

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**



**FACULTAD 8**

# **Análisis y diseño del módulo Ejercicios de la colección Multisaber.**

**Trabajo de diploma para optar por el título de  
Ingeniero en Ciencias Informáticas**

Autora: Sulay Matos Pérez.

Tutora: Ing. Greisy Gálvez George.

Co-Tutora: Ing. Celia María Soulyary.

Ciudad de la Habana, julio del 2008

“Año 50 de la Revolución”

## Declaración de Autoría

Declaro ser autora de la presente tesis “Análisis y Diseño del módulo Ejercicios de la colección Multisaber” y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_ días del mes de Julio del 2009.

Sulay Matos Pérez

---

Greisy Gálvez George

---

## **Pensamiento**

*"Nuestra libertad y su sostén cotidiano tienen color de sangre y están henchidos de sacrificio."*

*Ernesto Che Guevara de la Serna*

## **Agradecimientos**

*A mis padres, que me han dado todo el apoyo del mundo para que yo saliera adelante en mis estudios.*

*A mis hermanas que siempre han estado al tanto de mi vida.*

*A mi abuela, que tanto me ayudó en los años de carrera.*

*A mi esposo, mi cuñadita y suegros, de los que he tenido un gran apoyo en todo momento.*

*A Mailyn Cabrera y Deyaniris Depedro, quienes me ayudaron en la elaboración de la tesis siempre que lo necesité.*

*A todos mis amigos, que compartieron conmigo en la universidad, y estuvieron presentes para mí en todo momento.*

*A la profesora Greysi Gálvez y Celia María Soularý, mi tutora y cotutora, por su ayuda, disposición y preocupación.*

*A todas aquellas personas que han colaborado con el desarrollo de mi tesis.*

*A nuestra Revolución por hacer realidad mis sueños de llegar a ser una profesional.*

## Resumen

En el siguiente trabajo se muestra el resultado del análisis y diseño realizado al módulo Ejercicios de los nuevos productos de la Colección Multisaber; para dar atención al proceso educativo que se está llevando a cabo en la enseñanza primaria en Venezuela, donde los niños de las escuelas venezolanas contarán con una colección de software que incluyen imágenes, animaciones, videos, sonidos y textos, permitiéndoles una mayor interactividad con la tecnología multimedia, pues el estudiante podrá seleccionar el tema a ejercitar y dentro del mismo los ejercicios que va a realizar, y luego de responder los ejercicios solicitar: la respuesta, abordar acerca del tema, y los resultados alcanzados. Todo esto, teniendo en cuenta que esta aplicación debe ejecutarse sobre software libre y además que sea multiplataforma, se determinó usar la metodología RUP. Como lenguaje de modelado ApEM-L para el proceso de desarrollo del software y modelado de los diferentes artefactos del flujo de trabajo análisis y diseño mediante la herramienta CASE Visual Paradigm, cumpliendo así el objetivo propuesto en el trabajo.

# Índice

Introducción .....	4
Capítulo 1.Fundamentación Teórica .....	7
1.1- Introducción: .....	7
1.2- Hipertexto .....	7
1.2.1- Sus comienzos .....	7
1.2.2- Concepto de hipertexto .....	8
1.3- Multimedia .....	9
1.3.1-Historia de la multimedia .....	9
1.3.2- ¿Qué es multimedia? .....	9
1.3.3- Usos de la multimedia .....	10
1.3.4- Multimedia educativo .....	11
1.4- Hipermedia .....	12
1.4.1- ¿Qué es hipermedia? .....	12
1.4.3- Ventajas e Inconvenientes de los sistemas hipermediales .....	13
1.5- Posibles metodologías.....	15
1.5.1- Object Oriented Hypermedia Design Model (OOHDM).....	15
1.5.2- RelationShip Management Methodology (RMM).....	16
1.5.3- MultiMet .....	16
1.5.4-Análisis de la metodología a utilizar. El Proceso Unificado de Desarrollo de Software (RUP).....	17
1.6 Propuesta del Lenguaje de Modelado.....	21
1.6.1- Lenguaje Orientado a Objetos para el Modelado de Aplicaciones Multimedia (OMMMA-L). .....	21
1.6.2 - Lenguaje Unificado de Modelado (UML): .....	22
1.6.3- Object Constraint Language (OCL):.....	23
1.6.4-Análisis del lenguaje a utilizar. Lenguaje para la Modelación de Aplicaciones Educativas y Multimedia (ApEM – L).....	23
1.7-Herramientas de Modelado Visual .....	25
1.7.1-Rational Rouse .....	25
1.7.2-Herramienta CASE a utilizar: Visual Paradigm.....	26
1.8- Conclusiones Parciales.....	28
Capítulo 2.Características, Análisis y Diseño del Sistema.....	30

2.1- Introducción .....	30
2.2- Modelo del Dominio. ....	30
2.2.1- Identificación de los conceptos del dominio.....	31
2.3- Requerimientos del Sistema. ....	31
2.3.1- Requerimientos funcionales. ....	32
2.3.2- Requisitos no Funcionales. ....	33
2.4-Descripción del sistema propuesto. ....	35
2.4.1-Vistas de presentación por interfaz de comunicación con el usuario. ....	35
2.4.2- Actores del sistema. ....	35
2.4.3 Descripción textual de las vistas de presentación. ....	36
2.4.4 Vista de Gestión del Modelo.....	44
2.4.5 Diagrama de Estructura de Presentación por vista. ....	45
2.4.6 Diagrama de Estructura de Navegación. ....	49
2.4.7 Diagrama de Clases. ....	51
2.4.8 Diagrama de Interacción.....	54
2.5 Conclusiones Parciales.....	59
Capítulo 3.Estudio de Factibilidad.....	60
3.1 Introducción. ....	60
3.2 Estimación del Costo del Proyecto.....	60
3.3 Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.....	62
3.4- Análisis de costos y beneficios .....	65
3.5-Conclusiones Parciales.....	65
Conclusiones Generales. ....	66
Recomendaciones. ....	67
Glosario de términos: .....	68
Referencias Bibliográficas.....	71
Bibliografía.....	73

## Índice de Figuras

Figura 1 Representación de Hipermedia.....	13
Figura 2 Representación del ciclo de vida de RUP. ....	20
Figura 3 Lenguajes Básicos de ApEM-L. ....	24
Figura 4 Vista de la Herramienta CASE Visual Paradigm. ....	27
Figura 5 Modelo Conceptual. ....	31
Figura 6 Representación de los actores del sistema. ....	36
Figura 7 Representación de la Vista de Gestión de Modelo. ....	44
Figura 8 Diagrama de Estructura de Presentación de la vista Selección.....	46
Figura 9 Diagrama de Estructura de Presentación de la vista Ejercicios.....	47
Figura 10 Diagrama de Estructura de Presentación de la vista Respuesta. ....	48
Figura 11 Diagrama de Estructura de Presentación de la vista Resultados. ....	49
Figura 12 Diagrama de Estructura de Navegación.....	50
Figura 13 Diagrama de Clases del diseño de la vista Selección. ....	52
Figura 14 Diagrama de Clases del diseño de la vista Ejercicios.....	53
Figura 15 Diagrama de Clases del diseño de la vista Respuesta.....	53
Figura 16 Diagrama de Clases del diseño de la vista Resultados.....	54
Figura 17 Estereotipos de las clases interfaz, controladora y entidad. ....	55
Figura 18 Diagrama de secuencia de la vista Selección. ....	56
Figura 19 Diagrama de secuencia de la vista Ejercicios.....	57
Figura 20 Diagrama de secuencia de la vista Respuesta.....	58
Figura 21 Diagrama de secuencia de la vista Resultados.....	59

## Índice de Tablas

Tabla 1. Descripción Textual de la Vista de Presentación Selección.....	36
Tabla 2. Descripción Textual de la Vista de Presentación Ejercicios.....	39
Tabla 3. Descripción Textual de la Vista de Presentación Respuesta. ....	41
Tabla 4 .Consultas Externas. ....	61
Tabla 5 .Salidas Externas del módulo Ejercicios. ....	61
Tabla 6. Elementos de Información, valores y clasificación.....	62
Tabla 7. Puntos de Función. ....	62
Tabla 8. Multiplicadores de esfuerzo.....	62
Tabla 9. Factores de escala.....	62



# Introducción

La creación de software educativo se ha convertido en una tarea imprescindible para cualquier sistema educacional en la actualidad. Dado el gran auge que han tomado las tecnologías que se pueden usar para este fin, las nuevas Tecnologías de la Información y las Comunicaciones (TIC), han sido introducidas en el sistema educativo de nuestro país como parte de las transformaciones que lleva a cabo nuestra Revolución en el marco de la batalla de ideas.

El Ministerio de Educación (MINED), en colaboración con otras instituciones desarrolló la colección Multisaber. Esta colección surge a raíz de las grandes transformaciones en la educación en función de las TIC, desarrollada en el marco de la batalla de ideas, con el objetivo de resolver fundamentalmente de una manera didáctica y pedagógica, el problema de la “*integralidad*” en la enseñanza primaria. La misma consta de 32 multimedia de las diferentes materias que se imparten en este nivel. Cada una de ellas está compuesta por diferentes módulos específicos, respondiendo a las actividades que los estudiantes y profesores desarrollan a lo largo de este proceso docente.

## Situación problemática.

La hermana República Bolivariana de Venezuela, tiene especial interés en emplear software educativo de manera similar en finalidad a la experiencia cubana, ya que el proceso revolucionario que está llevando a cabo, incluye un sin número de cambios en el sector de la educación, por ello surge la necesidad de unificar todos los contenidos que deben impartir los docentes de este país. A Venezuela, como a otros países, no le es factible desde un punto de vista económico utilizar software propietario. Todo ello, debido al pago de las respectivas licencias, y por consiguiente, su política gira en torno a solo utilizar software libre.

En tal sentido se conoce que la colección Multisaber fue implementada sobre los preceptos de software propietario. Por ejemplo, utilizó como herramienta de apoyo: *Macromedia Flash 5*, *Macromedia Director*, *Tool Book* y *Borland Delphi*. Además es un software genuinamente cubano que utiliza aspectos nacionales que no son: ni identificados, ni conocidos por la población venezolana. Este problema de *contextualización*, unido a lo antes expuesto con respecto a las herramientas propietarias,

hace imprescindible la elaboración de un producto con características similares al anterior y el mismo, se debe ajustar a las exigencias educativas del proceso de aprendizaje venezolano.

Para que la creación de esta nueva colección tenga la calidad requerida y cumpla con todos los requerimientos exigidos por el cliente, es necesario desarrollar una serie de modelos que describan el software, documentando específicamente el módulo Ejercicios, uno de los módulos con que cuenta la colección Multisaber debido a su importancia en la práctica y aplicación en la actividad docente. Por lo que el **problema a resolver** se centra en ¿cómo obtener la documentación requerida en el flujo de trabajo de Análisis y Diseño del Módulo Ejercicios de los nuevos productos de la colección Multisaber?

El **objetivo general** que persigue este trabajo, consiste en: realizar el Análisis y Diseño del Módulo Ejercicios de la colección Multisaber. Y a raíz del mismo se derivaron los siguientes **objetivos específicos**:

- Seleccionar la metodología, lenguaje de modelado y herramientas más convenientes para el desarrollo de productos hipermediales.
- Identificar las funcionalidades requeridas para desarrollar el módulo Ejercicios de los nuevos productos de la Colección Multisaber.
- Realizar el Análisis del módulo Ejercicios de los nuevos productos de la Colección Multisaber.
- Realizar el Diseño del módulo Ejercicios de los nuevos productos de la Colección Multisaber.

Para dar cumplimiento a lo antes planteado se definieron las siguientes **tareas**:

- Estudio de las posibles metodologías, lenguajes de modelado y herramientas para el desarrollo de productos hipermediales.
- Estudio de los productos anteriores de la Colección Multisaber, haciendo énfasis, sólo en el Módulo Ejercicios.
- Definición de las funcionalidades del software en cuestión.
- Desarrollo de los entregables que faciliten la documentación del Módulo Ejercicios de la Colección Multisaber.

- Representación de datos, arquitectura, interfaces y componentes del Módulo Ejercicios de la Colección Multisaber.

El **objeto de estudio** lo constituye a su vez: la colección Multisaber desarrollada para Cuba, siendo el Módulo Ejercicios en específico el **campo de acción**.

En tal sentido, se **defiende la idea** de que con el análisis y diseño del módulo Ejercicios se obtendrá un producto final de mayor calidad, al poder contar los desarrolladores con todo lo necesario para implementarlo. Todo esto teniendo en cuenta, que será el grupo de desarrollo del módulo Ejercicios de la colección Multisaber, la **audiencia** más interesada.

Por tanto, este trabajo tendrá un **impacto social** muy elevado, ya que al obtener la documentación del software, se facilitará la implementación y posterior revisión del mismo por parte del grupo de calidad. Indudablemente, todo esto influye positivamente cuando se habla en estos términos, pues traerá consigo un producto final con menos errores, que optimizará el tiempo de cumplimiento de entrega del mismo.

# Capítulo 1. Fundamentación Teórica

## 1.1- Introducción

En Cuba se ha contemplado un gran auge en la incorporación de la informática al entorno educativo, dando lugar a un gran número de software que introduce el concepto de software educativo, lo cual es muy común en nuestros días. La colección Multisaber, desarrollada en nuestro país para perfeccionar el aprendizaje de los estudiantes en la enseñanza primaria es un ejemplo vivo de multimedia educativa. El propio impulso de este tipo de software ha derivado transformaciones significativas en los flujos de los procesos productivos y en los artefactos de modelación para concebir la documentación de productos hipermediales. En el presente capítulo se profundizará en los temas relacionados con los conceptos de hipertexto, multimedia, multimedia educativo e hipermedia, así como sus comienzos, aplicaciones, las ventajas y desventajas que proporcionan al proceso educativo.

## 1.2- Hipertexto

### 1.2.1- Sus comienzos

La idea original de hipertexto se debe a Vannevar Bush, cuando en 1945, en su artículo "As we may think", describe el dispositivo Memex (dispositivo para almacenar documentos) en el cual:

"...un individuo almacena sus libros, anotaciones, registros y comunicaciones, y esta colección de información es mecanizada de forma que puede ser consultada con alta velocidad y mucha flexibilidad".

Según [Bush, 1945] la característica esencial de MEMEX es su habilidad de "atar" o asociar dos ítems. En 1965, el término hipertexto es utilizado por primera vez por Ted Nelson para designar un sistema que permitía:

- ◆ Lectura no lineal.
- ◆ Escritura no lineal.

Este define hipertexto como:

"...un cuerpo de material escrito o pictórico interconectado en una forma compleja que no puede ser representado en forma conveniente haciendo uso de papel".

En el año 1960, Douglas Engelbart y Ted Nelson, desarrollaron un programa de computador que pudiera implementar las nociones de hipermedia e hipertexto. En los años ochenta, después de que

comenzaran a crearse las primeras computadoras personales, IBM lanzó el sistema de guía y enlace para sus computadoras.

El primer sistema hipertexto real fue el HES (Hypertext Editing System) construido en Brown University, en 1967, por Andries Van Dam. [1]

El hipertexto como herramienta de escritura hace posible:

- ◆ Planificar la escritura: Se pueden elaborar notas, cada una ellas con una forma y estructura propia.
- ◆ Preparar el material: Ofrece una plataforma para realizar composiciones no exclusivamente textuales.
- ◆ Componer el texto definitivo: Los enlaces permiten conectar fragmentos mediante la referencia adecuada.

### **1.2.2- Concepto de hipertexto**

En informática, es el nombre que recibe el texto que en la pantalla de una computadora conduce a su usuario a otro texto relacionado. La forma más habitual de hipertexto en documentos es la de hipervínculos o referencias cruzadas automáticas que van a otros documentos. Si el usuario selecciona un hipervínculo, hace que el programa de la computadora muestre inmediatamente el documento enlazado. [1]

Un hipertexto describe un tipo de funcionalidad de exploración en línea interactiva. Los vínculos incrustados en palabras o frases permiten al usuario escoger un texto concreto para que se muestre inmediatamente la información relacionada y el material multimedia asociado. [1]

Este consta de los siguientes elementos: nodos o secciones, enlaces o hipervínculos y anclajes. Los nodos son las partes del hipertexto que contienen información accesible para el usuario. Los enlaces son las uniones o vínculos que se establecen entre nodos y facilitan la lectura secuencial o no secuencial por los nodos del documento. Los anclajes son los puntos de activación de los enlaces. [1]

## 1.3- Multimedia.

### 1.3.1-Historia de la multimedia

La multimedia se inicia en 1984. En ese año, Apple Computer lanzó la Macintosh, la primera computadora con amplias capacidades de reproducción de sonidos equivalentes a los de un buen radio AM. Esta característica, unida a que su sistema operativo y programas, se desarrollaron en la forma que ahora se conocen como ambiente Windows, propicios para el diseño gráfico y la edición, hicieron de la Macintosh la primera posibilidad de lo que se conoce como Multimedia. También en 1968, Douglas Engelbart propone en la descripción de nls (online system) un sistema en donde no se procesan datos como números, sino ideas como texto estructurado y gráficos, dando mayor flexibilidad para manejar símbolos de manera natural. Esta propuesta condujo a Xerox a desarrollar un sistema computacional, donde se incluía al “mouse” como medio de manipulación de la información. [6]

El concepto de multimedia interactiva se empezó a asociar con las computadoras personales a partir del sistema creado por Bill Atkinson para la compañía Apple Computer llamado HyperCard. HyperCard atrajo tanto a programadores como a diseñadores, educadores, ejecutivos, músicos, productores de vídeo y todo tipo de gente creativa a desarrollar aplicaciones que reunían elementos gráficos, visuales y sonoros, en un sistema computacional controlable por el usuario. Después de que HyperCard se introdujo en el mercado en 1987, comenzaron a surgir múltiples aplicaciones desarrolladas con este paquete que incluían desde apoyos a presentaciones hasta sistemas educativos. Ya en 1992, durante la feria CES (Consumer Electronics Show) de Las Vegas, se anunció el CD multiusos. Este CD era un multiplayer interactivo capaz de reproducir sonido, animación, fotografía y video, por medio de la computadora o por vía óptica, en la pantalla de televisión. [6]

### 1.3.2- ¿Qué es multimedia?

Una multimedia no es más que la integración de al menos tres de los diversos datos o información manejados por la computadora: texto, gráficas, sonido, voz y video, que hace concurrir a diversas tecnologías: de expresión, comunicación, información, sistematización y documentación, para dar lugar a aplicaciones en la educación, la diversión y el entretenimiento, la información, la comunicación, la capacitación y la instrucción. [7]

Es un término que se aplica a cualquier objeto que usa simultáneamente diferentes formas de contenido informativo como texto, sonido, imágenes, animación y video para informar o entretener al usuario. También se puede calificar como multimedia a los medios electrónicos (u otros medios) que permiten almacenar y presentar contenido multimedia. [7]

Dentro de los materiales multimedia se encuentra la multimedia interactiva, cuando el usuario tiene cierto control sobre la presentación del contenido como: qué desea ver y cuándo desea verlo. Multimedia puede dividirse en dos categorías principales, multimedia lineal y multimedia no lineal. El contenido lineal avanza sin que el usuario tenga control sobre la navegación; una película de cine sería un ejemplo de esto. El contenido no lineal le ofrece al usuario la interactividad necesaria para controlar el progreso de la presentación tal como ocurre en los videojuegos. [7]

Existen diferentes tipos de información multimedia, como son:

- Texto: sin formatear, formateado, lineal e hipertexto.
- Gráficos: utilizados para representar esquemas, planos, dibujos lineales.
- Imágenes: son documentos formados por pixeles. Pueden generarse por copia del entorno (escaneado, fotografía digital) y tienden a ser ficheros muy voluminosos.
- Animación: presentación de un número de gráficos por segundo que genera en el observador la sensación de movimiento.
- Vídeo: Presentación de un número de imágenes por segundo, que crean en el observador la sensación de movimiento. Pueden ser sintetizadas o captadas.
- Sonido: puede ser habla, música u otros sonidos. [7]

### **1.3.3- Usos de la multimedia**

La multimedia encuentra su uso en varias áreas: arte, educación, entretenimiento, ingeniería, medicina, matemáticas, negocio, la investigación científica y en la industria del entretenimiento, para desarrollar especialmente efectos especiales en películas para la animación de los personajes de caricaturas. La multimedia puede incluir:

- Presentaciones e informaciones.
- La simulación.
- Terminales de información.
- Programas de aprendizaje.
- Juegos. [7]

El uso de los atractivos e interactivos materiales multimedia ha acontecido un gran auge en la esfera de la educación, pues favorece el aprendizaje de los estudiantes, ya que proporciona información de una manera rápida y dinámica, aviva el interés de los estudiantes pues utiliza datos o información manejados por la computadora como: gráficos, sonido, voz y video, por tanto acomoda el aprendizaje de los estudiantes con buenos gráficos, simulaciones o herramientas para el proceso de la información. El presente trabajo comprende el uso de la multimedia en la esfera de la educación.

