bases de datos tradicionales o aplicaciones legadas por poseer una alta volatilidad de la información.

Por último se realizó el estudio del Proceso Unificado de Desarrollo de Software (RUP por su nombre en inglés Rational Unified Process), que fue la seleccionada. Esta metodología es un proceso de Ingeniería de Software cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos, cubriendo el ciclo de vida de desarrollo del software.

2.2.3.1 Proceso Unificado de Desarrollo de Software (RUP).

También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de IBM, el cual incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el Rational Method Composer (Método de Composición, RMC), que permite la personalización de acuerdo a necesidades.

Originalmente se diseñó un proceso genérico y de dominio público, el Proceso Unificado, y una especificación más detallada, el Rational Unified Process (RUP), que se vendiera como producto independiente.

¿Por qué se escoge RUP?

Luego de un estudio profundo de las diferentes metodologías existentes se selecciona a RUP ya que este no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

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

Tabla 2: Elementos de RUP.

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

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como flujos de apoyo. En la Ilustración 2: RUP en dos dimensiones se representa el proceso en el que se grafican los flujos de trabajo, las fases, muestra la dinámica expresada en iteraciones y puntos de control.

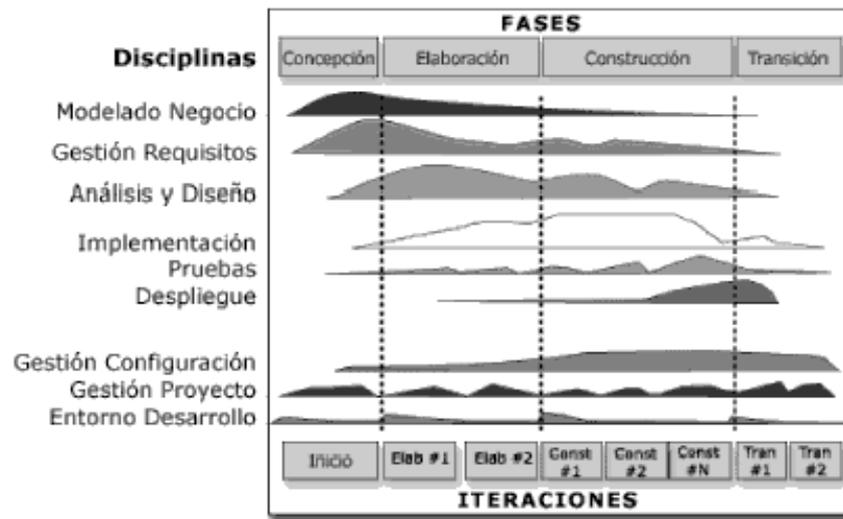


Ilustración 2: RUP en dos dimensiones.

Flujos de trabajo:

Modelación del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

Prueba (Testeo): Busca los defectos a lo largo del ciclo de vida.

Instalación: Produce la liberación del producto y realiza actividades (empaquete, instalación, asistencia a usuarios) para entregar el software a los usuarios finales.

Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etcétera.

Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Fases:

Conceptualización (Concepción o Inicio): Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.

Elaboración: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

Construcción: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene una o varias versiones del producto que han pasado las pruebas. Se ponen estas versiones a consideración de un subconjunto de usuarios.

Transición: El trabajo ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

El ciclo de vida de RUP se caracteriza por:

1. **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
2. **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los casos de usos relevantes desde el punto de vista de la arquitectura.
3. **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla

fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.(9)

2.2.4. Lenguajes de Modelado.

El lenguaje de modelado de objetos es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar (parte de) un diseño de software orientado a objetos.

Algunas organizaciones los usan extensivamente 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. El uso de un lenguaje de modelado es más sencillo que la auténtica programación, pues existen menos medios para verificar efectivamente el funcionamiento adecuado del modelo. Esto puede suponer también que las interacciones entre partes del programa den lugar a sorpresas cuando el modelo ha sido convertido en un software funcionando.

Algunos metodólogos del software orientado a objetos distinguen tres grandes "generaciones" cronológicas de técnicas de modelado de objetos:

- En la primera generación, tecnólogos aislados y grupos pequeños desarrollaban técnicas que resolvían problemas que se encontraban de primera mano en los proyectos de desarrollo Orientados a Objetos. En esta generación se incluye a autores y técnicas como Rumbaugh, Jacobson, Booch, los métodos formales, Shlaer-Mellor y Yourdon-Coad.
- En la segunda generación se reconoció que muchas de las mejores prácticas pertenecían a diferentes métodos del fragmentado terreno de la metodología orientada a objetos. En cualquier caso, la comunidad del software orientado a objetos empezaba a reconocer los beneficios que la estandarización de las técnicas conllevaría: abandono de las "buenas" formas de hacer las cosas en favor de "la" manera adecuada, que permitiría un lenguaje y unas prácticas comunes entre los diferentes desarrolladores.
- La tercera generación consiste en intentos creíbles de crear dicho lenguaje unificado por la industria, cuyo mejor ejemplo es UML.[10]

Entre los lenguajes de modelado que más se utilizan podemos encontrar:

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, está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar,

especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), 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.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar.

Lenguaje de Modelado para Aplicaciones Multimedia (OMMMA-L)

El Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia (OMMMA-L) constituye una extensión de UML que favorece el modelado para aplicaciones que utilicen técnicas multimedia.

Una aplicación de este tipo es un sistema de software interactivo con diferentes tipos de medias y que incluye una determinada funcionalidad con una interfaz de usuario. UML, puede representar la funcionalidad de un sistema a través de diagramas de clases y refinarla en el análisis y diseño con los diagramas de comportamiento, pero este lenguaje no ofrece estereotipos para el modelo de interfaz de usuario ni ningún artefacto que relacione los objetos de tipo media que presenta esta aplicación con las clases de la misma.

OMMMA-L para solucionar lo anterior crea su propia versión de algunos diagramas presentándolos de una manera más cómoda y más acorde al tipo de aplicación a realizar.

OMMMA-L representa lo anteriormente expuesto a través de diagramas basándose siempre en su fuente de inspiración UML que agrupa sus diagramas ya antes mencionados en cuatro categorías: Diagrama de Casos de Uso, Diagramas Estructurales, Diagramas de Comportamiento y Diagramas de Implementación, siendo el segundo y el tercero quienes interactúan directamente con las descripciones hechas anteriormente.

Para OMMMA-L podemos modelar la estructura lógica a través de diagramas de objetos y clases; la distribución espacial de media puede ser descrita a través de un nuevo artefacto propuesto para el lenguaje, el diagrama de presentación; el comportamiento temporal a través de diagramas de secuencia y por último el control interactivo a través de los diagramas de estado.[11]

Object Constraint Language Specification o Lenguaje de Restricción de Objetos (OCL).

OCL fue adoptado en octubre de 2003 por el grupo OMG como parte de UML 2.0. OCL es un lenguaje para la descripción formal de expresiones en los modelos UML. Sus expresiones pueden representar invariantes, precondiciones, postcondiciones, inicializaciones, guardias, reglas de derivación, así como consultas a objetos para determinar sus condiciones de estado. Se trata de un lenguaje sin efectos de borde, de manera que la verificación de una condición, que se presupone una operación instantánea, nunca altera los objetos del modelo. Su papel principal es el de completar los diferentes artefactos de la notación UML con requerimientos formalmente expresados.

El UML incorpora el lenguaje de restricción de objetos a fin de superar las deficiencias que poseen los elementos UML para definirse a sí mismos. Podría haberse utilizado el lenguaje natural, pero su obvia ambigüedad provocó la necesidad de la incorporación del OCL. Se puede decir que se trata de un lenguaje de expresión, porque se utiliza únicamente para expresar un valor, sin provocar cambios en el modelo; de un lenguaje de modelado, porque no es ejecutable, no es un lenguaje de programación; y si un lenguaje formal, porque todos sus elementos han sido definidos formalmente.

Según los autores, este lenguaje está destinado a la especificación de invariantes en clases y tipos de clases en el modelo de clases; para describir pre y post-condiciones en operaciones y métodos; para describir guardas; para ser empleado como un lenguaje de navegación; para definir reglas de formación; y para definir restricciones de operaciones.

2.2.4.1. Lenguaje para la modelación de Aplicaciones Educativas Multimedia ApEM-L.

ApEM – L es una extensión de UML, tomando como bases teóricas principales

OMMMA – L (2001) y OCL – 2.0 (2003).

ApEM-L se expresa de manera tal que puede ser utilizado con cualquier proceso de desarrollo. Aunque sugiere la utilización de procesos de desarrollo iterativos, incrementales y basados en prototipos, que permitan la modelación de sistemas orientados a objetos. Este lenguaje se ha dividido en varias vistas, que modela cada una de las construcciones que representan un aspecto del sistema. La división ha sido sobre la base de las áreas conceptuales: estructura lógica, comportamiento dinámico y gestión del modelo.[12]

¿Por qué se selecciona el lenguaje de modelado ApEM-L?

Incorpora los elementos más significativos de extensiones anteriores como OMMMA – L (2001) y a su vez respeta lo establecido por el estándar OCL (2003), de aquí se logra una extensión consistente y escalable en el tiempo, lo que produce las siguientes ventajas:

- 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.

ApEM-L incorpora la vista de presentación como aporte fundamental a la modelación de este tipo de aplicaciones con dos diagramas que permiten definir la estructura para la navegación y la estructura para la presentación.[12]

Tabla 3: Vistas y diagramas de ApEM – L.

Área	Vista	Diagramas	Conceptos Principales
Estructura Lógica	Vista Estática	Diagrama de clases	Clase, asociación, generalización, dependencia, realización, interfaz.
		Diagrama casos de uso	Caso de uso, actor, asociación, extensión, inclusión, generalización.
	Vista de Arquitectura	Diagrama de componente	Componente, interfaz, dependencia, realización.
		Diagrama de Despliegue	Nodo, componente, dependencia, localización.
Comportamiento Dinámico	Vista de Comportamiento	Diagrama de actividad	Estado, actividad, transición de terminación, división, unión.
		Diagrama de secuencia	Interacción, objeto, mensaje, activación.
		Diagrama de colaboración	Colaboración, interacción, rol de colaboración, mensaje.
		Máquina de datos	Estado, evento, transición, acción.
Gestión del Modelo	Vista de Presentación	Diagrama de Estructura de navegación	Clases de navegación, elementos de navegación, composición, asociación, dependencia, uso.
		Diagrama de presentación	Clases de presentación, clases Frameset, asociaciones, componentes medias, paquetes.

2.2.5. Herramientas.

Las herramientas de la ingeniería del software suministran un soporte automático o semiautomático para los métodos. Hoy existen herramientas para soportar cada uno de los métodos mencionados. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte de desarrollo del software, llamado ingeniería del software asistida por computadora (en inglés, CASE).

2.2.5.1. Herramientas CASE.

Rational Rose.

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML (Booch, Rumbaugh y Jacobson) y soporta de forma completa la especificación del UML. Es una herramienta que propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar vistas creando un modelo completo que representa el dominio del problema y el sistema de software. Dentro de sus características está el **desarrollo iterativo** pues utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. **Genera código** en distintos lenguajes de programación a partir de un diseño en UML y proporciona mecanismos para realizar la denominada **Ingeniería Inversa**. [13]

Visual Paradigm.

¿Por qué se escoge Visual Paradigm?

Luego del estudio realizado se decide escoger Visual Paradigm debido a que es una herramienta/extensión UML CASE ampliamente integrada. 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 UML en Visual Studio, realizar ingeniería inversa desde código C# a diagramas de clases, generar código C# y generar documentación. Presenta además:

- Entorno de creación de diagramas.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs.

- Disponibilidad en múltiples plataformas.

Los tipos de diagramas que presenta son:

- Casos de Uso
- Estado
- Actividad
- Secuencia
- Clase
- Componentes
- Despliegue[14]

2.3. Conclusiones

En este capítulo de forma precisa hemos mencionado todas las teorías que en el ámbito tecnológico e informático, respaldan el trabajo que se presenta. Se han argumentado y explicado claramente las bases teóricas fundamentales del presente trabajo, como son: el Proceso Unificado de Desarrollo de Software (RUP) y el Lenguaje para la modelación de Aplicaciones Educativas Multimedia (ApEM-L), teniendo como base esencial que actualmente en el mundo informático, son opciones de gran eficiencia y que brindan una excelente calidad. Igualmente se hace referencia al Visual Paradigm como herramienta CASE.

Capítulo 3

Descripción y Construcción de la solución propuesta.

3.1. Introducción

A continuación se realiza una descripción y construcción de la solución propuesta mediante la utilización del **Lenguaje para la modelación de Aplicaciones Educativas Multimedia (ApEM-L)**.

Se desarrolla todo lo referente al Modelo de Dominio y las Vistas del Sistema. Se realiza la modelación del dominio para el Módulo Mediateca de la Colección Multisaber, identificando los principales conceptos que lo integran, así como los principales eventos que suceden en el entorno en el que trabaja el sistema, integrándose todo en el Diagrama del Modelo del Dominio. Se realizan varios diagramas en los que podemos encontrar el de Estructura de Navegación para hacer más comprensible la aplicación. Se identifican los requerimientos funcionales y no funcionales del sistema, la descripción de las vistas de presentación del sistema, se realiza la representación de los diagramas de Clases para una mejor comprensión de cómo están relacionadas las clases del sistema, el diagrama de Estructura Presentación para tener una idea de cómo están distribuidos los elementos en las pantallas de interacción con el usuario, Diagrama de Despliegue que representa la disposición de las instancias de componentes en ejecución en instancias de nodos conectados por enlaces de comunicación.

3.2. Modelo Dominio.

Un modelo conceptual explica los conceptos más significativos en un dominio del problema, es una herramienta de comunicación, con la cual se intenta comprender los conceptos importantes y sus relaciones. Un modelo conceptual representa cosas del mundo real, no componentes del software. (12)

Tiene como objetivo comprender las clases más importantes dentro del contexto, contribuir a la comprensión del contexto del sistema y por tanto también contribuir a la comprensión de los requisitos del sistema que se desprenden de este contexto. Deberían contribuir a una comprensión del problema que se supone que el sistema resuelve en relación de de su contexto. (13)

3.3. Descripción del Sistema propuesto.

En la actualidad, son muchos los procesos de desarrollo de software que existen. Con el pasar de los años, la Ingeniería de Software ha introducido y popularizado una serie de estándares para medir y certificar la calidad, tanto del sistema a desarrollar, como del proceso de desarrollo en sí. Un número creciente de herramientas automatizadas han surgido para ayudar a definir y aplicar un proceso de desarrollo de software efectivo.

Sin embargo, ¿cómo explicamos la alta incidencia de fallos en los proyectos de software? ¿Por qué existen tantos proyectos de software víctimas de retrasos, presupuestos sobregirados y con problemas de calidad? Tal vez suene ilógico pero, a pesar de los avances que ha dado la tecnología, aún existen procesos de producción informales, parciales y en algunos casos no confiables.

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.

El reemplazo de plataformas y tecnologías obsoletas, la compra de sistemas completamente nuevos, las modificaciones de todos o de casi todos los programas que forman un sistema, entre otras razones, llevan a desarrollar proyectos en calendarios sumamente ajustados y en algunos casos irreales; esto ocasiona que se omitan muchos pasos importantes en el ciclo de vida de desarrollo, entre estos, la definición de los **requerimientos**, que no son más que una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

Estudios realizados muestran que más del 53% de los proyectos de software fracasan por no realizar un estudio previo de requisitos. Otros factores como falta de participación del usuario, requerimientos incompletos y el cambio a los requerimientos, también ocupan sitios altos en los motivos de fracasos.

Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales.

Los **requerimientos funcionales** definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Los **requerimientos no funcionales** tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.(14)

3.3.1. Requerimientos Funcionales

R1. Mostrar presentación del módulo.

R1.1. Mostrar una imagen de cámara fotográfica en representación de la galería de imágenes, una cámara de video en representación de la galería de videos, un librito abierto en representación del glosario de términos, y una estante en representación de la información de interés.

R2. Dar acceso a la Galería de Imágenes.

R2.1. Mostrar las categorías de imágenes existentes.

R2.2. Mostrar las imágenes existentes en la categoría seleccionada por el usuario.

