



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

Título: Multimedia sobre el framework Symfony

Autores: Ainedy Domínguez López
Felipe Lastra Moreno

Symfony

Tutor: Ing. Ismael Armando Nodarse Mora

Co-Tutor: Ing. Leonardo San Román Labaut

Ciudad de La Habana, junio de 2009

“Año del 50 aniversario del triunfo de la Revolución”

Declaración de Autoría

Declaramos ser los autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso de ella para lo que necesite, cediéndole de esta forma los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autores: Ainedy Domínguez López

Felipe Lastra Moreno

Tutor: Ing. Ismael Armando Nodarse Mora

Co-Tutor: Ing. Leonardo San Román Labaut



“Seamos realistas y hagamos lo imposible.”

Ernesto Che Guevara

Agradecimientos

A nuestros padres y abuelos por depositar su confianza y brindarnos todo su amor y apoyo, por vivir junto a nosotros cada momento de tensión y desvelo, por saber guiarnos en situaciones difíciles y enseñarnos que todo es posible, por ser, los mejores del mundo.

A nuestro tutor Ismael y a nuestro co-tutor Leonardo por su esfuerzo y ayuda incondicional.

A nuestros compañeros de grupo y amigos por compartir juntos momentos de preocupación y alegría.

A todos los que se preocuparon por el desarrollo de esta tesis.

Dedicatoria

A nuestros familiares y amigos que han estado presentes siempre, brindándonos su apoyo y ayuda incondicional.

Resumen

El desarrollo de las nuevas tecnologías de la informática y las comunicaciones ha significado un salto fundamental en la productividad de las actividades de la sociedad cubana. Estas son empleadas principalmente en el desarrollo de software, teniendo como objetivo resolver cualquier problemática. Es por ello que el presente trabajo está enmarcado en desarrollar una aplicación que brinde información sobre el framework Symfony a los estudiantes de la Universidad de las Ciencias Informáticas (UCI), principalmente a los estudiantes de la Facultad 8. Con este fin se realiza el levantamiento de los requerimientos, el análisis, diseño e implementación de la solución propuesta siguiendo la metodología RUP, modelando con ApEM-L. Para el desarrollo del producto se analizan las tendencias, tecnologías y las posibles herramientas y se realiza un estudio de factibilidad determinando entre otras cosas la cantidad de programadores y el tiempo que se emplea en la elaboración del mismo. Con este software se pretende aumentar la bibliografía existente y el interés de los estudiantes por el aprendizaje del framework Symfony, mostrando la información de una manera amena e interesante.

Palabras claves: Multimedia, Framework Symfony, Tecnología Multimedia, ApEM-L

Índice

Introducción	1
Capítulo 1. Fundamentación Teórica	4
1.1 Introducción	4
1.2 Las Tecnologías de la Información y la Comunicación en la educación	4
1.2.1 La tecnología Multimedia y su aplicación en el software educativo	5
1.3 Audiencia	7
1.4 Análisis de otras soluciones existentes	7
1.5 Metodologías de Desarrollo de Software	7
1.5.1 Proceso Unificado de Rational (RUP)	8
1.5.2 Extreme Programming (XP)	9
1.5.3 Metodología de desarrollo seleccionada	10
1.6 Lenguajes de modelado	11
1.6.1 UML: Lenguaje Unificado de Modelado	11
1.6.2 OMMMA – L: Lenguaje para la Modelación Orientada a Objetos de Aplicaciones Multimedia	12
1.6.3 ApEM – L: Lenguaje para la Modelación de Aplicaciones Educativas Multimedia	14
1.6.4 Lenguaje de Modelado seleccionado	16
1.7 Herramientas de autor	16
1.7.1 Macromedia Director	17
1.7.2 ToolBook Instructor	17
1.7.3 Macromedia Flash 8	18
1.7.4 Herramienta seleccionada	19
1.8 Lenguajes utilizados en el desarrollo del producto multimedia	20
1.8.1 ActionScript 2.0	20
1.8.2 XML	20
1.9 Conclusiones del capítulo	21
Capítulo 2. Descripción de la solución propuesta	22
2.1 Introducción	22
2.2 Modelo de Dominio	22
2.2.1 Identificación de los conceptos del Dominio	23
2.3 Solución propuesta	24

2.4	Requerimientos del Sistema	24
2.4.1	Requisitos Funcionales del Sistema	25
2.4.2	Requisitos no Funcionales del Sistema	26
2.5	Vista de Gestión del Modelo	27
2.5.1	Vistas de Presentación	28
2.6	Justificación de los actores del sistema	29
2.7	Descripción textual de las Vistas de Presentación	29
2.8	Diagramas de Estructura de Presentación	37
2.9	Diagramas de Estructura de Navegación	41
2.10	Conclusiones del capítulo	44
Capítulo 3. Construcción de la solución propuesta		45
3.1	Introducción	45
3.2	Diagramas de Clases	45
3.3	Diagramas de Interacción	49
3.4	Modelo de Implementación	53
3.4.1	Diagramas de Componentes	53
3.4.2	Diagrama de Despliegue	56
3.5	Conclusiones del capítulo	57
Capítulo 4. Estudio de factibilidad		58
4.1	Introducción	58
4.2	Planificación	58
4.2.1	Estimación por puntos de función	59
4.3	Estimación del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo	60
4.4	Beneficios Tangibles	64
4.5	Beneficios Intangibles	64
4.6	Análisis de Costos y Beneficios	64
4.7	Conclusiones del capítulo	65
Conclusiones		66
Recomendaciones		67
Referencias Bibliográficas		68

Bibliografía	72
Anexos	75
Glosario de Términos	133

Introducción

Es casi inimaginable la velocidad a la que avanza la investigación y el desarrollo en el campo de la Computación y la Informática. Todos los niveles, entiéndase, gobierno, empresas, educación y la población son un reflejo de lo antes expuesto, por como incorporan tecnología de punta y como buscan las soluciones a muchos de sus problemas con aplicaciones informáticas. Cuba no está ajena a esta revolución tecnológica y como parte del proceso de transformaciones educacionales y sociales en la que está inmersa, surge en el año 2002 la Universidad de las Ciencias Informáticas (UCI) con dos objetivos principales: informatizar al país y desarrollar la industria del software para así contribuir al desarrollo económico del mismo. En aras de apoyar el cumplimiento de estos objetivos, la universidad asigna a cada una de sus facultades diferentes proyectos productivos, en dependencia del perfil o perfiles de las mismas.

La Facultad 8 de la UCI, tiene como segundo perfil, Software Educativo, Calidad y Gestión. Para el desarrollo del software de gestión se trata de optimizar el trabajo mediante la utilización de librerías, CMS y frameworks. Uno de los frameworks utilizados en la universidad para el desarrollo de aplicaciones WEB con el lenguaje de programación PHP es el framework Symfony, sin embargo, los proyectos productivos de la facultad no optan por utilizarlo debido a la falta de conocimientos y complejidad del mismo. Actualmente la facultad no cuenta con un curso optativo que posibilite la capacitación de los estudiantes en el trabajo con este framework, ni con un material interactivo que sirva de apoyo o material de consulta para la preparación de los interesados, solamente existen algunos documentos en formato PDF y documentos WORD que en su mayoría contienen una información extensa y no se ajustan a las necesidades reales de las personas que los consultan.

Todo lo antes mencionado lleva al **Problema científico**: ¿Cómo contribuir a la existencia de materiales interactivos que sirvan de apoyo para impartir el curso optativo Framework Symfony en la facultad 8?

Este problema se enmarca en el **Objeto de estudio**: proceso de producción de software educativo, teniendo como **Campo de acción** el proceso de desarrollo de un software educativo con tecnología multimedia que muestre información inherente al framework Symfony.

El **Objetivo general** de esta investigación es: crear una aplicación educativa usando tecnología multimedia que sirva de apoyo para impartir el curso optativo Framework Symfony en la Facultad 8.

Para dar solución al problema se plantean los siguientes **Objetivos específicos**:

1. Recopilar la información inherente al contenido básico del framework Symfony.
2. Identificar la metodología y herramienta a utilizar para el desarrollo de la aplicación.
3. Realizar el análisis, diseño e implementación de un sistema desarrollado con tecnología multimedia.

Para dar cumplimiento a los objetivos planteados se proponen las siguientes **Tareas de investigación**:

1. Recopilar la información inherente al framework Symfony que va a ser incluida en la aplicación.
2. Estudiar las tendencias actuales para el desarrollo de aplicaciones con tecnología multimedia.
3. Estudiar las metodologías para desarrollar software con tecnología multimedia e identificar la más recomendable en el desarrollo de la aplicación que se desea implementar.
4. Realizar un estudio sobre las herramientas para el desarrollo de productos con tecnología multimedia.
5. Realizar el análisis y diseño del software que se desea implementar.
6. Realizar la implementación del software.

La **Idea a defender** del presente trabajo es: si se desarrolla una aplicación interactiva, usando tecnología multimedia con contenidos referentes al framework Symfony, se apoyará el curso optativo Framework Symfony que se imparta en la Facultad 8.

El presente trabajo está constituido por 4 capítulos, los cuales se presentan a continuación:

Capítulo 1: Fundamentación teórica. En este capítulo se resume el estudio realizado sobre software con tecnología multimedia, se aborda el tema de las metodologías de desarrollo de software y de los lenguajes de modelado y se fundamenta su selección para el proceso de desarrollo. También se tratan aspectos como las herramientas de desarrollo, así como los lenguajes de programación utilizados.

Capítulo 2: Descripción de la solución propuesta. En este capítulo se describe la solución propuesta para el problema planteado mediante la modelación del dominio para el software, identificándose los principales conceptos relacionados con el tema. Se describen además los requisitos establecidos por el cliente y se definen la Vista de Gestión del Modelo, los Diagramas de Estructura de Navegación y los Diagramas de Estructura de Presentación, en conjunto con las vistas de presentación del software y sus descripciones.

Capítulo 3: Construcción de la solución propuesta. En este capítulo se aborda el tema referente a la construcción de la solución propuesta, mostrando un conjunto de artefactos generados en los flujos de trabajo Análisis y Diseño e Implementación, como los diagramas de clases, de componentes y de despliegue.

Capítulo 4: Estudio de factibilidad. Este capítulo aborda el tema referente al estudio de la factibilidad del producto basado en los métodos Análisis de Puntos Función y COCOMO II.

Capítulo 1. Fundamentación Teórica

1.1 Introducción

El contenido de este capítulo constituye la base teórica de la investigación. En él se pretende abordar los aspectos relacionados con la tecnología multimedia y fundamentar el uso de esta tecnología como solución al problema científico planteado. Se hace un análisis de otras soluciones existentes así como una descripción de las herramientas y metodologías usadas para la creación de este tipo de aplicaciones. Se especifican las herramientas y la metodología seleccionadas para la confección del producto así como el lenguaje de modelado y los lenguajes de desarrollo que se utilizarán.

1.2 Las Tecnologías de la Información y la Comunicación en la educación

Desde sus comienzos el hombre ha estado creando métodos, técnicas y herramientas con el fin de organizar el tipo de sociedad que ha querido. Cada descubrimiento ha traído expectativas, dudas, temores y confusión, y los cambios en las relaciones sociales se han multiplicado. Hoy, es tal el desarrollo de la ciencia y la tecnología que no existe actividad humana en la que las Tecnologías de la Información y la Comunicación (TICs) no sean su principal referente. [1]

En la concepción de la educación como fuente del desarrollo, esta se enfrenta a nuevos desafíos como los de expandir y renovar permanentemente el conocimiento, dar acceso universal a la información y promover la capacidad de comunicación entre individuos y grupos sociales, entre otras cosas. Las políticas que implican la incorporación de las TICs en los establecimientos educacionales –y su utilización efectiva tanto en los procesos de enseñanza/aprendizaje como en la organización de la tarea docente– son una forma de dar respuesta a estos desafíos. [2]

Los recientes avances en tecnología, en especial los relacionados con las ideas de computación, ofrecen un atractivo potencial como medios para llevar a la práctica una visión rica y bastante profunda de la educación. El rápido desarrollo de la tecnología informática y su infiltración en todos los campos de la actividad humana han desencadenado profundos cambios culturales y sociales que requieren ajustes correspondientes en los sistemas educativos y en las habilidades de los estudiantes. [3]

Son innumerables los planteamientos que coinciden en que el desarrollo de las TICs ha significado un paso indispensable en el progreso de la sociedad. Los beneficios que ofrece son aprovechados para el desarrollo de muchos sectores, la educación, es indiscutiblemente uno de estos sectores en que se manifiesta satisfactoriamente su utilización.

Con el uso de las TICs en la educación se puede lograr despertar el interés en los estudiantes y profesores por la investigación científica y posibilitar el mejoramiento de las habilidades creativas, la imaginación, habilidades comunicativas y colaborativas pudiendo acceder a mayor cantidad de información y proporcionando los medios para un mejor desarrollo integral de los individuos. [1]

Ahora que empieza a hablarse de la posibilidad de aprender a lo largo de la vida, de una educación adquirida en cualquier lugar, dentro o fuera de salas de clases, de manera presencial o a distancia, a cualquier hora y a la medida de las necesidades del ciudadano, ¿no resulta acaso más fácil imaginar que nos dirigimos también hacia una compleja y sofisticada industria educacional? [4]

El software educativo como resultado de la aplicación de las TICs en la educación, ocupa un lugar evidente dentro del proceso de enseñanza aprendizaje. Este se puede definir como:

“Cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar aprender y administrar. Un concepto más restringido de Software Educativo lo define como aquél material de aprendizaje especialmente diseñado para ser utilizado con un computador en los procesos de enseñar y aprender”. [5]

1.2.1 La tecnología Multimedia y su aplicación en el software educativo

La palabra multimedia, que ha pasado por distintas acepciones a lo largo de los años, sugiere a la mayoría de las personas la idea de combinar diferentes medios que pueden ser controlados interactivamente por el usuario, y que han de servir para crear aplicaciones de interés educativo, comercial, de entretenimiento o para el trabajo cotidiano. [6]

La investigación educativa viene profundizando en la incidencia que herramientas como los multimedia pueden tener en los procesos de aprendizaje. Los sistemas multimedia favorecen una concentración a

más largo plazo por parte de los alumnos y mejora la constancia para la realización de tareas complejas. [7]

Los materiales multimedia se han convertido en los nuevos recursos para el profesorado y el alumnado. La labor en los próximos años es elaborar contenidos multimedia adecuados a las distintas etapas educativas y contar con bancos actualizados de dichos contenidos que puedan utilizarse con garantía pedagógica dentro del aula.

En los últimos años, se está dando cada vez mayor importancia al uso de las nuevas tecnologías en la educación. El desarrollo y estandarización del uso de Internet, que no sólo es un medio utilizado en los centros de trabajo sino también en todos los ámbitos sociales de las personas, ha convertido a este medio en un recurso esencial como instrumento en el proceso formativo y educativo.

Nos encontramos en un nuevo panorama de la educación en donde las nuevas posibilidades que ofrecen las nuevas tecnologías (Internet, todos sus recursos, contenidos multimedia, etc.) pueden ser un elemento de ayuda para facilitar la comprensión y el aprendizaje de los contenidos a aprender en las distintas materias o áreas de conocimiento. [8]

Entre las ventajas del software educativo se pueden mencionar:

- Da oportunidad a personas que de otro modo no les sería posible estudiar.
- Facilita la enseñanza individualizada. El alumno no se inhibe de preguntar o de repetir varias veces la misma lección.
- Aumenta la motivación y el gusto por aprender. Promueve la enseñanza y aprendizaje activos, se estimula a buscar el conocimiento.
- Proporcionan una enseñanza modular y muy adaptable.
- Reduce el tiempo de aprendizaje, cada uno va a su ritmo.
- Mejora el aprendizaje, el alumno avanza hacia nuevos temas únicamente cuando domina los que deben precederle.
- Incrementa la retención por la combinación de imágenes, gráficos, etc., así como la presentación de situaciones de la vida real.
- Consistencia pedagógica, la calidad de la instrucción no varía de un momento a otro.
- Facilitan la evaluación. [9]

1.3 Audiencia

La aplicación está dirigida a los estudiantes de la UCI que se interesen por aprender a desarrollar aplicaciones web utilizando el framework Symfony, en especial a los estudiantes de la Facultad 8. Los estudiantes y demás personas que interactúen con la aplicación deberán tener conocimientos sobre el lenguaje PHP, ya que Symfony está desarrollado en su totalidad con PHP 5 y las aplicaciones construidas a partir de dicho framework utilizarán también el lenguaje PHP como lenguaje servidor.

1.4 Análisis de otras soluciones existentes

Luego de realizar una búsqueda sobre soluciones alternativas al problema que se presenta, se puede concluir que no existe un producto interactivo en soporte multimedia que facilite el aprendizaje del framework Symfony. Dentro de las soluciones encontradas, se pueden mencionar: tutoriales, cursos, manuales, presentaciones, libros, pero toda esta información se encuentra dispersa y en ocasiones no está al alcance de todos los interesados, por ejemplo, el sitio oficial del framework Symfony (www.symfony-project.org), cuenta con una excelente documentación, pero es necesario tener una conexión a Internet para acceder a esta, igual sucede con otros sitios web que brindan tutoriales y cursos online. Además, en muchos casos la información contenida en estos materiales es demasiado extensa y no se ajusta a los objetivos perseguidos por los estudiantes y algunos de los materiales que están avalados no son accesibles a los estudiantes de la universidad porque hay que pagarlos.

1.5 Metodologías de Desarrollo de Software

Cuando se inicia el desarrollo de un software, siempre surge la pregunta ¿Qué metodología se debe utilizar? Esta es definitivamente, una pregunta muy importante. Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. En la actualidad se cuenta con las metodologías tradicionales o robustas pensadas para el uso exhaustivo de documentación durante todo el ciclo del proyecto y con las metodologías ágiles que ponen vital importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y en mantener una buena relación con el cliente.

El éxito del producto depende en gran parte de la metodología escogida por el equipo, ya sea tradicional o ágil, donde los equipos maximicen su potencial, aumenten la calidad del producto con los recursos y tiempos establecidos. [10]

Su finalidad es garantizar la eficacia, cumpliendo los requisitos iniciales y la eficiencia, minimizando las pérdidas de tiempo en el proceso de desarrollo del software. Todo desarrollo de software es riesgoso y difícil de controlar, si no se lleva una metodología de por medio, lo que se obtendrá, en la mayoría de los casos, será clientes y desarrolladores insatisfechos con el resultado.

Algunas de las metodologías existentes en la actualidad son:

- Proceso Unificado de Rational (RUP)
- Programación Extrema (XP)

1.5.1 Proceso Unificado de Rational (RUP)

RUP es un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational Software, está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación. [10]

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en cuatro fases el desarrollo del software: inicio, elaboración, construcción y transición, donde propone que cada una de ellas se desarrolle en iteraciones y se basa en dos disciplinas: la Disciplina de Desarrollo, en la que se encuentran los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación y Pruebas y la Disciplina de Soporte que incluye los flujos Configuración y administración del cambio, Administración del proyecto y Ambiente [Anexo 1]. RUP brinda un seguimiento detallado en cada una de las fases, funciona bien en proyectos de innovación y sigue los pasos intuitivos necesarios a la hora de desarrollar el software.

Esta metodología tiene tres características que le otorgan una significativa importancia: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

Estos conceptos - los de desarrollo dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental - son de igual importancia. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. [11]

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

1.5.2 Extreme Programming (XP)

Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar. [12]

XP es la más destacada de los procesos ágiles de desarrollo de software formulada por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. [10]

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos de corto plazo y corto equipo. La metodología consiste en una programación rápida o extrema, se basa en pruebas unitarias para ver las fallas que pudieran ocurrir en la refabricación, que no es más que la reutilización de código y en la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo.

Propone empezar en pequeño y añadir funcionalidad con retroalimentación continua, con ella, el manejo del cambio se convierte en parte sustantiva del proceso, el costo del cambio no depende de la fase o etapa, no introduce funcionalidades antes que sean necesarias y el cliente o el usuario se convierte en miembro del equipo. [13]

La metodología XP, como todas las metodologías ágiles existentes hoy, basa sus principios en el Manifiesto Ágil, por lo que cumple que:

- Los individuos e interacciones son más importantes que los procesos y herramientas.
- Software que funcione es más importante que documentación exhaustiva.
- La colaboración con el cliente es más importante que la negociación de contratos.
- La respuesta ante el cambio es más importante que el seguimiento de un plan. [14]

1.5.3 Metodología de desarrollo seleccionada

En el estudio para la determinación de la metodología a usar, se distinguieron ventajas de RUP sobre XP como: el establecimiento temprano de una arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento; su enfoque basado en modelos permite un buen entendimiento entre clientes y desarrolladores facilitando la obtención de un producto con altos niveles de calidad; otra ventaja es su enfoque iterativo: la metodología parte de que se trabajará en iteraciones cortas en tiempo y con metas muy claras.

Por otra parte, la metodología XP a pesar de sus ventajas como metodología ágil, no se ajusta al proceso actual, en primer lugar, exige que el cliente forme parte del equipo de desarrollo, requisito que no puede ser satisfecho ya que el mismo cuenta con un alto contenido de trabajo que no le permite integrarse al equipo de desarrollo, otro motivo que desfavorece el uso de esta metodología es que los requerimientos del sistema a desarrollar son totalmente estables, por lo que no haría falta el uso de XP, tampoco produce una potente documentación del sistema, lo que dificultaría el proceso de mantenimiento.

Siguiendo los aspectos mencionados anteriormente, la metodología RUP es la más idónea para guiar el proceso de desarrollo del software, además que por sus características se ajusta a las necesidades del producto que se va a desarrollar.

Como parte del proceso de desarrollo utilizando esta metodología se hace necesario el uso de un lenguaje de modelado para la modelación de los artefactos generados, motivo por el cual se realiza un estudio para seleccionar el más adecuado.

1.6 Lenguajes de modelado

Desde los inicios, en la informática se han estado utilizando distintas formas de representar los diseños de un modo más bien personal o con algún modelo gráfico. Como resultado, en el desarrollo de los proyectos de software es común la confusión y las malas interpretaciones en los requerimientos de los clientes y los usuarios. La falta de estandarización en la manera de representar gráficamente un modelo impedía una fácil comprensión del trabajo por parte de los desarrolladores que no estaban vinculados de manera activa en el proceso hasta ese momento, además de que no se ponían de acuerdo en lo que realmente estaban elaborando, haciendo difícil el trabajo en equipo e influyendo negativamente en el tiempo de desarrollo. De esta necesidad, es que surgen los lenguajes de modelado.

1.6.1 UML: Lenguaje Unificado de Modelado

Tal como indica su nombre, UML es un lenguaje de modelado. Un modelo es una simplificación de la realidad. El objetivo del modelado de un sistema es capturar las partes esenciales del sistema. Para facilitar este modelado, se realiza una abstracción y se plasma en una notación gráfica. Esto se conoce como modelado visual. [15]

UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Incluye aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.

- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: Relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones. [15]

El lenguaje UML se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software pero no especifica en sí mismo qué metodología o proceso usar.

1.6.2 OMMMA – L: Lenguaje para la Modelación Orientada a Objetos de Aplicaciones Multimedia

En los años recientes, varios lenguajes de modelación orientado a objetos han surgido de los cuales UML es uno de los más aceptados por la comunidad desarrolladora de sistemas informáticos de todo tipo. UML no soporta todos los aspectos de las aplicaciones multimedia de una forma adecuada. Especialmente, las características del lenguaje para modelar los aspectos de la interfaz de usuario, no se aplican explícitamente en los entornos multimedia. [16]

El Lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia (OMMMA - L) se lanza como una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos, y MVC (Modelo Vista Controlador) para la interfaz de usuario. [17]

OMMMA-L está sustentado en cuatro vistas fundamentales, donde cada una se asocia a un tipo de diagrama en particular. Estas vistas son:

- Vista Lógica: modelada a través del Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML, utilizando las mismas notaciones, pero incorporando las clases correspondientes a las

medias: media continua y media discreta, generalizadas en una clase medias. Divide en dos áreas dicho diagrama: una para la jerarquía de los tipos de media y otra para la modelación de la estructura lógica del dominio de la aplicación.

- Vista de Presentación espacial: modelada a través de los Diagramas de Presentación de OMMMA-L, los cuales son de nueva aparición en la extensión de UML, dado que este último no contiene un diagrama apropiado para esta tarea. Estos diagrama 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 (scrolls, barras de menú, botones, campos de entrada y salida, hipertextos con hipervínculos). Estos diagramas de presentación pueden ser divididos en capas virtuales de presentación donde en cada uno de ellas sólo se haga referencia al desarrollo de una clase específica de componentes (por ejemplo, una vista para los objetos de visualización y otra para los de interacción, u otro tipo de división para la representación de los intereses de los desarrolladores).
- Vista de Comportamiento temporal predefinido: modelada por el Diagrama de Secuencia de OMMMA-L, extendido a partir del diagrama de secuencia de UML. El Diagrama de secuencia modela una secuencia de una presentación predefinida dentro de una escena, donde todos los objetos dentro de un diagrama se relacionan al mismo eje del tiempo. En este diagrama se hace un refinamiento del eje del tiempo con la introducción de marcas de tiempo a través de diferentes tipos de intervalos; marcas de inicio y fin de ejecución que permite soportar su reusabilidad; marcas de activación y desactivación de demoras en objetos de tipo media, posibilitando la modelación de las tolerancias de la variación de las restricciones de sincronización para los objetos media; activación compuesta de objetos media para la agrupación de objetos concurrentemente activos.
- Vista de Control Interactivo: modelado a través del Diagrama de Estado, extendido a partir del diagrama de estado de UML, sintácticamente igual a este último, mas con la diferencia semántica de que en el orden de unir los controles interactivos y predefinidos, no interrumpidos de los objetos, las acciones internas de estados simples tienen que llevar nombres de diagrama de secuencia en vez de diagramas de estado empotrados; queriendo esto decir que el comportamiento especificado por el diagrama de secuencia se provoca automáticamente cuando se entra al estado correspondiente donde se hace referencia. [18]

Se puede decir entonces que OMMMA-L no es un lenguaje nuevo, sino una extensión del UML por lo que no es necesario aprenderlo, sino interpretar las características extendidas, centrados a la lógica de funcionamiento de una multimedia, que es por lo general, sencilla. Es robusto y altamente descriptivo, el proceso se refleja en todas sus etapas y hereda de RUP el ciclo de vida basado en iteraciones y el flujo de trabajo iterativo e incremental, centrado en casos de uso y en la arquitectura. [18]

1.6.3 ApEM – L: Lenguaje para la Modelación de Aplicaciones Educativas Multimedia

El impacto de las Tecnologías de la Información y las Comunicaciones (TIC) ha propiciado grandes y vertiginosos cambios en la modelación de los sistemas informáticos, incluso del software educativo. La UCI se ha fusionado hace algunos años a este amplio y competitivo mercado. Los estudios que han sido realizados sobre la documentación ingenieril de las soluciones que se presentan para este tipo de proyectos demuestran, además de su escasez, una generalidad en el uso de notaciones pobres y débiles en la representación de los requerimientos pedagógicos y cargadas con el uso de guiones y mapas de navegación que introducen ambigüedades en la interpretación por los equipos de desarrollo. [19]

ApEM-L 1.0 (Lenguaje para la Modelación de Aplicaciones Educativas Multimedia) está basado en el lenguaje de modelación UML, tomando elementos representativos de extensiones del mismo como las desarrolladas por (Berner, y otros, 2000), (Hennicker, y otros, 2000) y (Sauer, y otros, 2001) y descansa toda su estructura sobre los elementos planteados por el estándar OCL, en su versión actualizada 2.0 del 2003. Consta de tres áreas fundamentales: de estructura lógica, de comportamiento dinámico y de gestión del modelo, y a su vez de cuatro vistas distribuidas en dichas áreas: estática, de arquitectura, de comportamiento y de presentación; para la modelación de los productos, a través de un conjunto de diagramas distribuidos por cada una de estas vistas.

En la Vista Estática y de Arquitectura, ApEM-L 1.0 extiende el lenguaje base en el cual el diagrama de clases fue el artefacto principal de tales modificaciones, así como el diagrama de componentes que no modifica la semántica del lenguaje base y solamente incorpora elementos de organización en paquetes asociados al patrón arquitectónico Modelo-Vista-Controlador-Entidad (MVC-E) como una modificación de la variante MVCmm en el que está basado OMMMA-L. En la vista de comportamiento de todos sus diagramas solo el de secuencia fue el punto de mira de una sencilla modificación sin afectar la semántica

del mismo incorporando el estereotipo restrictivo asociado al tiempo y su representación siempre que sea necesario y para enriquecer la descripción funcional de este tipo de aplicaciones. Por último se crea una nueva vista en ApEM-L 1.0 llamada: vista de presentación y que ha sido incorporada completamente al lenguaje base de UML, para permitir utilizar la semántica original de dicho lenguaje en la construcción de estructuras lógicas de presentación y navegación, (...) construyendo nuevos diagramas dentro de la misma tal es el caso de: el diagrama de estructura de navegación y el diagrama de estructura de presentación, donde se enriquece el concepto de clase, definiendo una segunda clasificación sobre el concepto original. [20]

Necesariamente la implantación inminente y puesta en marcha de ApEM-L 1.0 en entornos productivos de las instituciones que hoy en día son productoras del SWE, ha permitido de alguna manera probar los conceptos manejados del lenguaje, dando paso al surgimiento de nuevas problemáticas y necesidades en la comprensión de sus artefactos en los diferentes equipos de desarrollo de software de estas empresas. Eso trajo consigo presentar algunas modificaciones importantes a ApEM-L 1.0 que posibilitara una mejora potencial para el ámbito productivo del SWE cubano y dar paso a una nueva versión del mismo (ApEM-L 1.5). [19]

El lenguaje de modelado ApEM-L en su nueva versión (ApEM-L 1.5), incorpora los siguientes cambios:

- Surgimiento del artefacto Vista de Presentación.
- La Descripción Textual del artefacto Vista de Presentación.
- La eliminación de los artefactos Caso de Uso y Diagrama de Casos de Uso.
- Incorporación del artefacto Actor al Diagrama de Estructura de Navegación.
- Incorporación del estereotipo para el tratamiento de la condición de transición.
- Una nueva área en ApEM-L: El área de Presentación.

Los cambios realizados a la versión inicial de ApEM-L, sobre la base del tratamiento más profundo de los elementos ingenieriles del software educativo, permiten hoy que aspectos como: interfaces de comunicación, transiciones de estado, concurrencia y la descripción multimedia de los sucesos y eventos, sean abordados más directamente, permitiendo mejores modelaciones del comportamiento de este tipo de aplicaciones. Aún se continúa trabajando en estereotipos restrictivos que incorporen mayor cantidad de

tratamiento de situaciones desde los conceptos abstractos de la programación y permitan mejoras sustantivas y escalables en las nuevas versiones. [19]

1.6.4 Lenguaje de Modelado seleccionado

El lenguaje seleccionado para la modelación de los artefactos producidos en la construcción del software, es ApEM-L 1.5, ya que los demás lenguajes de modelado estudiados no son capaces de modelar con claridad el proceso de desarrollo de un producto multimedia educativo. Otras razones que influyeron en esta selección son: es un lenguaje extensivo de UML por lo que incorpora sus fundamentos y estilos de trabajo; incorpora los elementos más significativos de otras extensiones de UML como OMMMA-L y respeta el estándar establecido por el Lenguaje de Restricción de Objetos (OCL), lo que permite modelar aplicaciones que utilicen el Paradigma Orientado a Objetos; permite usar para su modelado cualquiera de las herramientas CASE existentes en la actualidad para el modelado con UML; incluye nuevos elementos de modelado que permiten representar componentes de una interfaz de comunicación con el usuario y que ayudan tanto a los desarrolladores como a todo el equipo de trabajo a entender y familiarizarse más con la aplicación, con su entorno y funcionalidades. Además incorpora los diagramas de Estructura de Navegación y de Estructura de Presentación, con los que resuelve uno de los problemas existentes en la modelación de aplicaciones de este tipo: los elementos de presentación y navegación, que son de gran importancia para el desarrollo de estas aplicaciones.

1.7 Herramientas de autor

En los últimos años, todos han sido testigos del rápido desarrollo que han experimentado las herramientas de creación multimedia. Realizar gráficos, editar videos, añadir interacción o publicar en la red se han convertido en actividades al alcance de los usuarios, que disponen para ello de sencillas herramientas en el escritorio de su ordenador personal.

La facilidad para la creación de contenidos ha de ser visualizada como un elemento más en la emergencia de una sociedad dirigida por el conocimiento, en la que cada vez se requieren más personas generadoras de información, que como consecuencia se convierten en productores, más o menos conscientes, de contenidos multimedia. [6]

Las herramientas de autor son programas destinados a la creación de nuevos contenidos en formato multimedia, es decir, son software que manejan elementos medias, vinculados a la programación para lograr la funcionalidad interactiva de un producto multimedia. Las herramientas de autor permiten la generación de aplicaciones que funcionan de manera independiente al software que las creó.

1.7.1 Macromedia Director

La herramienta Director es un programa de autor de fácil manejo. Permite la combinación de texto, gráficos, sonido, animación y vídeo en un documento que se reproduce en el ordenador y que es presentado con múltiples detalles. [21]

Facilita el desarrollo de aplicaciones multimedia basándose en la metáfora del cine. Por esta razón, el desarrollador que realiza un proyecto con esta herramienta tiene que pensar en los elementos que participan en una película, tales como el escenario (espacio en el que se desarrollan las acciones), el reparto de actores (contenidos multimedia utilizados en la aplicación) y el guión, que define cuándo y dónde deben intervenir cada uno de los actores en el escenario, describiendo así las acciones que tendrán lugar en la película (aplicación multimedia). Todos estos elementos se pueden gestionar de forma totalmente visual o también de forma eficiente mediante Lingo, el lenguaje de programación que incorpora la herramienta. Director es ampliamente utilizado para construir todo tipo de aplicaciones multimedia interactivas. [22]

Este software permite generar presentaciones multimedia en archivos ejecutables que pueden ser distribuidas a través de CDs. Permite incorporar a las películas formatos, como imágenes JPEG, BMP, PNG, GIF, videos en formato MOV, AVI, sonidos WAV, AIFF o animaciones Flash. Incluye además editores básicos para texto, mapa de bits, vectores, sonido. Para la reproducción de sus ejecutables utiliza el reproductor Shockwave y como una de sus principales ventajas se puede mencionar el uso de los XTRAS, programas desarrollados por terceros, que proporcionan al usuario final múltiples utilidades.

1.7.2 ToolBook Instructor

ToolBook Instructor es una herramienta creada por la compañía Asymetrix que utiliza la metáfora del libro

como base para el desarrollo de aplicaciones, de tal forma que cualquier aplicación está formada por uno o más ficheros denominados libros. Al igual que un libro, los libros de ToolBook también están divididos en páginas, representando cada una de ellas una ventana de la aplicación. Cada página contiene campos, botones y gráficos, tanto las páginas como los componentes contenidos en ellas son denominados objetos. Posee un asistente para la creación de Libros llamado Book Specialist. Este asistente va guiando paso a paso al usuario por la creación de una nueva aplicación y una de las características más interesantes de este asistente es que permite ser configurado al gusto del desarrollador, permitiendo que la próxima vez que se cree un proyecto el asistente trabaje acorde a lo que se haya especificado. ToolBook también incorpora un potente lenguaje de programación basado en objetos denominado OpenScript. [22]

ToolBook puede publicar los proyectos en Internet de dos formas diferentes: HTML y Neuron. Permite la exportación directa de los proyectos a HTML, proporcionando así compatibilidad absoluta con cualquier navegador. Incluso más allá del HTML, ToolBook da la posibilidad de habilitar Java en aquellas partes en las que sea posible su utilización.

Para poder visualizar el contenido en otros ordenadores que no poseen ToolBook se proporciona un reproductor gratuito que trabaja sobre cualquier Windows. El inconveniente es, que únicamente existe reproductor para plataformas Windows.

Una de las cosas que se echan en falta en ToolBook son los editores de recursos. Únicamente se incluye un editor de mapas de bits que, por otro lado, es muy básico. [23]

1.7.3 Macromedia Flash 8

Flash debe sus raíces a una pequeña compañía llamada FutureSplash que fue adquirida por Macromedia en 1997 para complementar su programa Director que sirve para la creación de producciones multimedia interactivas, títulos de CD/DVD, etc., cuando deseaban darle un enfoque para el Web. [24]

Flash diseña gráficas de vectores; gráficas definidas como puntos y líneas en lugar de píxeles. Es decir que los vectores son como un conjunto de instrucciones matemáticas que por medio de valores le dan forma a una imagen. Además de las gráficas vectoriales, Flash permite incluir audio comprimido en

diversos formatos como el mp3, importar gráficas creadas con otros programas, formularios. Todo esto definido al igual que los vectores por un conjunto de instrucciones que mueven los objetos de posición y forma, y que dan como resultado archivos muy pequeños que se cargan en poco tiempo. [24]

Al igual que herramientas como Macromedia Director, Flash utiliza para la reproducción de sus películas el reproductor Flash Player y el lenguaje de programación ActionScript para controlar el comportamiento de los elementos que intervienen en estas películas.

