

**Universidad de las Ciencias Informáticas
Facultad 8**



**Título: “Análisis y Diseño del Módulo Resultados
de la Colección Multisaber”**



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

Autora: Dianick Velázquez Corona

Tutora: Ing. Lidisvey Herrero González

Ciudad de la Habana, Julio 2008

“Año del 50 de la Revolución”

Declaración de autoría

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

Para que así conste firmo la presente a los 8 días del mes de Julio del año 2008.

Dianick Velázquez Corona

Ing. Lidisvey Herrero González

Firma del Autor

Firma del Tutor

*“El árbol hace brotar de su tronco en los primeros años,
aquellas ramas principales que ha de tener, y así no tiene
después sino ir las desarrollando”*

Juan Ramos Comenio

Agradecimientos

A mi mamita querida y a mi hermanito del alma por quererme tanto, por estar siempre dándome todo su apoyo y todo su amor. Por darme mucha fuerza e impulsarme hacia el camino correcto y hacer que hoy me sienta satisfecha por el esfuerzo que he realizado.

A mi papito del alma por ser un ejemplo para mí, por ser la luz y guía en mi camino y brindarme todo su apoyo, amor y dedicación.

A mi novio Carlos Daniel, por amarme y malcriarme tanto, por ser muy especial en mi vida y estar a mi lado todos estos años, así como a su familia, en especial a Martica y a Memo por manifestarme su apoyo y dedicación.

A Yane, Isle, Arquí, Livan y David y mis compañeros de aula por ayudarme en los momentos difíciles, brindarme su amistad y compartir estos años a mi lado.

A mi tutora por su ayuda incondicional y significativa...

A los profes del Mined, en especial a Ponce por ayudarme en la redacción, transmitirme sus experiencias y alegrarme las mañanas.

Dedicatoria

*A mis padres queridos, mi hermanito del alma y a mi
amor Carlos Daniel, que juntos conforman el tesoro
más valioso de mi vida...*

Resumen

A lo largo de la historia se han producido avances científicos e innovaciones tecnológicas que han marcado profundamente la evolución de la humanidad, pues con la llegada de las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC) vinculadas con el desarrollo del software educativo como medio de enseñanza – aprendizaje, han modificado los paradigmas de la visión que antes tenían los seres humanos. Una de las primeras características que se deben de tener en cuenta en la realización de un software educativo es la conceptualización y el entorno del mismo, pues la concepción de las metodologías de desarrollo unida a los lenguajes de modelación hacen que éste se pueda representar, modelar y construir teniendo en cuenta los pilares pedagógicos para su representación y desarrollo. Por tal razón el presente trabajo de diploma aborda los principales aspectos que estuvieron presentes en los flujos de trabajo de Análisis y Diseño de la Colección Multisaber, específicamente del Módulo Resultados y la necesidad de migrar dicha colección a software libre. Además se utiliza el lenguaje de modelado ApEM – L para modelar la representación y conceptualización del entorno del producto y finalmente se realiza el estudio de la factibilidad para estimar el tiempo, los costos y beneficios del software a desarrollar.

Palabras Clave:

Análisis y Diseño, Software Educativo, Software Libre, ApEM – L.

Tabla de Contenidos

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Introducción	6
1.2 Las TIC en el proceso de enseñanza y aprendizaje.....	6
1.3 Tipologías de software.	7
1.3.1 Software libre vs software propietario.	8
1.4 La migración a software libre en Venezuela.	10
1.5 Ventajas y retos del software libre en Venezuela.	12
1.6 Descripción del Objeto de Estudio.	13
1.6.1 Descripción general.....	13
1.6.2 Identificación de la audiencia.....	14
1.7 Plataforma Metodológica.....	14
1.7.1 Objetivos Pedagógicos.....	14
1.8 Análisis del modelo de arquitectura de información utilizada.....	14
1.8.1 Estándares de la interfaz de la aplicación.....	14
1.9 Conclusiones	15
CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES	16
2.1 Introducción	16
2.2 Tendencias actuales para desarrollar la propuesta.....	16
2.2.1 Informatización del aprendizaje.	16
2.2.2 Informática Educativa	17
2.2.3 Repercusión del Software Educativo en la educación.	17
2.3 Tecnologías actuales para desarrollar la propuesta.	18
2.3.1 Tecnología Multimedia.	18
2.3.2 Tecnología Hipertexto.	19
2.3.3 Tecnología Hipermedia.	20

2.4 Aspectos de la Ingeniería de Software	20
2.4.1 Análisis de las metodologías existentes.	21
2.4.1.1 Metodología de Administración de Relaciones (RMM).....	21
2.4.1.2 Metodología de Diseño Hipermedia Orientada a Objetos (OOHDM).....	23
2.4.1.3 Programación Extrema (XP)	24
2.4.1.4 Proceso Unificado del Rational (RUP)	25
2.4.1.5 Selección y fundamentación de la metodología de desarrollo escogida.....	26
2.4.2 Análisis de los lenguajes de modelado existentes.	27
2.4.2.1 Lenguaje Unificado de Modelado (UML)	27
2.4.2.2 Lenguaje Orientado a Objetos para la Modelación de Aplicaciones Multimedia (OMMMA - L).....	28
2.4.2.3 Lenguaje para la modelación de Aplicaciones Educativas Multimedia (ApEM – L)	29
2.4.2.4 Selección y fundamentación del lenguaje de modelado escogido.....	30
2.5 Herramientas para la modelación	31
2.5.1 Rational Rose.....	31
2.5.2 Visual Paradigm	32
2.5.3 Fundamentación de la herramienta escogida.	32
2.6 Conclusiones	33
CAPÍTULO 3: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	34
3.1 Introducción	34
3.2 Especificación del contenido.	34
3.3 Modelo de Dominio.	35
3.3.1 Diagrama de clases del modelo de dominio.	36
3.3.2 Análisis de los conceptos de dominio.	36
3.4 Descripción del sistema propuesto.	37
3.4.1 Descripción de la funcionalidad.	37
3.4.1.1 Requisitos funcionales del sistema.	37
3.4.1.2 Requerimientos no funcionales.....	38
3.5 Actores del sistema.	40
3.5.1 Justificación del actor del sistema.	40
3.6 Área de Gestión de Modelo del sistema.	40

Tabla de Contenidos

3.6.1 Subsistemas y sus relaciones.	40
3.6.2 Diagrama de Estructura de Navegación.	42
3.6.3 Diagrama de Estructura de Presentación.	44
3.7 Conclusiones	69
CAPÍTULO 4: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA	70
4.1 Introducción	70
4.2 Área de Estructura Lógica	70
4.2.1 Vista Estática.....	70
4.2.1.1 Diagrama de Clases.	70
4.2.2 Vista de Arquitectura.	84
4.2.2.1 Diagrama de Despliegue.	84
4.3 Área de Comportamiento Dinámico	85
4.3.1 Vista de Comportamiento.	85
4.3.1.1 Diagrama de Secuencia.....	85
4.4 Conclusiones	97
CAPÍTULO 5: ESTUDIO DE LA FACTIBILIDAD	98
5.1 Introducción	98
5.2 Modelo Constructivo de Costes	98
5.3 Planificación mediante puntos de función	98
5.4 Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo del proyecto	101
5.5 Beneficios Tangibles e intangibles	104
5.5.1 Beneficios Tangibles	104
5.5.2 Beneficios Intangibles.....	104
5.6 Conclusiones	105
CONCLUSIONES GENERALES	106
RECOMENDACIONES	107

Tabla de Contenidos

REFERENCIAS BIBLIOGRÁFICAS	108
BIBLIOGRAFÍA	110
GLOSARIO DE TÉRMINOS Y SIGLAS	111

Índice de Ilustraciones

Ilustración 1 Diagrama de Modelo de Dominio	36
Ilustración 2 Actores del Sistema	40
Ilustración 3 Subsistemas y sus relaciones	42
Ilustración 4 Diagrama de Estructura de Navegación del sistema.....	44
Ilustración 5 Diagrama de estructura de presentación de vista Listado _ Estudiante	45
Ilustración 6 Diagrama de estructura de presentación de vista Filtros	47
Ilustración 7 Diagrama de estructura de presentación de Vista Identificación _ Estudiante	49
Ilustración 8 Diagrama de estructura de presentación de vista Datos _ Generales.....	51
Ilustración 9 Diagrama de estructura de presentación de vista Itinerario.....	53
Ilustración 10 Diagrama de estructura de presentación de vista Temas	55
Ilustración 11 Diagrama de estructura de presentación de vista Juegos	57
Ilustración 12 Diagrama de estructura de presentación de vista Ejercicios	59
Ilustración 13 Diagrama de estructura de presentación de vista Galería _ Videos	60
Ilustración 14 Diagrama de estructura de presentación de vista Galería _ Imágenes	62
Ilustración 15 Diagrama de estructura de presentación de vista Glosario	64
Ilustración 16 Diagrama de estructura de presentación de vista Información _ Interés	66
Ilustración 17 Diagrama de estructura de presentación de vista Sección.....	68
Ilustración 18 Diagrama de Clases de vista de presentación Listado _ Estudiante	72
Ilustración 19 Diagrama de Clases de vista de presentación Filtros	73
Ilustración 20 Diagrama de Clases de vista de presentación Identificación _ Estudiante	74
Ilustración 21 Diagrama de Clases de vista de presentación Datos _ Generales.....	75
Ilustración 22 Diagrama de Clases de vista de presentación Temas	76
Ilustración 23 Diagrama de Clases de vista de presentación Juegos	77
Ilustración 24 Diagrama de Clases de vista de presentación Ejercicios	78
Ilustración 25 Diagrama de Clases de vista de presentación Galería _ Imágenes	79
Ilustración 26 Diagrama de Clases de vista de presentación Galería _ Videos	80
Ilustración 27 Diagrama de Clases de vista de presentación Glosario	81
Ilustración 28 Diagrama de Clases de vista de presentación Información _ Interés.....	82

Índice de Ilustraciones

Ilustración 29 Diagrama de Clases de vista de presentación Itinerario	83
Ilustración 30 Diagrama de Clases de vista de presentación Sesión	84
Ilustración 31 Diagrama de Despliegue del sistema propuesto	85
Ilustración 32 Diagrama de Secuencia de Vista de Presentación Listado _ Estudiante.....	86
Ilustración 33 Diagrama de Secuencia de Vista de Presentación Filtros	87
Ilustración 34 Diagrama de Secuencia de Vista de Presentación Identificación _ Estudiante..	88
Ilustración 35 Diagrama de Secuencia de Vista de Presentación Datos _ Generales	89
Ilustración 36 Diagrama de Secuencia de Vista de Presentación Temas.....	90
Ilustración 37 Diagrama de Secuencia de Vista de Presentación Ejercicios	91
Ilustración 38 Diagrama de Secuencia de Vista de Presentación Juegos	92
Ilustración 39 Diagrama de Secuencia de Vista de Presentación Galería _ Imágenes.....	93
Ilustración 40 Diagrama de Secuencia de Vista de Presentación Galería _ Videos	94
Ilustración 41 Diagrama de Secuencia de Vista de Presentación Información _ Interés	95
Ilustración 42 Diagrama de Secuencia de Vista de Presentación Glosario	96
Ilustración 43 Diagrama de Secuencia de Vista de Presentación Sesión.....	97

Índice de Tablas

Tabla 1 Descripción Textual de Vista de Presentación Listado _ Estudiante	46
Tabla 2 Descripción Textual de Vista de Presentación Filtros	47
Tabla 3 Descripción Textual de Vista de Presentación Identificación _ Estudiante	49
Tabla 4 Descripción Textual de Vista de Presentación Datos _ Generales.....	51
Tabla 5 Descripción Textual de Vista de Presentación Itinerario.....	53
Tabla 6 Descripción Textual de Vista de Presentación Temas	55
Tabla 7 Descripción Textual de Vista de Presentación Juegos	57
Tabla 8 Descripción Textual de Vista de Presentación Ejercicios	59
Tabla 9 Descripción Textual de Vista de Presentación Galería _ Videos	61
Tabla 10 Descripción Textual de Vista de Presentación Galería _ Imágenes.....	62
Tabla 11 Descripción Textual de Vista de Presentación Glosario	64
Tabla 12 Descripción Textual de Vista de Presentación Información _ Interés.....	66
Tabla 13 Descripción Textual de la Vista de Presentación Sesión.....	68
Tabla 14: Entradas externas	98
Tabla 15: Salidas Externas	99
Tabla 16: Puntos de función	100
Tabla 17: Cantidad de instrucciones fuentes	101
Tabla 18: Valores de multiplicadores de esfuerzo	101
Tabla 19: Valores de factores de escala.....	102
Tabla 20: Resultados Estimados	105

Introducción

Antes de la tercera Revolución Educacional la dirección del Ministerio de Educación (MINED) decidió crear grupos de docentes con motivaciones informáticas para elaborar Software Educativo. Generalmente estos docentes eran especialistas de la facultad Matemática - Computación, Física, Inglés y Lenguas Españolas de los Institutos Superiores Pedagógicos (ISP). Lo que caracterizó este período fue la elaboración de software o programas educativos dedicados a problemas específicos de enseñanza – aprendizaje; como la solución de programas ortográficos, cálculo numérico y cinemática cuya elaboración no siempre estaba sustentada en bases teóricas bien fundamentadas, ni bajo preceptos que tuvieran que ver con la Ingeniería de Software, pues este era un proceso totalmente espontáneo. De esta manera se crearon 16 centros de desarrollo de Software Educativo 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 disolución de la Unión de Repúblicas Socialistas Soviéticas (URSS) entre otros factores, nuestro país se da a la tarea de realizar profundas transformaciones en la educación, las cuales estuvieron inspiradas en el marco de la conocida “Batalla de Ideas”. Esta revolución tiene como base la búsqueda de un justo equilibrio entre los componentes instructivos y educativos del proceso de enseñanza - aprendizaje en la escuela cubana, considerándose como fin primordial la elevación del componente educativo relacionada con la formación de valores, esencialmente con 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 mejor atención diferenciada a las diferentes individualidades de los educandos y de esta forma reforzar el trabajo educativo. Este aspecto tiene una repercusión económica, pues al reducirse a 15 el número de alumnos en las aulas implicaba mayor cantidad de profesores y por ende muchos más salarios, lo que obliga a elevar el diapasón de materias que el profesor debía impartir; esta transformación conllevó a una mayor integralidad por parte del docente.

Resulta obvio que la integridad, al menos en sus inicios, está en contraposición con el dominio de la materia, tal que para compensar este equilibrio natural aparece la tecnología acompañada de modernas teorías de aprendizaje para dar solución a la contradicción planteada; de esta manera surge

Introducción

la idea de la creación de una colección de software educativos para cada uno de los sistemas educacionales. Estos software son:

- Multisaber para la Enseñanza Primaria.
- El Navegante para la Enseñanza Secundaria.
- Futuro para la Enseñanza Preuniversitaria.

La colección Multisaber tiene un enfoque curricular y multidisciplinario, por su relación con los contenidos de los programas de cada asignatura del plan de estudio de la Educación Primaria. Cuenta con un grupo de software que contribuye a un aprendizaje desarrollador y a la formación de valores tales como: la dignidad, la responsabilidad, la laboriosidad, el humanismo, el patriotismo, la solidaridad, la honradez, la honestidad y la justicia social entre otros; lo que apunta hacia una educación más eficiente.

La confección de la Colección Multisaber que se encuentra actualmente en las escuelas primarias cubanas y que se encuentra desarrollada sobre software propietario estuvo a cargo de la dirección política de la Unión de Jóvenes Comunistas (UJC), bajo la asesoría del MINED por su implicación pedagógica, y del Ministerio de Informática y Comunicaciones (MIC) como instituciones que trabajan de forma coordinada para el logro de los objetivos trazados. A tal efecto se elaboró un modelo didáctico que responde a la tipología de Hiperentornos de Aprendizaje, que no son más que una mezcla sistemática de diversas tipologías del software, es decir, libros electrónicos, simuladores y bibliotecas virtuales; sustentados en tecnología hipermedia. Estos Hiperentornos de Aprendizaje integran armónicamente módulos como: Clases o Temas, Ejercicios, Juegos, Biblioteca y un Registro o Traza; este último registra el desenvolvimiento del estudiante en la interacción con los software de manera exhaustiva, y su análisis permite establecer estrategias de corrección y diagnóstico por parte del docente o tutor, así como el módulo dedicado al Maestro.

De esta manera se elaboraron 30 software educativos con el modelo pedagógico de la colección y se añadió un producto complementario sobre la vida y obra de nuestro Héroe Nacional denominado "El más puro de nuestra raza". Cada software está conformado por los módulos Ejercicios, Temas, Juegos, Biblioteca y Registro, donde este último, registra todo los resultados obtenidos por el estudiante en cada uno de los módulos antes mencionados.

Introducción

Pero la República Bolivariana de Venezuela no está exenta a esta aceptación que ha tenido el software educativo; pues en el sector de la educación, el intercambio y la colaboración de Cuba ayudó a la adquisición de programas y nuevas técnicas en el proceso docente educativo, un ejemplo de ello lo constituyen los nuevos software de la colección Multisaber, cual ya existe una versión en las escuelas primarias cubanas y se encuentra desarrollada en plataforma Windows 98 y con herramientas de apoyo como: Macromedia Flash 5, Macromedia Director, Toolbook, Borland Delphi, etc.; razón por lo cual se dificulta su despliegue para la República Bolivariana de Venezuela, pues como política nacional desde el 23 de Diciembre de 2004, fecha donde se promulga el Decreto 3390, se establece el uso prioritario del software libre en la administración pública nacional.

La confección de los nuevos software de la Colección Multisaber migrados a software libre se le asignó a la UCI, una universidad creada bajo el calor de la “Batalla de Ideas” y considerada la vanguardia en el dominio de los procesos de informatización y el uso de las nuevas tecnologías; que introduce un nuevo concepto basado en la vinculación estudio – trabajo, en la que la producción es un pilar fundamental y sus resultados favorecen el desarrollo económico, político y social del país.

Lo anteriormente planteado garantiza en los estudiantes de la universidad una eficiente preparación profesional sobre la base de la utilización tecnológica de la que dispone, la que ayuda a la materialización de los proyectos productivos que se desarrollan en cada una de las facultades de la misma.

En especial la Facultad 8, dedicada a la creación de multimedia o software educativo se encuentra formada por numerosos proyectos pertenecientes a ésta área de producción. El proyecto Multisaber formado por estudiantes de 3ro, 4to y 5to año de dicha facultad son los encargados de desarrollar la nueva colección migrada a software libre bajo la asesoría de profesores del MINED. Esta nueva colección contará con los módulos Ejercicios, Temas, Juegos, Mediateca y Resultados, donde este último, muestra un resumen de los resultados obtenidos por los estudiantes durante su interacción con el software.

Es importante señalar que la Colección Multisaber generalizada en las escuelas cubanas que sirve de guía para la confección de la nueva colección migrada a software libre, en algunos software no cuenta con los indicadores necesarios en el Módulo Registro, ya que no están todos los elementos que pudieran medir el desempeño de los escolares en la interacción con el software, y en el caso de los

Introducción

que lo poseen no brindaban abundante información para los usuarios careciendo dicha información de una adecuada organización y que se revierta en una búsqueda lógica para los alumnos y el personal docente; por lo que la nueva colección, necesita de un módulo que ayude a estos usuarios obtener toda la información concerniente a la trayectoria de los educandos durante la interacción con el software y que la misma le sirva al maestro para diseñar y proyectar el proceso de enseñanza – aprendizaje de forma óptima.

Este fenómeno planteado anteriormente unido a la necesidad de la migración de la Colección Multisaber a software libre, conlleva a plantear el siguiente **problema científico**: *¿Cómo realizar el análisis y diseño del Módulo Resultados de la nueva Colección Multisaber, de forma tal que le permita al maestro y a sus estudiantes observar el control académico y su trayectoria en su interacción con el software?*

Para resolver el problema planteado con anterioridad se traza como **objeto de estudio** de la investigación el “*proceso de desarrollo de los nuevos software de la Colección Multisaber migrados a software libre*”. De ello se deriva como **campo de acción** el “*proceso de desarrollo del Módulo Resultados de la Colección Multisaber migrada a software libre*”.

Por tanto el **objetivo general** de este trabajo de diploma es “*Analizar y diseñar el Módulo Resultados de la Colección Multisaber migrada a software libre*”.

Como **idea a defender** se define que “*Con el análisis y diseño del Módulo Resultados de la Colección Multisaber migrada a software libre, se obtendrá una aplicación de calidad, y se generará la documentación requerida del módulo, que servirá de guía para la creación o migración de productos similares*”.

Para darle cumplimiento al objetivo planteado con anterioridad y resolver los problemas mencionados se definen las siguientes **tareas de la investigación**:

- ✓ Caracterizar las tendencias actuales sobre el desarrollo del software educativo en el mundo.
- ✓ Investigar las metodologías existentes para el desarrollo de software educativo, establecer comparaciones entre ellas y escoger la más óptima para esta aplicación.

Introducción

- ✓ Analizar los lenguajes de modelado existentes, establecer una comparación entre ellos y seleccionar el que se adapte a las características de la aplicación que se quiere desarrollar.
- ✓ Analizar las herramientas CASE existentes y escoger la que se adapta a esta aplicación.
- ✓ Realizar los diagramas, vistas y modelos pertenecientes a los flujos de trabajo de Análisis y Diseño.

Como **aportes** de este trabajo de diploma se espera que *“Con el Análisis y Diseño del Módulo Resultados se pueda culminar la implementación del módulo, posibilitando la transferencia tecnológica y cursos de capacitación a futuros programadores que trabajen en el mantenimiento y el desarrollo de las nuevas versiones de los productos. Además, al concluir la implementación se distribuirán los software a las escuelas cubanas y venezolanas, permitiendo que dichas escuelas puedan contar con una importante herramienta de apoyo a la docencia”*.

Para el desarrollo de la investigación se han utilizado un conjunto de **métodos científicos** para la obtención, procesamiento y llegada a conclusiones. Dentro de estos se pueden mencionar los siguientes:

- **Métodos Teóricos:** que son los que nos permiten estudiar el objeto de la investigación. Estos se encuentran conformados por numerosos métodos, de los cuales se mencionan posteriormente los utilizados en este trabajo de diploma:
 - 1. Método de Análisis Histórico – Lógico:** para el estudio crítico de los trabajos anteriores y para utilizar estos trabajos como punto de referencia y comparar los resultados alcanzados en los mismos.
 - 2. Método de Modelación:** para modelar el sistema en cuestión y realizar todos los modelos pertenecientes a cada uno de los flujos de trabajo.

CAPÍTULO 1

Fundamentación Teórica

1.1 Introducción

A lo largo de la historia se han producido avances científicos e innovaciones tecnológicas que han marcado profundamente la evolución de la humanidad. A partir de la segunda mitad del siglo XX la rapidez sin precedentes en el desarrollo de estos dos aspectos modificó los paradigmas de la visión que los seres humanos teníamos acerca de las interacciones sociales; específicamente en la última década ocurrió una transformación en la manera de trabajar, aprender, comunicarnos, divertirnos, en definitiva, en nuestra manera de vivir, y en este proceso de cambio tuvieron un protagonismo decisivo las NTIC.

En el presente capítulo se hace alusión al significado de las Tecnologías de la Información y las Comunicaciones (TIC) mencionadas anteriormente y las ventajas que tienen estas en el proceso de enseñanza– aprendizaje. De la misma manera se hace referencia a los tipos de software existentes y se describe el objeto de estudio de este trabajo de diploma, los objetivos pedagógicos que persigue y la audiencia a la que va dirigida el mismo.

1.2 Las TIC en el proceso de enseñanza y aprendizaje.

(...) “Las Tecnologías de la Información y las Comunicaciones (TIC) no es más que el conjunto de avances tecnológicos que nos proporcionan la informática, las telecomunicaciones y las tecnologías audiovisuales. Estas comprenden los desarrollos relacionados con los ordenadores, Internet, la telefonía, las aplicaciones multimedia y la realidad virtual; proporcionando básicamente información, herramientas para su proceso y canales de comunicación” (...). (GRAELLS, 2005)

Las TIC se han convertido en un elemento básico de impulso y desarrollo de la denominada sociedad del conocimiento, pues sirven como medio de información, como escenarios y como fuente de motivación extrínseca para los estudiantes. Mucho se ha discutido en torno a las bondades y dificultades con la utilización de las TIC en la educación, pues estas permiten romper los clásicos escenarios formativos limitados a las instituciones educativas, proporcionando una mayor libertad en

Capítulo 1: Fundamentación Teórica

cuanto a los tiempos de estudio y desarrollo de actividades de aprendizaje, con un seguimiento periódico y riguroso.

A continuación se mencionan las principales ventajas que tiene la utilización de las TIC en el proceso de enseñanza y aprendizaje:

- ✓ Eliminan las barreras espacio-temporales entre el profesor y el estudiante.
- ✓ Flexibilizan la enseñanza, tanto en lo que respecta al tiempo, al espacio, a las herramientas de comunicación, como a los códigos con los cuales los alumnos pueden interactuar.
- ✓ Amplían la oferta formativa para el estudiante.
- ✓ Favorecen la creación de escenarios tanto para el aprendizaje cooperativo como para el auto-aprendizaje.
- ✓ Favorecen la interacción e interconexión de los participantes en la oferta educativa.
- ✓ Adaptan los medios y lenguajes a las necesidades, características, estilos de aprendizaje e inteligencia múltiples de los sujetos.
- ✓ Ayudan a comunicarse e interactuar con su entorno a los sujetos con necesidades educativas especiales.
- ✓ Y ofrecen nuevas posibilidades para la orientación y la tutorización de los estudiantes.

El éxito de estos procesos radica, no en poseer la tecnología más sofisticada, sino en implicar a los profesores en el diseño y desarrollo de innovaciones pedagógicas apoyadas en las TIC, para buscar el cambio en sus esquemas tradicionales de enseñanza, hacia la construcción de modelos didácticos más pertinentes y contextualizados.

El uso creciente de las TIC requiere de nuevas competencias en el individuo; su utilización debe conducir a un nuevo modelo de formación en el que el estudiante adopte un rol más activo en el proceso de enseñanza-aprendizaje y el profesor asuma como reto aplicar las TIC y participar activamente en su propio proceso de alfabetización digital y de cambio de sus paradigmas tradicionales. (ROA, 2006)

1.3 Tipologías de software.

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. El concepto de leer diferentes secuencias de

Capítulo 1: Fundamentación Teórica

instrucciones de la memoria de un dispositivo para controlar cálculos fue inventado por Charles Babbage como parte de su máquina diferencial. 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".

El software puede aplicarse a numerosas situaciones del tiempo real. En primer lugar, a todos aquellos problemas para los que se haya establecido un conjunto específico de acciones que lleven a su resolución. A continuación se muestran diferentes tipos de software:

Software de sistema: Es la parte que permite funcionar al hardware. Su objetivo es aislar tanto como sea posible al programador de aplicaciones de los detalles del computador particular que se use, especialmente de las características físicas de la memoria, dispositivos de comunicaciones, impresoras, pantallas, teclados, etcétera. Incluye, entre otros, sistemas operativos, controladores de dispositivos, herramientas de diagnóstico, servidores, sistemas de ventanas y utilidades.

Software de programación: Son los que proporcionan herramientas para ayudar al programador a escribir programas informáticos y a usar diferentes lenguajes de programación de forma práctica. Estos incluyen, entre otros, editores de texto, compiladores, intérpretes, enlazadores y depuradores. Los entornos de desarrollo integrados (IDE) agrupan estas herramientas mencionadas anteriormente, de forma que el programador no necesite introducir múltiples comandos para compilar, interpretar, depurar, etcétera, gracias a que habitualmente cuentan con una interfaz gráfica de usuario (GUI) avanzada.

Software de aplicación: permite a los usuarios llevar a cabo una o varias tareas más específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios. Incluye, entre otros, aplicaciones de automatización industrial como son: aplicaciones ofimáticas, software educativo, software médico, bases de datos, videojuegos y archivos de datos. (HISPALINUX, 2005)

1.3.1 Software libre vs software propietario.

El software propietario se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones). Este término fue utilizado por primera vez por Richard Stallman en el año 2003, en las conferencias sobre

Capítulo 1: *Fundamentación Teórica*

software libre, pues sería más adecuada que “software propietario” para definir en español al software que no es libre.

El software libre según la comunidad de desarrollo HISPALINUX, 2005 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.
- ✓ La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades. El acceso al código fuente es una condición previa para esto.
- ✓ La libertad de distribuir copias, con lo que puedes ayudar a tu vecino.
- ✓ La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

La libertad para usar un programa significa la libertad para cualquier persona u organización de usarlo en cualquier tipo de sistema informático, para cualquier clase de trabajo, y sin tener obligación de comunicárselo al desarrollador o a alguna otra entidad específica. Estas características que el software libre le otorga a los usuarios, empresas o a las administraciones públicas, hacen que éste tenga ciertas ventajas sobre el software propietario, las cuales se mencionan a continuación:

- ✓ **Bajo costo y libre uso:** Pues el software libre, como mercadería, por lo general no está a la venta, por lo que el usuario puede adquirir el software sin ninguna erogación monetaria o a muy bajo precio; con un conjunto de recursos muy amplios. Además cualquier persona con una computadora y una conexión a Internet puede utilizar el software libre, pues la mayoría de los usuarios individuales piensan que éste es una opción atractiva, por las libertades que garantiza sin la necesidad de verse agobiados por el precio. Todas estas razones no están disponibles en el software propietario pues hay que pagar una “licencia de uso” al creador (como el pago de derechos por el uso de una patente) y se está sujeto a las condiciones del fabricante. Normalmente estas condiciones no otorgan ningún derecho al usuario final.
- ✓ **Adaptación del software:** El software propietario habitualmente se vende en forma de paquete estándar, que muchas veces no se adapta a las necesidades específicas de empresas y administraciones. Una gran parte de la industria del software se basa en desarrollar proyectos donde se requiere software personalizado. El software libre permite personalizar, gracias al

Capítulo 1: *Fundamentación Teórica*

hecho de que disponemos del código fuente, los programas tanto como sea necesario hasta que cubran exactamente nuestra necesidad. La personalización es un área muy importante en que el software libre puede responder mucho mejor que el software propietario a unos costos mucho más razonables.

- ✓ **Innovación Tecnológica:** El software libre, tiene como objetivo principal compartir la información, trabajando de manera cooperativa, pues este es el principal modelo sobre el que la humanidad ha innovado y avanzado.
- ✓ **Escrutinio público:** Pues el modelo de desarrollo de software libre sigue un método a través de la cual trabajan de forma cooperativa los programadores, que en gran parte son voluntarios y trabajan coordinadamente en Internet. Lógicamente, el código fuente del programa está a la vista de todo el mundo, y son frecuentes los casos en que se reportan errores que alguien ha descubierto leyendo o trabajando con ese código. El proceso de revisión pública al que está sometido el desarrollo del software libre imprime un gran dinamismo al proceso de corrección de errores; pues los usuarios que tienen el software de todo del mundo, gracias a que disponen del código fuente de dicho programa, pueden detectar sus posibles errores, corregirlos y contribuir a su desarrollo y mejoras.
- ✓ **Independencia del proveedor:** El software libre garantiza una independencia con respecto al proveedor gracias a la disponibilidad del código fuente. Cualquier empresa o profesional, con los conocimientos adecuados, puede seguir ofreciendo desarrollo o servicios para nuestra aplicación.

1.4 La migración a software libre en Venezuela.

El software libre aparece en la República Bolivariana de Venezuela cuando el Gobierno Electrónico tiene como premisa fundamental la democratización del acceso a Internet y al Conocimiento en el marco del Plan Nacional de Tecnologías de Información y Comunicación; a través del cual se emprende el camino hacia el proceso de modernización del Estado, con la finalidad de proporcionar mayores y mejores servicios a la ciudadanía. Este plan, es un instrumento que orienta la acción del Ejecutivo Nacional, con el propósito de establecer los lineamientos y políticas nacionales en materia de ciencia, tecnología e innovación.

Capítulo 1: Fundamentación Teórica

El artículo 110 de la Constitución de la República Bolivariana de Venezuela, reconoce como de interés público la ciencia, la tecnología, el conocimiento, la innovación y los servicios de información, con el objeto de lograr el desarrollo económico, social y político del país. Esta disposición constitucional se expresa con fuerza en los artículos 1° de la Ley de Telecomunicaciones y 12° de la Ley Orgánica de la Administración Pública. Con el Decreto N° 825, emitido el 10 de mayo de 2000, se establece el acceso y el uso de Internet como política prioritaria para el desarrollo cultural, económico, social y político del Estado.

Este Decreto orienta la formulación de políticas públicas tendentes a la promoción y masificación del uso de Internet por parte de la ciudadanía, con el fin de impulsar la democratización del acceso a Internet. Asimismo, contempla que el Estado, a través del Ministerio de Ciencia y Tecnología (MCT), debe promover activamente el desarrollo académico, científico y cultural para así lograr el acceso adecuado y uso efectivo de Internet. Todo ello, con la intención de contribuir con el desarrollo de la investigación y el conocimiento en el sector de las tecnologías de la información y la prestación de servicios al ciudadano.

Basado en estos aspectos legales, el desarrollo del Gobierno Electrónico en Venezuela, comenzó sus primeros pasos creando acceso a los servicios públicos, por medio de la Internet. Es así, como la ciudadanía obtiene el derecho a disponer de trámites y servicios que deben ofrecerse de manera gratuita, clara, sencilla, ágil, racional, pertinente y de fácil entendimiento. Con esto se persigue mejorar las relaciones de las personas con el Estado y facilitar la eficiencia de la prestación de servicios.

En la lucha por lograr la independencia, soberanía y la autonomía tecnológica, el presidente de la República Bolivariana de Venezuela, Hugo Chávez Frías, emitió el Decreto N° 3.390, publicado en la Gaceta Oficial N° 38.095 de fecha 28/ 12/ 2004 sobre el uso obligatorio del software libre en el país para todas las dependencias públicas de carácter oficial. De esta forma, el Ejecutivo Nacional establece que es prioridad del Estado incentivar y fomentar la producción de bienes y servicios para satisfacer las necesidades de la población, mediante el uso de estas herramientas desarrolladas con estándares abiertos para robustecer la industria nacional, aumentando y aprovechando sus capacidades y fortaleciendo nuestra soberanía. La puesta en marcha de este instrumento legal, por parte del presidente Chávez, faculta al Ministerio de Ciencia y Tecnología, para capacitar y formar a los funcionarios públicos en esta materia.