R2.3. Mostrar la imagen seleccionada por el usuario.

R2.4. Cambiar de categoría.

R3. Permitir la salida de la galería de Imágenes regresando al mismo lugar donde estaba navegando el usuario cuando decidió acceder a esta.

R4. Dar acceso a la galería de videos.

R4.1. Mostrar las categorías de los videos existentes.

R4.2. Mostrar los videos existentes en la categoría seleccionada por el usuario.

R4.3. Mostrar el video seleccionado por el usuario.

R4.4. Cambiar de categoría.

R5. Permitir la salida de la galería de videos regresando al mismo lugar donde estaba navegando el usuario cuando decidió acceder a esta.

R6. Dar acceso al glosario de términos.

R6.1. Mostrar las letras del abecedario.

R6.2. Mostrar una lista de las palabras existentes en la letra seleccionada por el usuario.

R6.3. Mostrar el significado de la palabra seleccionada por el usuario.

R6.4. Cambiar de categoría.

- R7. Permitir la salida del glosario de términos regresando al mismo lugar donde estaba navegando el usuario cuando decidió acceder a este.
- R8. Dar acceso a los artículos de interés.
R8.1. Mostrar los artículos seleccionados por el usuario.
- R9. Permitir la salida de los artículos de interés regresando al mismo lugar donde estaba navegando el usuario cuando decidió acceder a este.
- R10. Activar o desactivar el sonido al oprimir el botón correspondiente en dependencia de si éste está desactivado o activado.
- R11. Permitir la salida de la aplicación.
R11.1. Mostrar una ventana de confirmación de salida.
- R12. Mostrar créditos una vez que el usuario confirme que desea salir.
- R13. Permitir ir a la pantalla principal de la multimedia presionando el botón de servicio para esto.
- R14. Permitir la navegación por los demás módulos de la multimedia.
- R15. Ofrecer ayuda sobre la página actual que se encuentra el usuario.
- R16. Buscar los contenidos que contiene el texto deseado.
- R17. Mostrar información acerca de un tema seleccionado por el usuario.

3.3.2 Requerimientos No Funcionales

- **Requerimientos de software.**

- ✓ Sistema Operativo Linux: Ubuntu 7.10.

- **Requerimientos de hardware.**

- ✓ Pentium 233 MHz (*Recomendado: Pentium 500MHz o mejor*)
- ✓ 64 MB RAM (*Recomendado: 128 MB RAM o mejor*)
- ✓ 1 GB espacio en Disco Duro
- ✓ Lector CD
- ✓ Dispositivo de audio
- ✓ Soporte de video que admita resolución de la menos 800x600 y 24 bits

- ✓ Dispositivo de red de al menos 10 megabits.

- **Requerimientos de apariencia o interfaz externa.**
 - ✓ En la pantalla de presentación del módulo aparecerán los iconos de Galería de imágenes, Galería de videos, Glosario de términos y el de Artículos de interés.
 - ✓ El texto de los menú o identificadores del módulo serán para los botones de servicios de color negro con iluminación blanca, además de un fondo blanco; para los identificadores de color blanco con un fondo azul.
 - ✓ En la Galería de imágenes, la Galería de videos, el texto que presenta cada uno será de color negro, con iluminación blanca, además con un fondo gris.
 - ✓ Los botones de servicios (Regresar a la pagina principal, Imprimir, Ayuda, Buscar, Acerca de..., Activar/Desactivar música, Salir) se mostrarán siempre visibles al usuario.
 - ✓ Las medias a visualizar siempre se mostrarán en la misma área de interfaz, para evitar la pérdida del usuario.
 - ✓ El idioma utilizado será el español, apoyándose en el uso de las palabras técnicas propias del tema de la aplicación.
 - ✓ El icono identificador de los botones de servicio serán de color verde y blanco con una estrellita amarilla en el centro, con iluminación blanca, además de mostrar el nombre del servicio y la estrellita moverse al pasarle por encima con el mouse.

- **Requerimientos de seguridad.**
 - ✓ A este módulo puede acceder cualquier tipo de usuario.

- **Requerimientos de usabilidad.**
 - ✓ Los usuarios que utilizarán el sistema deberán tener al menos un conocimiento básico del manejo de la computadora, así como conocimiento previo también del trabajo con sistemas operativos visuales.

- **Requerimientos de portabilidad.**
 - ✓ El software podrá ser ejecutado bajo los sistemas operativos SO Windows 9x, Me, NT, XP; Mac OS 9 o superior y Linux en sus distintas versiones, según decreto en el artículo 3.390 “todos los desarrollos deben seguir los estándares abiertos y los códigos

abiertos”, debiendo existir en el SO Linux el emulador para aplicaciones multimedia requerido para la ejecución de este tipo de aplicaciones.

- **Requerimientos de ayuda y documentación en línea.**

- ✓ El producto contará con un fichero texto nombrado Léeme.txt que permitirá especificar las necesidades de software antes de instalar el producto y el algoritmo a seguir para la instalación de este.

3.4. Subsistemas del módulo Mediateca.

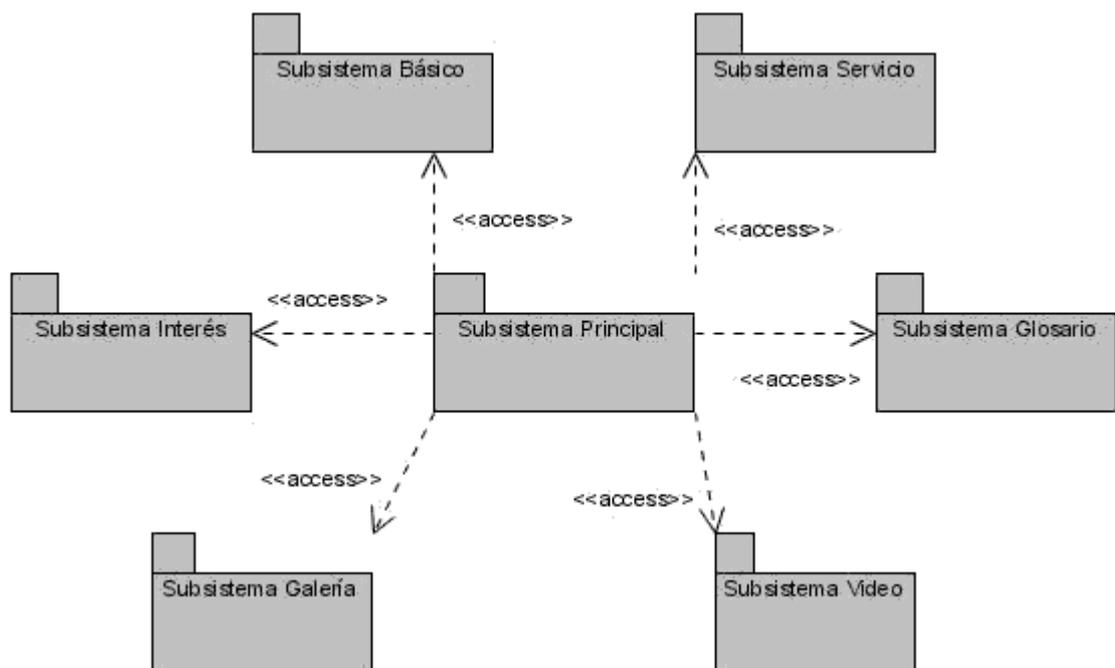


Ilustración 3: Vista Gestión del Modelo.

3.4.1. Vistas por Subsistemas.

1. Subsistema Básico.
 1. Vista Presentación Mediateca.
2. Subsistema Galería.
 1. Vista Galería_Categoría_Imágenes.
 2. Vista Galería_Selección_Categoría_Imágenes.

3. Subsistema Video.
 1. Vista Video_Categoría_Video.
 2. Vista Video_Selección_Categoría_Video.
4. Subsistema Glosario.
 1. Vista Glosario.
5. Subsistema Artículos.
 1. Vista Interés.
6. Subsistema de Servicios.
 1. Vista Salir.
 2. Vista Confirmar Salida.
 3. Vista Imprimir.
 4. Vista Buscar.
 5. Vista Acerca de...
 6. Vista Ayuda.
 7. Act/Desc Sonido.
7. Subsistema Principal.
 1. Vista Temas.
 2. Vista Ejercicios.
 3. Vista Juegos.
 4. Vista Mediateca.
 5. Vista Registro.
 6. Vista Maestro.

3.4.2. Justificación del actor.

Usuario: Es la persona que interactúa con la aplicación beneficiándose de ella, va a adquirir conocimientos mediante las consultas de información (puede ser un maestro, un alumno o invitado).

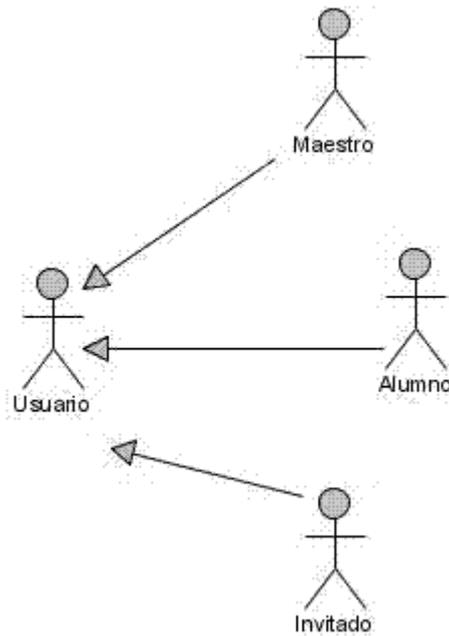


Ilustración 4: Diagrama de Relación de los Usuarios.

3.4.3. Descripción textual de las Vistas.

Tabla 4: Descripción Textual de la Vista de Presentación Módulo Mediateca.

Descripción Textual de la Vista de Presentación	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario de la aplicación seleccionar una de las opciones de trabajo dentro de la aplicación en su pantalla principal.
Objetivos pedagógicos	-----
Resumen	La vista se inicia cuando el usuario da clic en el elemento de representación de la Mediateca en el subsistema de presentación del software, se muestran las opciones de trabajo y termina cuando el usuario selecciona alguna de estas.

dentro de la aplicación.	<p>ejecuta la Vista Glosario y se culmina la Vista Mediateca.</p> <p>10. Si el usuario seleccionó la opción Galería de Imágenes se ejecuta la Vista Galería y se culmina la Vista Mediateca.</p> <p>11. Si el usuario seleccionó la opción Galería de Videos se ejecuta la Vista Videos y se culmina la Vista Mediateca.</p> <p>12. Si el usuario seleccionó la opción Información de Interés se ejecuta la Vista Interés y se culmina la Vista Mediateca.</p>	<p>9.1 Se ilumina la opción Glosario.</p> <p>9.2 Se reproduce la banda_selección.</p> <p>10.1 Se ilumina la opción Fotos.</p> <p>10.2 Se reproduce la banda_selección.</p> <p>11.1 Se ilumina la opción Videos.</p> <p>11.2 Se reproduce la banda_selección.</p> <p>12.1 Se ilumina la opción Información de Interés.</p> <p>12.2 Se reproduce la banda_selección.</p>	
Cursos Alternos de los Eventos			
Acción	Curso Alterno		
-	-		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo de la mediateca.	Existente
	I_02	Imagen de una cámara de video.	Existente

	I_03	Imagen de una cámara fotográfica.	Existente
	I_04	Imagen de un libro abierto.	Existente
	I_05	Imagen de un fichero (artículos).	Existente
	I_06	Imagen de una mesa de 4 patas donde está apoyado el librito.	Existente
	I_07	Imagen de una mesa de 3 patas donde va apoyada la cámara fotográfica.	Existente
	I_08	Imagen de una ventana	Existente
Video o Animación	A_01	Animación en donde la mascota 01 (trompo) conversa con el usuario y se desplaza a la parte inferior de la vista.	Existente
	A_02	Animación que ocurre cuando se pone el mouse encima de la mascota 01, esta se mueve de derecha a izquierda en el mismo lugar.	Existente
	A_03	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de la galería de imágenes es decir una cámara fotográfica cambiando el lente de la misma de color.	Existente
	A_04	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de la galería de video: una cámara de video cambiando el lente de la misma de color.	Existente
	A_05	Animación que ocurre cuando ponemos el mouse encima del elemento representativo del	Existente

		glosario de términos: el librito abierto cambiando de color algunas de sus partes.	
	A_06	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de los artículos de interés es decir un archivo cambiando de color.	Existente
	A_07	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de ir la pantalla principal.	Existente
	A_08	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de activar/desactivar la música.	Existente
	A_09	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de ofrecer ayuda.	Existente
	A_10	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de buscar.	Existente
	A_11	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de acerca de.	Existente
	A_12	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de salir.	Existente
Sonido	BS_01	Banda sonora de una canción de Venezuela	En localización
Texto	T_01	Texto identificador de la Vista "Mediateca".	Existente

	T_02	Texto que sale al pasar el mouse por encima del elemento identificador de la Galería de Imágenes “Galería de Imágenes”.	Existente
	T_03	Texto que sale al pasar el mouse por encima del elemento identificador de la Galería de Videos “Galería de Videos”.	Existente
	T_04	Texto que sale al pasar el mouse por encima del elemento identificador del Glosario de Términos “Glosario de Términos”.	Existente
	T_05	Texto que sale al pasar el mouse por encima del elemento identificador de la Información de Interés “Información de Interés”.	Existente
	T_06	Texto que aparece encima del identificado del Módulo Tema “Temas”.	Existente
	T_07	Texto que aparece encima del identificador del Módulo Ejercicios “Ejercicios”.	Existente
	T_08	Texto que aparece encima del identificador del Módulo Registro “Registro”.	Existente
	T_09	Texto que aparece encima del identificador del Módulo Juegos “Juegos”.	Existente
	T_10	Texto que aparece encima del identificador del Módulo Maestro “Maestro”.	Existente
	T_11	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de ir a la Pantalla Principal “Ir a la Pantalla Principal”.	Existente
	T_12	Texto que aparece en un cuadro de texto al	Existente

		pasar el mouse por encima del botón de Activar o Desactivar Sonido “Act/Des Sonido”.	
	T_13	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Buscar “Buscar”.	Existente
	T_14	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Ayuda “Ayuda”.	Existente
	T_15	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Salir “Salir”.	Existente
	T_16	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Imprimir “Imprimir”.	Existente
	T_17	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Acerca de “Acerca de...”.	Existente
	T_18	Texto que aparece en un cuadro de texto al poner el mouse encima de la mascota 01” Aquí estoy”.	Existente
	T_19	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de volver a la pantalla principal “Regresar a la pantalla Principal”.	Existente
Reglas pedagógicas			
-			

Tabla 5: Descripción Textual de la Vista Galería_Categoría_Imágenes.

Descripción Textual de la Vista Galería_Categoría_Imágenes.		
Actores de la Vista de Presentación	Usuario	
Propósito	Permitir al usuario una vez seleccionada la opción de trabajo, poder ver las categorías de imágenes que existen en la multimedia.	
Objetivos pedagógicos	-----	
Resumen	La vista se inicia cuando el usuario da clic en el elemento representativo de la Galería de Imágenes en la Vista Principal de la Mediateca, se muestran las diferentes categorías de imágenes existentes y culmina cuando el usuario decide ver las imágenes de alguna de las categorías.	
Vistas asociadas	Vista "Presentación_Mediateca" Vista "Galería_Selección_Categoría_Imágenes".	
Referencias	R2, R3.	
Precondiciones.	-	
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	Elementos de la Vista
1. El usuario da clic en el elemento representativo de la Galería de Imágenes.	2. Se muestra la imagen de fondo de la galería. 3. Se muestran los elementos representativos de las categorías de imágenes	

<p>7. El usuario da clic en un elemento representativo de una de las categorías.</p> <p>9. El usuario da clic en el elemento representativo de volver atrás para cambiar de categoría.</p> <p>11. El usuario da clic en el icono representativo volver atrás para llegar a la Vista "Presentación_Mediateca".</p>	<p>existentes.</p> <p>4. Se presentan las animaciones de la mascota de la aplicación.</p> <p>5. Se muestran las opciones del Subsistema Paquete Común.</p> <p>6. Se muestran las opciones del Subsistema de Servicio.</p> <p>8. Se muestran las imágenes correspondientes a dicha categoría, culmina la Vista Categorías_Imágenes.</p> <p>10. Se vuelve a la Vista Categorías_Imágenes para cambiar de categoría.</p> <p>12. Se vuelve a la Vista "Presentación_Mediateca".</p>		
Cursos Alternos de los Eventos			
Acción	Curso Alterno		
-	-		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado

Imagen	I_01	Imagen de fondo de la Mediateca.	Existente	
	I_010	Imagen representativa de cada carpeta de imágenes	Existente	
Video o Animación	A_01	Animación en donde la mascota 01 (trompo) se mueve.	Existente	
	A_07	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de ir la pantalla principal.	Existente	
	A_08	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de activar/desactivar la música.	Existente	
	A_09	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de ofrecer ayuda.	Existente	
	A_10	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de buscar.	Existente	
	A_11	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de acerca de.	Existente	
	A_12	Animación que ocurre cuando ponemos el mouse encima del elemento representativo de salir.	Existente	
	Sonido	BS_01	Música representativa de Venezuela	En localización.