1.7.4 Herramienta seleccionada

La herramienta seleccionada para la elaboración del producto multimedia es Macromedia Flash 8. Esta selección está justificada por el hecho que propicia la generación de una aplicación capaz de incorporar grandes volúmenes de datos con un tamaño resultante pequeño en comparación con las demás herramientas estudiadas. La libertad de su lenguaje de programación (ActionScript 2.0) es de vital importancia ya que los desarrolladores del producto a implementar, trabajan siguiendo el paradigma de la Programación Orientada a Objetos (POO), facilidad que ofrece ActionScript 2.0, además que la herramienta cuenta con un asistente para dicho lenguaje a diferencia de ToolBook Instructor y Director. Otro aspecto medido en la elección de la herramienta que constituye una ventaja sobre las estudiadas, es la posibilidad de crear diseños más atractivos y poseer una mayor potencia en la realización de animaciones, gracias a funcionalidades añadidas a esta versión donde destacan los filtros de efectos gráficos, modos de mezcla, suavizado de mapas de bits y caché de mapa de bits en tiempo de ejecución. También, las aplicaciones generadas con esta herramienta pueden ser visualizadas en cualquier plataforma, mientras que ToolBook Instructor es dependiente de la plataforma Windows. El reproductor de Flash (Flash Player) a diferencia del reproductor de Director, según estadísticas de Adobe, está instalado en el 98% de los ordenadores con acceso a internet mientras que el reproductor Shockwave en un 63%, lo que evidencia una mayor difusión del Flash Player.

Como la herramienta seleccionada es Macromedia Flash 8 se utilizará su lenguaje de programación ActionScript 2.0 y el lenguaje XML para la carga dinámica y almacenamiento de los contenidos.

1.8 Lenguajes utilizados en el desarrollo del producto multimedia

1.8.1 ActionScript 2.0

ActionScript es el lenguaje de scripts integrado en Flash, que posibilita el desarrollo de complejas aplicaciones multimedia y sitios web dinámicos bajo esta tecnología. Su campo de aplicación crece cada día, abarcando aplicaciones tan diversas como el desarrollo de juegos, simulaciones, presentaciones interactivas y animación dinámica con sorprendentes efectos visuales.

La facilidad de uso y aprendizaje del lenguaje ActionScript, junto a una perfecta integración con las herramientas de diseño de Flash, han abierto las puertas de la programación para un importante número de diseñadores, permitiéndoles un control absoluto sobre sus proyectos y ampliando sus horizontes.

La versión 2.0 de ActionScript implementa una estructura más fiel al modelo de Programación Orientada a Objetos, incorporando un nuevo set de elementos y características especiales que permiten al programador definir las clases y operar en base a ellas de una manera más práctica y adecuada.

El éxito de Flash no radica únicamente en su facilidad para la creación y edición de gráficos y animaciones vectoriales. Gracias a la incorporación de un potente lenguaje de scripts como lo es ActionScript, ha pasado a formar parte de una lista de herramientas imprescindibles para el desarrollo web y multimedia. [25]

1.8.2 XML

XML, es el estándar de Extensible Markup Language. XML no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados. [26]

Las ventajas que nos provee XML es que la flexibilidad que nos ofrece es altamente útil para diferentes propósitos, la información se nos provee de manera estructurada y descriptivamente visual, la información ya no es un conjunto de datos sin sentido, que solo entendía el sistema para el cual es enviada. Ahora puede ser leída tanto por aplicaciones ajenas a la aplicación que las envía (o provee) así como por

"humanos", vale decir que la información es accesible también para los desarrolladores y lo más importante que es, entenderla. [27]

Dentro de las principales características asociadas con XML se pueden destacar las siguientes:

- Es una arquitectura más abierta y extensible. No se necesitan versiones para que puedan funcionar en futuros navegadores. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en internet/intranet por medio de un validador de documentos (parser).
- Integración de los datos de las fuentes más dispares. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en el propio PC como en una red local o extensa.
- Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje nos permitirá agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.
- Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido web en XML consigue que la estructura de la información resulte más accesible. [26]

1.9 Conclusiones del capítulo

Analizando lo antes expuesto se puede concluir que la UCI, no cuenta con un producto que agrupe la documentación existente sobre el framework Symfony y que ajuste dicha información para la fácil comprensión de sus estudiantes. Se definieron las herramientas y los lenguajes a ser usados para la elaboración del producto. Como herramienta de autor se utilizará Macromedia Flash 8, determinado por ésta, el lenguaje ActionScript 2.0 y el XML para la carga dinámica y almacenamiento de datos. Para guiar el proceso de desarrollo de la aplicación con tecnología multimedia se determinó el uso de la metodología RUP, utilizando como lenguaje de modelado ApEM-L 1.5.

Capítulo 2. Descripción de la solución propuesta

2.1 Introducción

Este capítulo se centra en la descripción de la solución propuesta para resolver el problema planteado en la investigación, teniendo como guía de desarrollo el Proceso Unificado de Rational. Se identificarán los conceptos asociados al dominio y los requisitos funcionales y no funcionales del sistema. También se modela el dominio a partir de los conceptos identificados, se representa la Vista de Gestión del Modelo, se realizan los Diagramas de Estructura de Navegación (DEN) y los Diagramas de Estructura de Presentación (DEP) y se describen las vistas de presentación identificadas en el producto.

2.2 Modelo de Dominio

Existen varias alternativas para llevar a cabo el modelamiento de un sistema, RUP, como metodología de desarrollo propone la realización de un modelo de negocio para el caso en el que los procesos dentro del entorno estén claramente definidos, y la realización de un modelo de dominio para los escenarios en los que no puedan identificarse tales procesos del negocio.

Para la modelación del sistema con tecnología multimedia propuesto no se lograron definir procesos específicos en el entorno, solamente identificar conceptos y objetos relacionados con el mismo, por ello se propone la realización de un modelo de dominio.

El Modelo de Dominio (o Modelo Conceptual) es una representación visual de los conceptos u objetos del mundo real, significativos para un problema o área de interés. Captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Representa conceptos del mundo real, no de los componentes de software. [11]

2.2.1 Identificación de los conceptos del Dominio

- **Curso Optativo Symfony:** Se le llama así a la asignatura opcional donde se imparten los contenidos relacionados con el framework Symfony.
- **Usuario:** Se le denomina así a todas las personas involucradas en el curso optativo, estudiantes y profesores.
- **Materiales:** Conjunto de medios informativos que consultan los usuarios y sirven de apoyo al curso optativo.
- **Ejercicios:** Son las evaluaciones realizadas por los usuarios para comprobar sus conocimientos.
- **Videos:** Conjunto de presentaciones que describen algunos temas abordados en los contenidos.
- **Contenido:** Representa la información textual que compone los materiales.
- **Imagen:** Representa la información gráfica que compone los materiales.

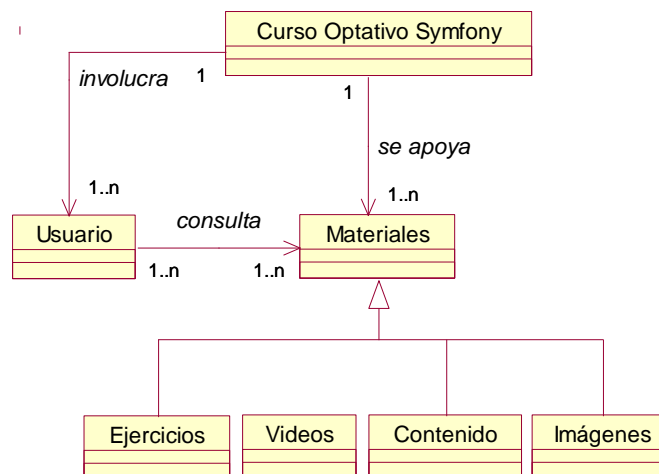


Figura 1. Modelo de Dominio

2.3 Solución propuesta

Para dar solución a la problemática planteada se decide desarrollar una aplicación interactiva con tecnología multimedia que muestre contenidos sobre el framework Symfony a un nivel básico, que llevará como título: Multimedia sobre el framework Symfony. Con ésta, se garantiza que todos los interesados tengan acceso a la información y también una mayor comprensión y asimilación de los contenidos.

La aplicación estará constituida por seis temas: Introducción al Symfony, Comenzando el proyecto, Configuración de Symfony, Cómo implementa Symfony el MVC, Trabajo con formularios y Proceso de despliegue de la instalación. Estos temas están conformados por subtemas, los cuales contienen la información que será visualizada. El sistema permite acceder a los contenidos a través de un menú “Temas”, el cual dispone de cada uno de los temas y subtemas mencionados anteriormente; permite imprimir el contenido del subtema que se esté mostrando, visualizar las imágenes asociadas al subtema así como una descripción de algunas palabras o frases del contenido, a través de “palabras especiales” en el texto de los subtemas (palabras calientes) y en el caso de las imágenes también es posible a través de un menú “Imágenes”. Brinda la opción de escuchar un tema musical de fondo, dando la posibilidad de desactivarlo o activarlo cuando se desee. Cuenta también con un “Glosario”, una “Galería de Videos”, “Ejercicios”, accesibles desde un menú “Opciones” y una “Ayuda” que muestra algunos detalles relacionados con la navegación de la aplicación.

2.4 Requerimientos del Sistema

La captura de requisitos es el proceso de averiguar, normalmente en circunstancias difíciles, lo que se debe construir. [11]

Antes de comenzar el diseño y la implementación de cualquier sistema, es necesario saber qué debe hacer dicho sistema y cómo debe hacerlo, dejar bien claro lo que los clientes desean del sistema a desarrollar. Para obtener exitosamente esta información y desarrollar con calidad el producto, RUP como parte del proceso de desarrollo de software cuenta con el Flujo de Trabajo Requerimientos.

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

Los **requerimientos funcionales** definen las condiciones que el sistema debe cumplir, o sea, las funciones que debe realizar para proporcionar un determinado resultado al usuario.

Los **requerimientos no funcionales** responden a las cualidades o propiedades que el sistema debe tener, es decir, son las características que permiten obtener un producto atractivo, usable, rápido o confiable. Pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

2.4.1 Requisitos Funcionales del Sistema

RF 1: Mostrar el contenido de la aplicación.

- RF 1.1: Mostrar la presentación de la aplicación.
- RF 1.2: Mostrar la bienvenida de la aplicación.
- RF 1.3: Mostrar el contenido en las diferentes pantallas según el subtema.
- RF 1.4: Mostrar las imágenes asociadas al subtema.
- RF 1.5: Mostrar las palabras calientes asociadas al subtema.
- RF 1.6: Mostrar los videos contenidos en la galería de la aplicación.
- RF 1.7: Mostrar el glosario.

RF 2: Permitir la navegabilidad dentro de la aplicación.

- RF 2.1: Permitir al usuario acceder a cualquier pantalla de la aplicación independientemente del lugar en que se encuentre.
- RF 2.2: Permitir la navegabilidad entre las pantallas internas de cada tema a través de botones: “Anterior” y “Siguiente”.

RF 3: Realizar acciones generales.

- RF 3.1: Mostrar la ayuda de la aplicación.
- RF 3.2: Permitir activar o desactivar el sonido de fondo de la aplicación.
- RF 3.3: Permitir controlar las opciones del reproductor de video: reproducir, pausar, detener, navegar y control de volumen.
- RF 3.4: Permitir imprimir el texto del subtema que se esté visualizando.
- RF 3.5: Permitir salir de la aplicación desde cualquier pantalla.
- RF 3.5.1: Mostrar confirmación de salida de la aplicación.
- RF 3.5.2: Mostrar créditos una vez confirmada la salida de la aplicación.

RF 4: Permitir interactuar con los ejercicios.

- RF 4.1: Mostrar los ejercicios organizados por temas.
- RF 4.2: Permitir realizar los ejercicios según la selección del usuario.
- RF 4.3: Mostrar evaluación de los ejercicios.

2.4.2 Requisitos no Funcionales del Sistema

Requerimientos de apariencia o interfaz externa:

- El sistema tendrá una interfaz sencilla, intuitiva y amigable para sus usuarios.
- El sistema proporcionará claridad y correcta organización de la información, permitiendo la interpretación inequívoca de esta, facilitando el uso de la aplicación.
- El sistema implementará la ejecución de acciones de una manera rápida, minimizando los pasos a ejecutar en cada proceso.
- El diseño de la interfaz gráfica deberá garantizar la distinción visual entre los elementos del sistema.

Usabilidad:

- Los usuarios que utilizarán el sistema deberán tener conocimiento básico del manejo de la computadora.
- La aplicación deberá poseer una interfaz y navegación asequibles y funcionales tanto para usuarios expertos como para los que no tienen conocimientos profundos de informática.

Requerimientos de Software:

- Macromedia Flash Player 8.0.

Requerimientos de Hardware:

- Procesador Pentium II, procesador AMD K6 o superiores.
- 64 MB de RAM (Mínimo).
- Kit de Multimedia y Mouse.
- Lector de CD-ROM.

Restricciones en el diseño y la implementación:

- La herramienta de desarrollo de la aplicación será Macromedia Flash 8.0.
- El lenguaje de programación será ActionScript 2.0.
- Se utilizará la Programación Orientada a Objetos.
- Todas las medias de la aplicación deben cargarse de forma dinámica.
- Los datos dinámicos deben almacenarse en ficheros externos como XML o TXT.

2.5 Vista de Gestión del Modelo

Para la modelación del producto multimedia siguiendo la metodología propuesta por ApEM-L 1.5, el primer paso es representar la Vista de Gestión del Modelo (VGM) que contendrá los subsistemas que se

identifiquen en la aplicación además de establecer las dependencias entre estos. Definidos por las características y funcionalidades de las vistas que lo conforman se han identificado 4 subsistemas: Subsistema Introducción, Subsistema Contenidos, Subsistema Opciones Generales y el Subsistema Servicios.

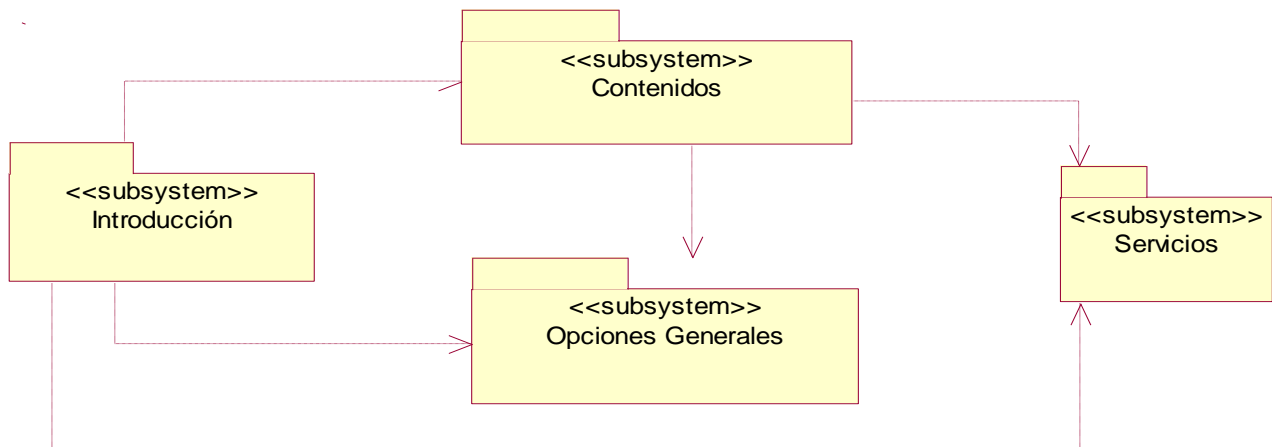


Figura 2. Vista de Gestión del Modelo

2.5.1 Vistas de Presentación

Una vista de presentación es aquel artefacto que recoge la descripción de los elementos y su relación para el cumplimiento de una determinada función del sistema. De ahí que desde el punto de vista estructural una interfaz de usuario esté constituida por varias vistas de presentación y que a su vez una propia vista de presentación puede contener otras en sí. Además toda interfaz gráfica de usuario es a su vez una vista de presentación por el mero hecho de que puede contener todas las otras vistas de presentación en su interior. [19]

A continuación se muestran los subsistemas identificados con las vistas correspondientes a cada subsistema.

Tabla 1: Subsistemas y Vistas de Presentación

Subsistemas	Vistas de Presentación
Subsistema Introducción	Vista "Presentación", Vista "Principal".
Subsistema Contenidos	Vista "Contenido", Vista "Imágenes", Vista "Palabra", Vista "Visor", Vista "Menú_Izquierdo", Vista "Menú_Derecho".
Subsistema Opciones Generales	Vista "GaleriaVideos", Vista "Ejercicios", Vista "Glosario", Vista "Ayuda", Vista "Créditos", Vista "Salir".
Subsistema Servicios	Vista "Servicios".

2.6 Justificación de los actores del sistema

Actores: Usuario

Descripción: Representa a todas las personas que interactúan con la aplicación, personas que serán fundamentalmente estudiantes. Es el que obtendrá los beneficios de la interacción, pues se nutrirá de los contenidos presentes en la misma.

2.7 Descripción textual de las Vistas de Presentación

Tabla 2: Descripción textual de la Vista Presentación

Descripción textual de la Vista de Presentación "Presentación"	
Actores de la Vista de Presentación	Usuario
Propósito	Mostrar al usuario a través de la presentación, la institución a cargo del desarrollo del producto y el nombre del mismo.
Objetivos pedagógicos	-

Resumen		La vista se inicia cuando el usuario ejecuta el fichero que da inicio a la aplicación. El sistema muestra la animación que da comienzo al producto y brinda la opción de saltar presentación, siendo la selección de esta opción la culminación de la vista o el tiempo de finalización de la animación.	
Vistas asociadas		Vista " <i>Principal</i> ".	
Referencias		R1.1	
Precondiciones		Fichero que da inicio a la aplicación ejecutado por el usuario.	
Poscondiciones		Se muestra la presentación de la aplicación.	
Curso normal de eventos			
Acciones del actor	Respuesta del sistema	Elementos de la vista	
1. El usuario ejecuta el fichero que da inicio a la aplicación.	2. Se carga la Vista " <i>Presentación</i> " mostrándose una animación (AnimacionPresentacion). Al concluir la animación de presentación, se muestra la Vista " <i>Principal</i> " de la aplicación concluyendo de esta forma la vista actual.	2.1 La animación se muestra en el centro de la pantalla.	
	3. Se muestra el botón "Saltar Presentación".	3.1 La opción de saltar presentación aparecerá en el borde inferior de la pantalla, centrada y representada por un botón de texto "Saltar Presentación."	
Curso alternativo de los eventos			
Acción	Curso alternativo		
2. El usuario selecciona el botón "Saltar Presentación".	Culmina la vista actual y se muestra la Vista " <i>Principal</i> ".		
Prioridad		Crítica	
Mejoras		-	
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	-	-	-

Video o Animación	AnimacionPresentacion	La animación inicia mostrando el nombre de la universidad, luego este nombre desaparece y muestra el nombre del producto.	Existente
Sonido	-	-	-
Texto	-	-	-
Reglas Pedagógicas			
-			

Tabla 3: Descripción textual de la Vista Principal

Descripción textual de la Vista de Presentación <i>“Principal”</i>		
Actores de la Vista de Presentación	Usuario, Vista <i>“Presentación”</i>	
Propósito	Permitir al usuario visualizar la pantalla principal de la aplicación.	
Objetivos Pedagógicos	Mostrar la bienvenida al usuario e indicarle los primeros pasos para su trabajo con la aplicación.	
Resumen	La vista se inicia cuando el usuario selecciona el botón <i>“Saltar Presentación”</i> de la vista de <i>“Presentación”</i> o cuando finaliza el tiempo de ejecución de la animación de presentación de la misma, se muestra la imagen de fondo y un texto de bienvenida y culmina cuando el usuario a través de la vista <i>“Menú_Izquierdo”</i> , accede a la vista <i>“Contenido”</i> .	
Vistas asociadas	Vista <i>“Menú_Izquierdo”</i> , Vista <i>“Menú_Derecho”</i> , Vista <i>“Servicios”</i> .	
Referencias	R1.2	
Precondiciones	Debe haberse ejecutado la Vista <i>“Presentación”</i> .	
Poscondiciones	Visualizada la Vista <i>“Principal”</i> .	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista

<p>1. El usuario selecciona la opción "Saltar Presentación" de la vista "Presentación" o finaliza la animación de presentación.</p>	<p>2. Se muestra la vista "Principal".</p>	<p>2.1 La vista muestra una imagen de fondo. (Imagen_FondoPrincipal)</p> <p>2.2 Se muestra el texto de bienvenida a la aplicación (Texto_Principal) ubicado en la parte izquierda de la pantalla, centrado verticalmente.</p> <p>2.3 Al iniciarse la vista se reproduce aleatoriamente un tema musical de fondo de cuatro existentes (Musica1, Musica2, Musica3, Musica4), que permanecerá en todas las vistas de la aplicación excepto en la Vista "GaleriaVideos".</p>
	<p>3. Se ejecuta la Vista "Servicios".</p>	<p>3.1 La vista de presentación "Servicios" está contenida dentro de la vista de presentación que se está describiendo y estará presente en el resto de las vistas de la aplicación exceptuando la Vista "Salir" y "Créditos". Está ubicada en la parte superior derecha de la pantalla.</p>
	<p>4. Se ejecuta la Vista "Menú_Izquierdo".</p>	<p>4.1 La vista de presentación "Menú_Izquierdo" está contenida dentro de la vista de presentación que se está describiendo y estará presente en la Vista "Contenido". Está ubicada en la parte</p>

		izquierda de la pantalla, centrada verticalmente.	
	5. Se ejecuta la Vista "Menú _ derecho".	5.1 La vista de presentación "Menú_Derecho" está contenida dentro de la vista de presentación que se está describiendo y estará presente en la Vista "Contenido". Está ubicada en la parte derecha de la pantalla, centrada verticalmente.	
Curso alternativo de los eventos			
Acción	Curso alternativo		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_FondoPrincipal	La imagen es de color carmelita, la parte superior e inferior de la imagen es una franja de color negro. El nombre del producto y el logotipo se muestran en la parte superior izquierda de la imagen y el nombre del framework está ubicado en la parte derecha de la imagen y centrado verticalmente.	Existente
Video o Animación	-	-	-
Sonido	<ul style="list-style-type: none"> - Musica1 - Musica2 - Musica3 - Musica4 	<ul style="list-style-type: none"> - Tema musical "So long my friend". - Tema musical "Face in the photograph". - Tema musical "Almost a whisper". - Tema musical "Enchantment". <p>Todos los temas</p>	Existente

		musicales pertenecen al autor Yanny Chrysomallis.	
Texto	Texto_Principal	Texto que da la bienvenida a la aplicación e informa sobre el objetivo de la misma. El texto estará justificado, la fuente a utilizar es Verdana con un tamaño igual a 11, color negro.	Existente
Reglas Pedagógicas			
-			

Tabla 4: Descripción textual de la Vista Contenido

Descripción textual de la Vista de Presentación “Contenido”	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario visualizar la pantalla de contenidos de la aplicación y así conocer sobre los diferentes temas abordados en la misma.
Objetivos Pedagógicos	Mostrar al usuario el contenido de los temas seleccionados, a través de los cuales realizará su estudio.
Resumen	La vista se inicia cuando el usuario a través de la vista “Menú_Izquierdo” accede a alguno de los temas abordados en la aplicación.
Vistas asociadas	Vista “Menú_Izquierdo”, Vista “Menú_Derecho”, Vista “Servicios”, Vista “Palabra”, Vista “Visor”, Vista “Imágenes”.
Referencias	R1.3, R1.4, R1.5, R2.2
Precondiciones	El usuario debe haber seleccionado uno de los botones subtemas en la vista “Menú_Izquierdo” que conducen a los temas abordados en la aplicación.
Poscondiciones	Vista de Presentación “Contenido” visualizada de acuerdo a la elección del usuario.

Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. El usuario hace clic sobre uno de los botones subtemas contenidos en la Vista "Menú_Izquierdo".	2. Se muestra la Vista "Contenido".	<p>2.1 La vista muestra una imagen de fondo (Imagen_FondoContenido).</p> <p>2.2 Se muestra un texto que contiene información sobre el subtema seleccionado. Está ubicado en el área izquierda de la pantalla, centrado verticalmente. Contiene palabras y frases significativas (palabras calientes), que aparecerán en color rojo y permitirán ser seleccionadas para conocer información adicional sobre ellas.</p> <p>2.3 Se muestran los botones de texto "Siguiete" y "Anterior", ubicados en la parte inferior izquierda de la pantalla.</p>
	3. Se ejecuta la Vista "Imágenes".	3.1 La vista de presentación "Imágenes" está contenida dentro de la vista de presentación que se está describiendo. La misma está ubicada en la parte superior derecha bajo la Vista "Servicios". Contiene las imágenes relacionadas con el contenido del subtema mostrado, en caso que el subtema no presente imágenes esta vista no se muestra.
4. El usuario selecciona el botón "Siguiete".	5. Se actualiza el texto de contenido, mostrando el texto del subtema siguiente y se	

	actualiza la Vista de Presentación “ <i>Imágenes</i> ”.		
6. El usuario selecciona el botón “Anterior”.	7. Se actualiza el texto de contenido, mostrando el texto del subtema anterior y se actualiza la Vista de Presentación “ <i>Imágenes</i> ”.		
8. El usuario selecciona una palabra caliente de tipo texto.	9. Se muestra la Vista de Presentación “ <i>Palabra</i> ”.		
7. El usuario selecciona una palabra caliente de tipo imagen.	8. Se muestra la Vista de Presentación “ <i>Visor</i> ”.		
Curso alternativo de los eventos			
Acción	Curso alternativo		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_FondoContenido	La imagen es de color carmelita y blanco, la parte superior e inferior de la imagen es una franja de color negro. El nombre del producto y el logotipo se muestran en la parte superior izquierda de la imagen y el nombre del framework está ubicado en la parte derecha de la imagen y centrado verticalmente.	Existente
Video o Animación	-	-	-
Sonido	-	-	-
Texto	Texto_Contenido	Texto que brinda información relacionada con el subtema abordado en dicha vista. El texto estará	Existente

		justificado, la fuente del mismo es Verdana con un tamaño igual a 11, color negro.	
Reglas Pedagógicas			
-			

- ✓ Descripción textual de la Vista Menú Izquierdo [Tabla 1, Anexos]
- ✓ Descripción textual de la Vista Menú Derecho [Tabla 2, Anexos]
- ✓ Descripción textual de la Vista Imágenes [Tabla 3, Anexos]
- ✓ Descripción textual de la Vista Palabra [Tabla 4, Anexos]
- ✓ Descripción textual de la Vista Visor [Tabla 5, Anexos]
- ✓ Descripción textual de la Vista Ayuda [Tabla 6, Anexos]
- ✓ Descripción textual de la Vista Ejercicios [Tabla 7, Anexos]
- ✓ Descripción textual de la Vista Galería de Videos [Tabla 8, Anexos]
- ✓ Descripción textual de la Vista Glosario [Tabla 9, Anexos]
- ✓ Descripción textual de la Vista Servicios [Tabla 10, Anexos]
- ✓ Descripción textual de la Vista Salir [Tabla 11, Anexos]
- ✓ Descripción textual de la Vista Créditos [Tabla 12, Anexos]

2.8 Diagramas de Estructura de Presentación

Los Diagramas de Estructura de Presentación (DEP), son otros de los diagramas definidos por ApEM-L, para modelar la estructura de las futuras interfaces de comunicación con el usuario en cuanto a sus componentes.

Permiten una organización lógica de los elementos conformantes de dichas interfaces, dejándole a los diseñadores gráficos la función de decidir dónde y cómo serán en términos visuales dichos elementos.

Para la mejor estructuración del modelo, se realiza una segunda clasificación sobre el concepto original de clase, definiendo dos nuevos tipos de estas: la *clase Estáticos* y la *clase Interacción*, las cuales seccionarán los elementos que cumplan con cada una de las características que denotan los propios nombres. [20]

A continuación se muestran los DEP identificados en la aplicación.

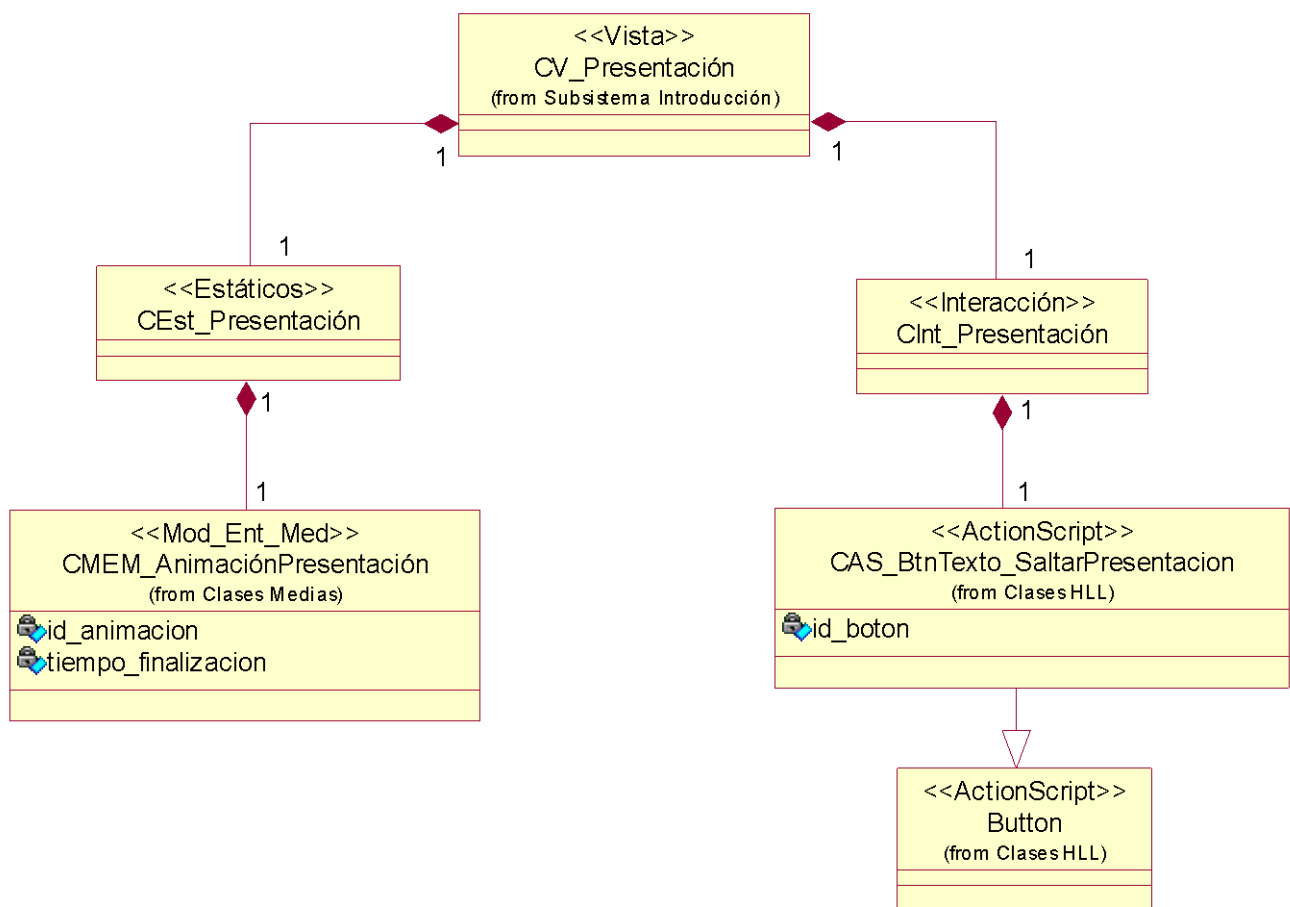


Figura 3. DEP Vista Presentación

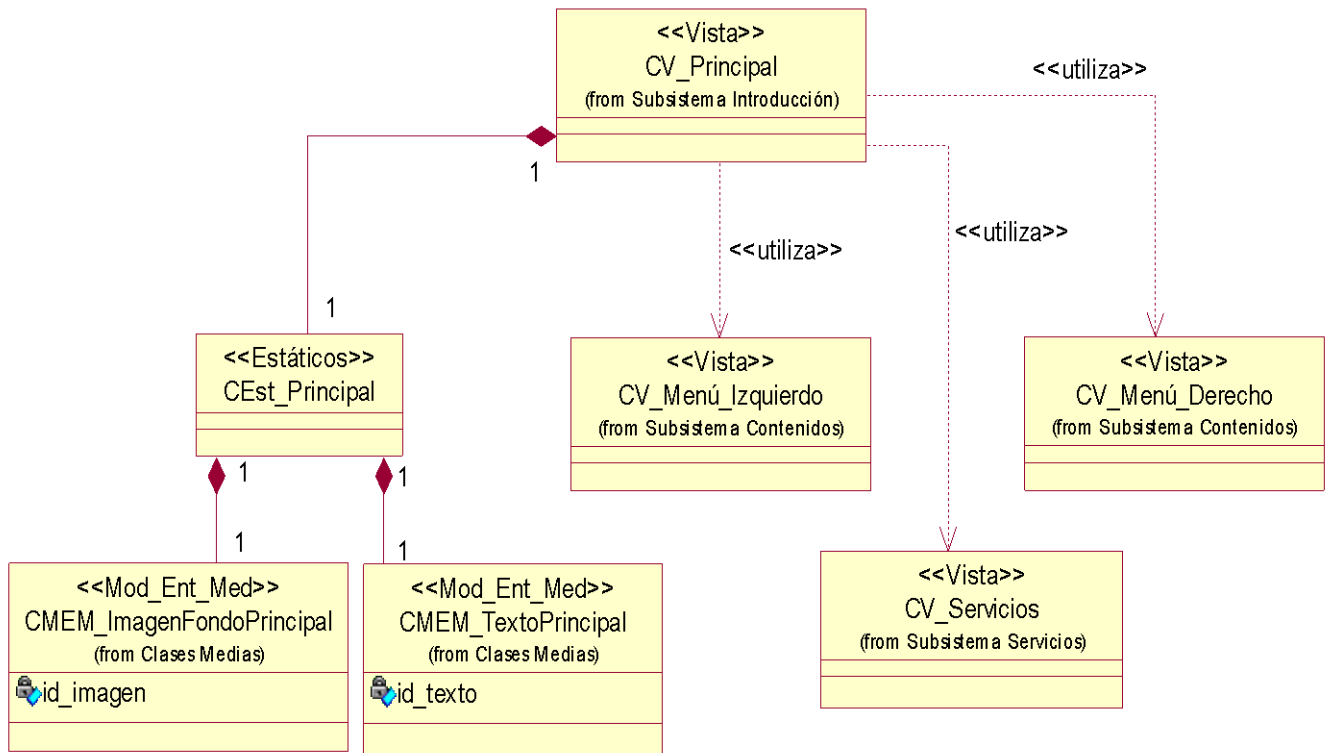


Figura 4. DEP Vista Principal

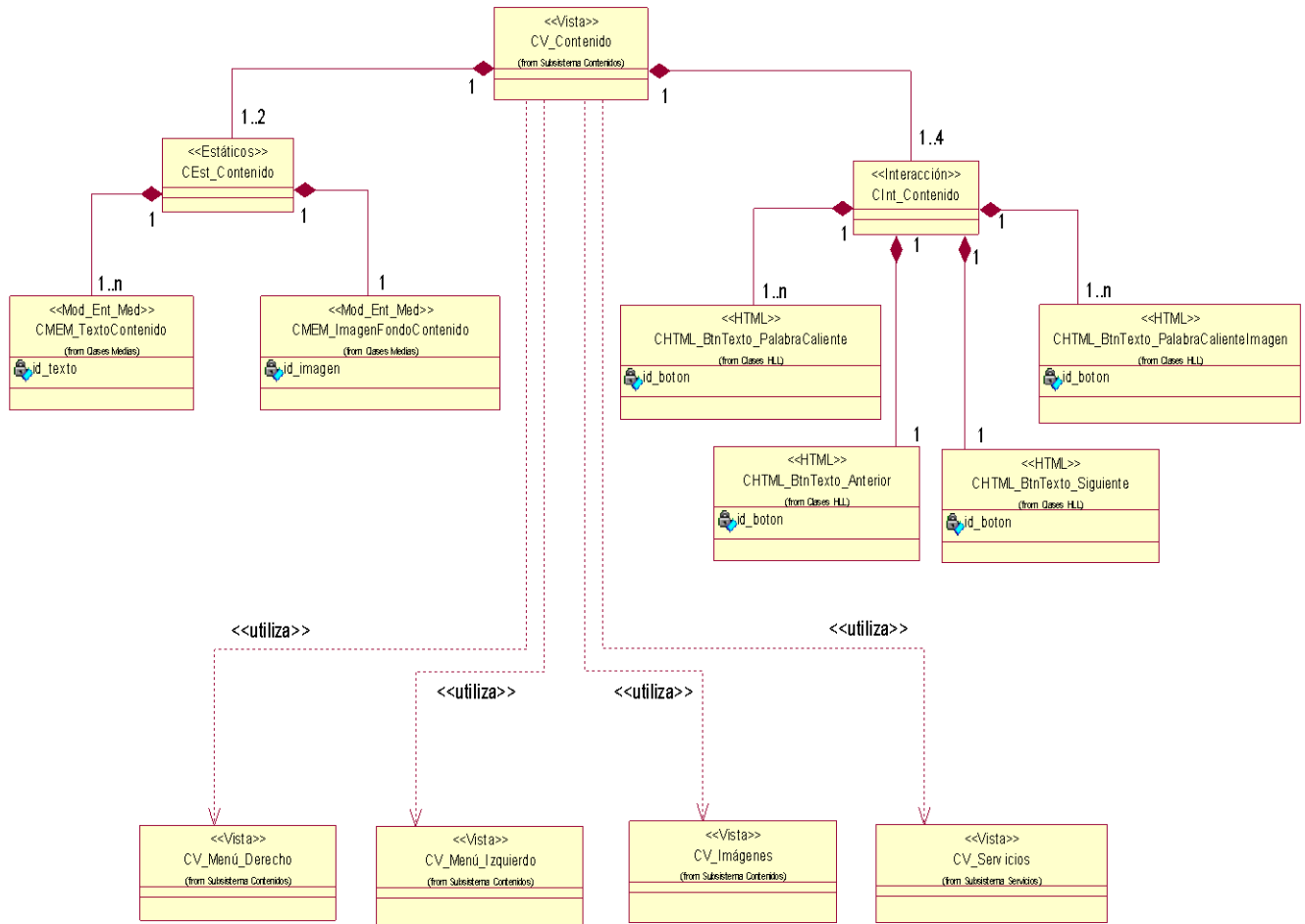


Figura 5. DEP Vista Contenido

- ✓ DEP Vista Imágenes [Anexo 2]
- ✓ DEP Vista Menú Derecho [Anexo 3]
- ✓ DEP Vista Menú Izquierdo [Anexo 4]
- ✓ DEP Vista Palabra [Anexo 5]
- ✓ DEP Vista Visor [Anexo 6]

- ✓ DEP Vista Ayuda [Anexo 7]
- ✓ DEP Vista Créditos [Anexo 8]
- ✓ DEP Vista Ejercicios [Anexo 9]
- ✓ DEP Vista Galería de Videos [Anexo 10]
- ✓ DEP Vista Glosario [Anexo 11]
- ✓ DEP Vista Salir [Anexo 12]
- ✓ DEP Vista Servicios [Anexo 13]

2.9 Diagramas de Estructura de Navegación

ApEM-L define un diagrama para modelar la estructura y organización de los elementos de navegación de sistemas multimedia, este es el Diagrama de Estructura de Navegación (DEN).