#### **1.3.4- Multimedia educativo**

El presente trabajo por sus características se corresponde con la modelación de un software multimedia educativo. Dentro del grupo de los materiales multimedia están los que se utilizan con un propósito pedagógico. Atendiendo a su estructura, los materiales didácticos multimedia se pueden clasificar en programas tutoriales, de ejercitación, simuladores, bases de datos, constructores, programas y herramientas, presentando diversas concepciones sobre el aprendizaje y permitiendo en algunos casos (programas abiertos, lenguajes de autor) la modificación de sus contenidos y la creación de nuevas actividades de aprendizaje por parte de los profesores y los estudiantes. [2]

El software multimedia educativo puede tener diferentes funciones en el proceso de enseñanza y aprendizaje. La colección Multisaber presenta las siguientes:

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

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

**Motivadora:** La interacción con el ordenador suele resultar por sí misma motivadora. Estos programas incluyen además elementos para captar la atención de los alumnos, mantener su interés y enfocarlo hacia los aspectos más importantes.

**Evaluadora:** La posibilidad inmediata a las respuestas y acciones de los alumnos, hace adecuados los programas para evaluarles. Esta evaluación puede ser:



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

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

Innovadora: Los programas educativos pueden desempeñar esta función ya que utilizan una tecnología actual y, en general, suelen permitir muy diversas formas de uso. Esta versatilidad abre amplias posibilidades de experimentación didáctica e innovación educativa en el aula. [7]

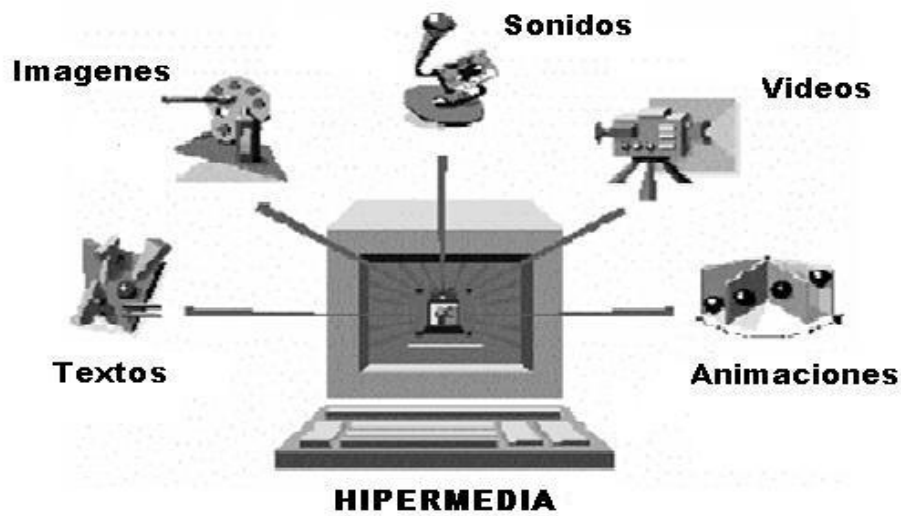
La multimedia educativo proporciona grandes ventajas al proceso educativo, debido a que cuando un software, un documento o una presentación combina adecuadamente los medios o herramientas de información manejados por la computadora, se mejora notablemente la atención, la comprensión y el aprendizaje del estudiantado, ya que se acercará algo más a la manera habitual en que los seres humanos nos comunicamos cuando empleamos varios sentidos para comprender un mismo objeto o concepto.

## **1.4- Hipermedia**

### **1.4.1- ¿Qué es hipermedia?**

Los primeros productos hipermediales reales surgen a partir de la segunda mitad de la década de los 80. El primer sistema hipermedial creado fue el Aspen Movie Map. Actualmente ejemplos de hipermedia pueden ser:

- La World Wide Web.
- Las películas almacenadas en un DVD.
- Las presentaciones en PowerPoint o en Flash, o productos informáticos similares. [9]



**Figura 1** Representación de Hipermedia.

Hipermedia conjuga los términos de hipertexto y multimedia. La hipermedia es la tecnología que nos permite estructurar la información de una manera no-secuencial, a través de nodos interconectados por enlaces. La información presentada en estos nodos podrá integrar diferentes medios. Esta se refiere a la conjunción de video, sonido, gráficos, animaciones y otros elementos para formar una asociación de temas no lineal. En lugar de forzar al usuario a desplazarse por la información de forma secuencial, la hipermedia permite simular los procesos humanos, saltando de un tema a otro por medio de asociaciones. Así, el usuario puede ir encontrando la información según sus propias asociaciones en lugar de tener que trabajar linealmente. [14]

### **1.4.3- Ventajas e Inconvenientes de los sistemas hipermediales**

La tecnología en función de la educación está posibilitando una nueva forma de efectuar los procesos de enseñanza-aprendizaje, la enseñanza a distancia, muy de moda en estos días, está sustituyendo la presencia de los educadores, sin embargo se persigue la misma finalidad, transmitir conocimientos a los estudiantes. A través de los productos hipermediales se desarrollan un gran número de actividades que originalmente eran llevadas a cabo por los educadores, es a su vez la pizarra, los alumnos, las sillas, las mesas y el contenido de la clase, por lo que gradualmente ha permitido ir cambiando los hábitos de aprender.

Hoy por hoy la educación a distancia es asistida por estos medios permitiendo cuestionar qué tanto pueden sustituir la presencia de un profesor, y en general, un aula de clases. Sin embargo, la combinación de medios (Multimedia) tiene como uno de sus objetivos principales no sustituir al profesor, sino enriquecer la enseñanza y el aprendizaje ya que la educación actual plantea al alumno como el eje de los procesos educativos. La multimedia tiene la función de proporcionarle al estudiante la información necesaria, a través de una presentación atractiva con el propósito de obtener un aprendizaje significativo.

Los productos hipermediales tienen en la actualidad gran utilización, debido a las disímiles ventajas que presentan, ejemplo de ello se tiene que:

1. Los textos hipermediales no son lineales. Los lectores pueden explorar la información como lo deseen. Esto significa, a su vez, que un mismo documento puede servir a públicos diferentes.
2. Tienen interfaces de usuario muy intuitivas.
3. El usuario no necesita realizar grandes esfuerzos para conseguir rápidamente resultados.
4. La información se recupera sencillamente, aunque distintos usuarios estén utilizando el mismo documento simultáneamente.
5. Permiten representar información poco o nada estructurada.
6. La creación de nuevas referencias es inmediata, independientemente del tipo de contenido involucrado.
7. Los usuarios pueden hacer crecer su hiperdocumento o simplemente anotarlo, sin cambiar por ello el documento referenciado.
8. Posibilitan la estructuración de la información.
9. Facilitan la modularidad y la consistencia de la información. [\[11\]](#)

Constituyen un marco idóneo para la autoría en colaboración, al permitir la compartición, distribución y personalización de la información.

- El tamaño excesivamente grande de algunos sistemas y su estructura, hace que en muchas ocasiones el lector sea incapaz de localizar aquello que está buscando.
- Menor velocidad de respuesta.
- Control real de datos postdatado.
- No es secuencial, depende de quien es el lector.

- El acceso depende del interés del lector, es totalmente variable.
- La desorientación y los problemas de sobrecarga de conocimiento constituyen los dos inconvenientes básicos en la utilización de este tipo de tecnología. [11]

## **1.5- Posibles metodologías:**

Existe un grupo de metodologías dirigidas al trabajo con productos del tipo multimedia e hipermedia con la finalidad de lograr que cualquier equipo de desarrollo logre un mejor entendimiento entre sí para generar todos los artefactos necesarios a la hora de analizar, diseñar e implementar un software de este tipo, con la calidad requerida y especialmente de mejorar las estructuras de navegación, haciéndolas más intuitivas y que al final el software tenga una de las características fundamentales de estos sistemas, su interactividad con el usuario. Algunas de las metodologías más usadas en productos multimedia e hipermedia lo constituyen sin lugar a dudas: RMM, OOHD, MultiMet y RUP.

### **1.5.1- Object Oriented Hypermedia Design Model (OOHDM)**

La metodología OOHDM fue concebida por D. Schwabe y G. Rossi y planteaba que el desarrollo de cualquier Hiperdocumento en general consta de cuatro fases en las que se combinan diferentes estilos de desarrollo como el incremental, iterativo y prototipado, en las tres primeras etapas. Las tres primeras fases son de diseño, en las que se obtiene un conjunto de modelos orientados a objeto que describen el sistema que será construido en la cuarta fase. [8]

OOHDM propone el diseño conceptual o análisis de dominio que utiliza el método de análisis orientado a objetos para obtener esquemas conceptuales de las clases. Realiza el diseño navegacional expresado también con un enfoque orientado a objetos a través de dos tipos de modelos, contempla una tercera fase denominada diseño de la interface abstracta, en la que se realiza un modelo, también orientado a objetos, para especificar la estructura y el comportamiento de la interface del sistema hipermedia con el usuario. Por último la fase de implementación construye una aplicación completamente orientada a objetos. [8]

Esta metodología tiene como desventaja que requiere de cierta sobrecarga para aprenderla, debido a los modelos que utiliza. Sin embargo, OOHDM contempla todos los aspectos que debemos modelar,

es completa para nuestro caso, los modelos subyacentes son ricos en representación estructural y semántica, más su utilización sin una herramienta CASE se hace compleja a medida que crece el tamaño y la complejidad de la aplicación a modelar.

### **1.5.2- Relationship Management Methodology (RMM):**

RMM es una metodología basada en los conceptos del modelo de diseño de Hipertexto (HDM) es decir, en entidades y tipos de entidades. Su objetivo es mejorar la navegación a través de un análisis de las entidades del sistema. Incorpora el concepto de slice como agrupación de datos de varias pantallas en una entidad. Es la primera metodología que se publica completa para la creación de un software multimedia. Su problema principal es que no permite realizar consultas a partir de dos entidades por su ligadura al modelo entidad relación, obligando a la descomposición de relaciones uno a muchos, no obstante muestra su fortaleza en los procesos de análisis y diseño para multimedia. [3]

El modelo no permite significar propiedades semánticas del dominio como por ejemplo el hecho de que los contenidos pertenecen a un tópico y de que la existencia de los primeros depende de la existencia del segundo. Presenta una desventaja, el hecho de tener que indicar las estructuras de acceso juntas con el esquema navegacional, ya que con esta especificación conjunta se ligan fuertemente los aspectos de navegación y de acceso a las estructuras, lo que las hace muy dependientes. La metodología es un poco restrictiva en el sentido que no se pueden incluir atributos de distintas entidades del dominio en una unidad de visualización. Finalmente, cabe destacar que RMM no provee herramientas conceptuales para el diseño de la interfaz. [3]

### **1.5.3- MultiMet.**

MultiMet es una metodología que describe las etapas generales de organización de un proyecto informático desarrollado con tecnología multimedia. Va a iniciar su desarrollo con un estudio de las necesidades que presentan los usuarios, los objetivos, las tecnologías y el personal de desarrollo. Se realiza un estudio detallado para determinar si es factible o no desarrollar el producto para no emplear tiempo y recursos en vano. Esta metodología de diseño nacional presenta una etapa donde se va a definir si el software es educativo, informativo o demostrativo, identificando también la audiencia final del producto. [4]

En el paso de implementación es donde se va a definir qué herramienta de autor se va a utilizar y se comienza entonces el montaje del software. La fase de prueba va a garantizar una revisión basada en la solidez de la información y el funcionamiento adecuado del producto. [4]

MultiMet no se centra en la especificación de la estructura al nivel de programación, llevando a un nivel elemental el análisis y diseño; fuerza al programador a llevar adelante el desarrollo de módulos que pueden resultar muy complejos y carece de herramientas de sostén para la descripción del proceso de implementación.

#### **1.5.4-Análisis de la metodología a utilizar. El Proceso Unificado de Desarrollo de Software (RUP):**

##### **Un poco de historia.**

Los orígenes de RUP se remontan al modelo espiral original de Barry Boehm. Ken Hartman, uno de los contribuidores claves de RUP colaboró con Boehm en la investigación. En 1995 Rational Software compró una compañía sueca llamada Objectory AB, fundada por Ivar Jacobson, famoso por haber incorporado los casos de uso a los métodos de desarrollo orientados a objetos. El Rational Unified Process fue el resultado de una convergencia de Rational Approach y Objectory (el proceso de la empresa Objectory AB). El primer resultado de esta fusión fue el Rational Objectory Process, la primera versión de RUP, fue puesta en el mercado en 1998, siendo el arquitecto en jefe Philippe Kruchten. [13]

##### **Características que hacen que el Proceso Unificado sea único:**

El Proceso Unificado de Desarrollo o RUP como su nombre bien indica, es un proceso de desarrollo de software. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado. Según los especialistas existen tres características que hacen que el Proceso Unificado sea único: iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura:

- **Iterativo e incremental:** El desarrollo de un producto software puede ser muy complejo e incluso requerir mucho tiempo en su desarrollo por lo que es muy efectivo dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos,

al crecimiento del producto. En cada iteración los desarrolladores seleccionan un grupo de casos de uso los cuales se analizan, diseñan, implementan y prueban, terminando los casos de uso que se desarrollan en dicha iteración en código ejecutable. La iteración controlada hace posible que se reconozca una realidad que a menudo es ignorada y es que las necesidades del usuario y sus correspondientes requisitos no pueden definirse completamente al principio.

- **Dirigido por casos de uso:** El objetivo de cada sistema es brindar algún tipo de servicio a los usuarios. RUP por su parte refleja lo que los usuarios necesitan y desean mediante los casos de uso. Las necesidades reales se captan cuando se modela el negocio y se representa a través de los requerimientos dichas necesidades. Los casos de uso se especifican, se diseñan y es a partir de los casos de uso finales que los ingenieros de prueba construyen sus casos de prueba.
- **Centrado en la arquitectura:** La arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción. Mediante la arquitectura se describen los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. En la arquitectura influyen factores como la plataforma en la que el software tiene que funcionar, el hardware, sistema operativo, sistema de gestión de base de datos y los protocolos para comunicaciones en red. La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo. [13]

Entre las principales características que distinguen a RUP se tienen:

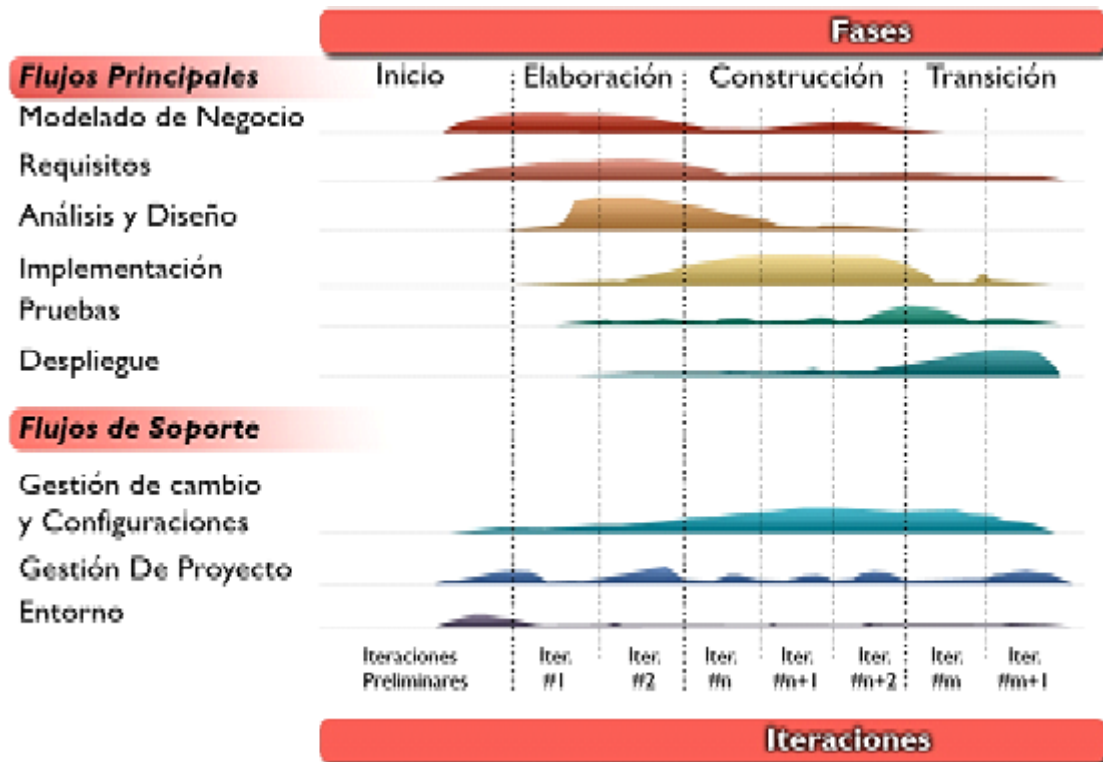
- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software. [13]

## Principios de Desarrollo de RUP.

RUP está basado en 5 principios claves que son:

- Adaptar el proceso: El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.
- Balancear prioridades: Los requerimientos de los diversos inversores pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.
- Demostrar valor iterativamente: Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados
- Elevar el nivel de abstracción: Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o esquemas (frameworks) por nombrar algunos. Esto previene a los ingenieros de software ir directamente de los requisitos a la codificación de software a la medida del cliente. Un nivel alto de abstracción también permite discusiones sobre diversos niveles arquitectónicos. Estos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML.
- Enfocarse en la calidad: El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. [13]





**Figura 2** Representación del ciclo de vida de RUP.

El ciclo de vida RUP es una implementación del desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas y organiza las tareas en fases e iteraciones.