Texto	T_01	Texto identificador de la Vista “Mediateca”.	Existente
	T_02	Texto que sale al pasar el mouse por encima del elemento identificador de la Galería de Imágenes “Galería de Imágenes”.	Existente
	T_03	Texto que aparece al pasar el mouse por encima del elemento identificador de la Galería de Videos “Galería de Videos”.	Existente
	T_04	Texto que sale al pasar el mouse por encima del elemento identificador del Glosario de Términos “Glosario de Términos”.	Existente
	T_05	Texto que sale al pasar el mouse por encima del elemento identificador de la Información de Interés “Información de Interés”.	Existente
	T_06	Texto que aparece encima del identificador del Módulo Tema “Temas”.	Existente
	T_07	Texto que aparece encima del identificador del Módulo Ejercicios “Ejercicios”.	Existente
	T_08	Texto que aparece encima del identificador del Módulo Registro “Registro”.	Existente
	T_09	Texto que aparece encima del identificador del Módulo Juegos “Juegos”.	Existente
	T_10	Texto que aparece encima del identificador del Módulo Maestro “Maestro”.	Existente
	T_11	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de ir a la Pantalla Principal “Ir a la Pantalla Principal”.	Existente

	T_12	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Activar o Desactivar Sonido “Act/Des Sonido”.	Existente
	T_13	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Buscar “Buscar”.	Existente
	T_14	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Ayuda “Ayuda”.	Existente
	T_15	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Salir “Salir”.	Existente
	T_16	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Imprimir “Imprimir”.	Existente
	T_17	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Acerca de “Acerca de...”.	Existente
	T_19	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de volver a la pantalla principal “Regresar a la pantalla anterior”.	Existente
	T_20	Texto que aparece en la parte inferior derecha de la vista mostrando el nombre de la carpeta de imágenes donde se posiciona el mouse.	Existente
Reglas pedagógicas			

Tabla 6: Descripción Textual de la Vista Galería_Selección_Categoría_Imágenes.

Descripción Textual de la Vista Galería_Selección_Categoría_Imágenes.		
Actores de la Vista de Presentación	Usuario	
Propósito	Permitir al usuario una vez seleccionado la opción de trabajo, poder ver las imágenes de la categorías que existen en la multimedia.	
Objetivos pedagógicos	-----	
Resumen	La vista se inicia cuando el usuario da clic en una de las categorías existentes, se nos muestran las diferentes imágenes, la vista culmina cuando el usuario decide cambiar de categoría.	
Vistas asociadas	Vista "Presentación_Mediateca", Vista "Galería_Categoría_Imágenes".	
Referencias	R2, R3.	
Precondiciones .	-	
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	Elementos de la Vista
1. El usuario da clic en el elemento representativo de una de las categorías.	2. Se muestra la imagen de fondo de la galería. 3. Se muestran las imágenes existentes en dicha categoría. 4. Se presentan las animaciones de la mascota de la aplicación. 5. Se muestran las opciones del	

<p>7. El usuario da clic en una de las imágenes.</p> <p>9. El usuario da clic en el elemento representativo de volver atrás para ver otra imagen.</p> <p>11. El usuario da clic en el elemento representativo de volver atrás para cambiar de categorías.</p> <p>13. El usuario da clic en el icono representativo volver a atrás para llegar a la Vista "Presentación_Mediateca".</p>	<p>Subsistema Paquete Común.</p> <p>6. Se muestran las opciones del Subsistema de Servicio.</p> <p>8. Se muestra dicha imagen en el toldo.</p> <p>10. Se vuelve a la Vista Galería_Imágenes_Selección_Categoría para ver otra imagen existente.</p> <p>12. Se vuelve a la Vista Categoría Imágenes, culmina la Vista "Galería _Selección_Categoría_Imágenes".</p> <p>14. Se vuelve a la Vista "Presentación_Mediateca", culmina la Vista "Galería_Selección_Categoría_Imágenes".</p>		
Cursos Alternos de los Eventos			
Acción	Curso Alterno		
-	-		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo de la Mediateca.	Existente

Video o Animación			
Sonido	BS_01	Canción representativa de Venezuela	En localización
Texto	T_11	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de ir a la Pantalla Principal “Ir a la Pantalla Principal”.	Existente
	T_12	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Activar o Desactivar Sonido “Act/Des Sonido”.	Existente
	T_13	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Buscar “Buscar”.	Existente
	T_14	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Ayuda “Ayuda”.	Existente
	T_15	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Salir “Salir”.	Existente
	T_16	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Imprimir “Imprimir”.	Existente
	T_17	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Acerca de “Acerca de...”.	Existente

	T_21	Texto que aparece en la parte inferior derecha de la vista mostrando el nombre de la imagen donde se tiene el mouse puesto.	Existente
	T_22	Texto que aparece en la pantalla de presentación como descripción de la imagen seleccionada por el usuario.	Existente
Reglas pedagógicas			

Tabla 7: Descripción Textual de la Vista Galería_Categoría_Video.

Descripción Textual de la Vista Galería_Categoría_Video.	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario una vez seleccionado la opción de trabajo, poder ver las categorías de videos que existen en la multimedia.
Objetivos pedagógicos	-----
Resumen	La vista se inicia cuando el usuario da clic en el elemento representativo de la Galería de Video en la Vista Principal de la Mediateca, se muestran las diferentes categorías de videos existentes y culmina cuando el usuario decide ver los videos de alguna de las categorías.
Vistas asociadas	Vista "Presentación_Mediateca" Vista "Galería_Categoría_Video".
Referencias	R4, R5.

Precondiciones .	-	
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	Elementos de la Vista
<p>1. El usuario da clic en el elemento representativo de la Galería de Video (cámara video).</p> <p>7. El usuario da clic en un elemento representativo de una de las categorías.</p> <p>9. El usuario da clic en el elemento representativo de volver atrás para cambiar de categoría.</p> <p>11. El usuario da clic en el icono representativo volver a atrás para llegar a</p>	<p>2. Se muestra la imagen de fondo de la galería.</p> <p>3. Se muestran los elementos representativos de las categorías de videos existentes.</p> <p>4. Se presentan las animaciones de la mascota de la aplicación.</p> <p>5. Se muestran las opciones del Subsistema Paquete Común.</p> <p>6. Se muestran las opciones del Subsistema de Servicio.</p> <p>8. Se muestran los videos correspondientes a dicha categoría, culmina la Vista "Categorías_Video".</p> <p>11. Se vuelve a la Vista "Categorías_Video" para cambiar de categoría.</p> <p>12. Se vuelve a la Vista "Presentación_Mediateca"</p>	

la Vista "Presentación_Mediateca"			
Cursos Alternos de los Eventos			
Acción		Curso Alterno	
-		-	
Prioridad		Crítica	
Mejoras		-	
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo de la Mediateca.	Existente
	I_11	Imagen de fondo de la Galería de Video.	Existente
	I_12	Imagen de representación en las carpetas de cada video.	Existente
Video o Animación			
Sonido	BS_01	Canción representativa de Venezuela	En localización
Texto	T_01	Texto identificador de la Vista "Mediateca".	Existente
	T_11	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de ir a la Pantalla Principal "Ir a la Pantalla Principal".	Existente
	T_12	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Activar	Existente

		o Desactivar Sonido “Act/Des Sonido”.	
	T_13	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Buscar “Buscar”.	Existente
	T_14	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Ayuda “Ayuda”.	Existente
	T_15	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Salir “Salir”.	Existente
	T_16	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Imprimir “Imprimir”.	Existente
	T_17	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Acerca de “Acerca de...”.	Existente
	T_23	Texto que aparece en la parte inferior derecha de la vista mostrando el nombre de la carpeta de videos donde se tiene el mouse puesto.	Existente
Reglas pedagógicas			

Tabla 8: Descripción Textual de la Vista Galería_Selección_Categoría_Video.

Descripción Textual de la Vista Glosario de Términos.		
Actores de la Vista de Presentación	Usuario	
Propósito	Permitir al usuario una vez seleccionado la opción de trabajo, poder ver los de videos que existen en la multimedia de una categoría determinada.	
Objetivos pedagógicos	-----	
Resumen	La vista se inicia cuando el usuario da clic en una de las categorías existentes, se nos muestran los diferentes videos, la vista culmina cuando el usuario decide cambiar de categoría.	
Vistas asociadas	Vista "Presentación_Mediateca" Vista "Galería_Categoría_Video".	
Referencias	R4, R5.	
Precondiciones .	-	
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	Elementos de la Vista
1. El usuario da clic en el elemento representativo de una de las categorías.	2. Se muestra la imagen de fondo de la galería. 3. Se muestran los videos existentes en dicha categoría. 4. Se presentan las	

<p>7. El usuario da clic en uno de los videos.</p> <p>9. El usuario da clic en el elemento representativo de volver atrás para ver otro video.</p> <p>11. El usuario da clic en el elemento representativo de volver atrás para cambiar de categoría.</p> <p>13. El usuario da clic en el icono representativo volver a atrás para llegar a la Vista "Presentación_Mediateca"</p>	<p>animaciones de la mascota de la aplicación.</p> <p>5. Se muestran las opciones del Subsistema Paquete Común.</p> <p>6. Se muestran las opciones del Subsistema de Servicio.</p> <p>8. Se muestra dicho video en la pantalla de presentación.</p> <p>10. Se vuelve a la Vista "Galería_Video_Selección - Categoría" para ver otro video existente.</p> <p>12. Se vuelve a la Vista Categoría Imágenes, culmina la Vista "Galería _Selección_ Categoría_Video".</p> <p>14. Se vuelve a la Vista "Presentación_Mediateca", culmina la Vista "Galería_Selección_ Categoría_Video".</p>	
Cursos Alternos de los Eventos		
Acción	Curso Alterno	
-	-	

Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo de la Mediateca.	Existente
	I_13	Imagen de fondo de la Galería de Video.	Existente
Video o Animación	A_01	Animación en donde la mascota 01 (trompo) se mueve.	Existente
Sonido	BS_01	Canción representativa de Venezuela	En localización
Texto	T_11	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de ir a la Pantalla Principal “Ir a la Pantalla Principal”.	Existente
	T_12	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Activar o Desactivar Sonido “Act/Des Sonido”.	Existente
	T_13	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Buscar “Buscar”.	Existente
	T_14	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Ayuda “Ayuda”.	Existente
	T_15	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Salir “Salir”.	Existente

	T_16	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Imprimir "Imprimir".	Existente
	T_17	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Acerca de "Acerca de...".	Existente
	T_24	Texto que aparece en la parte inferior derecha de la vista mostrando el nombre del video donde se tiene el mouse puesto.	Existente
Reglas pedagógicas			

Tabla 9: Descripción Textual de la Vista Glosario de Términos.

Descripción Textual de la Vista Glosario de Términos.	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario conocer el significado de las palabras que tenga duda que estén contenidas en la multimedia.
Objetivos pedagógicos	-----
Resumen	La vista se inicia cuando el usuario da clic en el elemento representativo del Glosario de Términos.
Vistas asociadas	Vista "Presentación_Mediateca",

Referencias	R6, R7.	
Precondiciones .	-	
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	Elementos de la Vista
1. El usuario da clic en el elemento representativo del glosario de Términos (librito abierto)	2. Se muestra la imagen de fondo del glosario. 3. Se presentan las animaciones de la mascota de la aplicación. 4. Se muestran las opciones del Subsistema Paquete Común. 5. Se muestran las opciones del Subsistema de Servicio. 6. Se muestra un área donde se encuentran las letras del alfabeto ordenadas. 7. Se muestra un ComboBox donde se mostrarán todas las palabras que se tiene en la Multimedia. 8. Se muestra un ListBox donde aparecerán los significados de las	

<p>9. El usuario da clic en una de las letras del alfabeto.</p> <p>11. El usuario da clic en una de las palabras.</p> <p>13. El usuario da clic en el icono representativo volver atrás para llegar a la Vista "Presentación_Mediateca".</p>	<p>palabras seleccionadas.</p> <p>10. Se muestra en el ComboBox una lista de las palabras existentes en la Multimedia que comienzan con dicha letra.</p> <p>12. Se muestra en el ListBox el significado de dicha palabra, si la palabra contiene imagen o video, también se muestra aquí.</p> <p>14. Se va a la Vista "Presentación_Mediateca" y culmina la Vista "Glosario_Términos".</p>		
Cursos Alternos de los Eventos			
Acción	Curso Alterno		
-	-		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo de la Mediateca.	Existente
	I_02	Imagen de fondo de la Galería de Video.	Existente

Video o Animación	A_01	Animación en donde la mascota 01 (trompo) se mueve.	Existente
Sonido	BS_01	Canción representativa de Venezuela	En localización
Texto	T_11	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de ir a la Pantalla Principal “Ir a la Pantalla Principal”.	Existente
	T_12	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Activar o Desactivar Sonido “Act/Des Sonido”.	Existente
	T_13	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Buscar “Buscar”.	Existente
	T_14	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Ayuda “Ayuda”.	Existente
	T_15	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Salir “Salir”.	Existente
	T_16	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Imprimir “Imprimir”.	Existente
	T_17	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Acerca de “Acerca de...”.	Existente
Reglas pedagógicas			

Tabla 10: Descripción Textual de la Vista Interés.

Descripción Textual de la Vista Glosario de Términos.		
Actores de la Vista de Presentación	Usuario	
Propósito	Permitir al usuario conocer alguna información de interés que se muestre con respecto al contenido de la multimedia	
Objetivos pedagógicos	-----	
Resumen	La vista se inicia cuando el usuario da clic en el elemento representativo de los Artículos de Interés.	
Vistas asociadas	Vista "Presentación_Mediateca"	
Referencias	R8, R9.	
Precondiciones	-	
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	Elementos de la Vista
1. El usuario da clic en el elemento representativo de los Artículos de Interés.	2. Se muestra la imagen de fondo de la vista. 3. Se presentan las animaciones de la mascota de la aplicación. 4. Se muestran las opciones del Subsistema Paquete Común.	

