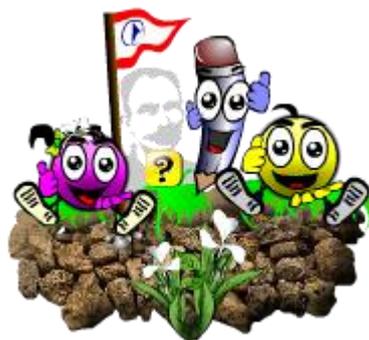


Universidad de las Ciencias Informáticas  
**Facultad 6**



**Título: "SOFTWARE EDUCATIVO PICO MESTRO"**



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

**Autor:** Norlen Rosales Velazquez

**Tutores:** Ing. Reynaldo Alvarez Luna  
Ing. Yosnel Herrera Martínez  
Lic. Yanelis Benítez Fernández

**Consultante:** MSc. Pedro Martínez Piñón

Ciudad de La Habana, junio del 2009

*El futuro de una sociedad depende en gran medida de la  
felicidad de sus niños.*

*Norlen Rosales*

## **DECLARACIÓN DE AUTORÍA**

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

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

---

Autor: Norlen Rosales Velazquez

---

Tutor: Ing. Reynaldo Alvarez Luna

---

Tutor: Ing. Yosnel Herrera Martínez

---

Tutor: Lic. Yanelis Benítez Fernández

## **DATOS DE CONTACTO**

### **Tutores:**

Ing. Reynaldo Alvarez Luna

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba.

Email: [rluna@uci.cu](mailto:rluna@uci.cu)

Ing. Yosnel Herrera Martínez

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba.

Email: [yherrera@uci.cu](mailto:yherrera@uci.cu)

Lic. Yanelis Benítez Fernández

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba.

Email: [jhanelis@uci.cu](mailto:jhanelis@uci.cu)

### **Consultante:**

MSc. Pedro Martínez Piñón

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba.

Email: [pedrito@uci.cu](mailto:pedrito@uci.cu)

## **AGRADECIMIENTOS**

A mi papá Románico Rosales Campo por ser mi guía, amigo y hermano a la vez. Papi a ti debo todo.

A mi mamá Belkis Velazquez Pardo por su cariño y por confiar siempre en mí. Mimi te quiero mucho.

A toda la gente linda que siempre me brindó su apoyo.

A la UCI y al Flash.

## **DEDICATORIA**

A mis hermanitos Jorgito y Noydel, ustedes fueron mi inspiración.

A mis padres y a toda mi familia.

A todos los niños de Cuba.

## **RESUMEN**

La introducción de las computadoras en las escuelas primarias cubanas ha posibilitado el uso de un nuevo medio de enseñanza: el Software Educativo, el que ha despertado en los pioneros mucho interés por aprender. Actualmente para desarrollar este tipo de aplicaciones informáticas destinadas a los niños se tiende a incorporar la actividad lúdica, debido a que los juegos potencian su proceso de aprendizaje. Pero hoy día en las escuelas primarias cubanas no se dispone de un software educativo que mediante la actividad lúdica contribuya con el estudio de la vida y obra martianas.

El presente trabajo aborda el desarrollo de un software educativo con tecnología multimedia que contribuye con el estudio de la vida y obra de José Martí en niños de entre 7 y 12 años de edad mediante la actividad lúdica. Se trata de un juego de aventuras en que el niño deberá escalar el Pico Turquino para rendir tributo al busto de Martí que se encuentra en su cima. Por el camino tendrá que ir acumulando puntos al responder determinadas incógnitas relacionadas con la vida y obra del apóstol, además de esquivar obstáculos que interrumpirán su paso.

Se hace un estudio de las tendencias actuales para el desarrollo de este tipo de aplicaciones. Se realiza un análisis y selección de las tecnologías adecuadas para el desarrollo del software, se identifican las funcionalidades que debe cumplir este, y se realiza el diseño del mismo. A partir de los artefactos obtenidos se procede a su implementación.

## **PALABRAS CLAVE**

Software educativo, juego educativo, multimedia, José Martí, Pico Maestro.

# ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.....	5
1.1 El Software Educativo .....	5
1.1.1 Actualidad del Software Educativo .....	5
1.1.2 Clasificación del Software Educativo .....	7
1.2 Los Juegos Educativos .....	9
1.2.1 Los Juegos Educativos con tecnología Multimedia.....	9
1.2.2 Ventajas y desventajas.....	10
1.3 Metodologías de desarrollo .....	11
1.3.1 XP .....	12
1.3.2 FDD.....	13
1.3.3 RUP.....	14
1.3.4 Metodología seleccionada .....	15
1.4 Arquitectura y patrones .....	16
1.4.1 Patrón arquitectónico Modelo Vista Controlador.....	16
1.4.2 Patrón arquitectónico Modelo Vista Controlador Entidad.....	18
1.4.3 Patrones de diseño.....	19
1.5 Lenguajes de modelado .....	20
1.5.1 UML.....	20
1.5.2 OMMMA-L .....	22
1.5.3 ApEM-L .....	23
1.5.4 Lenguaje seleccionado.....	24
1.6 Herramientas de modelado .....	25
1.6.1 Rational Rose.....	25
1.6.2 Visual Paradigm .....	27
1.6.3 Herramienta seleccionada .....	27

1.7 Herramientas de desarrollo de aplicaciones con tecnología Multimedia.....	28
1.7.1 ToolBook.....	28
1.7.2 Macromedia Director MX 2004.....	29
1.7.3 Macromedia Flash 8.....	30
1.7.4 Herramienta seleccionada.....	30
1.8 Conclusiones del capítulo.....	32
CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	33
2.1 La audiencia.....	33
2.2 Solución propuesta.....	33
2.3 Modelo del Dominio.....	35
2.3.1 Conceptos del Dominio.....	35
2.3.2 Diagrama de Clases del Modelo del Dominio.....	36
2.4 Descripción de la funcionalidad.....	37
2.4.1 Requerimientos Funcionales.....	37
2.4.2 Requerimientos No Funcionales.....	38
2.5 Vista de Gestión del Modelo y sus Subsistemas.....	40
2.5.1 Vistas de Presentación.....	42
2.5.2 Descripción Textual de las Vistas de Presentación.....	42
2.5.3 Diagramas de Estructura de Presentación.....	52
2.5.4 Diagramas de Estructura de Navegación.....	54
2.6 Conclusiones del capítulo.....	59
CAPÍTULO 3. DISEÑO DE LA SOLUCIÓN PROPUESTA.....	60
3.1 Patrón arquitectónico y patrones de diseño utilizados.....	60
3.2 Modelo del Diseño.....	61