Capítulo 1: Fundamentación Teórica

En el Centro Nacional de Tecnologías de Información (CNTI) el 28 de diciembre de 2004 fue oficializado el uso del software libre en la Administración Pública. A partir de esa fecha, el Ministerio de Ciencia y Tecnología en un plazo de tres meses, tuvo bajo su responsabilidad la elaboración del Plan Marco, el cual fue entregado el 28 de marzo de 2005 al Jefe de Estado para su revisión y aprobación. Una vez que dicho Plan recibió el visto bueno, todas las instituciones gubernamentales y sus entes adscritos, dispusieron de un lapso de tres meses para presentar su plan de migración que incluye sistemas operativos, plataforma, aplicaciones y servicios hacia los próximos dos años. (CHACÓN, 2008)

El Centro Nacional de Tecnologías de Información adscrito al Ministerio de Ciencia y Tecnología juega un papel importante en el desarrollo de las fases contempladas en el Decreto N° 3.390, a través de la gerencia de Asistencia al Estado, a cargo de Fernando Muro, donde se acompaña y orienta a los organismos gubernamentales en su proceso de migración. Este soporte consiste en capacitación y asesoramiento técnico, mediante la Academia del Software Libre la cual le proporcionará al funcionario público y al usuario, los conocimientos necesarios para alcanzar el dominio de este tipo de herramientas.

1.5 Ventajas y retos del software libre en Venezuela.

El uso del software libre contrarresta el individualismo que difunde el software licenciado porque se desarrolla bajo una concepción de cooperación entre un grupo de personas, cooperativas, instituciones del Estado, organizaciones sociales, etc. Con el Decreto N° 3.390 establecido en Venezuela se otorga prioridad a la gestión de Gobierno, mediante el uso de esta tecnología e igualmente se establece la importancia primordial del uso del software libre en la Administración Pública frente al Software propietario o licenciado.

Según el gerente de Educación e Investigación del Centro Nacional de Tecnologías de Información Domingo Monzón, “el software libre procura atender el uso de la Soberanía Tecnológica” pues cuando se trabaja con software propietario no se puede auditar el código fuente y en el momento de replicar las experiencias, las licencias impiden realizar esa acción. Por lo tanto, una de las ventajas que proporciona la tecnología abierta es que el resultado de esa práctica es replicable, porque la libertad permite la redistribución de la experiencia, en la que participa una comunidad de actores.

Capítulo 1: Fundamentación Teórica

Pero al mismo tiempo representa un reto el software libre en Venezuela pues con la utilización de éste se disminuye en gran parte lo que significa la brecha digital, que se define como la distancia que separa a diferentes entes, sectores, naciones o comunidades de la Sociedad del Conocimiento y de las Tecnologías de Información y Comunicación. La aplicación de estas tecnologías abiertas facilita a los usuarios estar en la constante búsqueda de información, y en contacto con el manejo de herramientas que proporcionan acceso a la información, al conocimiento y la educación y por consiguiente ayudar al desarrollo de los pueblos, mejorando la calidad de vida de sus ciudadanos.

Si este reto se aborda desde el punto de vista económico, el Estado debe proporcionar apoyo a la industria nacional de software, con la intención de garantizar la prestación de mejores servicios a la ciudadanía y contribuir a la generación de condiciones de vida más satisfactorias.

Finalmente, el software libre es un movimiento que se está gestando a nivel mundial, (una revolución tecnológica) y sirve de incentivo para que los entes públicos comiencen agilizar los procesos de cambios que se están llevando a cabo en esta materia. El Centro Nacional de Tecnologías de Información, como un organismo adscrito al MCT, debe promover el fomento y desarrollo del software libre, no sólo en las instituciones del Estado, sino también en la empresa privada; para que de este modo ambos sectores trabajen juntos y de manera muy estrecha con el fin último de alcanzar el desarrollo de esta gran nación. (CHACÓN, 2008)

1.6 Descripción del Objeto de Estudio.

1.6.1 Descripción general.

El **Software educativo** es el software destinando a la enseñanza y el auto aprendizaje; además permite el desarrollo de ciertas habilidades cognitivas. Así como existen profundas diferencias entre las filosofías pedagógicas, pues 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, computador.

El objeto de estudio de este trabajo de diploma está orientado como se mencionó anteriormente al proceso de desarrollo de los nuevos software de la Colección Multisaber migrados de software propietario a software libre.

Capítulo 1: Fundamentación Teórica

Se está en presencia de una colección con una función educativa, de aprendizaje en las diferentes materias para el alumnado de la República Bolivariana de Venezuela. Una colección que se encuentra en desarrollo haciendo uso de herramientas, metodologías, plataformas, entre otras, para software libre que es el principal objetivo para lograr la migración. Una colección que regulará el aprendizaje de los estudiantes encaminado a facilitar el logro de los objetivos educativos específicos, la interactividad propia de este, les permite responder inmediatamente a las respuestas y acciones de los estudiantes y lo que es adecuado para evaluar el trabajo que se va realizando con ellos, sus actividades presentan unos contenidos que proporcionan información a los estudiantes.

1.6.2 Identificación de la audiencia.

Uno de los aspectos más importantes a la hora de desarrollar un producto de alta calidad y que cumpla con los objetivos para los que se concibió es definir a la audiencia a la que va dirigida; en este caso la audiencia a la que va dirigida el Módulo Resultados de los nuevos software de la Colección Multisaber es a profesores y estudiantes de las escuelas bolivarianas de Venezuela. El primer ciclo de dicho módulo va dirigido a estudiantes entre primero y quinto grado y el segundo ciclo a estudiantes de quinto y sexto grado.

1.7 Plataforma Metodológica.

1.7.1 Objetivos Pedagógicos.

Los objetivos pedagógicos que tiene la Colección Multisaber específicamente el Módulo Resultados es mostrar un resumen de los resultados obtenidos por los estudiantes en su interacción con el software.

1.8 Análisis del modelo de arquitectura de información utilizada.

1.8.1 Estándares de la interfaz de la aplicación.

La interfaz gráfica del usuario es el medio por el cual este interactúa con el sistema, por lo que ésta debe ser lo más amigable posible y lograr que el mismo se sienta identificado con dicha interfaz.

Para el diseño de la interfaz del sistema se debe tener en cuenta que cada uno de los elementos que se repite en cada interfaz de comunicación con el usuario, debe de estar ubicado en el mismo lugar en

Capítulo 1: Fundamentación Teórica

cada una de estas interfaces; permitiéndole al mismo acostumbrarse al ambiente y que no se encuentre desorientado. Además cada interfaz de comunicación debe de contener la información necesaria para el usuario, y que estas informaciones sean uniformes logrando un balance entre los elementos que la componen.

1.9 Conclusiones

A lo largo de este capítulo se han ofrecido los elementos teóricos que sirven de sustento científico a esta investigación, se ofreció el concepto y las ventajas que tienen las nuevas Tecnologías de la Información y las Comunicaciones, las cuales han hecho emerger un nuevo tipo de sociedad, la sociedad de la información, que forma a sus integrantes en el conocimiento y la habilidad de explotar a futuras tecnologías.

Se analizaron además varias tipologías de software donde cada uno presentan características importantes; así como las ventajas y los retos que tiene el software libre en la República Bolivariana de Venezuela y quedaron claramente argumentadas las bases que sustentan este trabajo investigativo como son: la descripción del objeto de estudio, la identificación de la audiencia a la que va dirigida y los objetivos pedagógicos que tiene el mismo.

CAPÍTULO 2

Tendencias y Tecnologías actuales

2.1 Introducción

Una de las tendencias en la modernización de la clase en la actualidad, lo constituye la utilización de los recursos didácticos como fuentes de aprendizaje, especialmente, diseñados para ser utilizados con un computador en los procesos de enseñanza a los estudiantes. En el siguiente capítulo se hace referencia a las tendencias actuales que tiene la informatización del aprendizaje, así como el concepto de Informática Educativa como recurso didáctico y la repercusión que tiene el uso del software educativo como medio de enseñanza.

De la misma forma se hace alusión a diferentes metodologías apropiadas para obtener un buen producto de software, así como lenguajes de modelado y herramientas de modelación para el desarrollo del mismo y preceptos ingenieriles que se deben llevar a cabo para su desarrollo.

2.2 Tendencias actuales para desarrollar la propuesta.

2.2.1 Informatización del aprendizaje.

La Informatización del aprendizaje es la utilización integral de los recursos de la tecnología de la información en todas sus modalidades para potenciar la actividad de aprender. Todo recurso tecnológico que permita almacenar, procesar y recuperar información, ya sea datos numéricos, conceptos, imágenes, sonidos, etc., amplía el potencial de la inteligencia humana y puede ser aplicado para enriquecer el aprendizaje. El principal objetivo de la informatización del aprendizaje es extra computacional porque debe tender a mejorar la eficacia del proceso de enseñanza-aprendizaje. Significa potenciar la actividad del educando, la interacción con el docente y con sus educandos y la comprensión de los contenidos curriculares desde una concepción que parte del constructivismo pero no se circunscribe únicamente a él. (MAGLIO, 1999)

Capítulo 2: Tendencias y Tecnologías actuales

2.2.2 Informática Educativa

La informática es un recurso didáctico y abarca al conjunto de medios y procedimientos para reunir, almacenar, transmitir, procesar y recuperar datos de todo tipo. Abarca a las computadoras, teléfono, televisión, radio, etc. Estos elementos potencian las actividades cognitivas de las personas a través de un enriquecimiento del campo perceptual y las operaciones de procesamiento de la información.

Las Nuevas Tecnologías contribuyen, a través de una configuración sensorial más compleja que la tradicional, a esclarecer, estructurar, relacionar y fijar mejor los contenidos a aprender. Podemos vincular el recurso informático con la llamada tecnología del aprender a pensar, basada en:

La destreza para la planificación de estrategias de resolución de problemas por parte del docente y sus alumnos.

La creación del descubrimiento de principios y reglas lógicas de inferencia y deducción. De esta forma se aprenden conceptos básicos que pueden ser transferidos a situaciones nuevas.

El desarrollo de algoritmos para localizar información definida dentro de una gran masa de conocimientos.

Las condiciones de transferencia de conocimientos a campos diferentes y diferidos en el tiempo, en el espacio, etc.

En esta concepción, la computadora se considera como una extensión de la inteligencia humana. Las capacidades intelectuales de análisis, comparación, modelización, cálculo, graficación, deducción, etc., pueden amplificarse con el uso de la herramienta computacional. Pero hay que tener en cuenta que la disponibilidad de la herramienta no constituye en sí misma una experiencia de aprendizaje. (MAGLIO, 1999)

2.2.3 Repercusión del Software Educativo en la educación.

Un software educativo es todo programa que se desarrolla con la finalidad específica de ser utilizado como recurso didáctico en procesos de enseñanza y de aprendizaje. Los primeros intentos de desarrollo de software educativo se sitúan al final de la década del 60 con la aparición de los sistemas de instrucción programada, pero el verdadero auge se dio en la década del 80. En primera instancia con la producción de lenguajes para el aprendizaje, luego con el desarrollo de herramientas de autor para la producción de software educativo y ya más específicamente con la elaboración de programas tutoriales, de ejercitación y práctica, de cálculo, y de simulación.

Capítulo 2: Tendencias y Tecnologías actuales

Desde sus inicios y a través de los años se han ido incrementando las entidades encargadas del desarrollo de software educativo. En algunos casos, han sido editoriales especializadas las que han desarrollado este tipo de software en los últimos tiempos. En la actualidad es posible encontrar en la Web, gran variedad de software educativo desarrollados por aficionados con algunos conocimientos en el área, siendo esta una producción menos profesional.

En las últimas décadas el software educativo se ha convertido en una de las herramientas más poderosas que existen, pues vinculado con las tecnologías multimedia resultan un medio de aprendizaje tanto para los educandos como para los profesores ya que contienen materiales docentes provenientes de diferentes fuentes como son: textos, imágenes, audio, video, simulaciones, fotografías, esquemas, mapas conceptuales, entre otras. Estos recursos mencionados anteriormente se combinan a través de la interactividad y se crean las posibilidades para el desarrollo de un entorno educativo realmente efectivo y tan centrado en el estudiante que más que llamarlo medio de enseñanza, resultaría más correcto llamarlo medio de aprendizaje. (DANIELLE, 2005)

2.3 Tecnologías actuales para desarrollar la propuesta.

2.3.1 Tecnología Multimedia.

En el ámbito de la computación el término **multimedia** es nuevo y designa el uso de varios recursos o medios, es decir, es un programa que permite interactuar con imágenes, vídeo, sonido, animación y texto. Todos estos elementos han de organizarse y controlarse de forma que lleguen al usuario final como una aplicación cómoda de utilizar en la que puede desplazarse a voluntad por los distintos elementos que forman la aplicación.

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

Dicha tecnología toma auge, a partir de 1992, cuando se integran: audio (música, sonido estéreo y voz), video, gráficas, animación y texto al mismo tiempo. La principal idea multimedia desarrollada en los videos juegos era: que se podía navegar y buscar la información que se deseaba sobre un tema,

Capítulo 2: Tendencias y Tecnologías actuales

sin tener que recorrer todo el programa, de manera que se pueda interactuar con la computadora y la información de manera asociativa y no lineal. (GARCIA, 2003)

2.3.2 Tecnología Hipertexto.

(...) “La tecnología hipertexto se ha definido como un enfoque para manejar y organizar información, en el cual los datos se almacenan en una red de nodos conectados por enlaces, los cuales contienen textos, gráficos, imágenes, audio, animaciones y video, así como código ejecutable u otra forma de datos que se les da el nombre de **hipermedio**, es decir, una generalización de hipertexto”.
(...)(BIANCHINI, 2000)

El hipertexto es una tecnología que organiza una base de información en bloques distintos de contenidos, conectados a través de una serie de enlaces cuya activación o selección provoca la recuperación de información.

Un sistema hipertexto, en términos ideales, debe cumplir con las siguientes características:

- ✓ Esta tecnología debe proveer un medio adecuado para organizar y presentar información poco o nada estructurada, no ajustada a esquemas tradicionales y rígidos como es el caso de las bases de datos. Pueden utilizarse esquemas jerárquicos para la utilización de sistemas de documentación de texto tradicionales, muy organizados o simplemente creando estructuras de redes con poco o ningún atributo de precedencia.
- ✓ Tener asociada una interfaz de usuario muy intuitiva, pues se pretende imitar el funcionamiento de la mente humana, haciendo uso de modelos cognitivos, por lo que el usuario no debería realizar grandes esfuerzos para obtener la información requerida.
- ✓ La información se encuentra distribuida y puede ser accedida en forma concurrente por varios usuarios, por lo tanto es un ambiente compartido.
- ✓ Es un ambiente colaborativo: un usuario puede crear nuevas referencias entre dos documentos cualesquiera en forma inmediata e independiente de los tipos de contenido, haciendo crecer su hiperdocumento, sin generar cambios en el hiperdocumento referenciado. Estas referencias pueden estar embebidas en el documento, de modo que aunque éste se cambiara de instalación, el enlace seguiría proporcionando acceso a la información referenciada.

Capítulo 2: Tendencias y Tecnologías actuales

- ✓ Tiene asociados varios mecanismos de recuperación y búsqueda de información a través de las navegaciones, ya sean dirigidas o no dirigidas. (BIANCHINI, 2000)

2.3.3 Tecnología Hipermedia.

Muchos investigadores consideran que los términos Hipertexto e Hipermedia son sinónimos. En general, el Hipertexto se refiere a elementos de texto relacionados, mientras que Hipermedia; que es un acrónimo que combina las palabras Hipertexto y Multimedia; incluye relaciones entre elementos de cualquier tipo de medio (texto, imágenes, sonidos, animaciones, videos, etc.). Los conceptos son idénticos, pero el Hipertexto es más difícil de implementar en un medio no textual. (GARCIA, 2003)

Pero para el desarrollo de estas tecnologías hay que tener claro algunos aspectos ingenieriles que son de suma importancia cómo es: ¿Qué es la Ingeniería de Software? Y todo lo que respecta a ella.

2.4 Aspectos de la Ingeniería de Software.

Según la definición del IEEE citada por (Lewis, 1994), "**software** es la suma total de los programas de computadora, procedimientos, reglas, documentación asociada y los datos que pertenecen a un sistema de cómputo". En este contexto, la Ingeniería de Software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software, que en palabras más llana; se considera que es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software, es decir, permite elaborar consistentemente productos correctos, utilizables y costo-efectivos (Cota, 1994).

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". (Jacobson, 1998) 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 es lo que está haciendo, cuándo hacerlo y cómo alcanzar un cierto objetivo". (Jacobson, 1998)

Capítulo 2: Tendencias y Tecnologías actuales

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el **ciclo de vida del software** que comprende cuatro grandes fases: concepción o inicio, elaboración, construcción y transición. La concepción o inicio define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios. (ZAVALA, 2002)

La Ingeniería de Software está conformada por:

- **Herramientas:** Son el soporte automático o semiautomático a los métodos, orientadas a etapas particulares en el diseño de un software.
- **Métodos:** Son cómo se construye técnicamente el software.
- **Procedimientos:** Representan la secuencia en que se aplican los métodos, entregas y controles. Estos son los que unen los métodos con las herramientas. (CAMACHO and RODRÍGUEZ, 2006)

2.4.1 Análisis de las metodologías existentes.

En lo relativo a las metodologías, estas aparecen por la necesidad de poner orden al proceso de construcción del software. Resulta importante que el desarrollo de sistemas informáticos sea tratado bajo una disciplina ingenieril, con el fin de desarrollar e implantar sistemas realmente eficaces y eficientes.

A continuación se hace alusión a un grupo de metodologías y sus características que pueden ayudar a alcanzar un buen producto.

2.4.1.1 Metodología de Administración de Relaciones (RMM)

La **Metodología de Administración de Relaciones (RMM)** se define como un proceso de análisis, diseño y desarrollo de aplicaciones hipermedia. Los elementos principales de este método son el modelo Entidad-Relación (E-R) y el modelo RMDM (Relationship Management Data Model) basado en el modelo HDM. La metodología fue creada por Isakowitz, Stohr y Balasubramanian. Esta metodología

Capítulo 2: Tendencias y Tecnologías actuales

es apropiada para dominios con estructuras regulares (es decir, con clases de objetos bien definidas, y con claras relaciones entre esas clases). Según sus autores, está orientada a problemas con datos dinámicos que cambian con mucha frecuencia, más que a entornos estáticos.

El modelo en esta metodología propone un lenguaje que permite describir los objetos del dominio, sus interrelaciones y los mecanismos de navegación hipermedia de la aplicación, donde los objetos del dominio se definen con la ayuda de entidades, atributos y relaciones asociativas. El esquema completo del dominio y de la navegación de la aplicación se denomina esquema RMDM y se obtiene como resultado de las tres primeras etapas del método las cuales se mencionan a continuación:

- ✓ **Primera etapa:** representar los objetos del dominio con la ayuda del modelo Entidad-Relación ampliado con relaciones asociativas (aquéllas que permiten representar caminos de navegación entre entidades puestos en evidencia en la fase de análisis).
- ✓ **Segunda etapa:** determinar la presentación del contenido de las entidades de la aplicación así como su modo de acceso. El esquema obtenido como resultado de esta etapa se denomina esquema E.R+. Se trata de un esquema Entidad-Relación en el que cada entidad ha sido reemplazada por su esquema de entidad. Un esquema de entidad está constituido por nodos (los trozos o slides) unidos por relaciones estructurales.
- ✓ **Tercera etapa:** definir los caminos de navegación inducidos por las relaciones asociativas del esquema E-R+. A continuación, es posible definir estructuras de acceso de alto nivel (agrupaciones), lo que permite dotar a la aplicación de accesos jerárquicos a niveles diferentes de los trozos de información. El esquema RMDM resultante se obtiene añadiendo al esquema E-R+ las agrupaciones y caminos de navegación definidos en esta etapa.

En RMM, el modelo hipermedia retoma los elementos enlace, índice y visitas guiadas de HDM enriqueciéndolos con capacidades condicionales. Sin embargo, el método no permite al diseñador definir elementos hipermedia propios que tengan capacidades específicas ya que impone la utilización de metáforas preestablecidas. (LAPUENTE, 2007)

2.4.1.2 Metodología de Diseño Hipermedia Orientada a Objetos (OOHDM)

La **Metodología de Diseño Hipermedia Orientada a Objetos (OOHDM)** fue diseñado por D. Schawabe, G. Rossi y S. J. Barbosa y es una extensión de HDM con orientación a objetos, que se está convirtiendo en una de las metodologías más utilizadas. Ha sido usada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia y, sobre todo, numerosos sitios web.

Al igual que RMM, este método se inspira en el modelo HDM, pero lo que le distingue claramente del primero es el proceso de concepción orientado a objetos. OOHDM propone el desarrollo de aplicaciones hipermedia mediante un proceso de 4 etapas, las cuales se mencionan a continuación:

- ✓ **Diseño conceptual:** Es donde se construye un esquema conceptual representado por los objetos de dominio o clases y las relaciones entre dichos objetos. Se puede usar un modelo de datos semántico estructural (como el modelo de entidades y relaciones). El modelo OOHDM propone como esquema conceptual basado en clases, relaciones y subsistemas.
- ✓ **Diseño de navegación:** En esta etapa el diseñador define clases de navegación tales como nodos, enlaces y estructuras de acceso (índices y visitas guiadas) inducidas del esquema conceptual, donde los enlaces derivan las relaciones y los nodos representan ventanas lógicas sobre las clases conceptuales. De esta manera el diseñador describe la estructura de navegación en términos de contextos de dicha navegación, estos no son más que un conjunto de nodos, enlaces, clases de contexto, y otros contextos navegacionales (contextos anidados) que igual que en HDM definen agrupaciones que pueden ser definidos por compresión o extensión, o por enumeración de sus miembros. Durante esta etapa, es posible adaptar los objetos navegacionales para cada contexto, de forma similar a las perspectivas de HDM.

OOHDM no propone un modelo enriquecido para el dominio de la aplicación, por lo que deja libre al diseñador para elegir el modelo de especificación del dominio. Sin embargo, el modelo hipermedia está definido en dos niveles de abstracción: las clases navegacionales y los contextos navegacionales.

- ✓ **Diseño de interfaces abstractas:** En esta etapa OOHDM se dedica a la especificación de la interfaz abstracta. Así, se define la forma en la cual deben aparecer los contextos navegacionales. También se incluye aquí el modo en que dichos objetos de interfaz activarán la

navegación y el resto de funcionalidades de la aplicación, esto es, se describirán los objetos de interfaz y se los asociará con objetos de navegación. La separación entre el diseño de navegación y el diseño de interfaz abstracta permitirá construir diferentes interfaces para el mismo modelo de navegación.

- ✓ **Implementación:** Esta etapa se dedicada a la puesta en práctica, es decir, es donde se hacen corresponder los objetos de interfaz con los objetos de implementación. (LAPUENTE, 2007)

2.4.1.3 Programación Extrema (XP)

La metodología **Programación Extrema (XP)** nace como nueva disciplina de desarrollo de software hace algunos años atrás y ha causado un gran revuelo entre el colectivo de programadores del mundo. Kent Beck, su autor, es un programador que ha trabajado en múltiples empresas y que actualmente lo hace como programador en la conocida empresa automovilística DaimlerChrysler. Con sus teorías ha conseguido el respaldo de gran parte de la industria del software y el rechazo de otra parte. La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica.

XP es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo o corto equipo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. (SOLÍS, 2003)

La metodología XP se basa en las siguientes características, las cuales se mencionan a continuación:

- ✓ **Pruebas Unitarias:** La cual se basa es las pruebas realizadas a los principales procesos, de tal manera de adelantarnos en algo hacia el futuro, es decir, adelantándonos a los posibles errores.
- ✓ **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

Capítulo 2: Tendencias y Tecnologías actuales

- ✓ **Programación en pares:** Que es una particularidad de esta metodología, pues propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

Lo fundamental en esta metodología es que se basa en la comunicación entre los usuarios y los desarrolladores, así como en la simplicidad, al desarrollar y codificar los módulos del sistema y que logra la retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. (SANCHEZ, 2004)

2.4.1.4 Proceso Unificado del Rational (RUP)

El **Proceso Unificado de Rational (RUP)** es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto. Esta basado en componentes y utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado.

El **Proceso Unificado** "es un proceso de desarrollo de software configurable que se adapta a través de los proyectos variados en tamaños y complejidad. Se basa en muchos años de experiencia en el uso de la tecnología orientada a objetos en el desarrollo de software de misión crítica en una variedad de industrias por la compañía Rational", donde confluyen 'los tres amigos' como se llaman así mismos o los tres grandes OO: Grady Booch, James Rumbaugh e Ivar Jacobson. (M &R, 1998)

El Proceso Unificado del Rational guía a los equipos del proyecto en cómo administrar el desarrollo iterativo de un modo controlado, mientras se balancean los requerimientos del negocio, el tiempo al mercado y los riesgos del proyecto. El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura. El proceso describe qué entregables producir, cómo desarrollarlos y también provee patrones. Además éste es un proceso unificado soportado por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas.

Capítulo 2: Tendencias y Tecnologías actuales

El Proceso Unificado está compuesto por fases las cuales se mencionan a continuación:

- ✓ **Concepción o Inicio:** Es donde se define un plan de fases y se identifican los principales casos de usos. Además es donde se define la arquitectura y se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido y se identifican los riesgos.
- ✓ **Elaboración:** Es donde se hace un plan de proyecto, se completan los casos de usos y se eliminan los riesgos.
- ✓ **Construcción:** Se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- ✓ **Transición:** Se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados. (ZAVALA, 2002)

RUP se caracteriza en su ciclo de vida por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de usos. Además de incluir artefactos (que son los productos tangibles del proceso) y roles que son el papel que desempeña una persona en un determinado momento o una persona puede desempeñar distintos roles a lo largo del proceso.

2.4.1.5 Selección y fundamentación de la metodología de desarrollo escogida.

Después realizar algunas valoraciones respecto a las metodologías existentes, se llega a la conclusión que la metodología más óptima y adaptable a esta aplicación es RUP, pues es un proceso de la Ingeniería de Software que utiliza el paradigma orientado a objetos para su descripción, además de ser un proceso configurable para satisfacer necesidades específicas e implementa las mejores prácticas de desarrollo de software. También porque esta metodología presenta tres características principales que sirven de guía para cualquier producto de software, dichas características son: que es dirigido por casos de usos, pues estos son los que capturan los requerimientos funcionales y representan piezas de funcionalidad que brindan un resultado de valor al usuario; centrado en la arquitectura pues comprende los aspectos estáticos y dinámicos más importantes del sistema e iterativo e incremental

Capítulo 2: Tendencias y Tecnologías actuales

porque el trabajo se divide en piezas pequeñas o proyectos pequeños donde cada uno provee un producto incremental.

2.4.2 Análisis de los lenguajes de modelado existentes.

El lenguaje de modelado es un conjunto estandarizado de símbolos de modo de disponerlos para modelar parte de un diseño de software orientado a objetos. Estos se combinan con las metodologías de desarrollo para avanzar de una especificación inicial a un plan de implementación, 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 realizado. Esto puede suponer también que las interacciones entre partes del programa den lugar a sorpresas cuando el modelo ha sido convertido en un software. A continuación se mencionan diferentes lenguajes de modelado para así escoger cual es el más óptimo para esta aplicación.

2.4.2.1 Lenguaje Unificado de Modelado (UML)

El **Lenguaje Unificado de Modelado (UML)** es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software. Se ha convertido en el estándar de facto de la industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh, en la década de los 90.

El Lenguaje Unificado de Modelado se caracteriza por dividir el sistema en áreas conceptuales las cuales se mencionan a continuación: área de estructura estática, área de comportamiento dinámico, área de construcciones de implementación, área de organización del modelo y área de mecanismos de extensión; las cuales pretenden unificar y estandarizar las técnicas de modelado permitiendo conocer, configurar, mantener y controlar la información. Este lenguaje propone un vocabulario que incluye tres categorías: elementos, relaciones y diagramas, y estas a su vez están compuestas por diferentes categorías.

La notación de UML ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa: Booch, OMT y OOSE. UML ha puesto fin a las llamadas “guerras de métodos” que se han mantenido a lo largo de los 90, en las que los principales métodos sacaban nuevas versiones que incorporaban las técnicas de los demás. Con UML se fusiona la notación de estas técnicas para formar una herramienta

Capítulo 2: Tendencias y Tecnologías actuales

compartida entre todos los ingenieros de software que trabajan en el desarrollo orientado a objetos. (GRAU and SEGURA, 2006)

2.4.2.2 Lenguaje Orientado a Objetos para la Modelación de Aplicaciones Multimedia (OMMMA - L)

El **Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia (OMMMA-L)** se lanza como una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos, y el Modelo Vista Controlador (MVC) para la interfaz de usuario, siendo este un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos. Esto es útil ya que los modelos típicamente tienen cierto grado de estabilidad (dependiendo de la estabilidad del dominio del problema que está siendo modelado), donde el código de la interfaz de usuario sea más robusto, debido a que el desarrollador está menos propenso a "romper" el modelo mientras trabaja de nuevo en la vista. OMMMA-L está sustentado en cuatro vistas fundamentales, donde cada una se asocia a un tipo de diagrama en particular. Estas vistas son:

- **Vista Lógica:** modelada a través del Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML, utilizando las mismas notaciones, pero incorporando las clases correspondientes a las medias: media continua y media discreta, generalizadas en una clase medias. Divide en dos áreas dicho diagrama: una para la jerarquía de los tipos de media y otra para la modelación de la estructura lógica del dominio de la aplicación.
- **Vista de Presentación espacial:** modelada a través de los Diagramas de Presentación de OMMMA-L, los cuales son de nueva aparición en la extensión de UML, dado que este último no contiene un diagrama apropiado para esta tarea. Estos diagramas tienen el propósito de declarar las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área, dividiéndose en objetos de visualización (texto, gráfico, video, animación) e interacción (barras de menú, botones, campos de entrada y salida, scrolls, hipertextos con hipervínculos). Estos diagramas de presentación pueden ser divididos en capas virtuales de presentación donde en cada uno de ellas sólo se haga referencia a una clase específica de componentes (por ejemplo, una vista para los objetos de visualización y otra para los de interacción, u otro tipo de división para la representación de los intereses de los desarrolladores.

- **Vista de Comportamiento temporal predefinido:** modelada por el Diagrama de Secuencia de OMMMA-L, extendido a partir del diagrama de secuencia de UML. El Diagrama de secuencia modela una secuencia de una presentación predefinida dentro de una escena, donde todos los objetos dentro de un diagrama se relacionan al mismo eje del tiempo. En este diagrama se hace un refinamiento del eje del tiempo con la introducción de marcas de tiempo a través de diferentes tipos de intervalos; marcas de inicio y fin de ejecución que permite soportar su reusabilidad; marcas de activación y desactivación de demoras en objetos de tipo media, posibilitando la modelación de las tolerancias de la variación de las restricciones de sincronización para los objetos media; activación compuesta de objetos media para la agrupación de objetos concurrentemente activos.

- **Vista de Control Interactivo:** modelado a través del Diagrama de Estado, extendido a partir del diagrama de estado de UML, sintácticamente igual a este último, mas con la diferencia semántica de que en el orden de unir los controles interactivos y predefinidos, no interrumpidos de los objetos, las acciones internas de estados simples tienen que llevar nombres de diagrama de secuencia en vez de diagramas de estado empotrados; queriendo esto decir que el comportamiento especificado por el diagrama de secuencia se provoca automáticamente cuando se entra al estado correspondiente donde se hace referencia.

Actualmente, OMMMA – L se evalúa en diferentes escenarios, como proyectos industriales para la especificación de servicios de información multimedia, y se investiga características adicionales de sincronía para su especificación en el lenguaje y la formalización de un modelo para la composición dentro y entre los diferentes diagramas de comportamiento. (FERNÁNDEZ and CATALÁ, 2006)

2.4.2.3 Lenguaje para la modelación de Aplicaciones Educativas Multimedia (ApEM – L)

El **Lenguaje para la modelación de Aplicaciones Educativas Multimedia (ApEM - L)** se presenta como una extensión UML que toma como bases teóricas principales los lenguajes de modelado OMMMA – L (2001) y el estándar OCL – 2.0 (2003), lo cual hace que este nuevo lenguaje de modelado tenga 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.

Capítulo 2: Tendencias y Tecnologías actuales

- 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.

La concepción del Lenguaje para la Modelación de Aplicaciones Educativas (ApEM –L) tuvo los siguientes objetivos:

- Desarrollar una extensión del lenguaje de modelado UML, tomándolo como base e incorporando a este, a través de sus mecanismos de extensión, los elementos fundamentales del proceso productivo en la UCI. De esta forma se produjo un lenguaje de propósito particular para la modelación de aplicaciones educativas.
- Incorporar los elementos más significativos de extensiones anteriores como OMMMA – L (2001) y a su vez respetar lo establecido por el estándar OCL (2003), para de esta forma lograr una extensión consistente y escalable en el tiempo.
- No complicar ApEM – L con elementos que lo convirtieran o abarcaran un método de desarrollo de aplicaciones educativas, sino solo el área de la representación y la documentación de este tipo de aplicaciones.
- No circunscribir ApEM – L a un proceso de desarrollo en específico, sino expresarlo de manera tal que pueda ser utilizado con cualquiera de los existentes, aunque se sugiere la utilización de procesos de desarrollo iterativo, incremental y basado en prototipos, que permitan la modelación de sistemas orientados a objetos. (FEBE, 2007)

2.4.2.4 Selección y fundamentación del lenguaje de modelado escogido.

Después de haber estudiado las características de los diferentes lenguajes de modelado mencionados con anterioridad se llegó a la conclusión que el lenguaje de modelado conveniente para esta aplicación era ApEM – L pues este lenguaje se muestra como una extensión del lenguaje base UML que toma como bases teóricas lo planteado por OMMMA – L y el estándar OCL; que trabaja con el patrón arquitectónico Modelo Vista Controlador Entidad (MVC- E) que es una variante del patrón Modelo Vista Controlador (MVC) utilizado por OMMMA – L; que ApEM – L le agrega a este patrón la clase modelo entidad además de utilizar las clases controladoras, vistas y modelos que establece esta patrón y que UML no establece como patrón arquitectónico para regir su concepción de diseño.