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada uno de ellos, estos a la vez se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante:

- **Concepción:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo, 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.

- **Mantenimiento:** una vez instalado el producto, el usuario realiza requerimientos de ajuste, esto se hace de acuerdo a solicitudes generadas como consecuencia del interactuar con el producto. [5]

En el presente trabajo se utilizará RUP como metodología para el desarrollo de la documentación del módulo Ejercicios de la colección Multisaber, ya que pretende implementar las mejores prácticas en Ingeniería de Software, pues el Proceso Unificado de Desarrollo de Software se caracteriza por estar centrado en la arquitectura, que es donde se describen los elementos de modelado que son de mayor importancia para la construcción del módulo, además permite el modelado visual del software y la verificación de la calidad del software y control de cambios.

## **1.6 Propuesta del Lenguaje de Modelado.**

Para la modelación de software educativo existen disímiles tipos de lenguajes de modelado que permiten modelar los artefactos correspondientes a un software educativo. A continuación se realizará un estudio de determinados lenguajes de modelado concebidos con esta finalidad, así como sus características, ventajas y desventajas, posibilitando seleccionar el más conveniente para el desarrollo de la documentación requerida en el flujo de trabajo de análisis y diseño del módulo Ejercicios de la colección Multisaber.

### **1.6.1- Lenguaje Orientado a Objetos para el Modelado de Aplicaciones Multimedia (OMMMA-L).**

OMMMA-L permite modelar todos los aspectos de una aplicación desarrollada con tecnología multimedia e hipermedia y se lanza como una propuesta de extensión de UML para la integración de especificaciones de sistemas desarrollados con tecnología multimedia e hipermedia basados en el paradigma orientado a objetos, y MVC (Modelo Vista Controlador) para la interfaz de usuario. Este patrón distingue un componente modelo sosteniendo la funcionalidad del núcleo y los datos, un componente vista para mostrar la información al usuario y componente controlador para manipular los eventos de interacción. También representa la vista de presentación espacial que se modela a través de los Diagramas de Presentación de OMMMA-L. Estos diagramas son de nueva aparición en la extensión de UML y 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, scroll, hipertextos con hipervínculos). [15]

OMMMA-L es tomado por el lenguaje de modelado ApEM-L como lenguaje base, tomando los elementos más significativos del mismo, para la modelación de la programación orientada a objetos y el patrón arquitectónico MVC. Sin embargo no representa las modificaciones descriptivas y decorativas para un producto multimedia.

### **1.6.2 - Lenguaje Unificado de Modelado (UML):**

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.[6]

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

Aunque UML es uno de los lenguajes de modelación más utilizados, y de gran eficacia, no será utilizado en la realización del módulo Ejercicios ya que no permite representar de una aplicación educativa las modificaciones descriptivas y decorativas en la representación de los componentes del lenguaje base. UML no soporta todos los aspectos de las aplicaciones multimedia de una manera adecuada e intuitiva. Las características del lenguaje para el modelado de aspectos de la interfaz de usuario no están explícitamente proporcionadas. [2]

### **1.6.3- Object Constraint Language (OCL):**

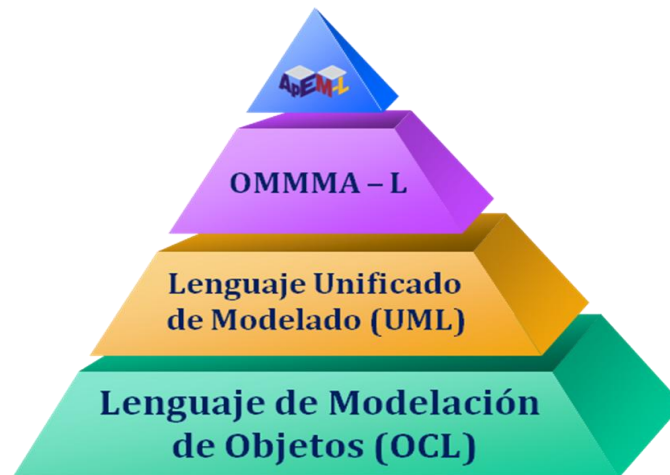
Fue adoptado en octubre de 2003 por el grupo OMG como parte de UML 2.0. OCL es un lenguaje para la descripción formal de expresiones en los modelos UML. Sus expresiones pueden representar invariantes, precondiciones, post-condiciones, inicializaciones, guardias, reglas de derivación, así como consultas a objetos para determinar sus condiciones de estado. Se trata de un lenguaje sin efectos de borde, de manera que la verificación de una condición, que se presupone una operación instantánea, nunca altera los objetos del modelo. Su papel principal es el de completar los diferentes artefactos de la notación UML con requerimientos formalmente expresados. [10]

OCL es tomado por el lenguaje de modelado ApEM-L para la modelación de la programación orientada a objetos. Aunque se trata de un lenguaje de especificación ya adoptado para UML 2.0, falta todavía definir los detalles de su utilización en los diferentes artefactos que conformarán la notación UML 2.0.

### **1.6.4-Análisis del lenguaje a utilizar. Lenguaje para la Modelación de Aplicaciones Educativas y Multimedia (ApEM – L):**

ApEM – L está basado en el lenguaje de modelación UML, tomando elementos representativos y extensiones del mismo; descansa toda su estructura sobre los elementos planteados por el estándar OCL, en su versión actualizada 2.0 del 2003. Se presenta como una extensión de UML, tomando como bases teóricas principales OMMMA – L y OCL – 2.0, lo que produce las siguientes ventajas:

- Puede utilizar para su representación todas las herramientas CASE que existen actualmente para la modelación de UML.
- Es un lenguaje que utiliza el estándar internacional OCL, para la modelación de la programación Orientada a Objetos.
- No modifica la semántica del lenguaje base UML, sino que trabaja en estereotipos restrictivos, por lo que a su vez produce modificaciones descriptivas y decorativas en la representación de los componentes del lenguaje base. [2]



**Figura 3** Lenguajes Básicos de ApEM-L.

ApEM-L tiene entre sus principales objetivos:

1. 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 UCI.
2. 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.
3. No complicar ApEM – L con elementos que lo convirtieran o abarcaran un método de desarrollo de aplicaciones educativas, sino sólo el área de la representación y la documentación de este tipo de aplicaciones.
4. 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. [2]

ApEM-L es el lenguaje que más se ajusta al presente trabajo, ya que establece el patrón arquitectónico MVC-E para la concepción del diseño de las aplicaciones educativas. Además plantea la semántica y los estereotipos restrictivos y descriptivos para las clases asociadas a las tecnologías multimedia e hipermedia y organiza la estructura del diagrama en secciones para la representación lógica de los distintos tipos de clases, incorporando las clases abstractas del modelo conceptual.

## 1.7-Herramientas de Modelado Visual

Para el modelado visual se utilizan las herramientas CASE, necesarias para la automatización de procesos de análisis, diseño e implementación, facilitando así la generación de los artefactos necesarios que permiten obtener un producto de mayor eficiencia. A continuación se realizará un estudio para determinar la herramienta CASE a utilizar para el modelado visual del módulo Ejercicios de la colección Multisaber en los flujos de trabajo de análisis y diseño.

### 1.7.1-Rational Rose

La primera herramienta CASE que salió a la luz en el año 1984 fue Excelerator y trabajaba bajo una plataforma PC. Estas alcanzan la cúspide a partir de los años 90.

Rational Rose es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases. Además, es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++® y Visual Basic®. Como todos los demás productos Rational Rose, proporcionan un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.[12]

Rational Rouse presenta características adicionales que lo distingue como son:

- El soporte para análisis de patrones ANSI, C++, Rose J y Visual C++ basado en "Diseño de Patrones: Elementos para reutilización de Software Orientado a Objetos".

- Característica de control por separado de los componentes de modelo que permite una mejor administración y el uso de modelos.
- La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- Soporte Enterprise Java Beans™ 2.0.
- Capacidad de análisis de calidad de código.
- El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web.
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación
- Integración con otras herramientas de desarrollo de Rational.
- Publicación Web y generación de informes para optimizar la comunicación dentro del equipo.

[12]

A pesar de que Rational Rose es una herramienta CASE muy potente, no es la más conveniente para el desarrollo del presente trabajo, pues no es multiplataforma, ya que solamente corre sobre el sistema operativo Windows. Al mismo tiempo no es una herramienta libre, y la universidad "UCI" no paga su licencia.

### **1.7.2-Herramienta CASE a utilizar: Visual Paradigm**

En el presente trabajo se utilizará Visual Paradigm como Herramienta CASE, ya que la misma da soporte al modelado visual con UML 2.0, incluye un gran número de estereotipos posibilitando la creación de diagramas de fácil entendimiento, además tiene disponibilidad en múltiples plataformas, múltiples versiones, y la universidad "UCI" posee su licencia.

Es una herramienta amigable para el usuario, puede ser usada en varios idiomas y cada componente utilizado en el diagrama que se esté creando, sugiere nuevos posibles componentes a utilizar, por lo que ya no es necesario localizarlos en la barra y así se crea fácilmente cualquier tipo de diagrama. Tiene un amplio número de estereotipos que proporciona la creación de diagramas de fácil entendimiento, además de que estos diagramas te los organiza automáticamente. Se puede obtener un Diagrama de Clases del Diseño a partir de un Diagrama de Entidad Relación y viceversa. Brinda la posibilidad de documentar todo el trabajo, en formato PDF o en forma de un sitio WEB, sin necesidad

de utilizar herramientas externas. Una vez que los diagramas han sido confeccionados, es posible generar documentos que combinan textos e imágenes obtenidas a partir de los diagramas de manera muy sencilla, incluso Visual Paradigm genera toda la documentación de lo que se ha hecho hasta el momento, cumpliendo con estándares establecidos. [4]

Visual Paradigm ofrece:

- Entorno de creación de diagramas para UML 2.0
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas. [4]

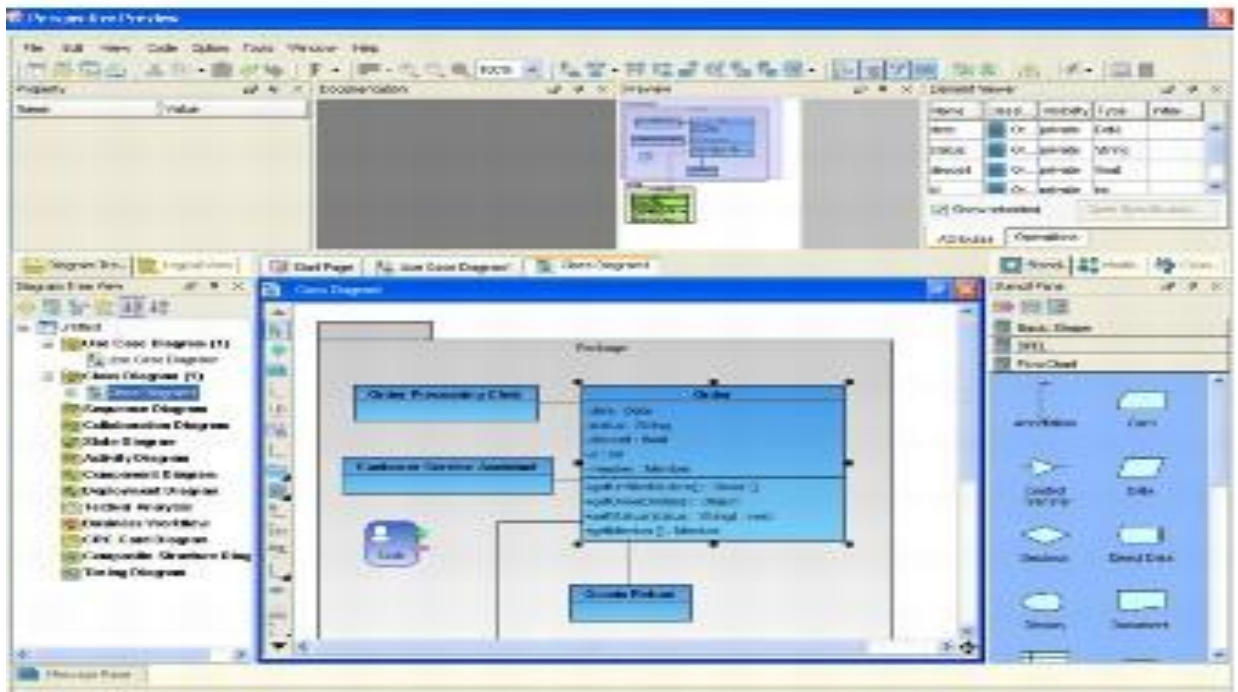


Figura 4 Vista de la Herramienta CASE Visual Paradigm.



Los diagramas en Visual Paradigm presentan:

- Representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo con vértices conectados por arcos.
- Diferentes tipos de diagrama permiten ver el sistema desde diferentes perspectivas
- Los tipos de diagramas que se pueden incluir son:
  - Componentes
  - Despliegue
  - Secuencia
  - Casos de Uso
  - Clase
  - Actividad
  - Estado [4]

## **1.8- Conclusiones Parciales:**

En este capítulo profundizamos en el conocimiento de algunos conceptos necesarios para la comprensión de este trabajo como son Multimedia, Hipertexto e Hipermedia, haciendo énfasis en los productos hipermediales ya que la manera de recorrer la información se hace de una forma no lineal, la interfaz por lo general llega a ser muy intuitiva y los usuarios, en este caso los estudiantes, no necesitan realizar grandes esfuerzos para conseguir rápidamente resultados.

ApEM-L será el lenguaje que se empleará en el Análisis y Diseño del Módulo Ejercicios de la colección Multisaber ya que este permite de una manera rápida y eficiente generar todos los artefactos necesarios para la obtención de la documentación que lleva el módulo en cuestión, pues facilita la modelación de multimedia educativas, incorporando los estereotipos restrictivos.

Se propone la metodología RUP, ya que implementa las mejores prácticas de Ingeniería de Software, para desarrollar el flujo de trabajo de Análisis y Diseño del Módulo Ejercicios de la colección Multisaber, además lleva a cabo un fuerte control de cambios, lo que es muy significativo para el desarrollo de grandes proyectos permitiendo así la verificación de la calidad del software.

Visual Paradigm es la Herramienta CASE que da soporte al modelado visual con UML 2.0 y el lenguaje de modelado a utilizar (ApEM-L) puede utilizar para su representación todas las herramientas CASE que existen actualmente para la modelación de UML. Además, es multiplataforma y posee un amplio número de estereotipos que facilita la representación de los diagramas de mayor entendimiento.

# Capítulo 2. Características, Análisis y Diseño del Sistema.

## 2.1- Introducción

El presente capítulo se dedica a la descripción de la solución propuesta para lograr nuestro objetivo general de “Realizar el Análisis y Diseño del módulo Ejercicios de la colección Multisaber”. Por tanto, se definirán las entidades principales relacionadas con el Modelo del Dominio, para concretar conceptos fundamentales presentes en la aplicación, requisitos funcionales, no funcionales del sistema, y la descripción de las diferentes pantallas que presenta el software. Se representará la Vista de Gestión del Modelo, los diagramas de Estructura de Presentación y los de Estructura de Navegación, los cuales permiten describir cómo interactúan las clases que posibilitan la navegación y cómo serán en términos visuales los elementos que componen cada una de las vistas. Se realizará un diagrama de clase por cada vista que establece el patrón arquitectónico MVC-E para la concepción del diseño de las aplicaciones educativas, y además se representarán los diagramas de interacción que representan la manera en la que los objetos de la aplicación intercambian mensajes para darle cumplimiento a sus responsabilidades.

## 2.2- Modelo del Dominio

Considerando que para modelar un sistema con tecnología multimedia no es necesario definir los procesos de negocio sino los conceptos relacionados con el mismo, se plantea la elaboración de un modelo de dominio que represente los tipos más importantes de objetos en el sistema que se modelará.

El modelo del dominio representa los conceptos (objetos) más importantes en el contexto del sistema, donde los objetos son entidades que existen en el mundo real y no de los componentes de software, además tienen identidad y son distinguibles entre sí. El modelo de dominio contiene clases de objetos, las que a su vez incluyen asociaciones entre sí y cada una posee atributos pero no métodos, ya que en el modelo de dominio no se muestra comportamiento.

La representación con UML del modelo de dominio es a través de diagramas de clases, que muestra a clientes, usuarios, revisores y desarrolladores cómo se relacionan las clases del dominio.