3.2.1	Diagramas de Clases .....	61
3.2.2	Diagramas de Secuencias.....	63
3.2.3	Diagrama de Despliegue .....	66
3.3	Conclusiones del capítulo .....	66
CAPÍTULO 4. IMPLEMENTACIÓN DE LA SOLUCIÓN PROPUESTA.....		67
4.1	Modelo de Implementación .....	67
4.1.1	Diagramas de Componentes .....	67
4.2	Descripción de archivos XML .....	69
4.2.1	Archivo XML Pregunta.....	70
4.2.2	Archivo XML CantidadTotalPreguntas .....	70
4.3	Pantallas Principales .....	71
4.4	Conclusiones del capítulo .....	74
CONCLUSIONES .....		75
RECOMENDACIONES .....		76
REFERENCIAS BIBLIOGRÁFICAS .....		77
BIBLIOGRAFÍA.....		79
ANEXOS.....		81
GLOSARIO .....		150

## INTRODUCCIÓN

En marco del creciente proceso de informatización de la sociedad en Cuba como parte de la Batalla de Ideas, se hace eminente el uso de las Tecnologías de la Información y la Comunicación (TIC) en el campo de la Educación. En consecuencia, esta última se encuentra inmersa en un proceso de transformaciones en las que los medios informáticos desempeñan un papel fundamental.

Una gran cantidad de computadoras han sido instaladas en escuelas y Joven Club de Computación de todo el país. Las localidades más intrincadas también se han favorecido con este programa de informatización con la instalación de celdas fotovoltaicas que han posibilitado el uso de estos dispositivos hasta en lo más alto una montaña donde existe una escuela rural. Varios Joven Club móviles se han creado para llevar la computación a estas zonas. Gracias a esto los niños cubanos pueden acceder al mundo cibernético, donde se divierten y aún más importante: aprenden.

A pesar de que el poder comunicativo de la informática contemporánea constituye una gran ventaja, también hay desventajas en el uso de la misma como es la amenaza a la pérdida de la identidad por la influencia foránea. Los niños son los principales amenazados debido al constante consumo de productos extranjeros como son los videojuegos.

La mejor alternativa para resolver tal problema es convertir a nuestro país de simple consumidor en productor. Por lo que cobra marcada importancia el desarrollo de software autóctonos que tributen al desarrollo de la enseñanza, la cultura y la formación de valores en la sociedad cubana como son los software educativos, medio que influye eficientemente en la esfera sensorial del individuo.

La tendencia actual para el desarrollo de este tipo de aplicaciones destinadas a los niños es incorporar la actividad lúdica. Los juegos permiten a los niños explorar e imitar el mundo que les rodea, con ellos ponen en marcha los mecanismos de su imaginación y desarrollan su creatividad. El hecho de posibilitar convertirse en partícipes y sobre todo en protagonistas de sus propias aventuras, como los juegos por computadoras, se asocia mucho con la diversión. Los expertos coinciden en que este elemento potencia su desarrollo cognitivo (intelectual) y afectivo (emocional). Es durante el proceso del juego cuando los niños ponen en práctica los conocimientos adquiridos. (GALLARDO 2008)

Debido a la potencialidad que ofrece este tipo de producto como instrumento pedagógico, durante los últimos años se han venido utilizando mucho en las escuelas primarias de nuestro país. Ha sido evidente el impacto causado por su uso en las diferentes asignaturas que se imparten en estos centros

educacionales. Lo que constituye la base de muchas de las transformaciones que en materia educacional se ha venido efectuando durante los últimos años en la esfera de la educación.

Como consecuencia del propósito descrito, el gobierno cubano ejecutó el proyecto de desarrollo de software educativo para las escuelas cubanas como un programa más de los numerosos priorizados por el Estado, entre los meses finales del año 2001 y el año 2002, consistiendo en el desarrollo de 30 paquetes de software para las escuelas primarias con temas que abarcaron todos los contenidos curriculares y extracurriculares de los educandos de 6 a 11 años de edad, temas dentro de los que no podía faltar José Martí.

El estudio de la obra martiana en edades tempranas, constituye un aspecto de vital importancia dentro del desarrollo espiritual y moral de un niño. El legado de la obra martiana fundamenta, orienta e inspira las posiciones más progresistas de todos los cubanos. Su figura se alza hoy como arquetipo ético: sus ideas, sirven de pedestal a los principios de la Revolución Cubana; de ahí que resulte de trascendental alcance su estudio para las nuevas generaciones, donde pueden encontrarse valores imperecederos como la honestidad, la honradez, la solidaridad, el patriotismo, importantísimos en la educación de la personalidad de niños, adolescentes y jóvenes. (VISPO 2006)

Por tal razón en todas las escuelas primarias de Cuba se estudia la vida y obra de José Martí. En este sentido se utilizan medios de enseñanza tradicionales como libros, cuadernos martianos, láminas y murales martianos. La introducción de las TIC en estos centros educacionales ha posibilitado el uso de otros medios de enseñanza como los materiales audiovisuales y el software educativo, este último ha despertado en los pioneros mucho interés por aprender.

Actualmente en las escuelas primarias de nuestro país se dispone de la colección de software educativos: Multisaber. La cual dedica una de sus secciones al estudio de la vida y obra de José Martí, a la que titula "El más puro de nuestra raza". Cabe señalar que dicho software no incorpora como elemento fundamental la actividad lúdica.

Tras una entrevista realizada al Ing. Abel Ernesto Lorente Rodríguez, especialista en la Dirección de Producción #2 de la Universidad de las Ciencias Informáticas (UCI) y a la compañera Karenia Álvarez especialista de la Dirección Nacional de Informática Educativa del Ministerio de Educación, se pudo comprobar que hoy día en las escuelas primarias cubanas no se dispone de un software educativo que vincule la actividad lúdica de los niños con el estudio de la vida y obra de José Martí.