2.5 Herramientas para la modelación

Las Herramientas CASE es la mejor base para el proceso de análisis y desarrollo de un software pues estas comprenden el conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores durante todo el ciclo de vida de desarrollo del software (Análisis, Diseño, Implementación e Instalación).

Se iniciaron en la década de los 70 cuando un procesador de palabras fue usado para crear y manipular documentación. La introducción de las herramientas CASE para ayudar en el proceso de análisis y desarrollo ha permitido que los diagramas puedan ser fácilmente creados y modificados, mejorando la calidad de los diseños de software. A continuación se mencionan las características de dos herramientas CASE y las ventajas que tienen cada una de estas al ser utilizadas, para así escoger la más adaptable para esta aplicación.

2.5.1 Rational Rose

Rational Rose es una herramienta de desarrollo basada en modelos que se integra con las bases de datos y los IDE de las principales plataformas del sector. Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática, una dinámica, una lógica y una física. Además permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema del software. (Anexo 6)

Rational Rose utiliza un proceso de desarrollo iterativo controlado, donde este desarrollo se lleva a cabo en una secuencia de iteraciones, donde cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos.

Rational Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. Además proporciona la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño. (PAVÓN, 2002)

2.5.2 Visual Paradigm

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue; además ayuda a una más rápida construcción de aplicaciones de calidad, con un menor costo y mejor calidad. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Además esta herramienta CASE es multiplataforma, es decir, que permite ser utilizada en cualquier sistema operativo, esto hace que tenga grandes aceptaciones en la actualidad. A continuación se mencionan las características principales de esta herramienta:

- ✓ Presenta un editor de Detalles de Casos de Uso incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- ✓ Permite la Ingeniería inversa de bases de datos desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- ✓ Genera los informes para la generación de la documentación.
- ✓ Distribuye automáticamente los diagramas y reorganiza las figuras y conectores de los diagramas del Lenguaje Unificado de Modelado (UML). (CABALLERO, 2007)

2.5.3 Fundamentación de la herramienta escogida.

Después de analizar las herramientas mencionadas anteriormente se llega a la conclusión que la más adaptable es Visual Paradigm pues esta herramienta es multiplataforma que puede ser utilizada en cualquier sistema operativo; además de que puede soportar el ciclo de vida completo de un software, permitiendo construir aplicaciones de calidad a un menor costo y genera informes para la generación de la documentación.

2.6 Conclusiones

En este capítulo que recién ha concluido se precisaron las teorías que en el ámbito tecnológico e informático respaldan el sistema en cuestión abordando claramente las bases teóricas fundamentales de la metodología de desarrollo escogida, RUP por ser un proceso configurable para satisfacer necesidades específicas e implementar las mejores prácticas de desarrollo de software, del lenguaje de modelado ApEM – L por incorporar los elementos más significativos establecidos por OMMMA - L y respetar lo establecido por el lenguaje base UML y la herramienta Visual Paradigm por ser multiplataforma y permitir modelar más rápido la construcción de una aplicación y a un menor costo.

CAPÍTULO 3

Descripción de la solución propuesta

3.1 Introducción

Desde el surgimiento mismo de la computación y a lo largo de toda su evolución se ha intentado modelar o simular el pensamiento humano y los procesos que ocurren en él. En los inicios solo se trataba de representar en las computadoras el pensamiento estructurado, los algoritmos de cálculos que podían definirse claramente como un conjunto de pasos que podían ser interpretados por las máquinas y de cierta forma sustituir o contribuir a elevar la eficiencia del ser humano en este tipo de actividades.

En el siguiente capítulo se realiza una descripción del entorno donde se usará la aplicación como respuesta al problema planteado. Además se hace referencia al Modelo de Dominio y la identificación de los principales conceptos que lo integran; así como el diagrama de estructura de navegación que le brindará una idea al usuario de cómo interactuar con la aplicación y se enumeran los requisitos funcionales y los no funcionales por los cuales se regirá el sistema propuesto.

3.2 Especificación del contenido.

El primer paso en el proceso de desarrollo de software es precisamente alcanzar cierto nivel de conocimientos sobre el problema el cual se desea modelar. En el problema en cuestión el contenido se encuentra estructurado de la siguiente manera:

En la primera interfaz de comunicación con el usuario la cual lleva el nombre de Resultados de los Estudiantes, se muestra el listado de estudiantes (escuela, grado y grupo) seleccionado por el usuario. Esta interfaz nos permite acceder después de haber seleccionado un estudiante determinado a la interfaz de comunicación Visor de Traza que es la interfaz que va a contener un menú de categorías las cuáles se registran en el módulo. Estas categorías son:

- **Identificación _ Estudiante:** Muestra en nombre(s) y apellidos del estudiante del cual se desea visualizar los resultados.

Capítulo 3: Descripción de la solución propuesta

- Datos _ Generales: Muestra los datos generales del estudiante seleccionado (Nombre(s), Apellidos, Grado, Grupo, Sección, Escuela, Fecha, Hora de entrada al software, Hora de salida del software y equipo de trabajo).
- Itinerario: Muestra el resumen del recorrido realizado por el estudiante en el software.
- Temas: Muestra la interacción del estudiante con el Módulo Temas.
- Ejercicios: Muestra la interacción del estudiante con el Módulo Ejercicios.
- Juegos: Muestra la interacción del estudiante con el Módulo Juegos.
- Galería _ Imágenes: Muestra la interacción del estudiante con la Galería de Imágenes del software.
- Galería _ Videos: Muestra la interacción del estudiante con la Galería de Videos del software.
- Información _ Interés: Muestra la interacción del estudiante con la información que se encuentra en la Mediateca del software: Curiosidades, Sabías que, entre otras.
- Glosario: Muestra la interacción del estudiante con las palabras del glosario.

3.3 Modelo de Dominio.

Un modelo conceptual o modelo de dominio explica (a sus creadores) los conceptos significativos en el dominio del problema y constituyen la esencia del análisis orientado a objetos según Larman en el artículo "Modelo Conceptual" en el año 2000. Además el modelo de dominio se realiza cuando existe poco conocimiento y poca estructuración en los procesos de negocio que plantea la metodología RUP.

La identificación de conceptos forma parte de una investigación del dominio del problema. Un modelo conceptual puede mostrar conceptos, asociaciones y atributos de los conceptos; además descomponen el espacio del problema en unidades comprensibles y contribuyen a esclarecer la terminología de dominio. A continuación se muestra el diagrama de clases del modelo de dominio del sistema en cuestión y los conceptos que a él se integran. (Ilustración 1)

Capítulo 3: Descripción de la solución propuesta

3.3.1 Diagrama de clases del modelo de dominio.

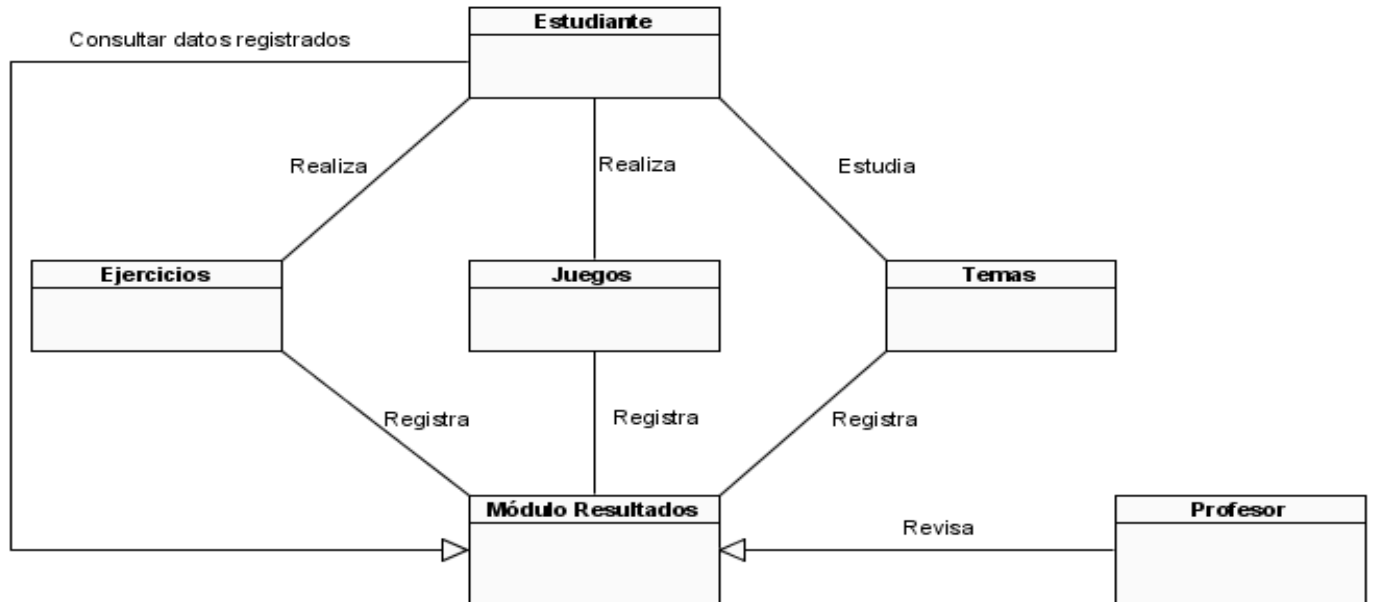


Ilustración 1 Diagrama de Modelo de Dominio

3.3.2 Análisis de los conceptos de dominio.

A continuación se presentan los principales conceptos utilizados en el diagrama del modelo de dominio:

Estudiante: Hace referencia al objeto que interactúa con la aplicación y realiza los ejercicios y juegos que existen en ella y estudia todos los temas pertenecientes a dicha aplicación.

Profesor: Hace referencia al objeto que revisa todos los datos (ejercicios, juegos, temas) consultados y realizados por el estudiante registrados en el Módulo Resultados.

Módulo Resultados: Se le denomina Módulo Resultados al objeto que registra toda la trayectoria del estudiante durante su interacción con la aplicación.

Temas: Es el objeto que contiene el contenido de todos los temas que el estudiante debe estudiar acerca de un tema determinado.

Capítulo 3: Descripción de la solución propuesta

Ejercicios: Es el objeto que contiene todos los ejercicios que el estudiante debe realizar acerca de un tema determinado.

Juegos: Es el objeto que contiene todos los juegos que el estudiante debe realizar para profundizar algún contenido indicado por el profesor.

3.4 Descripción del sistema propuesto.

3.4.1 Descripción de la funcionalidad.

Un proyecto no puede ser exitoso sin una descripción detallada, correcta y exhaustiva de los requerimientos, pues estos definen las funciones que debe de realizar el sistema y la forma en que debe hacerlo.

Un **requisito (o requerimiento)** es una característica de diseño, una propiedad o un comportamiento de un sistema. Los requerimientos constituyen cualidades o capacidades que son especificadas por el cliente y que debe de cumplir el sistema. Estos requerimientos se pueden clasificar en:

- Requerimientos funcionales: representan las capacidades o funciones que el sistema será capaz de realizar.
- Requerimientos no funcionales: representan las características o propiedades que debe presentar el sistema y de alguna forma lo limitan.

3.4.1.1 Requisitos funcionales del sistema.

R1. Permitir al usuario ejecutar la aplicación.

R2. Mostrar la visualización de los módulos.

R3. Permitir al usuario autenticarse.

R4. Permitir al usuario acceder al módulo escogido por éste, en este caso al Módulo Resultados.

R5. Permitir seleccionar el grado y grupo deseado por el usuario.

R6. Mostrar el listado de estudiantes del grado y grupo seleccionado.

R7. Permitir identificar el estudiante del cual se desea visualizar los resultados.

Capítulo 3: Descripción de la solución propuesta

R8. Mostrar un menú de categorías donde se muestre la información relacionada con los resultados obtenidos por el estudiante seleccionado.

R9. Permitir visualizar los resultados obtenidos por el estudiante seleccionado, en una categoría determinada.

R10. Permitir seleccionar la fecha y hora de la sección de trabajo de la que se desea visualizar los resultados del estudiante.

R11. Permitir al profesor insertar o eliminar un estudiante en determinado grupo.

R12. Permitir al usuario conectar o desconectar el audio del sistema.

R13. Permitir la navegación por toda la aplicación.

R14. Permitir la salida del sistema cuando sea solicitada

3.4.1.2 Requerimientos no funcionales.

Apariencia o Interfaz externa

1. En la interfaz de presentación se mostrará el nombre del producto en este caso: "Colección Multisaber Versión Multiplataforma".
2. Las opciones de servicios además de tener su icono identificador tendrá el texto que muestre la opción en cuestión con su función descrita para un reconocimiento rápido por el usuario.
3. Los módulos de la colección siempre estarán visibles y podrán ser accedidos por el usuario desde cualquier interfaz de comunicación; además se mostrarán con el logotipo de un libro.
4. Las medias a visualizar siempre se utilizará en la misma área de la interfaz.
5. Se utilizará un vocabulario reconocido por los estudiantes venezolanos.
6. Los textos que identifican las interfaces u otros contenidos se visualizarán de color predominante para que sea fácil su identificación.

Navegación

7. Desde cualquier interfaz se podrá acceder a cualquier módulo del software.
8. Se podrá salir del programa desde cualquier interfaz, posterior a una confirmación del usuario.

Capítulo 3: Descripción de la solución propuesta

Software y Sistema Operativos

9. Se utilizará el sistema operativo Linux Ubuntu 7.0
10. Se necesitará para lograr ejecutar la aplicación tener instalado Mozilla Firefox.

Hardware

11. Las máquinas donde se ejecutará la aplicación deben de tener los siguientes elementos de hardware mencionados posteriormente:
 - PENTIUM 233 MHz (Recomendado: 500 MHz o mejor).
 - 64 MB RAM (Recomendado: 128 MB RAM o mejor)
 - 1 GB de espacio de Disco Duro.
 - Lector de CD.
 - Dispositivo de audio.
 - Soporte de video que admita resolución de pantalla de al menos 800x600 y 24 bits.
 - Dispositivo de red de al menos 10 megabits.

Restricciones del diseño y la implementación

12. El lenguaje de programación utilizado será JavaScript 2.0.
13. Se utilizará como herramienta para la modelación Visual Paradigm.
14. La herramienta de desarrollo será el CMS Joomla.

Soporte

15. La máquina donde se ejecutará el software deberá disponer de los siguientes requerimientos:
 - Tarjeta de video integrada de 128 Mb (como mínimo).
 - Tarjeta de sonido digital integrada de 128 BIT.

Servicios Generales

16. El audio, inicio, ayuda, busca de contenido y salir del programa siempre estarán visibles al cliente durante toda la navegación que realice por el software.

Capítulo 3: Descripción de la solución propuesta

3.5 Actores del sistema.

Un **actor** es cualquier usuario que interactúe con el sistema. En esta aplicación los actores que intervienen es el usuario, que el mismo puede ser estudiante, maestro o algún invitado que desee conocer como funciona el producto. (Ilustración 2)

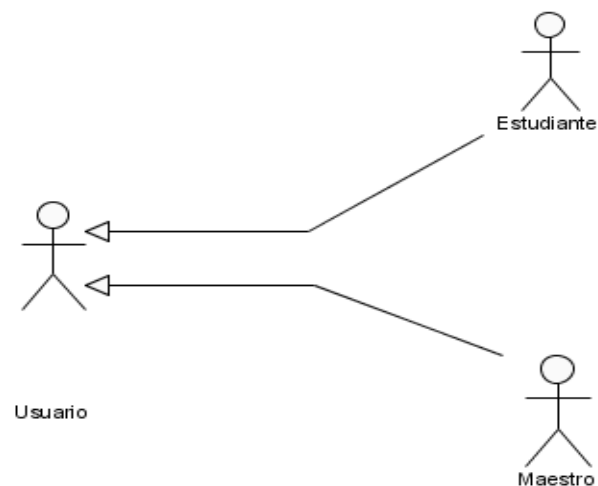


Ilustración 2 Actores del Sistema

3.5.1 Justificación del actor del sistema.

En el sistema en cuestión se representa al usuario como la persona que va a utilizar el software en este caso el estudiante es el usuario que va a interactuar con cada uno de los módulos presentes en el software (Temas, Ejercicios, Juegos y Mediateca) y al profesor como el usuario que evalúa o chequea los contenidos visitados por el estudiante y los ejercicios y juegos realizados por este.

3.6 Área de Gestión de Modelo del sistema.

El área de Gestión de Modelo según ApEM – L ha sufrido numerosos cambios pues se han incorporado nuevos estereotipos restrictivos en todos los diagramas a partir de los nuevos conceptos incorporados a los diagramas de clases originales o básicos de UML. Esta área se encuentra conformada por el diagrama de estructura de navegación, el diagrama de estructura de presentación y por los subsistemas y sus relaciones como se muestra a continuación.

3.6.1 Subsistemas y sus relaciones.

En todas las aplicaciones, salvo en las más pequeñas, el primer paso para diseñar un sistema consiste

Capítulo 3: Descripción de la solución propuesta

en dividir el mismo en pequeños componentes, donde a cada uno de estos componentes principales se les denominará **subsistema**. Un subsistema no es más que un subconjunto del sistema que está compuesto por muchas vistas de presentaciones y éstas a su vez también representan una vista. Cada subsistema abarca aspectos del sistema que comparten alguna propiedad común. A continuación se muestra los diferentes subsistemas que conforman el sistema en cuestión y las relaciones que existen entre los mismos, además de las vistas que se encuentran asociadas a cada uno de estos subsistemas. Los subsistemas son: (Ilustración 3)

- Subsistema Aplicación: Se encuentra relacionado con el Subsistema Básico Módulos, donde dentro del mismo se encuentran todos los módulos que presenta el producto y con el subsistema Paquete Común donde se encuentran todas las opciones de servicio que muestra el software.
- Subsistema Básico Módulos: Se encuentra relacionado con el Subsistema Informativo Resultados, donde dentro del mismo se encuentra un resumen de los resultados obtenidos por el estudiante durante su interacción con el software y con el Subsistema Paquete Común. El Subsistema Básico Módulos está conformado por 4 vistas. (Cargar _ Presentación, Inicio, Ventana _ Auxiliar y Mascota.)
- Subsistema Informativo Resultados: Se encuentra relacionado con el Subsistema Paquete Común y con el Subsistema Menú de Categorías, donde dentro del mismo se encuentran todas las categorías relacionadas con los resultados de los estudiantes durante la interacción con el software. Este subsistema está conformado por 3 vistas (Resultados, Listado _ Estudiantes, Grado y Grupo.)
- Subsistema Menú de Categorías: Se encuentra relacionado con el Subsistema Paquete Común, con el Subsistema Opciones y con el Subsistema Gestionar Estudiante. Este Subsistema está compuesto por 11 vistas, las cuales se mencionan a continuación: (Identificación _ Estudiante, Datos _ Generales, Itinerario, Temas, Ejercicios, Juegos, Galería _ Imágenes, Galería _ Videos, Información _ Interés, Glosario y Sesión.)
- Subsistema Gestionar Estudiante: Este subsistema está conformado la vista Insertar _Estudiante y por la vista Eliminar_ Estudiante.

Capítulo 3: Descripción de la solución propuesta

- Subsistema Opciones: Está conformado por el subsistema por la vista Pantalla _ Principal.
- Subsistema Paquete Común: Está conformado por 6 vistas: Audio, Ayuda, Salir, Anterior, Acerca de... y de la vista Busca _ Contenido.

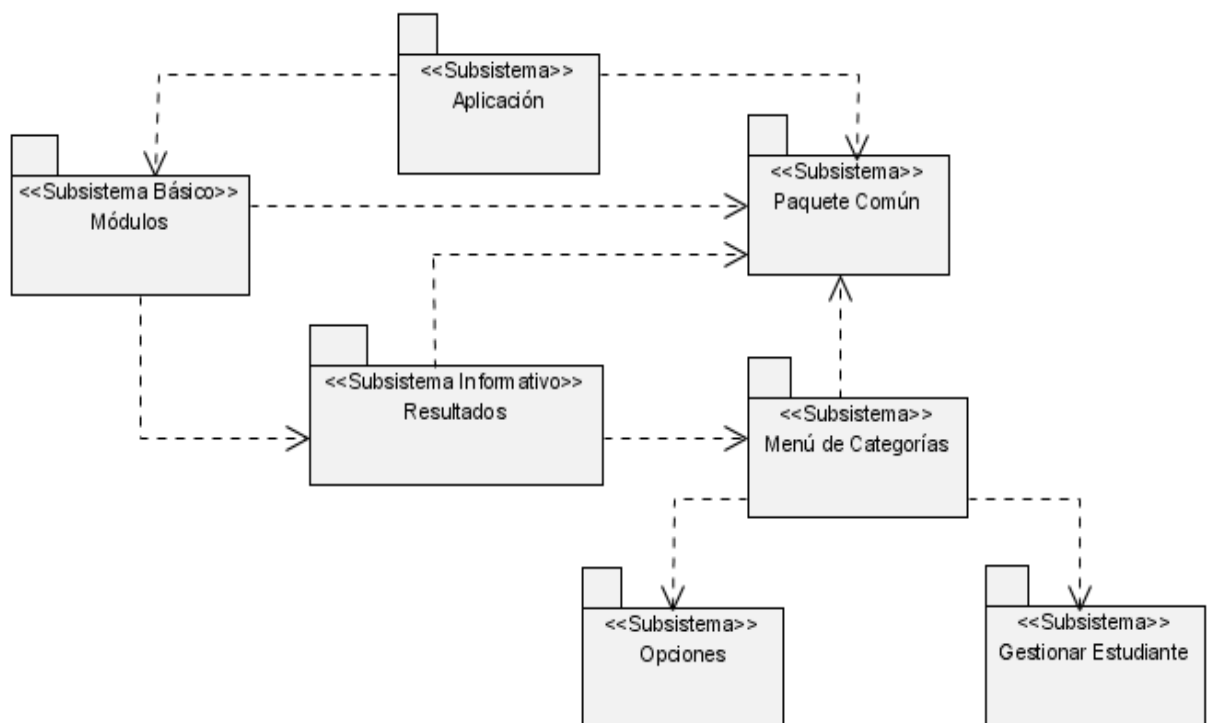


Ilustración 3 Subsistemas y sus relaciones

3.6.2 Diagrama de Estructura de Navegación.

El **diagrama de estructura de navegación** utilizando el lenguaje de modelado ApEM – L utiliza para la modelación *la clase menú*, *clase índice*, *clase botón* y *la clase consulta*; donde cada una dentro del diagrama realizan funciones diferentes, permitiendo que la navegación en el sistema sea más comprensible por el usuario. La **clase menú** permite llegar a diferentes clases vistas con las cuales se conecta el menú, la **clase índice** permite denotar el valor de direccionamiento hacia una clase vista a partir de una opción determinada, la **clase botón** permite denotar la existencia de un elemento interactivo del tipo botón para producir un camino en la navegación hacia la clase vista y la **clase**

Capítulo 3: Descripción de la solución propuesta

consulta permite el enlace a través de un valor directo (variable de consulta) asignando a la búsqueda el elemento que se desea visualizar.

A continuación se muestra el diagrama de estructura de navegación del sistema donde primeramente el usuario ejecuta la aplicación, luego se muestra la clase vista Presentación la cual está compuesta por la clase botón Módulos, donde se encuentran todos los módulos del software. Esta clase botón está relacionada con una clase índice que es la que permite el acceso a la clase vista Módulos, donde de ésta se accede a la clase vista Ventana _ Auxiliar para que el usuario se identifique utilizando la clase botón; si éste no desea identificarse oprime el botón cancelar y no puede acceder a realizar ninguna otra operación, sino se identifica oprime el botón aceptar y accede a consultar mediante la clase consulta el grado y grupo deseado por él y se muestra el listado de estudiantes del grado y grupo seleccionado por el usuario.

Esta clase listado de estudiantes se encuentra compuesta por una clase menú Categorías donde se encuentran todas las categorías que se desean saber del estudiante; pero para consultar estas categorías, se necesita acceder a la clase índice Categorías que es la que me va a permitir el acceso a la vista Categorías y así a cada una de las categorías deseadas por el usuario. Todas estas clases tienen acceso al Subsistema Paquete Común, que en él se encuentran todas las opciones de servicio que muestra el software como: Ayuda, Salir del sistema, Buscar Contenido Deseado y retroceder a la página anterior. (Ilustración 4)

Capítulo 3: Descripción de la solución propuesta

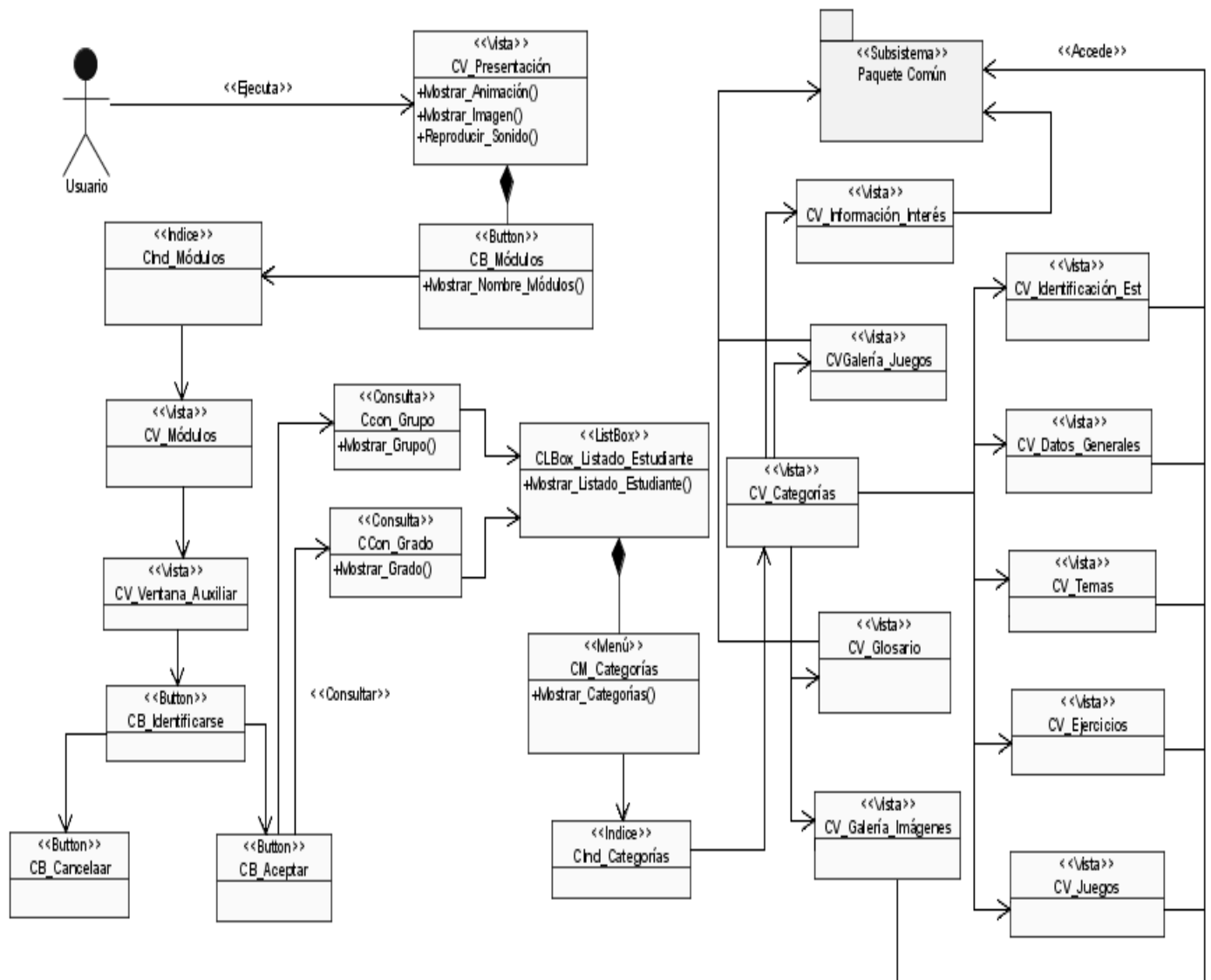


Ilustración 4 Diagrama de Estructura de Navegación del sistema.

3.6.3 Diagrama de Estructura de Presentación.

Un aspecto esencial a la hora de modelar un software es la definición que tendrá sus futuras interfaces de comunicación con el usuario. Según el lenguaje de modelado ApEM – L un **diagrama de estructura de presentación (DEP)** utilizando este lenguaje está compuesto por **clases estáticas** y con **clases interactivas**; donde las clases estáticas agrupan los componentes que tienen que ver con la visualización de la información (imágenes, texto, sonido ó animación) y las clases interactivas son los elementos que permiten la interacción del usuario con el sistema modelado.

Capítulo 3: Descripción de la solución propuesta

A continuación se muestra los diagramas de estructura de presentación de las vistas que tienen prioridad crítica en el sistema y la descripción textual de cada una de estas vistas.

El diagrama de estructura de presentación que se muestra a continuación (Ilustración 5) representa el DEP de la vista Listado de Estudiantes del subsistema Resultados, el cual se encuentra relacionado con los subsistemas Módulos, Paquete Común y Opciones. Este diagrama se encuentra formado según define el lenguaje de modelado ApEM – L por las vistas estáticas (CEst _ Listado _ Estudiante), la cual está asociada con la clase modelo entidad media (CMEM _ Imagen _ Fondo); que agrupa las clases que identifican las medias, en este caso la imagen y todos los atributos y métodos correspondientes a la misma y por la clase interactiva (CInt _ Listado _ estudiante) que es la que me va a permitir la interacción con el listado de estudiante; la cual según el lenguaje de alto nivel (HLL) JavaScript permitirá mostrar el nombre y los apellidos de los estudiantes.

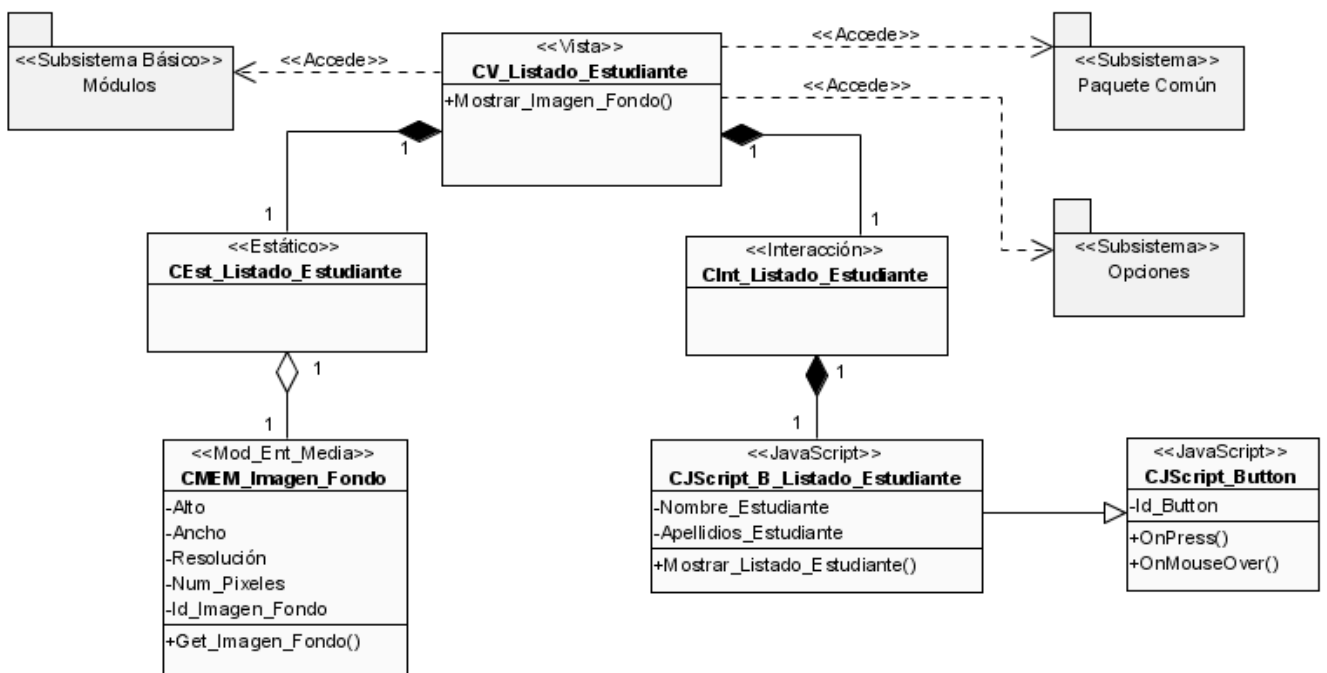


Ilustración 5 Diagrama de estructura de presentación de vista Listado _ Estudiante

Capítulo 3: Descripción de la solución propuesta

Tabla 1 Descripción Textual de Vista de Presentación Listado _ Estudiante

Descripción Textual de la Vista de Presentación: Listado _ Estudiante			
Actores de la Vista de Presentación		Usuario (profesor o estudiante)	
Propósito		Mostrar al usuario de la aplicación el listado de estudiantes que ha interactuado con la aplicación.	
Objetivos pedagógicos		-	
Resumen		La vista se inicia cuando el usuario accede a la aplicación, para ver el listado de estudiantes que ha interactuado con la misma y termina cuando se muestran dichos datos.	
Vistas asociadas		Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.	
Referencias		R1, R2, R3, R4, R5, R6	
Precondiciones		-	
Poscondiciones		Vistas de presentación visualizada de acuerdo a la elección del usuario.	
Curso Normal de los Eventos			
Acciones del Actor		Respuesta del Sistema	Elementos de la Vista
1. El usuario selecciona la opción de trabajo Listado de estudiantes. 3. El usuario selecciona en el listado el nombre y apellido del estudiante deseado y culmina la vista.		2. Muestra el Listado de estudiantes.	2.1 Listado de estudiantes.
Prioridad		Crítica	
Mejoras		-	
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo	Existente.
Video o Animación	-		
Sonido	-		
Texto			

Capítulo 3: Descripción de la solución propuesta

Después de haber observado el DEP de la vista de presentación anterior se mostrará el DEP de la vista de presentación Filtros la cual permitirá mostrar el grado y grupo de un estudiante determinado. Este diagrama se encuentra formado (Ilustración 6) por la clase vista estática (CEst _ Filtros), la cual está asociada con la clase modelo entidad media (CMEM _ Imagen _ Fondo) y todos los atributos y métodos correspondientes a la misma y por la clase interactiva (CInt _ Filtros) que es la que va a permitir la mostrar los datos deseados.