<p>9. El usuario da clic en uno de los artículos.</p> <p>11. El usuario da clic en el icono representativo volver atrás para llegar a la Vista "Presentación_Mediateca".</p>	<p>5. Se muestran las opciones del Subsistema de Servicio.</p> <p>6. Se muestra un área donde se encuentran los principales artículos de la multimedia.</p> <p>7. Se muestra un ListBox donde aparecerá el artículo seleccionado por el usuario.</p> <p>10. Se muestra en el ListBox el artículo seleccionado por el usuario.</p> <p>12. Se va a la Vista "Presentación_Mediateca" y culmina la Vista "Interés".</p>		
Cursos Alternos de los Eventos			
Acción	Curso Alterno		
-	-		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo de la Mediateca.	Existente

	I_14	Imagen de fondo de los artículos de interés.	Existente
Video o Animación			
Sonido	BS_01	Canción representativa de Venezuela	En localización
Texto	T_11	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de ir a la Pantalla Principal "Ir a la Pantalla Principal".	Existente
	T_12	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Activar o Desactivar Sonido "Act/Des Sonido".	Existente
	T_13	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Buscar "Buscar".	Existente
	T_14	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Ayuda "Ayuda".	Existente
	T_15	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Salir "Salir".	Existente
	T_16	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Imprimir "Imprimir".	Existente
	T_17	Texto que aparece en un cuadro de texto al pasar el mouse por encima del botón de Acerca de "Acerca de...".	Existente

	T_18	Texto representativo de los artículos de interés que se encuentran en la multimedia.	Existente
	T_19	Texto representativo de la sección donde se encuentran los artículos de interés	Existente
Reglas pedagógicas			

3.5. Diagrama de Estructura de Presentación.

Diagrama de Estructura de Presentación extiende la semántica de los diagramas de clases para poder utilizarlos en la representación de la estructura de las presentaciones, incorporando los estereotipos restrictivos de clases: estáticos e interacción y un árbol jerárquico a partir de estos estereotipos que agrupa los componentes visuales de acuerdo a su función en la interfaz a la que pertenece.[12]

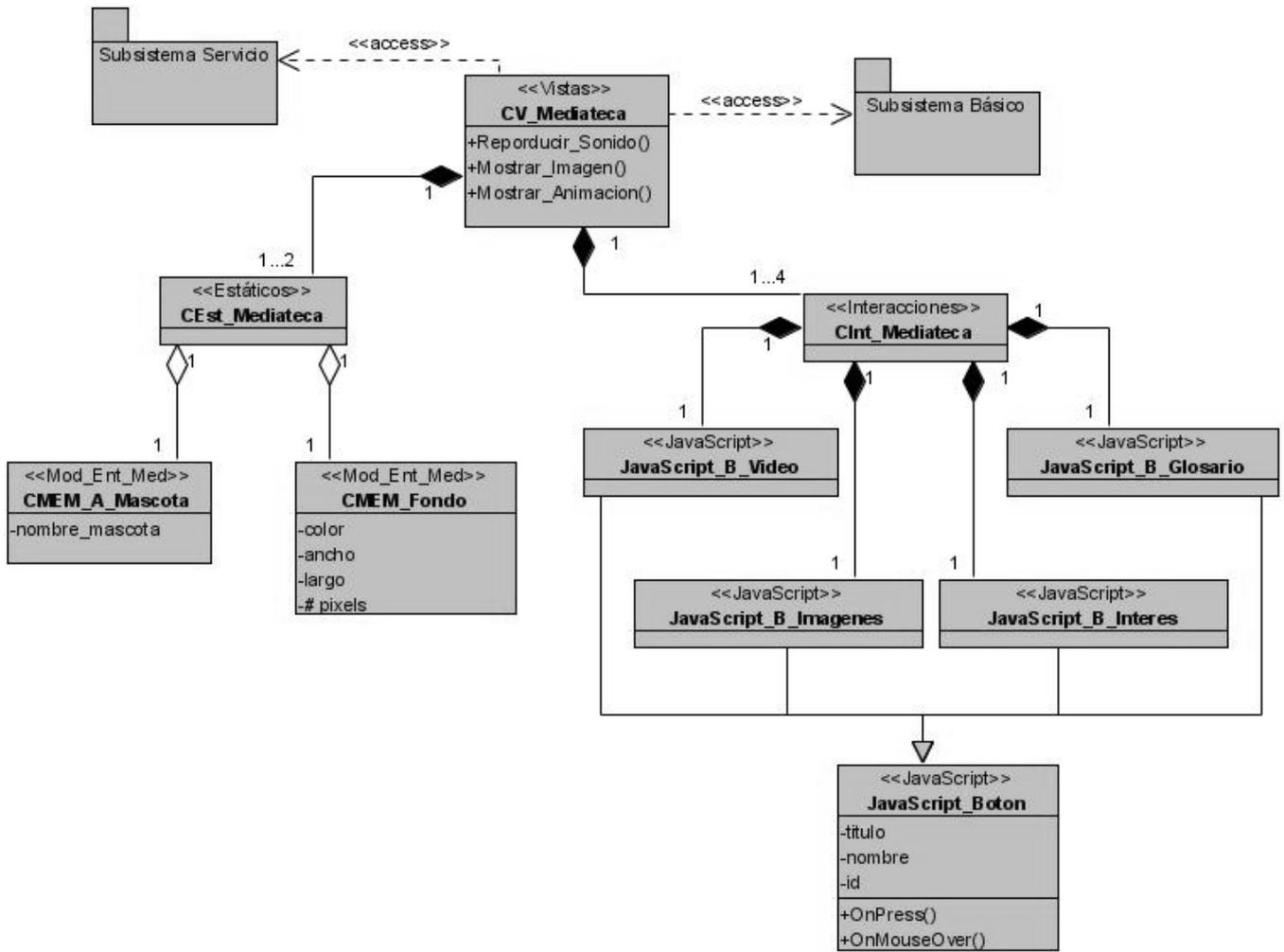


Ilustración 5: Diagrama de Estructura de Presentación de la Vista Presentación.

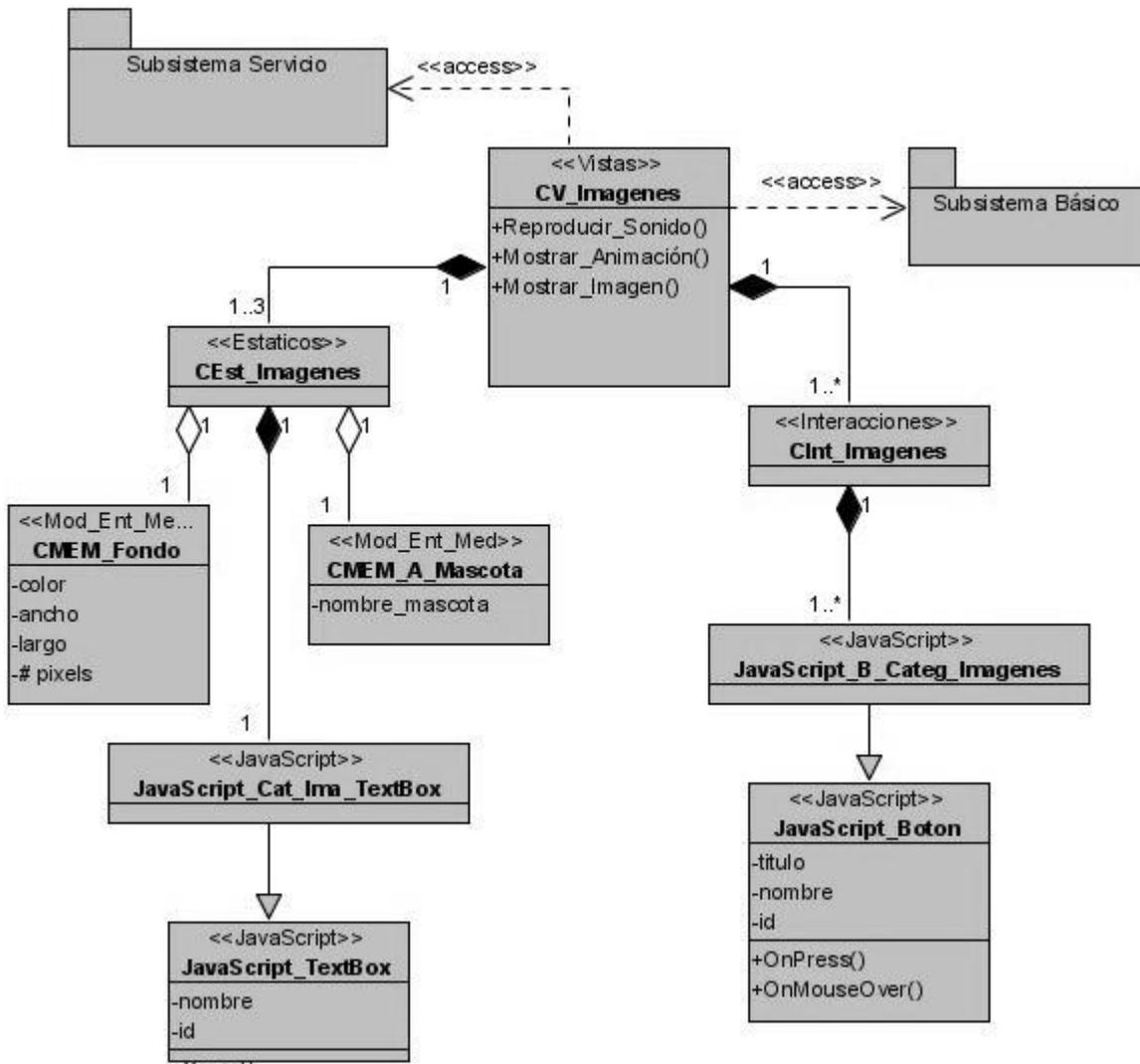


Ilustración 6: Diagrama de Estructura de Presentación de la Vista_Galería_Categoría_Imágenes.

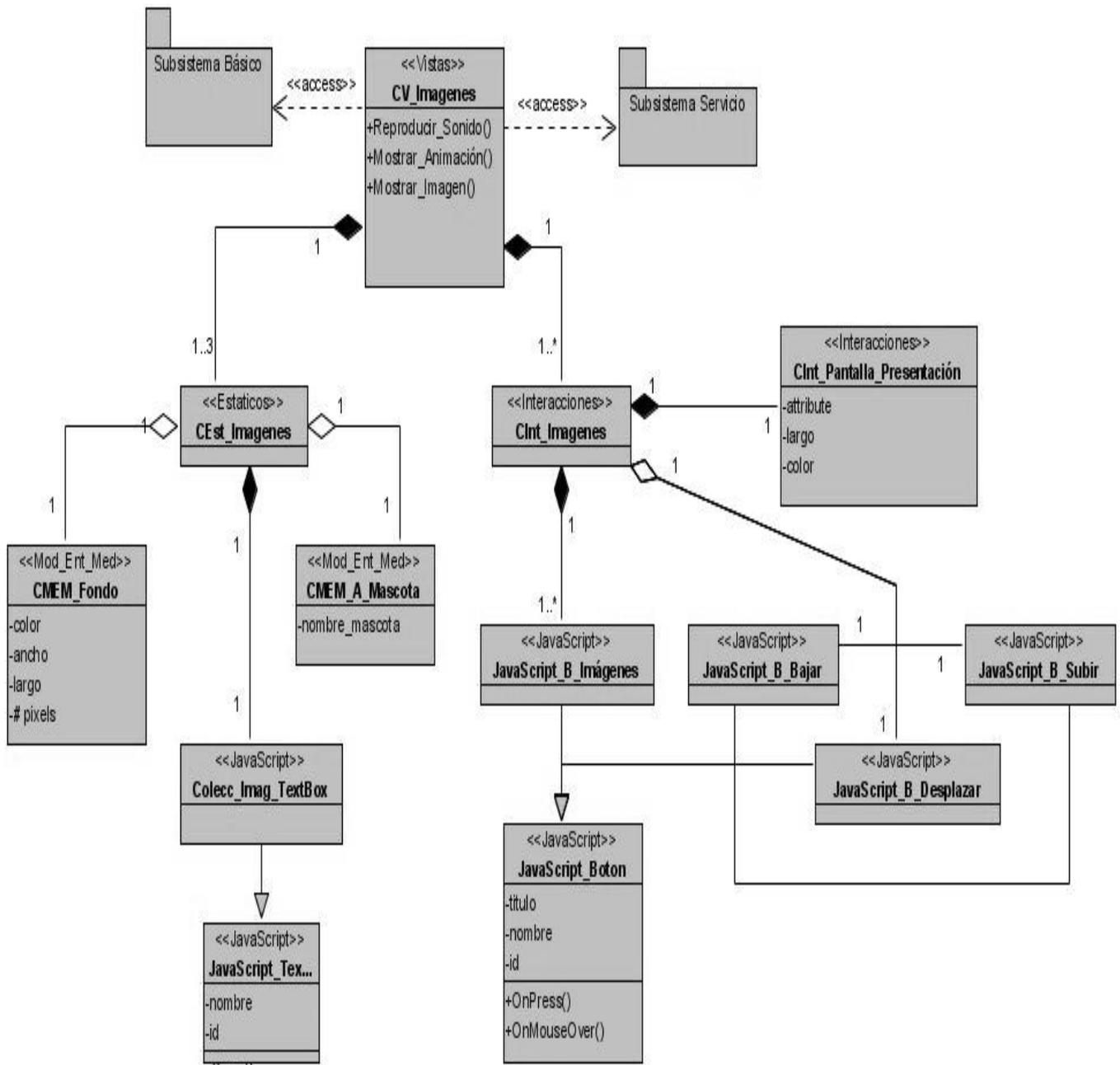


Ilustración 7: Diagrama de Estructura de Presentación de la Vista_Galeria_Selección_Categoría_Imágenes.

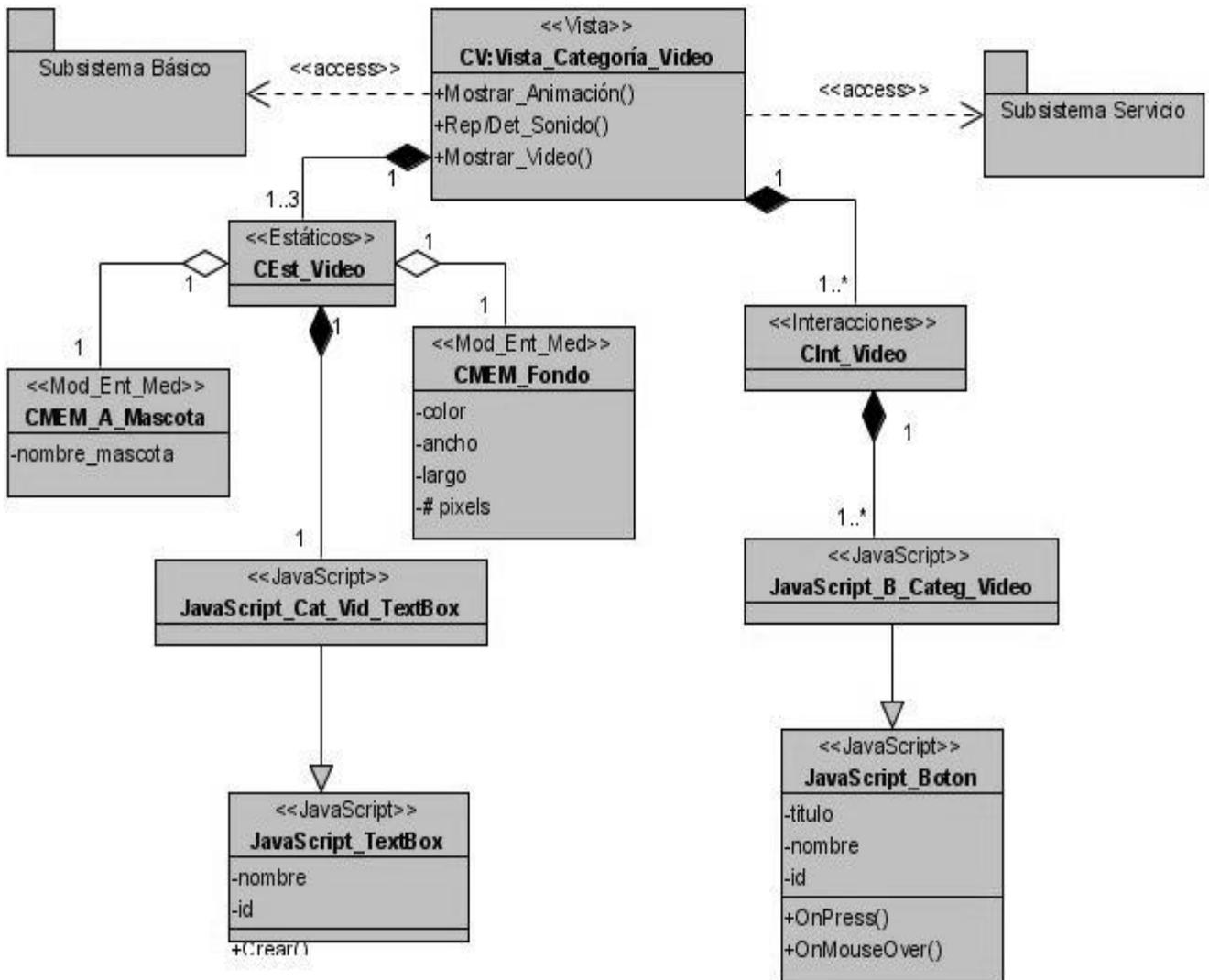


Ilustración 8: Diagrama de Estructura de Presentación de la Vista_Galería_Categoría_Videos.