A continuación se muestran los DEN identificados en la aplicación.

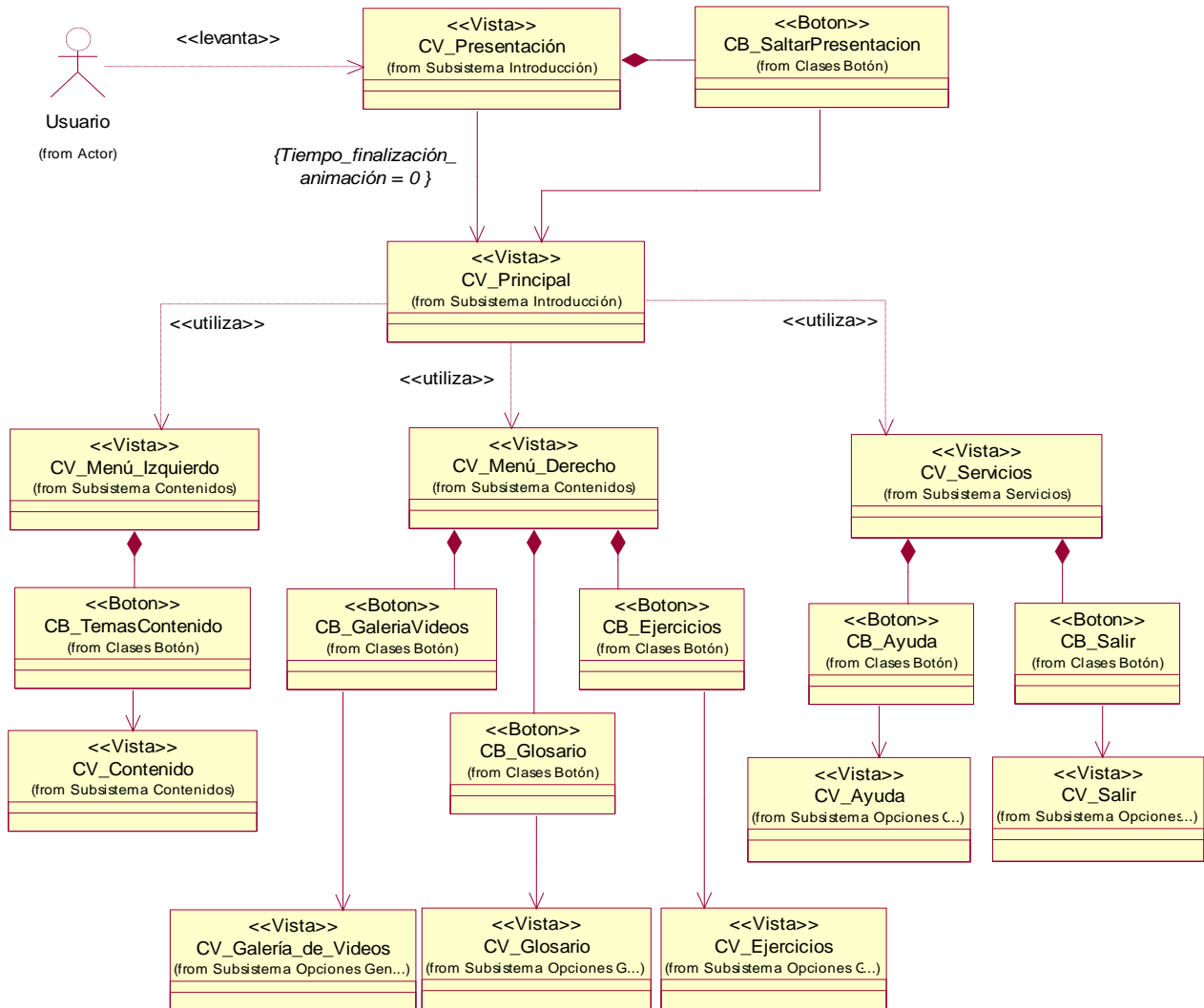


Figura 6. DEN Subsistema Introducción

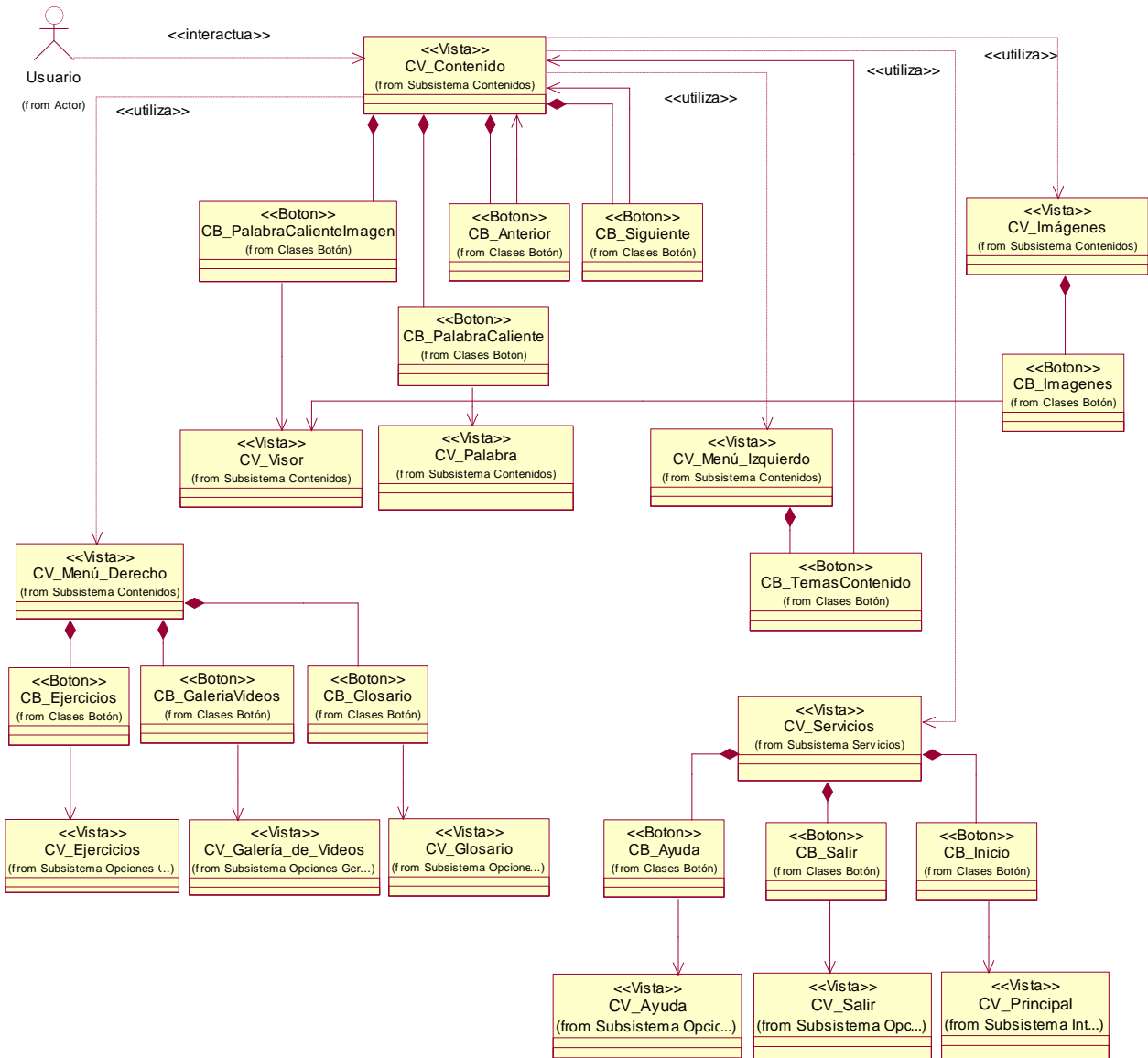


Figura 7. DEN Subsistema Contenidos

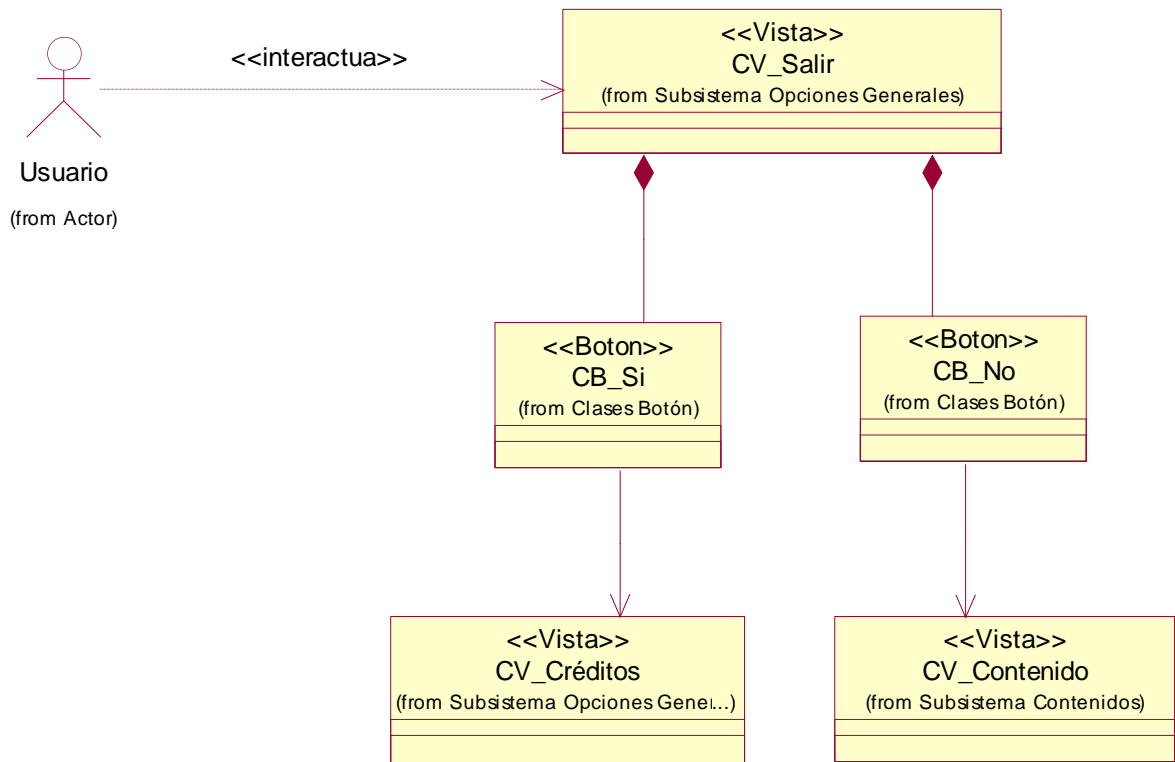


Figura 8. DEN Subsistema Opciones Generales

Con el objetivo de mostrar toda la navegación del sistema en un mismo diagrama, se confeccionó además de los DEN propuestos por ApEM – L, un mapa de navegación general. [Anexo 14]

2.10 Conclusiones del capítulo

En este capítulo se realizó la modelación del dominio del problema real, definiéndose los conceptos fundamentales asociados al mismo. Se especificaron también los requisitos funcionales y no funcionales que debe cumplir el sistema. Se definieron los subsistemas que compondrán la aplicación y las vistas de presentación que conformarán estos subsistemas, realizándose sus descripciones textuales, Diagramas de Estructura Presentación y los Diagramas de Estructura de Navegación para los subsistemas.

Capítulo 3. Construcción de la solución propuesta

3.1 Introducción

Una vez finalizada la primera iteración del flujo de trabajo de Requerimientos que establece RUP para el desarrollo de software, se tienen los elementos necesarios para realizar el diseño y la implementación del mismo, y es precisamente sobre estos flujos de trabajo que trata este capítulo. En el mismo se procede a la construcción de la solución propuesta mediante los flujos de trabajo de diseño e implementación. Se presentan los Diagramas de Clases del Diseño, los Diagramas de Interacción y el Modelo de Implementación que permitirán construir la solución propuesta.

3.2 Diagramas de Clases

El diagrama de clases constituye uno de los diagramas del área de Estructura Lógica que define ApEM-L para el modelado del software, específicamente en la Vista Estática. El Diagrama de Clases no es más que la representación de las clases identificadas y sus relaciones. En la etapa de diseño las clases que componen estos diagramas son descritas con un mayor nivel de detalle, concebidas para satisfacer los requisitos funcionales y no funcionales.

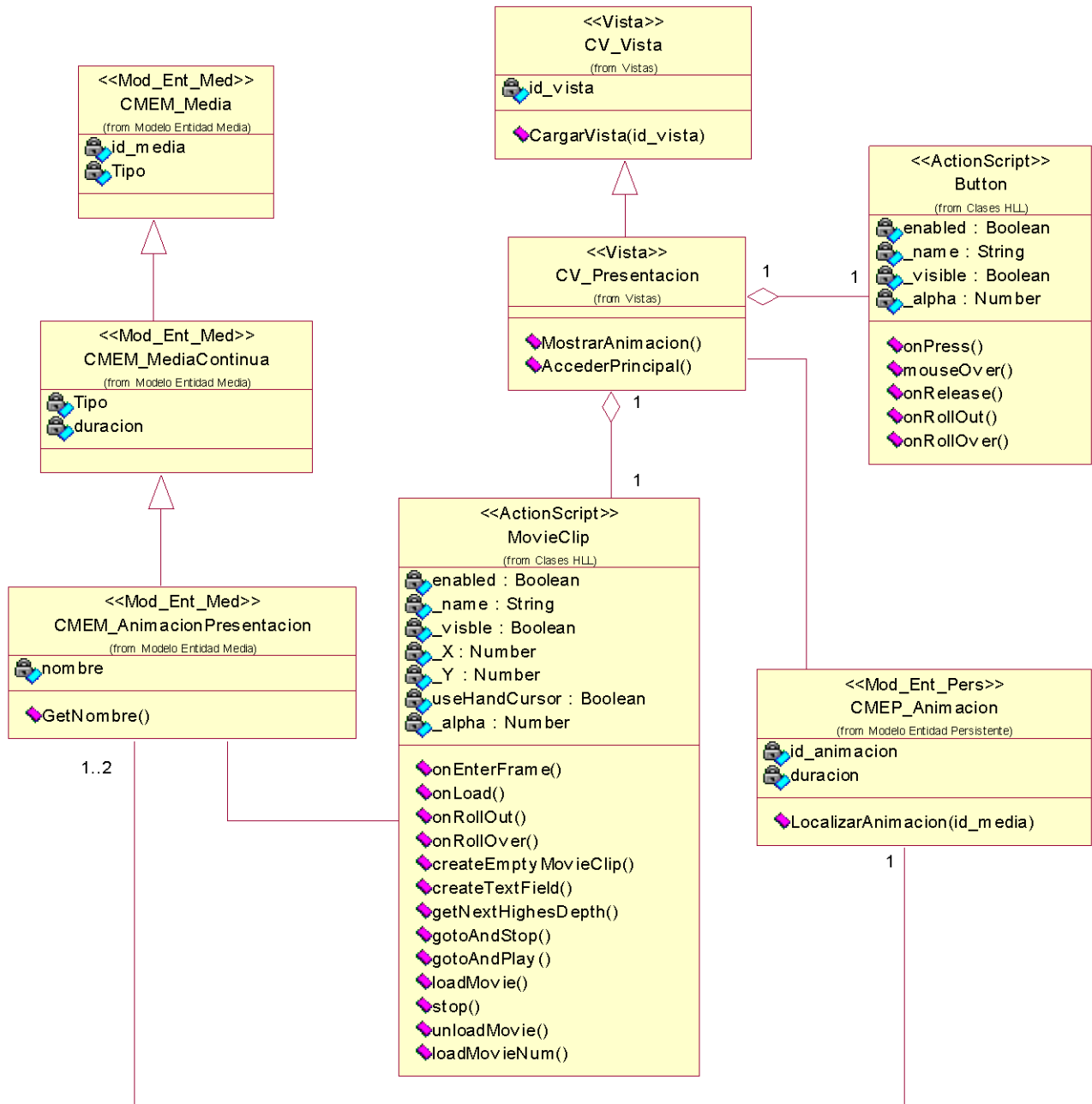


Figura 9. Diagrama de Clases de la Vista Presentación

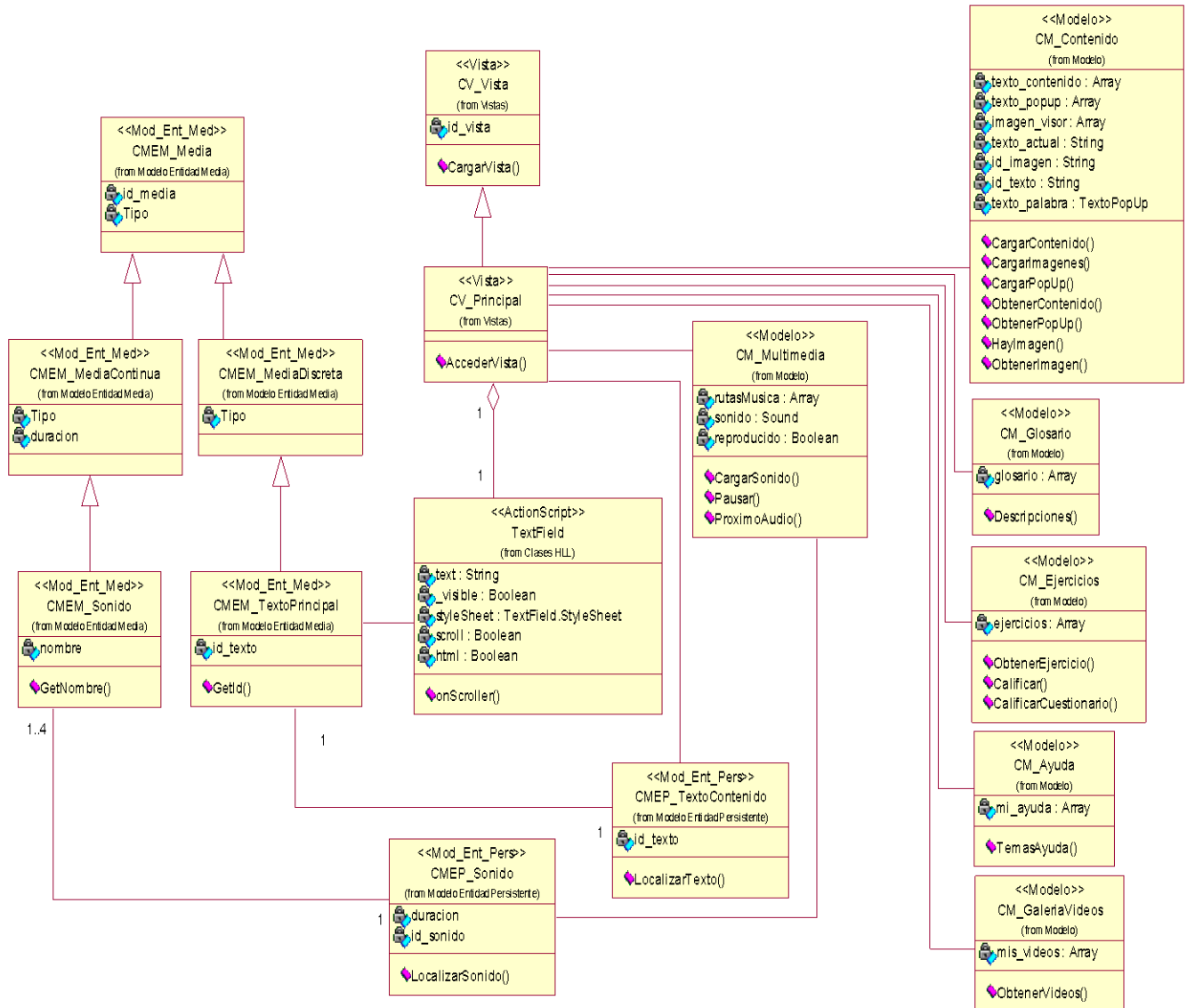


Figura 10. Diagrama de Clases de la Vista Principal

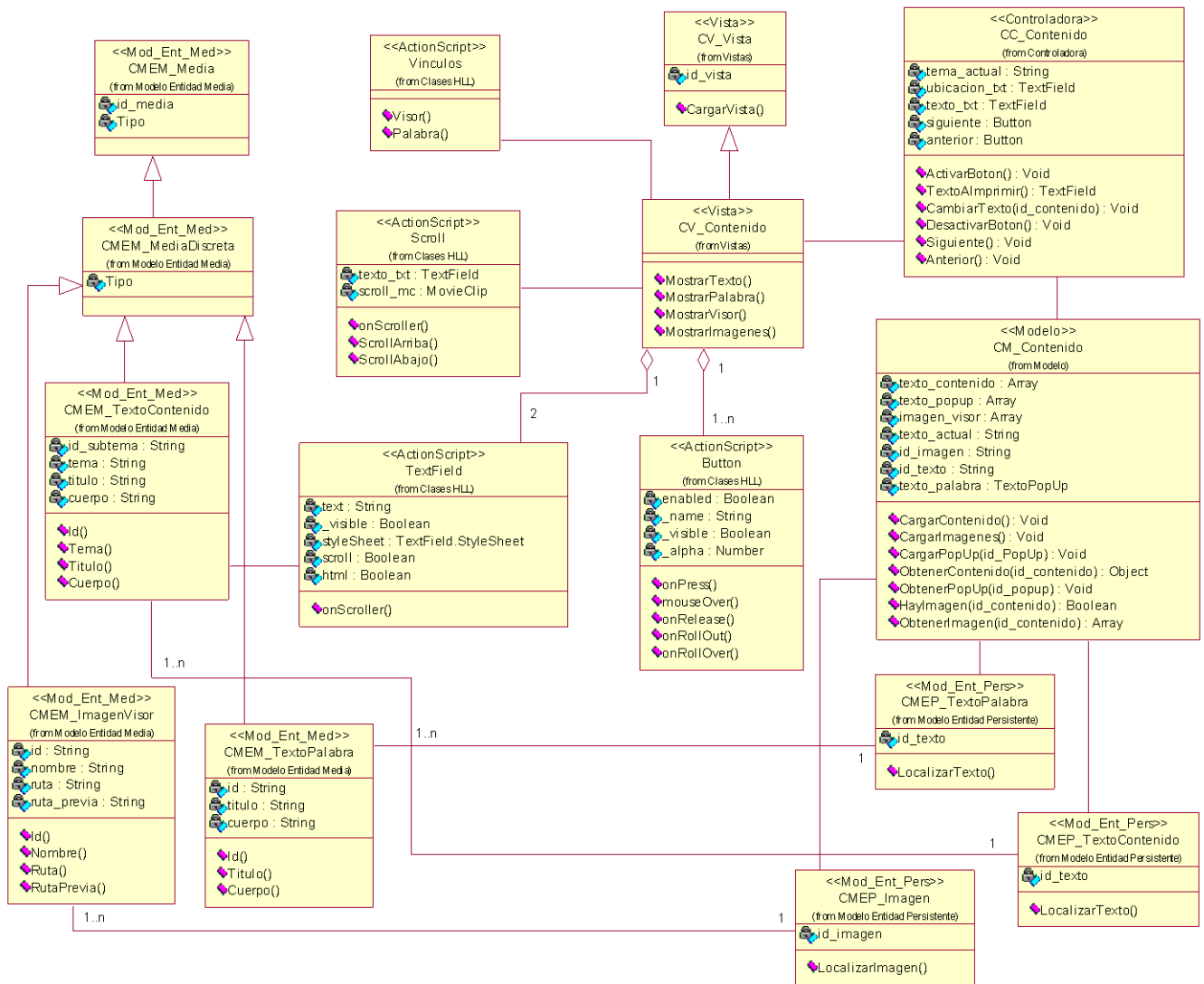


Figura 11. Diagrama de Clases de la Vista Contenido

- ✓ Diagrama de Clases de la Vista Ejercicios [Anexo 15]
- ✓ Diagrama de Clases de la Vista Galería de Videos [Anexo 16]
- ✓ Diagrama de Clases de la Vista Glosario [Anexo 17]

- ✓ Diagrama de Clases de la Vista Palabra [Anexo 18]
- ✓ Diagrama de Clases de la Vista Salir [Anexo 19]
- ✓ Diagrama de Clases de la Vista Visor [Anexo 20]
- ✓ Diagrama de Clases de la Vista Menú Izquierdo [Anexo 21]
- ✓ Diagrama de Clases de la Vista Menú Derecho [Anexo 22]
- ✓ Diagrama de Clases de la Vista Imágenes [Anexo 23]
- ✓ Diagrama de Clases de la Vista Servicios [Anexo 24]
- ✓ Diagrama de Clases de la Vista Ayuda [Anexo 25]

3.3 Diagramas de Interacción

Los Diagramas de Interacción de UML se utilizan para modelar los aspectos dinámicos de un sistema. Explican gráficamente, a través de mensajes, las interacciones existentes entre las instancias de las clases. UML define dos tipos de estos diagramas, los diagramas de colaboración y los diagramas de secuencia.

A continuación se representan los Diagramas de Secuencia, que muestran los mensajes entre los objetos a lo largo del tiempo.

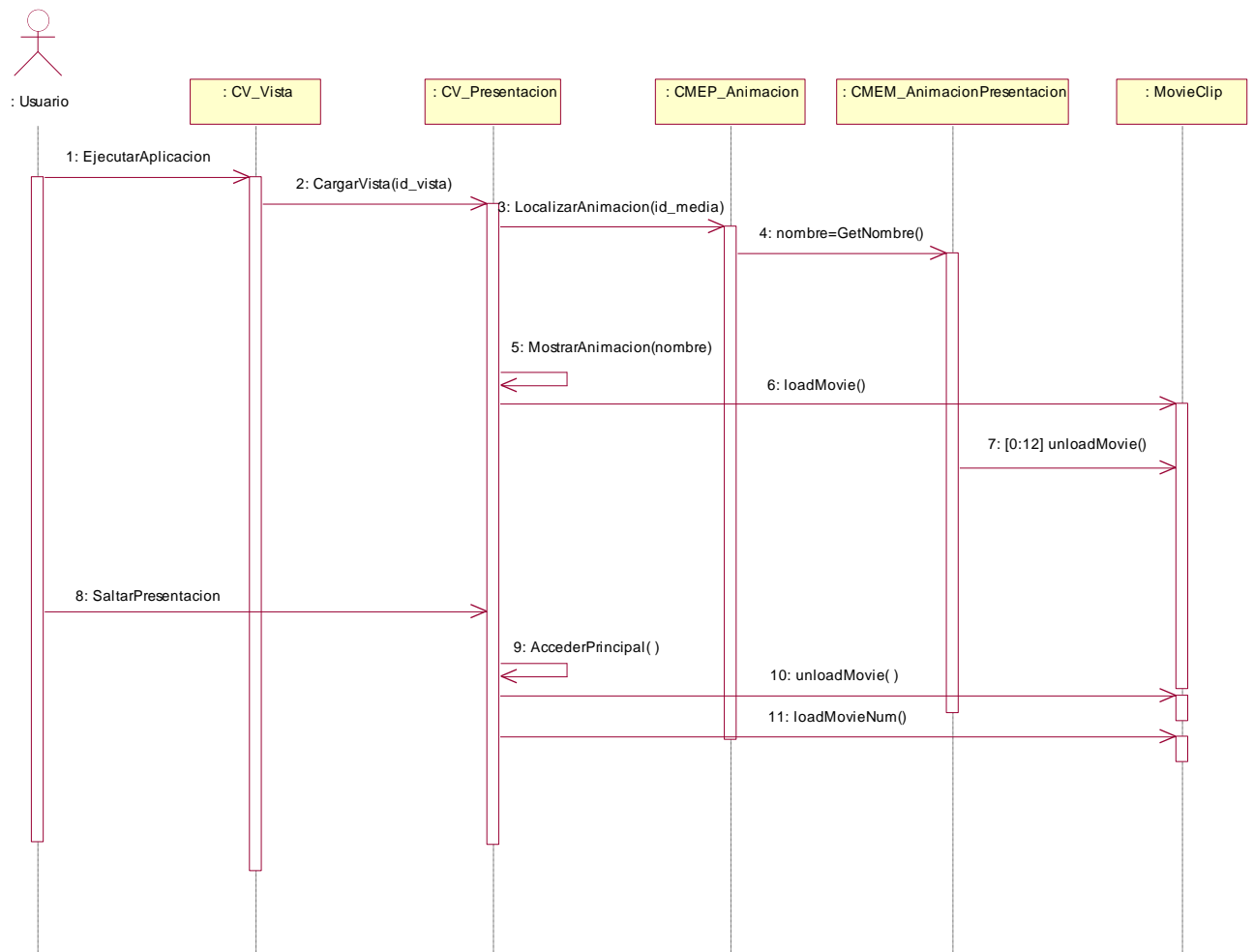


Figura 12. Diagrama de Secuencia de la Vista Presentación

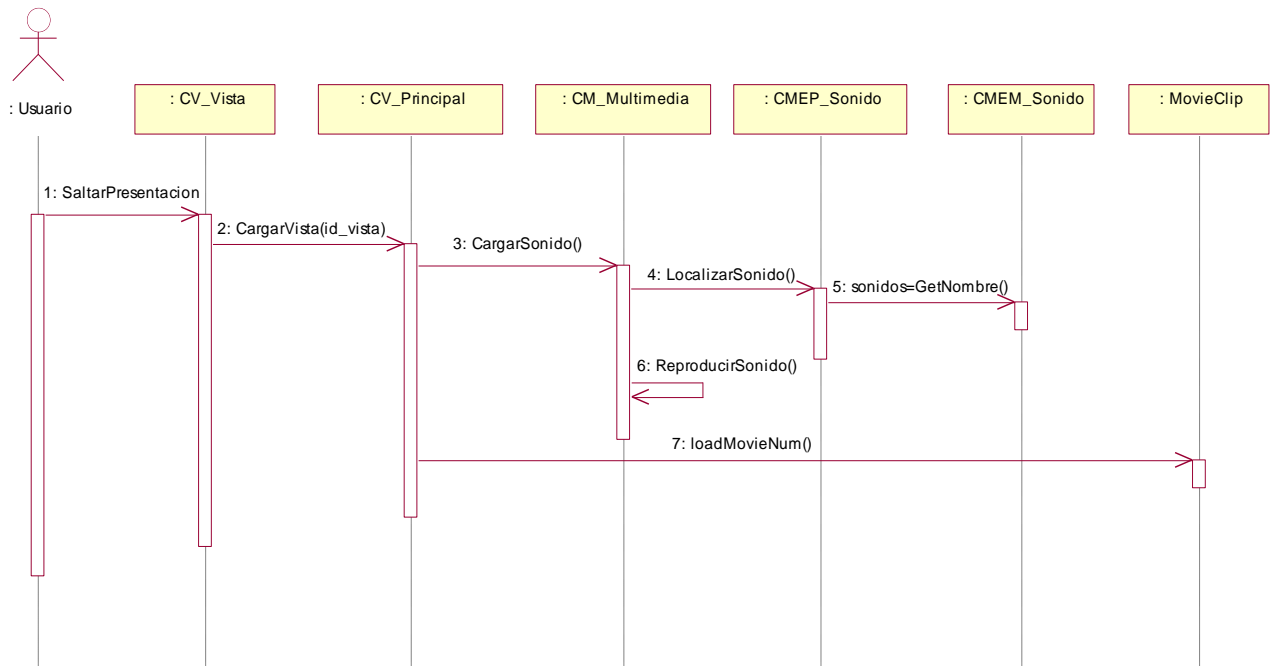


Figura 13. Diagrama de Secuencia de la Vista Principal

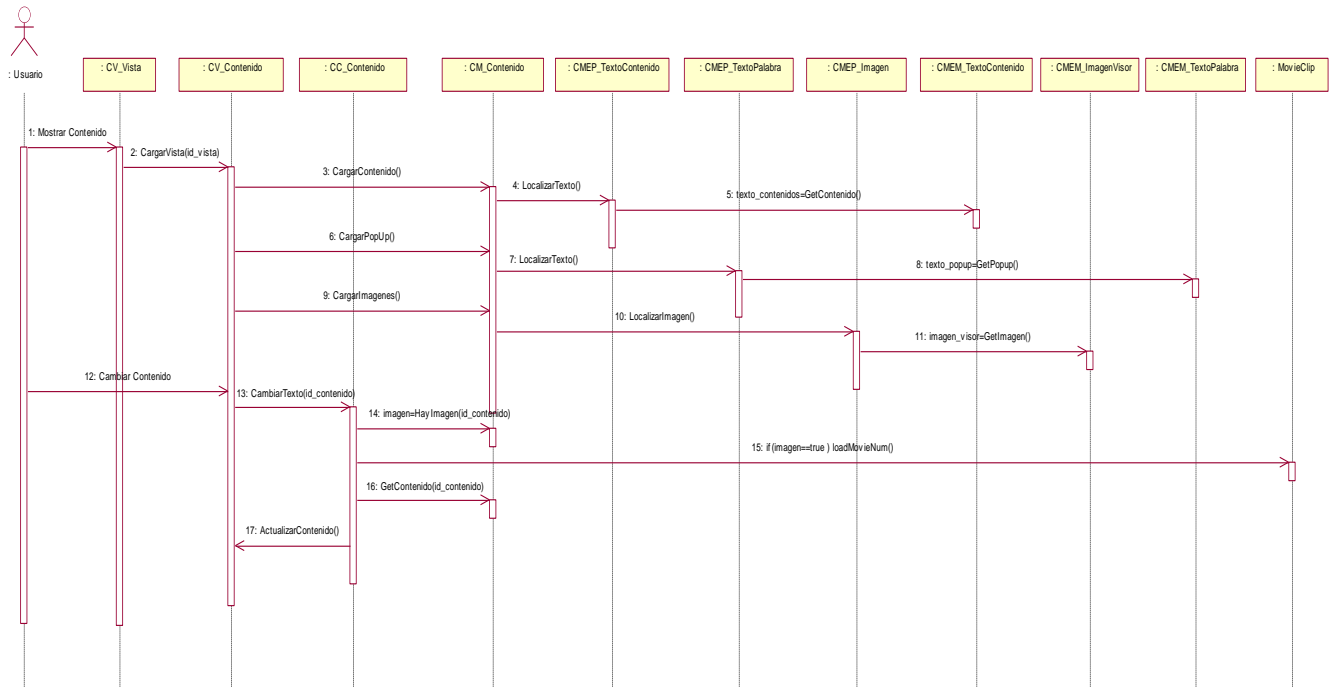


Figura 14. Diagrama de Secuencia de la Vista Contenido

- ✓ Diagrama de Secuencia de la Vista Ejercicios [Anexo 26]
- ✓ Diagrama de Secuencia de la Vista Galería de Videos [Anexo 27]
- ✓ Diagrama de Secuencia de la Vista Glosario [Anexo 28]
- ✓ Diagrama de Secuencia de la Vista Palabra [Anexo 29]
- ✓ Diagrama de Secuencia de la Vista Salir [Anexo 30]
- ✓ Diagrama de Secuencia de la Vista Visor [Anexo 31]
- ✓ Diagrama de Secuencia de la Vista Menú Izquierdo [Anexo 32]
- ✓ Diagrama de Secuencia de la Vista Menú Derecho [Anexo 33]
- ✓ Diagrama de Secuencia de la Vista Imágenes [Anexo 34]

- ✓ Diagrama de Secuencia de la Vista Servicios [Anexo 35]
- ✓ Diagrama de Secuencia de la Vista Ayuda [Anexo 36]

3.4 Modelo de Implementación

El Modelo de Implementación es considerado como el artefacto más significativo del flujo de trabajo de Implementación, esto está dado por la importancia que tiene para los desarrolladores entender claramente el funcionamiento del sistema desde el punto de vista de componentes y sus relaciones antes de escribir líneas de código. Este modelo está conformado por los Diagramas de Componentes y de Despliegue, los cuales describen los componentes a construir, y su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

Su objetivo fundamental es describir cómo las clases del modelo de diseño se implementan en términos de componentes, tales como ejecutables, ficheros de código fuentes o tablas de una base de datos.