### 2.2.1- Identificación de los conceptos del dominio

**Tema:** Se llama el contenido que se va a ejercitar.

**Ejercicio:** El (los) ejercicios a realizar por el estudiante de un tema seleccionado.

**Maestro:** Es la persona que indica a cada estudiante qué ejercicio (ejercicios) realizará.

**Estudiante:** Es la persona que realiza los ejercicios de un tema seleccionado para aprender o ejercitar un contenido.

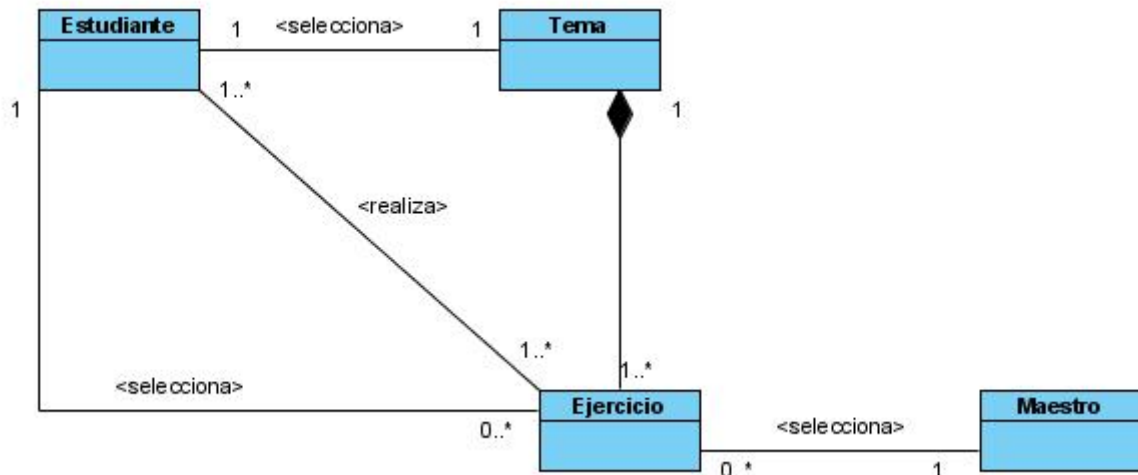


Figura 5 Modelo Conceptual.

### 2.3- Requerimientos del Sistema

El propósito del flujo de trabajo de Requisitos es regir a los desarrolladores a construir un sistema correcto. Mediante la descripción de los requisitos del sistema, que no es más que las condiciones o capacidades que debe cumplir el mismo, se logra un acuerdo entre el cliente (incluyendo usuarios) y los desarrolladores sobre qué debe cumplir el sistema, así como las restricciones bajo las cuales debe operar. Por tanto, para el levantamiento de requisitos debemos utilizar el lenguaje del cliente, ya que este debe ser capaz de comprender el resultado de la captura de requisitos. Alcanzar un entendimiento entre los clientes que solicitan el software y el equipo de desarrollo del mismo, es la clave para lograr

el triunfo en la producción de un software, y por esta razón es indispensable la correcta aplicación de la Ingeniería de Requerimientos de Software. Para lograr lo antes expuesto, en este flujo de trabajo se siguen conjuntamente los siguientes pasos:

- Enumerar los requisitos candidatos.
- Comprender el contexto del sistema.
- Capturar requisitos funcionales.
- Capturar requisitos no funcionales.

Los requerimientos pueden clasificarse en dos grandes grupos: funcionales y no funcionales.

**Requisitos Funcionales:** Concretan las condiciones que el sistema debe cumplir, o sea, las funciones que debe realizar para proporcionar un determinado resultado al usuario.

**Requisitos no Funcionales:** Responden a las cualidades o propiedades que el sistema debe cumplir, es decir, son las características que permiten obtener un producto atractivo, usable, rápido o confiable.

### **2.3.1- Requerimientos funcionales.**

**RF 1:** Cargar presentación del módulo Ejercicios.

**RF 1.1:** Seleccionar el tema y dentro del mismo, los ejercicios a realizar.

**RF 1.2:** Mostrar la introducción del ejercicio, la pregunta, una imagen (opcional), el área de respuesta y forma de operar.

**RF 1.3:** Mostrar el porcentaje de efectividad, temporizador, evaluación (B, R, M) y el número del ejercicio que se está realizando.

**RF 1.4:** Revisar respuesta del ejercicio.

**RF 1.4.1:** Al responder correctamente se emitirá un sonido de confirmación y se pasará a la pregunta siguiente.

**RF 1.4.2:** Mostrar respuesta correcta (si el estudiante falla se muestra un mensaje con la respuesta correcta).

**RF 1.4.3:** Mostrar mensaje de premio (al estudiante responder un número determinado de preguntas correctamente se le mostrará un mensaje con algún premio y una locución de la mascota con felicitaciones).

**RF 1.5:** Se mostrará una ventana al estudiante con las notas obtenidas, los puntos alcanzados y el tiempo que tardó para responder el conjunto de ejercicios y la posibilidad de continuar o cancelar.

**RF 2:** Mostrar botón de ayuda, donde explica toda la funcionalidad de la pantalla en curso.

**RF 3:** Mostrar botón: Poner/Quitar Música.

**RF 4:** Mostrar botón: Página Principal, que permite ir a la página principal.

**RF 5:** Mostrar botón: Salir.

**RF 5.1:** Al usuario oprimir este botón se le debe mostrar un mensaje de confirmación.

### 2.3.2- Requisitos no Funcionales

➤ **Requerimientos de Software.**

Sistema Operativo Linux: Ubuntu 7.10

➤ **Requerimientos de Hardware.**

Se necesita como mínimo un procesador PENTIUM 233 MHz con tarjeta de sonido estándar de 11.025 Khz. y lector de CD. La resolución de pantalla debe ser de 800x600 y 24 bits en lo adelante. Memoria de 64 MB RAM (*Recomendado:* 128 MB RAM o mejor). Disco duro de 1GB.

➤ **Restricciones en el diseño y la implementación.**

El lenguaje de programación será Java Script. La herramienta de desarrollo de la aplicación será el CMS Joomla!

El texto de los menús o identificadores del módulo serán para los botones de servicios de color negro con iluminación blanca, además de un fondo blanco; para los identificadores de color blanco con un fondo azul.

Los botones de servicios (Regresar a la página principal, Ayuda, Buscar, Acerca de..., Activar/Desactivar música, Salir) se mostrarán siempre visibles al usuario.

Las medias a visualizar siempre se mostrarán en la misma área de interfaz, para evitar la pérdida del usuario.

El idioma utilizado será el español, apoyándose en el uso de las palabras técnicas propias del tema en cuestión en la aplicación.

El ícono identificador de los botones de servicios serán de color verde y blanco con una estrellita amarilla en el centro, con iluminación blanca, además de mostrar el nombre del servicio y la estrellita moverse al pasarle por encima con el Mouse.

➤ **Requerimientos de Seguridad.**

Solo el gobierno de Venezuela está autorizado a la distribución del software.

➤ **Requerimientos de Usabilidad.**

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

➤ **Requerimientos de Soporte.**

La máquina donde se ejecutará la aplicación deberá disponer de una tarjeta de sonido, además de los dispositivos necesarios para la reproducción del mismo.

➤ **Requerimientos de Rendimiento.**

- El tiempo de verificación de las respuestas, en los ejercicios que lo requieran, no deberá exceder los 5s.
- El tiempo de visualización de las medias no deberá exceder los 5s.
- El tiempo de ejecución de un hipervínculo no deberá exceder los 5s.

➤ **Requerimientos de Portabilidad.**

El software podrá ser ejecutado bajo los sistemas operativos SO Windows 9x, Me, NT, XP; Mac OS 9 o superior y Linux en sus distintas versiones, según decreto en el artículo 3.390 “todos los desarrollos deben seguir los estándares abiertos y los códigos abiertos”, debiendo existir en el SO Linux el emulador para aplicaciones multimedia requerido para la ejecución de este tipo de aplicaciones.

➤ **Requerimientos de Ayuda y documentación en línea.**

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

➤ **Requerimientos Políticos-Culturales.**

La aplicación responderá a las líneas de políticas a las que se adscribe el proyecto MULTISABER y que fueron dictaminadas por el Ministerio de Ciencia y Tecnología.

## 2.4-Descripción del sistema propuesto

### 2.4.1-Vistas de presentación por interfaz de comunicación con el usuario

Una **vista de presentación** es una descripción lógica de un fragmento de funcionalidad del sistema que expresa una sucesión de mensajes intercambiados entre esta fracción y los actores del sistema. La vista de presentación representa un área de la interfaz gráfica de usuario, y contiene elementos gráficos e interactivos como son: imágenes, botones, animaciones, textos, entre otros. Por tanto, una interfaz de usuario estará constituida por varias vistas de presentación, y una vista de presentación a su vez puede contener en sí otras vistas de presentación, por lo que toda interfaz gráfica de usuario es una vista de presentación. [2]

Una interfaz de comunicación con el usuario es la forma en que los usuarios se comunican con la computadora, por lo que comprende todos los puntos de contacto entre el usuario y el cliente.

- Interfaz de comunicación con el usuario Selección.
  - Vista Selección.
  - Vista Ejercicios.
- Interfaz de comunicación con el usuario Ejercicios.
  - Vista Ejercicios.
  - Vista Respuesta.
- Interfaz de comunicación con el usuario Resultados.
  1. Vista Resultados.

### 2.4.2- Actores del sistema

**Usuario:** Es la generalización de los actores: estudiante y maestro, que realiza las actividades comunes a ellos.



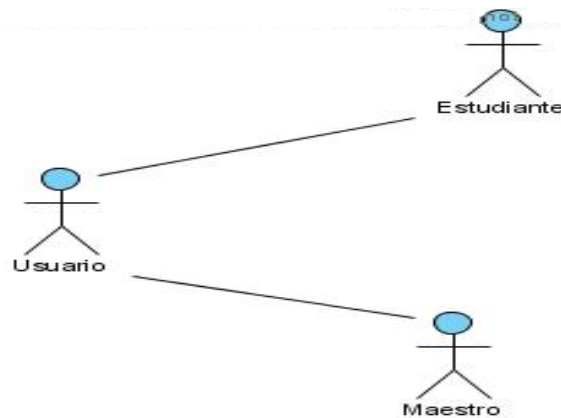


Figura 6 Representación de los actores del sistema.

### 2.4.3 Descripción textual de las vistas de presentación

La descripción textual de la vista de presentación permite expresar y describir cómo ocurre el flujo de acciones de la interacción del usuario con el sistema, así como las características esenciales de la vista. Esta descripción textual de una vista de presentación es solo de una vista de presentación a otra vista.

Tabla 1. Descripción Textual de la Vista de Presentación Selección.

Descripción Textual de la vista de Presentación Selección.	
<b>Actores de la vista de presentación</b>	<b>Usuario</b>
<b>Propósito</b>	Permitir al usuario de la aplicación seleccionar el Tema a ejercitar, y cuántos o cuáles ejercicios va a realizar.
<b>Objetivos pedagógicos</b>	Presentarle al usuario los diferentes temas que se deben ejercitar para el aprendizaje del estudiante, así como los ejercicios que incluye cada tema para desarrollar.
<b>Resumen</b>	La vista tiene inicio cuando se carga el módulo Ejercicios, y se muestran los textos que representan los temas que el estudiante puede ejercitar, y dentro del tema la cantidad y cuáles ejercicios el mismo puede realizar. La vista termina con la selección del usuario, del tema y los ejercicios a realizar.



<p>7. El usuario selecciona una de las opciones que permite determinar cuántos o cuáles ejercicios va a realizar.</p> <p>8. El usuario selecciona la opción de comenzar.</p>	<p>9. Se ejecuta la vista de "Ejercicios", y culmina la vista.</p>	<p>estudiante es el que decide; la mascota decide, a partir de una cantidad especificada por el usuario se escogen los ejercicios de forma aleatoria; el maestro decide, el maestro selecciona el estudiante y cuáles ejercicios son los que debe realizar.</p>	
<b>Curso alternativo de los eventos</b>			
<b>Acción</b>	<b>Curso Alterno</b>		
-	-		
<b>Prioridad</b>	Crítica		
<b>Mejoras</b>	-		
<b>Medias a utilizar</b>	<b>Tipo de Media</b>	<b>Descripción</b>	<b>Estado</b>
	<b>Imagen</b>	-	-
	<b>Video o animación</b>	-	-
	<b>Sonido</b>	Canción para estimular la selección del módulo por parte del usuario.	En localización.
	<b>Texto</b>	Textos que describen cada uno de los temas a	Existente

		ejercitar por el estudiante.	
<b>Reglas Pedagógicas</b>	-		
<b>Representación de la vista</b>	-		

**Tabla2. Descripción Textual de la Vista de Presentación Ejercicios.**

<b>Descripción Textual de la vista Ejercicios</b>	
<b>Actores de la vista de presentación Ejercicios</b>	Estudiante.
<b>Propósito</b>	Permitir al estudiante desarrollar los ejercicios en su interfaz de comunicación con el usuario.
<b>Objetivos pedagógicos</b>	Presentarle al estudiante situaciones problemáticas las cuales debe resolver a través de los conocimientos que posee para ejercitarlos.
<b>Resumen</b>	La vista tiene inicio cuando el usuario ha seleccionado el tema y los ejercicios a realizar y da clic en el botón comenzar (elemento interactivo) se muestra una interfaz de comunicación con el usuario mostrando el enunciado del primer ejercicio, donde el estudiante da respuesta al mismo de forma interactiva. De esta manera, mediante las opciones anterior o siguiente (elementos interactivos) se desplaza para culminar la realización de los ejercicios correspondientes, solicita revisión al sistema dando clic en el botón Revisar (elemento interactivo) y termina la vista.
<b>Vistas asociadas</b>	Vista"Respuesta".
<b>Referencias</b>	RF1.2, RF1.3, RF1.4, RF2, RF3, RF4, RF5, RF5 y RF5.1
<b>Precondiciones</b>	Debe haberse ejecutado la vista " <i>Selección</i> ".
<b>Poscondiciones</b>	Se muestra la vista" <i>Respuesta</i> ".
<b>Curso Normal de los eventos</b>	

<b>Acción del actor</b> <ol style="list-style-type: none"> <li>1. El estudiante da respuesta al primer ejercicio.</li> <li>2. Mediante los elementos interactivos anterior y siguiente se desplaza al siguiente ejercicio para realizarlo.</li> <li>3. El estudiante culmina y pide revisión mediante el elemento interactivo Revisar (botón).</li> </ol>	<b>Respuesta del sistema</b> <ol style="list-style-type: none"> <li>4. Se ejecuta la vista "Respuesta" y culmina la vista.</li> </ol>	<b>Elementos de la Vista</b>	
<b>Curso alternativo de los eventos</b>			
<b>Acción</b>	<b>Curso Alterno</b>		
-	-		
<b>Prioridad</b>	Crítica		
<b>Mejoras</b>	-		
<b>Medias a utilizar</b>	<b>Tipo de Media</b>	<b>Descripción</b>	<b>Estado</b>
	<b>Imagen</b>	Imágenes referentes a las diferentes tipologías de los ejercicios en su representación.	Existente
	<b>Video o animación</b>	Animación en donde la mascota se muestra con una expresión de estimulación en función de la respuesta.	Existente

	<b>Sonido</b>	Cada vez que sale la mascota se produce un sonido afectivo.	Localización
	<b>Texto</b>	Textos que determinan los enunciados de los ejercicios.	Existente
<b>Reglas Pedagógicas</b>	-		
<b>Representación de la vista</b>	-		

**Tabla3. Descripción Textual de la Vista de Presentación Respuesta.**

<b>Descripción Textual de la vista Respuesta</b>		
<b>Actores de la vista de presentación Respuesta.</b>	<b>Estudiante</b>	
<b>Propósito</b>	Permitir al estudiante ver la respuesta de los ejercicios que realizó, así como un resumen donde se le presenta una estadística con la evaluación de los ejercicios.	
<b>Objetivos pedagógicos</b>	Permite que el estudiante sea evaluado según sus conocimientos, y además le brinda una <i>retroalimentación</i> para cada opción de respuesta, así como profundizar en el tema.	
<b>Resumen</b>	La vista se inicia cuando el estudiante solicita la respuesta del ejercicio mediante la opción <i>Ver respuesta</i> , el sistema le muestra la respuesta; la estadística con la cantidad de preguntas, los intentos y el tiempo que tardó en realizarlas.	
<b>Vistas asociadas</b>	Vista " <i>Resultados</i> ".	
<b>Referencias</b>	RF1.4, RF1.4.1, RF1.4.2, RF1.4.3, RF2, RF3, RF4, RF5, RF6 y RF6.1	
<b>Precondiciones</b>	Debe haberse ejecutado la vista " <i>Ejercicios</i> ".	
<b>Poscondiciones</b>	Vista de presentación Resultados visualizada.	
<b>Curso Normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	<b>Elementos de la Vista</b>



6. El usuario selecciona el elemento interactivo Terminar (botón).	7. Se ejecuta la vista "Resultados", y culmina la vista.		
<b>Curso alternativo de los eventos</b>			
<b>Acción</b>	<b>Curso Alterno</b>		
-	-		
<b>Prioridad</b>	Crítica		
<b>Mejoras</b>	-		
<b>Medias a utilizar</b>	<b>Tipo de Media</b>	<b>Descripción</b>	<b>Estado</b>
	<b>Imagen</b>	Imágenes referentes a las diferentes tipologías de los ejercicios en su representación.	Existente
	<b>Video o animación</b>	Animación en donde la mascota se muestra con una expresión de estimulación en función de la respuesta.	Existente
	<b>Sonido</b>	Cada vez que sale la mascota se produce un sonido afectivo.	Existente
	<b>Texto</b>	Textos que determinan las respuestas de los ejercicios.	Existente
<b>Reglas Pedagógicas</b>	-		
<b>Representación de la vista</b>	-		



#### 2.4.4 Vista de Gestión del Modelo

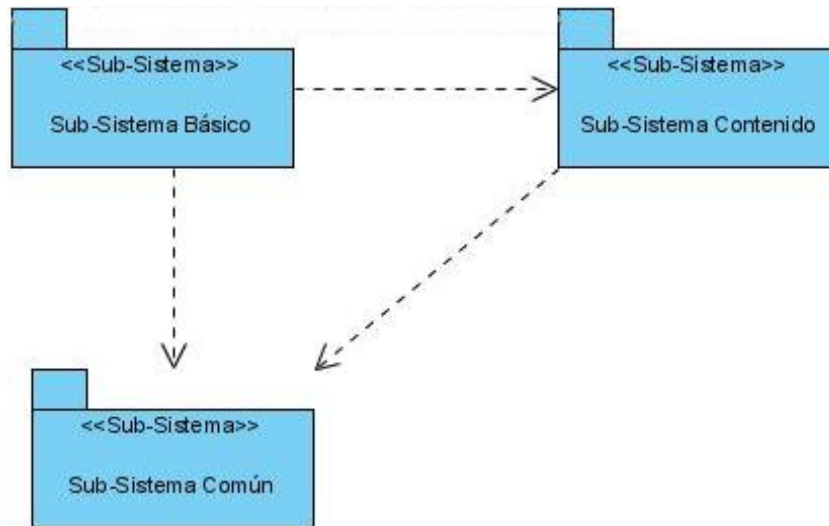


Figura 7 Representación de la Vista de Gestión de Modelo.