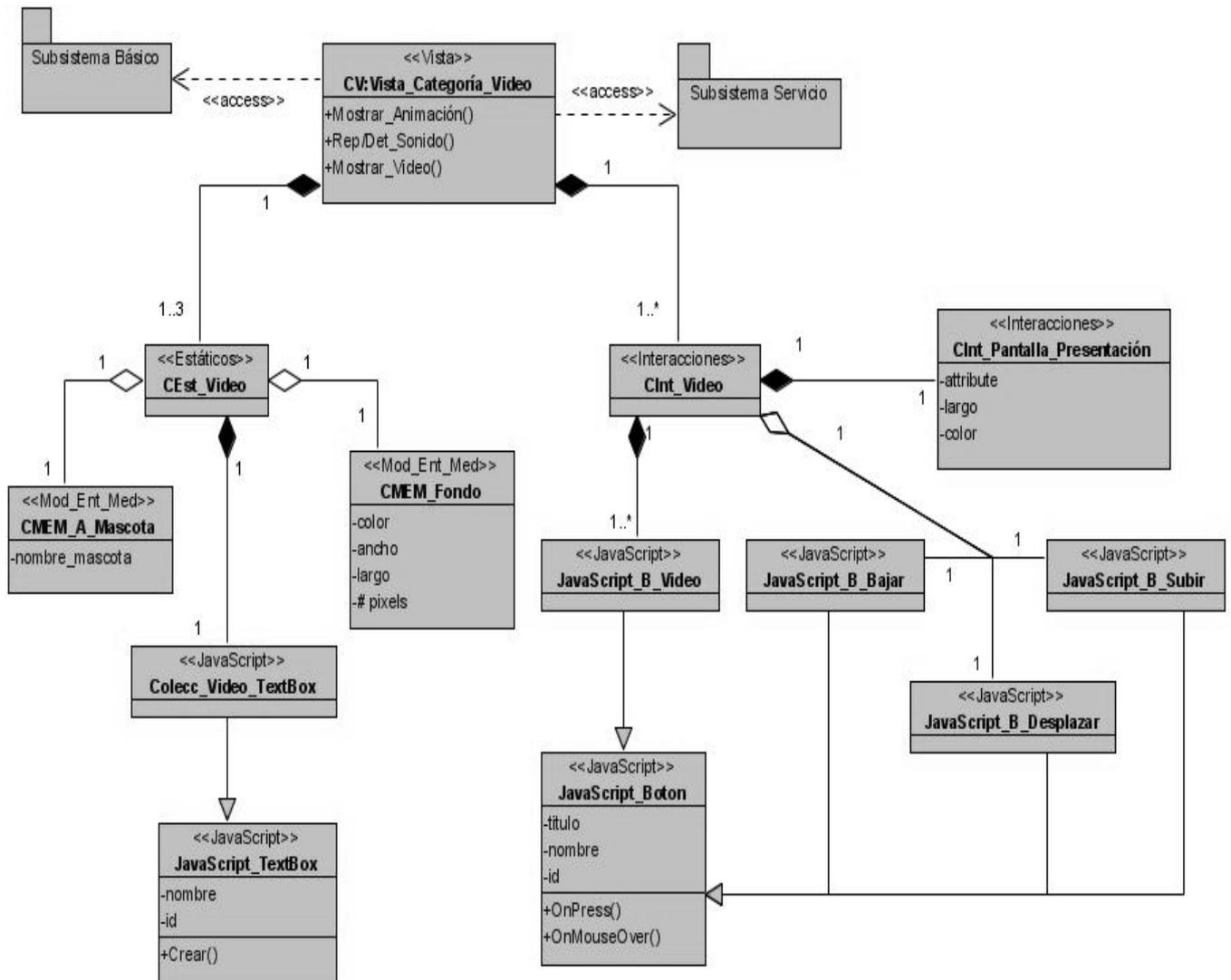


Ilustración 9: Diagrama de Estructura de Presentación de la Vista_Galería_Selección_Categoría_Videos.

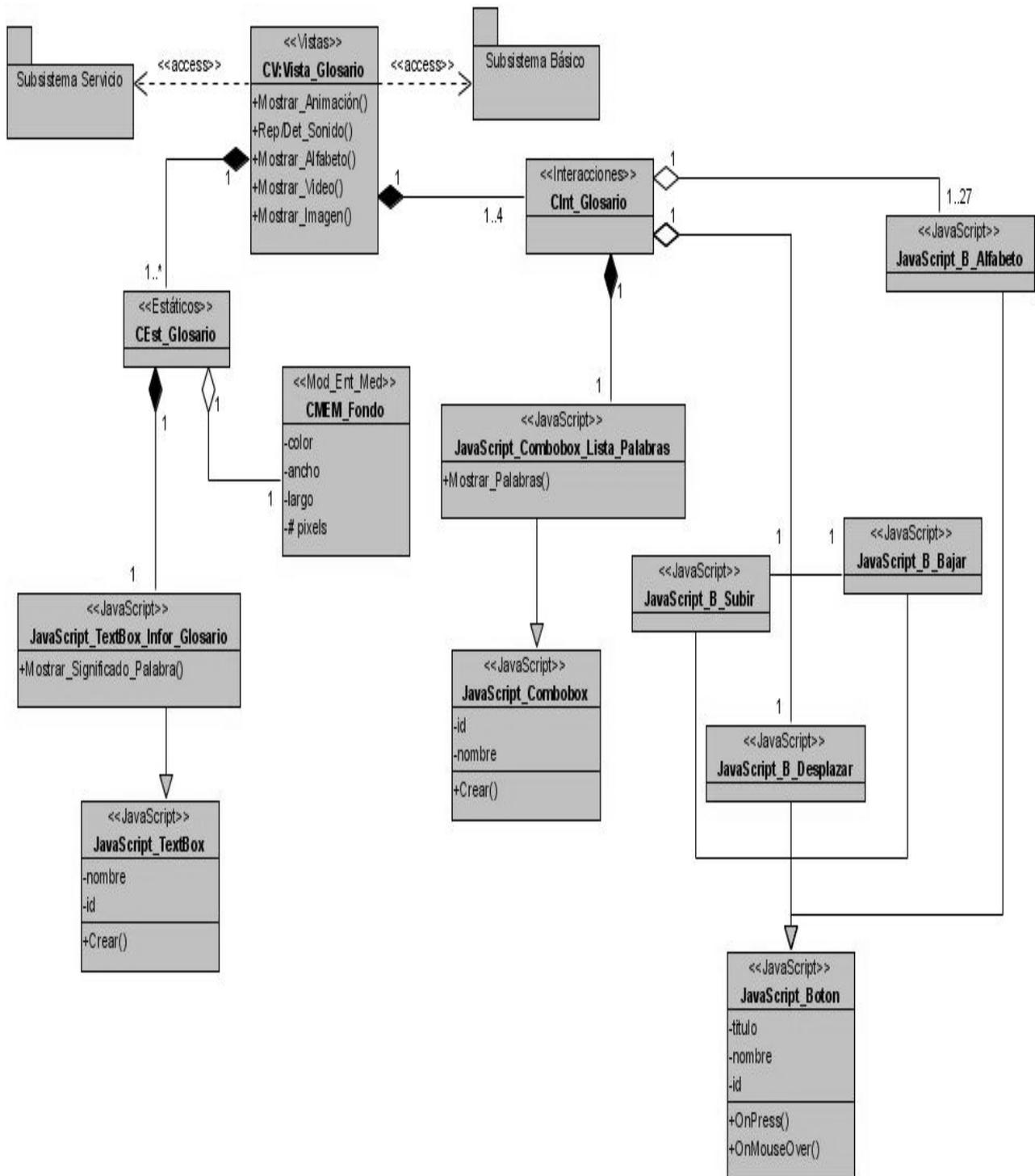


Ilustración 10: Diagrama de Estructura de Presentación de la Vista_Glosario_Términos.

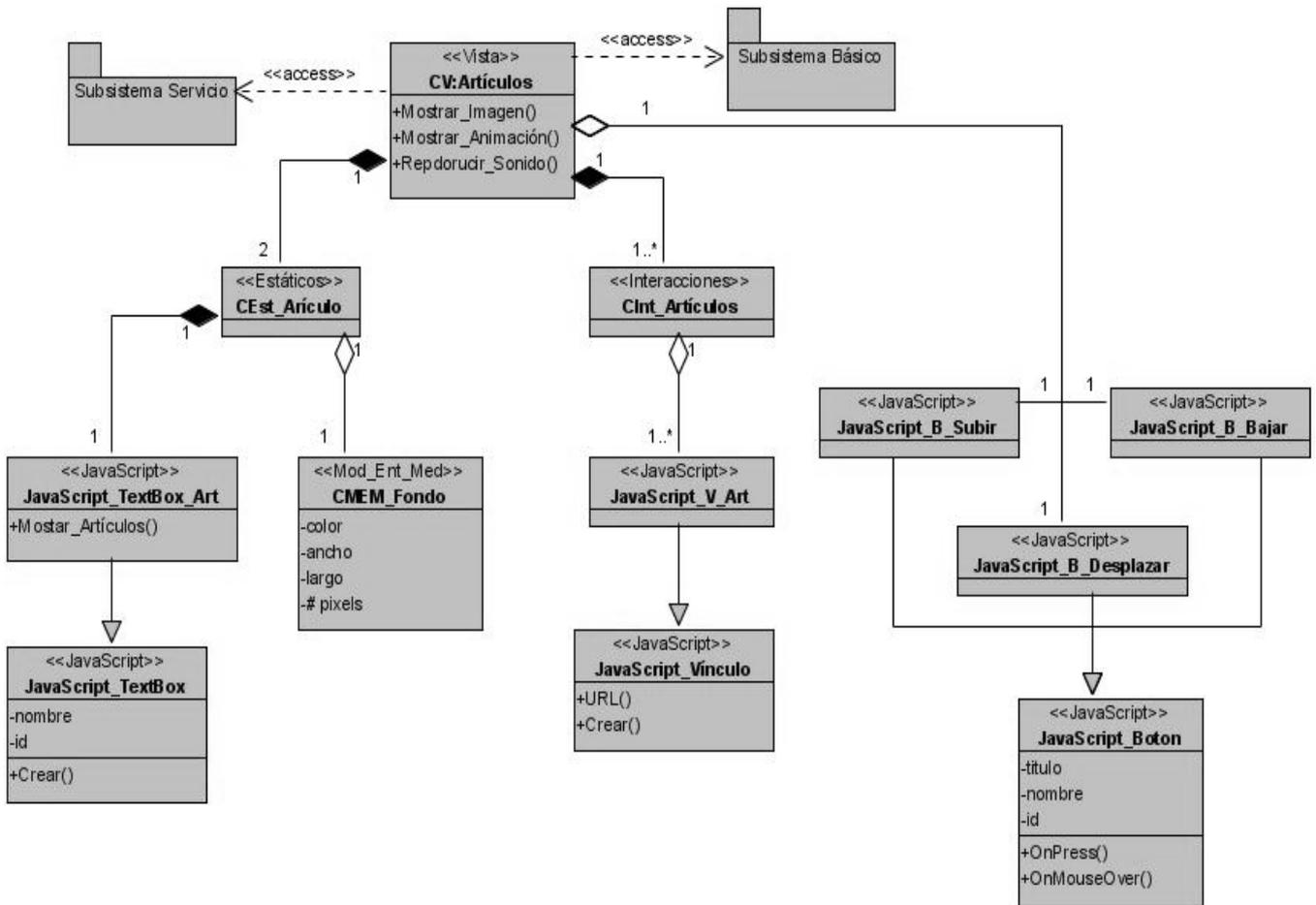


Ilustración 11: Diagrama de Estructura de Presentación de la Vista_Interés .

3.6. Diagrama de Estructura de Navegación.

Diagrama de Estructura de Navegación :extiende la semántica de los diagramas de clases para poder utilizarlos en este tipo de representación, incorporando los estereotipos restrictivos de clases: menú, índice, consulta y botón, además de utilizar las ya definidas: clase modelo entidad media texto y clase modelo entidad media imagen.[12]

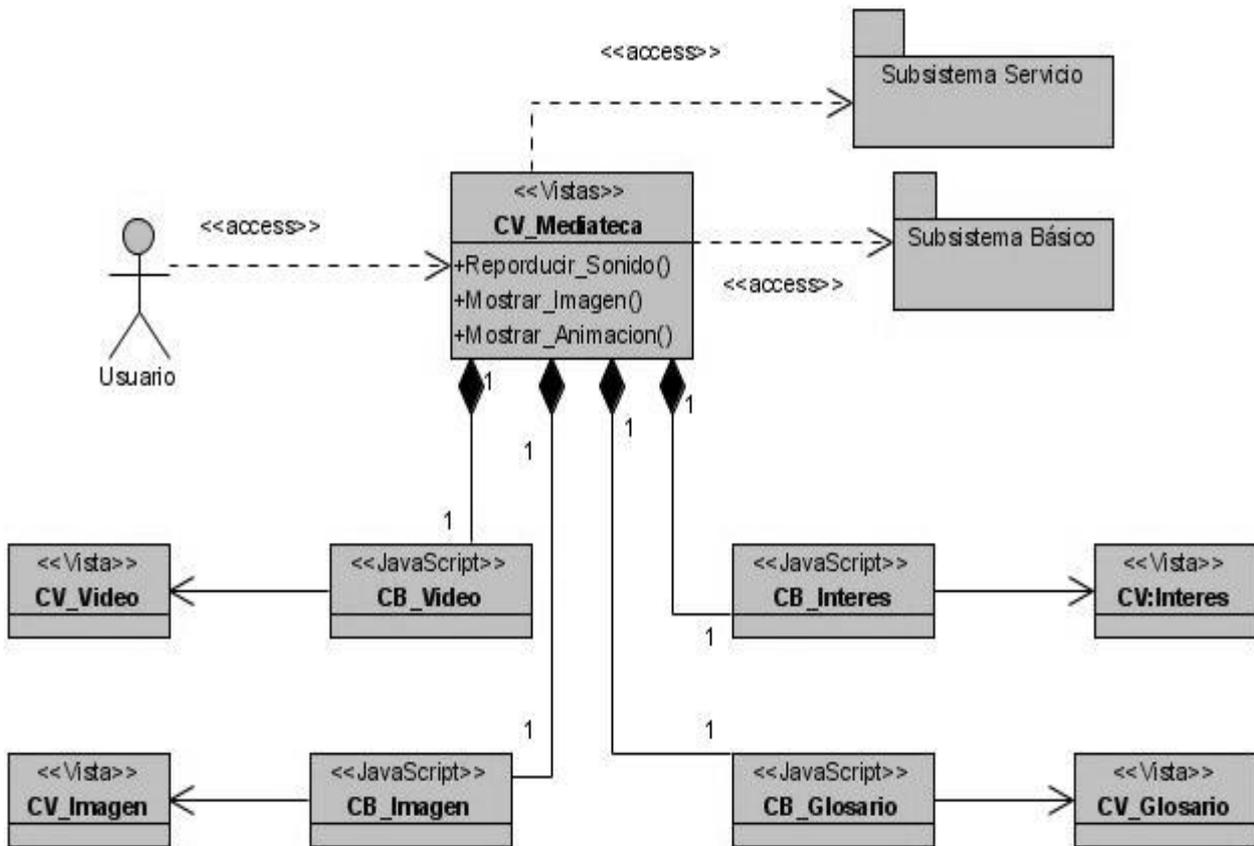


Ilustración 12: Diagrama de Estructura de Navegación del módulo Mediateca.