3.4.1 Diagramas de Componentes

Los Diagramas de Componentes muestran tanto los componentes software (código fuente, binario y ejecutable) como las relaciones lógicas entre ellos en un sistema. Los componentes representan todos los tipos de elementos software implicados en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc.

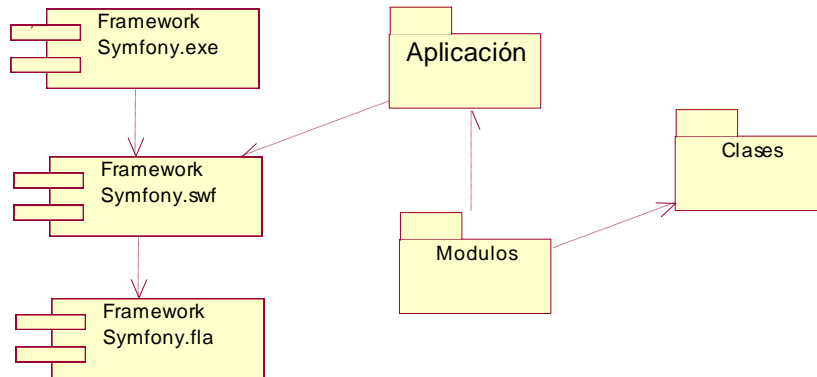


Figura 15. Diagrama de Componentes General

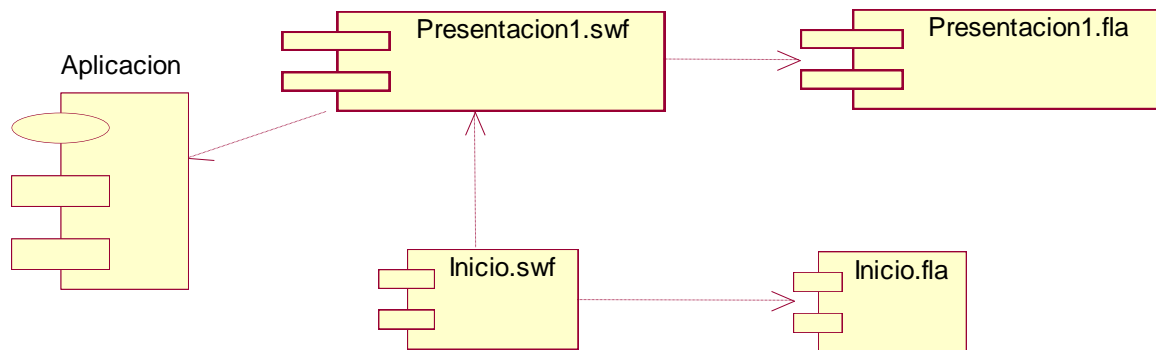


Figura 16. Diagrama de Componentes del Paquete Aplicación

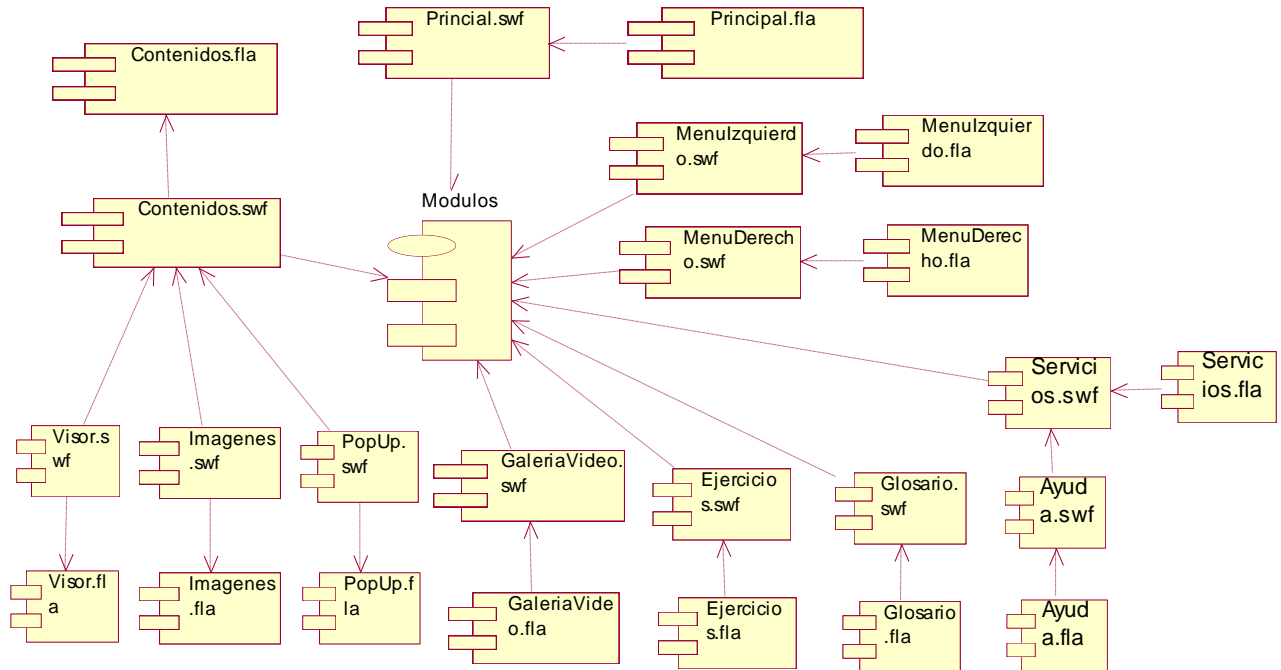


Figura 17. Diagrama de Componentes del Paquete Módulos

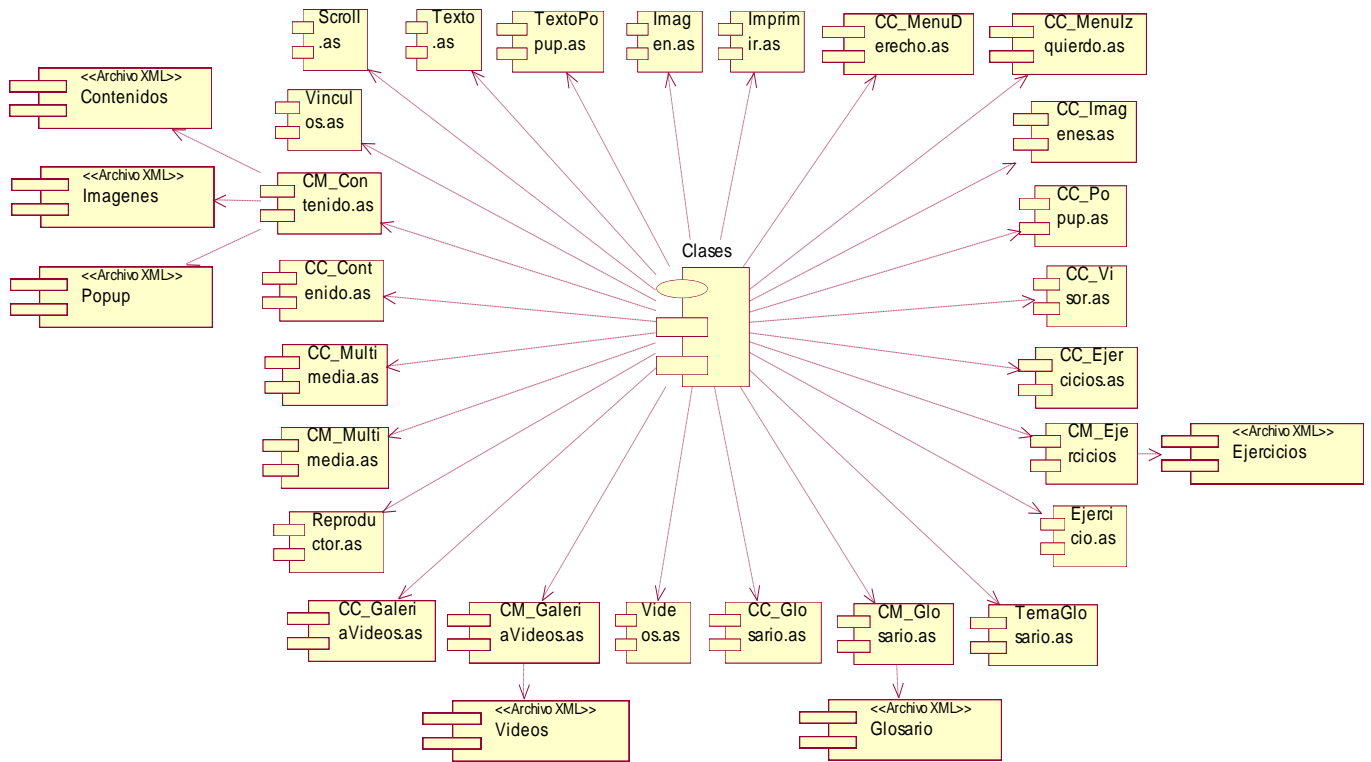


Figura 18. Diagrama de Componentes del Paquete Clases

3.4.2 Diagrama de Despliegue

Un Modelo de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Forma parte de la vista de despliegue de UML, la cual representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

El Modelo de Despliegue en sí mismo representa una correspondencia entre la arquitectura software y la arquitectura del sistema (el hardware). [28]

El Modelo de Despliegue planteado por los autores está compuesto por dos nodos, uno representa la computadora donde se ejecuta la aplicación y la impresora como dispositivo usado por el sistema una vez que el usuario selecciona la opción de imprimir un determinado contenido.

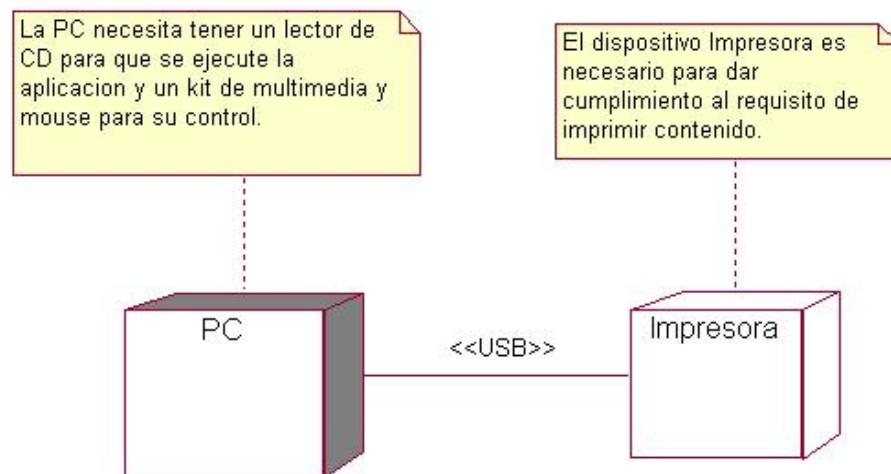


Figura 19. Diagrama de Despliegue

3.5 Conclusiones del capítulo

En este capítulo se diseñó la construcción de la solución propuesta mediante los flujos de trabajo de diseño e implementación. Se representaron los Diagramas de Clases del Diseño, los Diagramas de Interacción y el Modelo de Implementación, que representan los principales artefactos y que permitirán construir la solución propuesta.

Capítulo 4. Estudio de factibilidad

4.1 Introducción

En el presente capítulo se muestra todo lo referente al estudio de la factibilidad del producto a desarrollar dada la importancia y necesidad de saber previamente una serie de aspectos relacionados con la producción del mismo. Con este fin se realiza una planificación, se calcula el tiempo estimado de desarrollo, el costo del producto, los recursos humanos, los gastos económicos y los bienes necesarios a emplear en la producción. Este estudio de factibilidad está basado en el método Análisis de Puntos Función y COCOMO II.

4.2 Planificación

El objetivo de la Planificación del proyecto de Software es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos costos y planificación temporal. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente a medida que progresa el proyecto. Además las estimaciones deberían definir los escenarios del mejor caso, y peor caso, de modo que los resultados del proyecto pueden limitarse. [29]

En la actualidad existen técnicas para la planificación y estimación de sistemas informáticos, una de ellas es el Análisis de Puntos Función que es una técnica estructurada para la clasificación de los componentes de un Sistema. La misma cuantifica los datos que usa la aplicación y las transacciones de entrada-salida como base para estimar el esfuerzo de construcción de esa aplicación. También existen métodos para la estimación del esfuerzo, uno de ellos es el COCOMO II (Modelo Constructivo de Costes II), basado en ecuaciones matemáticas que permiten calcular el esfuerzo a partir de ciertas métricas de tamaño estimado.

4.2.1 Estimación por puntos de función

Tabla 5: Salidas Externas

Salidas Externas			
Nombre de la Salida Externa	Cantidad de Ficheros	Cantidad de Elementos de Datos	Clasificación (Simple, Media y Compleja)
Mostrar la pantalla principal del producto.	1	1	Simple
Mostrar el contenido del subtema seleccionado.	3	4	Simple
Mostrar el contenido de la palabra caliente seleccionada.	3	5	Simple
Mostrar el contenido de la opción Galería de Videos.	3	4	Simple
Mostrar el contenido de la opción Ejercicios.	3	3	Simple
Mostrar el contenido de la opción Glosario de Términos.	2	2	Simple
Mostrar el contenido de la opción Ayuda.	1	1	Simple
Reproducir la música de fondo de la aplicación.	1	1	Simple
Mostrar la pantalla de confirmación de salida de la aplicación.	1	1	Simple
Mostrar Créditos de la aplicación.	1	1	Simple

Tabla 6: Archivos Lógicos Internos

Nombre del archivo interno	Tipos de registro	Cantidad de Elementos de Datos	Clasificación (Simple, Media y Compleja)
Música	1	1	Simple
Contenidos	3	10	Simple
Glosario	1	2	Simple
Videos	1	4	Simple

Ejercicios	1	2	Simple
------------	---	---	--------

Tabla 7: Puntos de Función

Elementos de Información	Simples		Medios		Complejos		Subtotal de puntos de Función
	Cuenta	Peso	Cuenta	Peso	Cuenta	Peso	
Salidas Externas	10	4	0	5	0	7	40
Archivos Lógicos Internos	5	7	0	10	0	15	35
Subtotal de puntos de función sin ajustar(UFP)							75

Tabla 8: Resultados Obtenidos

Características	Valor
Puntos de función desajustados	75
Lenguaje	ActionScript 2.0
Instrucciones fuentes por puntos de función(SLOC/UFP)	66
Instrucciones fuentes	4950

4.3 Estimación del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo

$$PM_{\text{nominal}} = A * (\text{Size})^B$$

Donde:

PM_{nominal}: es el esfuerzo nominal requerido en meses-hombre.

Size: es el tamaño estimado del software, en miles de líneas de código (KSLOC) o en Puntos de Función sin ajustar (convertibles a KSLOC mediante un factor de conversión que depende del lenguaje y la tecnología).

A: es una constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento del tamaño del software. El modelo la calibra inicialmente con un valor de 2.94.

B: es una constante denominada Factor escalar, la cual tiene un impacto exponencial en el esfuerzo y su valor está dado por la resultante de los aspectos positivos sobre los negativos que presenta el proyecto.

Cálculo del Factor Escalar (B)

$$B = 0.91 + 0.01 * \Sigma (SF_i)$$

Tabla 9: Variables Escalares

Variable	Valor
PREC	3.72
FLEX	3.04
RSEL	4.24
TEAM	1.10
PMAT	6.24

SF_i= valor de la variable escalar.

$$\Sigma SF_i = 3.72 + 3.04 + 4.24 + 1.10 + 6.24 = 18.34$$

$$B = 0.91 + 0.01 * 18.34 = 1.0934$$

A = Se toma el valor por defecto del modelo = 2.94

Size: se calcula como el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema.

$$\text{Size} = 75 * 66 = 4950 \text{ SLOC} = 4.9 \text{ KSLOC}$$

$$\text{PM}_{\text{nominal}} = 2.94 * (4.9)^{1.0934}$$

$$\text{PM}_{\text{nominal}} = 2.94 * 5.684 = 16.7109 \text{ Meses-hombre}$$

Para completar la estimación, hay que ajustar el esfuerzo nominal de acuerdo a las características del proyecto.

$$\text{PM}_{\text{ajustado}} = \text{PM}_{\text{nominal}} * \prod (\text{ME}_i)$$

Donde:

Los ME_i (multiplicadores de esfuerzo) varían en función del modelo de estimación seleccionado (Diseño Preliminar o Post arquitectura). Representan las características del proyecto y expresan su impacto en el desarrollo total del producto.

Tabla 10: Multiplicadores de esfuerzo

Multiplicador	Valor
PERS (ACAP, PCAP, PCON)	1.00
RCPX (RELY, DATA, CPLX, DOCU)	1.00
RUSE	1.00
PDIF (TIME, STOR, PVOL)	1.00
PREX (AEXP, PEXP, LTEX)	1.00
SCED	1.00
FCIL (TOL, SITE)	0.87

$$ME = \pi (ME_i) = 1*1*1*1*1*1*0.87 = 0.87$$

$$PM_{\text{ajustado}} = 16.7109 * 0.87 \approx 14.54$$

$$PM_{\text{ajustado}} \approx \mathbf{14.54 \text{ Meses-hombre}}$$

Ya encontrado el esfuerzo (PM), se calcula el tiempo de desarrollo de la aplicación (TDEV).

$$TDEV = 3.67 * (PM)^{0.28 + 0.002 * \sum SFi}$$

$$TDEV = 3.67 * (14.54)^{0.28 + 0.002 * 18.34}$$

$$TDEV = 3.67 * (14.54)^{0.31668} = 8.5669$$

$$TDEV \approx \mathbf{8.5 \text{ meses}}$$

Para estimar cuántas personas requiere el desarrollo:

$$CH = PM / TDEV$$

$$CH = 14.54 / 8.5 = 1.7105$$

$$CH \approx \mathbf{2 \text{ hombres}}$$

Costo

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

$$CHM = CH * \text{Salario mínimo}$$

Considerando que el producto es desarrollado por estudiantes de 5to año de la Universidad de las Ciencias Informáticas, el salario mínimo a considerar es de \$100.

$$CHM = 2 * \$100 = \$200$$

$$\text{Costo} = 200 * 14.54 = 2908$$

$$\text{Costo} = \mathbf{\$ 2908 \text{ o } 116 \text{ CUC.}}$$

4.4 Beneficios Tangibles

- ✓ Documentación necesaria para implementar la aplicación.
- ✓ Reusabilidad de código.
- ✓ Multimedia sobre el framework Symfony.
- ✓ Su distribución podrá realizarse a través de CD-ROMS por lo que se podrá prescindir del uso de recursos de red.

4.5 Beneficios Intangibles

- ✓ Motivar a los estudiantes en el estudio del framework Symfony.
- ✓ Proporcionar contenidos que ayuden y faciliten el aprendizaje sobre el framework Symfony.
- ✓ Mayor aprovechamiento de las tecnologías de la información.

4.6 Análisis de Costos y Beneficios

El producto Multimedia sobre el framework Symfony, no necesita grandes gastos monetarios ni de recursos, y se puede desarrollar en un período de tiempo relativamente breve. El mismo presenta una interfaz amigable, por lo que resulta agradable al usuario y es además fácil de usar y navegar por todas las personas que interactúen con él. Como el aporte más valioso se tiene la obtención de una aplicación interactiva con tecnología multimedia que brinda información referente al framework Symfony. La misma, proporciona a todos los interesados en el tema un material de apoyo para su estudio y auto preparación.

Según los cálculos realizados anteriormente, el costo de desarrollo de este software es \$2908 (moneda nacional), lo que equivale a 116 pesos convertibles aproximadamente. Teniendo en cuenta los resultados tangibles e intangibles que aportaría la realización del producto y el costo del mismo, se considera factible el desarrollo de la aplicación.

4.7 Conclusiones del capítulo

En este capítulo se efectuó el estudio de factibilidad del producto, donde se puntualizaron diferentes aspectos que son de vital importancia en todo proceso de desarrollo de un software. Según cálculos realizados se estima entre otros datos, que el costo de la aplicación es de \$2908, con un tiempo de desarrollo de 8 meses aproximadamente. También se identificaron los beneficios tangibles e intangibles, y se hizo un análisis de los costos y beneficios donde se evidencia claramente la factibilidad de realizar el producto.

Conclusiones

- ✓ Para un exitoso desarrollo de la aplicación con tecnología multimedia que se implementó, se hizo un estudio sobre la ingeniería de software aplicada a productos multimedia, donde se escogió como metodología RUP unido al lenguaje de modelado ApEM-L. Mientras la metodología establecía las actividades y los criterios para conducir el sistema por todo el proceso, el lenguaje de modelado ofrecía la notación gráfica necesaria para representar los sucesivos modelos que se obtenían en el proceso de refinamiento.
- ✓ Se investigó sobre las posibles herramientas para el desarrollo de aplicaciones con tecnología multimedia y se seleccionó Macromedia Flash en su versión 8.0.
- ✓ Se realizó el análisis, el diseño y la implementación del producto *Multimedia sobre el framework Symfony*, obteniendo la documentación necesaria para garantizar el futuro mantenimiento del sistema.
- ✓ Se obtuvo un producto interactivo que cumple con los requisitos establecidos.
- ✓ Con la realización de esta aplicación, en la Facultad 8 de la Universidad de las Ciencias Informáticas aumenta la bibliografía existente sobre el framework Symfony y los materiales interactivos que sirven de apoyo para impartir sus cursos.

Recomendaciones

- ✓ Promover el producto obtenido en las restantes facultades de la UCI.
- ✓ Mantener actualizado el contenido de la aplicación, incorporando una mayor cantidad de textos, videos y ejercicios.

Referencias Bibliográficas

- [1] Cardozo, G. *Tecnologías de la Información y la Comunicación en la Educación. Un punto de vista para discusión*. [21/05/2009]. Disponible en: <http://www.rieoei.org/jano/2085Cardozo.pdf>
- [2] Sunkel, G. *Las tecnologías de la información y la comunicación (TIC) en la educación en América Latina. Una exploración de indicadores*, 2006. [21/05/2009]. Disponible en: <http://www.eclac.org/publicaciones/xml/7/27817/Serie126final.pdf>
- [3] Palacios, L. L. *La Revolución Científico Técnica y el impacto de la Informática en la Educación*, 2000. [21/05/2009]. Disponible en: <http://fcmjtrigo.sld.cu/materiales/informatica/rctinformaticayeducacion.doc>
- [4] Brunner, J. *Educación: Escenarios de Futuro. Nuevas Tecnologías y Sociedad de la Información*. [21/05/2009].
- [5] Sánchez, J. *Construyendo y Aprendiendo con el Computador*, 1999. [22/05/2009].
- [6] Fernández, J. *Herramientas y Métodos para la Producción Multimedia*, 2005. [26/01/2009]. Disponible en: http://www.tdx.cat/TESIS_UPC/AVAILABLE/TDX-0416108-105158//01_fernandezSanchez.pdf
- [7] Pons, J. *Las tecnologías de la información y la comunicación: un punto de vista educativo*. [21/05/2009]. Disponible en: <http://www.ucm.es/info/multidoc/multidoc/revista/num8/jpablos.html>
- [8] Mascetti, R. *Uso de los multimedia en la educación*, 2007. [21/05/2009]. Disponible en: <http://www.eliceo.com/consejos/usos-de-los-multimedia-en-la-educacion.html>
- [9] Berro, P. *Ventajas del Software Educativo Interactivo*. [21/05/2009]. Disponible en: http://www.multisoft.com.uy/servicios_educativo_interactivo_ventajas.htm

- [10] Figueroa, R.; Solís, C.; Cabrera, A. *Metodologías Tradicionales Vs. Metodologías Ágiles*, 2008. [02/02/2009]. Disponible en: <http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato.doc>
- [11] Booch, G.; Jacobson, I.; Rumbaugh, J. *El Proceso Unificado de Desarrollo de Software*, Madrid, Editorial Pearson Educación, S. A., 2000. [05/02/2009].
- [12] Beck, K. *Extreme Programming Explained: Embrace Change*. Editorial Addison Wesley Longman, 2000. [23/05/2009].
- [13] Aguilar, A. *Programación Extrema y Software Libre*, 2002. [21/05/2009]. Disponible en: <http://www.seguridad.unam.mx/eventos/datos/ev11/semi18/mat.7.pon19.semi18.pdf>
- [14] Ramos, Y. *Conferencia: Metodologías Ágiles de Desarrollo de Software*. Universidad de las Ciencias Informáticas. Ciudad de la Habana, 2008. [23/05/2009].
- [15] Hernández, E. *El Lenguaje Unificado de Modelado (UML)*, 2004. [02/02/2009]. Disponible en: http://www.acta.es/articulos_mf/26067.pdf
- [16] Santos, M. C. *Prevención de las disgrafías escolares: una necesidad de la escuela actual, para la atención a la diversidad*. [23/05/2009]. Disponible en: www.rmm.cl/usuarios/vagui/doc/200505301105330.Curso%209.doc
- [17] Sauer, S.; Engels G. *MVC-Based Modeling Support for Embedded Real - Time Systems*. [22/05/2009]. Disponible en: <http://wwwcs.uni-paderborn.de/cs/ag-engels/Papers/1999/SauerOMER99.pdf>
- [18] Sauer, S.; Engels, G. *Extending UML for Modeling of Multimedia Applications*, 2001. [22/05/2009]. Disponible en: <http://www.itec.uni-klu.ac.at/~harald/proseminar02/sauer1.pdf>

- [19] Ciudad, F. A.; Herrera, Y. *ApEM – L 1.5: Una Nueva Versión del Lenguaje de Modelación para Aplicaciones Educativas Multimedia*, 2007. [05/02/2009].
- [20] Ciudad, F. *ApEM – L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI*, 2007. [05/02/2009].
- [21] Ruiz, J. *Comparativa de Sistemas Multimedia en el Diseño de Unidades Didácticas en Ciencias, Multimedia en la enseñanza de las Ciencias*, 1999. [09/01/2009]. Disponible en: <http://www.sociedadelainformacion.com/fisica/multimedia/multimedia.htm>
- [22] Suárez, F. J. *Desarrollo de Aplicaciones Multimedia*, 2008-2009. [22/05/2009]. Disponible en: <http://www.atc.uniovi.es/teleco/5tm/archives/5prog.pdf>
- [23] López, V. *Herramientas profesionales para desarrollar productos multimedia*. [22/05/2009]. Disponible en: <http://espejos.unesco.org.uy/simplac2002/Ponencias/ambientes%20digitales/AD023.doc>
- [24] Cavazos, J. *Tecnología Flash en Sitios Web*, 2006. [22/05/2009]. Disponible en: <http://www.espacios.net.mx/tecnolog-a-flash-en-sitios-web.html>
- [25] *Curso Programación Con ActionScript 2.0*. [22/05/2009]. Disponible en: http://www.virtual-formac.com/curso_programacion_con_actionscript_2_0-c6408.html?idcurso=6408
- [26] González, B. *XML: el lenguaje de los Servicios Web*, 2005. [22/05/2009]. Disponible en: <http://www.desarrolloweb.com/articulos/1574.php>
- [27] Velázquez, M. A. *XML como definición y manejo de datos*, 2002. [22/05/2009]. Disponible en: http://www.informatizate.net/articulos/xml_como_definicion_y_manejo_de_datos_20030808.html
- [28] Díaz, S. *Ingeniería para productos con tecnología multimedia*, II Taller de Software Educativo y Multimedia, 2007. [26/01/2009].

- [29] Concepción, P. *Planificación de Proyectos de Software*, 2005. [20/04/2009]. Disponible en:
<http://www.getec.etsit.upm.es/articulos/gproyectos/art4.htm>

Bibliografía

Aguilar, A. *Programación Extrema y Software Libre*, 2002. [21/05/2009]. Disponible en: <http://www.seguridad.unam.mx/eventos/datos/ev11/semi18/mat.7.pon19.semi18.pdf>

ATNOVA. *Macromedia Flash 8.0*, 2008. [29/01/2009]. Disponible en: <http://www.eruano.com/aulavirtual/FCCCurso.aspx?IDAreaTematicaPrincipal=48&IDAreaTematica=48&IDCurso=427>

Berro, P. *Ventajas del Software Educativo Interactivo*. [21/05/2009]. Disponible en: http://www.multisoft.com.uy/servicios_educativo_interactivo_ventajas.htm

Booch, G.; Jacobson, I.; Rumbaugh, J. *El Proceso Unificado de Desarrollo de Software*, Madrid, Editorial Pearson Educación, S. A., 2000. [05/02/2009].

Ciudad, F. A. *ApEM – L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI*, 2007. [05/02/2009].

Ciudad, F. A.; Herrera, Y. *ApEM – L 1.5: Una Nueva Versión del Lenguaje de Modelación para Aplicaciones Educativas Multimedias*, 2007. [05/02/2009].

deHann, J.; Skinner, G. *Guía de aprendizaje de los efectos gráficos de Flash*, 2005. [09/01/2009]. Disponible en: http://www.adobe.com/es/devnet/flash/articles/graphic_effects_guide.html

DESARROLLOWEB. *Director MX*, 2002. [01/02/2009]. Disponible en: <http://www.desarrolloweb.com/articulos/978.php>

Díaz, S. *Ingeniería para productos con tecnología multimedia*, II Taller de Software Educativo y Multimedia, 2007. [26/01/2009].

EDUCARCHILE. *Funciones y características del formato video*. [06/02/2009].
Disponible en: <http://www.educarchile.cl/Portal.Base/Web/verContenido.aspx?ID=138831>

EDUTEKA. Hipertexto: *Qué es y cómo utilizarlo para escribir en medios electrónicos*. [29/01/2009].
Disponible en: <http://www.eduteka.org/Hipertexto1.php>

Figueroa, R; Solís, C; Cabrera, A. *Metodologías Tradicionales Vs. Metodologías Ágiles*, 2008 [02/02/2009].
Disponible en: <http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato.doc>

González, B. *XML: el lenguaje de los Servicios Web*, 2005. [22/05/2009]. Disponible en:
<http://www.desarrolloweb.com/articulos/1574.php>

Hassan, Y. *Diseño Hipermedia centrado en el usuario*, 2002. [26/01/2009].
Disponible en: <http://www.nosolousabilidad.com/articulos/hipermedia.htm>

Hernández, E. *El Lenguaje Unificado de Modelado (UML)*, 2004. [02/02/2009]. Disponible en:
http://www.acta.es/articulos_mf/26067.pdf

ISPETP. *Revolución Multimedia*. [06/02/2009]. Disponible en:
<http://www.ispetp.rimed.cu/pages/gesetp/sitios/tutoriales/tutoriales/softlibre/Revolution.pdf>

José, H. C. Patricio, L. María Carmen, P. *Metodologías Ágiles en el Desarrollo de Software*, 2007.
[20/02/2009]. Disponible en: www.willydev.net/descargas/prev/TodoAgil.Pdf

KIOSKEA. *Métodos rápidos*, 2008. [06/02/2009]. Disponible en: <http://es.kioskea.net/contents/genie-logiciel/methodes-agiles.php3>

López, V. *Herramientas profesionales para desarrollar productos multimedia*, 6to Congreso de Ingeniería Electromecánica y de Sistemas. [05/02/2009]. Disponible en:
<http://espejos.unesco.org.uy/simplac2002/Ponencias/ambientes%20digitales/AD023.doc>

Palacios, L. L. *La Revolución Científico Técnica y el impacto de la Informática en la Educación*, 2000. [21/05/2009]. Disponible en: <http://fcmjtrigo.sld.cu/materiales/informatica/rctinformaticayeducacion.doc>

Quintana, J. *Multimedia en la Enseñanza*, 1997. [01/02/2009]. Disponible en: <http://www.dionisiodiaz.com/multimensenanza/multimediaensenanza.html#02>

Ruiz, J. *Comparativa de Sistemas Multimedia en el Diseño de Unidades Didácticas en Ciencias, Multimedia en la enseñanza de las Ciencias*, 1999. [09/01/2009]. Disponible en: <http://www.sociedadelainformacion.com/fisica/multimedia/multimedia.htm>

Santiso, M A; González, B. *Diseño multimedia en e-learning para el ámbito universitario*, 2005. [20/01/2009]. Disponible en: http://www.nosolousabilidad.com/articulos/multimedia_elearning.htm

Sauer, S.; Engels, G. *Extending UML for Modeling of Multimedia Applications*, 2001. [22/05/2009]. Disponible en: <http://www.itec.uni-klu.ac.at/~harald/proseminar02/sauer1.pdf>

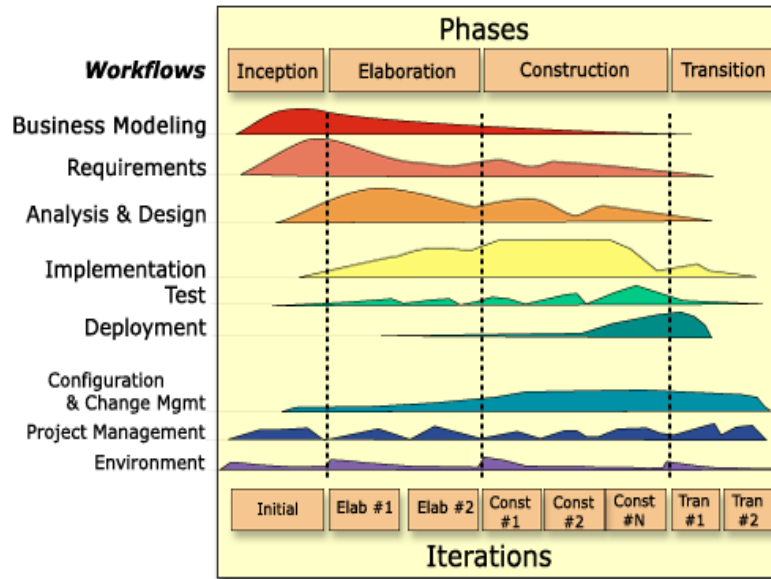
Sauer, S.; Engels G. *MVC-Based Modeling Support for Embedded Real - Time Systems*. [22/05/2009]. Disponible en: <http://wwwcs.uni-paderborn.de/cs/ag-engels/Papers/1999/SauerOMER99.pdf>

Teruel, A. *COCOMO II: Una Familia de Modelos de Estimación*, 2001. [14/4/2009]. Disponible en: www ldc.usb.ve/~teruel/ci4713/clases2001/cocomo2.html