De esta problemática se deriva el **problema científico** de esta investigación: ¿Cómo contribuir al estudio de la vida y obra de José Martí mediante la actividad lúdica?

Por lo que se ha definido como **objeto de estudio**: El Proceso de desarrollo de software educativo. Y como **campo de acción** se identificó: El Proceso de desarrollo de software educativo con tecnología multimedia.

Se propone como **objetivo general** de esta investigación: desarrollar un software educativo con tecnología multimedia que contribuya al estudio de la vida y obra de José Martí en niños de entre 7 y 12 años de edad, mediante la actividad lúdica.

Para dar cumplimiento a este objetivo general se plantea como **objetivos específicos**:

- Definir las funcionalidades que deberá cumplir el software.
- Diseñar el software orientado a la tecnología multimedia.
- Implementar el software orientado a la tecnología multimedia.

A los que se le dará cumplimiento mediante las siguientes **tareas de la investigación**:

- Análisis del estado del arte del software educativo.
- Investigación de las tendencias actuales del software educativo con tecnología multimedia.
- Estudio y selección de la metodología de desarrollo.
- Estudio y selección de los lenguajes de modelado y de programación a utilizar.
- Estudio y selección de las herramientas para desarrollar el software.
- Modelación del negocio.
- Especificación de los requerimientos funcionales y no funcionales del software.
- Selección de los patrones de diseño.
- Elaboración de los diagramas de clases del diseño.
- Elaboración de los diagramas de interacción.
- Elaboración de los diagramas de componentes.
- Implementación del software.

**Aportes prácticos esperados de la investigación:**

El resultado de la presente investigación será un software educativo con tecnología multimedia que mediante la actividad lúdica contribuya al estudio la vida y obra de José Martí en niños de entre 7 y 12 años de edad.

El contenido de este documento está estructurado de la siguiente manera:

**Capítulo 1:** *Fundamentación del tema:* Explica el estado del arte del software educativo, las metodologías, las herramientas y los lenguajes utilizados para el desarrollo de la aplicación.

**Capítulo 2:** *Descripción de la solución propuesta:* Muestra el modelado del negocio a través de un Modelo del Dominio. Muestra los requerimientos funcionales y no funcionales de la aplicación, así como los diagramas correspondientes a la gestión del modelo y sus subsistemas.

**Capítulo 3:** *Diseño de la Solución Propuesta:* Incluye la descripción de los estilos arquitectónicos y patrones utilizados. Define el modelo de diseño del sistema. Muestra los diagramas de clases, de secuencias y el modelo de despliegue.

**Capítulo 4:** *Implementación de la Solución Propuesta:* Incluye la definición del modelo de implementación. Muestra los diagramas de componentes, describe la estructura de la información almacenada y muestra las pantallas principales de la aplicación.

# CAPÍTULO 1 . FUNDAMENTACIÓN DEL TEMA

En este capítulo se hará un estudio del estado actual del software educativo en Cuba y en el mundo. Se hará énfasis en uso de la tecnología multimedia para este tipo de software. Dentro de las clasificaciones de los software educativos se encuentran los juegos educativos, en los que se hará énfasis, señalando las ventajas y desventajas de su uso como medio de enseñanza. Se analizará el estado actual de la tecnología para el desarrollo de aplicaciones multimedia, estudiando diferentes estilos arquitectónicos, patrones, metodologías, lenguajes y herramientas para el desarrollo de software educativo con tecnología multimedia.

## 1.1 EL SOFTWARE EDUCATIVO

Según (SÁNCHEZ 1999) el Software Educativo se define 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 aquel material de aprendizaje especialmente diseñado para ser utilizado con una computadora en los procesos de enseñar y aprender.

Otros autores lo definen como una aplicación informática, que soportada sobre una bien definida estrategia pedagógica, apoya directamente el proceso de enseñanza aprendizaje constituyendo un efectivo instrumento para el desarrollo educacional.

Se puede concluir que un Software Educativo es un producto informático diseñado con un fin pedagógico, que mediante actividades atractivas para el alumno puede facilitar la consecución de determinados objetivos educativos, favoreciendo en alto grado al proceso de enseñanza - aprendizaje.

### 1.1.1 ACTUALIDAD DEL SOFTWARE EDUCATIVO

A nivel mundial y desde hace varias décadas se discute bastante acerca de cómo evaluar los diferentes software desarrollados, con vistas a ser utilizados como apoyo al proceso de enseñanza - aprendizaje, existe dos tendencias fundamentales: (BOTANA 2007)

La primera plantea desarrollar la evaluación como un proceso integral en forma de sistema, con el fin de que abarque todo el ciclo de vida (diseño, desarrollo y explotación) del software educativo. Esta debe permitir constatar el resultado obtenido con el que se previó, teniendo en cuenta los contextos en

los cuales van a ser utilizados y las características de los estudiantes y docentes que interaccionan con los mismos, donde es importante tener presentes aspectos técnicos, estéticos y pedagógicos y una perspectiva de evaluación colaborativa entre las diferentes personas que intervienen en el proceso de diseño, producción y utilización del mismo.

La segunda plantea evaluar el software educativo como un producto ya terminado, para seleccionar aquel que cumpla con los objetivos del proceso de enseñanza aprendizaje. Esta permite orientar el uso del software por los docentes, dándose énfasis, por tanto, a las potencialidades pedagógicas y metodológicas más que a los aspectos informáticos o técnicos, para definir si el software puede ser usado y cómo usarlo en el desarrollo del proceso docente educativo.

Indistintamente, cada una de estas tendencias ha generado metodologías, criterios, modelos, planillas, guías de uso, instrumentos, sobre los que se puede encontrar diversas escalas evaluativas. La mayoría de las propuestas a nivel mundial se enmarcan en la segunda tendencia ya que a diferencia de nuestro país, el software educativo llega a las escuelas usualmente por la vía comercial o individualmente el docente lo obtiene de Internet.