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

El subsistema **Común** está compuesto por los elementos interactivos *Música*, *Buscar*, *Pantalla Principal*, *Ayuda*, *Salir*. *Música* es el elemento interactivo para conectar o desconectar la música. *Buscar* brinda un servicio de búsqueda mediante la vista de presentación *Búsqueda*. *Pantalla Principal* es un elemento interactivo que lleva al usuario a la interfaz de comunicación con el usuario principal de la aplicación. *Ayuda* contiene la vista de presentación *Ayuda*. *Salir* permite al usuario salir de la aplicación e incluye la vista de presentación *Confirmar*.

El subsistema **Contenido** está compuesto por: la vista de presentación *Selección*, conformada por los elementos interactivos que permiten seleccionar el tema y dentro del mismo, cuántos o cuáles ejercicios el usuario va a realizar; la vista de presentación *Ejercicios* que plantea el enunciado del ejercicio a resolver, y contiene los elementos interactivos que permiten al usuario desplazarse al ejercicio anterior o siguiente, *Revisar* analiza si la respuesta está correcta y *Ver Respuesta* que permite ver cuál es la respuesta correcta del ejercicio e incluye la vista de presentación *Respuesta*, además

presenta los elementos interactivos *Saber más* que ofrece la oportunidad de profundizar en un tema determinado, *Estadística* que presenta la cantidad de preguntas e intentos realizados y el tiempo que tardó en elaborarlas, y *Terminar* que contiene la vista de presentación Resultados donde muestra el resultado del estudiante mediante un texto resumen.

#### **2.4.5 Diagrama de Estructura de Presentación por vista**

El diagrama de *Estructura de Presentación* surge como parte de la vista de Presentación en ApEM-L. El mismo, extiende la semántica de los diagramas de clases para la representación de la estructura de las presentaciones, estableciendo una organización lógica de los elementos que conforman cada interfaz de comunicación con el usuario en el futuro. Define 2 nuevos tipos de clases:

- Clase Estática: Agrupará los componentes que tienen como función visualizar información, pero no permiten interacción con el usuario.
- Clase Interacción: Agrupará los elementos de la vista que permiten la interacción del usuario con el sistema informático creado.

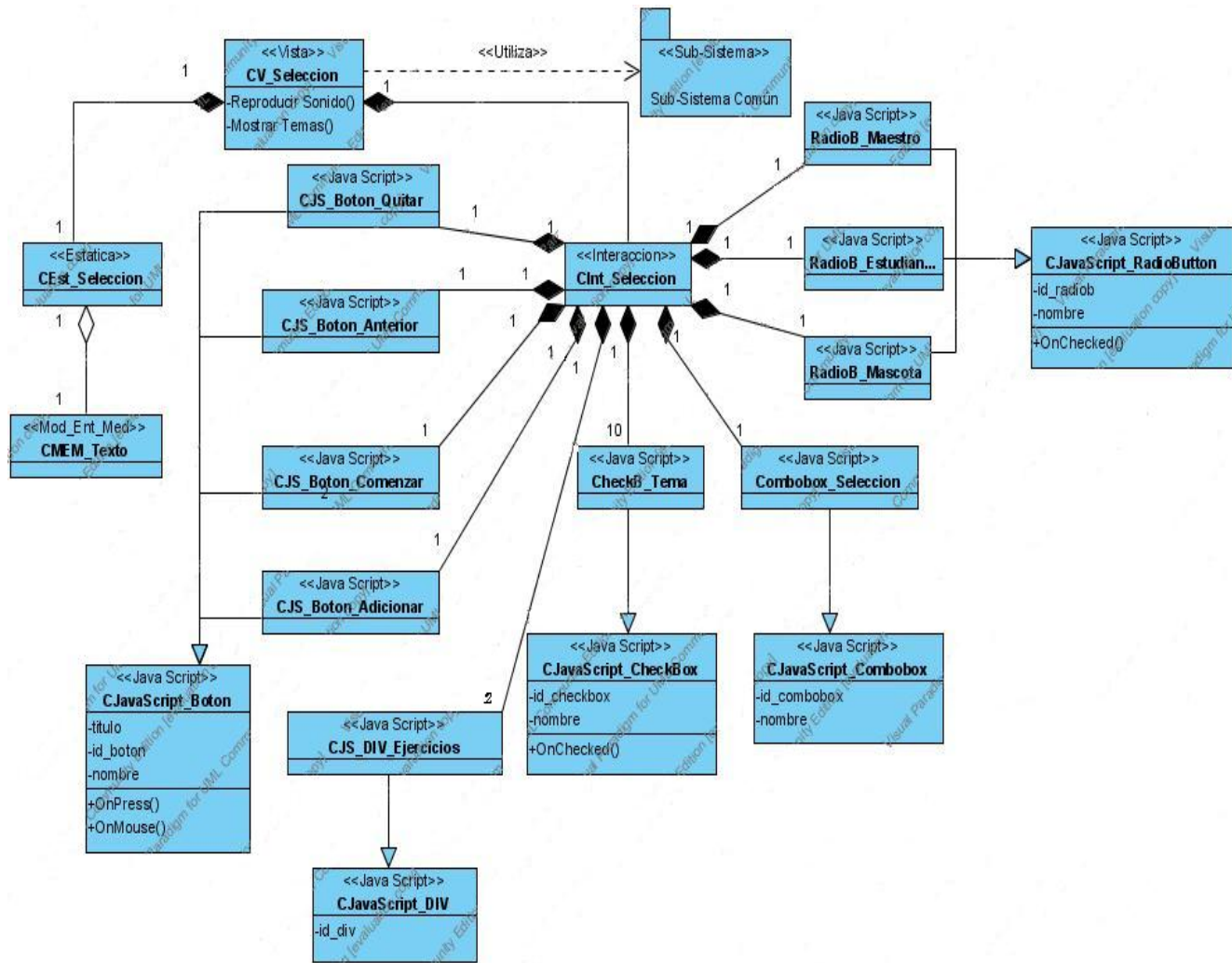


Figura 8 Diagrama de Estructura de Presentación de la vista Selección.

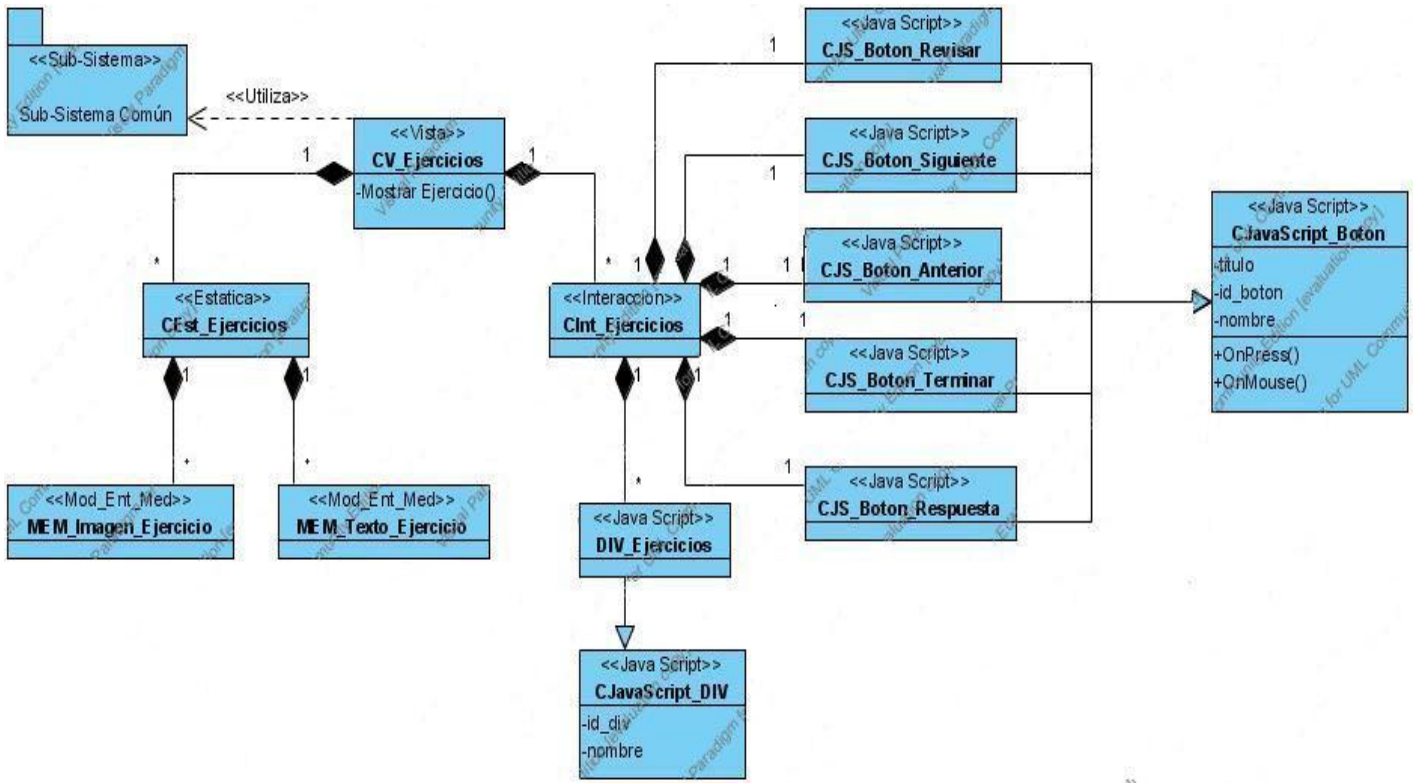


Figura 9 Diagrama de Estructura de Presentación de la vista Ejercicios.

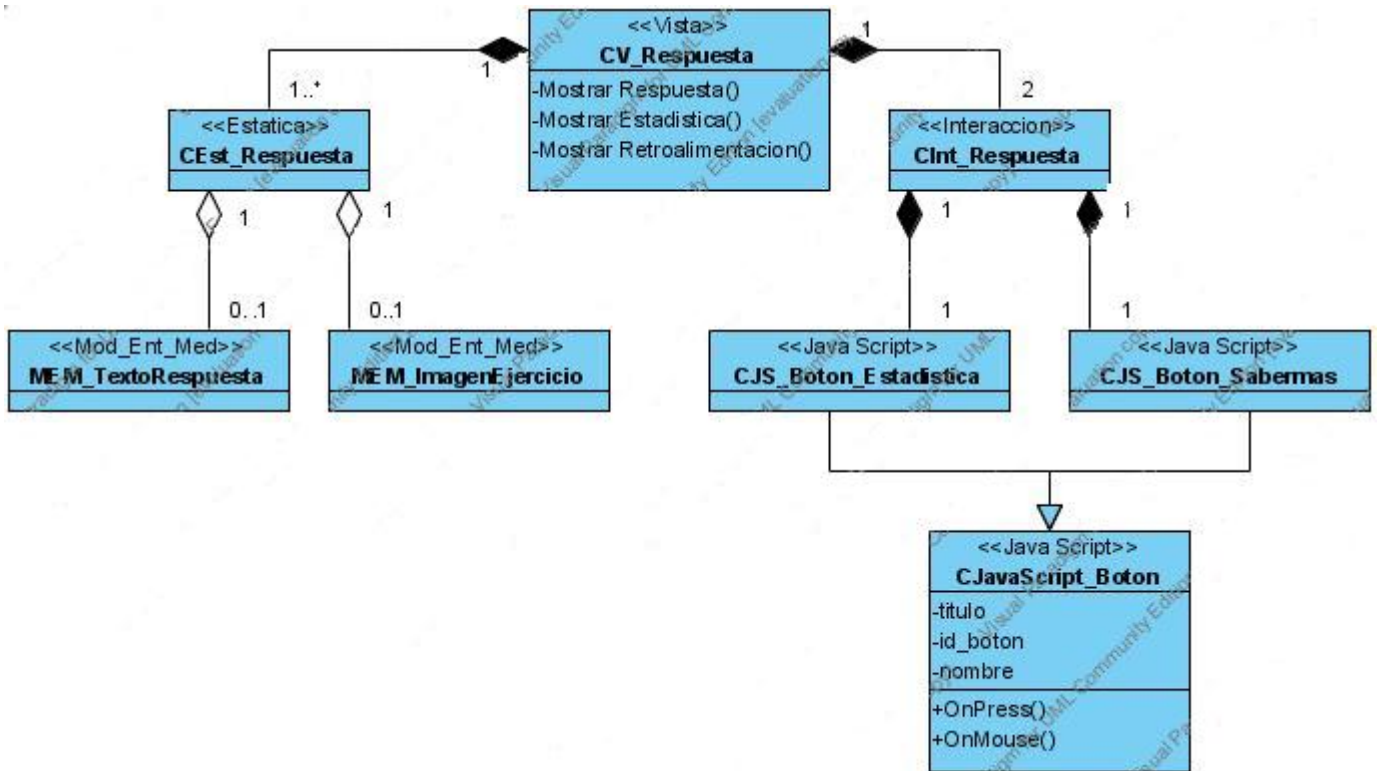


Figura 10 Diagrama de Estructura de Presentación de la vista Respuesta.

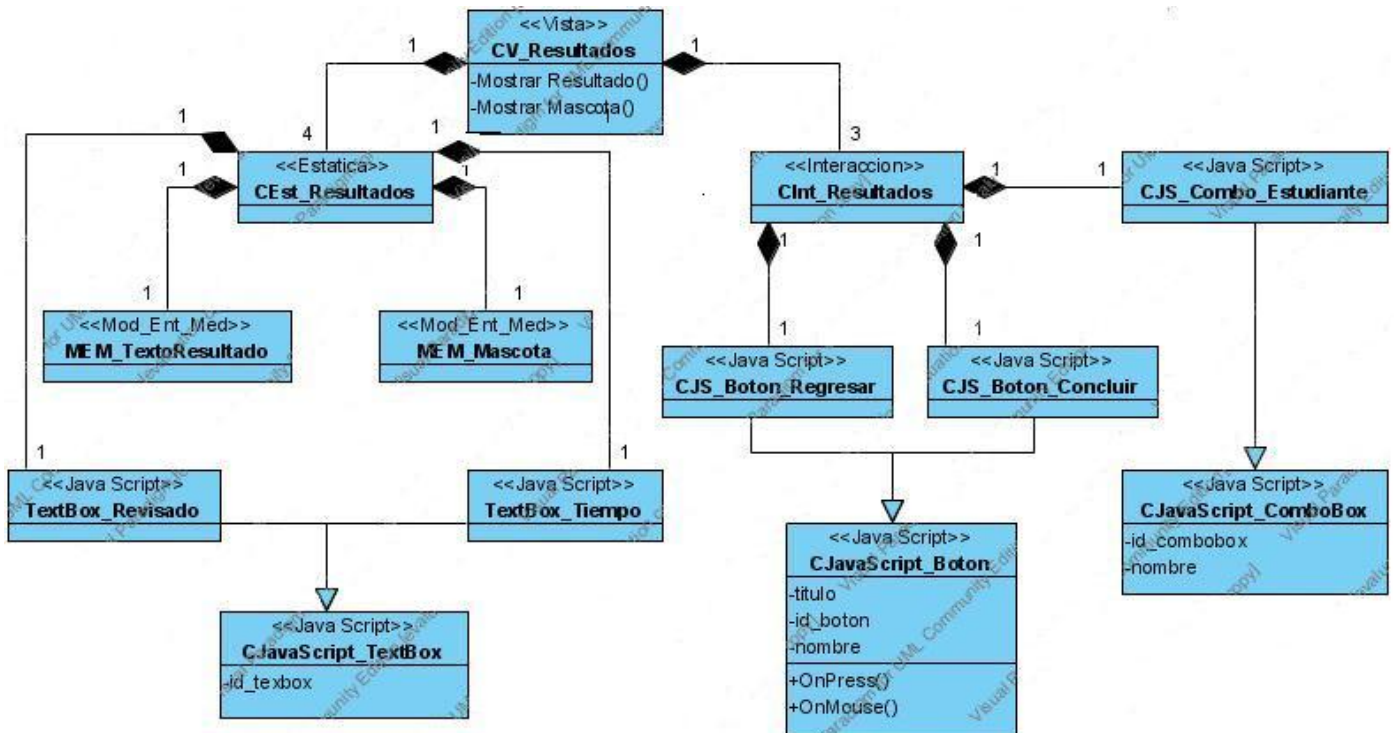


Figura 11 Diagrama de Estructura de Presentación de la vista Resultados.

## 2.4.6 Diagrama de Estructura de Navegación

El Diagrama Estructura de navegación permite representar las clases que posibilitan la navegación entre las distintas clases que componen la aplicación. Para la modelación de este diagrama ApEM\_L define las clases: menú, índice, consulta y botón, además de las ya definidas modelo-entidad-media texto y modelo-entidad-media imagen.