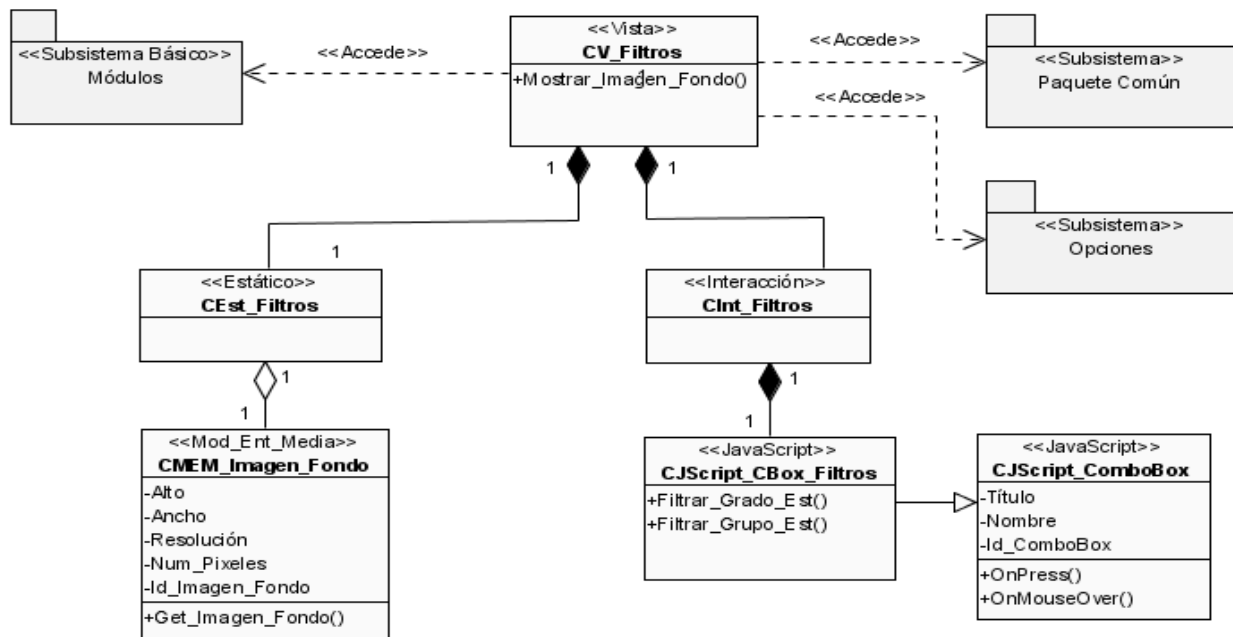


Ilustración 6 Diagrama de estructura de presentación de vista Filtros

Tabla 2 Descripción Textual de Vista de Presentación Filtros

Descripción Textual de la Vista de Presentación: Filtros	
Actores de la Vista de Presentación	Usuario (profesor o estudiante)
Propósito	Filtrar el grado y grupo de los estudiantes para poder acceder al listado de estudiantes del grupo y grupo en cuestión.
Objetivos pedagógicos	-
Resumen	La vista se inicia cuando el usuario selecciona el grado y grupo deseado para ver el listado de estudiante del mismo y termina cuando se filtra dichos eventos.
Vistas asociadas	Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.

Capítulo 3: Descripción de la solución propuesta

Referencias		R1, R2, R3, R4, R5.	
Precondiciones			
Poscondiciones		Vistas de presentación visualizada de acuerdo a la elección del usuario.	
Curso Normal de los Eventos			
Acciones del Actor	Respuesta del Sistema	Elementos de la Vista	
1. El usuario selecciona el grado y grupo deseado.	2. Filtra el grado y grupo deseado por el usuario. 3. Muestra grado y grupo deseado por el mismo y así culmina la vista.	2.1 Vista Grado y Vista Grupo.	
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de Fondo	Existente
Video o Animación	-		
Sonido	-		
Texto	-		

El diagrama de presentación que se muestra (Ilustración 7) representa el DEP de vista Identificación de Estudiante la cual va a permitir identificar al estudiante del cual se desea visualizar los resultados; la misma está compuesta por la clase estática (CEst _ Identificación _ Estudiante) la cual se encuentra asociada a la clase (CMEM _ Imagen _ Fondo) que es la que va a visualizar la media imagen y por la clase interactiva (CInt _ Identificación _ Estudiante) que es la que mostrar a través del lenguaje JavaScript el estudiante identificado por el usuario.

Capítulo 3: Descripción de la solución propuesta

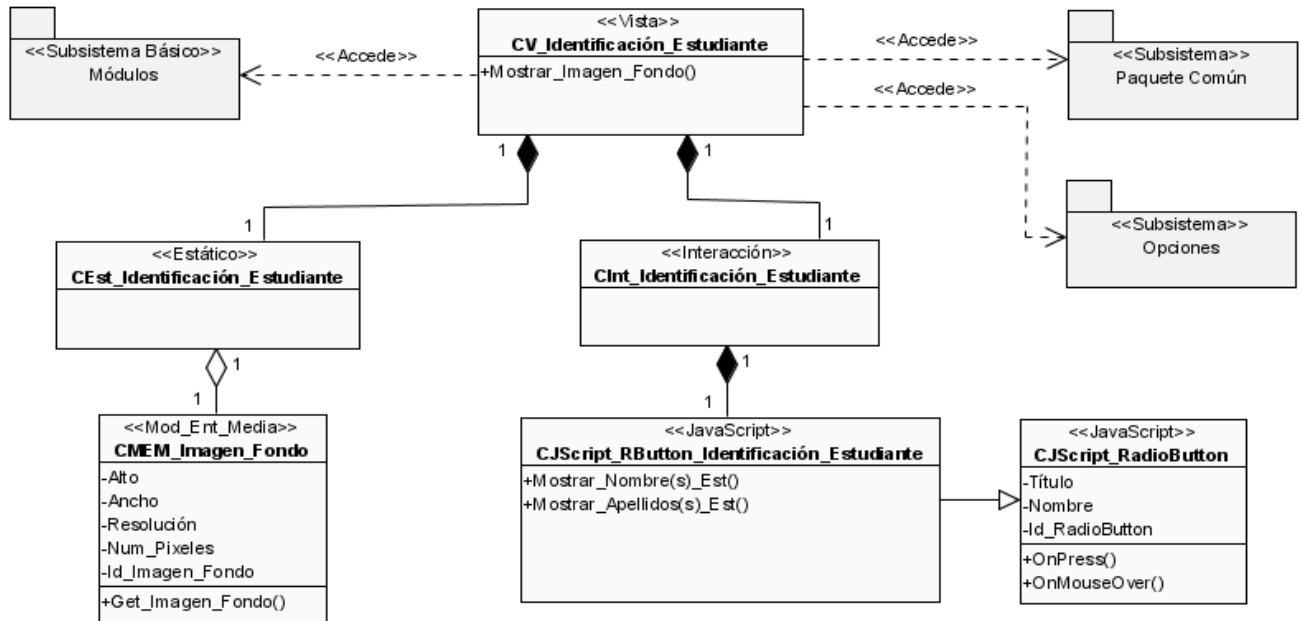


Ilustración 7 Diagrama de estructura de presentación de Vista Identificación _ Estudiante

Tabla 3 Descripción Textual de Vista de Presentación Identificación _ Estudiante

Descripción Textual de la Vista de Presentación: Identificación _ Estudiante	
Actores de la Vista de Presentación	Usuario (profesor o estudiante)
Propósito	Mostrar al usuario la identificación del estudiante deseado por el mismo.
Objetivos pedagógicos	-
Resumen	La vista se inicia cuando el usuario selecciona la opción de trabajo identificación y se muestra la identificación del estudiante deseado por este y culmina cuando se muestre la identificación de dicho estudiante.
Vistas asociadas	Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.
Referencias	R1, R2, R3, R4, R5, R6, R7.
Precondiciones	-
Poscondiciones	Vistas de presentación visualizada de acuerdo a la elección del usuario.
Curso Normal de los Eventos	

Capítulo 3: Descripción de la solución propuesta

Acciones del Actor		Respuesta del Sistema	Elementos de la Vista
1. El usuario selecciona la opción de trabajo Identificación para ver la identificación del estudiante.		2. Busca la identificación deseada por el usuario. 3. Muestra la Identificación y así culmina la vista.	2.1 Vista Identificación.
Prioridad		Crítica	
Mejoras		-	
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de Fondo	Existente
Video o Animación	-		
Sonido	-		
Texto	-		

El diagrama de presentación que se muestra (Ilustración 8) representa el DEP de vista Datos_Generales la cual va a mostrar nombre, apellidos, sesión, fecha, hora de entrada, hora de salida y equipo de trabajo de determinado estudiante. El mismo se encuentra compuesto por la clase estática (CEst _ Datos _ Generales) la cual se encuentra asociada a la clase (CMEM _ Imagen _ Fondo) que es la que va a visualizar la media imagen y por la clase interactiva (CInt _ Datos _ Generales) que es la que mostrar a través del lenguaje JavaScript dichos datos.

Capítulo 3: Descripción de la solución propuesta

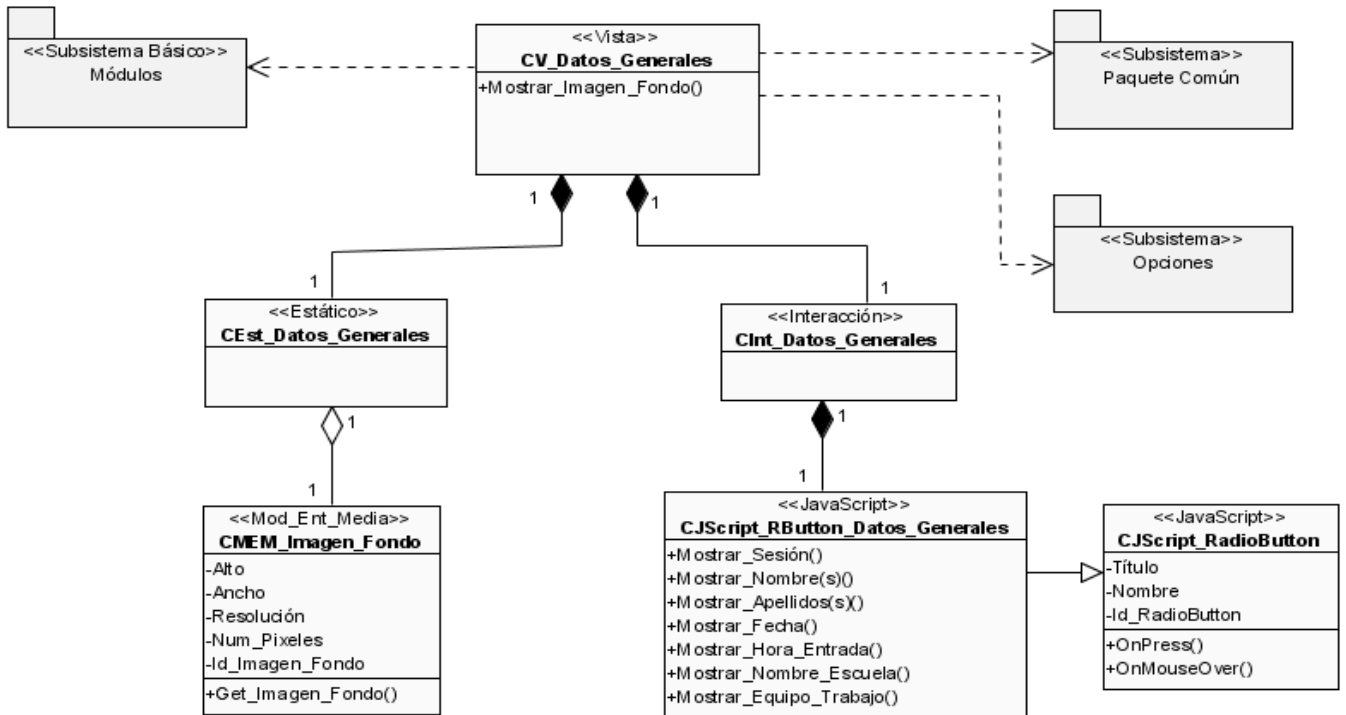


Ilustración 8 Diagrama de estructura de presentación de vista Datos _ Generales

Tabla 4 Descripción Textual de Vista de Presentación Datos _ Generales

Descripción Textual de la Vista de Presentación: Datos _ Generales		
Actores de la Vista de Presentación	Usuario (profesor o estudiante)	
Propósito	Mostrar los datos generales del estudiante seleccionado por el usuario.	
Objetivos pedagógicos	-	
Resumen	La vista se inicia cuando el usuario selecciona la opción de trabajo Datos generales y se muestra los datos del estudiante deseado y culmina así dicha vista.	
Vistas asociadas	Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.	
Referencias	R1, R2, R3, R4, R5, R7, R8.	
Precondiciones		
Poscondiciones	Vistas de presentación visualizada de acuerdo a la elección del usuario.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	Elementos de la Vista

Capítulo 3: Descripción de la solución propuesta

1. El usuario selecciona la opción de trabajo Datos generales para ver los datos del estudiante deseado por él.	2. Busca los datos deseados por el usuario. 3. Muestra los datos generales del estudiante y así culmina la vista.	2.1 Vista Datos Generales.	
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo	Existente
Video o Animación	-		
Sonido	-		
Texto	-		

Después de haber observado el DEP de la vista de presentación anterior se mostrará el DEP de la vista de presentación Itinerario (Ilustración 9) la cual permitirá mostrar el recorrido del estudiante durante su interacción con el software. Este diagrama se encuentra formado por la clase vista estática (CEst _ Itinerario), la cual está asociada con la clase modelo entidad media (CMEM _ Imagen _ Fondo) y todos los atributos y métodos correspondientes a la misma y por la clase interactiva (CInt _ Itinerario) que es la que va a permitir la mostrar el recorrido de dicho estudiante.

Capítulo 3: Descripción de la solución propuesta

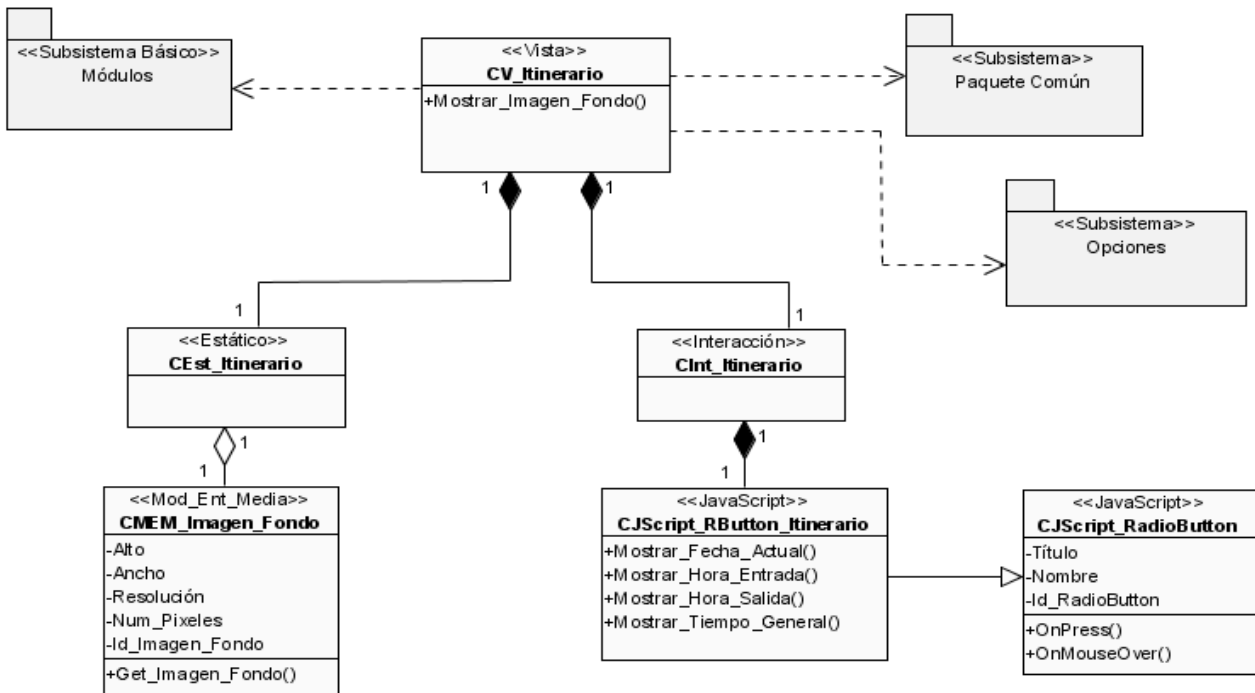


Ilustración 9 Diagrama de estructura de presentación de vista Itinerario

Tabla 5 Descripción Textual de Vista de Presentación Itinerario

Descripción Textual de la Vista de Presentación: Itinerario	
Actores de la Vista de Presentación	Usuario (profesor o estudiante)
Propósito	Mostrar el recorrido realizado por el estudiante por el software.
Objetivos pedagógicos	-
Resumen	La vista se inicia cuando el usuario selecciona la opción de trabajo Itinerario y se culmina cuando se muestra la fecha, hora de entrada, hora de salida y el tiempo de general que el estudiante estuvo interactuando con el software.
Vistas asociadas	Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.
Referencias	R1, R2, R3, R4, R5, R6, R7, R7, R8, R9, R10.
Precondiciones	
Poscondiciones	Vistas de presentación visualizada de acuerdo a la elección del usuario.

Capítulo 3: Descripción de la solución propuesta

Curso Normal de los Eventos			
Acciones del Actor		Respuesta del Sistema	Elementos de la Vista
1. El usuario selecciona la opción de trabajo Itinerario.		2. Busca los elementos pertenecientes a esta vista. 3. Muestra la fecha, hora de entrada, hora de salida y el tiempo que estuvo el estudiante interactuando con el software.	2.1 Vista Itinerario.
Prioridad		Crítica	
Mejoras		-	
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo	Existente
Video o Animación	-		
Sonido	-		
Texto	-		

El diagrama de presentación que se muestra (Ilustración 10) representa el DEP de vista Temas la cual va a mostrar todos los temas visitados por el estudiante. El mismo se encuentra compuesto por la clase estática (CEst _ Temas) la cual se encuentra asociada a la clase (CMEM _ Imagen _ Fondo) y todos los atributos y métodos que a ella respectan y por la clase interactiva (CInt _ Temas) que es la que mostrar a través del lenguaje JavaScript los temas consultados por el estudiante.

Capítulo 3: Descripción de la solución propuesta

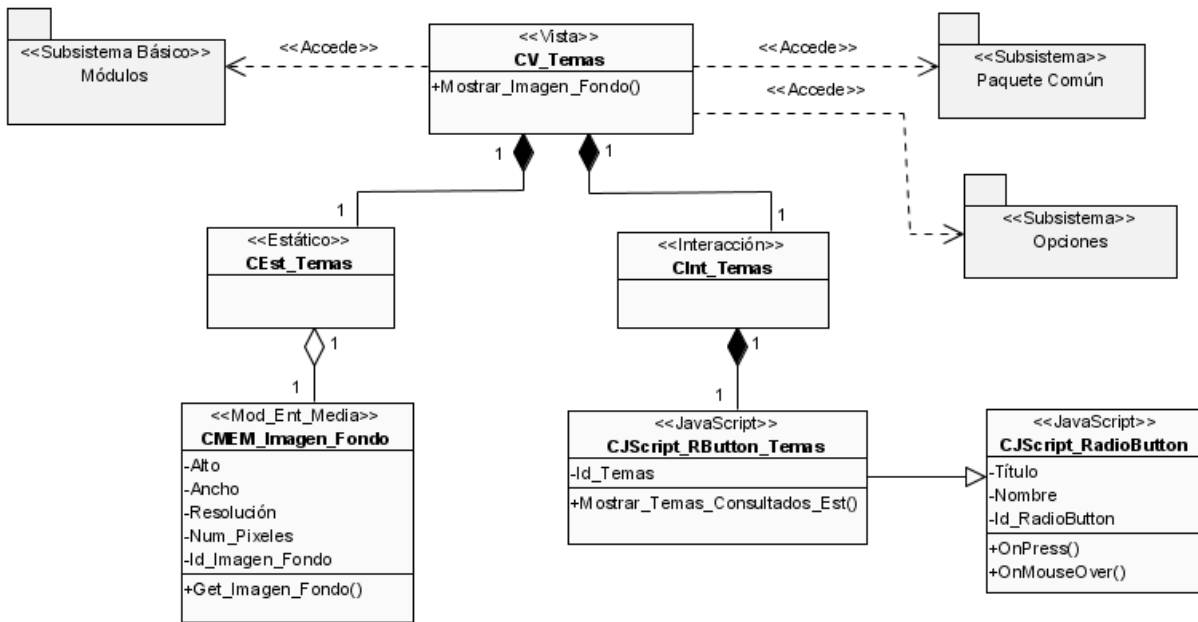


Ilustración 10 Diagrama de estructura de presentación de vista Temas

Tabla 6 Descripción Textual de Vista de Presentación Temas

Descripción Textual de la Vista de Presentación: Temas	
Actores de la Vista de Presentación	Usuario (profesor o estudiante)
Propósito	Mostrar al usuario los temas visitados por el estudiante durante su trayectoria en la aplicación.
Objetivos pedagógicos	Muestra los resultados de la interacción del estudiante con el Módulo Temas.
Resumen	La vista se inicia cuando el usuario selecciona la opción de trabajo Temas para ver la trayectoria del estudiante durante su interacción en la aplicación y se culmina cuando se muestran los temas visitados por el mismo.
Vistas asociadas	Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.
Referencias	R1, R2, R3, R4, R5, R6.
Precondiciones	-
Poscondiciones	Vistas de presentación visualizada de acuerdo a la elección del usuario.

Capítulo 3: Descripción de la solución propuesta

Curso Normal de los Eventos			
Acciones del Actor		Respuesta del Sistema	Elementos de la Vista
1. El usuario selecciona la opción de trabajo Temas para ver los temas visitados por el estudiante.		2. Busca los temas visitados por el estudiante. 3. Muestra los temas visitados por el estudiante y así culmina la vista.	2.1 Vista Temas.
Prioridad		Crítica	
Mejoras		-	
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo	Existente
Video o Animación	-		
Sonido	-		
Texto	-		

En el DEP que a continuación se muestra (Ilustración 11) representa al igual que los diagramas expuestos anteriormente la clase estática (CEst _ Temas) la cual se encuentra asociada a la clase (CMEM _ Imagen _ Fondo) y todos los atributos y métodos que a ella respectan y por la clase interactiva (CInt _ Juegos) que es la que mostrar a través del lenguaje JavaScript los juegos realizados por el estudiante.

Capítulo 3: Descripción de la solución propuesta

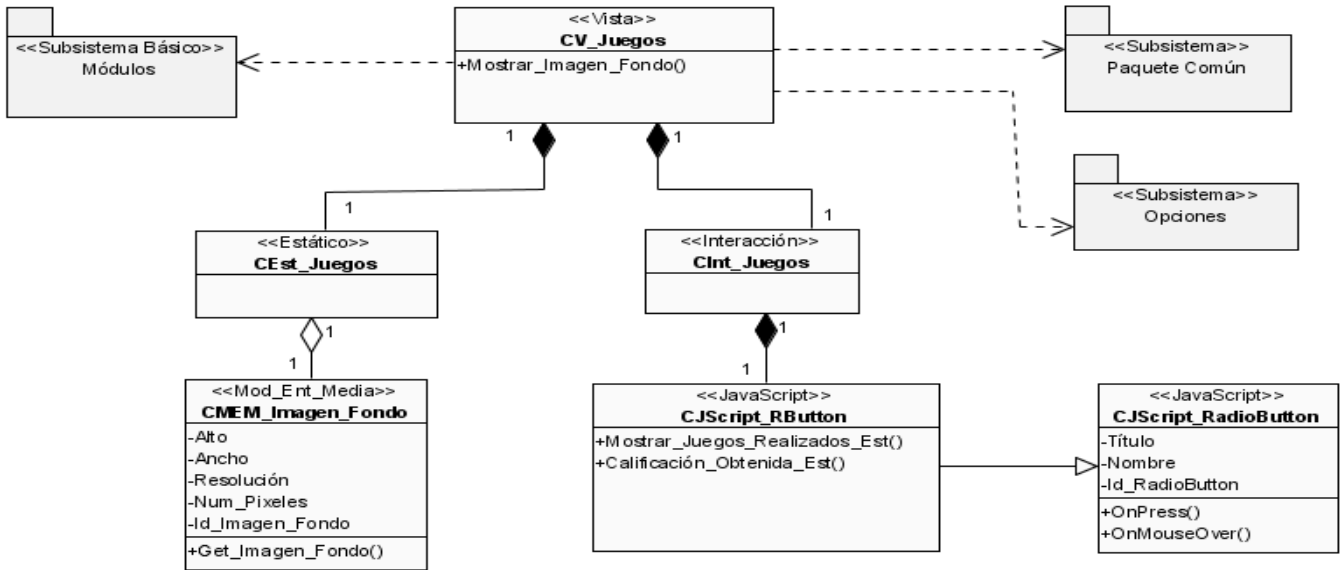


Ilustración 11 Diagrama de estructura de presentación de vista Juegos

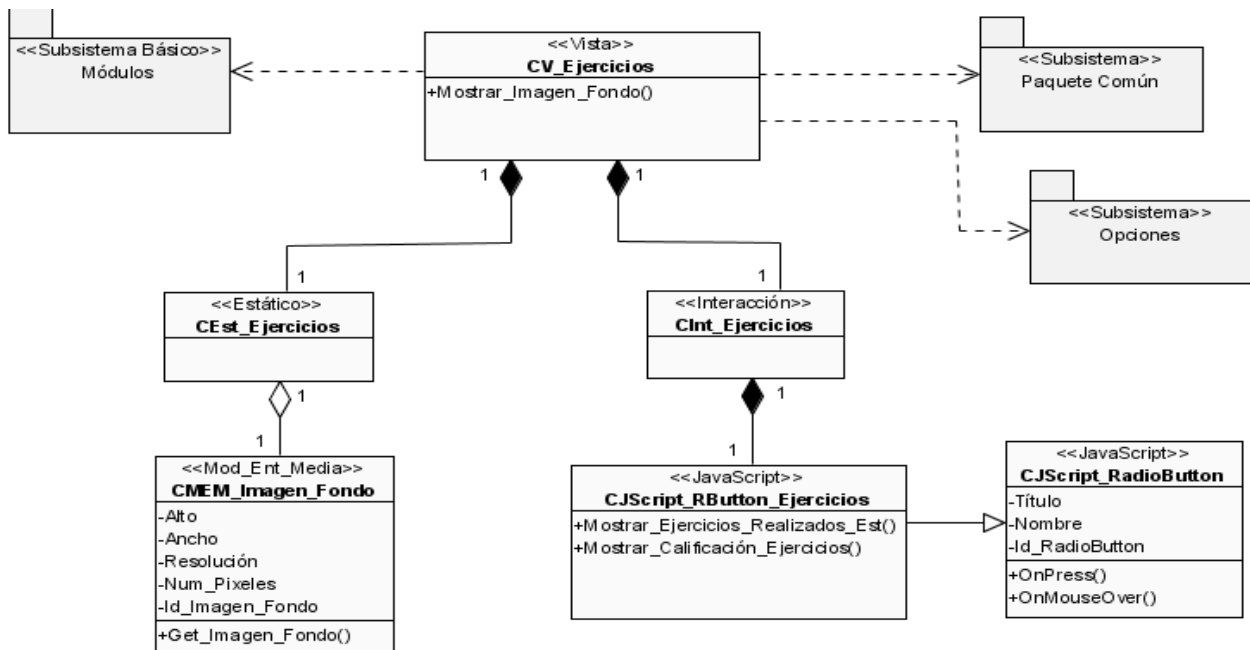
Tabla 7 Descripción Textual de Vista de Presentación Juegos

Descripción Textual de la Vista de Presentación: Juegos	
Actores de la Vista de Presentación	Usuario (profesor o estudiante)
Propósito	Mostrar al usuario los juegos realizados por el estudiante.
Objetivos pedagógicos	Muestra los resultados de la interacción del estudiante con el Módulo Juegos.
Resumen	La vista se inicia cuando el usuario selecciona la opción de trabajo Juegos para ver los juegos realizados por el estudiante y se culmina cuando se muestran los mismos.
Vistas asociadas	Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.
Referencias	R1, R2, R3, R4, R5, R6.
Precondiciones	-
Poscondiciones	Vistas de presentación visualizada de acuerdo a la elección del usuario.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
	Elementos de la Vista

Capítulo 3: Descripción de la solución propuesta

1. El usuario selecciona la opción de trabajo Juegos para ver los juegos realizados por el estudiante.	2. Busca los juegos realizados por el estudiante.	2.1 Vista Juegos.	
	3. Muestra las definiciones, imágenes y videos consultados por el estudiante y así culmina la vista.		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo	Existente
Video o Animación	-		
Sonido	-		
Texto	-		

En el DEP que a continuación se muestra (Ilustración 12) representa al igual que los diagramas expuestos anteriormente la clase estática (CEst _ Ejercicios) la cual se encuentra asociada a la clase (CMEM _ Imagen _ Fondo) y todos los atributos y métodos que a ella respectan y por la clase interactiva (CInt _ Ejercicios) que es la que mostrar a través del lenguaje JavaScript los juegos realizados por el estudiante.



Capítulo 3: Descripción de la solución propuesta

Ilustración 12 Diagrama de estructura de presentación de vista Ejercicios

Tabla 8 Descripción Textual de Vista de Presentación Ejercicios

Descripción Textual de la Vista de Presentación: Ejercicios			
Actores de la Vista de Presentación		Usuario (profesor o estudiante)	
Propósito		Mostrar al usuario los ejercicios realizados y la puntuación obtenida por el estudiante.	
Objetivos pedagógicos		Muestra los resultados de la interacción del estudiante con el Módulo Ejercicios.	
Resumen		La vista se inicia cuando el usuario selecciona la opción de trabajo Ejercicios para ver los ejercicios realizados por el estudiante y la puntuación que alcanzó el mismo y se culmina cuando se muestran los mismos.	
Vistas asociadas		Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.	
Referencias		R1, R2, R3, R4, R5, R6.	
Precondiciones		-	
Poscondiciones		Vistas de presentación visualizada de acuerdo a la elección del usuario.	
Curso Normal de los Eventos			
Acciones del Actor	Respuesta del Sistema		Elementos de la Vista
1. El usuario selecciona la opción de trabajo Ejercicios para ver los ejercicios realizados por el estudiante y la puntuación alcanzada por este.	2. Busca los ejercicios realizados y la puntuación obtenida. 3. Muestra los ejercicios realizados y la puntuación obtenida por el estudiante y así culmina la vista.		2.1 Vista Ejercicios.
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo	Existente

Capítulo 3: Descripción de la solución propuesta

Video o Animación	-		
Sonido	-		
Texto	-		

El diagrama de presentación que se presenta (Ilustración 13) representa el DEP de vista Galería _ Videos la cual va a mostrar los resultados del estudiante en su interacción con la Galería de Videos del software. El mismo se encuentra compuesto por la clase estática (CEst _ Galería _ Videos) la cual se encuentra asociada a la clase (CMEM _ Imagen _ Fondo) que es la que va a visualizar la media imagen y por la clase interactiva (CInt _ Galería _ Videos) que es la que mostrar a través del lenguaje JavaScript los videos consultados.