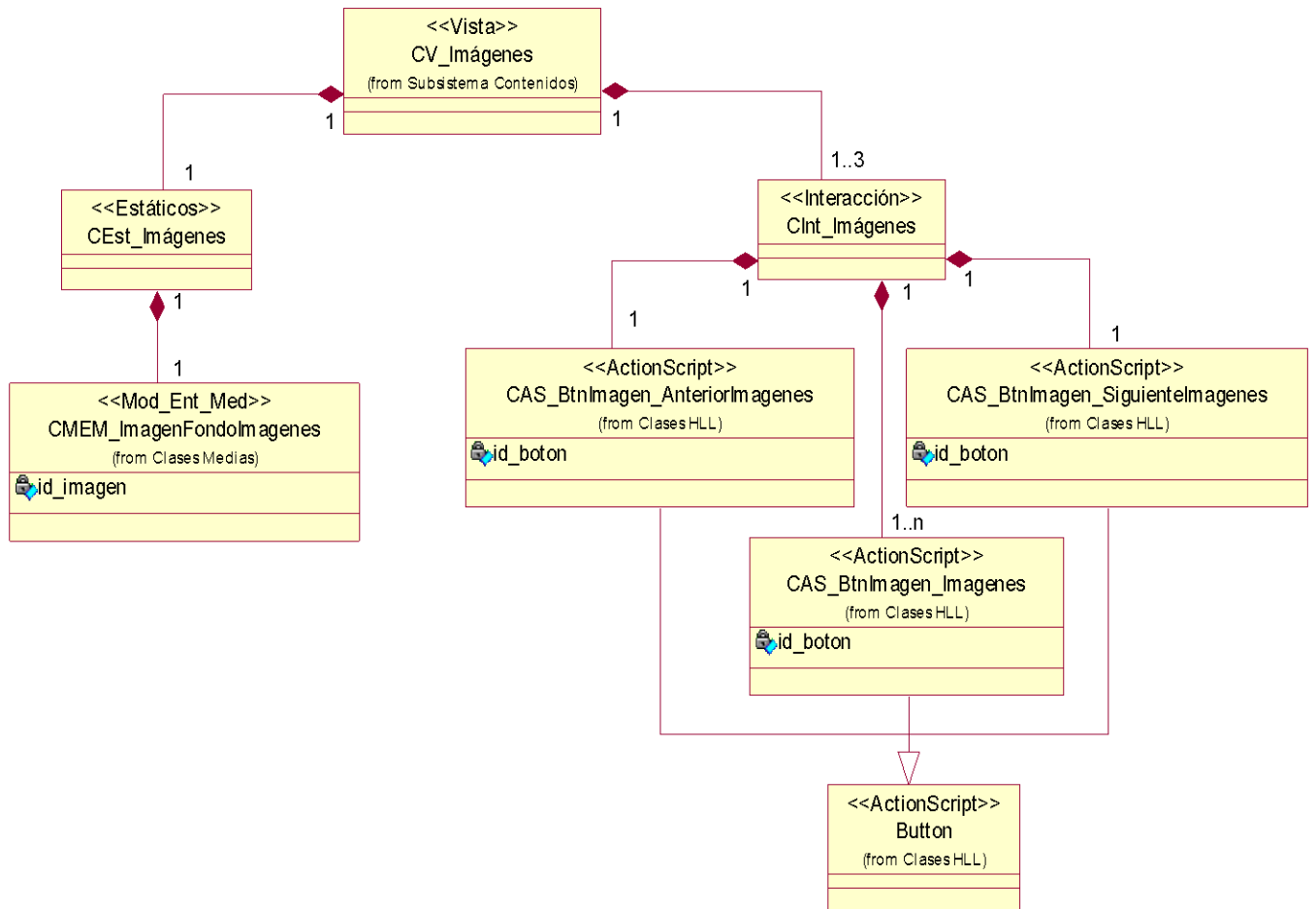
Torrens. R. *Arquitectura de la información*, 1999. [05/02/2009]. Disponible en: <http://oai.saber.ula.ve/db/ssaber/Edocs/postgrados/computacion/ArquitecturadelaInformacion.pdf>

Valverde, D. M. *Introducción a la programación extrema rápidos*, 2007. [16/02/2009]. Disponible en: www.davidvalverde.com/blog/introduccion-a-la-programacion-extrema-xp

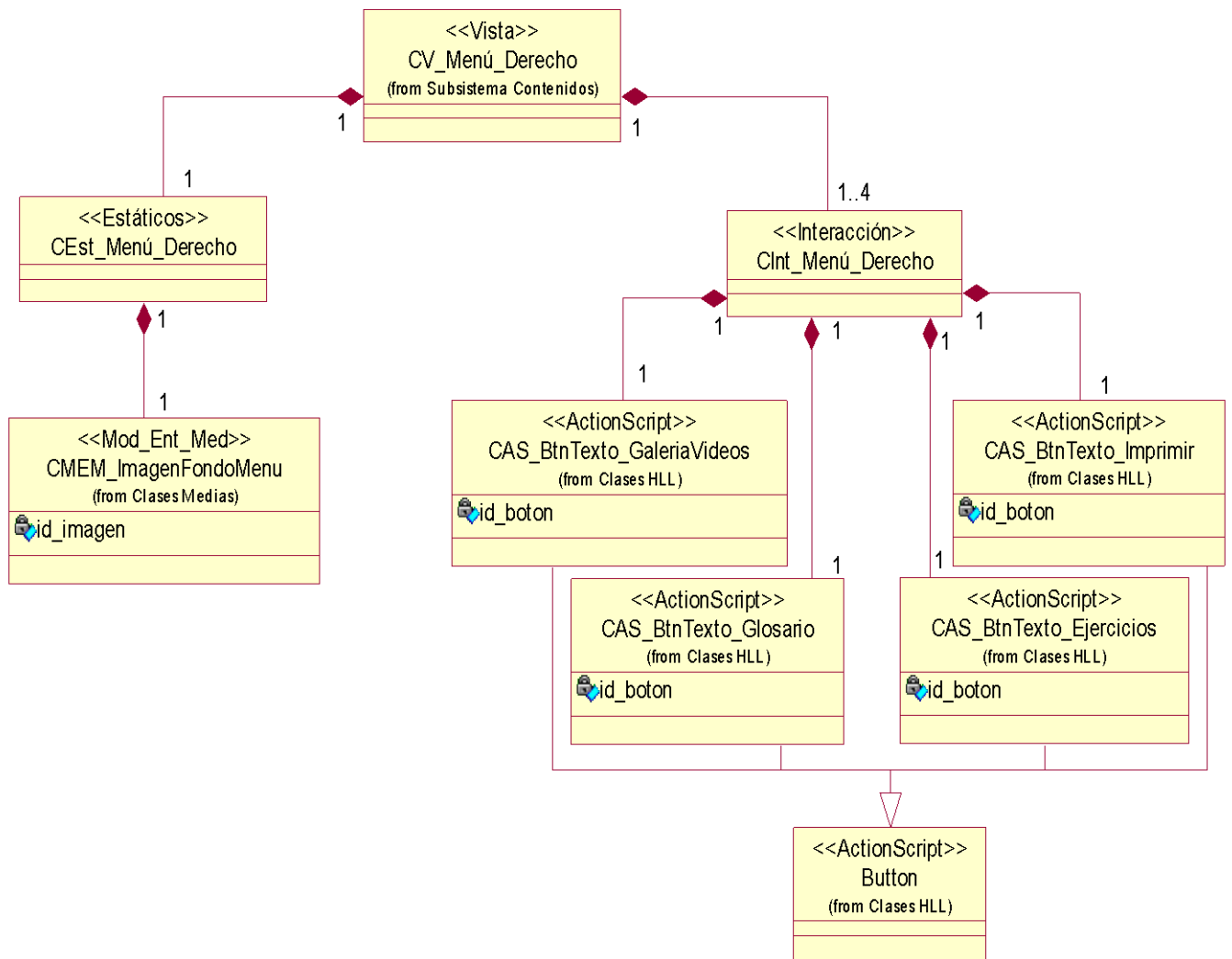
Anexos



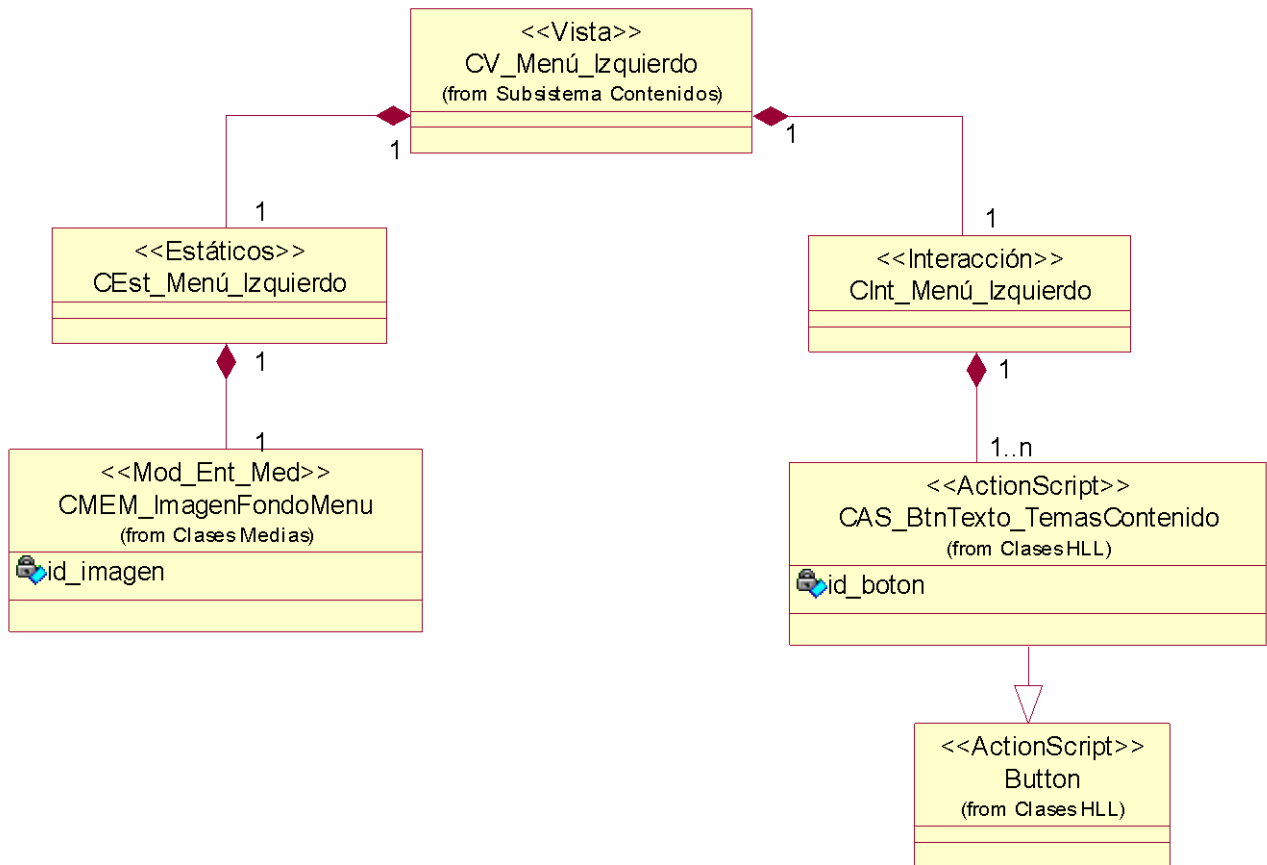
Anexo 1. Proceso Unificado de Rational (RUP)