Cuba enfrenta actualmente el desafío de las TIC, lo que requiere de cambios cualitativos profundos en el sistema educativo cubano para continuar elevando la calidad de la educación para todos. El uso del software educativo como medio de enseñanza constituye un elemento de vital importancia para el logro de los objetivos planteados en las transformaciones de la escuela cubana y en un período relativamente corto el MINED ha organizado el proceso de producción del software e introducido los mismos en todos los niveles de educación.

En nuestro país, es meritoria la labor realizada por los centros de estudio en la creación de las colecciones *El Navegante*, para Secundaria Básica, y *Futuro*, para la educación Pre-universitaria y Politécnica, donde ha existido un trabajo colegiado de diferentes especialistas según corresponde en las diferentes etapas del ciclo de vida del software educativo.

Por otro lado, el uso del software educativo en nuestras escuelas es aún joven y en el empeño de lograr un desarrollo en este sentido, el MINED ha establecido cuáles son los software que deben usarse por niveles de educación, facilitando materiales para la orientación a los maestros y profesores sobre el uso de los mismos en el proceso de enseñanza - aprendizaje. En este bregar se han propuesto, basados en la práctica pedagógica, varios trabajos para la superación de los estudiantes en el uso de los software educativos y las tareas docentes utilizando los mismos, que el maestro ha aplicado en mayor o menor medida, aunque no siempre ha resultado una feliz experiencia.

La Universidad de las Ciencias Informáticas según expresó Fidel Castro el referirse a esta, debe ser una universidad de constantes cambios pedagógicos; en la cual la investigación y la experimentación de nuevos conceptos, técnicas, métodos, procedimientos y contenidos sean lo cotidiano.

La UCI, al ser hoy un centro con solo 7 años de creado, consta con un personal especializado de pocos años de experiencia en su mayoría, inmerso en el impulso de la Industria Cubana del Software, aplicando las mejores prácticas de la ingeniería informática en su totalidad e investigando en aspectos relacionados con la generalización de soluciones y adecuación de estas al contexto productivo UCI. (FEBE 2007)

A pesar del poco tiempo que lleva de creada esta universidad, cuenta con una notación para representar los elementos estructurales, lógicos, funcionales, pedagógicos y de patrones de ingeniería al modelar los software educativos, permitiendo una mejor documentación técnica y una más clara representación ingenieril en este tipo de aplicaciones.

### 1.1.2 CLASIFICACIÓN DEL SOFTWARE EDUCATIVO

Los software educativos a pesar de tener rasgos esenciales básicos y una estructura general común se presentan con características muy diversas: algunos aparentan ser un laboratorio o una biblioteca, otros se limitan a ofrecer función instrumental del tipo máquina de escribir o calculadora, otros se presentan como juego o como libro, muchos tienen vocación de examen, unos pocos de expertos.

A continuación se muestran las clasificaciones del software educativo según la función educativa que asumen dentro del proceso de enseñanza - aprendizaje: (CURBELO *et al.* 2007)

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

Considerando la función educativa se pueden clasificar en:

#### **Sistemas Tutoriales:**

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

**Sistemas Entrenadores:**

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

**Libros Electrónicos:**

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

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

Considerando la función educativa se pueden clasificar en:

**Simuladores**

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

**Sistemas Expertos**

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

**Sistemas Tutoriales Inteligentes de enseñanza**

Pueden detectar errores, clasificarlos, y explicar por qué se producen, favoreciendo así el proceso de retroalimentación del estudiante.

**Juegos Educativos**

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

De las anteriores clasificaciones del software educativo, en la presente investigación se hará énfasis en esta última.

## 1.2 LOS JUEGOS EDUCATIVOS

El concepto de Juego Educativo involucra un producto digital que incorpora una aventura en un entorno audiovisual donde el usuario debe resolver ciertos problemas que conforman la esencia del juego en cuestión en pro de su desarrollo. Se crea entonces una reciprocidad entre el ámbito educativo y el recreacional que fortalece aún más la accesibilidad constante del usuario al conocimiento y a su formación integral.

Los Juegos Educativos representan una herramienta pedagógica de muy alto valor ya que con ellos el usuario puede aplicar los conocimientos adquiridos en el aula o en el juego como instrumento para llegar hasta la victoria. Esto aumenta en gran medida el rendimiento del estudiante pues este se verá mucho más interesado en captar la mayor cantidad de conocimientos que le sea posible.

Los Juegos Educativos estimulan el desarrollo de habilidades en los estudiantes como la coordinación motriz y los reflejos, la capacidad observación, de concentración, la creatividad, la imaginación y la memoria.

### 1.2.1 LOS JUEGOS EDUCATIVOS CON TECNOLOGÍA MULTIMEDIA

Se puede llamar Multimedia a un producto informático capaz de hacer uso simultáneo de diferentes tipos de información, como audio, video, animaciones, texto y gráficos en una computadora.

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

La tecnología de multimedia toma auge en los juegos por computadora, a partir de 1992, cuando se integran: audio (música, sonido estéreo y voz), video, gráficas, animación y texto al mismo tiempo. La principal idea de desarrollar multimedia en forma de juego es que se pueda navegar y buscar la información que se desea sobre un tema, sin tener que recorrer todo el programa, que se pueda interactuar con la computadora y que la información no sea lineal sino asociativa.

Los juegos educativos multimedia son una de las mayores atracciones de los niños, los adolescentes y los jóvenes de la última década. La explicación de este fenómeno, se basa principalmente, en que estamos frente a una nueva generación, a la que Marc Prensky, denominó nativos digitales. Sus

características e intereses, los llevan a querer ser partícipes y sobre todo protagonistas de sus propias aventuras, más que meros observadores pasivos de las historias creadas por otros. Este escenario, pone al sistema educativo y al mundo laboral, ante el dilema de incorporar las nuevas tecnologías y en especial los juegos por computadoras al aula, al trabajo y al aprendizaje; enfrentarse a él o acompañar a la nueva generación de los nativos digitales, en la exploración, el análisis crítico, la comprensión y el buen uso de estas herramientas, con el fin de potenciar su proceso de aprendizaje.