La clase **menú** es el elemento de composición de una clase vista, desde donde se puede llegar a otras diferentes clases vistas con las cuales se puede conectar este menú, pues contiene una lista de las opciones de movimiento siguiente, pero no los valores de esas opciones, los cuales serán ofrecidos por el resto de los tipos de clases. La clase **consulta** es aquel tipo de clase que permite el enlace con una clase vista y a través de un valor directo (variable de consulta) asignado a la búsqueda para la

visualización del elemento a mostrar, que sitúa como identificador en la relación entre las clases. La clase **índice** denota el valor de direccionamiento hacia una clase vista a partir de una opción determinada. La clase **botón** es aquella clase que denota la existencia de un elemento interactivo del tipo botón para producir un camino en la navegación hacia una clase vista, y utilizando como intermediarias a las clases de tipo índice o consulta.[2]

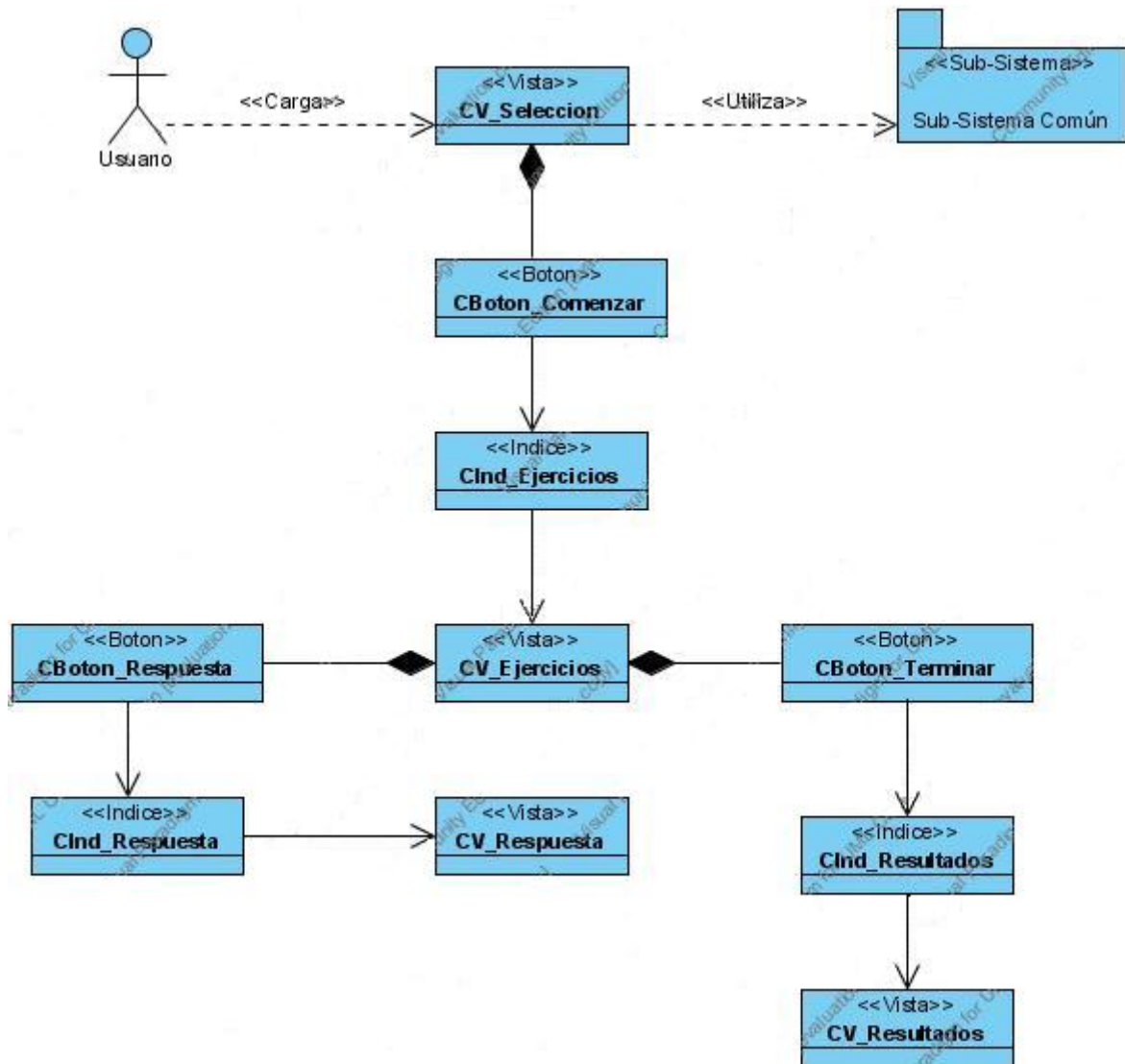


Figura 12 Diagrama de Estructura de Navegación.

### 2.4.7 Diagrama de Clases

El diagrama de clases pertenece a la vista estática que define ApEM-L y muestra un conjunto de clases, interfaces, colaboraciones y sus relaciones. Los diagramas de clases se usan en el diseño del modelo estático para ver el sistema. Su representación permite conocer y relacionar los conceptos que sustentan el funcionamiento del software educativo en cuestión, además establece el patrón arquitectónico MVC-E para la concepción del diseño de aplicaciones educativas. Este diagrama va a tener dos grandes partes. En la parte izquierda, dedicada a la representación del árbol jerárquico de medias, y en la derecha del diagrama, las clases que controlan la lógica del negocio de la aplicación, donde se tienen las siguientes clases:

**Clase vista:** Recibirán las peticiones del usuario al sistema y mostrarán los mensajes de salida o respuestas.

**Clase Controladora:** Gestionarán las peticiones y la muestra de las respuestas.

**Clase Modelo:** Contendrán la lógica de negocio para el procesamiento de la información.

**Clase Modelo Entidad Persistente:** Procesarán la información persistente.

**Clase correspondiente al Lenguaje de Alto Nivel:** Seleccionado para la programación del software.

[2]



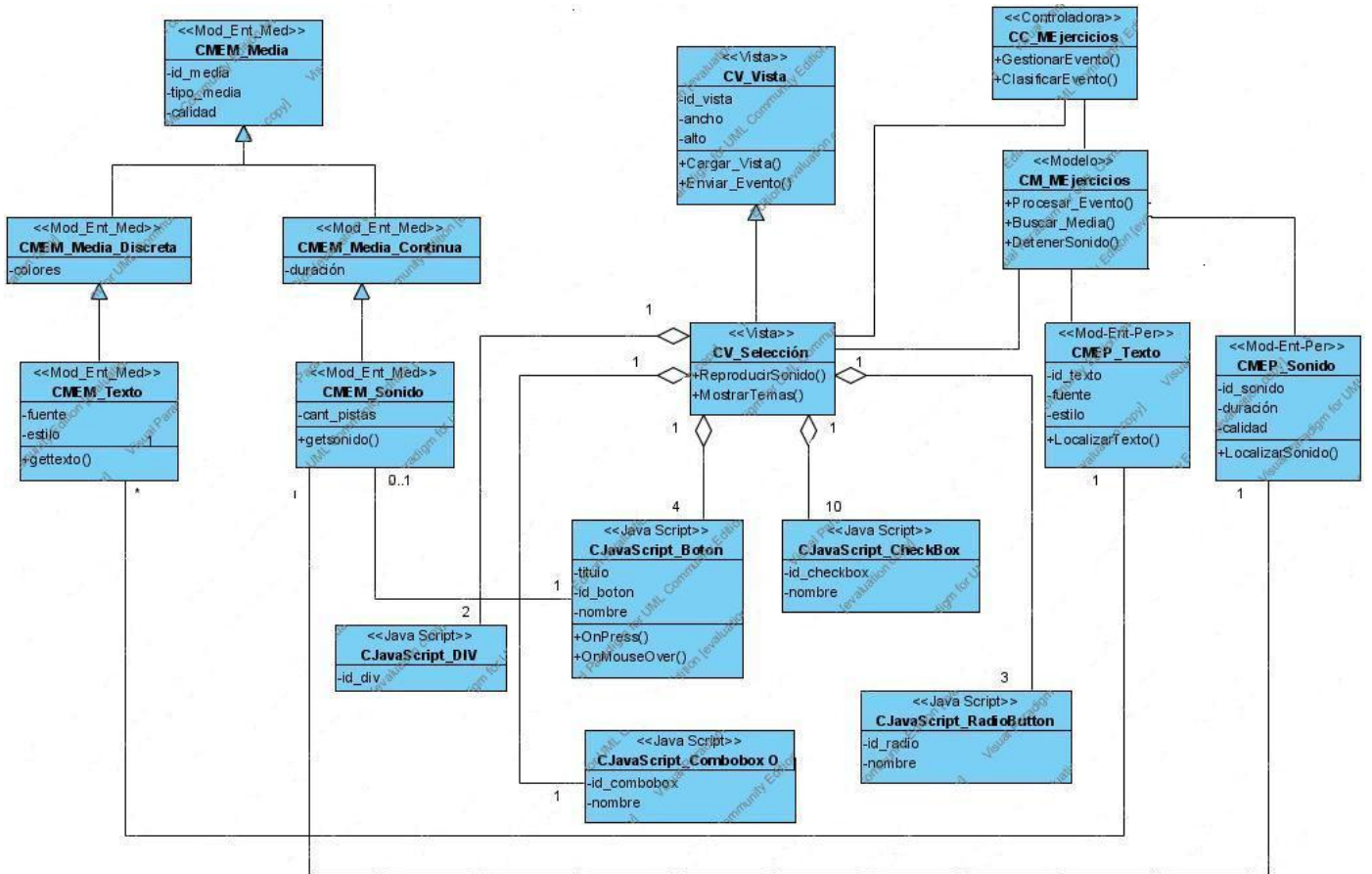


Figura 13 Diagrama de Clases del diseño de la vista Selección.

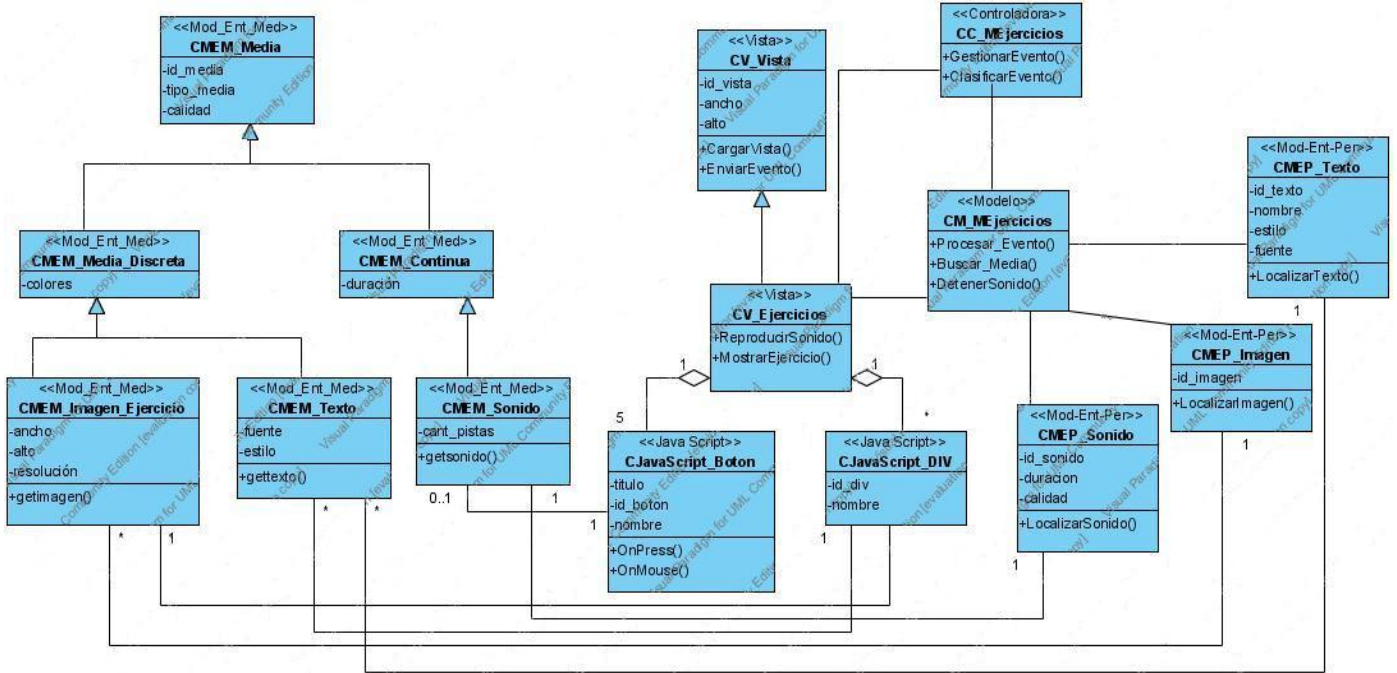


Figura 14 Diagrama de Clases del diseño de la vista Ejercicios.

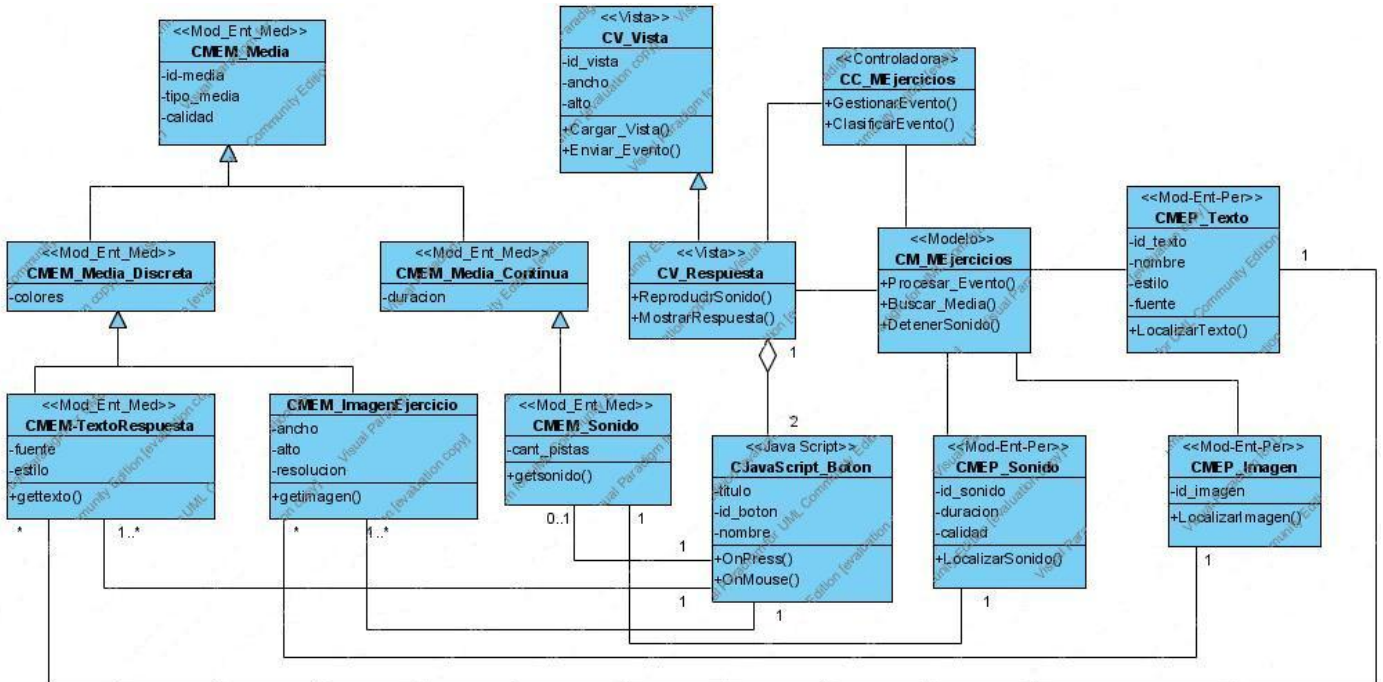


Figura 15 Diagrama de Clases del diseño de la vista Respuesta.

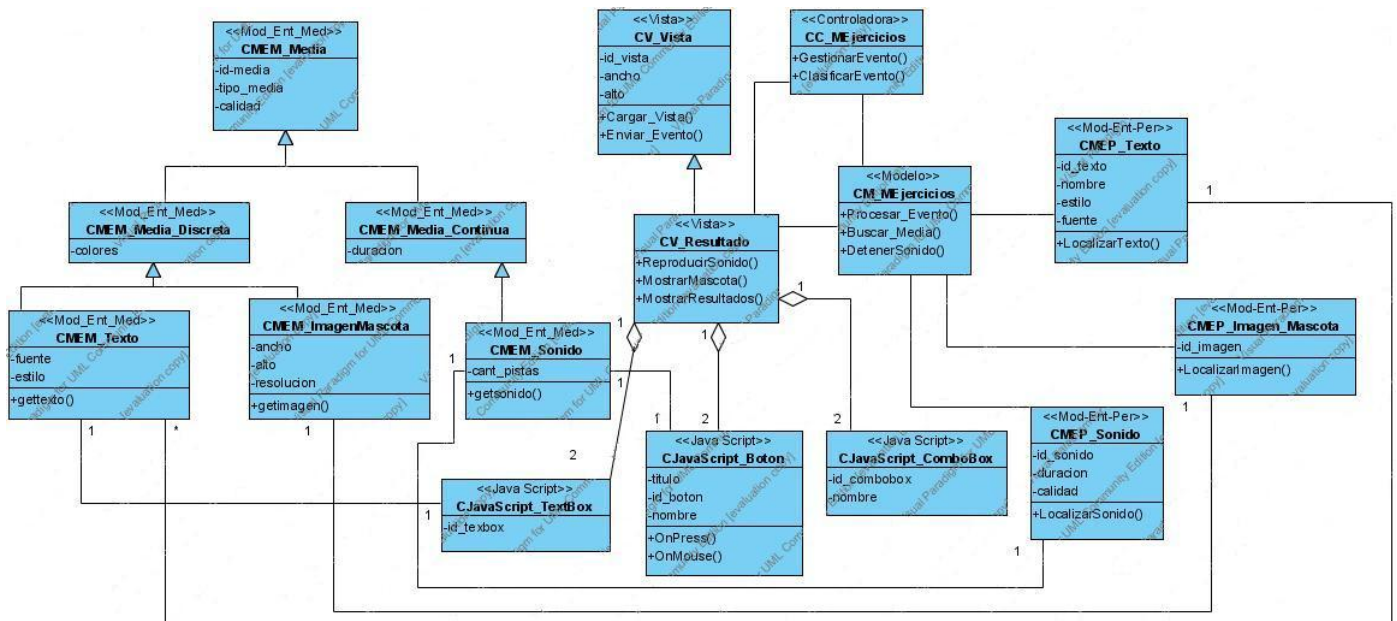


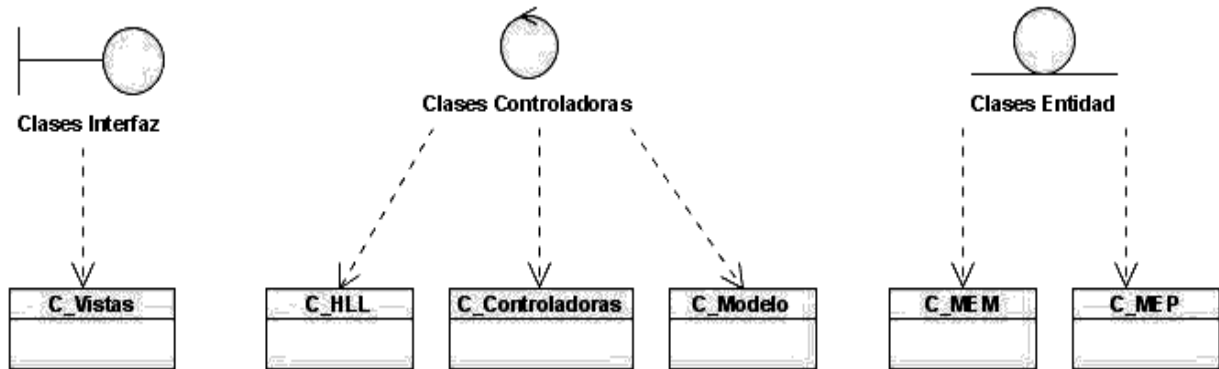
Figura 16 Diagrama de Clases del diseño de la vista Resultados.