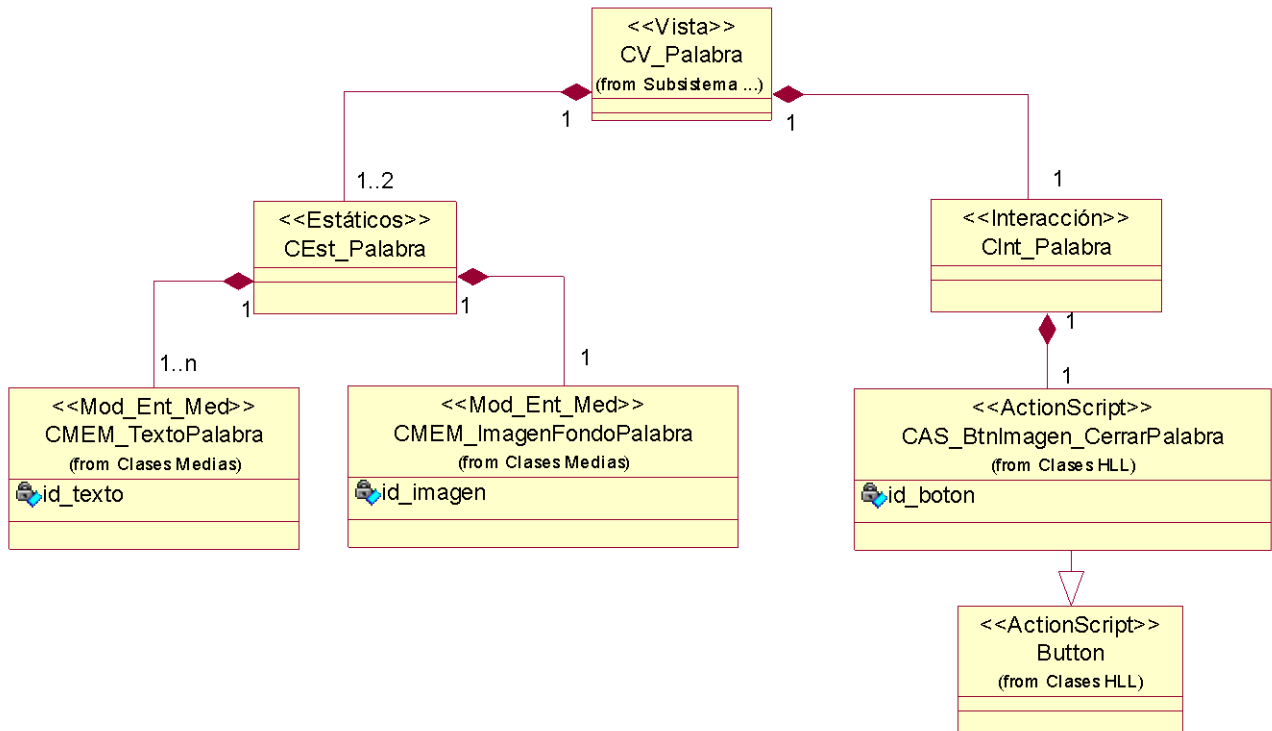
Anexo 2. DEP Vista Imágenes



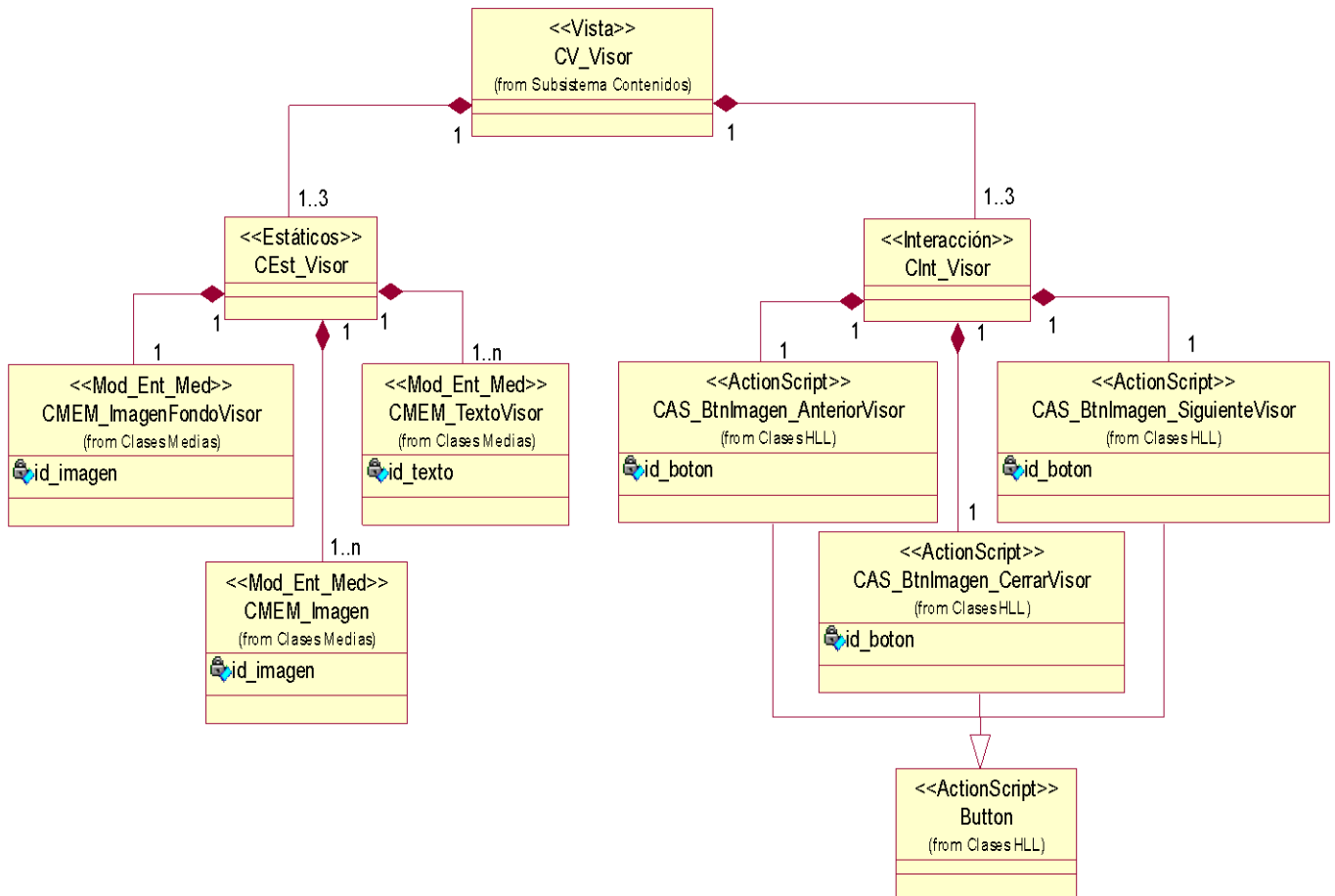
Anexo 3. DEP Vista Menú Derecho



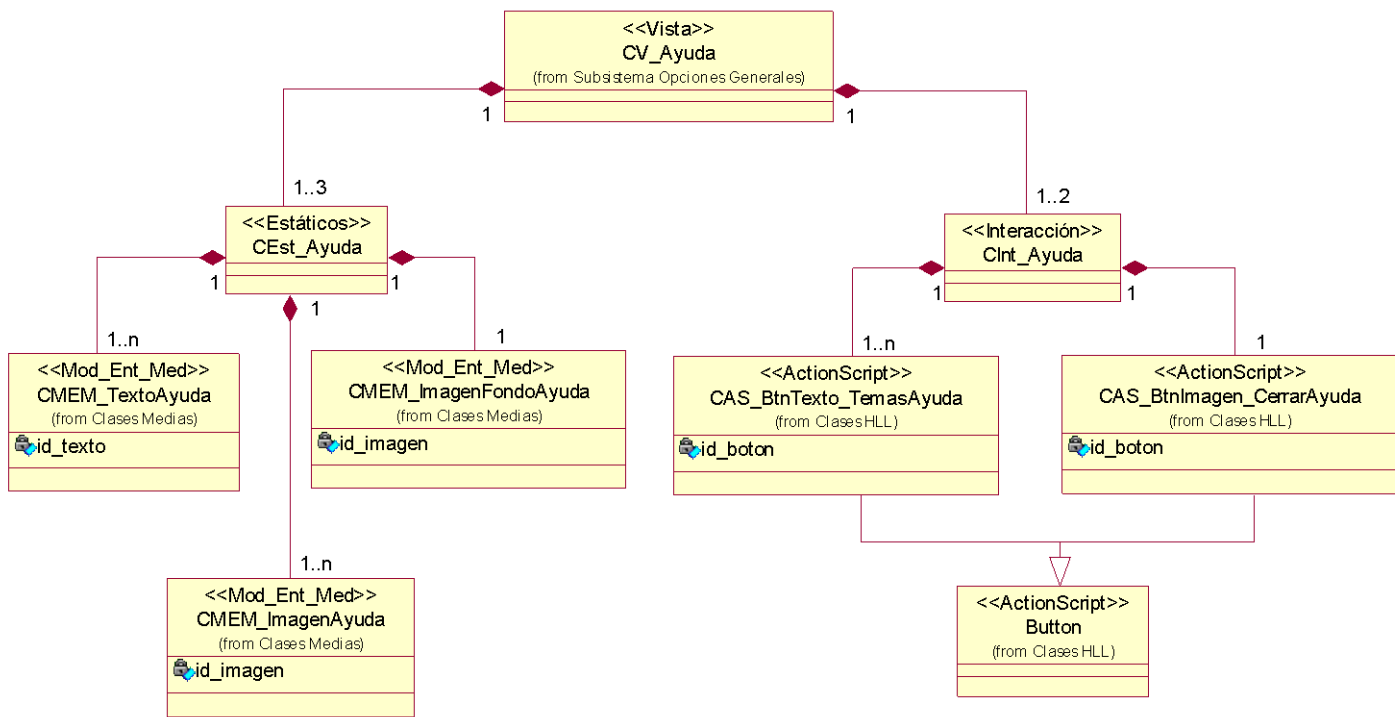
Anexo 4. DEP Vista Menú Izquierdo



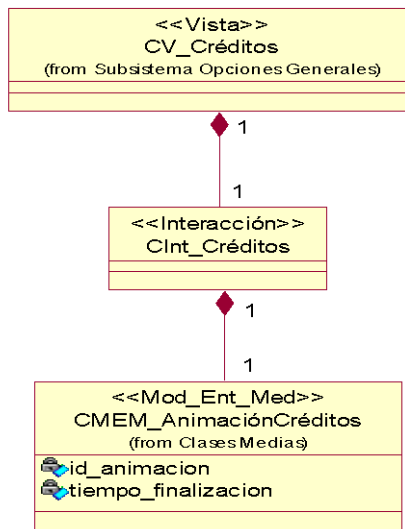
Anexo 5. DEP Vista Palabra



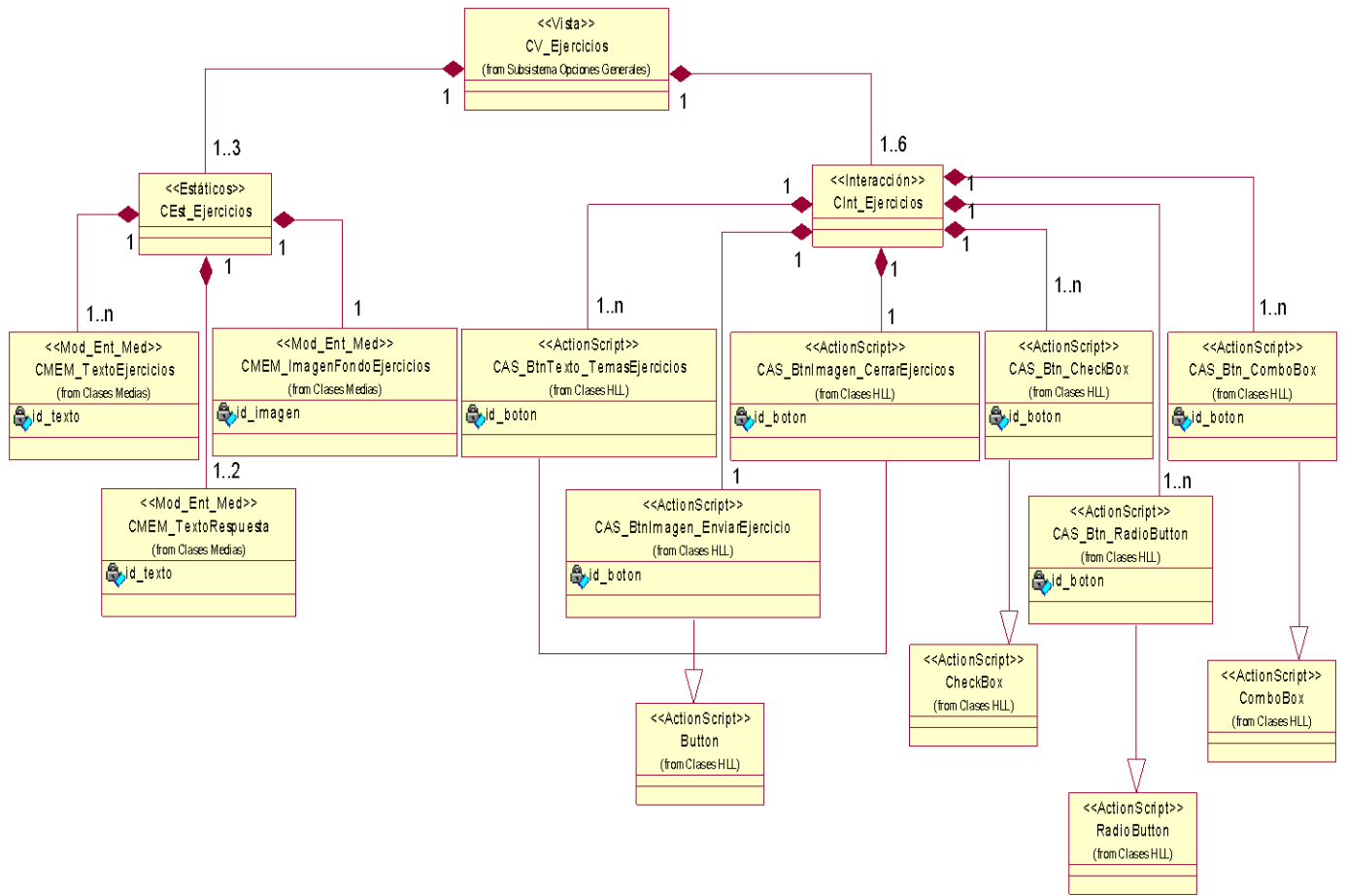
Anexo 6. DEP Vista Visor



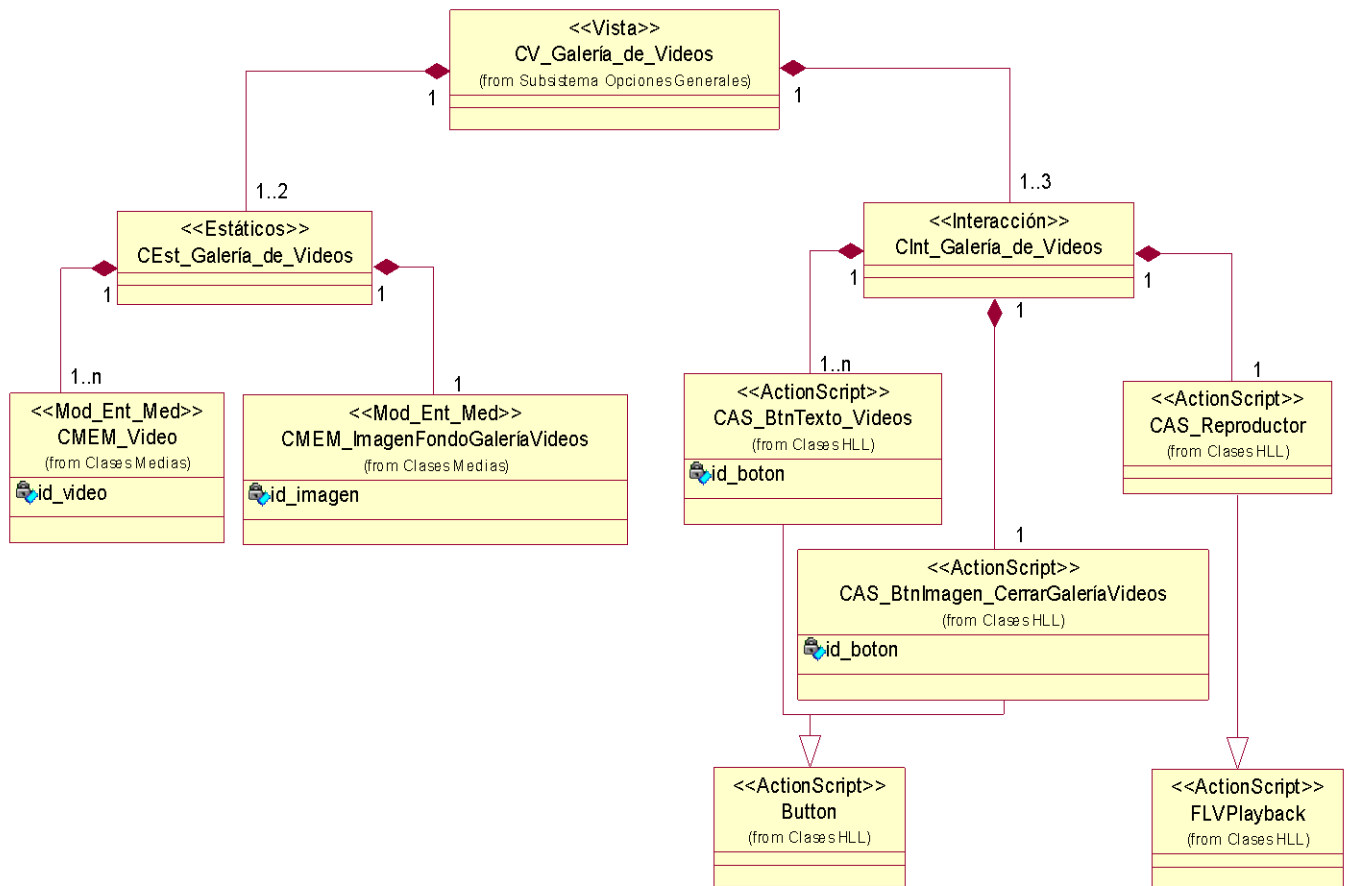
Anexo 7. DEP Vista Ayuda



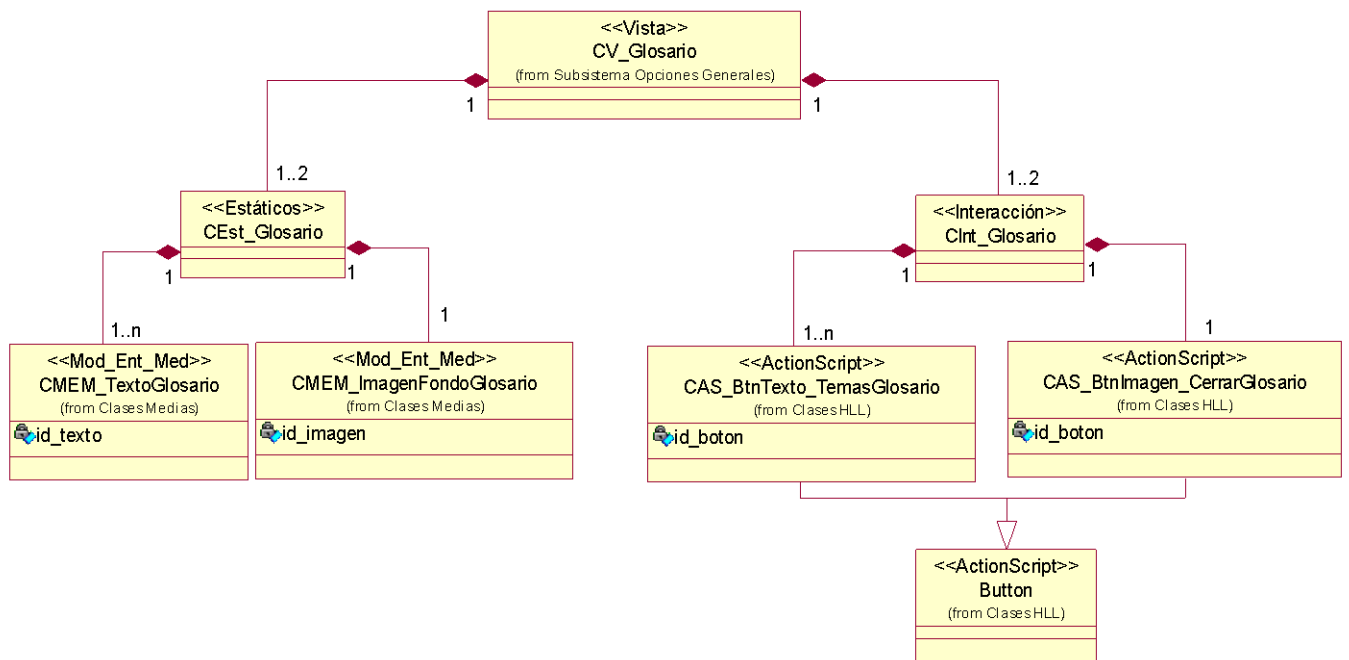
Anexo 8. DEP Vista Créditos



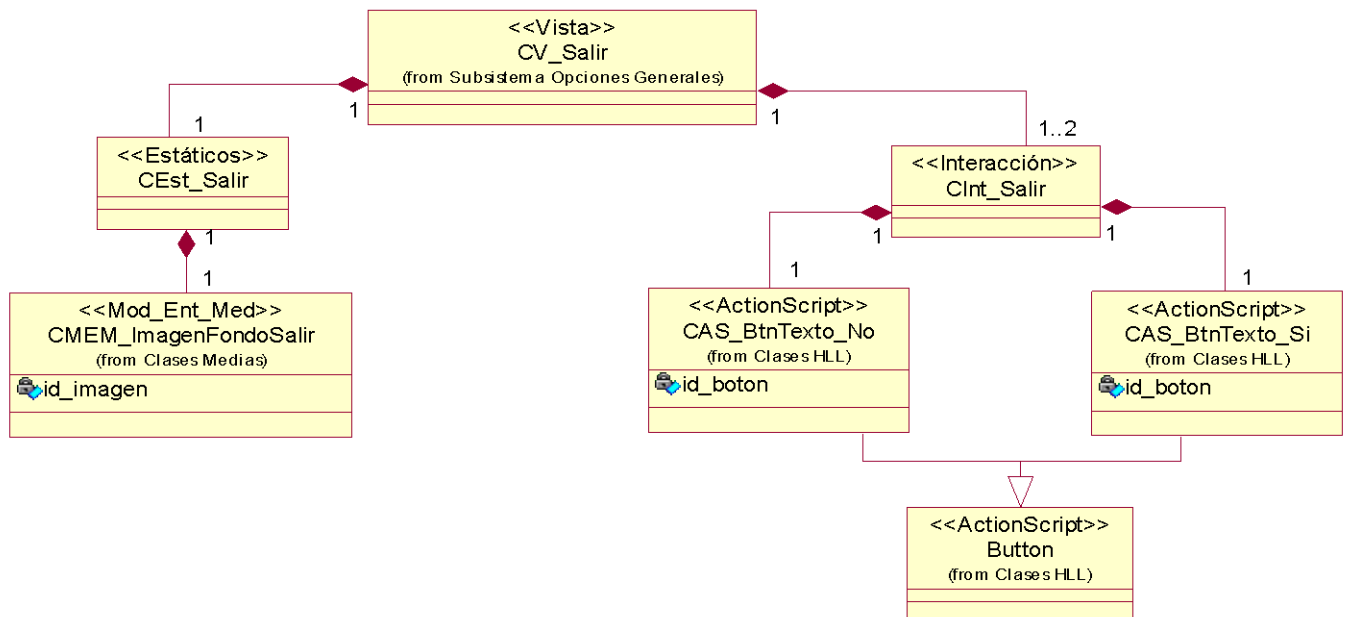
Anexo 9. DEP Vista Ejercicios



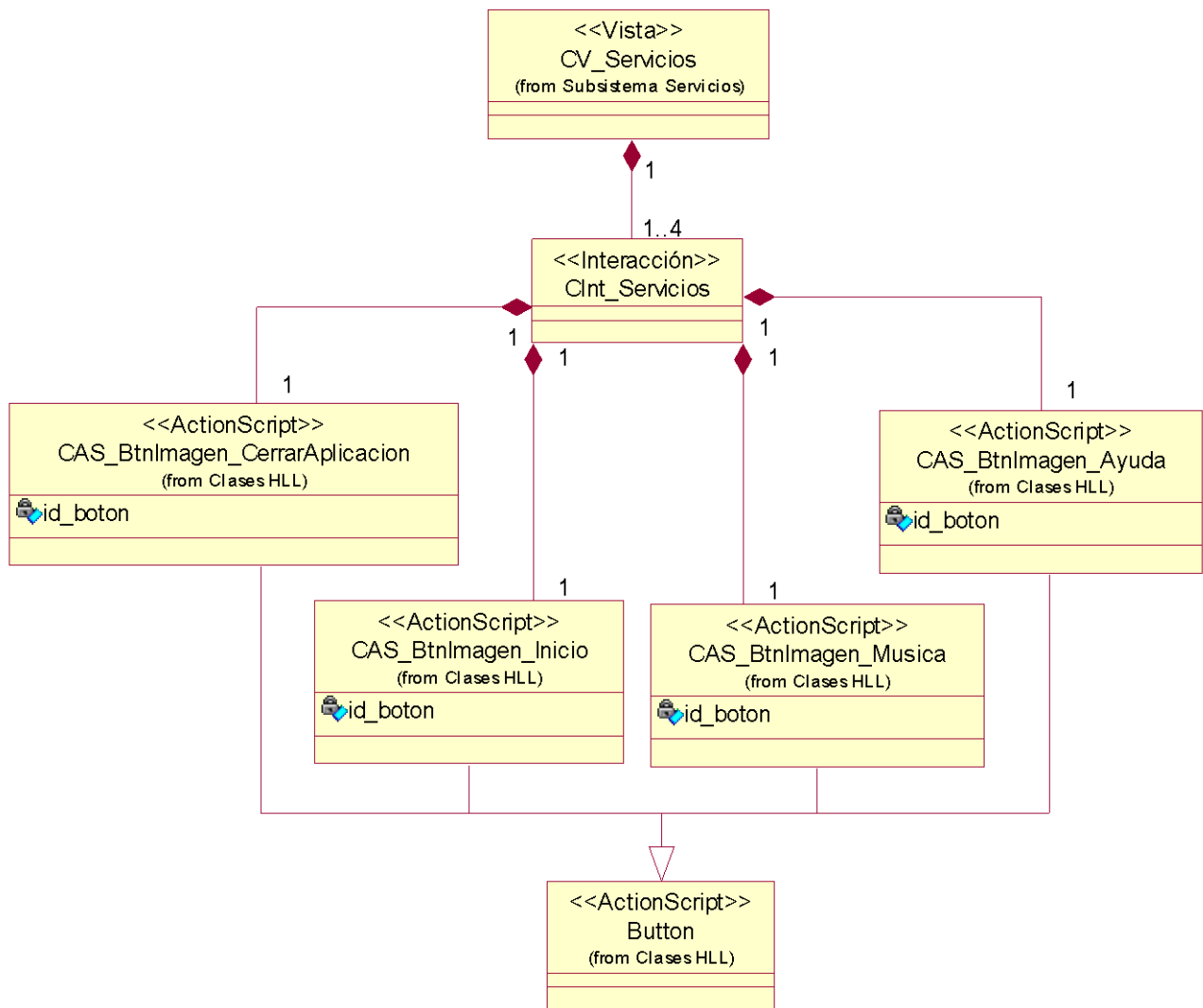
Anexo 10. DEP Vista Galería de Videos



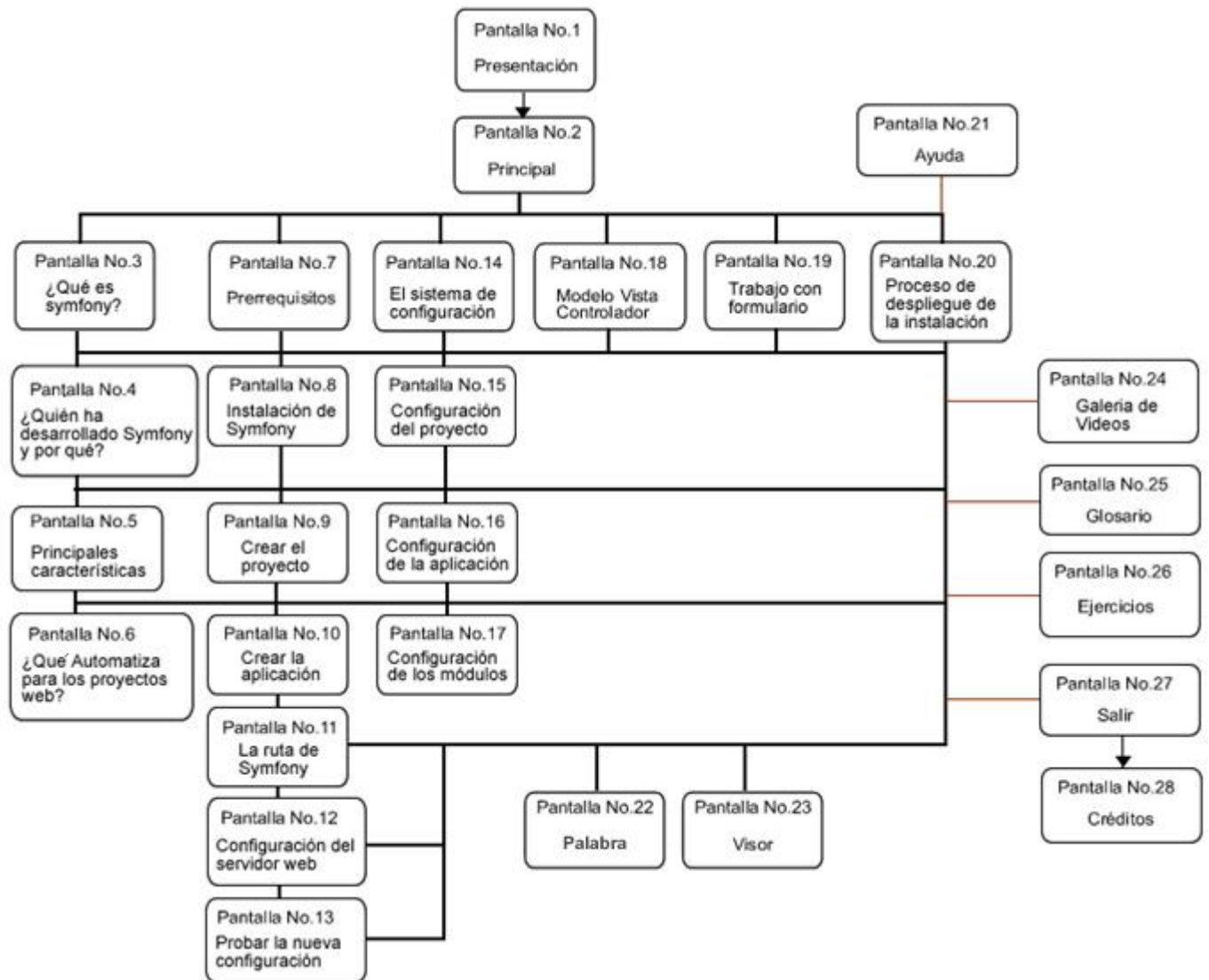
Anexo 11. DEP Vista Glosario



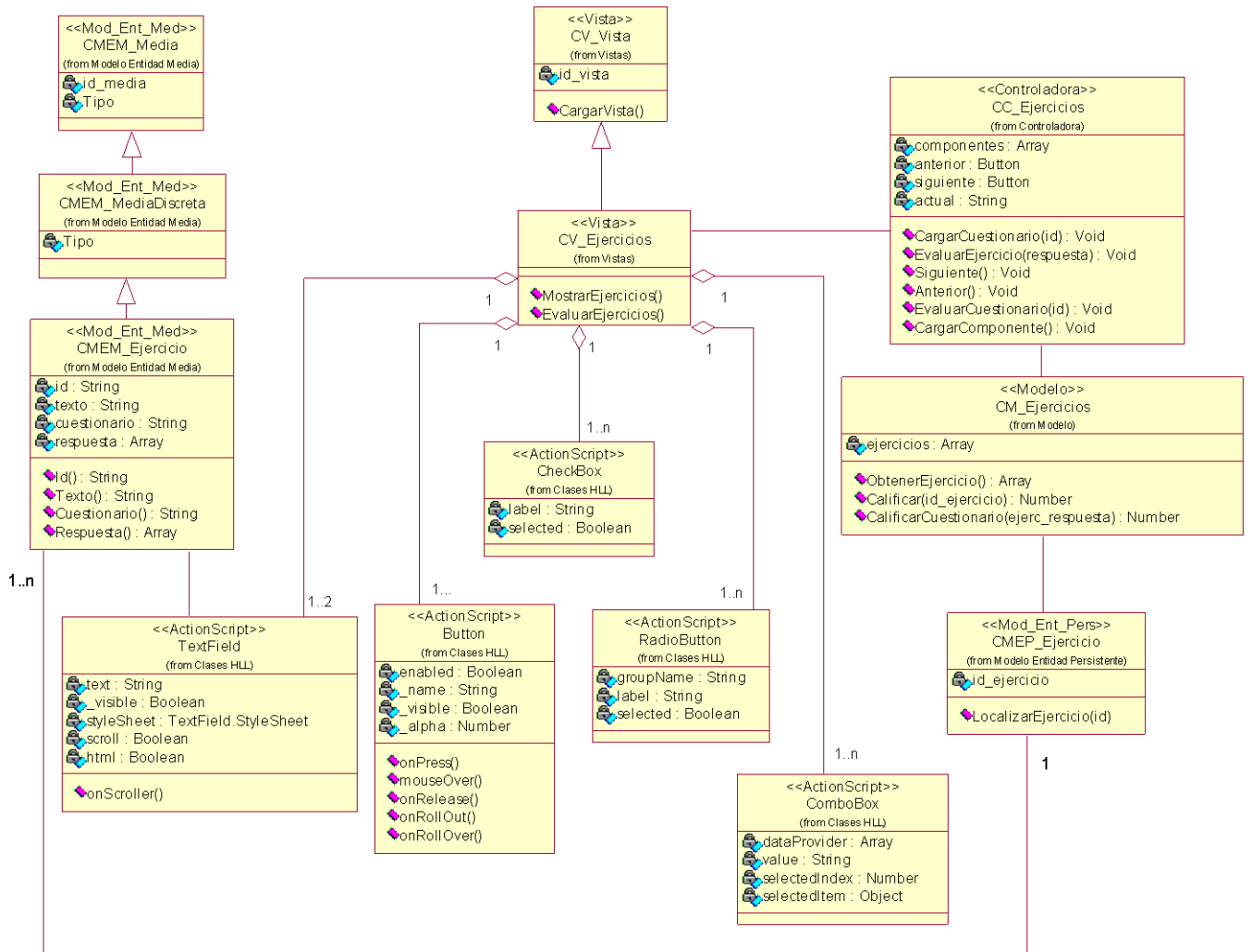
Anexo 12. DEP Vista Salir



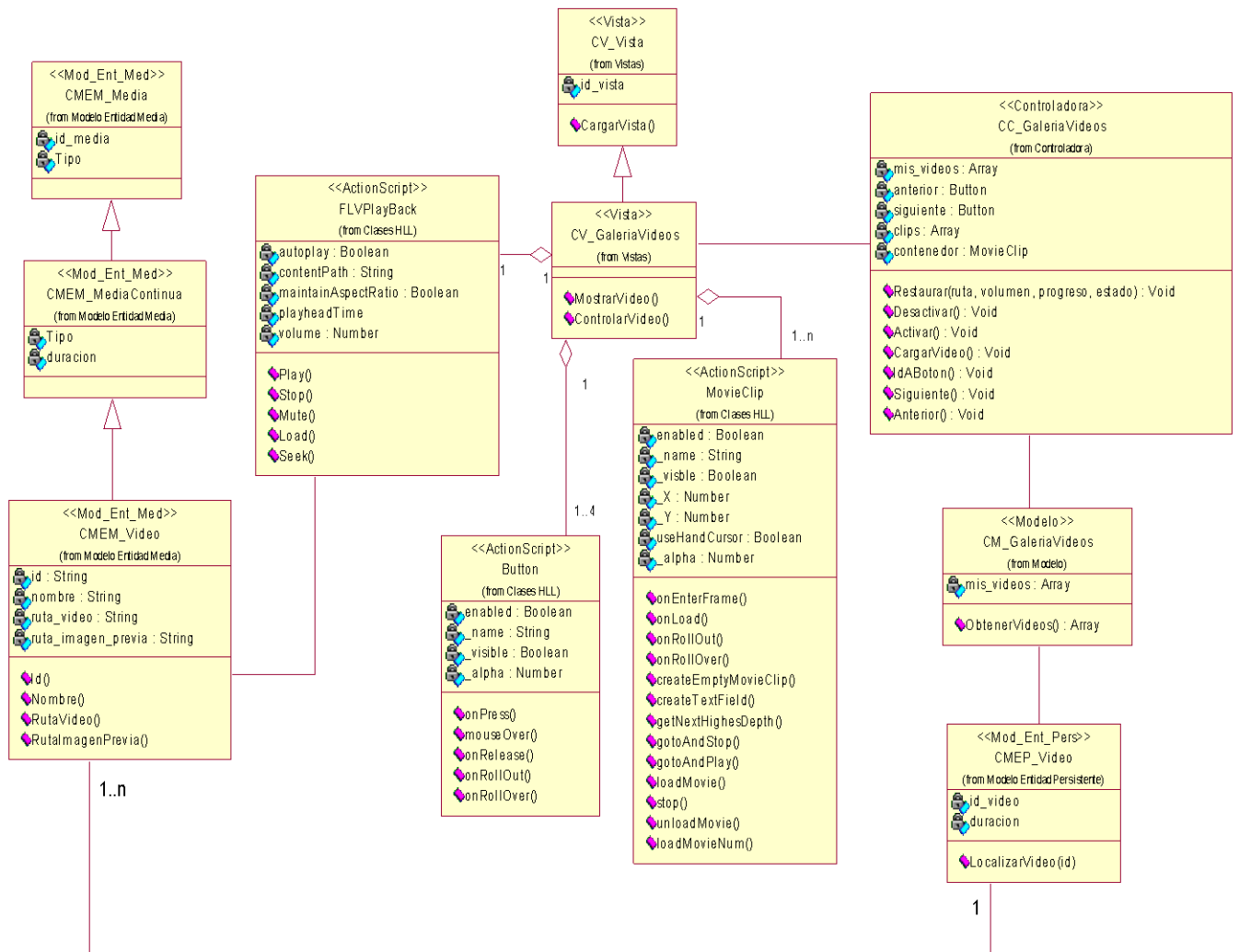
Anexo 13. DEP Vista Servicios



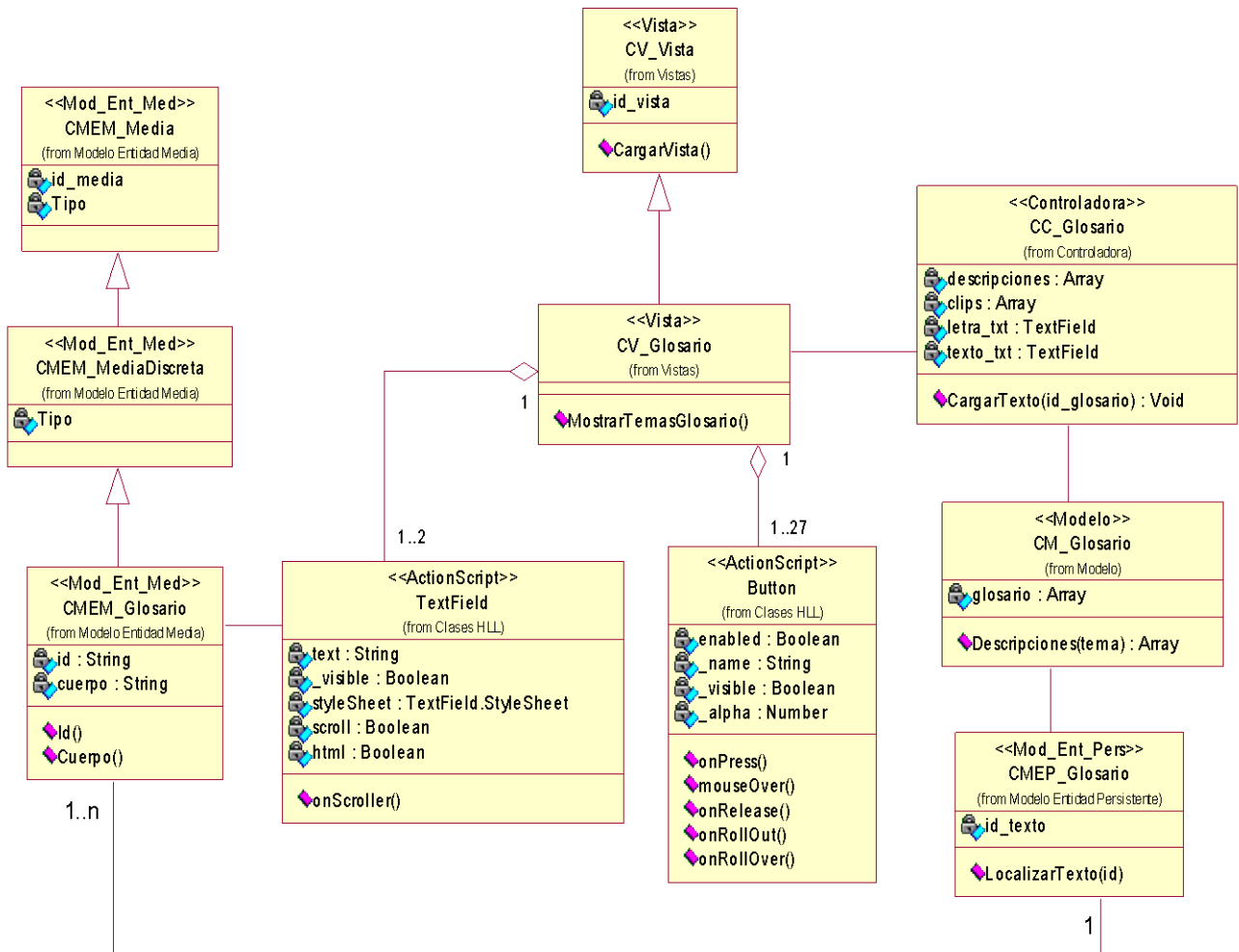
Anexo 14. Mapa de navegación general



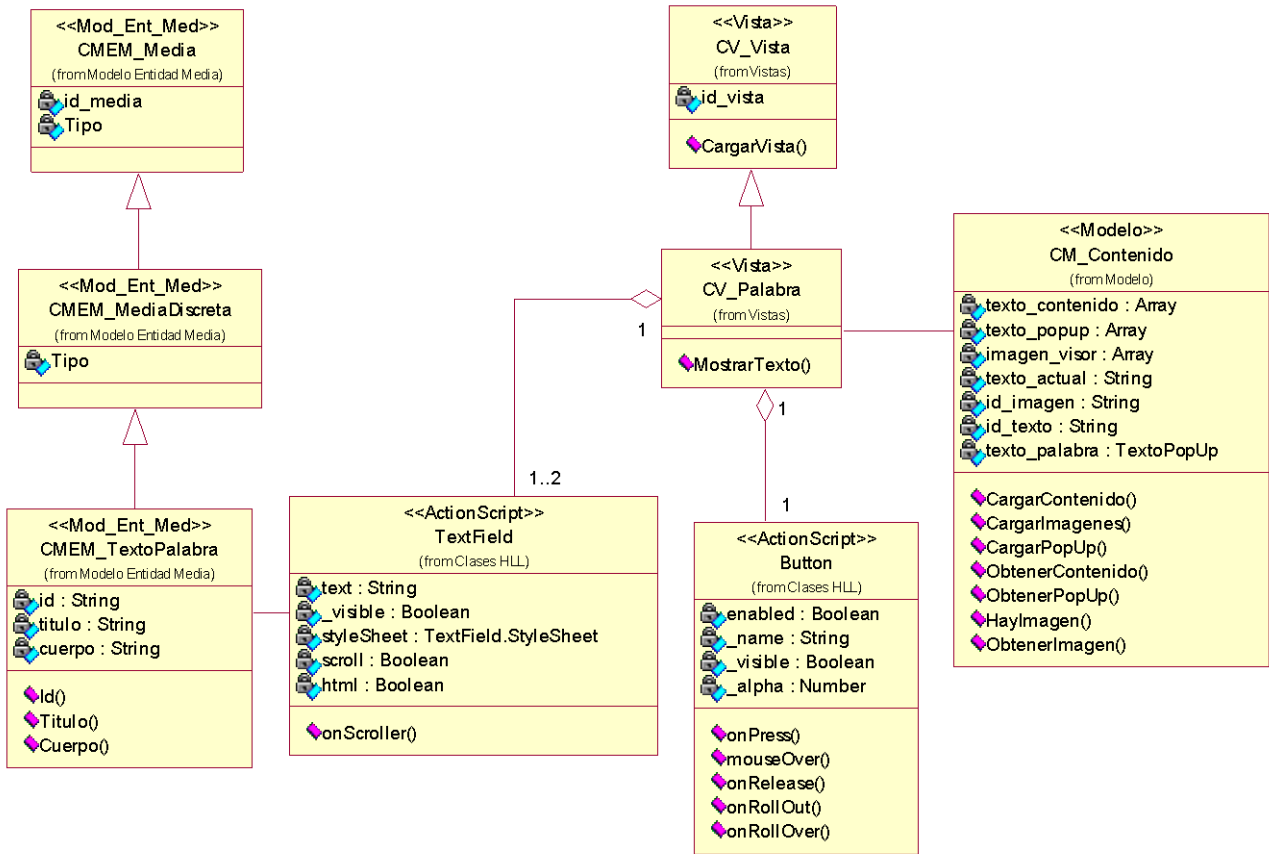
Anexo 15. Diagrama de Clases de la Vista Ejercicios



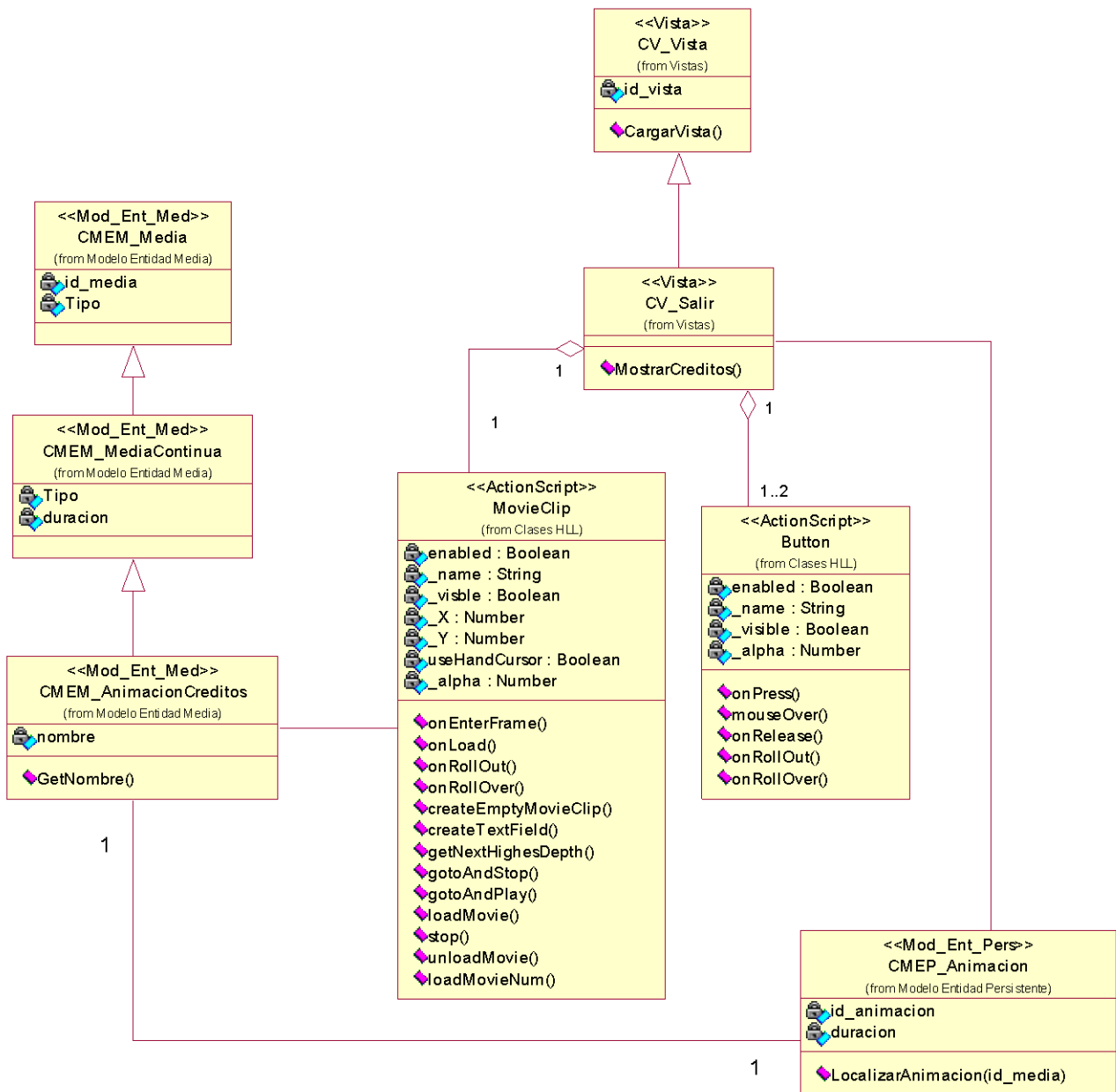
Anexo 16. Diagrama de Clases de la Vista Galería de Videos



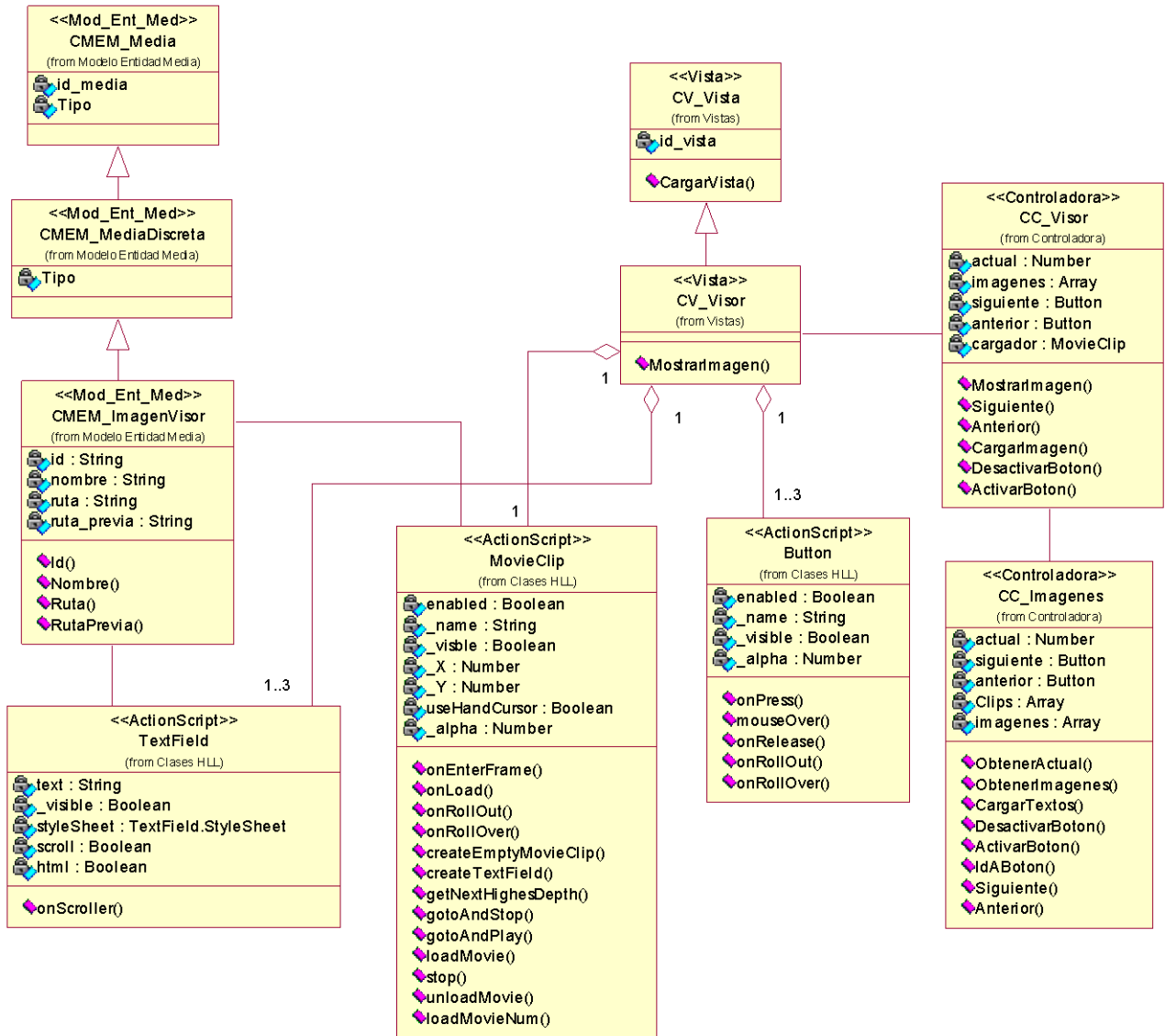
Anexo 17. Diagrama de Clases de la Vista Glosario



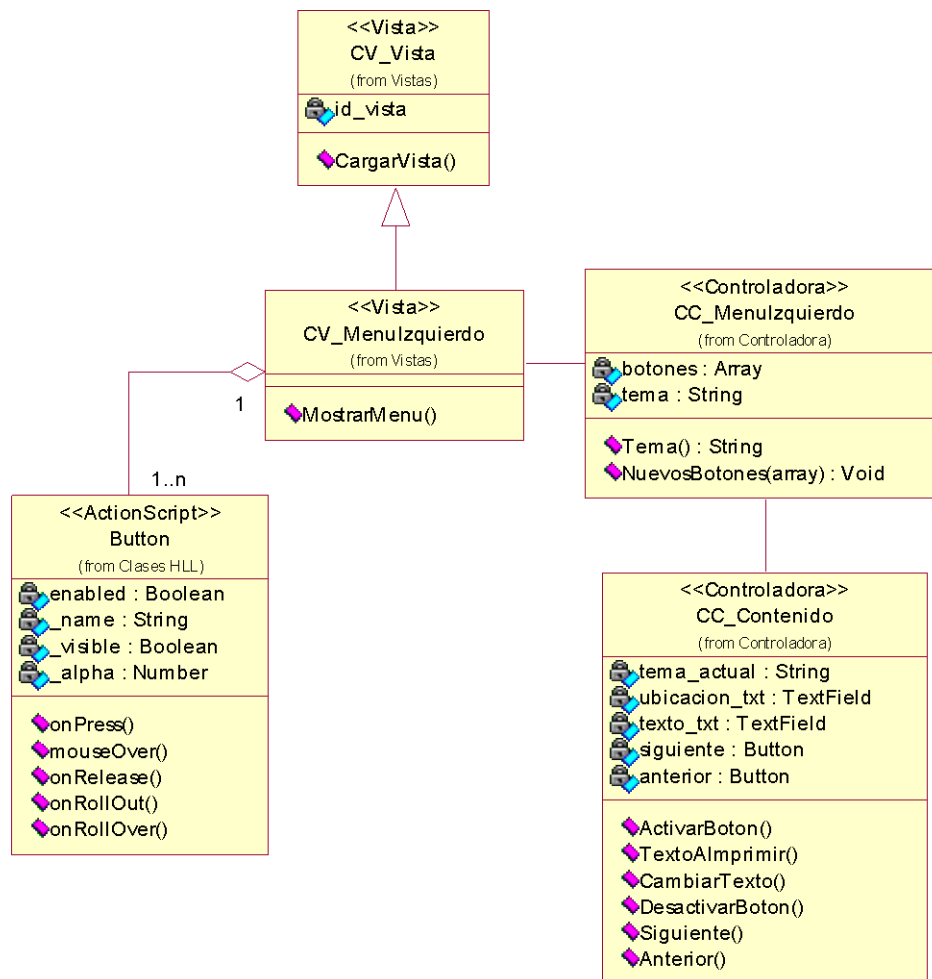
Anexo 18. Diagrama de Clases de la Vista Palabra



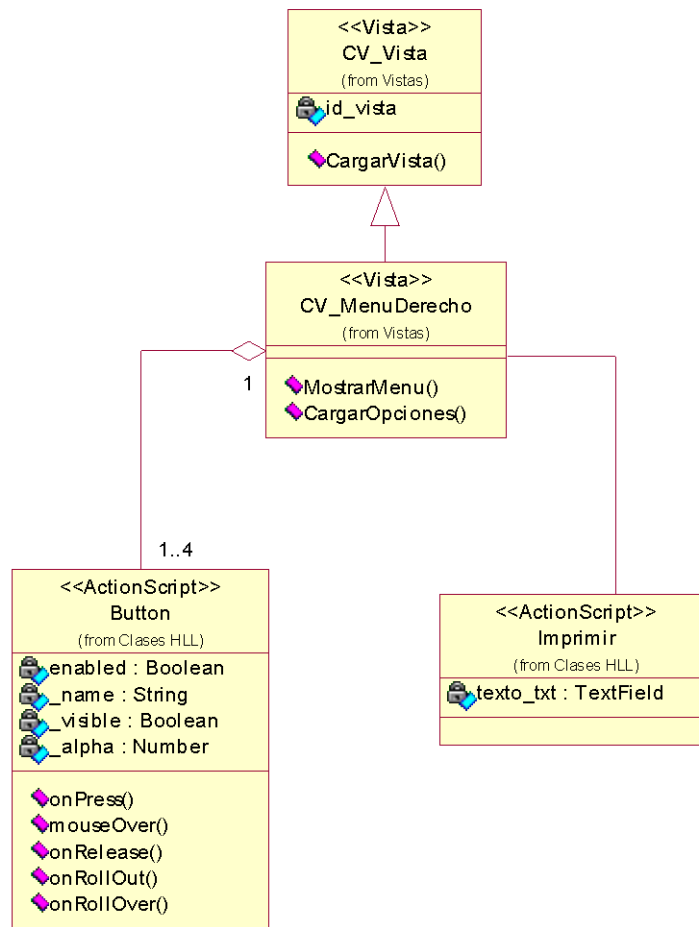
Anexo 19. Diagrama de Clases de la Vista Salir



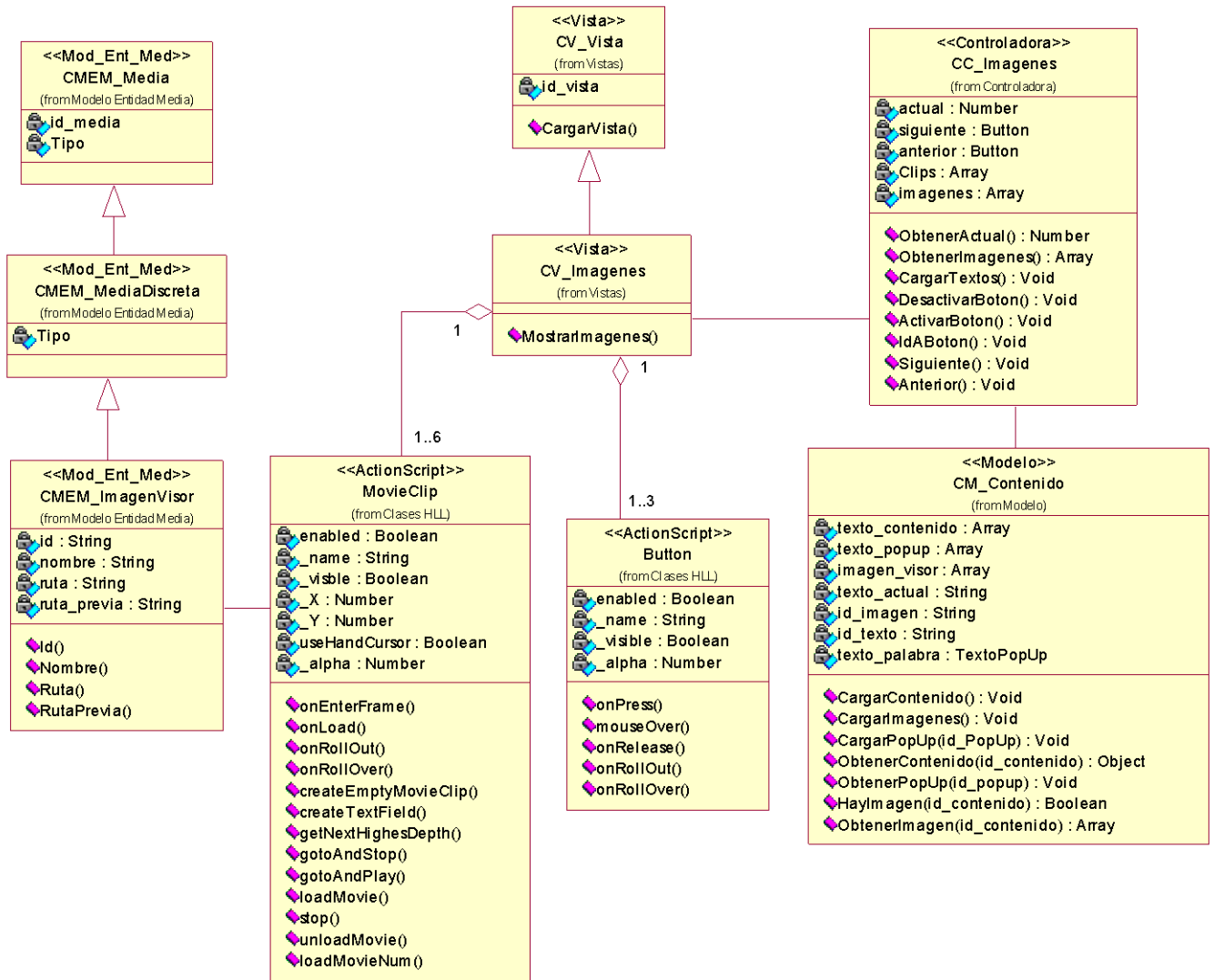
Anexo 20. Diagrama de Clases de la Vista Visor



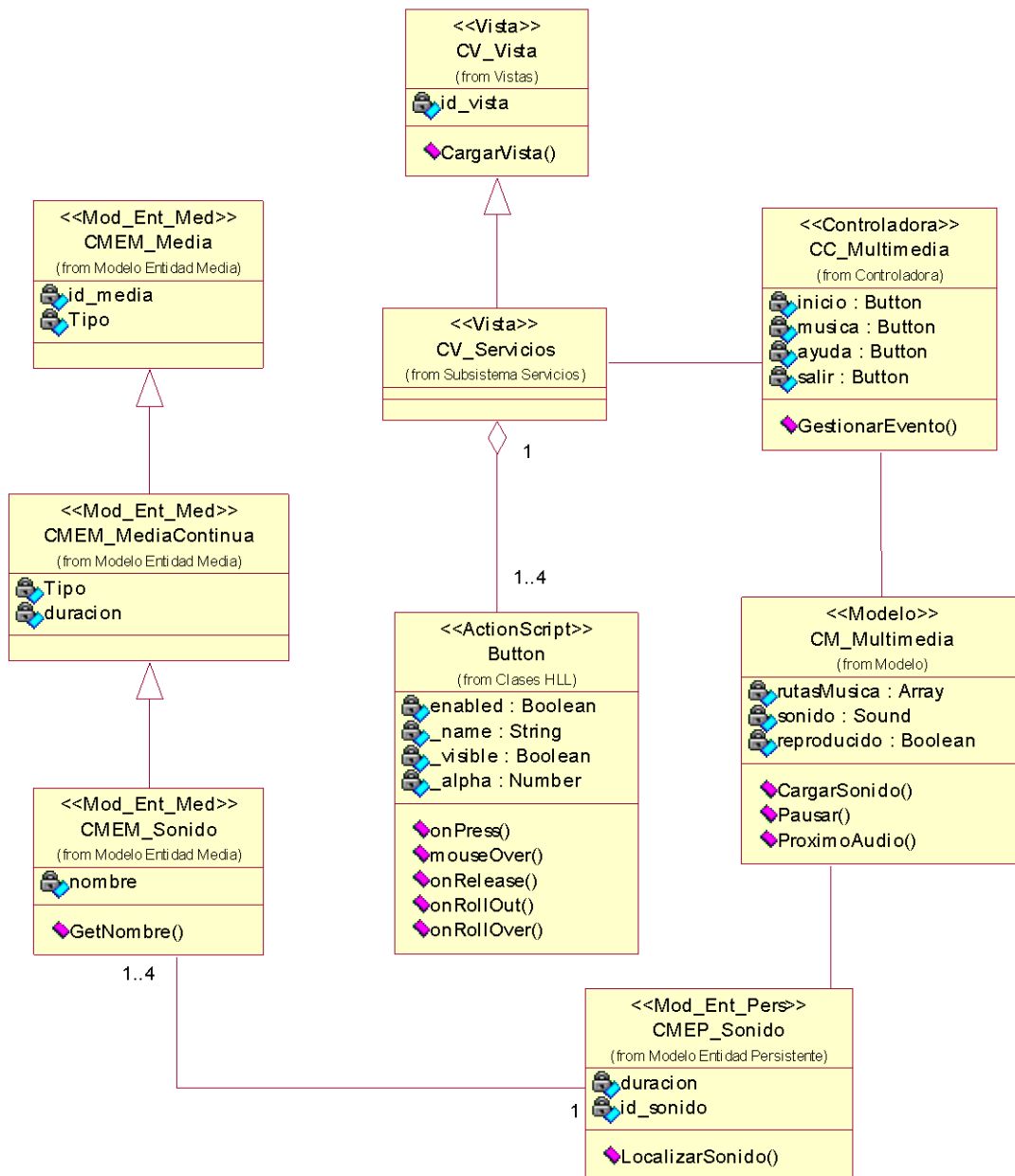
Anexo 21. Diagrama de Clases de la Vista Menú Izquierdo