### 1.2.2 VENTAJAS Y DESVENTAJAS

A continuación se muestran las principales ventajas y desventajas que trae consigo el uso de juegos educativos con tecnología multimedia según (MARQUÉS 2000) y (HEREDIA 2007):

#### **Ventajas:**

- **Motivación:** La motivación es uno de los motores del aprendizaje, ya que incita a la actividad y al pensamiento. Hace que los estudiantes dediquen más tiempo a trabajar y, por tanto, es probable que aprendan más.
- **Interacción:** Continua actividad intelectual. Los estudiantes están permanentemente activos al interactuar con el ordenador y mantienen un alto grado de implicación en el trabajo.
- **Desarrollo de la iniciativa:** Los estudiantes se ven obligados a tomar continuamente nuevas decisiones ante las respuestas del ordenador a sus acciones.
- **Aprendizaje a partir de los errores:** Los estudiantes pueden conocer sus errores justo en el momento en que se producen y generalmente el software les ofrece la oportunidad de ensayar nuevas respuestas o formas de actuar para superarlos.
- Los alumnos a menudo aprenden con menos tiempo.
- Permiten controlar las tareas docentes de forma individual o colectiva.
- Facilitan la autoevaluación del estudiante.
- Permiten elevar la calidad del proceso docente - educativo.
- Constituyen una nueva, atractiva, dinámica y rica fuente de conocimientos.
- Marca las posibilidades para una nueva clase más desarrolladora.

**Desventajas:**

- Adicción: El software educativo interactivo resulta motivador, pero un exceso de motivación puede provocar adicción.
- Distracción: Los alumnos a veces se dedican a jugar en vez de trabajar.
- Ansiedad: La continua interacción ante el ordenador puede provocar ansiedad en los estudiantes.
- Aprendizajes incompletos y superficiales: La libre interacción de los alumnos con estos materiales (no siempre de calidad) a menudo proporciona aprendizajes incompletos con visiones de la realidad simplista y poco profunda.
- Aislamiento: Los juegos educativos permiten al alumno aprender solo, hasta le animan a hacerlo, pero este trabajo individual, en exceso, puede acarrear problemas de sociabilidad.
- Cansancio visual y otros problemas físicos: Un exceso de tiempo trabajando ante el ordenador o malas posturas pueden provocar diversas dolencias.
- Problemas con los ordenadores: A veces los alumnos desconfiguran o contaminan con virus los ordenadores.

Aun considerando las desventajas que pueda implicar el uso de los juegos educativos con tecnología multimedia en las escuelas primarias cubanas, se imponen los beneficios que pueden aportar este tipo de software en el proceso de enseñanza – aprendizaje de los educandos.

### **1.3 METODOLOGÍAS DE DESARROLLO**

La construcción de un software implica la toma de decisiones sobre la arquitectura del sistema (definir los componentes del sistema de software y sus interacciones). Estas decisiones pueden ser cruciales para el éxito o fracaso del sistema resultante, por lo que se requiere seleccionar un proceso de desarrollo de software con el fin de obtener la calidad del sistema de software deseada y cumpla con los requerimientos establecidos. Metodologías vigentes de ingeniería de software atienden muy bien estos requerimientos y permiten al equipo encargado de dicha labor asumir con propiedad su función. (ANTÓN *et al.* 2002)

### 1.3.1 XP

La metodología de Programación Extrema (XP, eXtreme Programming) es uno de los ejemplos más exitosos de metodología ágil. Se centra en el trabajo orientado directamente al objetivo, basándose para esto en las relaciones interpersonales y en la velocidad de reacción para la implementación y para los cambios que puedan surgir durante el desarrollo del proceso.

Una característica saltante de XP, es que el código siempre se produce en parejas, estas van cambiando constantemente para lograr así que todo el equipo sepa y pueda modificar según necesidades el código generado, esto logra en el equipo que los integrantes aprendan entre sí y compartan todo el código.

Esta metodología consta de cuatro fases:

**Planificación:** Definir las historias de usuario, crear un plan de publicaciones, iteraciones y reuniones bastante frecuentes, estimar la velocidad, XP aconseja la programación en parejas pues incrementa la productividad y la calidad del software desarrollado.

**Diseño:** Se sugieren diseños simples y sencillos, elaborar glosarios de términos, identificar riesgos en pareja y reducirlo al máximo, revisar de nuevo estos códigos para procurar optimizar su funcionamiento.

**Desarrollo o codificación:** El cliente es la parte más importante del equipo de desarrollo, su presencia es indispensable en las distintas fases de XP. La codificación debe hacerse teniendo en cuenta estándares de codificación ya creados. Se sugiere un modelo de trabajo colectivo, usando repositorios de código.

**Pruebas:** Se realizan pruebas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir, o sea obtener los posibles errores.

La metodología XP está muy orientada a la implementación, esto lo hace bueno para el equipo de desarrollo ya que no debe preocuparse de la documentación y los equipos rotan, todos aprenden de todos, por otro lado el cliente también se siente satisfecho pues recibe un software que se adapta exactamente a sus deseos, esto significa que se debió designar a una persona totalmente involucrada en el negocio, lo que podría implicar que esta persona deje de hacer sus funciones para estar totalmente disponible al equipo de desarrollo, razón por la cual se considera mejor la utilización de este proceso para desarrollos internos, pues debe haber una gran confianza entre el cliente y el equipo de

desarrollo, es poco probable que el cliente pueda prescindir de los desarrolladores, esto incurriría un coste adicional para el cliente. Como se puede representar todo lo que se debe sin documentación alguna, si no se anota ni se archiva nada, se hace muy difícil que alguien más pueda tomar el lugar de uno de los miembros del equipo, o hacer mejoras en el sistema, esto crearía una dependencia para con el equipo de desarrollo.

Otro de los inconvenientes que presenta es que solo se recomienda aplicar en proyectos a corto plazo.

### 1.3.2 FDD

La metodología de Desarrollo Manejado por Rasgos (FDD, Feature Driven Development), se considera como punto medio entre las metodologías pesadas y ágiles, aunque en la práctica es más similar a este último. Pensado para proyectos relativamente cortos, está basado en iteraciones que producen un software funcional que puede ser visto, probado y monitorizado por el cliente.

Estas iteraciones son decididas en base a las funcionalidades que el software debe tener, funcionalidades definidas por el cliente.