## 2.4.8 Diagrama de Interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva a modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. En el contexto de las clases describen la forma en que grupos de objetos colaboran para proveer un comportamiento. Existen dos tipos de diagramas de interacción:

- Diagramas de secuencia (sequence diagrams).
- Diagramas de colaboración (collaboration diagrams).

ApEM-L ha agregado al diagrama de secuencia un estereotipo descriptivo para denotar el tiempo de demora para que se produzca un evento determinado como variable para este tipo de aplicaciones. Las clases que se utilizan para modelar este tipo de diagramas se estereotipan de la siguiente manera:



**Figura 17** Estereotipos de las clases interfaz, controladora y entidad.

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

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

Las clases *Entidad* serían las clases *Modelo Entidad* tanto, las que son para el procesamiento de la información *persistente*, como las correspondientes a la representación del árbol jerárquico de medias.[2]

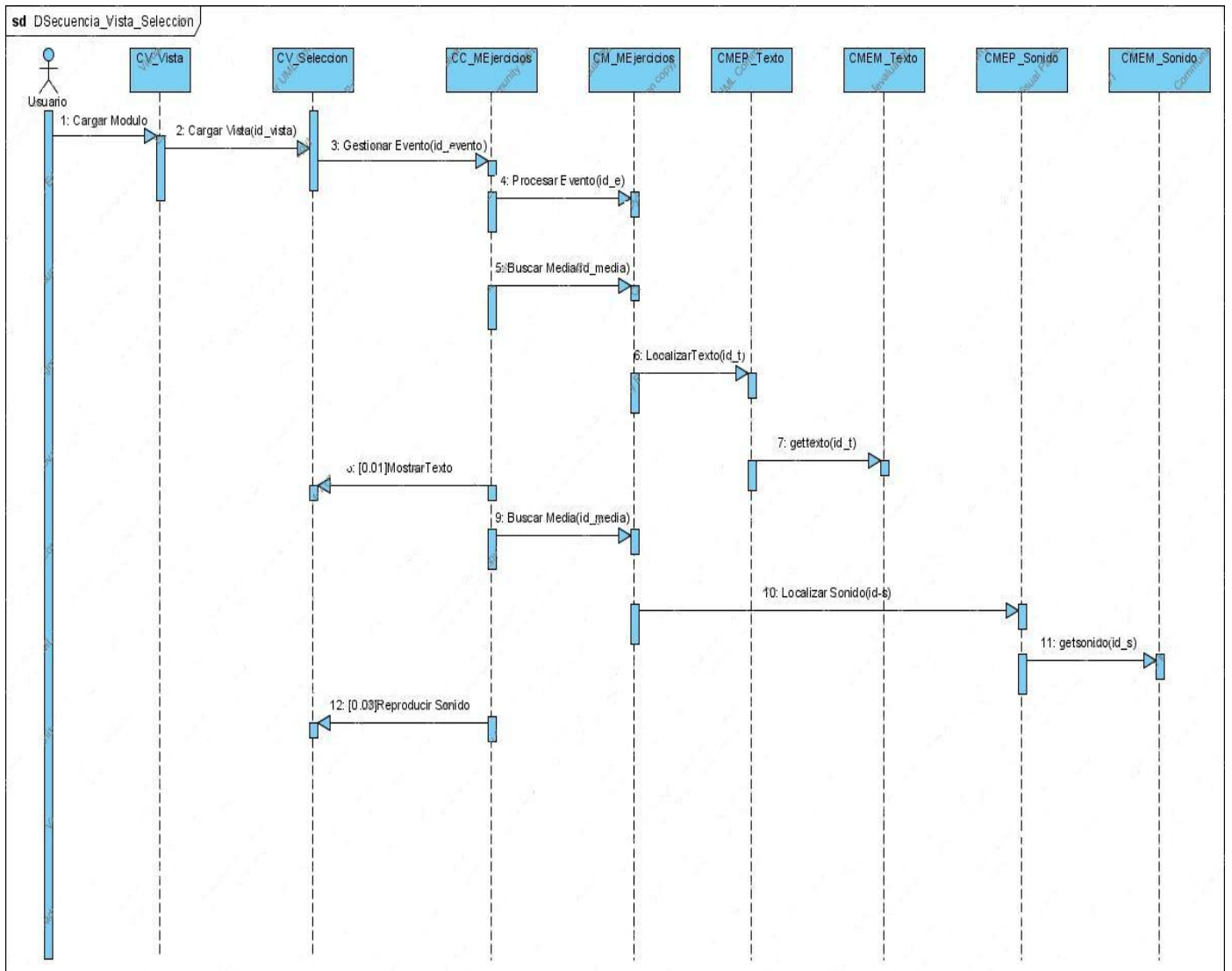


Figura 18 Diagrama de secuencia de la vista Selección.

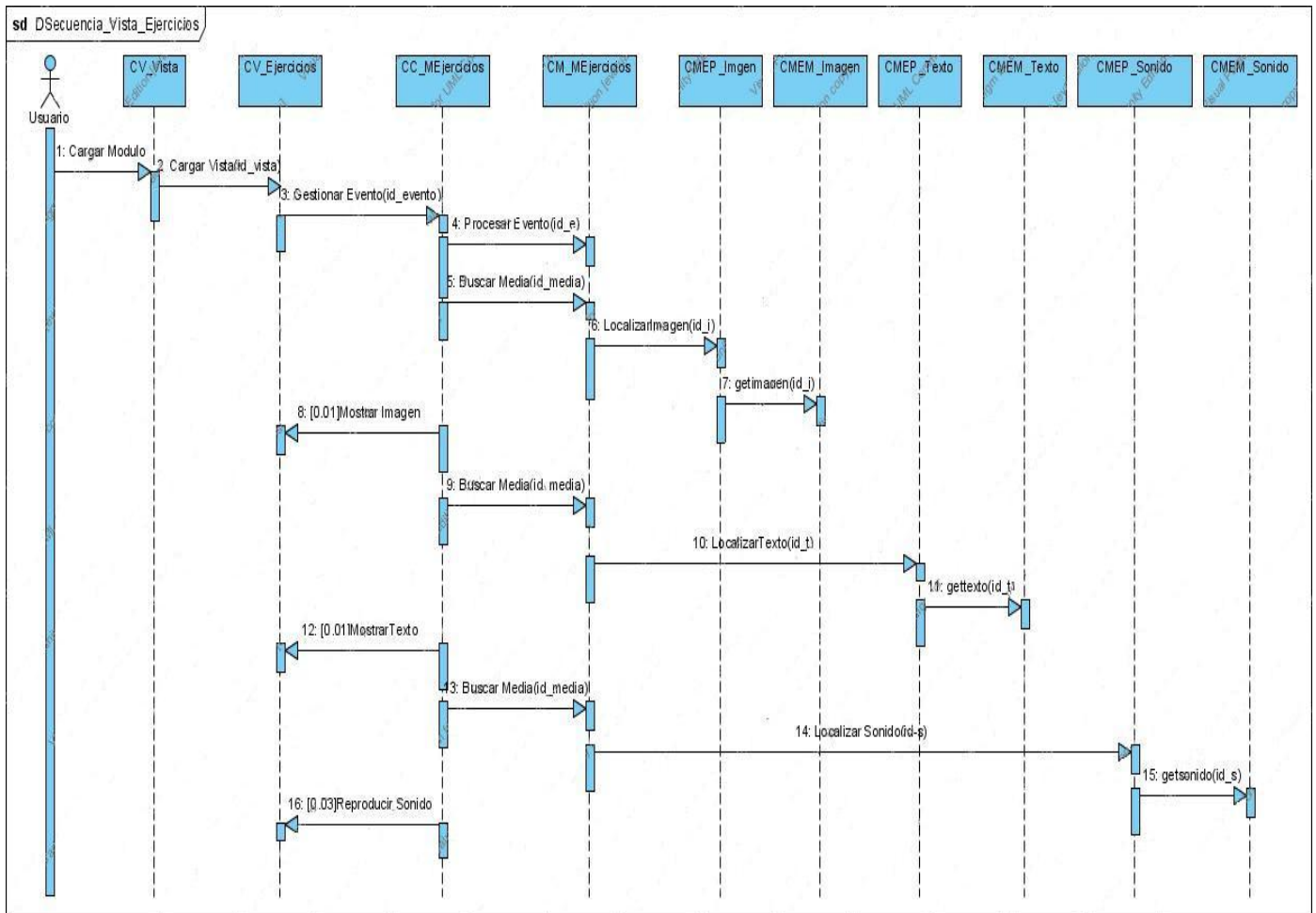


Figura 19 Diagrama de secuencia de la vista Ejercicios.

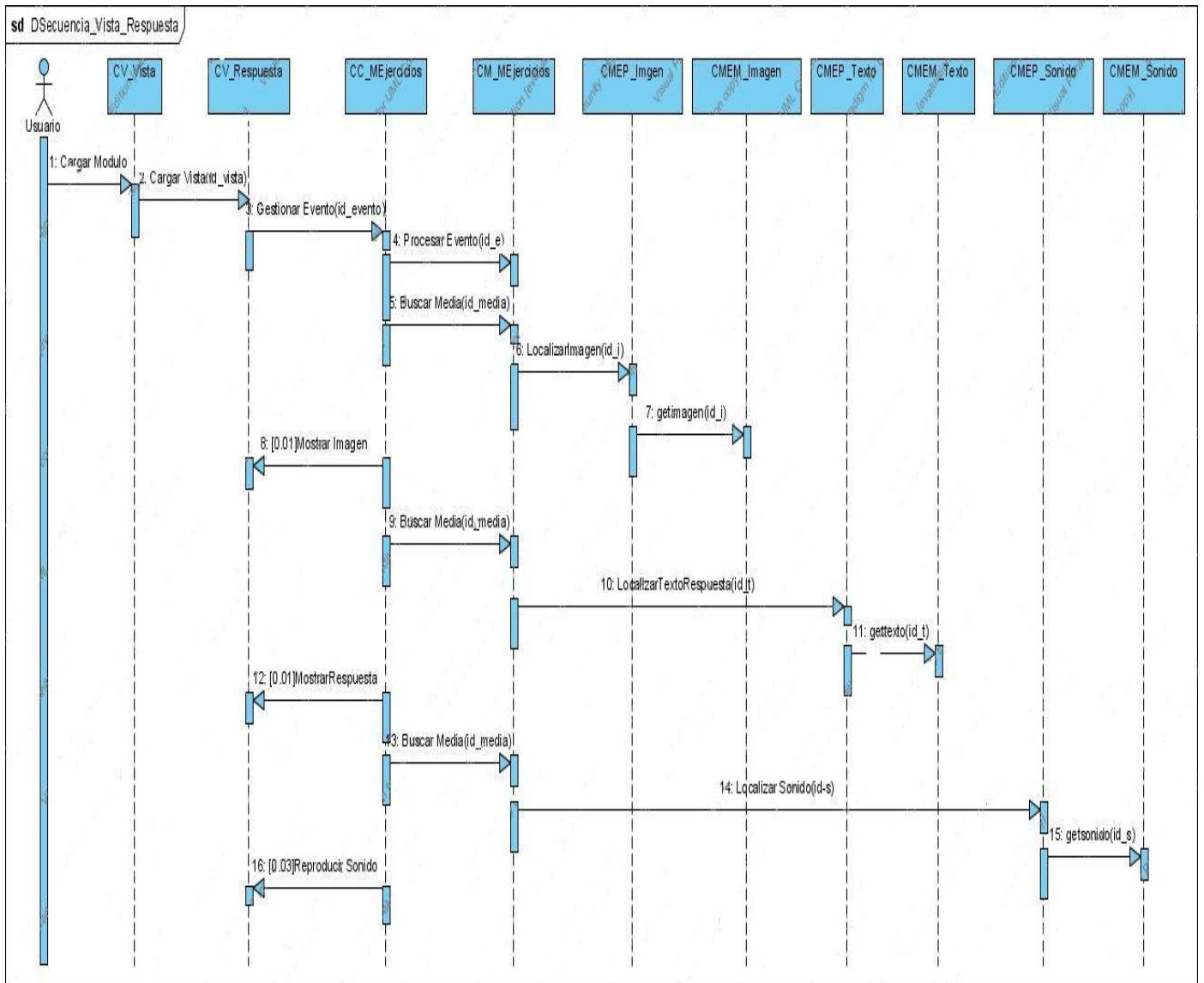


Figura 20 Diagrama de secuencia de la vista Respuesta.

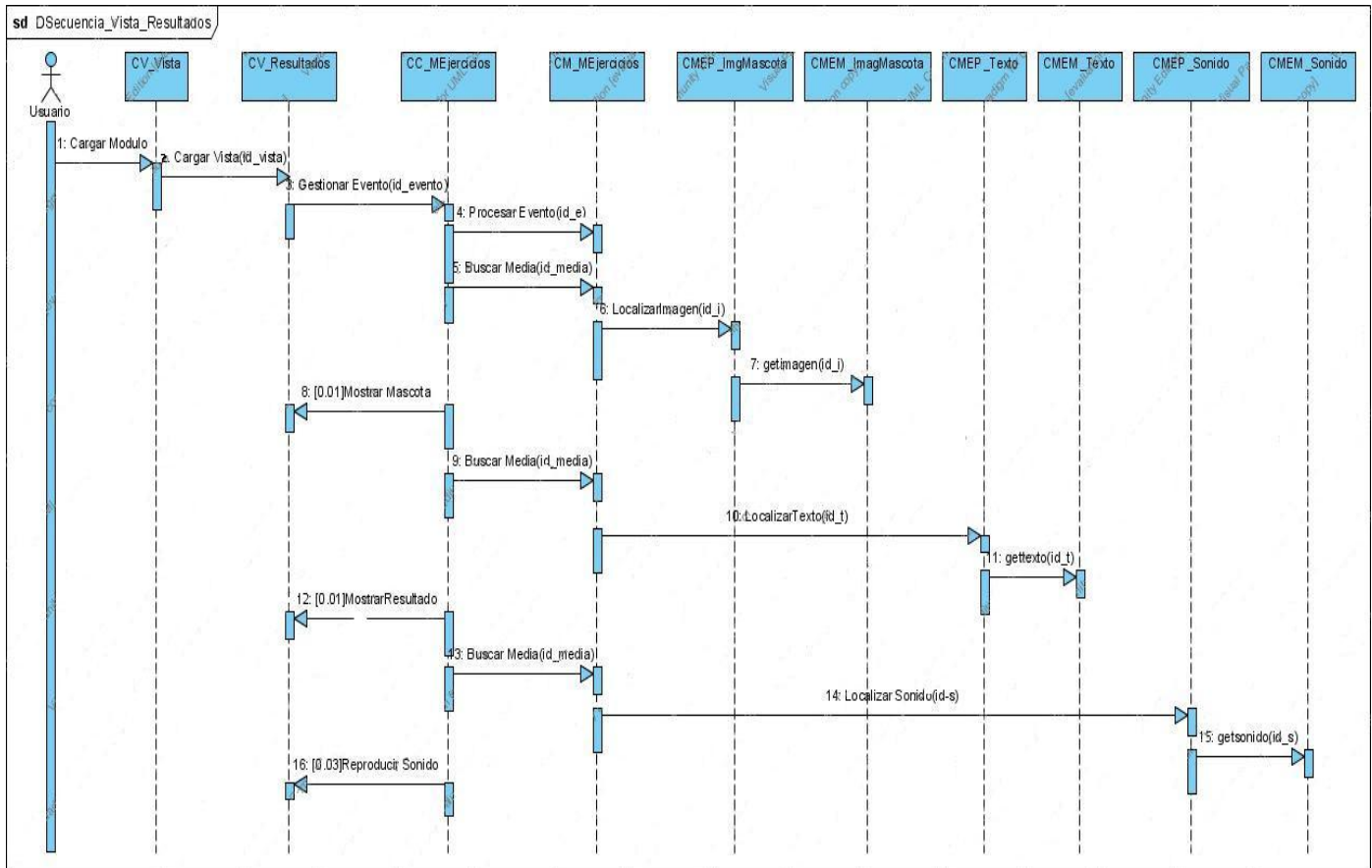


Figura 21 Diagrama de secuencia de la vista Resultados.

## 2.5 Conclusiones Parciales

En el capítulo se obtuvieron los artefactos pertinentes del flujo de trabajo de análisis y diseño del sistema, donde se representaron los diagramas necesarios para la documentación del módulo en cuestión. Además, se realiza el modelo de dominio para la comprensión del contexto del sistema; se definieron los requisitos funcionales y no funcionales que debe cumplir la aplicación; se elaboró la vista de gestión del modelo que brinda una visión más completa del sistema; y se hace la descripción textual para cada una de las vistas de presentación del módulo Ejercicios. Indudablemente, todo esto posibilita una mejor visión de la solución propuesta.



# Capítulo 3. Estudio de Factibilidad

## 3.1 Introducción

Es muy necesario para la elaboración de un software estar al tanto de los beneficios que reportará, el análisis del costo, el tiempo de desarrollo en meses y la cantidad de personas que se necesitarán para desarrollarlo, y así, tomar la decisión de si es factible o no la implementación del mismo. Para definir la factibilidad de un proyecto existen varios métodos de estimación, siendo los más conocidos, Análisis por puntos de casos de uso, Análisis de puntos de función y COCOMO II (Constructive Cost Model).

En este capítulo se realizará el estudio para determinar si es viable o no desarrollar el software multimedia que se propone como solución al problema planteado. Debido a la utilización del lenguaje de modelado ApEM\_L, el cual emplea vistas de presentación, se utilizará el método COCOMO II (Modelo Constructivo de Costes) y Puntos de Función, para realizar el estudio de factibilidad.