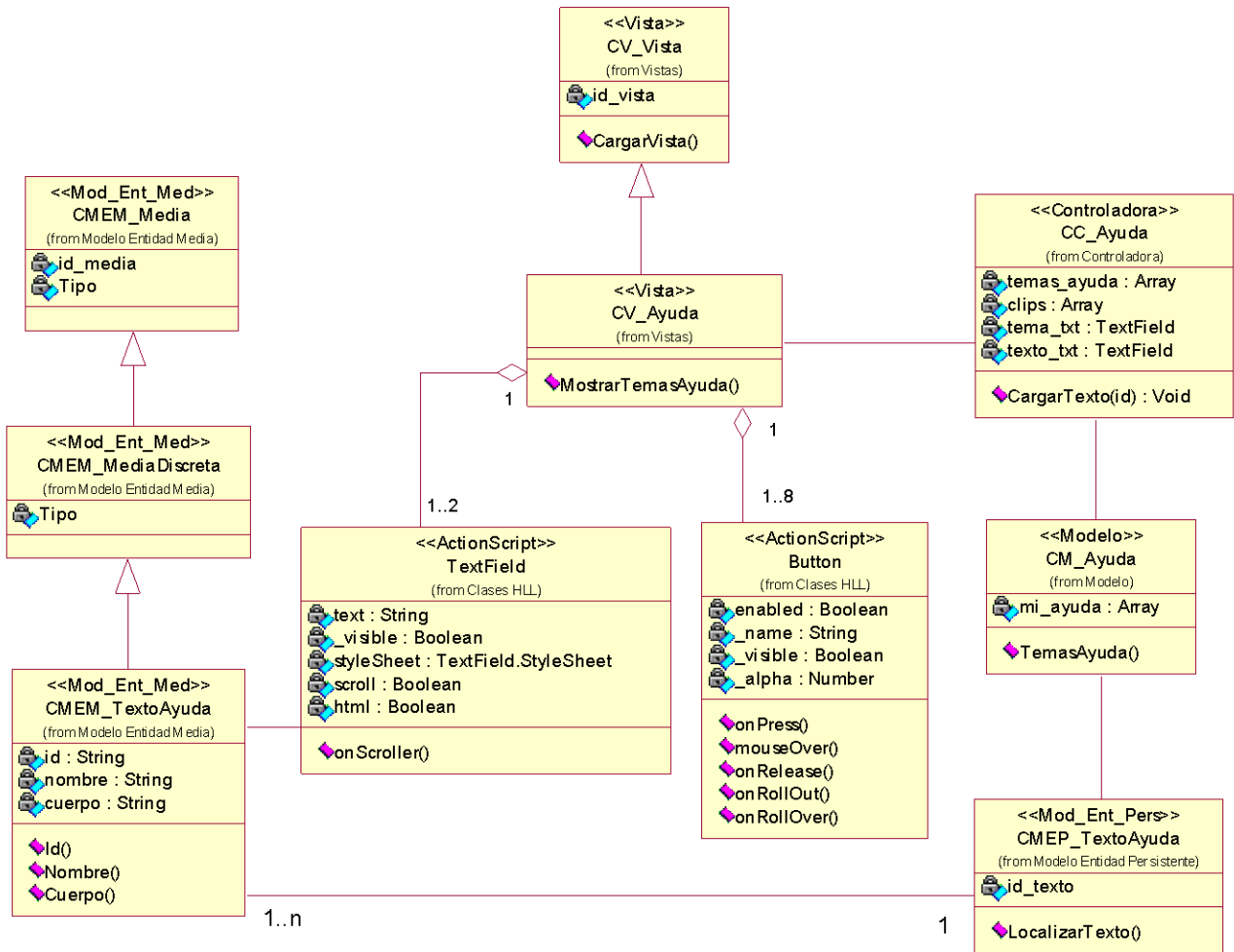
Anexo 22. Diagrama de Clases de la Vista Menú Derecho



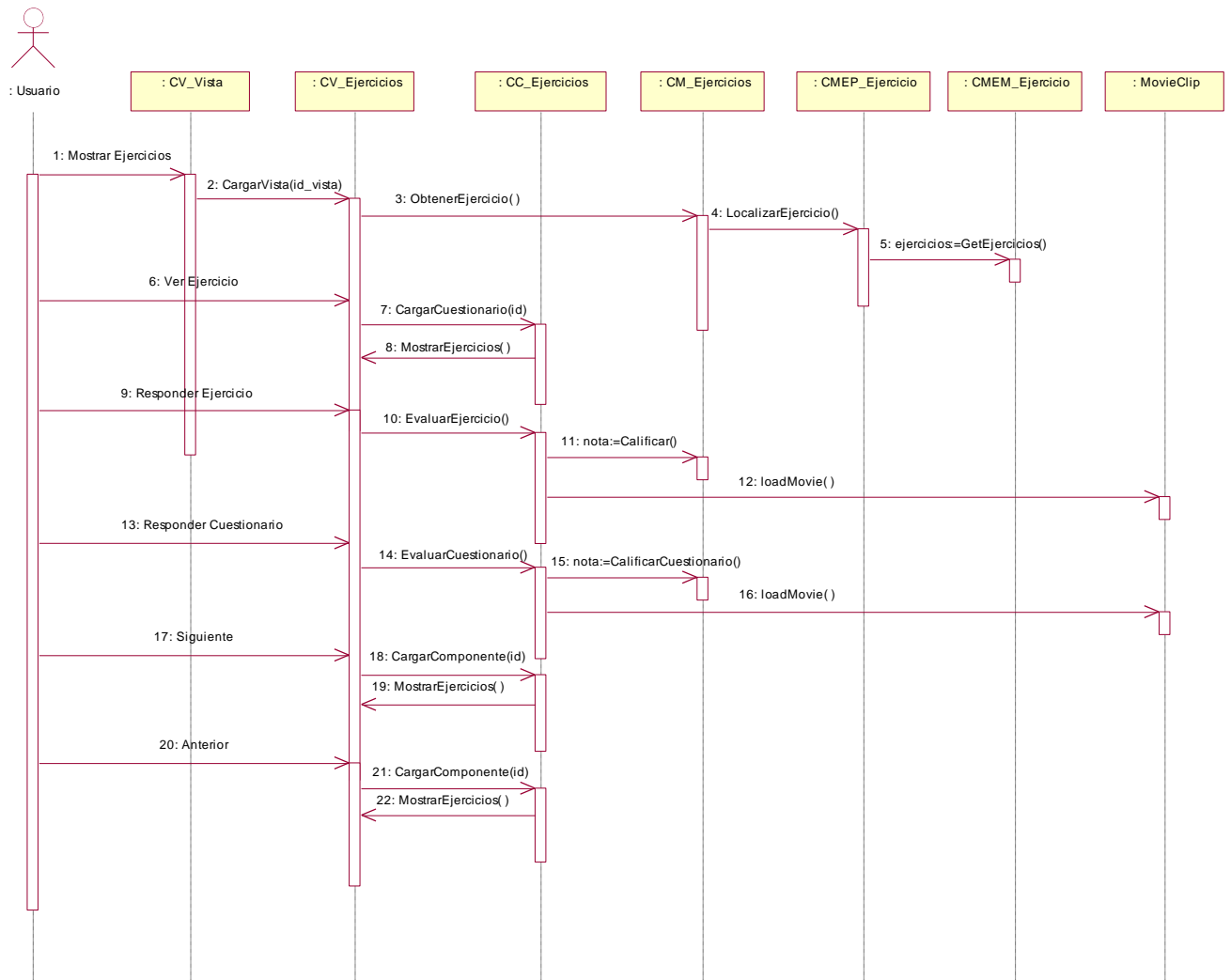
Anexo 23. Diagrama de Clases de la Vista Imágenes



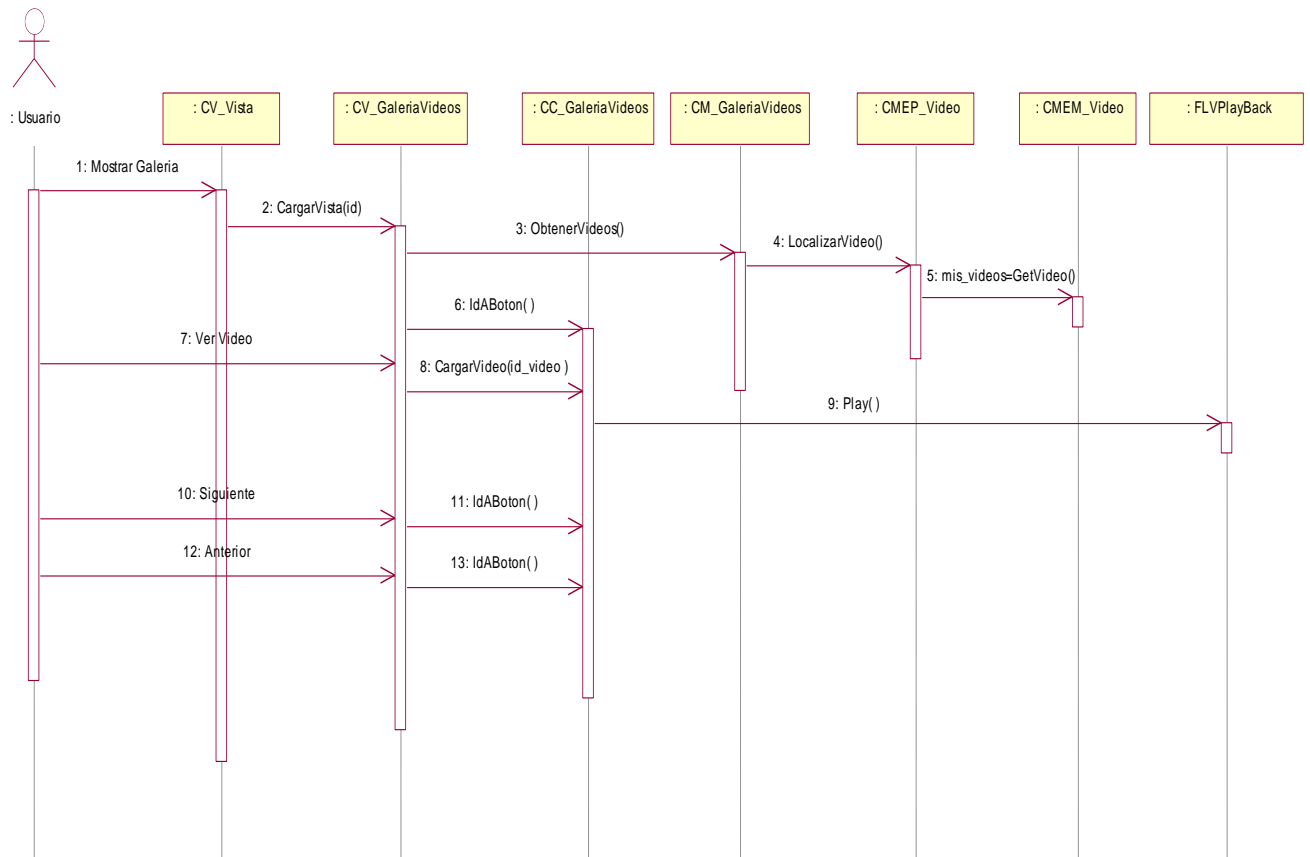
Anexo 24. Diagrama de Clases de la Vista Servicios



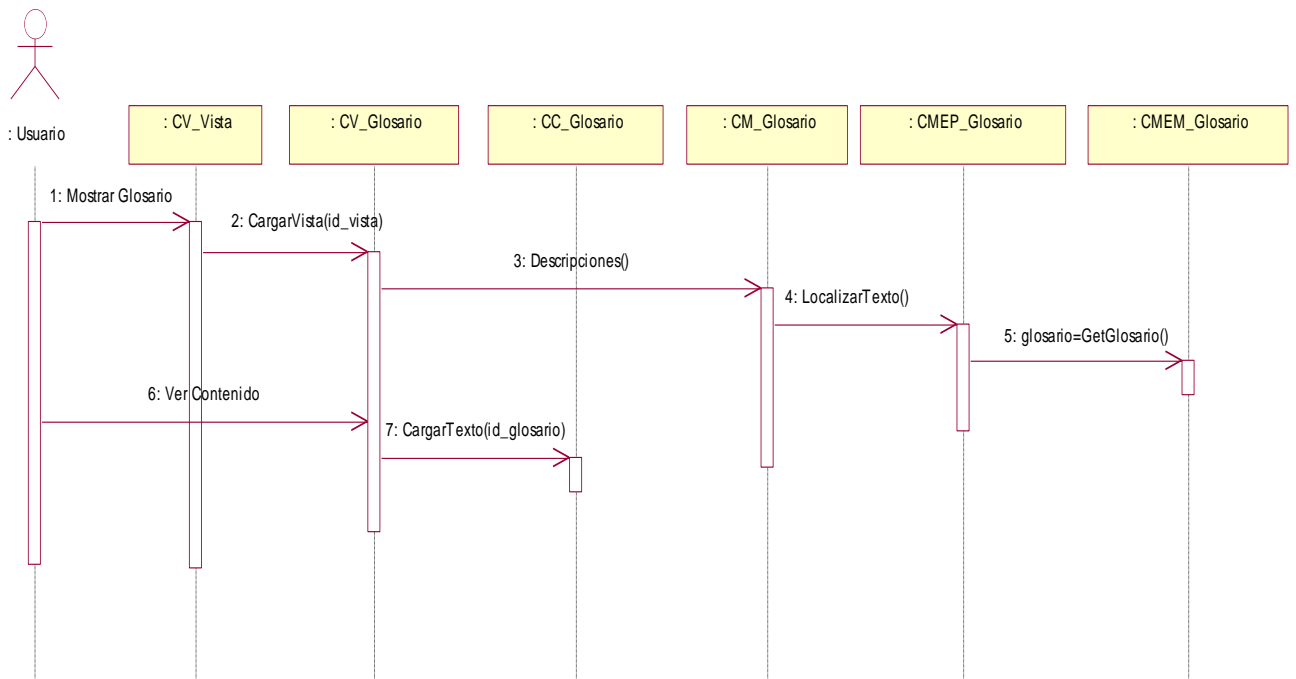
Anexo 25. Diagrama de Clases de la Vista Ayuda



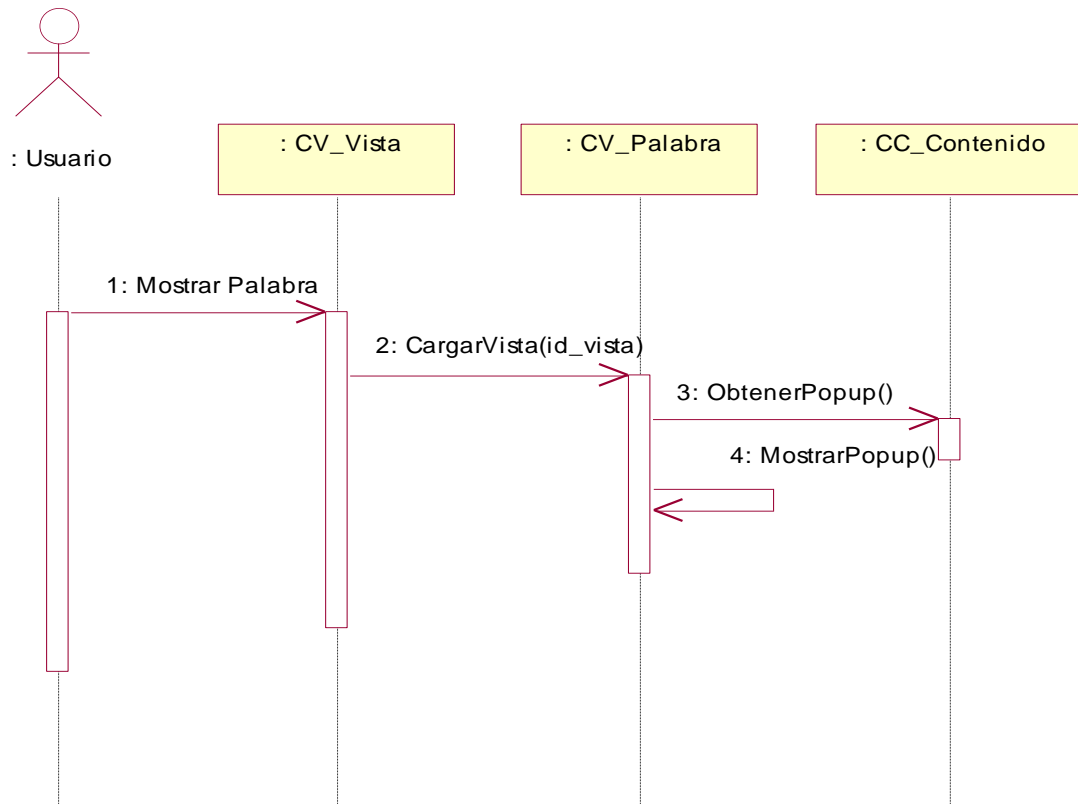
Anexo 26. Diagrama de Secuencia de la Vista Ejercicios



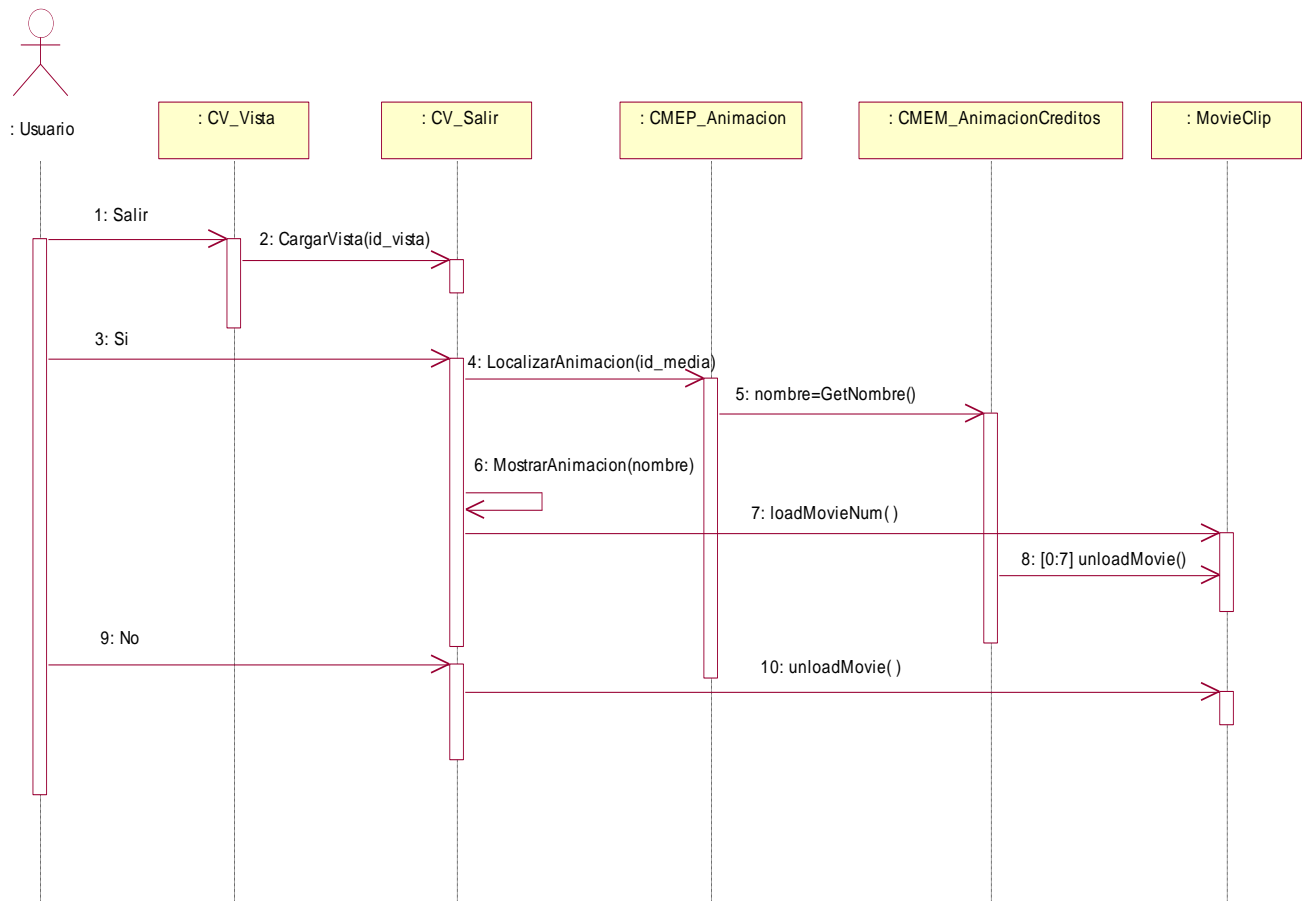
Anexo 27. Diagrama de Secuencia de la Vista Galería de Videos



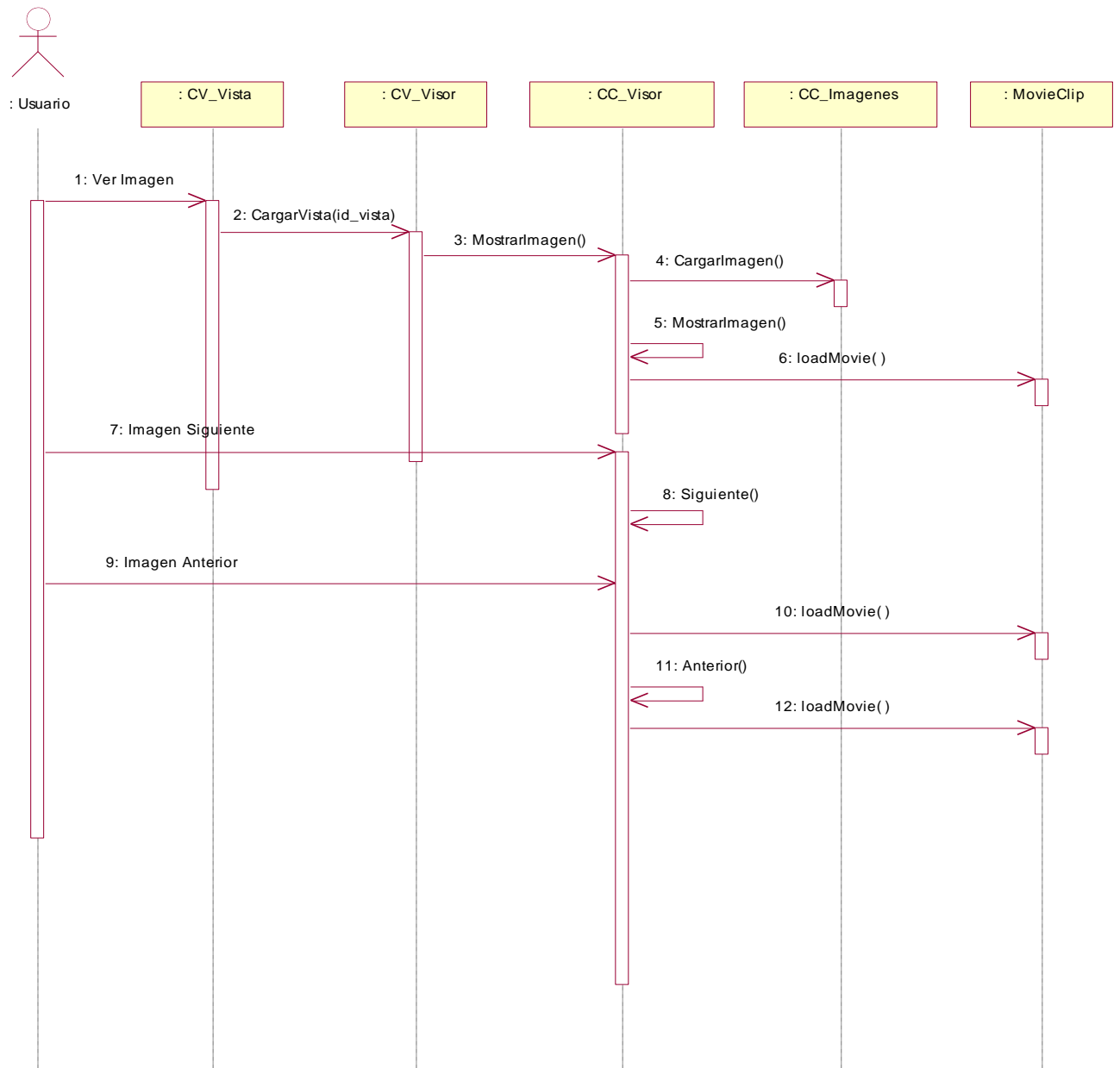
Anexo 28. Diagrama de Secuencia de la Vista Glosario



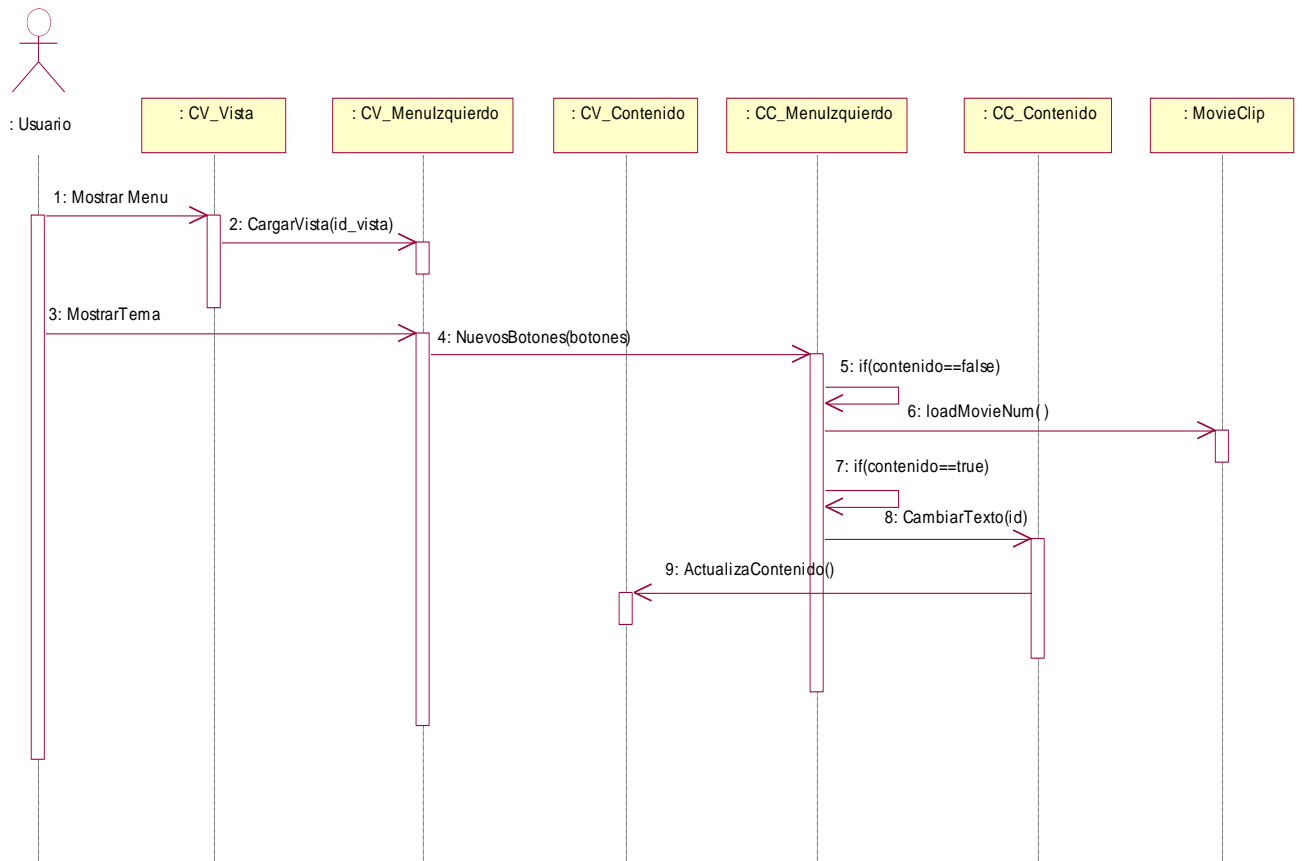
Anexo 29. Diagrama de Secuencia de la Vista Palabra



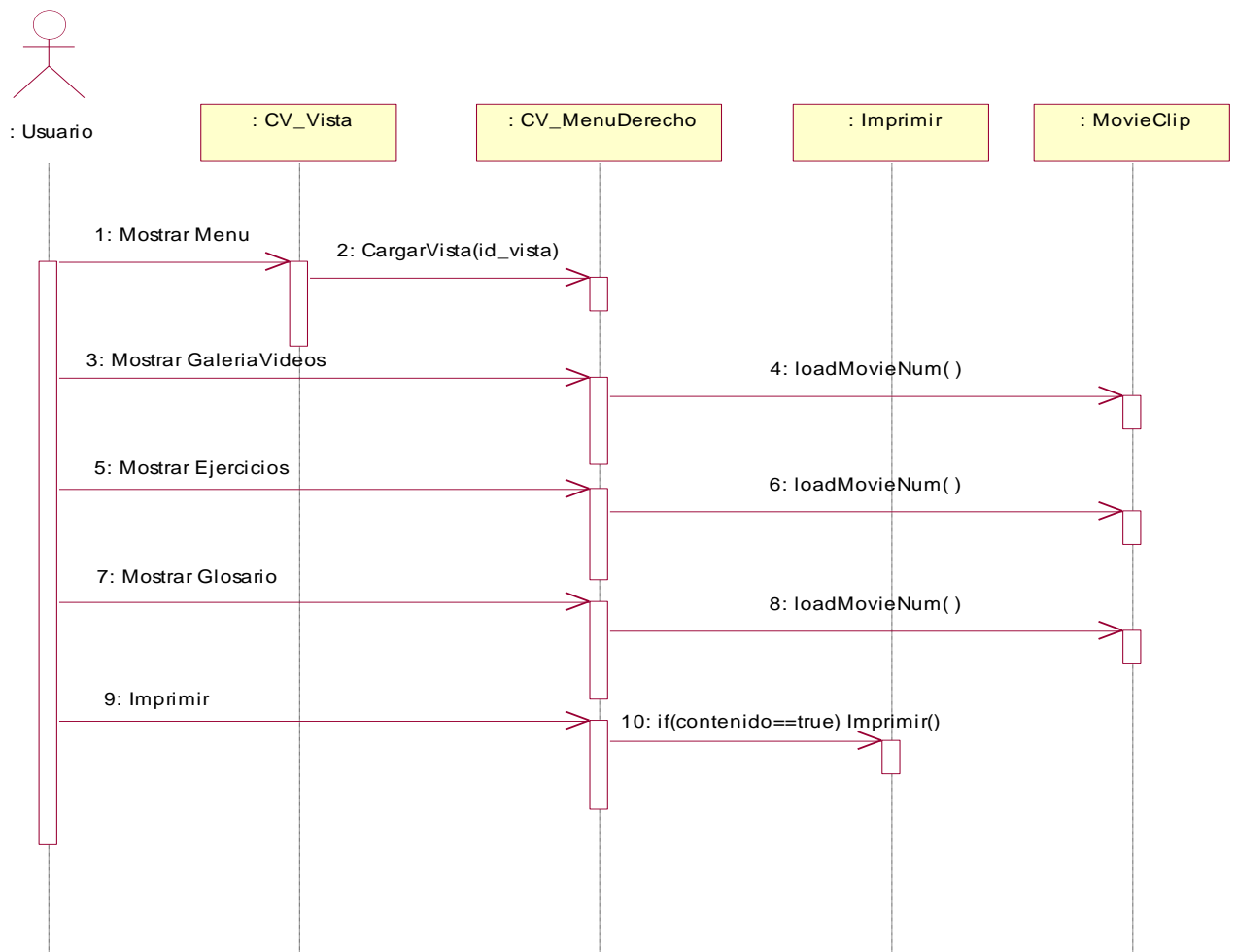
Anexo 30. Diagrama de Secuencia de la Vista Salir



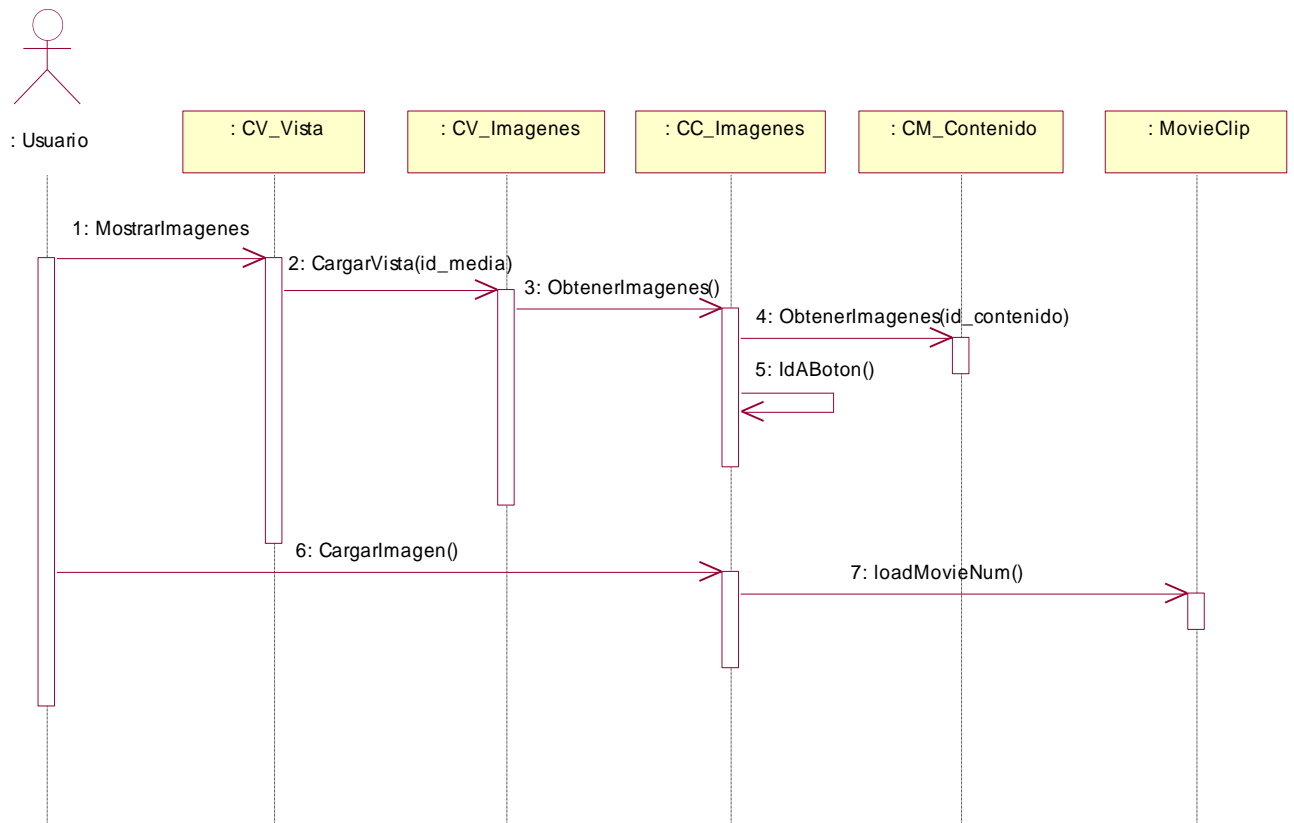
Anexo 31. Diagrama de Secuencia de la Vista Visor