Esta metodología consta de cinco fases:

1. Desarrollo de un Modelo General.
2. Construcción de la Lista de Funcionalidades.
3. Plan de liberaciones basadas en las funcionalidades a implementar.
4. Diseñar en base a las funcionalidades definidas.
5. Implementar en base a las mismas funcionalidades.

Existen jerarquías en el equipo de trabajo, siempre debe haber un jefe de proyecto, y aunque es un proceso considerado ligero también incluye documentación, la mínima necesaria para que algún nuevo integrante pueda entender el desarrollo de inmediato.

FDD está pensada para proyectos y equipos medianos a pequeños, con la flexibilidad que a mayor necesidad de código, mayor organización.

Pero el mayor problema de FDD está en la necesidad de tener en el equipo miembros con experiencia que marquen el camino a seguir desde el principio, con la elaboración del modelo global, puesto que no es tan ágil como podría serlo XP.

### 1.3.3 RUP

El Proceso Unificado RUP (Rational Unified Process), constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Proporciona una aproximación disciplinada a la asignación de tareas y responsabilidades. RUP actúa como modelo y puede ser adaptado y extendido. (JACOBSON, I. *et al.* 2004)

Características esenciales de RUP:

- **Dirigido por los Casos de Uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, estos representan los requisitos funcionales. Los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- **Centrado en la arquitectura:** La arquitectura muestra una visión común del sistema completo en la que el equipo de trabajo y los usuarios deben de estar de acuerdo, por lo que describe los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
- **Iterativo e incremental:** Propone que cada fase se desarrolle en iteraciones. Una iteración implica actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

La metodología RUP proporciona disciplinas en las cuales se encuentran artefactos con lo cual se podrá contar con guías para poder documentar e implementar de una manera fácil y eficiente, todas las guías para un buen desarrollo, todo esto dentro de las respectivas fases con las cuales cuenta:

- **Concepción o inicio:** Comprender los requisitos y determinar visión y alcance del proyecto.
- **Elaboración:** Asignar recursos, especificar las características y definir la arquitectura.
- **Construcción:** Implementación, construir el producto operacional.
- **Transición:** Hacerlo operativo para los usuarios, nivel correcto de calidad para entregar.

En RUP se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales.

Los seis primeros son los flujos de ingeniería, estos son:

- **Modelamiento de negocio:** se entienden las necesidades del negocio.
- **Requerimientos:** se trasladan las necesidades del negocio a un sistema automatizado.
- **Análisis y diseño:** se trasladan los requerimientos dentro de la arquitectura de software.
- **Implementación:** se crea un software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Prueba (Testeo):** se debe asegurar que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- **Instalación del software.**

Los tres últimos son de apoyo:

- **Administración del proyecto:** para el control de los horarios y recursos.
- **Administración de configuración y cambios:** permite guardar todas las versiones del proyecto.
- **Ambiente:** se realiza la administración del ambiente de desarrollo.

RUP se basa mucho en la documentación, que aunque se ve afectada con los posibles cambios volátiles que los clientes soliciten en cuanto a funcionalidades del software, representa una gran ventaja, ya que gracias a su plan de desarrollo se pueden reconocer los problemas y fallos de forma temprana y corregirlos. Pero además, RUP no es un sistema con pasos firmemente establecidos, sino una metodología adaptable al contexto y necesidades.

#### 1.3.4 METODOLOGÍA SELECCIONADA

Anteriormente se analizaron algunas metodologías como XP, que a pesar de ser muy ágil en el proceso de desarrollo de software debido a que utiliza el mínimo de documentación, tiene como desventaja que el cliente debe de estar fuertemente ligado al equipo de desarrollo, cosa que se hace muy difícil en este caso.

También se analizó la metodología FDD, que puede ser muy ventajosa puesto que aunque también genera documentación, es la mínima necesaria para comprender el código. Pero FDD requiere de miembros experimentado en el equipo de desarrollo, cosa que no siempre puede hacerse realidad.

Finalmente se escoge como metodología de desarrollo a RUP, debido a que es una metodología abierta y adaptable al desarrollo de software educativo, lo cual garantiza que se lleven a cabo sólo aquellas actividades y modelos que sean necesarios o útiles para el proyecto a desarrollar. El hecho de que RUP sea iterativo e incremental, garantiza que en cada iteración se va obteniendo un producto más acabado. La ventaja principal de esta metodología es que se basa en las mejores prácticas que se han intentado y se han probado en el campo del desarrollo de software.

## 1.4 ARQUITECTURA Y PATRONES

En la última década cambió la visión que los desarrolladores tienen de los sistemas de software. Esta nueva visión se llamó "arquitectura". La arquitectura de software básicamente indica la estructura, funcionamiento e interacción entre las partes del software (REYNOSO 2004). Desde los pequeños programas hasta los sistemas más grandes poseen una estructura y un comportamiento que los hace clasificables según su arquitectura.

El diseño arquitectónico define la relación entre los elementos estructurales principales del software, los patrones de diseño que se pueden utilizar para lograr los requisitos que se han definido para el sistema, y las restricciones que afectan a la manera en que se pueden aplicar los patrones de diseño arquitectónicos.

Los patrones expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos; así como ayudan a especificar la estructura fundamental de una aplicación. (FEBE 2006)

### 1.4.1 PATRÓN ARQUITECTÓNICO MODELO VISTA CONTROLADOR

Modelo Vista Controlador (MVC en adelante) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

- **Modelo:** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- **Vista:** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.

- **Controlador:** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón y pulsaciones de teclas. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente: (FEBE 2006)

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón: enlace)
2. El controlador recibe (por parte de los objetos de la interfaz - vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Algunos de sus principales beneficios son:

- Menor acoplamiento.
  - a) Desacopla las vistas de los modelos.
  - b) Desacopla los modelos de la forma en que se muestran e ingresan los datos.
- Mayor cohesión.
  - a) Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio).