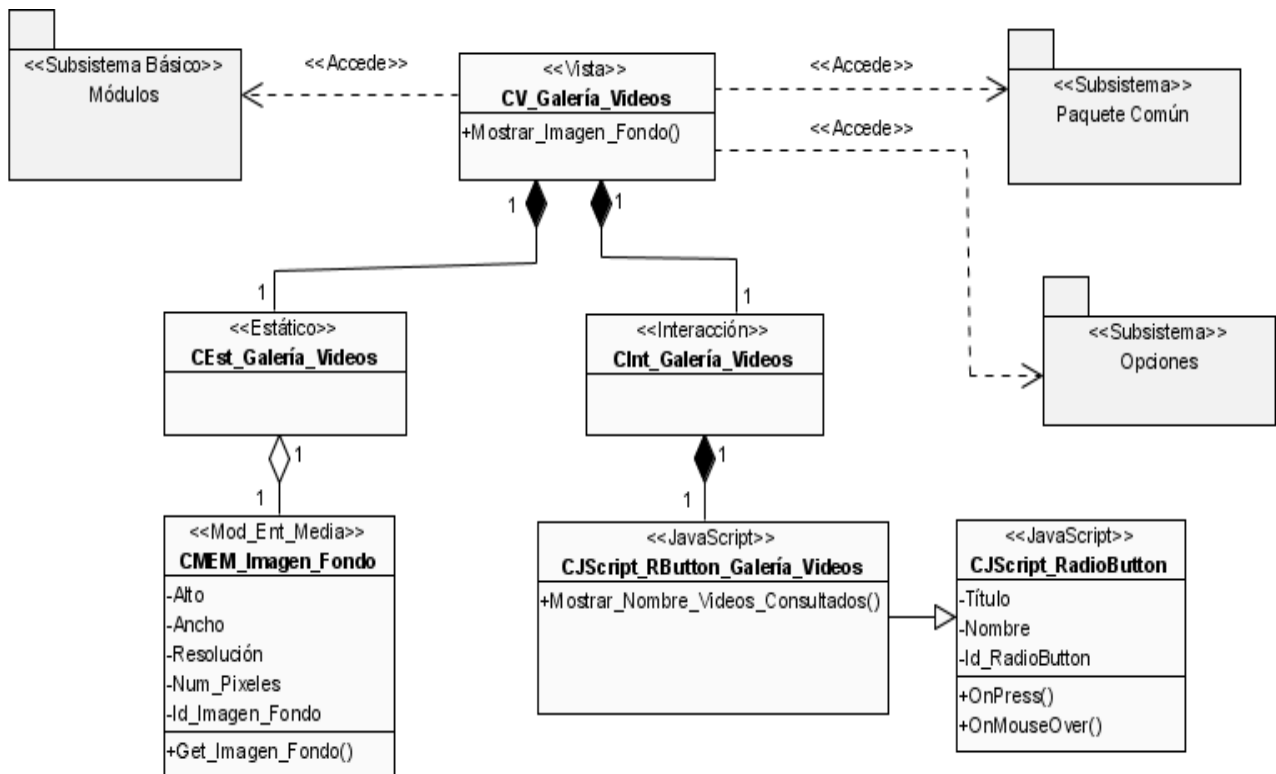


Ilustración 13 Diagrama de estructura de presentación de vista Galería _ Videos

Capítulo 3: Descripción de la solución propuesta

Tabla 9 Descripción Textual de Vista de Presentación Galería _ Videos

Descripción Textual de la Vista de Presentación: Galería _ Videos			
Actores de la Vista de Presentación	Usuario (Profesor o Estudiante)		
Propósito	Mostrar al usuario los videos consultados por el estudiante.		
Objetivos pedagógicos	Muestra los resultados de la interacción del estudiante con la Galería de Videos del software.		
Resumen	La vista se inicia cuando el usuario selecciona la opción de trabajo Galería de Videos y se culmina cuando muestra los videos visitadas por el estudiante.		
Vistas asociadas	Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.		
Referencias	R1, R2, R3, R4, R5, R6.		
Precondiciones	-		
Poscondiciones	Vistas de presentación visualizada de acuerdo a la elección del usuario.		
Curso Normal de los Eventos			
Acciones del Actor	Respuesta del Sistema	Elementos de la Vista	
1. El usuario selecciona la opción Galería _ Videos.	2. Busca los videos visitadas por el estudiante. 3. Muestra las videos consultadas por el estudiante	2.1 Vista Galería _ Videos.	
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo	Existente
Video o Animación	-		
Sonido	-		
Texto	-		

Capítulo 3: Descripción de la solución propuesta

El diagrama de presentación que se presenta (Ilustración 14) representa el DEP de vista Galería _ Imágenes la cual va a mostrar los resultados del estudiante en su interacción con la Galería de Imágenes del software. El mismo se encuentra compuesto por la clase estática (CEst _ Galería _ Imágenes) la cual se encuentra asociada a la clase (CMEM _ Imagen _ Fondo) que es la que va a visualizar la media imagen y por la clase interactiva (CInt _ Galería _ Imágenes) que es la que mostrar a través del lenguaje JavaScript las imágenes consultadas.

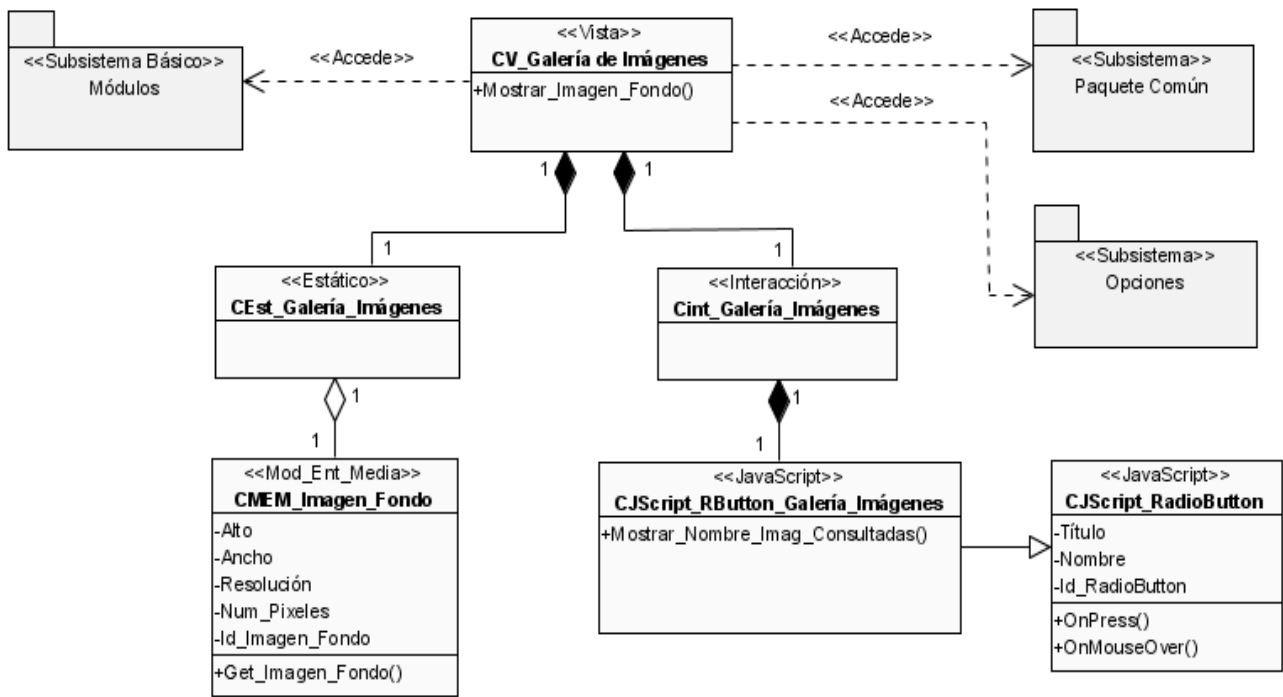


Ilustración 14 Diagrama de estructura de presentación de vista Galería _ Imágenes

Tabla 10 Descripción Textual de Vista de Presentación Galería _ Imágenes

Descripción Textual de la Vista de Presentación: Galería _ Imágenes	
Actores de la Vista de Presentación	Usuario (Profesor o Estudiante)
Propósito	Mostrar al usuario las imágenes visitadas por el estudiante.
Objetivos pedagógicos	Muestra los resultados de la interacción del estudiante con la Galería de Imágenes del software.
Resumen	La vista se inicia cuando el usuario selecciona la opción de trabajo Galería de Imágenes y se culmina cuando muestra las imágenes visitadas por el estudiante.

Capítulo 3: Descripción de la solución propuesta

Vistas asociadas	Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.		
Referencias	R1, R2, R3, R4, R5, R6.		
Precondiciones	-		
Poscondiciones	Vistas de presentación visualizada de acuerdo a la elección del usuario.		
Curso Normal de los Eventos			
Acciones del Actor	Respuesta del Sistema		Elementos de la Vista
1. El usuario selecciona la opción Galería _ Imágenes.	2. Busca las imágenes visitadas por el estudiante. 3. Muestra las imágenes consultadas por el estudiante		2.1 Vista Galería _ Imágenes.
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo	Existente
Video o Animación	-		
Sonido	-		
Texto	-		

El presente DEP corresponde a la vista Glosario (Ilustración 15) la cual va a mostrar los resultados del estudiante durante su interacción con las palabras calientes del glosario de términos del software. El mismo se encuentra compuesto por la clase estática (CEst _ Glosario) la cual se encuentra asociada a la clase (CMEM _ Imagen _ Fondo) que es la que va a visualizar la media imagen y por la clase interactiva (CInt _ Glosario) que es la que mostrar a través del lenguaje JavaScript las palabras calientes visitadas.

Capítulo 3: Descripción de la solución propuesta

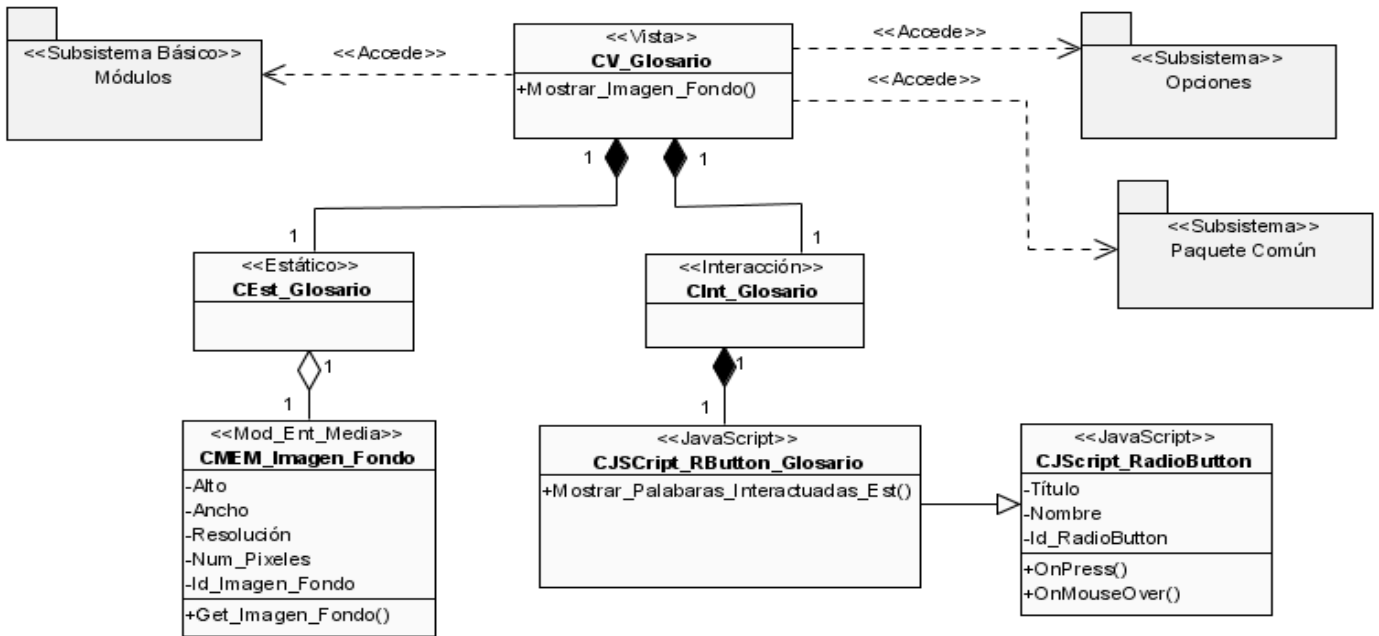


Ilustración 15 Diagrama de estructura de presentación de vista Glosario

Tabla 11 Descripción Textual de Vista de Presentación Glosario

Descripción Textual de la Vista de Presentación: Glosario	
Actores de la Vista de Presentación	Usuario (Profesor o Estudiante)
Propósito	Mostrar al usuario las palabras calientes interactuadas por el estudiante.
Objetivos pedagógicos	Muestra los resultados de la interacción del estudiante con las palabras del glosario.
Resumen	La vista se inicia cuando el usuario selecciona la opción de trabajo Glosario y se culmina cuando se muestra las palabras consultadas por el estudiante.
Vistas asociadas	Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.
Referencias	R1, R2, R3, R4, R5, R6.
Precondiciones	-
Poscondiciones	Vistas de presentación visualizada de acuerdo a la elección del usuario.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema Elementos de la Vista

Capítulo 3: Descripción de la solución propuesta

1. El usuario selecciona la opción Glosario.	2. Busca las palabras calientes visitadas por el estudiante. 3. Muestra las palabras calientes consultadas por el estudiante.	2.1 Vista Glosario.	
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo	Existente
Video o Animación	-		
Sonido	-		
Texto	-		

El presente DEP corresponde a la vista Información _ Interés (Ilustración 16) la cual va a mostrar los resultados del estudiante durante su interacción con la Mediateca del software. El mismo se encuentra compuesto por la clase estática (CEst _ Información _ Interés) la cual se encuentra asociada a la clase (CMEM _ Imagen _ Fondo) que es la que va a visualizar la media imagen y por la clase interactiva (CInt _ Información _ Interés) que es la que mostrar a través del lenguaje JavaScript curiosidades, sabías que, entre otros consultados por el estudiante.

Capítulo 3: Descripción de la solución propuesta

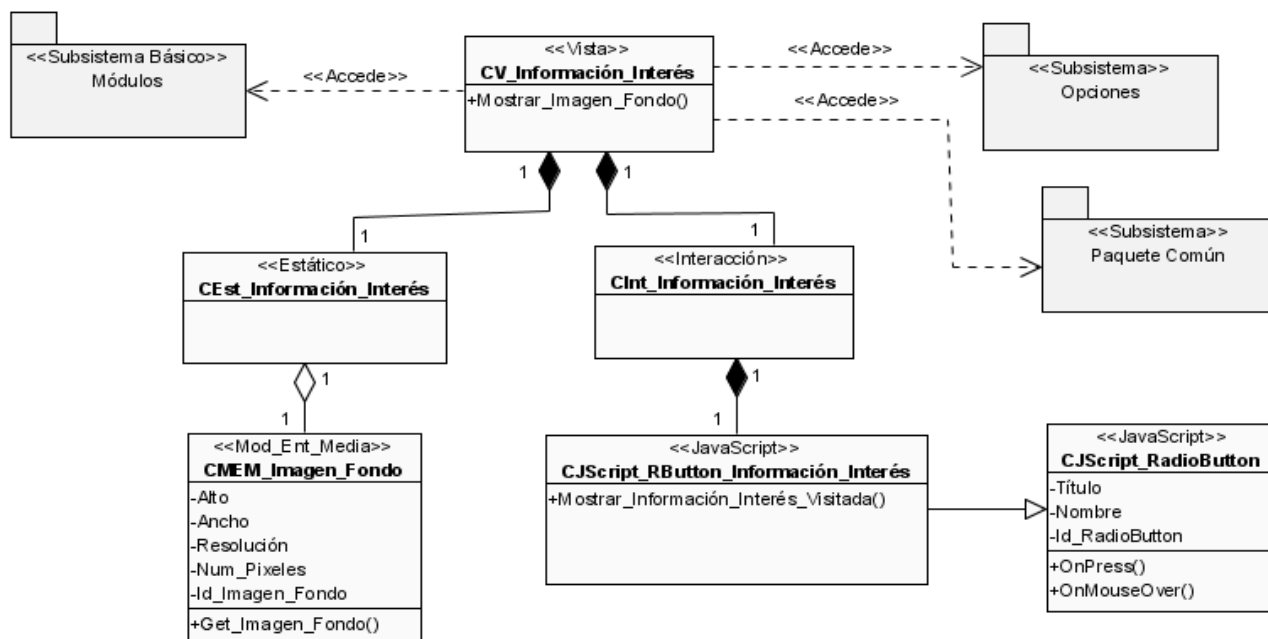


Ilustración 16 Diagrama de estructura de presentación de vista Información _ Interés

Tabla 12 Descripción Textual de Vista de Presentación Información _ Interés

Descripción Textual de la Vista de Presentación: Información _ Interés	
Actores de la Vista de Presentación	Usuario (Profesor o Estudiante)
Propósito	Mostrar al usuario la información de interés que se presenta en la Mediateca del software.
Objetivos pedagógicos	Muestra los resultados de la interacción del estudiante con la información de interés que se presenta en la Mediateca del software: Curiosidades, Sabías que y otras.
Resumen	La vista se inicia cuando el usuario selecciona la opción de trabajo Información _ Interés y se culmina cuando se muestra la información deseada por el usuario.
Vistas asociadas	Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.
Referencias	R1, R2, R3, R4, R5, R6.
Precondiciones	-
Poscondiciones	Vistas de presentación visualizada de acuerdo a la elección del usuario.
Curso Normal de los Eventos	

Capítulo 3: Descripción de la solución propuesta

Acciones del Actor		Respuesta del Sistema	Elementos de la Vista
1. El usuario selecciona la opción Información _ Interés.		2. Busca la información deseada por el usuario. 3. Muestra la información deseada por el usuario.	2.1 Vista Información _ Interés.
Prioridad		Crítica	
Mejoras		-	
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo	Existente
Video o Animación	-		
Sonido	-		
Texto	-		

El presente DEP corresponde a la vista Sesión (Ilustración 17) la cual va a mostrar la fecha y la hora de la sesión de trabajo que se desea visualizar de determinado estudiante. El mismo se encuentra compuesto por la clase estática (CEst _ Sesión) la cual se encuentra asociada a la clase (CMEM _ Imagen _ Fondo) que es la que va a visualizar la media imagen y por la clase interactiva (CInt _ Sesión) que es la que mostrar a través del lenguaje JavaScript la fecha y hora de la sesión consultada.

Capítulo 3: Descripción de la solución propuesta

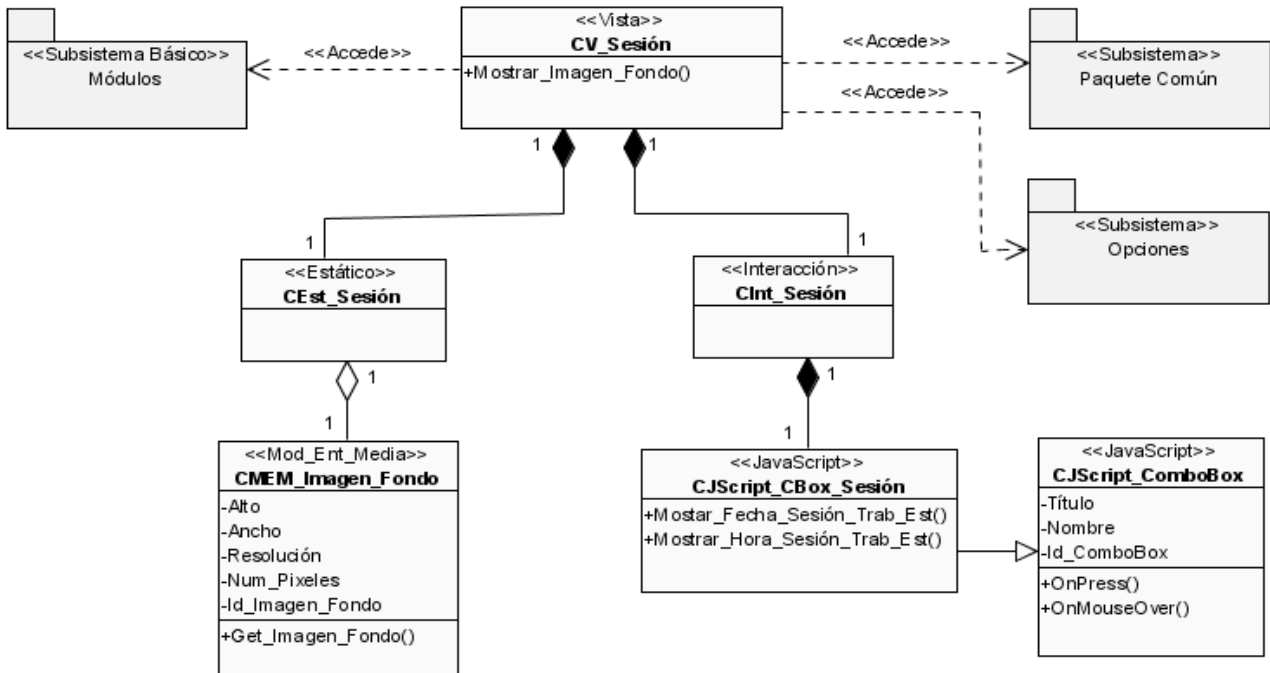


Ilustración 17 Diagrama de estructura de presentación de vista Sección

Tabla 13 Descripción Textual de la Vista de Presentación Sesión

Descripción Textual de la Vista de Presentación: Sesión	
Actores de la Vista de Presentación	Usuario (profesor o estudiante)
Propósito	Mostrar al usuario la fecha y hora en la cual el estudiante consultó la traza seleccionada.
Objetivos pedagógicos	-
Resumen	La vista se inicia cuando el usuario selecciona la opción de trabajo Sesión para observar la fecha y hora en la cual el estudiante consultó la traza seleccionada y se culmina cuando se muestran los mismos.
Vistas asociadas	Vistas del Subsistema Paquete Común, Vista del Subsistema Opciones, Vistas del Subsistema Módulos.
Referencias	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10.
Precondiciones	-
Poscondiciones	Vistas de presentación visualizada de acuerdo a la elección del usuario.
Curso Normal de los Eventos	

Capítulo 3: Descripción de la solución propuesta

Acciones del Actor		Respuesta del Sistema	Elementos de la Vista
1. El usuario selecciona la opción de trabajo Sesión.		2. Busca la fecha y la hora en la cual el estudiante consultó la traza seleccionada. 3. Muestra la fecha y la hora en la cual el estudiante consultó la traza seleccionada	2.1 Vista Sesión.
Prioridad		Crítica.	
Mejoras		-	
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I_01	Imagen de fondo	Existente
Video o Animación	-		
Sonido	-		
Texto	-		

3.7 Conclusiones

En este capítulo se hizo una descripción de la propuesta de solución exponiendo todo lo concerniente a las vistas de presentación del sistema. Se realizó además el modelo de dominio del entorno que presenta el producto y los principales conceptos que lo integran; así como los requerimientos funcionales y no funcionales que son el resultado del análisis de las solicitudes de los clientes. Se presentó además el diagrama de estructura de navegación del sistema para lograr un mejor entendimiento del mismo tanto para los desarrolladores del software como para los usuarios.

CAPÍTULO 4

Construcción de la solución propuesta

4.1 Introducción

A partir de ApEM – L que fue el lenguaje de modelación seleccionado como extensión de UML para modelar el Módulo Resultados de la Colección Multisaber se presentan en este capítulo artefactos que posibilitaron una mejor construcción del sistema.

En él se muestran el diagrama de clases de cada una de las vistas de presentación modeladas en el capítulo anterior; así como el diagrama de colaboración de cada una de ellas y el diagrama de despliegue pertenecientes al sistema actual.

4.2 Área de Estructura Lógica.

El área de estructura lógica es una de las áreas conceptuales de ApEM - L la cual está conformada por la vista estática y la vista de arquitectura del sistema. La primera de ellas está compuesta por el diagrama de clases de cada una de las vistas del sistema y la segunda la componen el diagrama de componentes y el diagrama de despliegue.

4.2.1 Vista Estática.

4.2.1.1 Diagrama de Clases.

El diagrama de clases según el lenguaje de modelado ApEM - L se encuentra dividido en dos grandes zonas, la primera dedicada a las **clases modelo entidad medias** que representan los recursos mediáticos de la aplicación (imagen, texto, sonido, etc.) y la segunda las clases que controlan la *lógica del negocio* de la aplicación propiamente dicha. Esta última zona se encuentra subdividida en cuatro zonas: la primera dedicada a las clase **vista**, que son las encargadas de recibir las peticiones del usuario al sistema y mostrar la información requerida por el mismo, la contigua a ésta dedicada a las clases **controladoras**, las cuales son las encargadas de gestionar las peticiones y la muestra de las respuestas al usuario; la tercera destinada a las clases **modelo**, que son las que encapsulan los datos y sus funcionalidades, la cuarta dedicada a las clases **modelo entidad persistentes** para el tratamiento de la información persistente y por último, las clases correspondientes a lenguaje de alto nivel con que se programe.

Capítulo 4: Construcción de la solución propuesta

A continuación se muestran los diagramas de clases de las vistas que tienen prioridades críticas descritas en el capítulo anterior y una breve descripción del mismo para lograr su entendimiento.

En el diagrama de clases que se presenta a continuación (Ilustración 18) se muestra el diagrama de clases de la vista de presentación Listado _ Estudiante. En el mismo se muestra como se encuentran relacionadas las clases utilizando en lenguaje de modelado ApEM – L y el área que representan cada una de ellas. Primeramente para lograr visualizar una vista se hace falta de la presencia de una clase permita que se ejecute, esta clase es la clase (CV _ Vista). La cual está generalizada por la clase (CV _ Listado _ Estudiante) que se encuentra asociada a la clase controladora (CC _ Listado _ Estudiante) la cual es la encargada de gestionar el modelo así como la vista que generará la respuesta al sistema, la clase modelo (CM _ Listado _ Estudiante) es la que va a procesar y buscar la media existente y la clase modelo entidad persistente (CMEP _ Imagen), es la encargada de localizar la media, en este caso la imagen. La clase (CV _ Listado _ Estudiante) va a tener además agregadas las clases programables del lenguaje de alto nivel (JavaScript), en este caso las clases (CJScript _ Frame) y (CJScript _ TextBox) que son las que van a permitir que se muestran cada uno de los eventos de la vista y por la clase modelo entidad media (CMEM _ Media) que es la que va a mostrar como tal las medias existentes.

Capítulo 4: Construcción de la solución propuesta

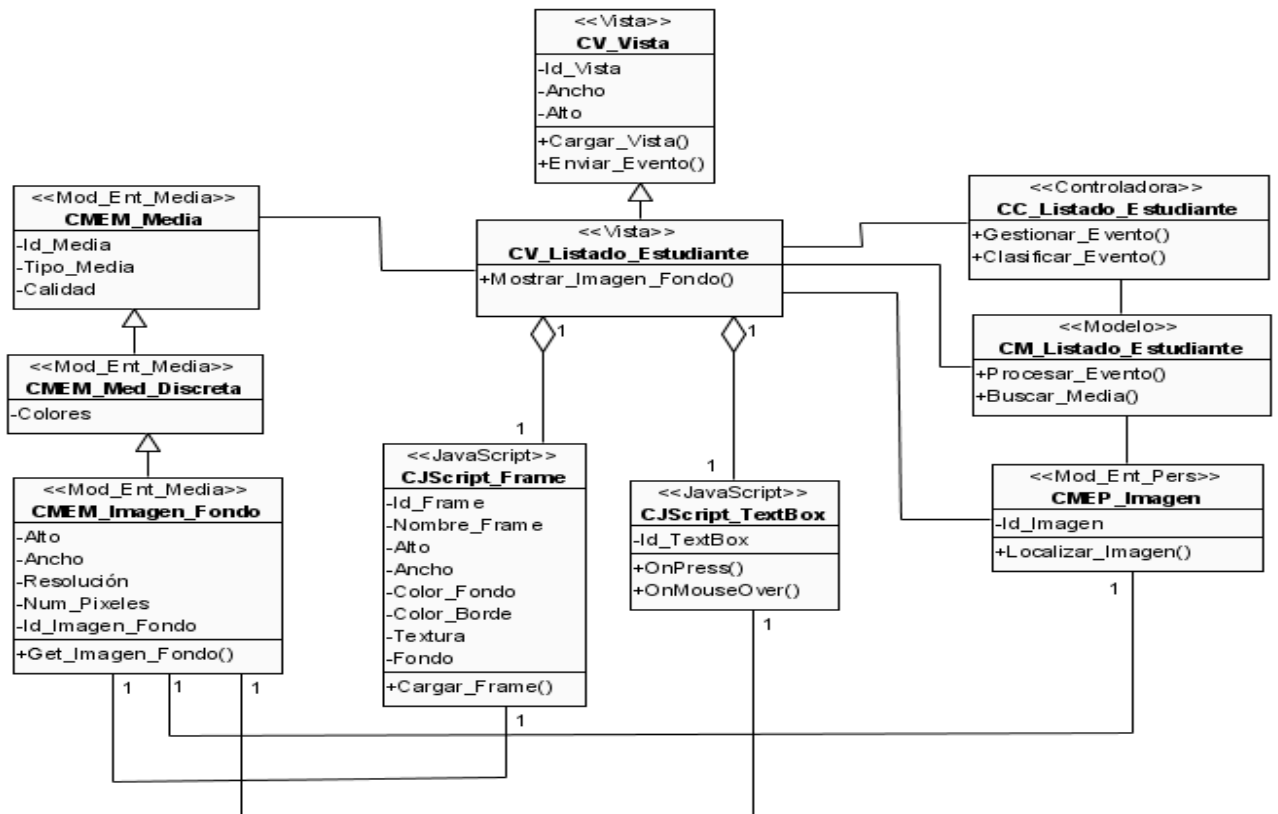


Ilustración 18 Diagrama de Clases de vista de presentación Listado _ Estudiante

En el diagrama que se muestra a continuación (Ilustración 19) al igual que en el diagrama anterior se muestra la clase (CV_Vista) la cual es está generalizada por la clase (CV_Filtros) que está asociadas a las clase controladora (CC_Filtros), a la clase modelo (CM_Filtros) y la clase modelo persistente (CMEP_Imagen). Además va a tener agregadas a ella las clases programables (CJScrip_Combobox) y (CJScrip_Frame) y asociada la clase modelo entidad media (CEMEM_Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CEMEM_Med_Discreta) que a su vez está generalizada por la clase (CEMEM_Imagen_Fondo).

Capítulo 4: Construcción de la solución propuesta

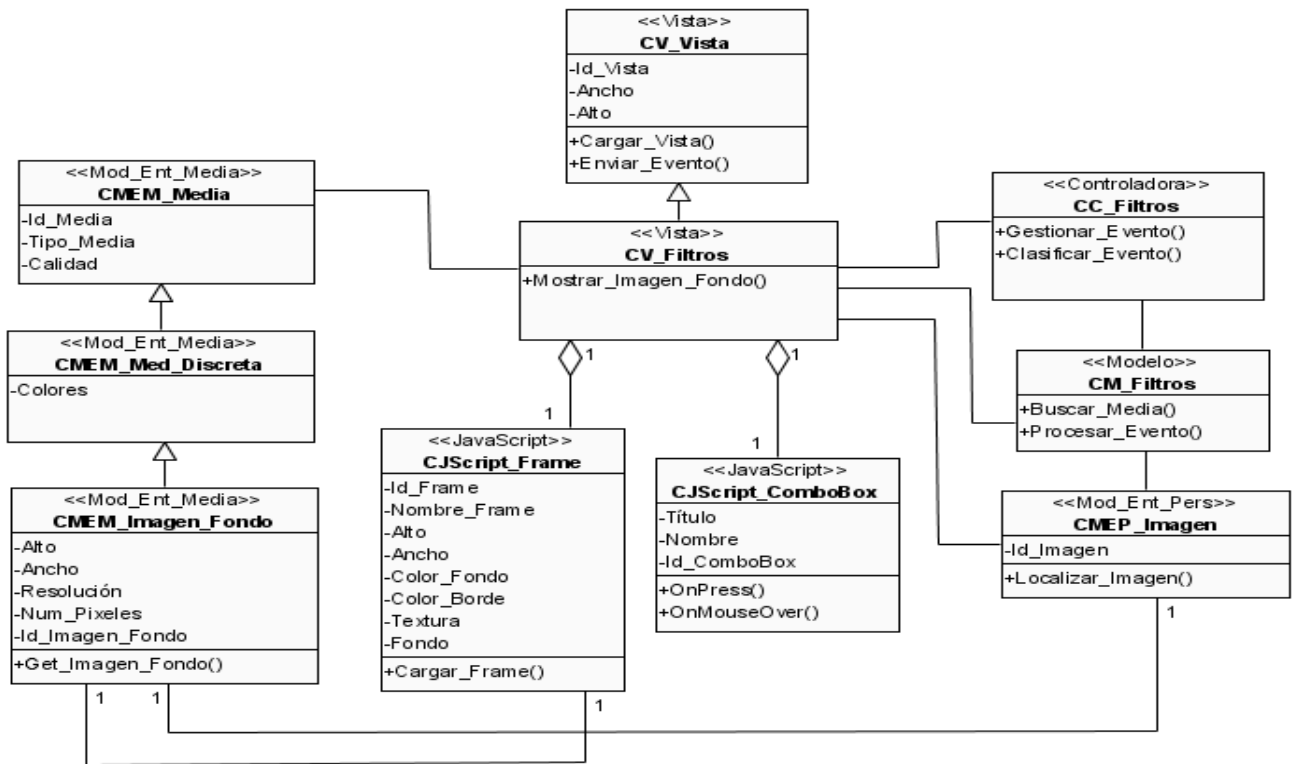


Ilustración 19 Diagrama de Clases de vista de presentación Filtros

En el diagrama que se muestra (Ilustración 20) al igual que en el diagrama anterior se muestra la clase (CV_Vista) la cual es está generalizada por la clase (CV_ Identificación _ Estudiante) que está asociadas a las clase controladora (CC_ Identificación _ Estudiante), a la clase modelo (CM_ Identificación _ Estudiante) y a la clase modelo entidad persistente (CMEP_ Imagen). Además va a tener agregadas a ella las clases programables (CJScrip_ RadioButton) y (CJScrip_ Frame) y asociada la clase modelo entidad media (CMEM_ Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CMEM_ Med_ Discreta) que a su vez está generalizada por la clase (CMEM_ Imagen_ Fondo).