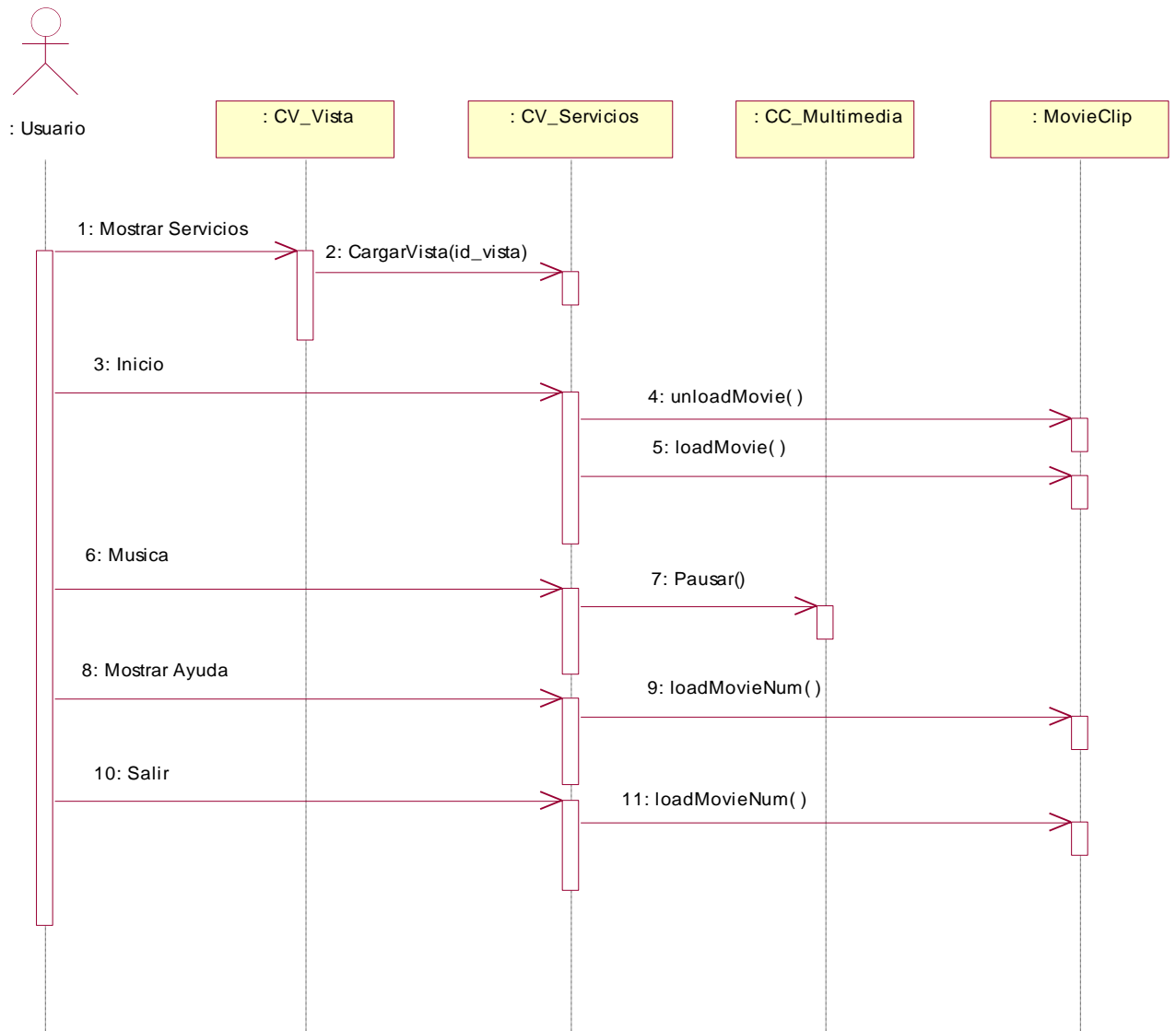
Anexo 32. Diagrama de Secuencia de la Vista Menú Izquierdo



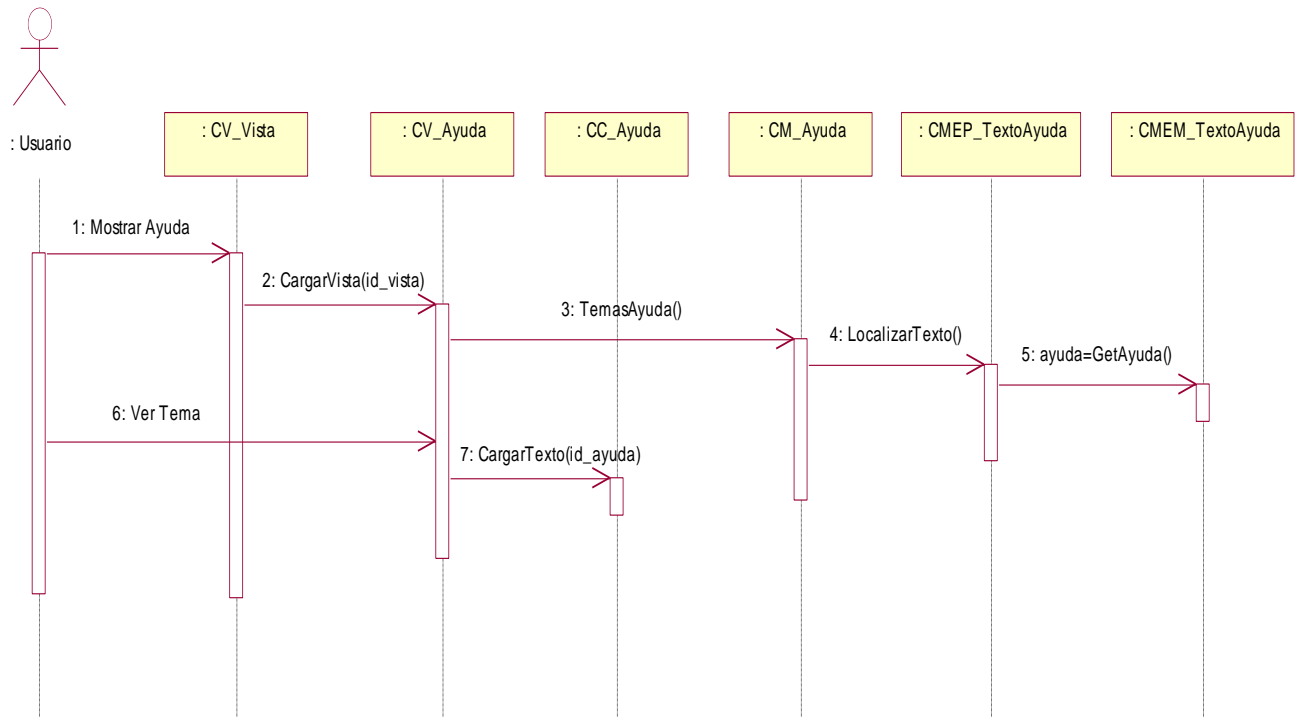
Anexo 33. Diagrama de Secuencia de la Vista Menú Derecho



Anexo 34. Diagrama de Secuencia de la Vista Imágenes



Anexo 35. Diagrama de Secuencia de la Vista Servicios



Anexo 36. Diagrama de Secuencia de la Vista Ayuda

Tabla 1: Descripción textual de la Vista Menú Izquierdo

Descripción textual de la Vista de Presentación “Menú Izquierdo”		
Actores de la Vista de Presentación	Usuario, Vista “Principal”	
Propósito	Permitir al usuario acceder a la pantalla de contenidos de la aplicación y así conocer sobre los diferentes temas abordados en la misma.	
Objetivos Pedagógicos	-	
Resumen	La Vista se inicia cuando se muestra la Vista “Principal”.	
Vistas asociadas	Vista “Contenido”, Vista “Principal”	
Referencias	R2.1	
Precondiciones	Debe haber iniciado la vista “Principal”.	
Poscondiciones	Vista de Presentación visualizada de acuerdo a la elección del usuario.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
	1. Se ejecuta la Vista “Menú_Izquierdo”.	<p>1.1 La vista tendrá una imagen de fondo (Imagen_FondoMenú).</p> <p>1.2 Se muestran los botones del menú. El mismo estará compuesto por un botón “Temas” y los botones de tema “Introducción al Symphony” que contiene a los botones subtemas: “¿Qué es Symphony?”, “¿Quién ha desarrollado Symphony y por qué?”, “Principales características”, “¿Que automatiza para los proyectos WEB?”; el botón tema “Comenzando el proyecto” que contiene</p>

		<p>los botones subtemas: “Prerrequisitos”, “Instalación de Symfony”, “Crear el proyecto”, “Crear la aplicación”, “La ruta de Symfony”, “Configuración del servidor WEB”, “Probar la nueva configuración”; el botón de tema “Configuración de Symfony” que contiene los botones subtemas: “El sistema de configuración”, “Configuración del proyecto”, “Configuración de la aplicación”, “Configuración de los módulos”; y los botones de tema “¿Cómo implementa Symfony el MVC?”, “Trabajo con formularios” y “Proceso de despliegue de la instalación”.</p>
2. El usuario selecciona el botón “Temas”.	3. Se despliega el menú.	3.1 El menú se desliza horizontalmente de izquierda a derecha mostrando los botones contenidos en el mismo.
4. El usuario selecciona un botón tema.	5. Se muestran los botones subtemas correspondientes al tema.	
6. El usuario selecciona un botón subtema.	7. Se ejecuta la Vista “ <i>Contenido</i> ” correspondiente al subtema seleccionado por el usuario.	
Curso alternativo de los eventos		

Acción		Curso alternativo	
6		Si la Vista "Contenido" ya está visualizada no se vuelve a cargar, solo se actualiza su contenido de acuerdo al subtema seleccionado.	
Prioridad		Crítica	
Mejoras		-	
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_FondoMenú	Imagen de color negro semitransparente, de forma rectangular con las esquinas redondeadas.	Existente
Video o Animación	-	-	-
Sonido	-	-	-
Texto	-	-	-
Reglas Pedagógicas			
-			

Tabla 2: Descripción textual de la Vista Menú Derecho

Descripción textual de la Vista de Presentación "Menú Derecho"	
Actores de la Vista de Presentación	Usuario, Vista "Principal"
Propósito	Permitir al usuario acceder a la Galería de Videos, al Glosario y a los Ejercicios y acceder también al servicio de imprimir el subtema que se esté visualizando.
Objetivos Pedagógicos	-
Resumen	La Vista se inicia cuando se muestra la Vista "Principal".
Vistas asociadas	Vista "Principal", Vista "GaleriaVideos", Vista "Glosario", Vista "Ejercicios".
Referencias	R2.1, R3.4
Precondiciones	Debe haber iniciado la vista "Principal".
Poscondiciones	Vista de Presentación visualizada de acuerdo a la elección del usuario.

Curso normal de eventos			
Acciones del actor	Respuesta del sistema	Elementos de la vista	
	1. Se ejecuta la Vista "Menú_Derecho".	1.1 La vista tendrá una imagen de fondo (Imagen_FondoMenú). 1.2 Se muestra un botón "Opciones" y cuatro botones de texto, "Galería de Videos", "Ejercicios", "Glosario" e "Imprimir".	
2. El usuario selecciona el botón "Opciones".	3. Se despliega el menú.	3.1 El menú se desliza horizontalmente de derecha a izquierda mostrando los botones contenidos en el mismo.	
4. El usuario selecciona la opción "Galería de Videos".	5. Se ejecuta la Vista "GaleriaVideos".		
6. El usuario selecciona la opción "Ejercicios".	7. Se ejecuta la Vista "Ejercicios".		
8. El usuario selecciona la opción "Glosario".	9. Se ejecuta la Vista "Glosario".		
10. El usuario selecciona la opción "Imprimir".	11. Se manda a imprimir el contenido del subtema que el usuario esté visualizando.		
Curso alternativo de los eventos			
Acción	Curso alternativo		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_FondoMenú	Imagen de color negro semitransparente, de forma rectangular con las esquinas redondeadas.	Existente
Video o Animación	-	-	-
Sonido	-	-	-

Texto	-	-	-
Reglas Pedagógicas			
-			

Tabla 3: Descripción textual de la Vista Imágenes

Descripción textual de la Vista de Presentación “Imágenes”		
Actores de la Vista de Presentación	Usuario, Vista “Contenido”	
Propósito	Mostrar al usuario las imágenes relacionadas con cada tema de la aplicación.	
Objetivos Pedagógicos	-	
Resumen	La vista se inicia cuando se ejecuta o actualiza la Vista “Contenido” y el subtema mostrado tiene imágenes asociadas.	
Vistas asociadas	Vista “Visor”, Vista “Contenido”	
Referencias	R1.4	
Precondiciones	Debe haber iniciado la vista “Contenido”.	
Poscondiciones	Imagen visualizada de acuerdo a la elección del usuario.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
	1. Se ejecuta la Vista “Imágenes”.	1.1 La vista tendrá una imagen de fondo (Imagen_FondoImagenes). 1.2 Se muestra una cantidad de botones de imágenes de acuerdo a la cantidad de imágenes asociadas al subtema, como máximo 6, que conducirán a las imágenes presentes en el tema, también tendrá un botón “Siguiente Imágenes” y “Anterior Imágenes”

		(representados con flechas orientadas a la derecha y a la izquierda respectivamente y ubicados en la parte inferior de la Vista) que permanecerán activos en caso de existir más de 6 imágenes asociadas al subtema que se esté visualizando, permitiendo al usuario ver todas las imágenes.	
2. El usuario selecciona un botón de imagen.	3. Se ejecuta la vista “ <i>Visor</i> ” mostrando la imagen correspondiente al botón seleccionado por el usuario.		
4. El usuario selecciona el botón “Siguiente”.	5. Se actualiza el contenido de la Vista “ <i>Imágenes</i> ”, mostrando los botones imágenes siguientes.		
6. El usuario selecciona el botón “Anterior”.	7. Se actualiza el contenido de la Vista “ <i>Imágenes</i> ”, mostrando los botones imágenes anteriores.		
Curso alternativo de los eventos			
Acción	Curso alternativo		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_Fondolimagenes.	Imagen de color negro semitransparente, de forma rectangular con las esquinas redondeadas.	Existente
Video o Animación	-	-	-
Sonido	-	-	-
Texto	-	-	-
Reglas Pedagógicas			

-

Tabla 4: Descripción textual de la Vista Palabra

Descripción textual de la Vista de Presentación “Palabra”		
Actores de la Vista de Presentación	Usuario	
Propósito	Ampliar la información mostrada en los contenidos de la aplicación.	
Objetivos Pedagógicos	Mostrar al usuario el significado de algunas palabras de difícil comprensión (palabras calientes de tipo texto) relacionadas con el tema.	
Resumen	La vista se inicia cuando el usuario accede a la palabra caliente de tipo texto ubicada en el texto de la Vista “Contenido”.	
Vistas asociadas	Vista “Contenido”	
Referencias	R1.5	
Precondiciones	Haberse ejecutado la Vista “Contenido”.	
Poscondiciones	Vista “Palabra” visualizada de acuerdo a la elección del usuario.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. El usuario acciona una palabra caliente de tipo texto del contenido.	2. Se ejecuta la Vista “Palabra”.	2.1 La vista tendrá una imagen de fondo (Imagen_FondoPalabra). 2.2 Se muestra un texto que contiene el título (TextoTítulo) de la palabra caliente, ubicado en la parte superior de la Vista y además se muestra otro texto con la descripción de la misma (TextoPalabra), ubicado en el centro de la Vista. 2.3 Se muestra el botón de imagen “CerrarPalabra” que permite salir de la vista, ubicada en la parte superior derecha de la Vista.

3. El usuario acciona el botón "CerrarPalabra".	4. Culmina la Vista de Presentación.		
Curso alternativo de los eventos			
Acción	Curso alternativo		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_FondoPalabra.	Imagen de color negro, la parte superior de la imagen es una barra de color carmelita con un icono que representa una hoja en la parte izquierda.	Existente
Video o Animación	-	-	-
Sonido	-	-	-
Texto	- TextoPalabra. - TextoTítulo.	- Muestra una descripción de la palabra. El formato del mismo es: fuente Verdana, con un tamaño igual a 11, color blanco y alineación justificado. - Texto que muestra el título, alineado a la izquierda, de color blanco, fuente Tahoma, tamaño 11.	Existente

Reglas Pedagógicas
-

Tabla 5: Descripción textual de la Vista Visor

Descripción textual de la Vista de Presentación “Visor”		
Actores de la Vista de Presentación	Usuario	
Propósito	Mostrar al usuario las diferentes imágenes ampliadas relacionadas con el tema.	
Objetivos Pedagógicos	Mostrar al usuario las diferentes imágenes que enriquecen el texto.	
Resumen	La vista se inicia cuando el usuario accede al botón imagen contenido en la vista “Imágenes” o a las palabras calientes de tipo imagen contenidas en los textos de la Vista “Contenido”.	
Vistas asociadas	Vista “Contenido”, Vista “Imágenes”	
Referencias	R1.4	
Precondiciones	Haberse ejecutado la Vista “Imágenes”.	
Poscondiciones	Vista “Visor” mostrada de acuerdo a la elección del usuario.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. El usuario acciona una palabra caliente de tipo imagen de la Vista “Contenidos” o un botón imagen de la vista “Imágenes”.	2. Se ejecuta la Vista “Visor”.	2.1 La vista tendrá una imagen de fondo (Imagen_FondoVisor). 2.2 Se muestra un texto con el nombre de la imagen que se esté visualizando (TextoVisor). 2.3 Se muestra una imagen (Imagen_Tema). 2.4 Se muestran un botón “SiguienteVisor” y “AnteriorVisor” (representados con flechas orientadas a la derecha y a la izquierda

		respectivamente y ubicados en la parte inferior de la Vista) que permiten al usuario ver todas las imágenes asociadas al subtema que se esté visualizando. Se muestra también el botón "CerrarVisor" que permite salir de la vista, ubicado en la parte superior derecha de la Vista.
3 El usuario acciona el botón "SiguienteVisor".	4 Se actualiza la vista Visor, mostrando la Imagen_Tema siguiente.	
5 El usuario acciona el botón "AnteriorVisor".	6 Se actualiza la vista Visor, mostrando la Imagen_Tema anterior.	
7 El usuario acciona el botón "CerrarVisor".	8 Finaliza la Vista de Presentación.	

Curso alternativo de los eventos

Acción	Curso alternativo
Prioridad	Crítica
Mejoras	-

Medias a utilizar

Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_FondoVisor.	Imagen de color negro, la parte superior de la imagen es una barra de color carmelita con un icono que representa un dibujo en la parte izquierda.	Existente
Imagen	Imagen_Tema	Imagen relacionada con el subtema. Las dimensiones de la misma son 470x320 pixel.	Existente
Video o Animación	-	-	-

Sonido	-	-	-
Texto	TextoVisor	Muestra el nombre de la imagen que se esté visualizando. El texto está alineado a la izquierda, con fuente Tahoma, un tamaño igual a 11 y de color blanco.	Existente
Reglas Pedagógicas			
-			

Tabla 6: Descripción textual de la Vista Ayuda

Descripción textual de la Vista de Presentación “Ayuda”		
Actores de la Vista de Presentación	Usuario	
Propósito	Mostrar al usuario la información de ayuda que necesita para navegar en la aplicación.	
Objetivos Pedagógicos	Enseñar al usuario como trabajar con la aplicación y utilizar todas sus prestaciones.	
Resumen	La vista se inicia cuando el usuario accede a la opción Ayuda ubicada en la vista “Servicios”.	
Vistas asociadas	Vista “Servicios”	
Referencias	R3.1	
Precondiciones	-	
Poscondiciones	Vista “Ayuda” visualizada.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. El usuario acciona el botón Ayuda de la vista “Servicios”.	2. Se carga la Vista “Ayuda”.	2.1 La vista tendrá una imagen de fondo (Imagen_FondoAyuda). 2.2 Se muestra un texto (TextoAyuda) que describe la información de ayuda necesaria para los usuarios e imágenes

		<p>(ImagenAyuda) acompañando a dicho texto para una mejor comprensión del mismo.</p> <p>2.3 Tendrá 8 botones de texto que conducen a los diferentes temas (“¿Cómo ver un contenido?”, “¿Cómo ver un video?”, “¿Cómo acceder a los Ejercicios?”, “¿Cómo imprimir un contenido?”, “¿Cómo trabajar con el Glosario?”, “¿Cómo controlar la música?”, “¿Cómo volver al inicio?”, “¿Cómo salir de la aplicación?”) abordados en la ayuda y un botón de imagen “CerrarAyuda” que permite salir de la Vista, ubicado en la parte superior derecha.</p>	
3	El usuario selecciona uno de los botones de texto contenidos en la vista.	4 Se carga el tema correspondiente de ayuda, de acuerdo a la elección del usuario.	
5	El usuario selecciona el botón “CerrarAyuda”.	6 Finaliza la Vista de Presentación.	
Curso alterno de los eventos			
Acción		Curso alterno	
Prioridad		Secundaria	
Mejoras		-	
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_FondoAyuda	Imagen de color negro, la parte superior de la	Existente

		imagen es una barra de color carmelita con un icono que muestra una hoja con un signo de interrogación en la parte izquierda.	
Imagen	ImagenAyuda	Imagen relacionada con el tema de ayuda seleccionado.	Existente
Video o Animación	-	-	-
Sonido	-	-	-
Texto	TextoAyuda	Muestra los diferentes temas abordados en la vista. El texto es de color blanco, está justificado, la fuente es Verdana con un tamaño igual a 11.	Existente
Reglas Pedagógicas			
-			

Tabla 7: Descripción textual de la Vista Ejercicios

Descripción textual de la Vista de Presentación “Ejercicios”	
Actores de la Vista de Presentación	Usuario
Propósito	Mostrar al usuario una serie de ejercicios que le ayudarán a consolidar todo lo visto en los temas de la aplicación.
Objetivos Pedagógicos	Comprobar los conocimientos adquiridos por el usuario a través del uso de la aplicación.
Resumen	La vista se inicia cuando el usuario accede a la opción “Ejercicios” a través de la Vista “Menú_Derecho”.
Vistas asociadas	Vista “Menú_Derecho”.
Referencias	R4.1, R4.2, R4.3
Precondiciones	El usuario debe haber seleccionado la opción “Ejercicios” contenida en la Vista “Menú_Derecho”.

Poscondiciones		Vista "Ejercicios" visualizada.
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. El usuario acciona el botón "Ejercicios" de la Vista "Menú_Derecho".	2. Se carga la Vista "Ejercicios".	<p>2.1 La Vista tendrá una imagen de fondo (Imagen_FondoEjercicios).</p> <p>2.2 Se muestra un texto que describe los ejercicios (TextoEjercicios) y un texto de confirmación de la validez de los ejercicios una vez realizados y enviados (TextoRespuesta).</p> <p>2.3 Se muestran 6 botones de texto que conducen a los diferentes cuestionarios (conformados por varios ejercicios) de acuerdo a los temas de contenido. Se muestra además un botón "EnviarEjercicio", ubicado al pie de los botones de textos, un botón "CerrarEjercicios" que permite salir de la Vista, ubicado en la parte superior derecha y un botón "Siguiete" y "Anterior" (representados con flechas orientadas a la derecha y a la izquierda respectivamente y ubicados en ambos extremos del botón "EnviarEjercicio")</p> <p>2.4 Para los ejercicios se utilizarán los componentes checkbox, combobox y radiobutton.</p>

3. El usuario selecciona uno de los botones de textos presentes en la vista.	4. Se carga el cuestionario de acuerdo al tema seleccionado por el usuario.	4.1 Se muestra el primer ejercicio del cuestionario.	
5. El usuario selecciona la opción "EnviarEjercicio".	6. Se muestra un texto con la calificación del ejercicio enviado.	6.1 El texto se muestra en un MovieClip en el centro de la pantalla.	
7. El usuario selecciona la opción "Siguiente".	8. Se muestra el ejercicio siguiente del cuestionario.		
9. El usuario selecciona la opción "Anterior".	10. Se muestra el ejercicio anterior del cuestionario.		
11. El usuario selecciona la opción "CerrarEjercicios".	12. Finaliza la vista de presentación.		
Curso alternativo de los eventos			
Acción	Curso alternativo		
Prioridad	Secundaria		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_FondoEjercicios	Imagen de color negro, la parte superior de la imagen es una barra de color carmelita con un icono que muestra una hoja con lápiz en la parte izquierda.	Existente
Video o Animación	-	-	-
Sonido	-	-	-
Texto	- TextoEjercicios - TextoRespuesta	Muestra los diferentes Ejercicios contenidos en la Vista. El color de la letra es blanco, el texto está justificado, la fuente es Verdana con un tamaño igual a 11. Muestra un mensaje de confirmación de validez del ejercicio enviado. El color del texto es	Existente

		carmelita oscuro.	
Reglas Pedagógicas			
-			

Tabla 8: Descripción textual de la Vista Galería de Videos

Descripción textual de la Vista de Presentación “ <i>GaleríaVideos</i> ”		
Actores de la Vista de Presentación	Usuario	
Propósito	Mostrar al usuario diferentes videos que servirán de apoyo para la comprensión del contenido.	
Objetivos Pedagógicos	Facilitar el proceso de aprendizaje a través de videos.	
Resumen	La vista se inicia cuando el usuario accede a la opción “Galería de Videos” a través de la vista “ <i>Menú_Derecho</i> ”.	
Vistas asociadas	Vista “ <i>Menú_Derecho</i> ”	
Referencias	R1.6, R3.3	
Precondiciones	El usuario debe haber seleccionado la opción “Galería de Videos” contenida en la vista “ <i>Menú_Derecho</i> ”.	
Poscondiciones	Vista “ <i>GaleríaVideos</i> ” visualizada.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. El usuario acciona el botón “Galería de Videos” contenido en la vista “ <i>Menú_Derecho</i> ”.	2. Se carga la Vista “ <i>GaleríaVideos</i> ”.	2.1 La vista tendrá una imagen de fondo (Imagen_GaleríaVideos). 2.2 Se muestran 3 botones de tipo imagen, que permiten cargar los videos, ubicados en la parte derecha de la Vista. Se muestra además un botón “Anterior” y un botón “Siguiete” (representados por flechas orientadas

		<p>hacia arriba y abajo respectivamente), estos botones están ubicados en el extremo superior e inferior de los botones imágenes. Se muestra también un botón “CerrarGaleríaVideos” que permite salir de la Vista.</p> <p>2.3 Se muestra un reproductor para la visualización de los videos.</p>
3 El usuario selecciona uno de los botones de tipo imagen ubicados en la Vista.	4 Se carga el video correspondiente en el reproductor de acuerdo a la elección del usuario.	
5 El usuario selecciona el botón “Siguiente”.	6 Se actualizan los botones de imágenes mostrados.	6.1 El contenido de los botones se desplaza hacia arriba, permitiendo que el tercer botón permita la visualización del video próximo.
7 El usuario selecciona el botón “Anterior”.	8 Se actualizan los botones de imágenes mostrados.	8.1 El contenido de los botones se desplaza hacia abajo, permitiendo que el primer botón permita la visualización del video anterior.
9 El usuario selecciona la opción “CerrarGaleríaVideos”.	10 Finaliza la Vista de Presentación.	
Curso alterno de los eventos		
Acción	Curso alterno	
Prioridad	Secundaria	
Mejoras	-	
Medias a utilizar		

Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_GaleríaVideos	Imagen de color negro, la parte superior de la imagen es una barra de color carmelita con un icono que muestra una cinta de video en la parte izquierda.	Existente
Video o Animación	Video	Videos relacionados con los temas abordados en la aplicación.	Existente
Sonido	-	-	-
Texto	-	-	-
Reglas Pedagógicas			
-			

Tabla 9: Descripción textual de la Vista Glosario

Descripción textual de la Vista de Presentación “Glosario”	
Actores de la Vista de Presentación	Usuario
Propósito	Mostrar al usuario las palabras de difícil comprensión relacionadas con los temas de la aplicación.
Objetivos Pedagógicos	Mostrar al usuario el significado de algunas palabras que pueden resultar de difícil comprensión relacionadas con el tema.
Resumen	La vista se inicia cuando el usuario accede a la opción “Glosario” a través de la vista “Menú_Derecho”.
Vistas asociadas	Vista “Menú_Derecho”
Referencias	R1.7
Precondiciones	El usuario debe haber seleccionado la opción “Glosario” contenida en la vista “Menú_Derecho”.
Poscondiciones	Vista “Glosario” visualizada.
Curso normal de eventos	

Acciones del actor	Respuesta del sistema	Elementos de la vista	
1. El usuario acciona el botón "Glosario" contenido en la vista "Menú_Derecho".	2. Se carga la Vista "Glosario".	2.1 La vista tendrá una imagen de fondo (Imagen_FondoGlosario). 2.2 Se muestra un texto que describe las palabras que comienzan con una misma letra. 2.3 Se muestran 27 botones de texto (letras del abecedario) que muestran de acuerdo a la letra las diferentes palabras del "Glosario" y un botón "CerrarGlosario".	
3 El usuario acciona uno de los botones de texto contenidos en la Vista.	4 Se actualiza el texto del glosario.	4.1 Se muestran en el texto del glosario las palabras que comiencen con la letra seleccionada por el usuario.	
5 Selecciona la opción "CerrarGlosario".	6 Finaliza la Vista de Presentación.		
Curso alternativo de los eventos			
Acción	Curso alternativo		
Prioridad	Secundaria		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_FondoGlosario	Imagen de color gris, la parte superior de la imagen es una barra de color carmelita con un icono que muestra libro en la parte izquierda.	Existente
Video o Animación	-	-	-
Sonido	-	-	-

Texto	TextoGlosario	Texto que describe cada una de las palabras de difícil comprensión. El texto es de color negro, justificado, con fuente Verdana y tamaño igual a 11.	Existente
Reglas Pedagógicas			
-			

Tabla 10: Descripción textual de la Vista Servicios

Descripción textual de la Vista de Presentación “Servicios”		
Actores de la Vista de Presentación	Usuario, Vista “Principal”	
Propósito	Permitir al usuario acceder a algunos servicios que brinda la aplicación (cerrar la aplicación, música, ayuda, volver a la pantalla de presentación).	
Objetivos Pedagógicos	-	
Resumen	La vista se inicia cuando se carga la vista “Principal” y estará presente en todas las vistas de la aplicación. Culmina cuando se cierra la aplicación.	
Vistas asociadas	Vista “Principal”, Vista “Ayuda”, Vista “Salir”	
Referencias	R2.1, R3.1, R3.2, R3.5	
Precondiciones	Debe haberse cargado la vista “Principal”.	
Poscondiciones	Vista “Servicios” visualizada.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
	1. Se ejecuta la Vista “Servicios”.	1.1 La vista tendrá cuatro botones de imagen estos serán, “Cerrar”, “Ayuda”, “Música”, e “Inicio”. Los mismos están ubicado en la parte superior derecha de la Vista.

2. El usuario selecciona la opción "Ayuda".	3. Se muestra la Vista "Ayuda" de la aplicación.		
4. El usuario selecciona la opción "Inicio".	5. Se muestra la Vista "Principal" de la aplicación.		
6. El usuario selecciona la opción "Música".	7. Se detiene la música de la aplicación.		
8. El usuario selecciona la opción "Cerrar".	9. Se muestra la Vista "Salir".		
Curso alternativo de los eventos			
Acción	Curso alternativo		
6	Si la música de la aplicación está detenida se reproduce.		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	-	-	-
Video o Animación	-	-	-
Sonido	-	-	-
Texto	-	-	-
Reglas Pedagógicas			
-			

Tabla 11: Descripción textual de la Vista Salir

Descripción textual de la Vista de Presentación "Salir"	
Actores de la Vista de Presentación	Usuario
Propósito	Brindar al usuario la posibilidad de confirmar si desea salir realmente de la aplicación o no.
Objetivos Pedagógicos	-

Resumen		La vista se inicia cuando el usuario accede a la opción "Cerrar" de la Vista "Servicios". El sistema muestra las opciones de confirmación y culmina cuando el usuario selecciona una de estas opciones.	
Vistas asociadas		Vista "Servicios", Vista "Créditos"	
Referencias		R3.5.1	
Precondiciones		El usuario debe haber seleccionado la opción "Cerrar" contenida en la vista "Servicios".	
Poscondiciones		Vista "Salir" visualizada.	
Curso normal de eventos			
Acciones del actor	Respuesta del sistema	Elementos de la vista	
1. El usuario acciona el botón "Cerrar" contenido en la vista "Servicios".	2. Se carga la Vista "Salir".	2.1 La vista tendrá una imagen de fondo (Imagen_FondoSalir), ubicada en el centro de la pantalla. 2.2 Se muestran dos botones de texto: "Sí" y "No", ubicados en el centro de la vista, uno al lado del otro.	
3 El usuario selecciona el botón de texto "No".	4 Se cierra la Vista "Salir" mostrándose nuevamente la Vista anterior.		
5 El usuario selecciona el botón de texto "Sí".	6 Se muestra la Vista "Créditos" y culmina la Vista "Salir".		
Curso alternativo de los eventos			
Acción	Curso alternativo		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	Imagen_FondoSalir	Imagen de color negro, de forma rectangular con las esquinas redondeadas.	Existente

Video o Animación	-	-	-
Sonido	-	-	-
Texto	-	-	-
Reglas Pedagógicas			
-			

Tabla 12: Descripción textual de la Vista Créditos

Descripción textual de la Vista de Presentación “Créditos”		
Actores de la Vista de Presentación	Usuario	
Propósito	Muestra al usuario los nombres de los autores de la aplicación, el nombre de la universidad y los agradecimientos.	
Objetivos Pedagógicos	-	
Resumen	La vista se inicia cuando el usuario selecciona la opción “Si” de la Vista “Salir” y culmina cuando el usuario presiona una tecla ó cuando finaliza el tiempo de reproducción de los créditos.	
Vistas asociadas	Vista “Salir”	
Referencias	R3.5.2	
Precondiciones	El usuario debe haber seleccionado la opción “Si” contenida en la Vista “Salir”.	
Poscondiciones	Vista “Créditos” visualizada.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. El usuario selecciona la opción “Si” de la Vista “Salir”.	2. Se carga la Vista “Créditos”, mostrándose una animación (AnimaciónCréditos). Al concluir la animación de los créditos finaliza la ejecución de la aplicación	2.1 La animación se muestra en el centro de la pantalla.
Curso alternativo de los eventos		
Acción	Curso alternativo	
2	Si presiona una tecla culmina los créditos y la aplicación.	
Prioridad	Crítica	

Mejoras		-	
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	-	-	-
Video o Animación	AnimaciónCréditos	La animación mostrará los nombres de los autores de la aplicación, el nombre de la universidad y los agradecimientos.	Existente
Sonido	-	-	-
Texto	-	-	-
Reglas Pedagógicas			
-			

Glosario de Términos

AIFF: Audio Interchange File Format (AIFF ó Formato de Archivo de Intercambio de Audio) es un estándar de formato de audio usado para vender datos de sonido para computadoras personales.

Animación: Es la presentación de un número de gráficos por segundo que genera en el observador la sensación de movimiento. La animación es el arte visual en el que se representa movimiento. En un sentido amplio, incluye todos los cambios, incluidos los de posición, forma, color y otras características de los objetos.

Avi: Audio/Video Interlaved. Audio/Video Entrepaginado. Formato tradicional de archivos de video digital creado por Microsoft para las plataformas PC compatible.

Bmp: Bitmap. Mapa de bits. Formato tradicional de archivos de imágenes digitales creado por Microsoft para ser utilizado por el sistema operativo Windows.

Framework: Es una estructura de soporte definida mediante la cual un proyecto de software puede ser organizado y desarrollado.

Gif: Graphics Interchange Format. Formato de Intercambio de Gráficos. Formato de archivos de imágenes digitales muy utilizado en la Web por ser de reducidas dimensiones.

Gráficos Vectoriales: Es una imagen digital formada por objetos geométricos independientes (segmentos, polígonos, arcos, etc.), cada uno de ellos definido por distintos atributos matemáticos de forma, de posición, de color, etc.

Herramienta de Autor: Software que manejan elementos de media asociados a la programación para lograr la funcionalidad interactiva de un producto multimedia.

HTML: Lenguaje de marcas hipertextuales.

JPG: Extensión que identifica a los archivos con formato de archivo digital según estándares del JPEG.

Lenguaje de Marcas: Un lenguaje de marcado o lenguaje de marcas es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

MPEG: Moving Picture Experts Group (Grupo de Expertos de Imágenes en Movimiento) referido comúnmente como MPEG, es un grupo de trabajo encargado de desarrollar estándares de codificación de audio y vídeo.

MP3: Formato de archivos de audio digital que utiliza uno de los estándares propuestos por MPEG. Actualmente muy famoso por las altas tasas de compresión que proporciona archivos de muy reducido tamaño y excelente calidad.

Nodo: Recurso de ejecución tal como un computador, un dispositivo o memoria.

Píxeles: Son cada uno de los puntos que forman las imágenes. Cada píxel, tiene un color diferente, de forma que al observar el conjunto de puntos se forma la imagen. El número de puntos que forma la imagen indica su resolución.

TICs: Tecnologías de la Información y las Comunicaciones. Se denominan así al conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética. Las TICs incluyen la electrónica como tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual.

XML: De las siglas en inglés Extensible Markup Language (lenguaje de marcas ampliable), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).