- Las vistas proveen mayor flexibilidad y agilidad.
  - a) Se puede crear múltiples vistas de un modelo.
  - b) Las vistas pueden anidarse.
  - c) Se puede cambiar el modo en que una vista responde al usuario sin cambiar su representación visual.
  - d) Se puede sincronizar las vistas.
  - e) Las vistas pueden concentrarse en diferentes aspectos del modelo.
- Mayor facilidad para el desarrollo de clientes ricos en múltiples dispositivos y canales.
  - a) Una vista para cada dispositivo que puede variar según sus capacidades.
  - b) Una vista para la Web y otra para aplicaciones de escritorio.
- Más claridad de diseño.
- Facilita el mantenimiento.
- Mayor escalabilidad.

Desde el surgimiento de MVC y con el paso de los años se han desarrollado diferentes variantes, dentro de las que se encuentra la modificada para Aplicaciones Multimedia del MVC, conocida como MVC<sub>mm</sub>. En esta variante se diversifica las funcionalidades del Modelo teniendo en cuenta las características de las aplicaciones multimedia, donde tienen un gran peso las medias utilizadas en estas.

#### **1.4.2 PATRÓN ARQUITECTÓNICO MODELO VISTA CONTROLADOR ENTIDAD**

Las aplicaciones educativas cubanas explotan grandemente los conceptos del entorno hipermedia, así como incorporan de forma profunda las técnicas y conocimientos de bases de datos relacionales y la Programación Orientada a Objetos más avanzada que se utiliza hoy día; como consecuencia de la necesidad de implementar diversos y complejos métodos pedagógicos desarrollados por nuestros educadores en las distintas ramas de la enseñanza cubana. Es por esto que los diseñadores han buscado y aplicado soluciones ya trabajadas internacionalmente y con un gran número de aplicaciones prácticas en este sentido, dentro de las que se encuentra la utilización del Patrón MVC. No obstante

este patrón tal y como está descrito actualmente, aún con su variante más actual no responde completamente a las características mencionadas.

El patrón Modelo Vista Controlador Entidad (MVC-E) es una variante de solución al MVC<sub>mm</sub> para las aplicaciones educativas cubanas, donde se descarguen las responsabilidades de la clase modelo concernientes al procesamiento y almacenamiento de la información persistente de las aplicaciones, incorporando una nueva clase al modelo denominada *Modelo Entidad*, con dos tipos fundamentales, la clase *Modelo-Entidad-Media* y *Modelo-Entidad-Persistente* (Consultar Anexo 1). La primera de estas con la responsabilidad de agrupar las clases que identifican las medias y su árbol de jerarquía en la aplicación y la segunda tiene como responsabilidad la gestión de la información persistente, que antes sobrecargaba a la clase *Modelo* del patrón MVC original. Esta variación a su vez sustenta las características actuales de los sistemas multimedia como son: comunicación con bases de datos, archivos XML, o sistemas externos. (FEBE 2006)

### 1.4.3 PATRONES DE DISEÑO

Los Patrones de Diseño son directrices y principios estructurados que describen un problema común y entregan una buena solución ya probada a la que le dan un nombre.

Dentro de las principales ventajas que trae el uso de estos patrones están:

- Ayudan a diseñar correctamente en menos tiempo.
- Ayudan a construir problemas reutilizables.
- Facilitan la documentación.

Existe un grupo de patrones de diseño denominado Patrones de Principio Generales para Asignar Responsabilidades (en sus siglas en inglés GRASP). Dentro de este grupo se encuentra el patrón:

- **Experto:**

Problema que resuelve: Determinar cuál es la clase que debe asumir una responsabilidad a partir de la información que posee cada una.

Solución: Asignar una responsabilidad al experto en información (clase que cuenta con la información necesaria para cumplir la responsabilidad).

- **Creador:**

Problema que resuelve: ¿Quién es el responsable de crear alguna nueva instancia de alguna clase?

Solución: Asignarle a la clase B la responsabilidad de crear las instancias de la clase A en uno de los siguientes casos:

- a) B agrega los objetos de A.
- b) B contiene a los objetos de A.
- c) B registra las instancias de los objetos de A.
- d) B utiliza específicamente los objetos de A.
- e) B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado.

- **Singleton:**

Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella.

## 1.5 LENGUAJES DE MODELADO

Un Lenguaje de Modelado es un conjunto estandarizado de símbolos y modos de disponerlos para modelar un diseño de software o parte de él. Seguidamente se muestran algunas propuestas importantes, de las que se hará una breve descripción y se mostrarán sus principales características.

### 1.5.1 UML

El Lenguaje Unificado de Modelado (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. Cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas:

#### Diagramas de estructura estática:

- **Diagrama de clases:** Clases, interfaces y colaboraciones; así como sus colaboraciones.
- **Diagrama de objetos:** Objetos y sus relaciones.
- **Diagrama de casos de uso:** Casos de uso, actores y sus relaciones.

#### Diagramas de comportamiento:

- **Diagramas de interacción (secuencia y colaboración):** Objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos.

- **Diagrama de estados:** Muestra una máquina de estado que consta de estados, transiciones, eventos y actividades.
- **Diagrama de actividad:** Es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.

#### Diagramas de implementación:

- **Diagrama de componentes:** Organización y las dependencias entre un conjunto de componentes.
- **Diagrama de despliegue:** Configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones: (ORALLO 2001)

- **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.

#### Inconvenientes de UML: (BOOCH *et al.* 2000)

- Falta de integración con otras técnicas (P.ej. diseño de interfaz de usuario)
- UML es excesivamente complejo (y no está del todo libre de ambigüedades): “el 80% de los problemas pueden modelarse usando alrededor del 20% de UML”.

A pesar de que UML permite modelar casi cualquier sistema orientado a objetos, no soporta todos los aspectos de las aplicaciones multimedia de una forma adecuada e intuitiva. Especialmente, las características del lenguaje para modelar los aspectos de la interfaz de usuario, no se aplican explícitamente en los entornos multimedia. Pero gracias a la facilidad de extensión que brinda, surge el Lenguaje Orientada a Objetos para la Modelación de Aplicaciones Multimedia (OMMMA-L), que facilita el modelado de un gran rango de aspectos de aplicaciones multimedia interactivas de una forma integrada y comprensiva.