3.7. Diagrama de Clases.

Diagrama de Clases: Plantea la semántica y los estereotipos restrictivos y descriptivos para las clases asociadas a las tecnologías multimedia e hipermedia. Organiza la estructura del diagrama en secciones para la representación lógica de los distintos tipos de clases, incorporando las clases abstractas del modelo conceptual.[12]

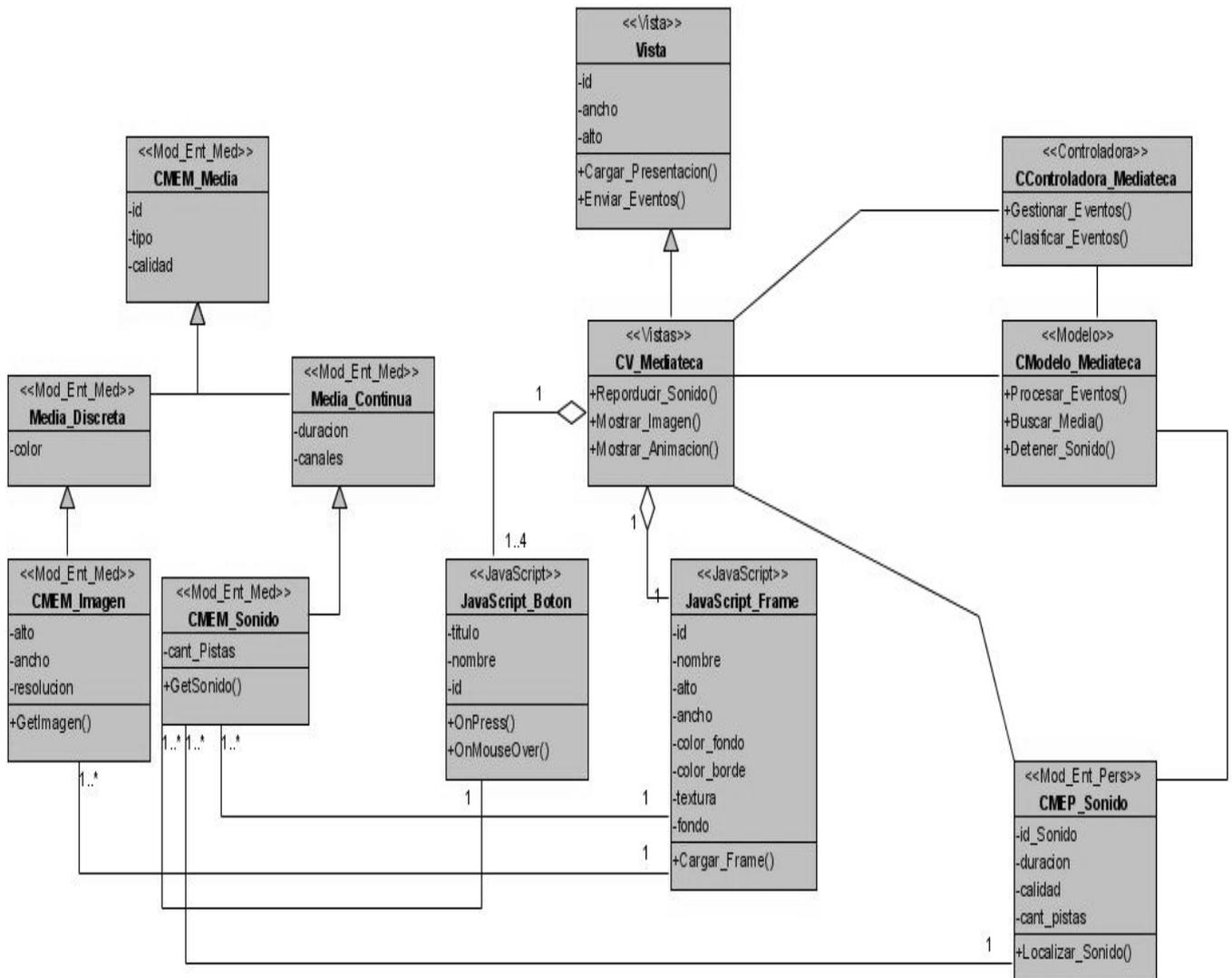


Ilustración 13: Diagrama de Clases del Diseño de la Vista_Presentación.

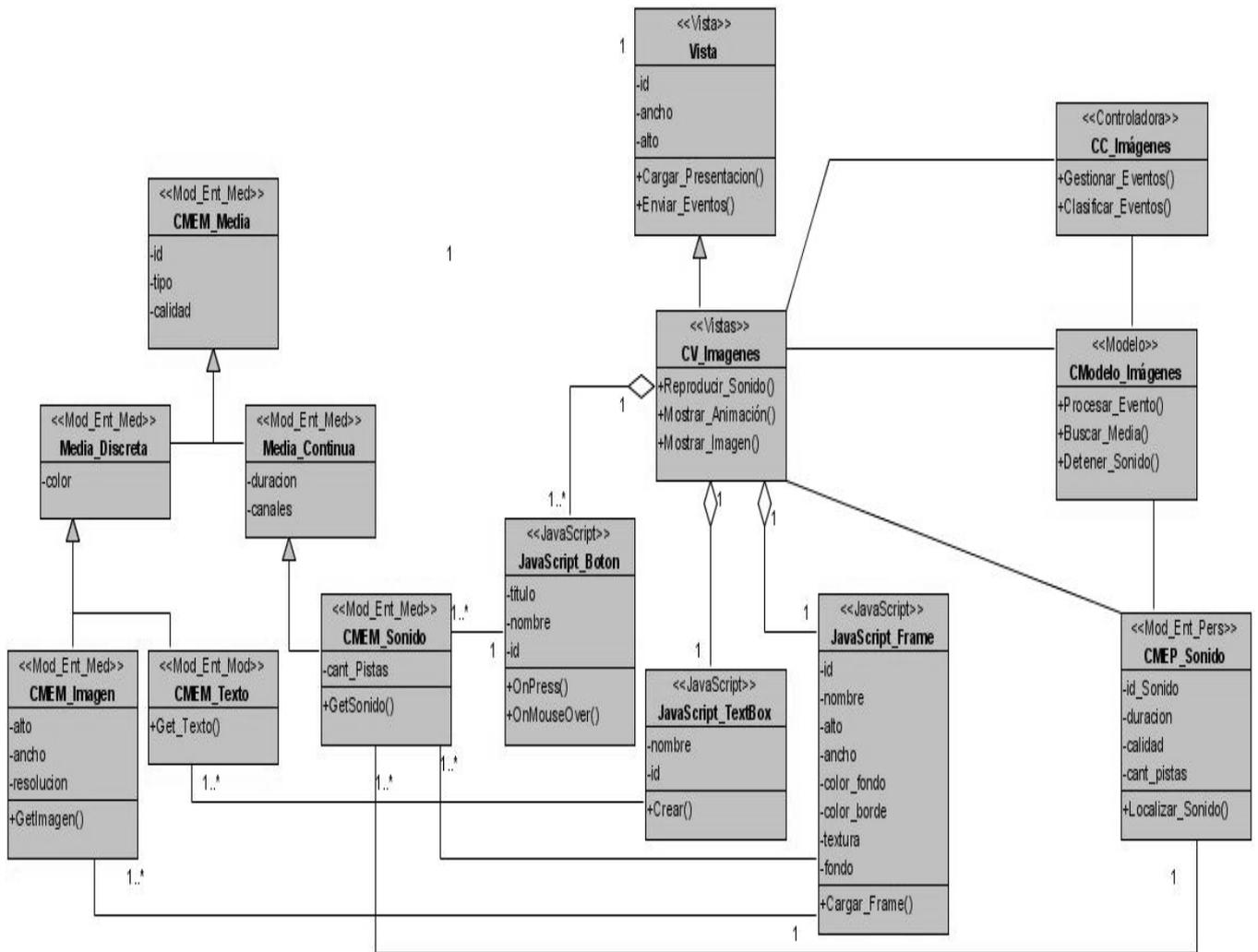


Ilustración 14: Diagrama de Clases del Diseño de la Vista_Categoría_Imágenes.

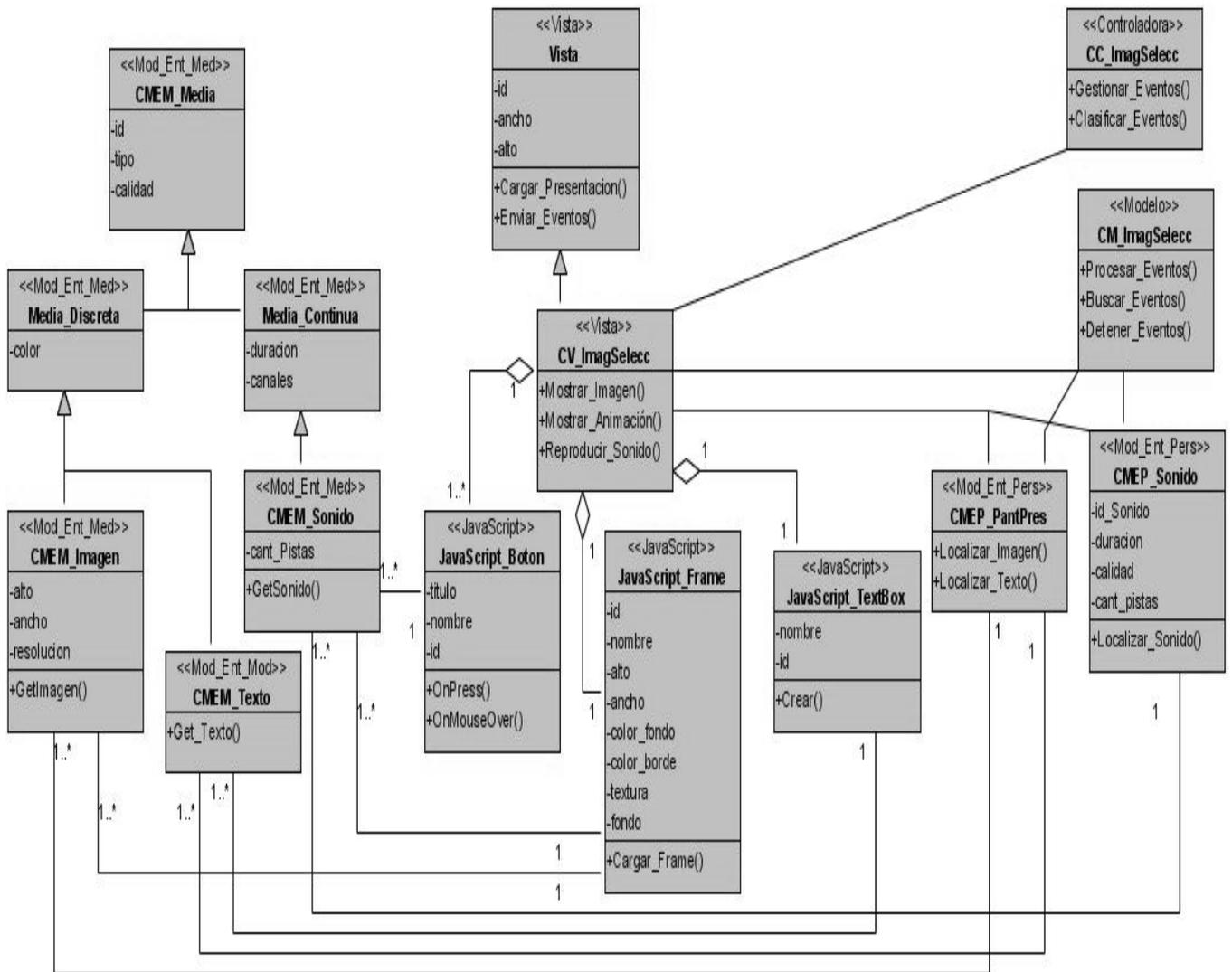


Ilustración 15: Diagrama de Clases del Diseño de la Vista_Categoría_Selección_Imágenes.

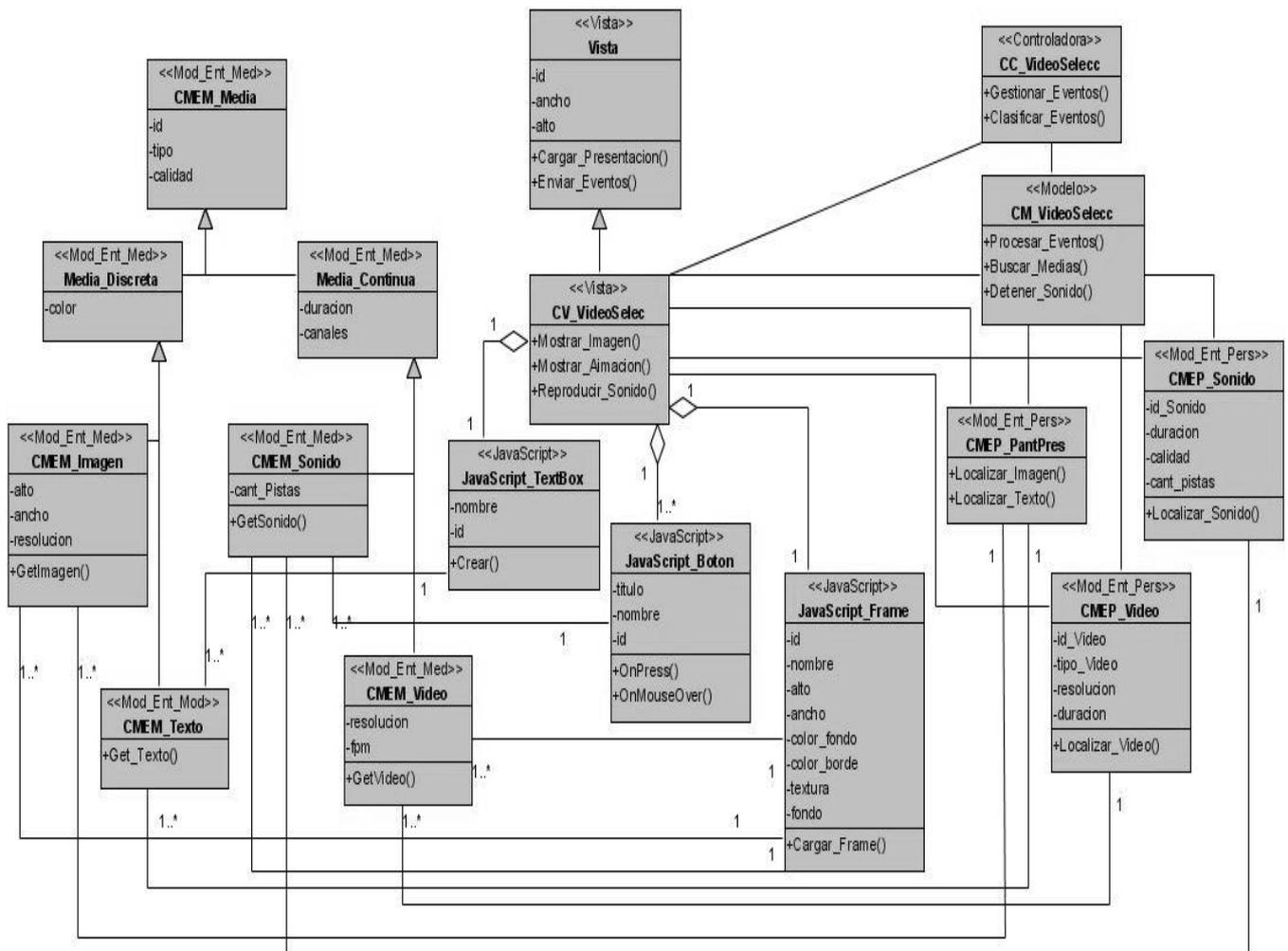


Ilustración 17: Diagrama de Clases del Diseño de la Vista_Categoría_Selección_Videos.