Capítulo 4: Construcción de la solución propuesta

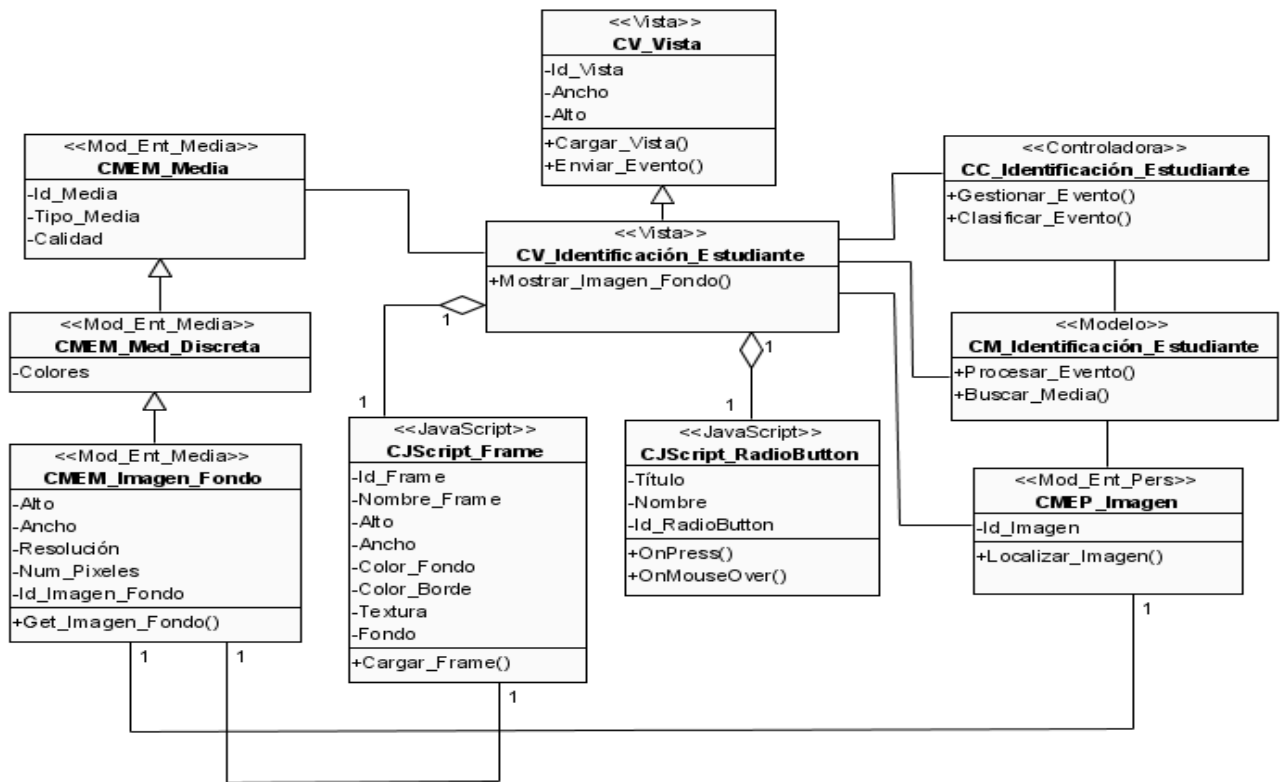


Ilustración 20 Diagrama de Clases de vista de presentación Identificación _ Estudiante

En el diagrama que se muestra (Ilustración 21) al igual que en el diagrama anterior se muestra la clase (CV_Vista) la cual es está generalizada por la clase (CV_Datos_Generales) que está asociadas a las clase controladora (CC_Datos_Generales), a la clase modelo (CM_Datos_Generales) y a la clase modelo entidad persistente (CMEP_Imagen). Además va a tener agregadas a ella las clases programables (CJScrip_RadioButton) y (CJScrip_Frame) y asociada la clase modelo entidad media (CMEM_Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CMEM_Med_Discreta) que a su vez está generalizada por la clase (CMEM_Imagen_Fondo).

Capítulo 4: Construcción de la solución propuesta

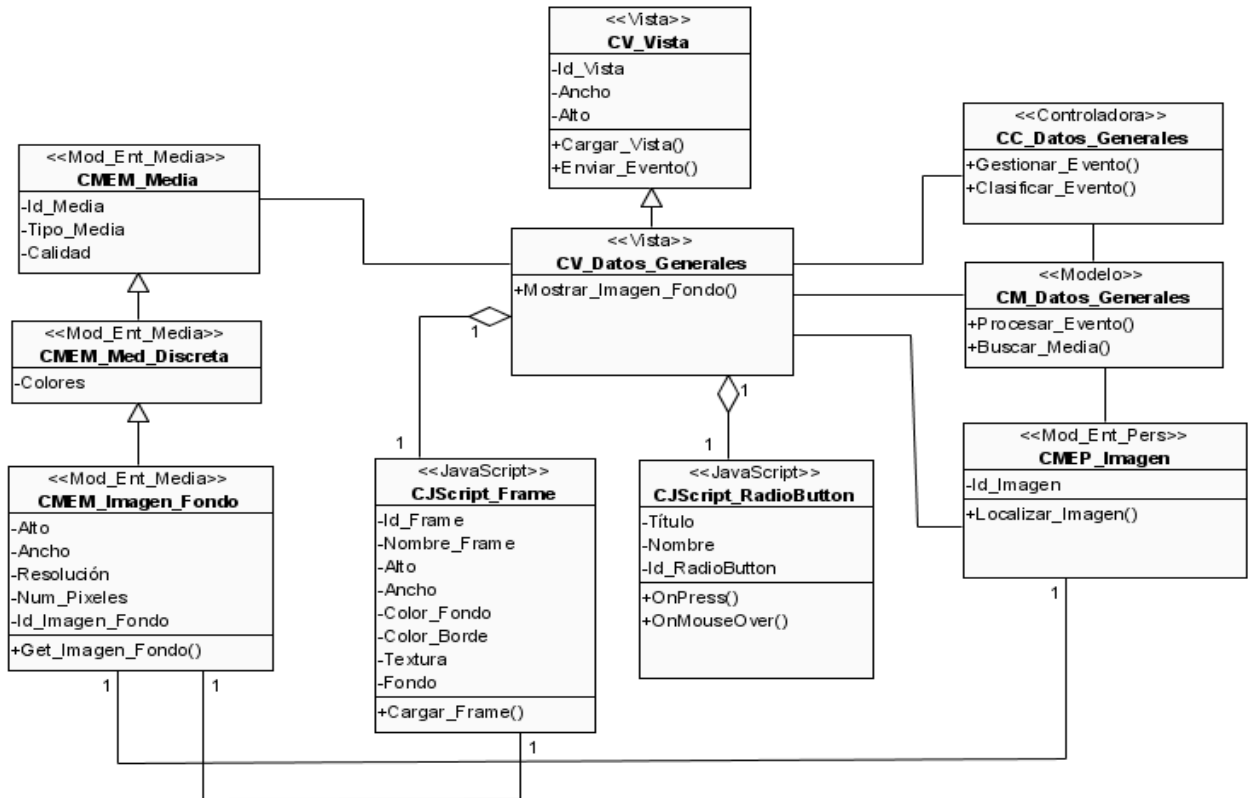


Ilustración 21 Diagrama de Clases de vista de presentación Datos _ Generales

En el diagrama que se muestra (Ilustración 22) al igual que en el diagrama anterior se muestra la clase (CV_ Vista) la cual es está generalizada por la clase (CV _ Temas) que está asociadas a las clase controladora (CC _ Temas), a la clase modelo (CM _ Temas) y a la clase modelo entidad persistente (CMEP _ Imagen). Además va a tener agregadas a ella las clases programables (CJScript _ RadioButton) y (CJScript _ Frame) y asociada la clase modelo entidad media (CMEM _ Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CMEM _ Med _ Discreta) que a su vez está generalizada por la clase (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

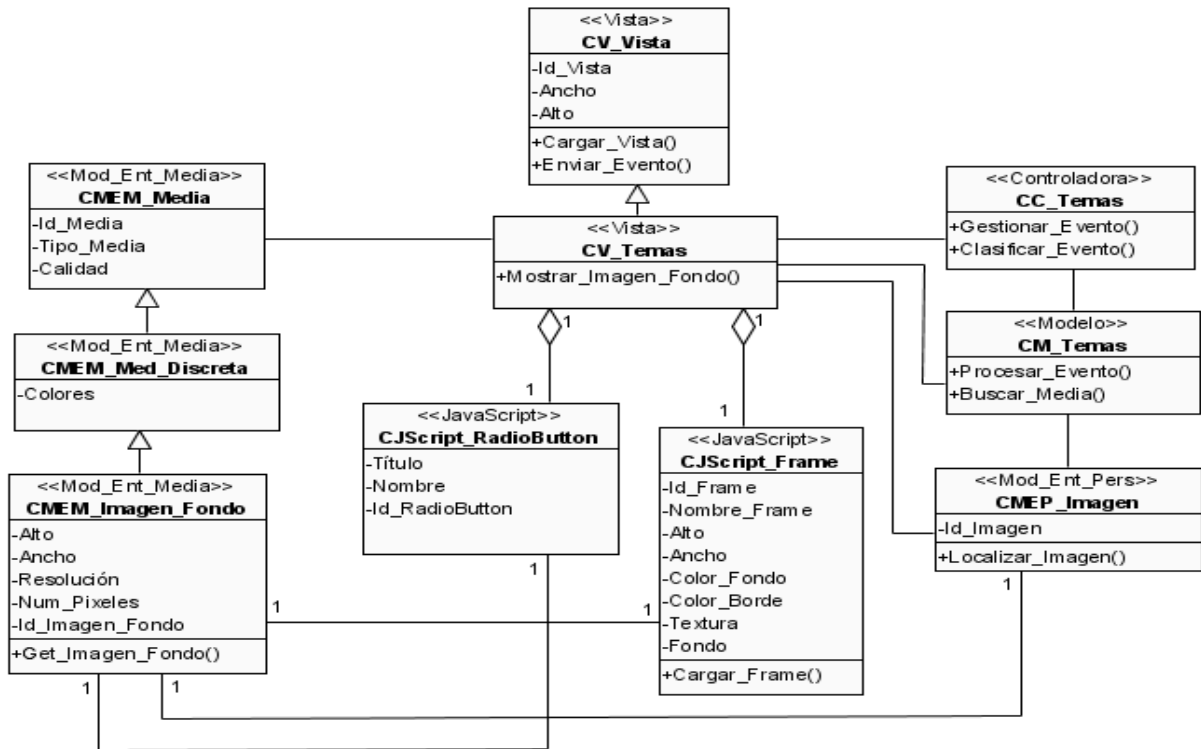


Ilustración 22 Diagrama de Clases de vista de presentación Temas

En el diagrama que se muestra (Ilustración 23) al igual que en el diagrama anterior se muestra la clase (CV_Vista) la cual es está generalizada por la clase (CV_Juegos) que está asociadas a las clase controladora (CC_Juegos), a la clase modelo (CM_Juegos) y a la clase modelo entidad persistente (CMEP_Imagen). Además va a tener agregadas a ella las clases programables (CJScrip_RadioButton) y (CJScrip_Frame) y asociada la clase modelo entidad media (CMEM_Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CMEM_Med_Discreta) que a su vez está generalizada por la clase (CMEM_Imagen_Fondo).

Capítulo 4: Construcción de la solución propuesta

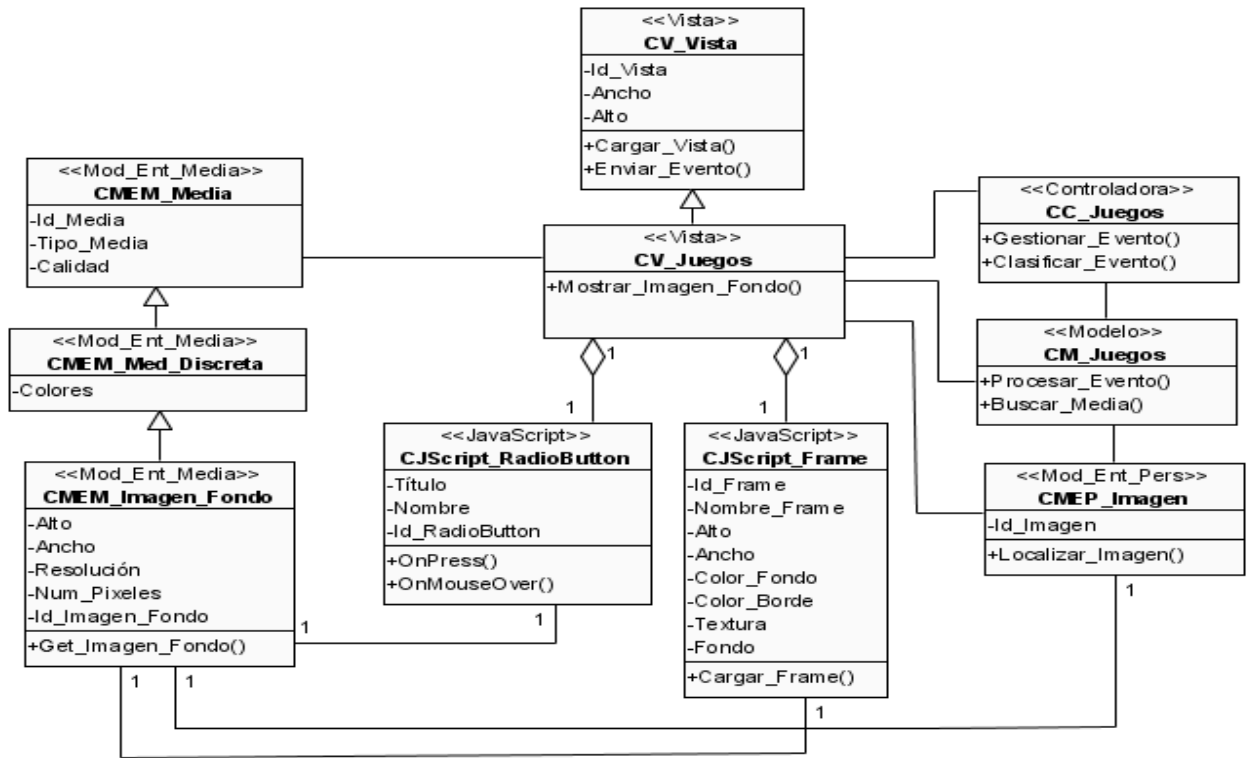


Ilustración 23 Diagrama de Clases de vista de presentación Juegos

En el diagrama que se muestra (Ilustración 24) al igual que en el diagrama anterior se muestra la clase (CV_Vista) la cual es está generalizada por la clase (CV_Ejercicios) que está asociadas a las clase controladora (CC_Ejercicios), a la clase modelo (CM_Ejercicios) y a la clase modelo entidad persistente (CMEP_Imagen). Además va a tener agregadas a ella las clases programables (CJavaScript_RadioButton) y (CJavaScript_Frame) y asociada la clase modelo entidad media (CEMEM_Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CEMEM_Med_Discreta) que a su vez está generalizada por la clase (CEMEM_Imagen_Fondo).

Capítulo 4: Construcción de la solución propuesta

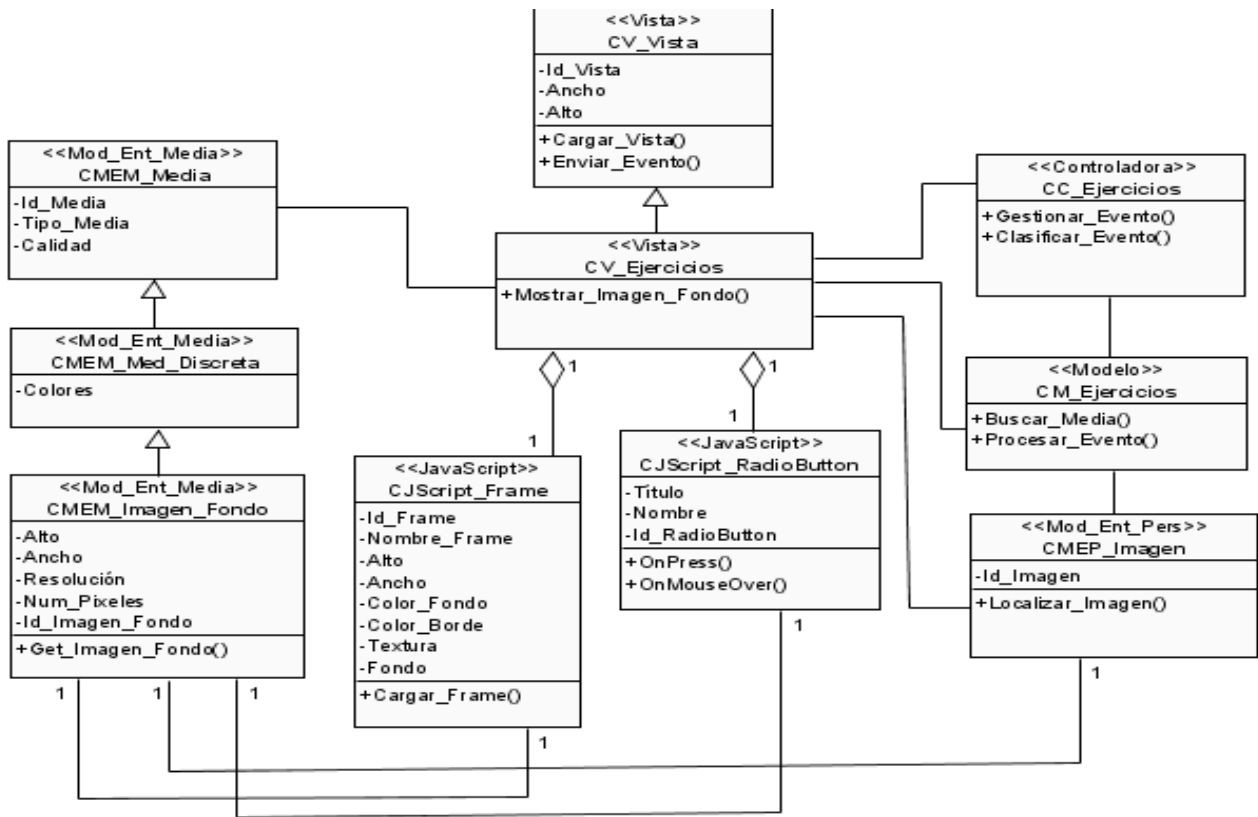


Ilustración 24 Diagrama de Clases de vista de presentación Ejercicios

En el diagrama que se muestra (Ilustración 25) al igual que en el diagrama anterior se muestra la clase (CV_Vista) la cual es está generalizada por la clase (CV_Galería_Imágenes) que está asociadas a las clase controladora (CC_Galería_Imágenes), a la clase modelo (CM_Galería_Imágenes) y a la clase modelo entidad persistente (CMEP_Imagen). Además va a tener agregadas a ella las clases programables (CJScrip_RadioButton) y (CJScrip_Frame) y asociada la clase modelo entidad media (CMEM_Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CMEM_Med_Discreta) que a su vez está generalizada por la clase (CMEM_Imagen_Fondo).

Capítulo 4: Construcción de la solución propuesta

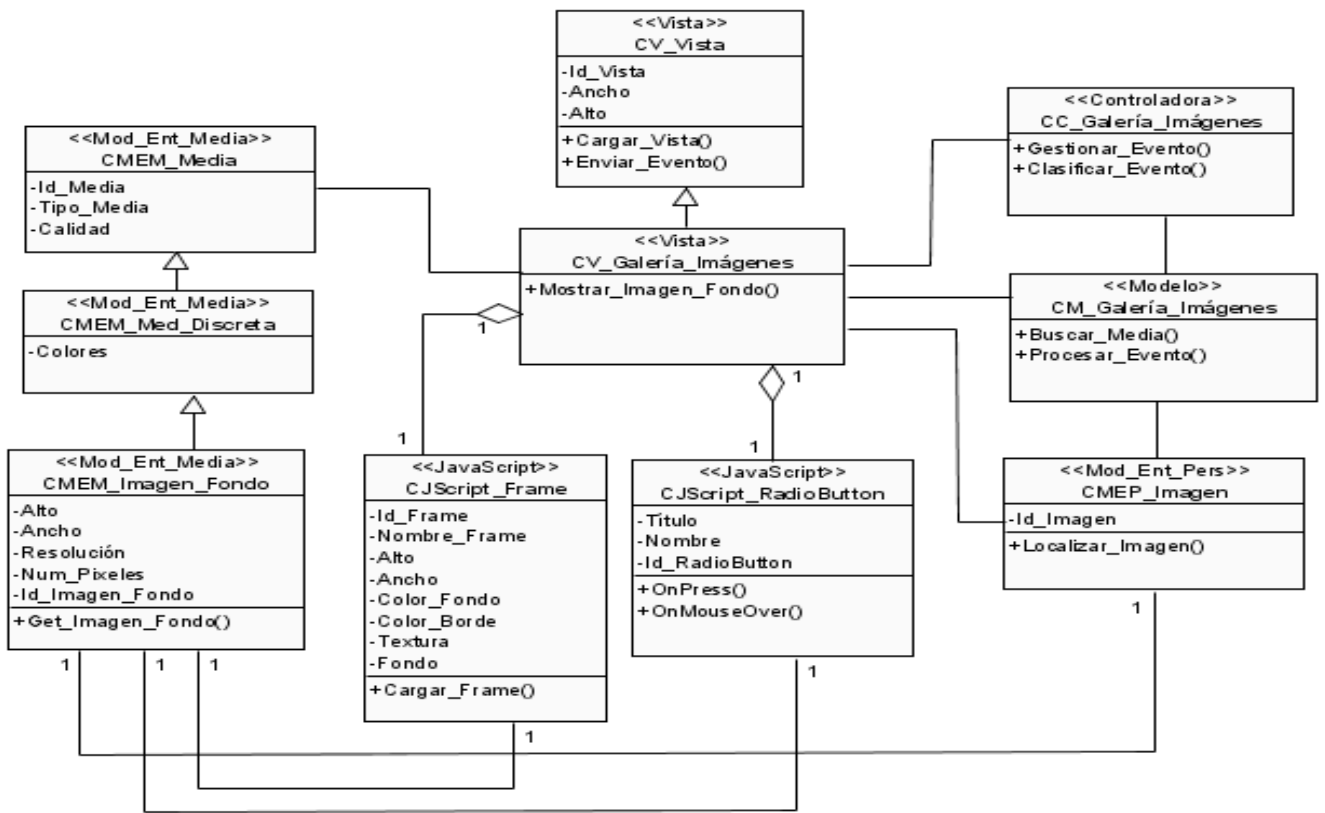


Ilustración 25 Diagrama de Clases de vista de presentación Galería _ Imágenes

En el diagrama que se muestra (Ilustración 26) al igual que en el diagrama anterior se muestra la clase (CV_Vista) la cual es está generalizada por la clase (CV _ Galería _ Videos) que está asociadas a las clase controladora (CC _ Galería _ Videos), a la clase modelo (CM _ Galería _ Videos) y a la clase modelo entidad persistente (CMEP _ Imagen). Además va a tener agregadas a ella las clases programables (CJScrip _ RadioButton) y (CJScrip _ Frame) y asociada la clase modelo entidad media (CMEM _ Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CMEM _ Med _ Discreta) que a su vez está generalizada por la clase (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

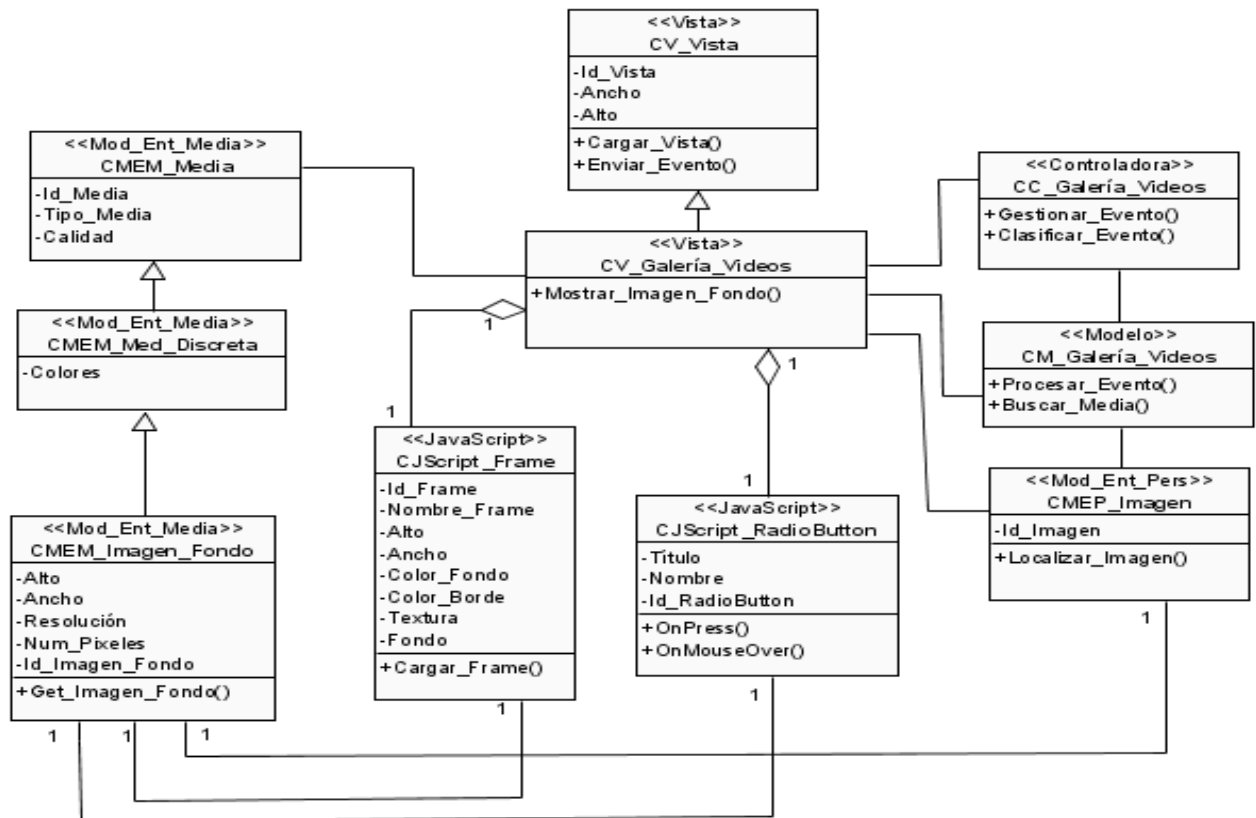


Ilustración 26 Diagrama de Clases de vista de presentación Galería _ Videos

En el diagrama que se muestra (Ilustración 27) al igual que en el diagrama anterior se muestra la clase (CV_Vista) la cual es está generalizada por la clase (CV_Glosario) que está asociadas a las clase controladora (CC_Glosario), a la clase modelo (CM_Glosario) y a la clase modelo entidad persistente (CMEP_Imagen). Además va a tener agregadas a ella las clases programables (CJScript_RadioButton) y (CJScript_Frame) y asociada la clase modelo entidad media (CMEM_Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CMEM_Med_Discreta) que a su vez está generalizada por la clase (CMEM_Imagen_Fondo).

Capítulo 4: Construcción de la solución propuesta

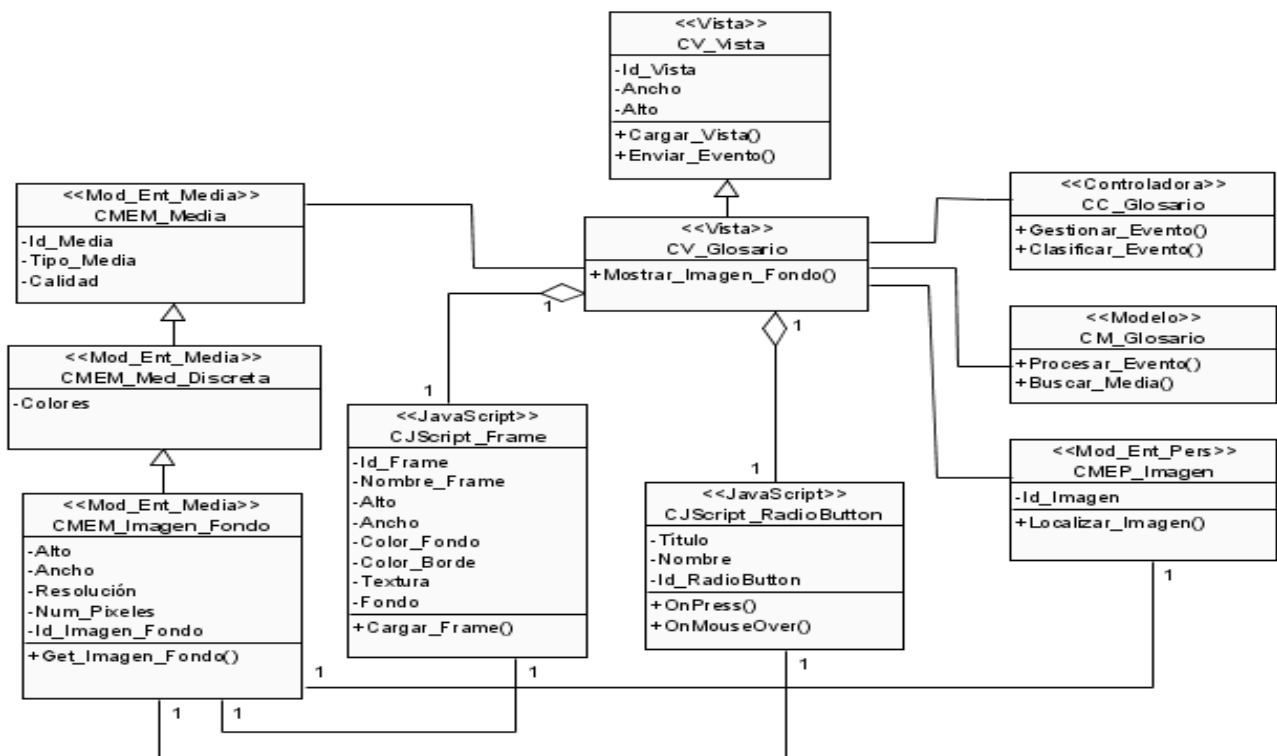


Ilustración 27 Diagrama de Clases de vista de presentación Glosario

En el diagrama que se muestra (Ilustración 28) al igual que en el diagrama anterior se muestra la clase (CV_Vista) la cual es está generalizada por la clase (CV_ Información _ Interés) que está asociadas a las clase controladora (CC_ Información _ Interés), a la clase modelo (CM_ Información _ Interés) y a la clase modelo entidad persistente (CMEP_ Imagen). Además va a tener agregadas a ella las clases programables (CJScrip_ RadioButton) y (CJScrip_ Frame) y asociada la clase modelo entidad media (CEMEM_ Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CEMEM_ Med_ Discreta) que a su vez está generalizada por la clase (CEMEM_ Imagen_ Fondo).

Capítulo 4: Construcción de la solución propuesta

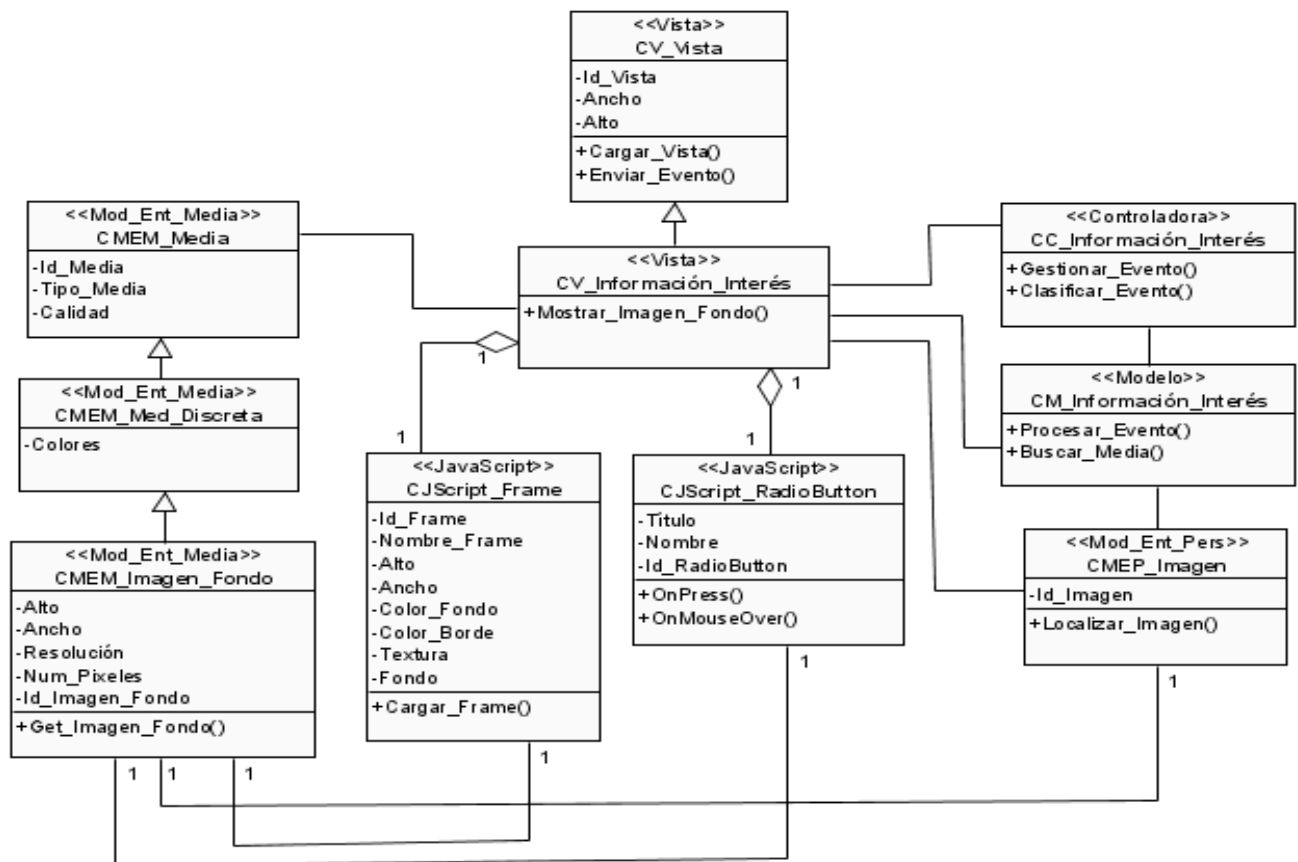


Ilustración 28 Diagrama de Clases de vista de presentación Información _ Interés

En el diagrama que se muestra (Ilustración 29) al igual que en el diagrama anterior se muestra la clase (CV_Vista) la cual es está generalizada por la clase (CV _ Itinerario) que está asociadas a las clase controladora (CC _ Itinerario), a la clase modelo (CM _ Itinerario) y a la clase modelo entidad persistente (CMEP _ Imagen). Además va a tener agregadas a ella las clases programables (CJScrip _ RadioButton) y (CJScrip _ Frame) y asociada la clase modelo entidad media (CMEM _ Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CMEM _ Med _ Discreta) que a su vez está generalizada por la clase (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

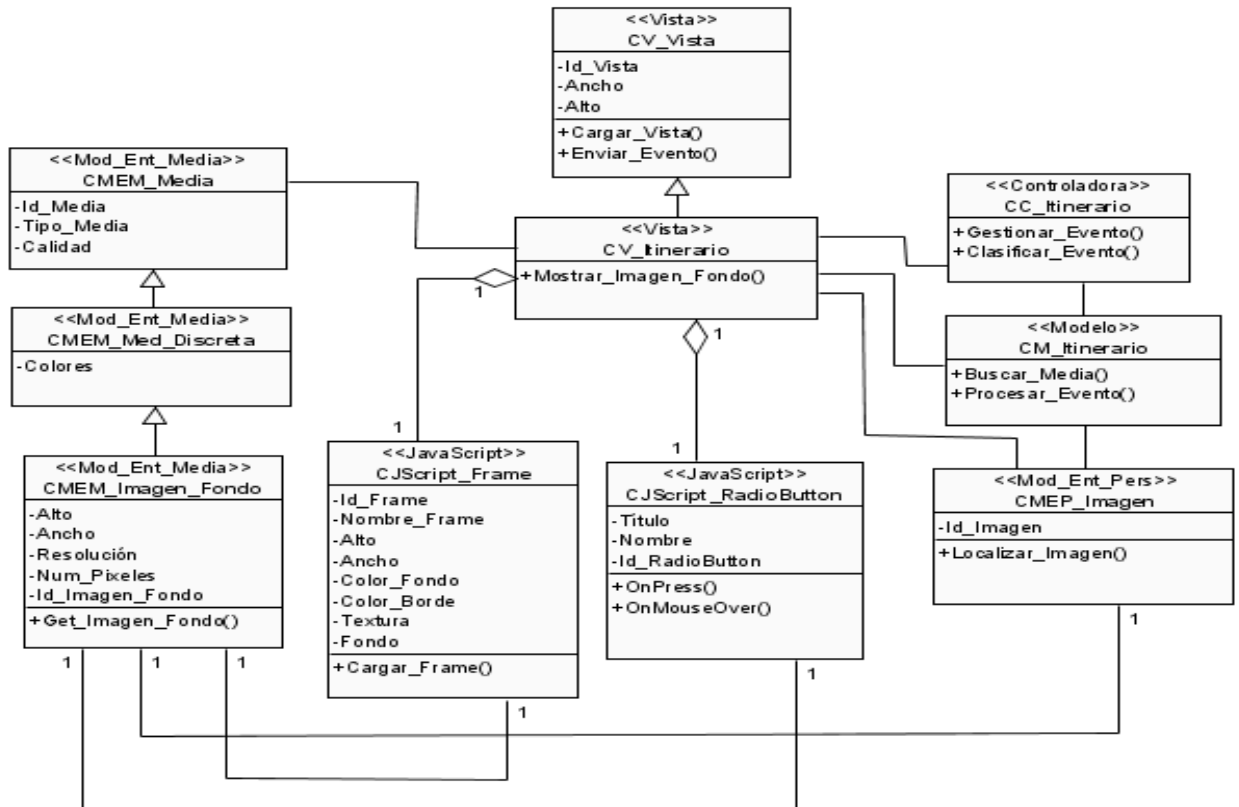


Ilustración 29 Diagrama de Clases de vista de presentación Itinerario

En el diagrama que se muestra (Ilustración 30) al igual que en el diagrama anterior se muestra la clase (CV_Vista) la cual es está generalizada por la clase (CV_Sección) que está asociadas a las clase controladora (CC_Sección), a la clase modelo (CM_Sección) y a la clase modelo entidad persistente (CMEP_Imagen). Además va a tener agregadas a ella las clases programables (CJavaScript_RadioButton) y (CJavaScript_Frame) y asociada la clase modelo entidad media (CMEM_Media), la cual va a estar generalizada por la clase modelo entidad media discreta (CMEM_Med_Discreta) que a su vez está generalizada por la clase (CMEM_Imagen_Fondo).