## 3.2 Estimación del Costo del Proyecto

El método COCOMO II se basa en la aplicación de ecuaciones matemáticas sobre los Puntos de Función o la cantidad de líneas de código (SLOC, Source Lines Of Code) estimados para un proyecto. Para la estimación del tamaño de un sistema a partir de sus requerimientos, una de las técnicas más difundidas es el Análisis de Puntos de Función. Ésta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación, las metodologías, plataformas y/o tecnologías utilizadas, denominadas Puntos de Función. [9]

En relación a los Puntos de Función, las transacciones se clasifican de la siguiente manera:

- Entradas Externas (EI)
- Salidas Externas (EO)
- Consultas Externas (EQ)

En relación a los Puntos de Función, los archivos se clasifican de la siguiente manera:

- Archivos Lógicos Internos (ILF)
- Archivos de Interfaz Externos (EIF)

**Entradas Externas:** No hay.

**Tabla 4 .Consultas Externas.**

<b>Nombre de la consulta externa</b>	<b>Cantidad de archivos referenciados</b>	<b>Cantidad de elementos de datos</b>	<b>Clasificación (Baja, Media, Alta)</b>
Buscar Información	1	1	Baja

**Tabla 5 .Salidas Externas del módulo Ejercicios.**

<b>Nombre de la salida externa</b>	<b>Cantidad de archivos referenciados</b>	<b>Cantidad de elementos de datos</b>	<b>Clasificación (Simple, Media, Alta)</b>
Mostrar la vista de selección de los Ejercicios	1	1	Baja
Mostrar los temas de los Ejercicios	1	1	Baja
Mostrar alumnos que realizarán los ejercicios.	1	1	Baja
Mostrar imagen del ejercicio.	1	1	Baja
Mostrar texto del ejercicio.	1	1	Baja
Mostrar la respuesta del ejercicio.	1	1	Baja
Mostrar estadística.	1	1	Baja
Mostrar retroalimentación (Saber más).	1	1	Baja
Mostrar anterior/próximo ejercicio.	1	1	Baja
Mostrar resumen de los resultados.	1	1	Baja
Mostrar imagen de la mascota.	1	1	Baja
Permitir al usuario ir a inicio	1	1	Baja
Permitir el acceso al resto de los módulos en todo momento.	1	1	Baja
Permitir al usuario salir de la aplicación en cualquier momento.	1	1	Baja

**Tabla 6. Elementos de Información, valores y clasificación.**

Elementos de Información	Simples		Medios		Complejos		Subtotal de Puntos de Función
	Cuenta	Peso	Cuenta	Peso	Cuenta	Peso	
Consultas externas	1	4	0	4	0	7	4
Salidas externas	14	4	0	5	0	7	56
<b>Subtotal (UFP)</b>							<b>60</b>

**Tabla 7. Puntos de Función.**

Características	Valor
Puntos de función desajustados	60
Lenguaje	Java Script 2.0
Instrucciones fuentes por puntos de función	56
Instrucciones fuentes	3360

**Resultados:**

Puntos de función del proyecto: 60

Total de instrucciones fuentes: 3360.

Miles de instrucciones fuentes (*MF*): 3.4 (aproximadamente)

### 3.3 Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo

**Tabla 8. Multiplicadores de esfuerzo.**

Multiplicadores	Multiplicadores	Descripción
RCPX (RELY, DATA, CLPX, DOCU )		Las exigencias de confiabilidad, documentación y volumen de datos son

	1.00	moderados, y la complejidad del producto es simple.
<b>RUSE</b>	1.07	El código se puede reutilizar para futuros programas.
<b>PDIF (TIME, STOR, PVOL)</b>	1.00	La plataforma es estable.
<b>PERS (ACAP, PCAP, PCON)</b>	1.00	La capacidad de los programadores es normal.
<b>PREX (AEXP, PEXP, LTEX)</b>	1.00	La experiencia del equipo de trabajo, dígame programadores es moderada.
<b>FCIL (TOOL, SITE)</b>	1.00	Se tienen herramientas CASE simples e infraestructura de comunicación básica.
<b>SCED</b>	1.00	Se requiere terminar el proyecto en el tiempo estimado, alta planificación.

**Tabla 9. Factores de Escala.**

<b>Factores de Escala</b>	<b>Valor</b>	<b>Descripción</b>
PREC	2.48	Precedencia: Tiene precedencia el trabajo.
FLEX	3.04	Flexibilidad: Algo de relajación en cuanto a la flexibilidad del desarrollo.
RESL	2.83	Riesgos: el plan identifica riesgos críticos y establece hitos para resolverlos.
TEAM	2.19	Cohesión de equipo: experiencia previa media operando como equipo.
PMAT	4.68	Madurez de las capacidades: normal

**Multiplicadores de esfuerzo:**

7

$$EM = \prod_{i=1}^7 E_{mi} = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED = 1.1449$$

i=1

**Factores de escala:**

5

$$SF = \sum_{i=1}^5 SF_i = PREC + FLEX + RESL + TEAM + PMAT = 15.22$$

**Valores calibrados:**

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

$$E = B + 0.01 * SF$$

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

$$E = 0.91 + 0.01 * 15.22$$

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

$$E = 1.0622 \approx 1.1$$

$$F = 0.278 \approx 0.3$$

**Esfuerzo:**

7

$$PM = A * (2.5)^E * \prod_{i=1}^7 Em_i$$

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

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

**Cálculo del tiempo:**

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

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

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

**Cálculo de la cantidad de hombres:**

$$CH = PM / TDEV$$

$$CH = 8 / 6.6 = 1.$$

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

Teniendo en cuenta que el módulo Ejercicios fue desarrollado por 2 hombres, se hace un reajuste del tiempo de desarrollo según la cantidad de hombres real, arrojando como resultado un tiempo de **4 meses**.

**Salario promedio:**

Para determinar el salario promedio se tiene en cuenta que el desarrollador es un trabajador de la UCI (Universidad de las Ciencias Informáticas), que tiene como salario correspondiente: **\$680.00**

**Costo:****CHM = 3\* Salario Promedio****CHM = 3\* \$680****CHM = \$2040****Costo = CHM \* PM****Costo = \$2040 \* 8****Costo = \$16320****Beneficios tangibles.**

El desarrollo del proceso que se propone: Análisis y diseño del módulo Ejercicios de la colección Multisaber, aportará numerosos beneficios perceptibles, ya que se contará con la documentación del módulo para la creación de una aplicación interactiva, dinámica y de interfaz agradable que posibilitará a los alumnos perfeccionar su aprendizaje, logrando integrar la tecnología multimedia a la educación.

**Beneficios intangibles.**

La realización del análisis y diseño del módulo Ejercicios de la colección Multisaber, permite al equipo de desarrollo implementar un producto de mayor calidad, capacitado para cumplir con los intereses del proceso educativo venezolano, e indudablemente, se reduce el tiempo de implementación y entrega del mismo.

### 3.4- Análisis de costos y beneficios

Por cálculos realizados anteriormente el costo de desarrollo de este software es \$16320 (moneda nacional), lo que equivale a 652.80 cuc. Tomando en cuenta los resultados tangibles e intangibles que aportaría la realización del mismo, se llegó a la conclusión que es factible desarrollar la aplicación y comercializarla.

### 3.5-Conclusiones Parciales

En este capítulo se efectuó el estudio de factibilidad del producto, donde se puntualizaron diferentes aspectos que son de trascendental importancia en todo proceso de desarrollo de un software como son el costo del producto, el tiempo de desarrollo, los beneficios tangibles e intangibles. A partir de las estimaciones realizadas se pudo llegar a la conclusión que es factible desarrollar el producto, el cual, según el tiempo estimado debe estar listo en **4** meses, con el trabajo de **2** hombres y un costo aproximado de **\$16320** en moneda nacional.

## Conclusiones Generales

Al culminar el presente trabajo, se puede afirmar que se alcanzaron resultados satisfactorios de acuerdo con el objetivo trazado.

- Se realizó un amplio estudio sobre la metodología, lenguaje de modelado y herramienta CASE existentes para la construcción de productos multimedia, donde se seleccionó como la metodología más adecuada para desarrollar la documentación en el flujo de trabajo de análisis y diseño del módulo Ejercicios de la colección Multisaber, al Proceso Unificado de Desarrollo de Software (RUP), como lenguaje de modelado ApEM-L, y como herramienta CASE Visual Paradigm.
- Se definieron las características y funcionalidades que deberá poseer el software, enmarcadas en un contexto pedagógico.
- Se obtuvieron los artefactos pertinentes al flujo de trabajo de análisis y diseño del módulo Ejercicios, que sirven de guía para realizar la implementación del producto.
- Se comprobó la factibilidad de realizar y comercializar el producto a partir del estudio de factibilidad realizado.

## Recomendaciones

- Aprovechar el estudio investigativo realizado para la posterior implementación del módulo Ejercicios de la colección Multisaber.
- Continuar indagando acerca de las necesidades del proceso educativo venezolano, para agregar nuevas funcionalidades al módulo.
- Utilizar la documentación del proceso análisis y diseño del módulo Ejercicios de la colección Multisaber para la creación de productos similares.



## Glosario de términos

### A

**API** (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.

**Actor del Sistema:** Rol o función que asume una persona, sistema o entidad que interactúa con el sistema que estamos construyendo de la misma forma. Tiene la propiedad de ser externo al sistema.[5]

**Archivo Lógico Interno:** Un archivo lógico interno es un grupo de datos lógicamente relacionados o informaciones de control, identificado por el usuario y mantenido dentro de las fronteras de la aplicación.[9]

**Archivo de interfaz externa:** según la visión del usuario, es un grupo de datos lógicamente relacionados o informaciones de control utilizadas por la aplicación pero que es mantenida por otra (por eso es de interfase externa). [9]

### C

**CASE:** (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diferentes aplicaciones informáticas que permiten aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar el diseño de un proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores.[4]

**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. (Verga y Miceli, 1994).

**Consulta Externa:** El EQ es un proceso elemental o una acción que envía datos o información de control hacia fuera de la frontera de la aplicación. El objetivo principal de un EQ es presentar información al usuario a través de la mera obtención del dato o de la información de control. A diferencia del EO, el procesamiento lógico no debe contener ninguna fórmula o cálculo matemático, ni tampoco debe crear datos derivados de los obtenidos. [9]

## **D**

**Diagrama:** Es la representación gráfica en la que se muestran las relaciones existentes entre las diferentes partes de un conjunto o sistema.

## **E**

**Entrada Externa:** (EI) es un proceso elemental o una acción que procesa datos o información de control que vienen desde afuera de la frontera de la aplicación hacia adentro. Los datos pueden venir desde otra aplicación o desde una pantalla de ingreso de datos. El objetivo fundamental es mantener uno o más archivos lógicos internos (ILF de Internal Logical File) y/o alterar el comportamiento del sistema. [9]

## **I**

**Integralidad (en el proceso docente cubano):** Al disminuirse la cantidad de alumnos por profesores, para lograr una mayor atención a las diferencias individuales de los alumnos y de esta forma reforzar el trabajo educativo, se obliga a elevar el rango de materias que el docente debe dominar para impartir todas la asignaturas.

**Interfaz de Comunicación con el Usuario:** Forma en que los usuarios pueden comunicarse con una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. [2]

## **R**

**Retroalimentaciones:** Cada opción de respuesta necesita de una retroalimentación, en el caso de las opciones de la respuesta correcta esta retroalimentación puede explicar brevemente porque es correcta, profundizar en el tema o hacer algún comentario al respecto. En el caso de los distractores la retroalimentación será el señalamiento del error cometido, una breve explicación de en que casos esa sería la respuesta correcta o un comentario al respecto, según corresponda.

## **S**

**Salida Externa:** (EO) es un proceso elemental o una acción que envía datos o información de control hacia fuera de la frontera de la aplicación. El objetivo fundamental es presentar información al usuario a través del procesamiento lógico de los datos o de la información de control obtenida. El procesamiento lógico debe contener al menos una fórmula o cálculo matemático o crear datos derivados de los obtenidos. Un EO podría mantener uno o más ficheros lógicos internos y/o alterar el comportamiento del sistema. [9]

**Software Libre** (free software) es la dominación del software que brinda libertad a los usuarios sobre el producto adquirido; no confundirlo con software gratis, es decir Software libre es la libertad con que cuentan los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software de modo más efectivo, que este se ajuste al contexto y peticiones del usuario. Software libre se refiere a cuatro libertades de los usuarios del software. [15]

## Referencias Bibliográficas

1. Bianchini, P.A. *Hipertexto*. 2000. 29 enero 2009 [cited; Available from: <http://www ldc.usb.ve/~abianc/hipertexto.html>]
2. Ciudad, A. F. R. 2007. ApEM – L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI. Tesis para optar por el título de Máster en Informática Aplicada.
3. Fabelo., D.C.M.d.C.S. *Prevención de las disgrafías escolares: una necesidad de la escuela actual, para la atención a la diversidad*. [cited; Available from: <http://www.monografias.com/trabajos55/disgrafia-escolar/disgrafia-escolar2.shtml>]
4. Herramienta CASE,2003.15 julio[cited; Available from: [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(Iglesia\\_Anglicana\)\\_%5BMac\\_OS\\_X\\_cuenta\\_14717\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5BMac_OS_X_cuenta_14717_p/)]
5. Ivar Jacobson, G.B., James Rumbaugh, *El Proceso Unificado de Desarrollo de Software (Volumen I y II)*.
6. Mora., A.J.H. *Multimedia*. [cited; Available from: <http://www.monografias.com/trabajos7/mult/mult2.shtml>]
7. Multimedia. [cited; Available from: <http://es.tech-faq.com/multimedia.shtml>]
8. *Object Oriented Hypermedia Design Model (OOHDM)*.2009 [cited; Available from: [http://es.answers.yahoo.com/question/index;\\_ylt=AkHOHFdjU5RdbPtv2tt10ZqR.gt;\\_ylv=3?qid=20080810154050AApYJwr](http://es.answers.yahoo.com/question/index;_ylt=AkHOHFdjU5RdbPtv2tt10ZqR.gt;_ylv=3?qid=20080810154050AApYJwr)]
9. Peralta, Mario. "Estimación del esfuerzo basada en casos de uso". [cited; Available from: [http://www.itba.edu.ar/index.php?seccion\\_generica\\_id=494](http://www.itba.edu.ar/index.php?seccion_generica_id=494)]
10. Pons, R.S.G.C.F. Un lenguaje para Transformación de Modelos basado en MOF y OCL [cited; Available from: <http://www.lifia.info.unlp.edu.ar/papers/2006/Giandini2006.pdf>]

11. Puig, C.T.i. *Hipermedia*. [cited; Available from: Carles Tomàs i Puig.
12. Rational Rose. [cited; Available from: <http://www.rational.com.ar/herramientas/rup.html>.
13. Conexion IT. Qué es RUP?. 2009 [cited; Available from: <http://www.conexionit.com/blog/metodologias/que-es-rup.html>
14. Wikilearning. *Hipermedia*. [cited; Available from: [http://www.wikilearning.com/curso\\_gratis/internet\\_y\\_la\\_web\\_definiciones-que\\_es\\_hipermedia/6065-1](http://www.wikilearning.com/curso_gratis/internet_y_la_web_definiciones-que_es_hipermedia/6065-1).
15. Yusef Hassan, F.J.M.y.G.I. *Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información*. [cited; Available from: <http://www.nosolousabilidad.com/articulos/hipermedia.htm>.

## Bibliografía

1. Adelaide Bianchini. Conceptos y Definiciones de Hipertexto. 2009 [cited; Available from: <http://www ldc.usb.ve/~abianc/hipertexto.html>
2. Ciudad, A. F. R. 2007. ApEM – L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI. Tesis para optar por el título de Máster en Informática Aplicada.
3. Colección Multisaber. 2008. Disponible en: <http://www.insted.rimed.cu/multisaber.asp> .
4. *Diagramas de Interacción*. 18 abril 2009 [cited; Available from: <http://tvdi.det.uvigo.es/~avilas/UML/node41.html>.
5. Funciones Tipo de Datos. 26 abril 2009 [cited; Available from: <http://www.geocities.com/gimenezpy/datos.htm>.
6. Herramienta CASE, 2003. 15 julio [cited; Available from: [http://www.freownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(Iglesia\\_Anglicana\)\\_%5BMac\\_OS\\_X\\_cuenta\\_14717\\_p/](http://www.freownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5BMac_OS_X_cuenta_14717_p/)
7. Instructivo para la Cuenta de Puntos de Función. [cited; Available from: <http://www.dicyt.gub.uy/pdt/files/6.2.5%20-%20puntosFuncion.pdf>.
8. Ivar Jacobson, G.B., James Rumbaugh, El Proceso Unificado de Desarrollo de Software (Volumen I y II).
9. Mora., A.J.H. *Multimedia*. [cited; Available from: <http://www.monografias.com/trabajos7/mult/mult2.shtml>
10. Multimedia. [cited; Available from: <http://es.tech-faq.com/multimedia.shtml>
11. *Object Oriented Hypermedia Design Model (OOHDM)*. 2009 [cited; Available from: <http://es.answers.yahoo.com/question/>
12. (Osjobu), G.B.O.J. *MIS TAREAS DE INGENIERIA DE SOFTWARE*. [cited; Available from: <http://diario.de/osjobu/uml.html>
13. Scribd. RUP. 2009 [cited; Available from: <http://www.scribd.com/search?cx=007890693382555206581%3A7fgc6et2hmk&cof=FORID%3A10&ie=UTF-8&c=all&ft=&q=ciclo+de+vida+de+RUP&sa=Search#1055>

14. Wikilearning. *Hipermedia*. [cited; Available from: [http://www.wikilearning.com/curso\\_gratis/internet\\_y\\_la\\_web\\_definiciones-que\\_es\\_hipermedia/6065-1](http://www.wikilearning.com/curso_gratis/internet_y_la_web_definiciones-que_es_hipermedia/6065-1)].
15. Yancy Martínez Pérez, A.D.D., Abel Ernesto Lorente Rodríguez. *Plantilla para el montaje dinámico de los productos de la colección Multisaber*. [cited; Available from: [http://www.informaticahabana.com/evento\\_virtual/files/MUL067.pdf](http://www.informaticahabana.com/evento_virtual/files/MUL067.pdf)].