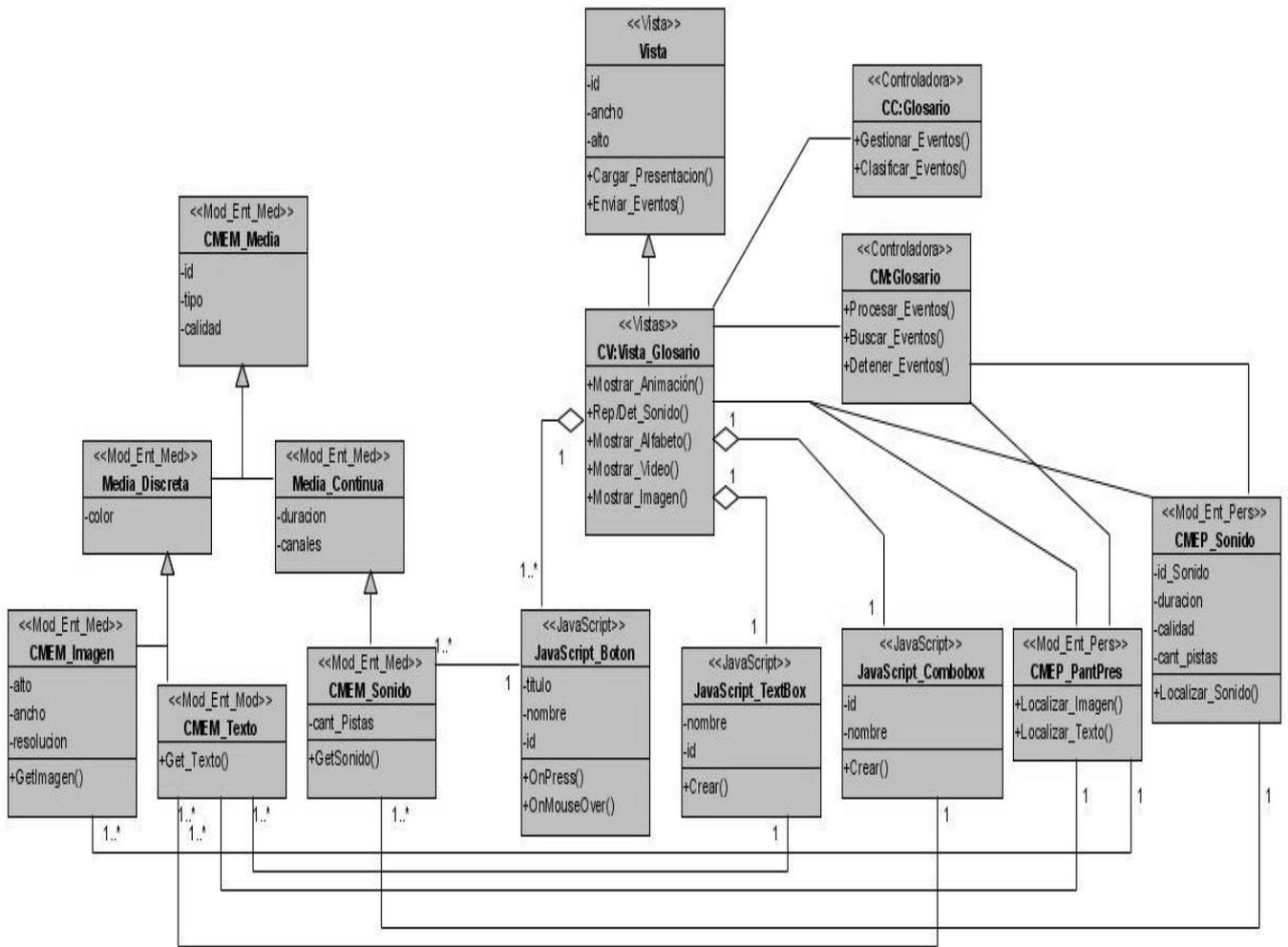


Ilustración 18: Diagrama de Clases del Diseño de la Vista_Glosario_Términos.

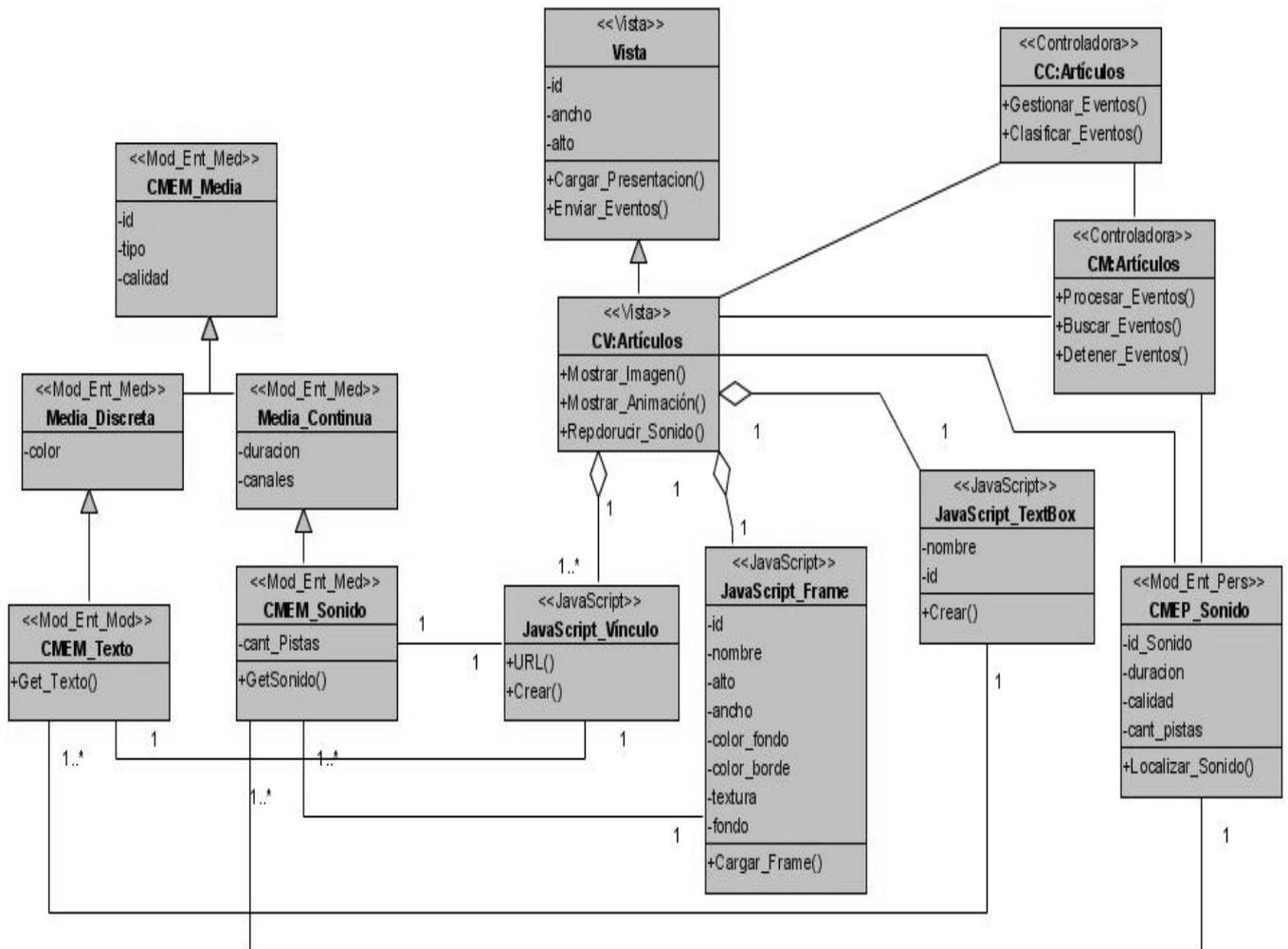


Ilustración 19: Diagrama de Clases del Diseño de la Vista_Interés.

3.8. Diagrama de Secuencia.

Diagrama de Secuencia: muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario. (11)

Ilustración 20: Diagrama de Secuencia Vista Presentación.

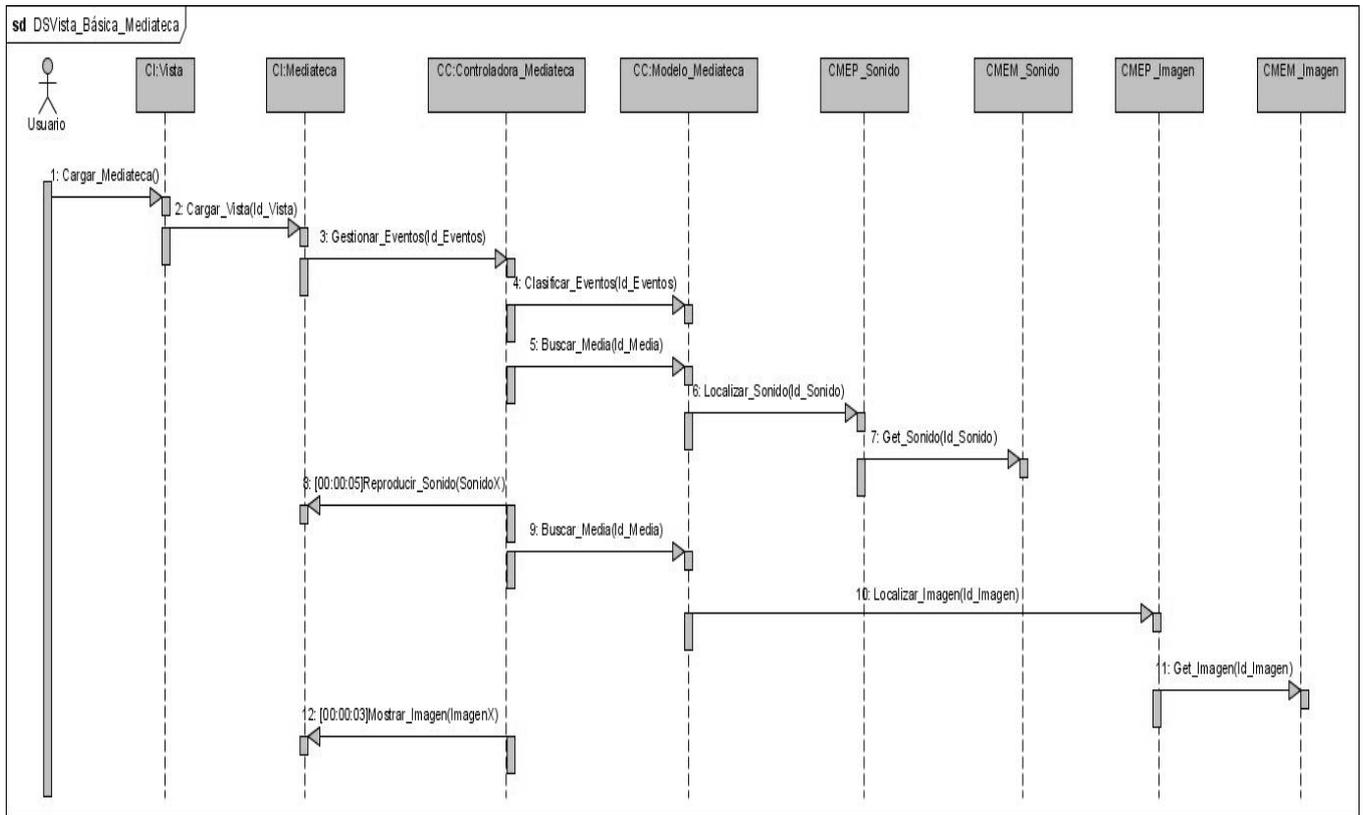


Ilustración 21: Diagrama de Secuencia Vista_Categoría_Imágenes.

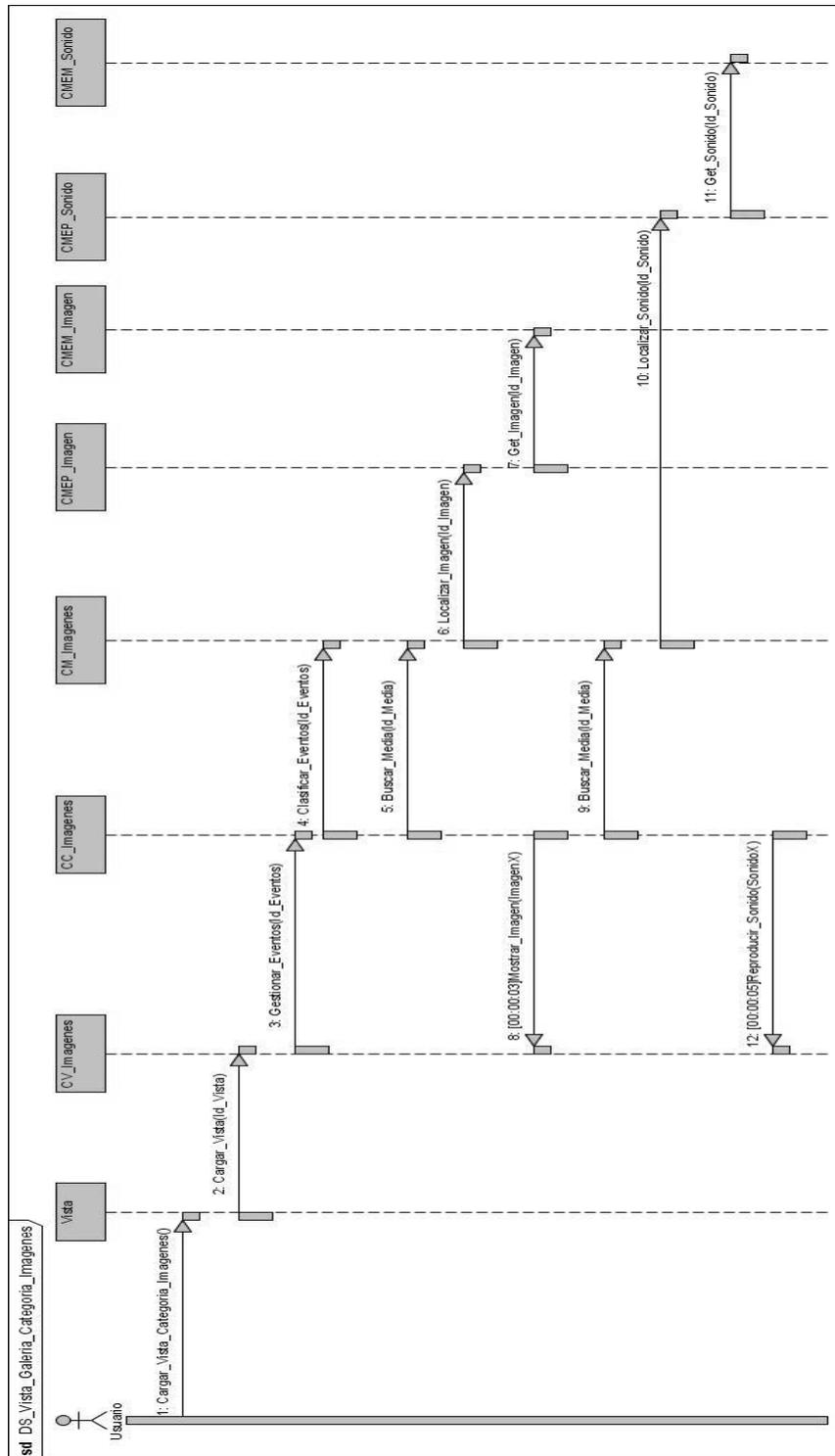
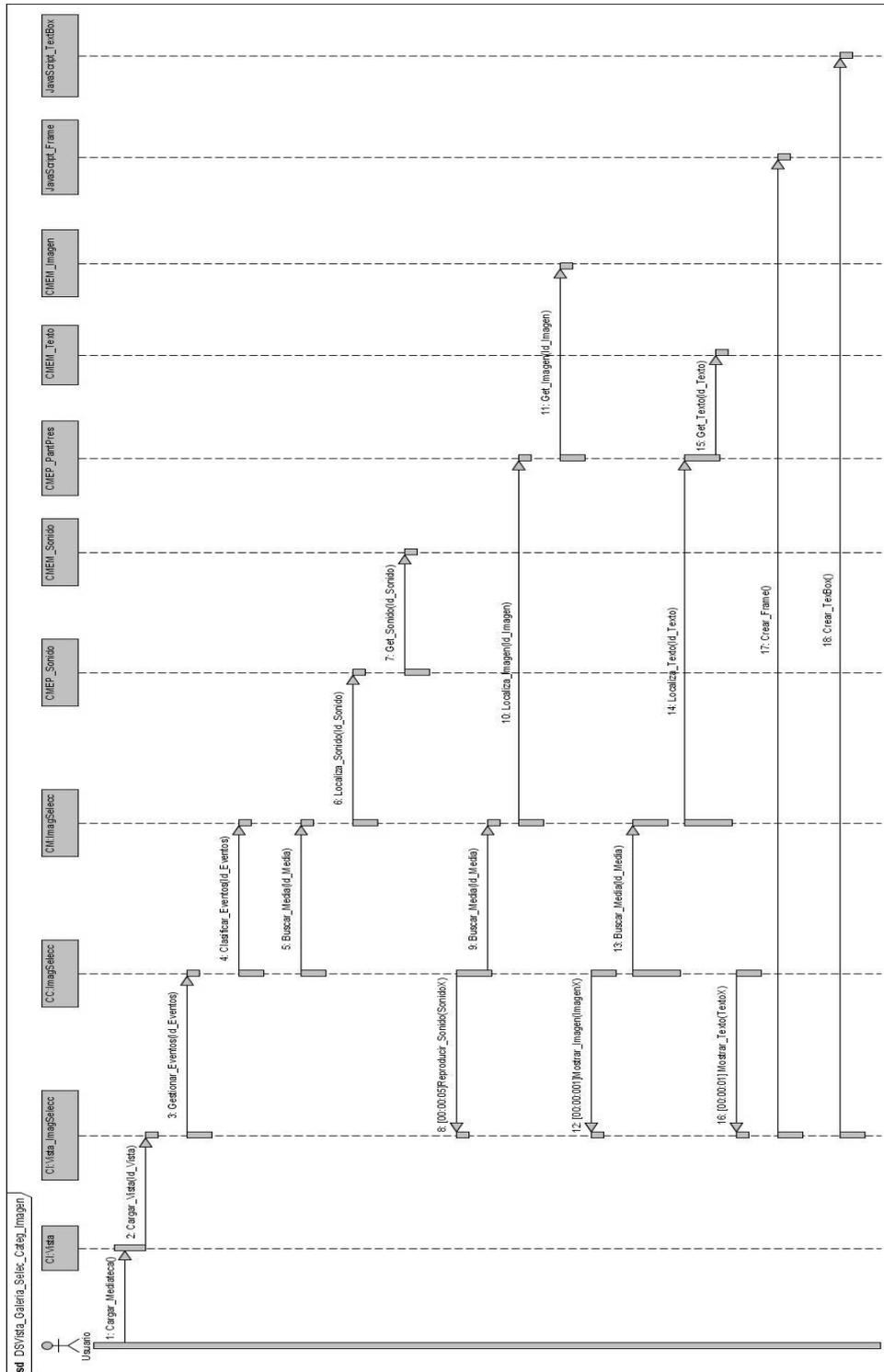


Ilustración 22: Diagrama de Secuencia Vista_Selección_Categoría_Imágenes.