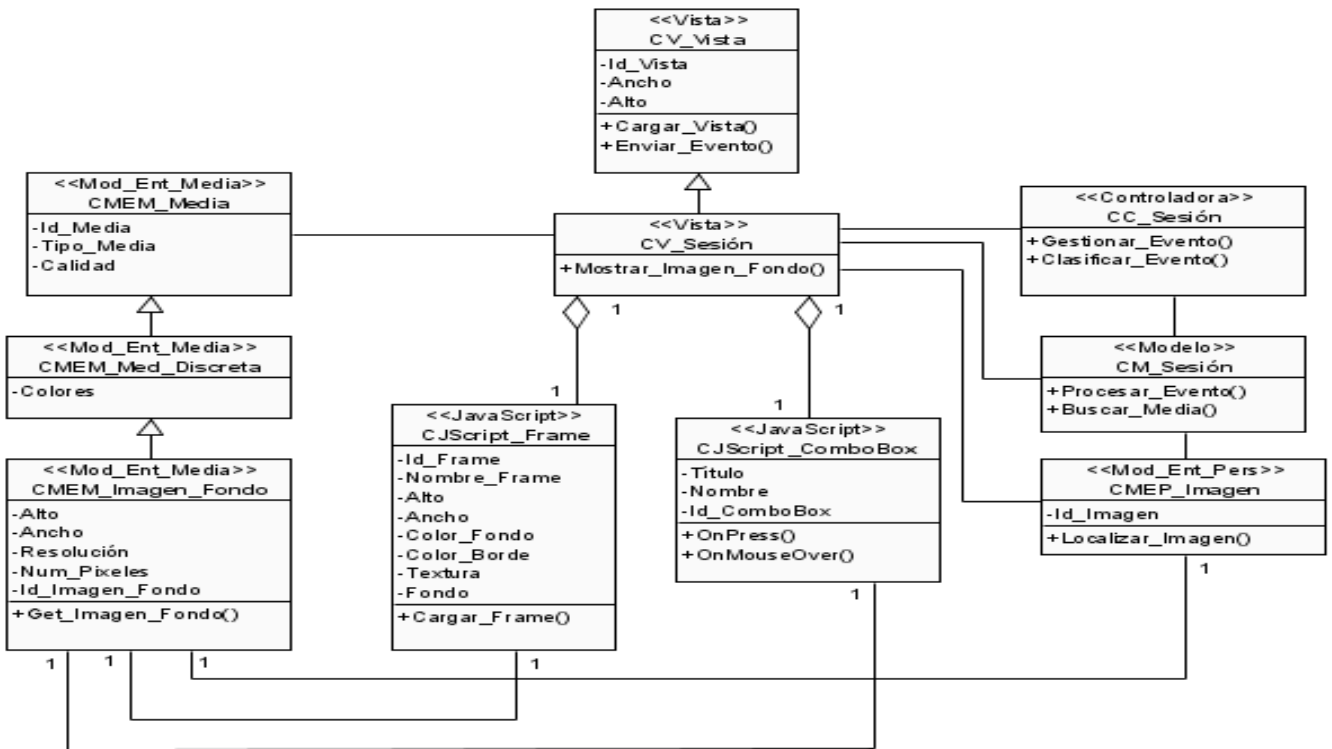


Ilustración 30 Diagrama de Clases de vista de presentación Sesión

4.2.2 Vista de Arquitectura.

4.2.2.1 Diagrama de Despliegue.

El **diagrama de despliegue** utilizando el lenguaje de modelado ApEM – L utiliza lo establecido por el lenguaje base UML, es decir, que este diagrama no sufre modificaciones por lo que utiliza para modelar, el hardware utilizado en las implementaciones del sistema y las relaciones entre sus componentes. Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema, esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. Además este diagrama representa los tipos de nodos del sistema y sus relaciones.

El diagrama de despliegue para el módulo “Resultados” de la Colección Multisaber utiliza una PC cliente la cual está conectada a través del protocolo << TCP / IP >> al servidor de bases de datos, donde se encuentra almacenado todos los datos correspondientes a los estudiantes de las escuelas venezolanas y a través del protocolo << HTTP >> para acceder al servidor Web. (Ilustración 31)

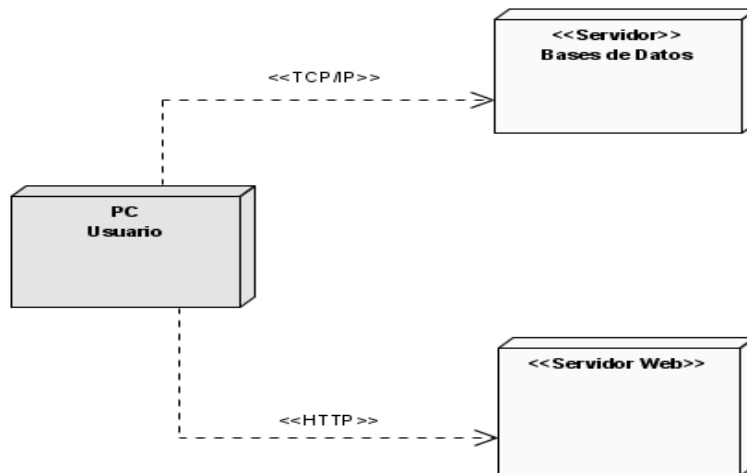


Ilustración 31 Diagrama de Despliegue del sistema propuesto

4.3 Área de Comportamiento Dinámico.

El área de comportamiento dinámico no ha sufrido numerosos cambios en cuanto a lo establecido por el lenguaje base UML sino que ApEM – L ciertamente ha enriquecido la semántica original de los diagramas de este lenguaje. El área de comportamiento dinámico según ApEM – L se encuentra compuesta por la vista de comportamiento, la cual está formada por los diagramas de secuencia, colaboración, los cuáles se muestran a continuación.

4.3.1 Vista de Comportamiento.

4.3.1.1 Diagrama de Secuencia.

Los **diagramas de secuencia** se utilizan en un sistema para mostrar la relación entre los distintos objetos que participan en un escenario de software, en este caso las vistas de presentación. En el diagrama de secuencia la relación se establece a través del paso de mensajes; además estos diagramas son los que dan una idea cronológica de cómo ocurren las interacciones con el usuario y cada objeto viene mostrado por una barra vertical que es la que da idea de tiempo. A continuación se muestra el diagrama de secuencia de las vistas de presentación que tienen prioridad crítica del sistema utilizando el lenguaje de modelado ApEM – L, las cuales según este lenguaje sufre modificaciones pues se adiciona el estereotipo descriptivo y por ende decorativo tiempo para denotar el tiempo como variable de sumo interés en el diseño que muestran las medias en el sistema.

Capítulo 4: Construcción de la solución propuesta

El diagrama que a continuación se muestra (Ilustración 32) es el diagrama de secuencia de la vista Listado _ Estudiante donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista para mostrar el listado de estudiantes del grado y grupo deseado por él, ésta le envía un mensaje a la clase (CV _ Listado _ Estudiante) para cargar la vista y luego la clase (CC _ Listado _ Estudiante) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Listado _ Estudiante) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEM _ Imagen _ Fondo)

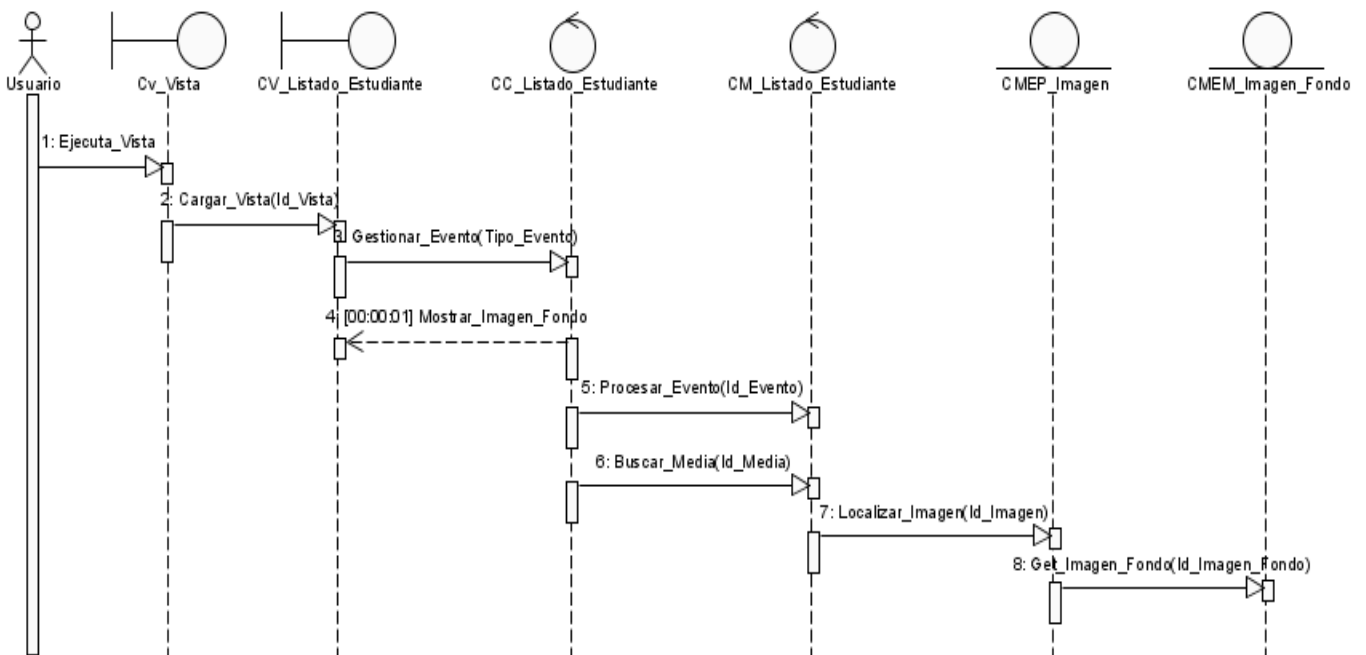


Ilustración 32 Diagrama de Secuencia de Vista de Presentación Listado _ Estudiante

A continuación se muestra (Ilustración 33) el diagrama de secuencia de la vista Filtros donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista, ésta le envía un mensaje a la clase (CV _ Filtros) para cargar la vista y luego la clase (CC _ Filtros) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Filtros) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

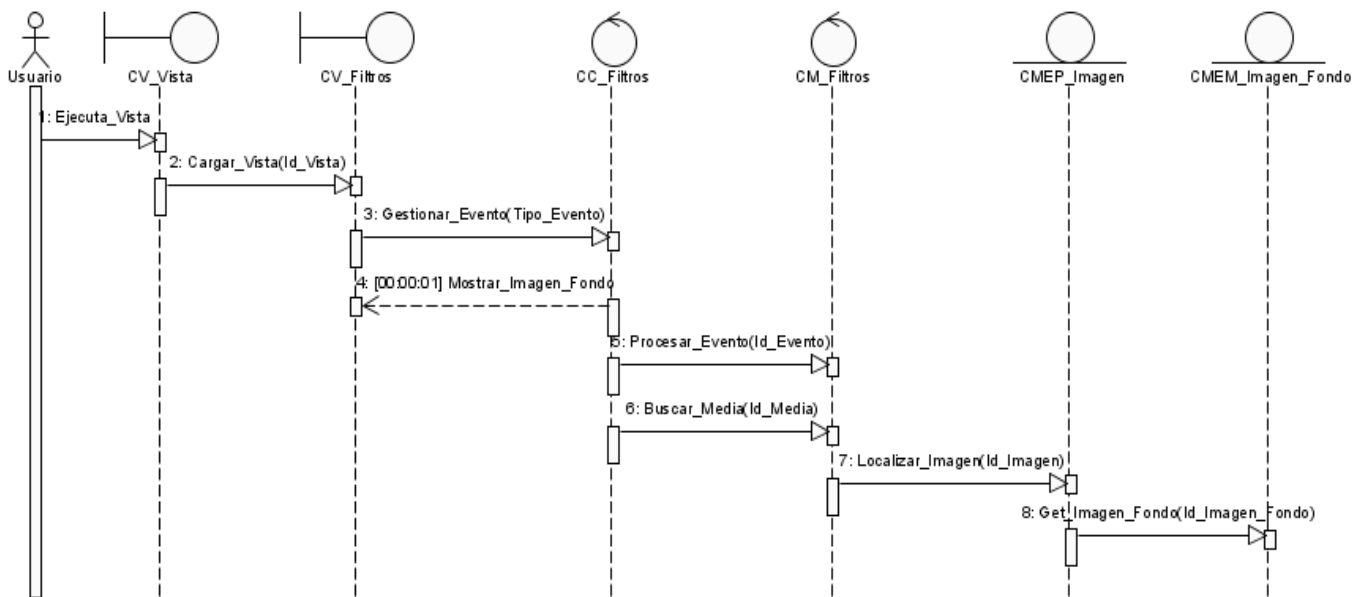


Ilustración 33 Diagrama de Secuencia de Vista de Presentación Filtros

El diagrama de secuencia que a continuación aparece (Ilustración 34) es el correspondiente a la vista Identificación _ Estudiante donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista para mostrar la identificación del estudiante deseado, ésta le envía un mensaje a la clase (CV _ Identificación _ Estudiante) para cargar la vista y luego la clase (CC _ Identificación _ Estudiante) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Identificación _ Estudiante) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

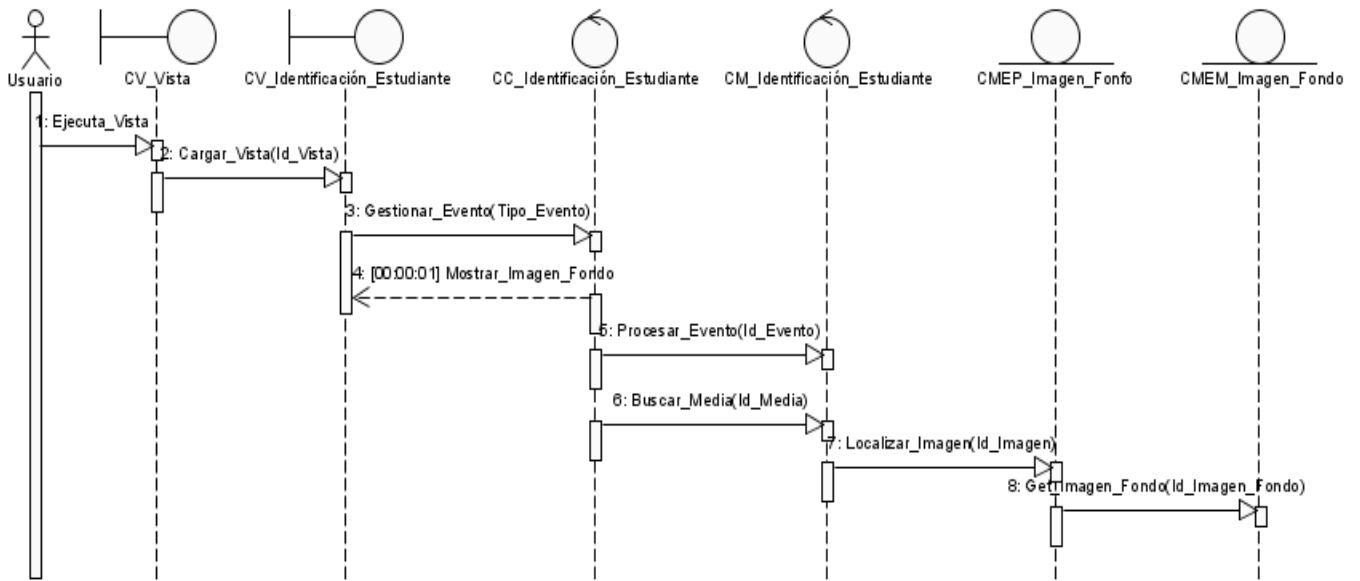


Ilustración 34 Diagrama de Secuencia de Vista de Presentación Identificación _ Estudiante

El diagrama de secuencia que a continuación aparece (Ilustración 35) es el correspondiente a la vista Datos _ Generales donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista para mostrar los datos generales del estudiante seleccionado, ésta le envía un mensaje a la clase (CV _ Datos _ Generales) para cargar la vista y luego la clase (CC _ Datos _ Generales) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Datos _ Generales) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

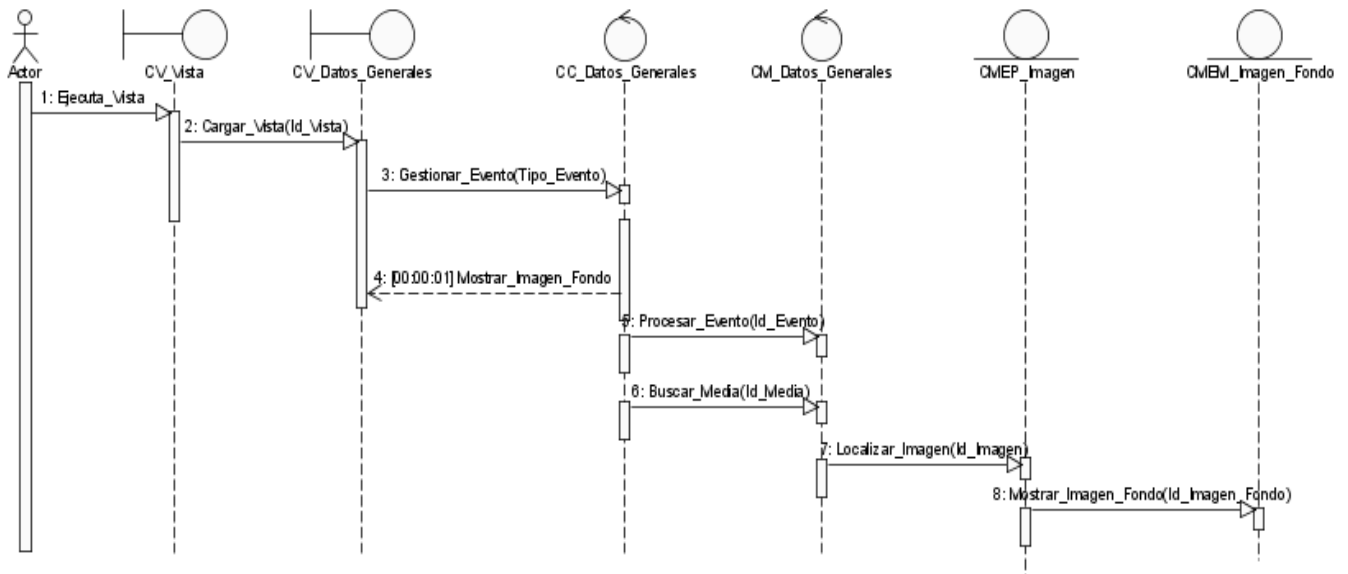


Ilustración 35 Diagrama de Secuencia de Vista de Presentación Datos _ Generales

El diagrama de secuencia que a continuación aparece (Ilustración 36) es el correspondiente a la vista Temas donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista para mostrar los temas visitados por el estudiante; ésta a su vez le envía un mensaje a la clase (CV _ Temas) para cargar la vista y luego la clase (CC _ Temas) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Temas) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEV _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

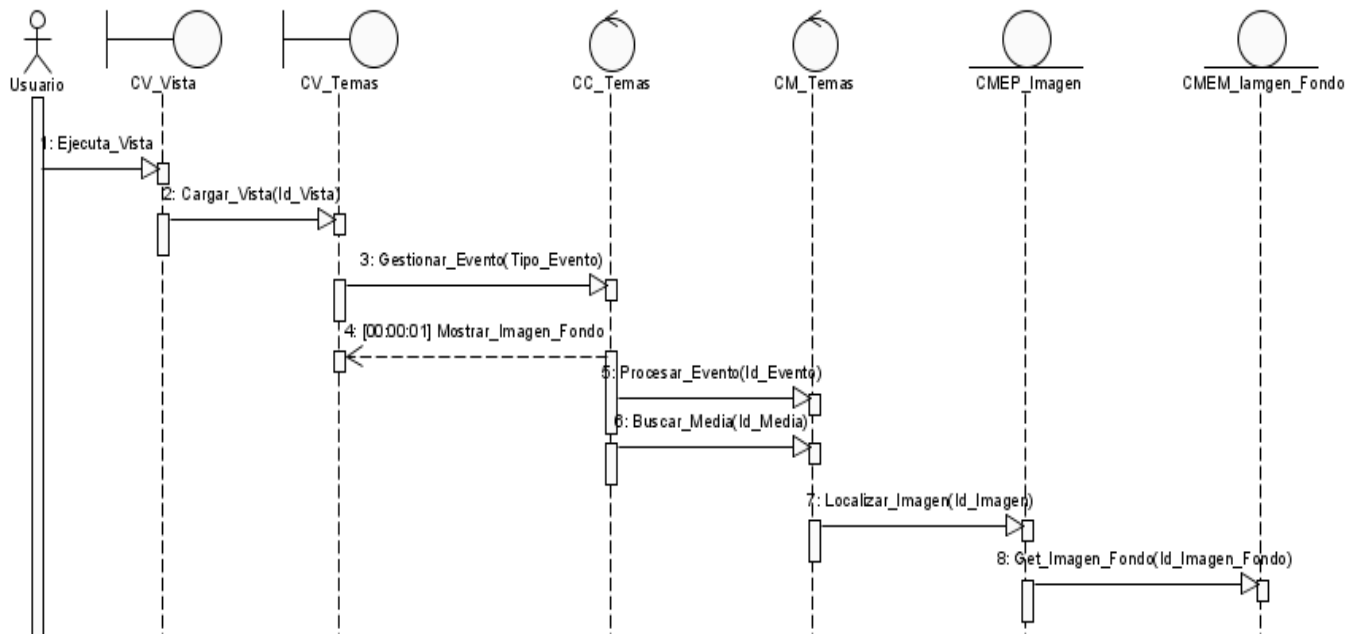


Ilustración 36 Diagrama de Secuencia de Vista de Presentación Temas

El diagrama de secuencia que a continuación aparece (Ilustración 37) es el correspondiente a la vista Ejercicios donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista para mostrar los ejercicios realizados por el estudiante; ésta a su vez le envía un mensaje a la clase (CV _ Ejercicios) para cargar la vista y luego la clase (CC _ Ejercicios) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Ejercicios) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

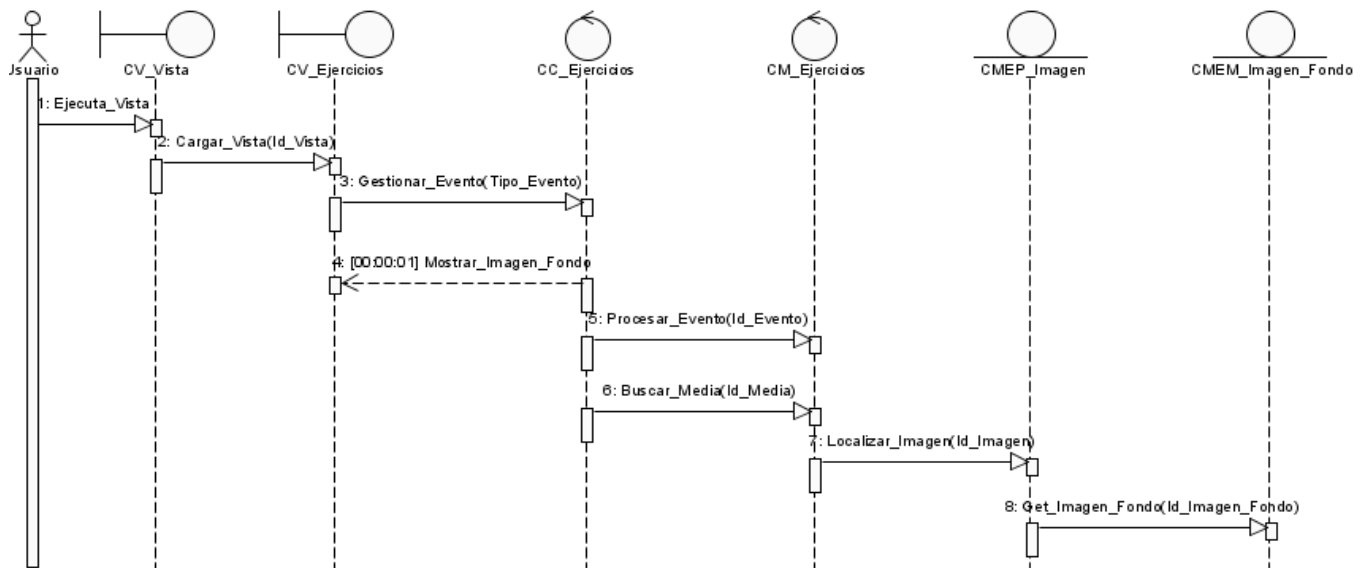


Ilustración 37 Diagrama de Secuencia de Vista de Presentación Ejercicios

El diagrama de secuencia que a continuación aparece (Ilustración 38) es el correspondiente a la vista Juegos donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista para mostrar los juegos realizados por el estudiante; ésta a su vez le envía un mensaje a la clase (CV _ Juegos) para cargar la vista y luego la clase (CC _ Juegos) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Juegos) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

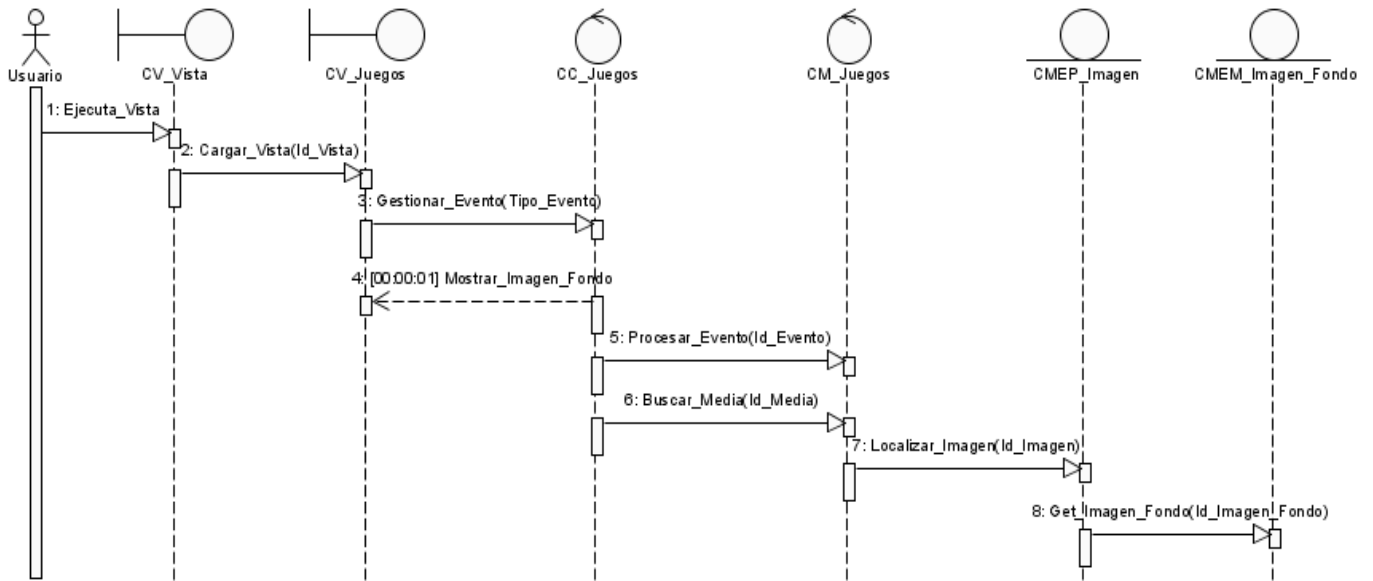


Ilustración 38 Diagrama de Secuencia de Vista de Presentación Juegos

El diagrama de secuencia que a continuación aparece (Ilustración 39) es el correspondiente a la vista Galería _ Imágenes donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista para mostrar las imágenes visitadas por el estudiante; ésta a su vez le envía un mensaje a la clase (CV _ Galería _ Imágenes) para cargar la vista y luego la clase (CC _ Galería _ Imágenes) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Galería _ Imágenes) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

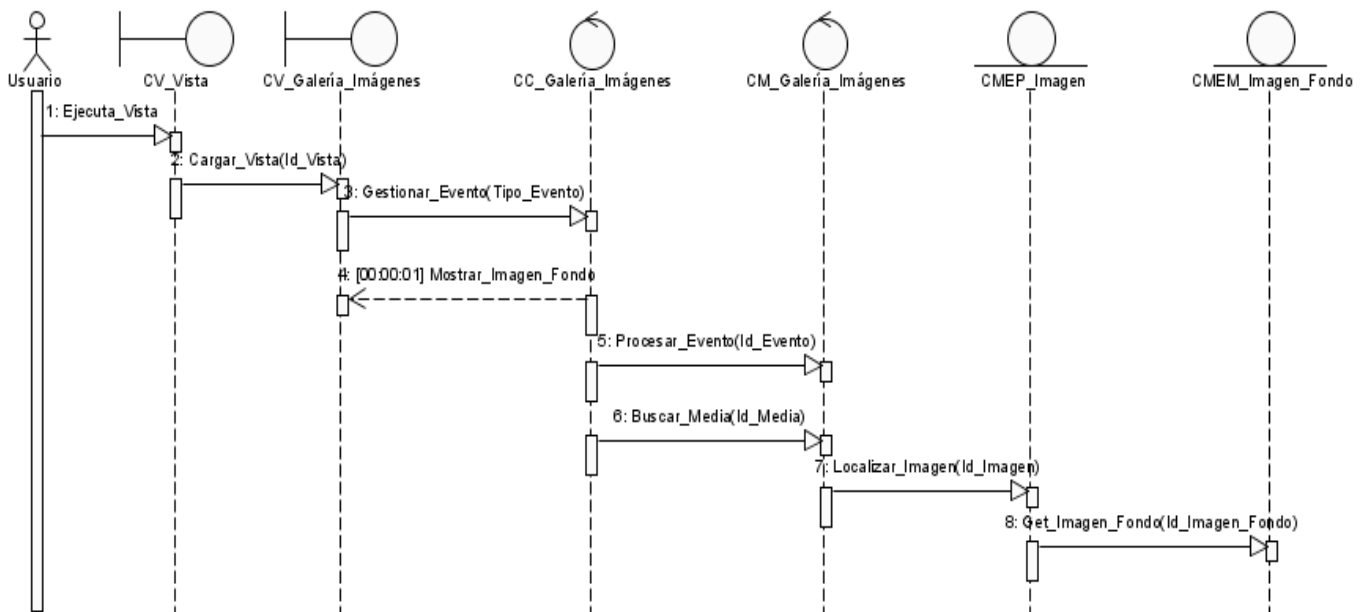


Ilustración 39 Diagrama de Secuencia de Vista de Presentación Galería _ Imágenes

El diagrama de secuencia que a continuación aparece (Ilustración 40) es el correspondiente a la vista Galería _ Videos donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista para mostrar los videos visitados por el estudiante; ésta a su vez le envía un mensaje a la clase (CV _ Galería _ Videos) para cargar la vista y luego la clase (CC _ Galería _ Videos) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Galería _ Videos) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