3.9. Diagrama de Despliegue.

Diagrama de Despliegue: describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre nodos del cómputo. Representa en sí mismo una correspondencia entre la arquitectura software y la arquitectura del sistema (hardware).

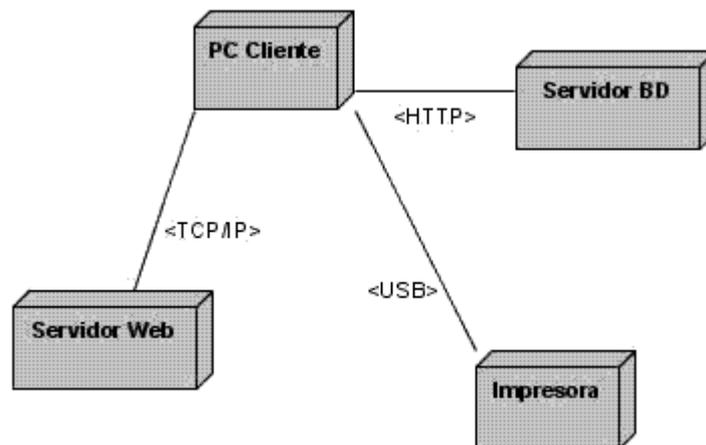


Ilustración 27: Diagrama de Despliegue Módulo Mediateca.

3.10. Conclusiones.

En este capítulo se hizo una descripción de la propuesta de solución exponiendo todo lo concerniente a las vistas del sistema. Se realizó además la modelación del dominio del entorno donde se presenta el problema que dio origen a la construcción de este módulo; la identificación de los objetos del dominio del sistema y sus relaciones y como paso fundamental para la modelación del sistema, los Requerimientos funcionales y los no funcionales, que son el resultado del análisis de las solicitudes del cliente y fueron modelados en términos de vistas del sistema. Con el desarrollo de este flujo de trabajo y los artefactos obtenidos a partir de este, se puede pasar al flujo de diseño para comenzar la construcción de la solución propuesta.

Capítulo 4

Estudio de factibilidad.

4.1. Introducción

Para el desarrollo de cualquier proyecto en la esfera productiva, incluyéndose las aplicaciones informáticas no sólo debe tenerse en cuenta la información y los medios a utilizar, y las potencialidades informáticas para su implementación; sino que al mismo tiempo es necesario conocer de antemano, el tiempo que se empleará en su desarrollo, los recursos humanos a utilizar, los insumos necesarios y los gastos económicos en los que se incurrirá. De igual manera se debe analizar si posterior a su ejecución existirá alguna forma de aminorar dichos gastos para una recuperación económica factible. Es por esta razón que se hace indispensable la realización del estudio de factibilidad al “Módulo Mediateca” de la Colección Multisaber.

4.2. Planificación

En el caso del producto en cuestión se utilizó COCOMO II (Modelo Constructivo de Costes) por su acrónimo del inglés Constructive Cost Model , es un modelo de estimación de costes de software que incluye tres submodelos, donde cada uno ofrece un nivel de detalle y aproximación cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado. Se escogió este método debido a que es uno de los modelos más documentados en la actualidad, es muy fácil de utilizar y está basado en las experiencias del COCOMO I. (10)

Tabla 11: Consultas Externas

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

Tabla 12: *Salidas Externas*

Nombre de la salida externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (Simple, Media y Compleja)
Mostrar el nombre del Software.	1	1	Simple
Mostrar el nombre de las secciones cuando alguna sea seleccionada.	1	1	Simple
Mostrar la pantalla de la sección seleccionada.	1	1	Simple
Mostrar ventana auxiliar comprobando si el usuario desea Salir.	1	1	Simple
Mostrar los créditos del producto al ser confirmada la salida.	1	1	Simple
Permitir al usuario controlar el audio del sistema.	1	1	Simple
Permitir en todo momento volver a la pantalla principal del producto.	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 una de sus secciones.	1	1	Simple

Tabla 13: Total de Consultas Externas y Salidas Externas

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	9	4	0	5	0	7	36
Subtotal (UFP)							40

Tabla 14: Puntos de Función

Características	Valor
Puntos de función desajustados	40
Lenguaje	JavaScript 2.0
Instrucciones fuentes por puntos de función	36
Instrucciones fuentes	1360

Resultados:

Puntos de función del proyecto: 40

Total de instrucciones fuentes: 1360

Miles de instrucciones fuentes (*MF*): 1.4 (aproximadamente)

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

Multiplicadores de esfuerzo	Valor
RCPX (RELY, DATA, CLPX, DOCU)	0.80
RUSE	1.00
PDIF (TIME, STOR, PVOL)	0.87
PERS (ACAP, PCAP, PCON)	0.83
PREX (APEX, PLEX, LTEX)	1.50
FCIL (TOOL, SITE)	1.00
SCED	0.70

Factores de Escala	Valor
PREC	3.50
FLEX	3.04
RESL	5.70
TEAM	2.19
PMAT	4.68

Multiplicadores de esfuerzo:

7

$$EM = \prod_{i=1}^7 Em_i = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED = \mathbf{0.60}$$

i=1

Factores de escala:

5

$$SF = \sum_{i=1}^5 SFi = PREC + FLEX + RESL + TEAM + PMAT = \mathbf{19.11}$$

i=1

Valores calibrados:

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

$$E = B + 0.01 * SF$$

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

$$E = 0.91 + 0.01 * 19.11$$

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

$$E = 1.1011 \approx 1.1$$

$$F = 0.278 \approx 0.28$$

Esfuerzo:

7

$$PM = A * (MF)^E * \prod_{i=1}^7 Em_i$$

i=1

$$PM = 2.94 * (1.4)^{1.1} * 0.60 = 2.554 \approx 2.55$$

$$PM \approx 3 \text{ Hombres mes.}$$

Cálculo del tiempo:

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

$$TDEV = 3.67 * (3)^{0.28} = 4.991$$

$$TDEV \approx 5 \text{ meses}$$

Cálculo de la cantidad de hombres:

$$CH = PM / TDEV$$

$$CH = 3 / 5 = 0.6$$

$$CH \approx 1 \text{ hombres.}$$

Al ser 2 la cantidad real de hombres ha desarrollar la aplicación al reajustar el tiempo de desarrollo según la cantidad de hombres, resultó un tiempo de **0.45 meses (45 días)**.

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: **\$270.00**.

Costo:

$$CHM = 10 * \text{Salario Promedio}$$

$$CHM = 10 * \$270$$

$$CHM = \$ 2700$$

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

$$\text{Costo} = \$2700 * 3$$

$$\text{Costo} = \$8100$$

4.4. Beneficios tangibles e intangibles

Como principales beneficios **tangibles** asociados al desarrollo del Módulo Mediateca de la colección Multisaber se señalan los siguientes:

- Se contará con una aplicación interactiva, dinámico y de interfaz agradable que posibilite a los estudiantes perfeccionar su aprendizaje.
- Las escuelas venezolanas contarán con un medio tecnológico que integrará la educación con las TICs.

Como principales beneficios **intangibles** asociados al desarrollo del Módulo Mediateca de la Colección Multisaber se señalan los siguientes:

- Mayor satisfacción al cliente.

4.5. Conclusiones.

Una vez terminado el estudio de factibilidad llegamos a la conclusión de que:

Cantidad de Hombres	1 hombres
Tiempo de desarrollo	5 meses
Costo	\$8100

Tabla 15: Resumen del estudio de factibilidad.

Este capítulo se ha dedicado la exposición a los aspectos que de una forma u otra influyen en la ejecución de una aplicación multimedia. Se puntualiza de una forma clara los costos a incurrir, los recursos humanos implicados, el tiempo de desarrollo y los beneficios que aporta la terminación del producto en cuestión.

Conclusiones

Al finalizar el presente trabajo de diploma se concluye que se han cumplido los objetivos propuestos.

- Se realizó un estudio de la descripción del módulo “Mediateca” de cada una de las 14 multimedias con las que contará la Colección Multisaber para Venezuela, demostrando que la similitud del módulo entre todas las multimedias permite que el presente documento sea aplicable a toda la colección de productos.
- Se realizó un estudio de las metodologías de desarrollo de software y de los lenguajes de modelado existentes para el desarrollo de multimedias seleccionando como metodología el Proceso Unificado de Desarrollo de Software (RUP) y ApEM-L como lenguaje de modelado.
- Se logró la elaboración de la documentación del módulo “Mediateca” de la Colección Multisaber para las escuelas primarias venezolanas aplicando Proceso Unificado de Desarrollo de Software (RUP) como metodología de desarrollo y apoyándose en el uso del lenguaje extendido de UML: ApEM-L para la modelación específica de aplicaciones multimedia educativas.
- Se realizó el estudio de factibilidad donde se obtuvieron buenos resultados de modo que se pudo comprobar que el desarrollo del módulo será factible y de bajo costo.

Recomendaciones

Después de haber concluido el trabajo y haber logrado dar cumplimiento a los objetivos propuestos al inicio del mismo se añaden a continuación algunas sugerencias o recomendaciones que deben tenerse en cuenta para proyectos futuros:

- El estudio de modificaciones al modelado de software de este tipo que incorporen características y artefactos más específicos para la modelación de aplicaciones Web.
- Coordinar con el MINED para lograr la orientación a la confección de guiones con la utilización de ApEM-L.
- Que se mantenga copia del documento por parte del MINED y la UCI para que sirva de consulta y documentación de la colección de multimedias a las que tributa.

Referencias Bibliográficas

- (1) Multisaber, Disponible en : <http://www.insted.rimed.cu/>
- (2) SANCHEZ .J."Construyendo y Aprendiendo con el computador" 1999.
- (3) SCRUM, Disponible en : www.controlchaos.com
- (4) Crystal, Disponible en : www.crystallmethodologies.org.
- (5) DSDM, Disponible en : www.dsdm.org.
- (6) ASD, Disponible en : www.adaptivesd.com.
- (7) FDD, Disponible en : www.featuredrivendevelopment.com.
- (8) LD, Disponible en : www.poppendieck.com.
- (9) Rational Unified Process, Wikipedia, la enciclopedia libre, Disponible en http://es.wikipedia.org/wiki/Proceso_Unificado
- (10) TERUEL. A. Cocomo II: Una familia de modelos de estimación, Disponible en : <http://es.wikipedia.org/wiki/COCOMO>
- (11) Disponible en : <http://www-gris.det.uvigo.es/~avilas/UML/node42.html>
- (12) GUERRERO, L.A., Análisis y Diseño Orientado a Objetos. Modelo conceptual. Disponible en [:http://www.dcc.uchile.cl/~luquerre/cc40b/clase4.html](http://www.dcc.uchile.cl/~luquerre/cc40b/clase4.html).
- (13) JACOBSON I, B. G., RUMBAUGH J. *El Proceso Unificado de Desarrollo de Software*.
- (14) *Herrera.L,J., Ingeniería De Requerimientos. Ingeniería De Software.*

Bibliografía

1. Pressman., "Software Engineering Resources." 2006.
2. Software Disponible en : http://es.wikipedia.org/wiki/Computer_software
3. Educativo, Software. Disponible en : http://es.wikipedia.org/wiki/Software_educativo.
4. Educativo, Software. Disponible en : <http://www.monografias.com/trabajos31/software-educativo-cuba/software-educativo-cuba.shtml>
5. Multimedia, Disponible en : <http://es.wikipedia.org/wiki/Multimedia>
6. Libre, Software., Disponible en : <http://www.gnu.org/home.es.html>
7. Libre, Software., Disponible en : http://es.wikipedia.org/wiki/C%C3%B3digo_libre
8. Jacobson, I., "Applying UML in The Unified Process" Presentación. Rational Software. . 1998.
9. Marqués, "Metodología para la elaboración de software educativo." 1995.
10. UML, Disponible en : <http://es.wikipedia.org/wiki/UML>.
11. Hernández, Y.M.L., Trabajo de Diploma para obtener el título de Ingeniero en Ciencias Informáticas : APLICACIÓN DE LA NOTACIÓN OMMMA-L PARA SOFTWARE EDUCATIVO. 2007, Universidad de las Ciencias Informáticas: Ciudad Habana.
12. Ciudad Ricardo, Ing. Febe .Ángel., ApEM-L como una nueva solución a la modelación de aplicaciones educativas multimedias en la UCI.
13. PÉREZ, Y.M., PLANTILLA PARA EL MONTAJE DINÁMICO DE LOS PRODUCTOS DE LA COLECCIÓN MULTISABER. 2006.
14. Visual, [http://www.freownloadmanager.org/es/downloads/SDE_para_Estudio_Visual_\(Iglesia_Anglicana\)_para_Windows_3965_p/](http://www.freownloadmanager.org/es/downloads/SDE_para_Estudio_Visual_(Iglesia_Anglicana)_para_Windows_3965_p/).

Glosario

Actor: Abstracción de las entidades externas a un sistema, subsistemas o clases que interactúan directamente con el sistema. Un actor participa en un caso de uso o en conjunto coherente de casos de uso para llevar a cabo un propósito global.

Copyright: El Derecho de autor (del francés *droit d'auteur*) es un conjunto de normas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores (los *derechos de autor*), por el solo hecho de la creación de una obra literaria, artística o científica, tanto publicada o que todavía no se haya publicado.

Modelo: Es una abstracción semánticamente completa de un sistema.

Módulo: Término que denota una unidad para el almacenamiento y manipulación del software. La palabra no corresponde a una única estructura de UML, sino que incluye varias estructuras.

Pantalla: grupo de elementos de medias visuales que están comprendidos en una vista determinada.

Requerimiento: Una característica, propiedad o comportamiento que se desea para el sistema.

Sistema: Colección de unidades conectadas que se organiza para lograr un propósito. El sistema es el “modelo completo”.

Vista: Interfaz de presentación que se le presenta al usuario cuando interactúa con el sistema