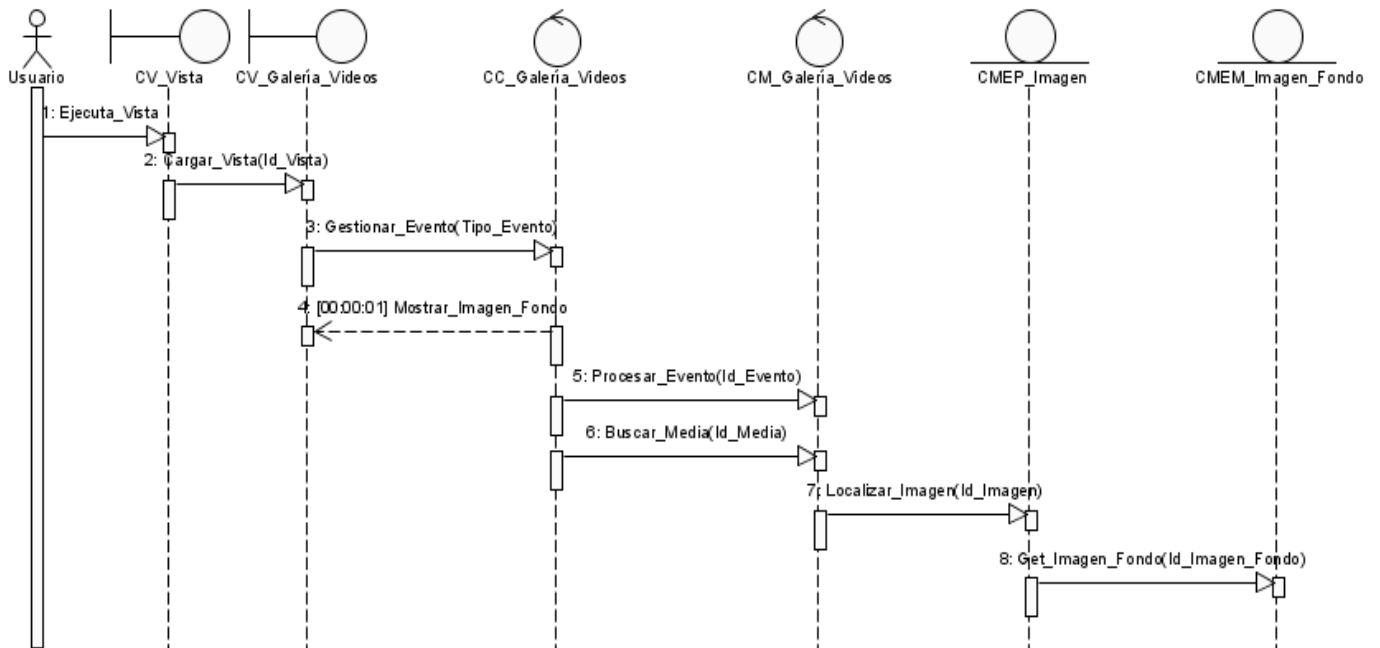


Ilustración 40 Diagrama de Secuencia de Vista de Presentación Galería _ Videos

El diagrama de secuencia que a continuación aparece (Ilustración 41) es el correspondiente a la vista Información _ Interés donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista para mostrar la información de interés consultada por el estudiante; ésta a su vez le envía un mensaje a la clase (CV _ Galería _ Información _ Interés) para cargar la vista y luego la clase (CC _ Información _ Interés) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Información _ Interés) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

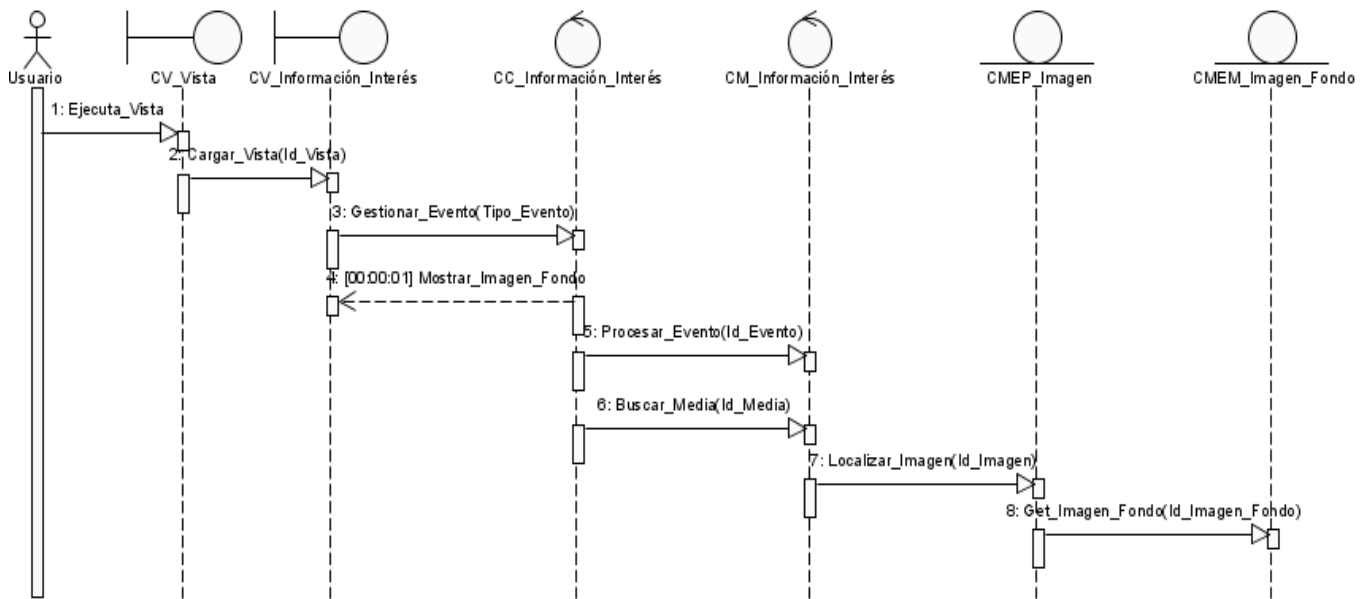


Ilustración 41 Diagrama de Secuencia de Vista de Presentación Información _ Interés

El diagrama de secuencia que a continuación aparece (Ilustración 42) es el correspondiente a la vista Glosario donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista para mostrar las palabras calientes visitadas por el estudiante en la Mediateca del software; ésta a su vez le envía un mensaje a la clase (CV _ Galería _ Glosario) para cargar la vista y luego la clase (CC _ Glosario) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Glosario) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

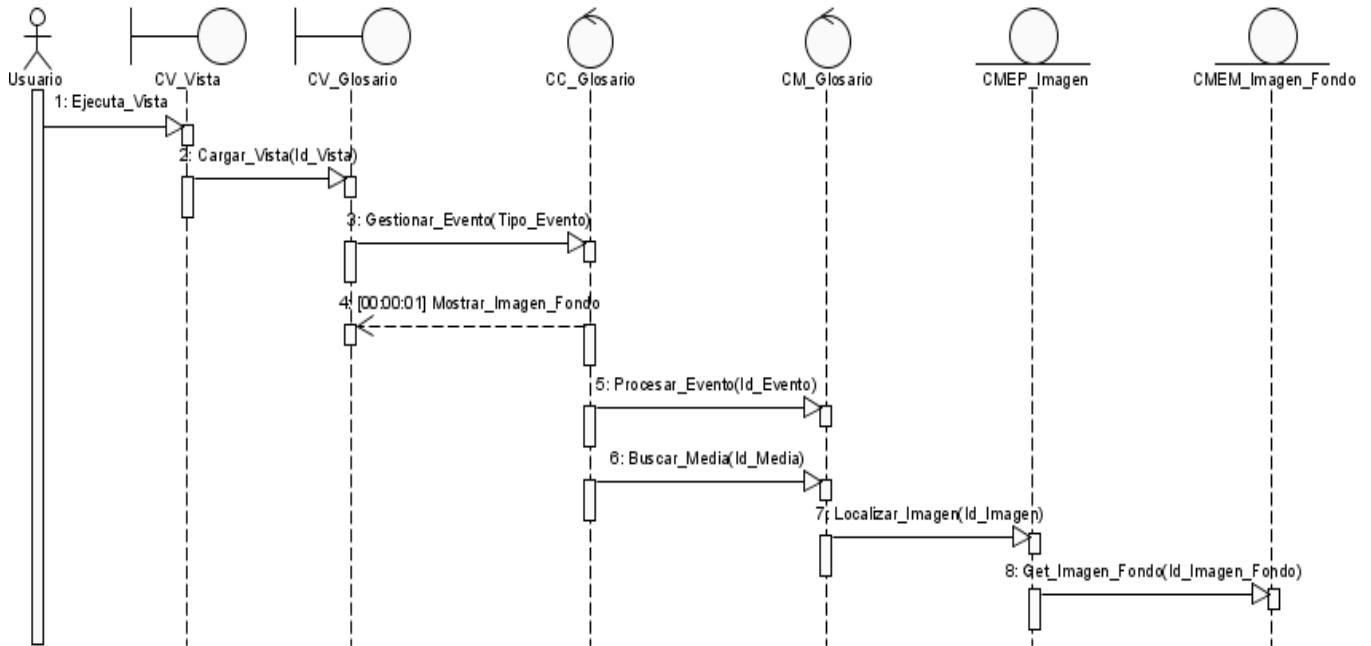


Ilustración 42 Diagrama de Secuencia de Vista de Presentación Glosario

El diagrama de secuencia que a continuación aparece (Ilustración 43) es el correspondiente a la vista Sesión donde el usuario es el que inicia la misma, diciéndole a la clase interfaz (CV _ Vista) que ejecute la vista para mostrar las sesiones especificadas por el estudiante; ésta a su vez le envía un mensaje a la clase (CV _ Sesión) para cargar la vista y luego la clase (CC _ Sesión) es la encargada de gestionar el evento y responde mostrando la Imagen _ Fondo en el tiempo [00:00:01]; donde la clase (CM _ Sesión) es la encargada de procesar el evento y buscar la media correspondiente, la cual es localizada por la clase entidad persistente (CMEP _ Imagen) y mostrada por la clase entidad media (CMEM _ Imagen _ Fondo).

Capítulo 4: Construcción de la solución propuesta

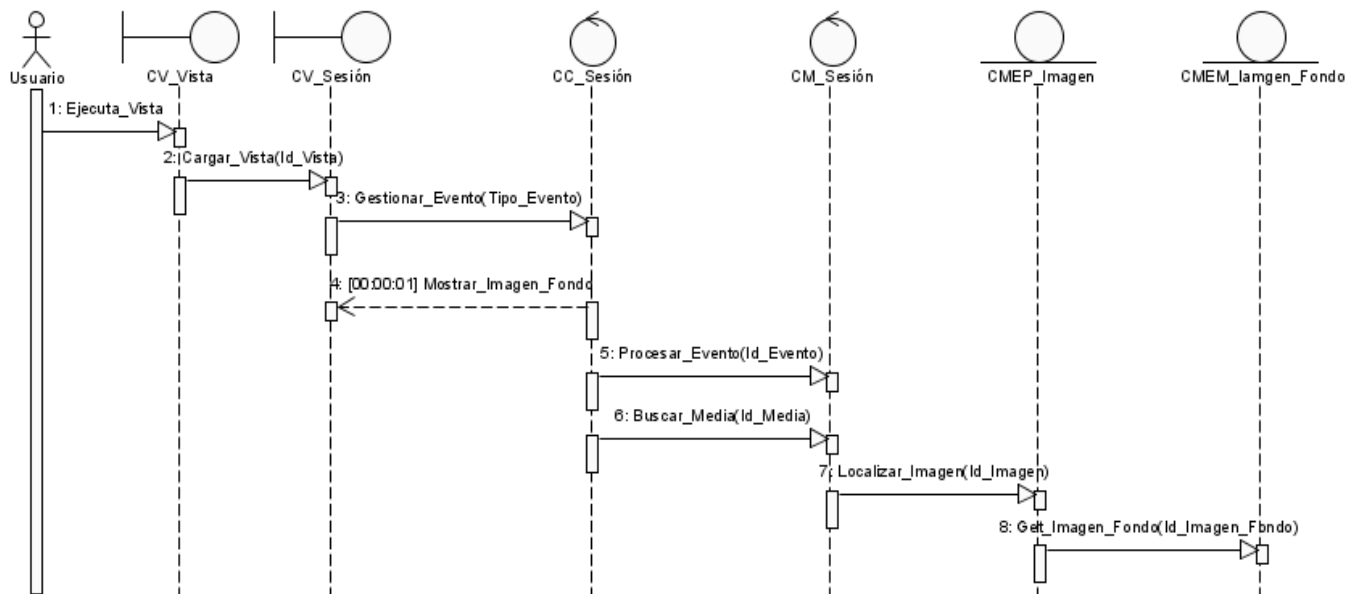


Ilustración 43 Diagrama de Secuencia de Vista de Presentación Sesión

4.4 Conclusiones

Al concluir este capítulo en el que se analizaron las áreas de estructura lógica y área de comportamiento dinámico propuestos por el lenguaje de modelado ApEM – L se pudo apreciar que los diagramas de clases de cada una de las vistas de presentación se encuentran estrechamente relacionadas pues brindan la interacción entre cada una de las clases que la conforman y todas las asociaciones que existen entre ellas. Además con el diagrama de colaboración de cada una de estas vistas se pudo apreciar la interacción entre cada una de los objetos que la componen y observar claramente el flujo de control de ellas; así como en el diagrama de despliegue se observó la arquitectura de los elementos de hardware que presenta la aplicación.

CAPÍTULO 5

Estudio de la Factibilidad

5.1 Introducción

Antes de decidir realizar un proyecto en la esfera productiva es necesario de antemano conocer el tiempo que se utilizará para su desarrollo; así como los recursos humanos a utilizar, los insumos necesarios y los gastos económicos en que se incurrirá; para ello se hace necesario analizar si un proyecto es fiable o no. Es por tal razón que se hace indispensable el estudio de la factibilidad del sistema propuesto. En el presente capítulo se hace un estudio de la factibilidad del sistema utilizando el modelo de COCOMO II y los beneficios que traería su utilización al proyecto.

5.2 Modelo Constructivo de Costes.

El Modelo Constructivo de Costes o Constructive Cost Model (COCOMO II) desarrollado por Barry W. Boehm a finales de los años 70, es un modelo de estimación de costes que incluye tres modelos, donde cada uno ofrece un nivel de detalle y aproximación a medida que avanza el proyecto de desarrollo del software: básico, intermedio y detallado.

El **modelo básico** se basa exclusivamente en el tamaño expresado en LDC, el **modelo intermedio** además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes y los **modelos detallados o avanzados** incluyen todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo.

5.3 Planificación mediante puntos de función.

Tabla 14: Entradas externas

Nombre de la Entrada Externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (Simple, Media y Compleja)
Entrar nombre y apellidos del estudiante.	1	1	Simple
Entrar contraseña en			

Capítulo 5: Estudio de la Factibilidad

caso de ser profesor.	1	1	Simple
-----------------------	---	---	--------

Tabla 15: Salidas Externas

Nombre de la Salida Externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (Simple, Media y Compleja)
Mostrar el nombre de los módulos.	1	1	Simple
Mostrar el nombre de las escuelas venezolanas.	1	1	Simple
Mostrar las opciones de trabajo grado y grupo.	1	1	Simple
Mostrar el listado de estudiantes de un grado y grupo seleccionado.	1	1	Simple
Mostrar la identificación del estudiante.	1	1	Simple
Mostrar los datos generales del estudiante.	1	1	Simple
Mostrar el Itinerario del estudiante.	1	1	Simple
Mostrar los temas visitados por el estudiante.	1	1	Simple
Mostrar los ejercicios realizados por el estudiante.	1	1	Simple
Mostrar los juegos realizados por el estudiante.	1	1	Simple
Mostrar la interacción de los estudiantes con la	1	1	Simple

Capítulo 5: Estudio de la Factibilidad

Galería de Imágenes.			
Mostrar la interacción del estudiante con la Galería de Videos.	1	1	Simple
Mostrar la información de Interés consultada por el estudiante.	1	1	Simple
Mostrar las palabras consultadas por el estudiante en el Glosario de términos.	1	1	Simple
Mostrar la sesión de trabajo del estudiante.	1	1	Simple
Mostrar la interfaz de inicio.	1	1	Simple
Mostrar el contenido de la ayuda cuando sea consultada.	1	1	Simple
Mostrar el contenido del texto buscado por el estudiante.	1	1	Simple
Mostrar el contenido del Acerca de... solicitado por el estudiante.	1	1	Simple

Tabla 16: Puntos de función

Elementos de información	Simples		Medios		Complejos		Subtotal de puntos de función
	Cuenta	Peso	Cuenta	Peso	Cuenta	Peso	
Entradas externas	2	3	0	4	0	6	6
Salidas externas	19	4	0	5	0	7	76
Subtotal (UFP)							82

Capítulo 5: Estudio de la Factibilidad

Tabla 17: Cantidad de instrucciones fuentes

Características	Valor
Puntos de función desajustados	82
Lenguajes	JavaScript 2.0
Instrucciones fuentes por puntos de función	52
Instrucciones fuentes	4264

Los resultados obtenidos son:

- Puntos de función desajustados: 82
- Total de instrucciones fuentes: 4264
- Miles de instrucciones fuentes (**MF**): 4.3 (aproximadamente)

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

Tabla 18: Valores de multiplicadores de esfuerzo

Multiplicadores	Valores
RCPX	1.00
RUSE	0.95
PDIF	0.87
PREX	1.33
FCIL	1.10
SCED	1.00
PERS	1.00
$\pi EM = 1.2$	

Capítulo 5: Estudio de la Factibilidad

Tabla 19: Valores de factores de escala

Factores de escala	Valores
PREC	3.72
FLEX	3.04
RESL	5.65
TEAM	2.19
PMAT	4.68
$\sum SF = 19.28$	

Multiplicadores de esfuerzo:

7

$$EM = \prod_{i=1}^7 E_{mi} = RCPX * RUSE * PDIF * PREX * FCIL * SCED * PERS = 1.2$$

i=1

Multiplicadores de escala:

5

$$SF = \sum_{i=1}^5 SF_i = PREC * FLEX * RESL * TEAM * PMAT = 19.28$$

i=1

Valores calibrados:

$$A = 2.94 \quad B = 0.91 \quad C = 3.67 \quad D = 0.24$$

$$E = B + 0.01 * SF \quad F = D + 0.2 * (E - B)$$

$$E = 0.91 + 0.01 * 19.28 \quad F = 0.24 + 0.2 (1.1 - 0.91)$$

$$E = 1.10 \approx 1.1 \quad F = 0.278 \approx 0.3$$

Capítulo 5: Estudio de la Factibilidad

Esfuerzo:

$$E = 7$$

$$PM = A * (MF) * \sum_{i=1}^{E} E_{mi}$$

$$1.1$$

$$PM = 2.94 * (4.3) * 1.2$$

$$PM = 17.28 \approx 17.3 \text{ hombres / mes.}$$

Cálculo del tiempo:

$$F$$

$$TDEV = c * (PM)$$

$$0.3$$

$$TDEV = 3.67 * (17.3)$$

$$TDEV = 3.67 * 2.35 = 8.62 \approx 8.6 \text{ meses}$$

Cálculo de cantidad de hombres:

$$CH = PM / TDEV$$

$$CH = 17.3 / 8.6 = 2.01 \approx 2 \text{ hombres}$$

Como la cantidad real de hombres a desarrollar la aplicación es 2 desarrolladores, por tanto al reajustar el tiempo de desarrollo según la cantidad de hombres reales, se tiene un tiempo de duración de la aplicación de aproximadamente 8 meses.

Salario Promedio:

Para determinar el salario promedio se tiene en cuenta que los desarrolladores del sistema son profesores del MINED los cuales reciben un salario de \$ 350, por lo que el cálculo del costo del proyecto sería como a continuación se muestra:

Capítulo 5: Estudio de la Factibilidad

Costo:

CHM = 2 * Salario Promedio

CHM = 2 * \$350

CHM = \$ 700

Costo = CHM * PM

Costo = \$ 700 * 17.3

Costo = \$ 12 110 MN

5.5 Beneficios Tangibles e intangibles.

5.5.1 Beneficios Tangibles

Como principales beneficios **tangibles** asociados al Módulo Resultados de la Colección Multisaber se señalan los siguientes:

- La nueva Colección Multisaber contendrá el Módulo Resultados que le permitirá al maestro venezolano y a sus estudiantes observar el resumen de la interacción de la interacción de los educandos en el software.
- El Módulo Resultados de la nueva Colección Multisaber contará con la documentación requerida obtenida en los flujos de trabajo de Análisis y Diseño.

5.5.2 Beneficios Intangibles

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

- Centralizar en el módulo un resumen de los resultados alcanzados por el estudiante durante su interacción con el software.

5.5.3 Análisis de costos y beneficios.

Según el estudio de la factibilidad realizado el costo para realizar el sistema es aproximadamente \$12110 en monedas nacional. Además de que el desarrollo del sistema no supone de grandes gastos de recursos y la base de datos que contiene la información referente al sistema puede ser alojada sin problema alguno, ya que los mismos tienen buenas prestaciones y acceso rápido.

Capítulo 5: Estudio de la Factibilidad

La fácil utilización y navegabilidad del software no genera daño físico alguno a los manipuladores de este, ni los dispositivos utilizados para su uso. La interfaz está diseñada cuidadosamente y resulta agradable el entorno del usuario.

5.6 Conclusiones

En el capítulo que recién ha concluido se realizó el estudio de la factibilidad del sistema donde se estimó los valores mostrados en la siguiente tabla:

Tabla 20: Resultados Estimados

Parámetros	Valores
Esfuerzo	17.3 Hombres / Mes
Tiempo de desarrollo	8.6 Meses
Cantidad de hombres	2 Hombres
Salario Medio	\$ 700 MN
Costo	\$ 12 110 MN

También este capítulo se dedicó a la exposición de los aspectos que de una forma u otra influyen en la ejecución de un proyecto y se expresa claramente la ventaja que implica la implementación de esta aplicación, que permite ahorrar recursos humanos, tiempo de desarrollo así como la necesidad de centralizar la información correcta.

Conclusiones Generales

A partir de la investigación realizada para la elaboración del Análisis y Diseño de esta aplicación utilizando el Proceso Unificado de Desarrollo de Software (RUP), el Lenguaje de Modelación para Aplicaciones Educativas (ApEM – L) como extensión del Lenguaje Unificado de Modelado (UML) que toma como bases teóricas el Lenguaje de Modelado Orientado a Objetos para Aplicaciones Multimedia (OMMMA - L) y el Lenguaje de el estándar OCL, se arriba a las siguientes conclusiones:

- Se estudiaron las herramientas, metodologías y lenguajes existentes en la actualidad utilizados para la modelación de aplicaciones educativas determinando la más óptima para el desarrollo del Módulo Resultados de la Colección Multisaber.
- Se cumplió correctamente el objetivo general planteado para la investigación pues se logró analizar y diseñar el Módulo Resultados de la Colección Multisaber migrada a software libre.
- El Módulo Resultados de la Colección Multisaber cuenta con una correcta documentación garantizando el mantenimiento futuro del módulo y la creación o migración de productos similares.
- Con el estudio de la factibilidad se observó los gastos económicos y los insumos necesarios para realizar el proyecto mostrando los beneficios que traería la utilización del mismo.

Con la propuesta y el estudio realizado del Módulo Resultados de la Colección Multisaber se materializa el objetivo propuesto: Analizar y Diseñar el Módulo Resultados de los nuevos software de la Colección Multisaber definiéndose los artefactos en correspondencia con el nivel descriptivo necesario, que permitirán la comprensión del software y facilitando la documentación requerida para la migración del producto.

Recomendaciones

Al concluir el presente trabajo investigativo se hace necesario realizar un conjunto de recomendaciones para perfilar de esta forma investigaciones futuras para mejorar el resultado científico que se ha obtenido, las cuales se mencionan a continuación:

- Aplicar el Lenguaje de Modelado para Aplicaciones Educativas (ApEM - L) propuesto en este trabajo de diploma en los proyectos productivos de la UCI que desarrollan software educativo pues este lenguaje garantiza la comprensión del entorno de trabajo donde se desarrolla la aplicación.
- Continuar el estudio del módulo con el objetivo de encontrar nuevas funcionalidades para refinar la documentación y lograr una herramienta más completa y general.

Referencias Bibliográficas

- BIANCHINI, A. *Conceptos y definiciones de hipertexto*, 2000. [2008]. Disponible en: <http://www ldc.usb.ve/~abianc/hipertexto.html>
- CABALLERO, I. *"Una herramienta CASE para ADOO: Visual Paradigm"*, 2007. [2008]. Disponible en: http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf
- CAMACHO, M. R. M. and J. P. F. RODRÍGUEZ. *Experiencias de la Aplicación de la Ingeniería de Software en Sistema de Gestión.*, 2006. [2008]. Disponible en:
- CHACÓN, C. G. *Software Libre en Venezuela: Independencia o Soberanía Tecnológica*, 2008. [2008]. Disponible en: <http://www.softwarelibre.cl/drupal/>
- DANIELLE. *"Desarrollo de un Software Educativo para el proceso de fotosíntesis"*, 2005. [Disponible en: <http://cs.uns.edu.ar/jeitics2005/Trabajos/pdf/54.pdf>
- FEBE. *ApEM-L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI*, 2007. p.
- FERNÁNDEZ, G. S. and S. D. CATALÁ. *MULTIMEDIA AUTO-APRENDE*, 2006. [2008]. Disponible en: <http://www.ilustrados.com/documentos/tesis-multimedia-auto-aprende-130907.pdf>
- GARCIA, F. G. *HIPERTEXTO: QUÉ ES Y CÓMO UTILIZARLO PARA ESCRIBIR EN MEDIOS ELECTRÓNICOS*, 2003. [2008]. Disponible en: http://Reddigital.cnice.mecd.es/3/firmas/firmas_francisco_ind.html
- GRAELLS, P. M. *LAS TIC Y SUS APORTACIONES A LA SOCIEDAD*, 2005. [2008]. Disponible en: <http://dewey.uab.es/PMARQUES/tic.htm>
- GRAU, X. F. and M. I. S. SEGURA. *Desarrollo Orientado a Objetos con UML*, 2006. [2008]. Disponible en:
- HISPALINUX. *¿Qué es el software libre?*, 2005. [2008]. Disponible en: <http://www.hispalinux.es/SoftwareLibre>
- LAPUENTE, M. J. L. *Hipertexto: El nuevo concepto de documento en la cultura de la imagen*, 2007. [2008]. Disponible en: <http://www-di.inf.puc-rio.br/schwabe//papers/TAPOSRevised.pdf>
- MAGLIO, F. *La informatización del aprendizaje*, 1999. [2007]. Disponible en: <http://www.fmmeduccion.com.ar>

Referencias Bibliográficas

- PAVÓN, J. *Rational Rose*, 2002. [2008]. Disponible en:
<http://www.fdi.ucm.es/profesor/jpavon/is2/Lab01RationalRose.pdf>
- ROA, S. *Las TIC en los procesos de enseñanza aprendizaje*, 2006. [2008]. Disponible en:
<http://www.isp.fuac.edu.co/encuentros/?p=17>
- SANCHEZ, M. A. M. *¿Qué metodología debo usar para el desarrollo de un Software?*, 2004. [2008].
Disponible en:
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
- SOLÍS, M. C. *Una explicación de la programación extrema (XP)*, 2003. [2008]. Disponible en:
<http://www.willydev.net/descargas/prev/ExplicaXP.pdf>
- ZAVALA. *La Ingeniería de Software*, 2008, 2002. [5/02]. Disponible en:
<http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>

Bibliografía

- AGUILAR, A. *Programación Extrema y Software Libre.*, 2002. [2008]. Disponible en: <http://www.seguridad.unam.mx/eventos/datos/ev11/semi18/mat.7.pon19.semi18.pdf>
- CALERO, M. "*Una explicación de la programación extrema (XP)*", 2003. [2008]. Disponible en: <http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/explicaxp.pdf>
- CAMACHO, M. "*Experiencias de la aplicación de la Ingeniería de Software en sistemas de gestión.*" 2000. [2008]. Disponible en: http://www.cecam.sld.cu/pages/rcim/revista_1/articulos_pdf/r0100a01.pdf
- CONSTITUCIÓN. *Gaceta Oficial de la República Bolivariana de Venezuela*, 2000. [2008]. Disponible en: <http://www.mijuicio.com/leyes/organicas/22.pdf>
- FRESNO, C. "*En pos de la creación de la Sociedad Mundial de la Información / Conocimiento.*" 2006. [2008]. Disponible en: http://www.cecam.sld.cu/pages/rcim/revista_10/articulos_htm/poscreacion.htm
- GRAELLS, P. *Multimedia Educativa*, 2007]. Disponible en: <http://dewey.uab.es/pmarques/funcion.htm>.
- JAUQUE, M. *V Jornadas del Software Libre en Asturias*, 2005. [2008]. Disponible en: <http://www.asturlinux.org/jornadas2005/programacion-extrema.php>
- MENDOZA, M. A. *Qué metodología debo usar para el desarrollo de un software?*, 2004. [2008]. Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
- MONTSERRAT, C. "*Software Libre vs Software Propietario*" *Ventajas y desventajas.*, 2006. [2008]. Disponible en: <http://www.softwarelibre.cl/drupal//files/32693.pdf>

Glosario de términos y siglas

Términos

Borland Delphi: Lenguaje de Programación diseñado para la programación de propósito general con énfasis en la programación visual; producido por empresas estadounidenses y en sus diferentes variantes permite producir archivos ejecutables.

Contextualización: (Contextualizar una información) Significa llevar elementos nuevos y paralelos, al hecho central que origina la noticia, realizando aportes que permitan aclararlo y ponerlo en relación con otros hechos laterales. Lo que se busca de este modo, es profundizar los niveles de información que se brindan, aportando al lector la mayor cantidad de elementos posibles que contribuyan a la comprensión de los mensajes transmitidos.

Hiperdocumento: Es el contenido de información, incluyendo los fragmentos de información y las conexiones entre esos fragmentos a través de un nodo.

Hipermedio: Es una generalización de hipertexto que se utiliza al igual que él datos que se almacenan en una red de nodos conectados por enlaces, los cuales contienen textos, gráficos, imágenes, animaciones, audio, video.

Hardware: Es el substrato físico en el cual existe el software. El hardware abarca todas las piezas físicas de un ordenador (CPU, placa base, etc.).

Macromedia Flash 5: Herramienta orientada para crear aplicaciones y contenidos dinámicos, es decir, utilidades interactivas y multimedia con una amplia posibilidad de animación.

Macromedia Director: Es un programa de Adobe Systems Incorporated para la producción de películas ejecutables en Adobe Shockwave, usando mapas de bits y en programación Lingo. Este software permite generar presentaciones multimedia que pueden ser distribuidas a través de CDs. Permite incorporar a las películas múltiples formatos, como imágenes vídeos, sonidos o animaciones Flash. Incluye editores básicos para texto, mapa de bits, vectores, sonido.

Paradigma: Un modelo o patrón en cualquier disciplina científica.

Glosario de Términos y Siglas

Programador: Un programador es un individuo que ejerce la programación, es decir, que escribe programas de computadora u ordenador. Los programadores también reciben el nombre de desarrolladores de software.

Sistema Operativo: El sistema operativo es el programa (o software) más importante de un ordenador (computadora).

Software: Se refiere a los programas y datos almacenados en un ordenador.

Software Educativo: Es el software destinado a la enseñanza y autoaprendizaje que permite el desarrollo de ciertas habilidades cognitivas.

Toolbook: Sistema para desarrollo de aplicaciones destinado a Windows, de Asymetrix Corporation, Bellevue, WA, que emplea una metáfora de "página y libro", análoga a la de "tarjeta y pila" de HyperCard. Su lenguaje OpenScript es similar a HyperTalk.

Siglas

CNTI (Centro Nacional de Tecnologías de Información): Es una institución adscrita al Ministerio del Poder Popular para las Telecomunicaciones y la Informática, con el propósito de contribuir a la eficiencia y efectividad del Estado en Venezuela.

ISP (Institutos Superiores Pedagógicos): Son centros universitarios de la República de Cuba encargados de la formación del personal pedagógico para las distintas educaciones del país (pregrado). También atiende la capacitación y estudios de posgrado y la formación de máster y doctores.

MCT (Ministerio de Ciencia y Tecnología): Es el organismo en Venezuela encargado de conformar y mantener el Sistema Nacional de Ciencia, Tecnología e Innovación.

MES (Ministerio de Educación Superior): Ministerio de Educación Superior de la República de Cuba creado desde 1976, para lograr integrar en una concepción pedagógica coherente el legado de nuestros más ilustres educadores; con las actuales exigencias científico – técnicas de la formación de profesionales.

Glosario de Términos y Siglas

MIC (Ministerio de Informática y Comunicaciones): Es el organismo rector de las disciplinas de tecnología de la información, industria electrónica, telecomunicaciones, radiodifusión, servicio postal y espectro radioeléctrico de estas disciplinas en Cuba.

MINED (Ministerio de Educación): Es un organismo de la República de Cuba encargado de gestionar las tareas administrativas relacionadas con la educación en los diferentes sectores educacionales (primaria, secundaria y preuniversitario).

UJC (Unión de Jóvenes Comunistas): Es la organización juvenil del Partido Comunista de Cuba, comprende jóvenes entre los 15 y 30 años, los cuales deben mantener una excepcional conducta ante el estudio, trabajo y su desarrollo principalmente como estudiantes.

URSS (Unión de Repúblicas Socialistas Soviéticas): Actualmente Eurasia después de la caída del campo socialista de este país.