## 1.5.2 OMMMA-L

En medio de una búsqueda para una modelación adecuada, 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. (LORENTE *et al.* 2006)

OMMMA-L está sustentado en cuatro vistas fundamentales, donde cada una se asocia a un tipo de diagrama en particular. Estas vistas son: (TORRES and ESPINO 2008)

**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. Divide en dos áreas dicho diagrama: una para la jerarquía de los tipos de media y otra para la modelación de la estructura lógica del dominio de la aplicación.

**Vista de Presentación espacial:** Modelada a través de los Diagramas de Presentación de OMMMA-L, los cuales son de nueva aparición en la extensión de UML, dado que este último no contiene un diagrama apropiado para esta tarea. Estos diagramas tienen el propósito de declarar las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área, y separa los objetos de visualización e interacción.

**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.

**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 el comportamiento especificado por el diagrama de secuencia se provoca automáticamente cuando se entra al estado correspondiente donde se hace referencia.

El lenguaje OMMMA-L soporta el modelado de los aspectos estructurales, funcionales y dinámicos de un sistema interactivo y su interfaz de usuario. Además de concentrarse en la funcionalidad desde la perspectiva del sistema de software su sintaxis es definida explícitamente puesto que su semántica es informal e intuitiva.

Es por ello que la extensión del lenguaje notacional UML a OMMMA-L para el tratamiento de aplicaciones educativas multimedia en los últimos años ha sido de gran aceptación.

### 1.5.3 ApEM-L

El Lenguaje para la Modelación de Aplicaciones Educativas y Multimedia (ApEM-L) surge con motivo de dar una nueva solución a la modelación de aplicaciones educativas multimedia en el entorno productivo de la Universidad de las Ciencias Informáticas. Se presenta como una extensión de UML y tomando como base los elementos más significativos de OMMMA-L (2001) y respetando los estándares OCL-2.0 (2003). (Consultar Anexo 2).

La concepción del ApEM-L tuvo los siguientes objetivos:(FEBE 2007)

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

Los conceptos y modelos de ApEM-L pueden agruparse en las siguientes áreas conceptuales:

- **Estructura lógica:** está compuesta por la vista estática y la vista de arquitectura. La primera de ellas está compuesta por el diagrama de clases y el diagrama de casos de uso. Cualquiera de los modelos presentados por ApEM-L define los conceptos claves de la aplicación que modela, las propiedades internas de estos y sus relaciones. Estos conceptos son modelados como clases, describiendo cada una un conjunto de objetos que almacenan información y se comunican para implementar su comportamiento. La información almacenada se representa como atributos de estas clases y las operaciones a través de los métodos de dichas clases. A su vez la vista de arquitectura la componen el diagrama de componentes y el diagrama de

despliegue. De los diagramas mencionados solo han sido modificados los siguientes: diagrama de clases y el diagrama de componentes, el resto mantuvo lo establecido por UML. Para los diagramas de casos de uso, se adicionaron un conjunto de elementos a la descripción textual de los casos de uso propuesta por UML, para una mejor descripción del contexto productivo de los software educativos.

- **Comportamiento dinámico:** realmente esta área no ha sido grandemente modificada, pues solo se ha hecho una pequeña adición al diagrama de secuencia, salvo que ciertamente se ha enriquecido la semántica original de UML para estos diagramas. El comportamiento de la aplicación está descrito por la vista de comportamiento, la cual está compuesta por los diagramas: de actividad, de secuencia, de colaboración y de estados, donde solo ha sido modificado el segundo de los listados anteriormente; adicionando una variable de tiempo donde quiera que es necesario su especificación para un mejor entendimiento.
- **Gestión del modelo:** esta área es la que ha sufrido grandes cambios tanto en su carácter semántico como sintáctico, con la incorporación de estereotipos restrictivos en todos los diagramas a partir de nuevos conceptos incorporados a los diagramas de clases originales o básicos de UML. Se crean dos nuevos diagramas: el de estructura de la presentación y el de estructura de la navegación.

#### 1.5.4 LENGUAJE SELECCIONADO

Después de haber analizado algunos lenguajes de modelado de software como UML y su extensión OMMMA-L, se ha llegado a la conclusión de que a pesar de que en la última década se ha logrado incorporar estos lenguajes notacionales en la modelación de entornos educativos, no logran denotar todos los elementos, tanto ingenieriles como pedagógicos de las aplicaciones educativas cubanas en sus gráficos y semánticas utilizadas.

Por su parte ApEM-L surge con motivo de dar una nueva solución a la modelación de aplicaciones educativas multimedia y dentro de sus principales ventajas están:

- Puede utilizar para su representación todas las herramientas CASE que existen actualmente para la modelación de UML.
- Es un lenguaje que utiliza el estándar internacional OCL, para la modelación de la programación Orientada a Objetos.

- No modifica la semántica del lenguaje base UML, sino que trabaja en estereotipos restrictivos, por lo que a su vez produce modificaciones descriptivas y decorativas en la representación de los componentes del lenguaje base.

En los Anexo 3, 4 y 5 se muestran diferentes tablas donde se comparan determinados elementos de los lenguajes UML y ApEM-L, donde se evidencia una mayor adecuación de ApEM-L como lenguaje para modelar aplicaciones educativas.

Por las razones anteriormente descritas se escoge como lenguaje de modelado a ApEM-L.

## **1.6 HERRAMIENTAS DE MODELADO**

Las herramientas de modelado de sistemas informáticos, son herramientas que se emplean para la creación de modelos de sistemas que ya existen o que se desarrollarán.

Las herramientas de modelado, permiten crear un "simulacro" del sistema, a bajo costo y riesgo mínimo. A bajo costo porque, al fin y al cabo, es un conjunto de gráficos y textos que representan el sistema, pero no son el sistema físico real (el cual es más costoso). Además minimizan los riesgos, porque los cambios que se deban realizar (por errores o cambios en los requerimientos), se pueden realizar más fácil y rápidamente sobre el modelo que sobre el sistema ya implementado.

Las buenas herramientas de modelado cumplen con determinadas características:

- Permiten una visión descendente del sistema.
- Permiten particionar el sistema.
- Poseen componentes gráficos con algo de apoyo textual.
- El modelo resultado debe ser fácil de comprender.
- Poseen mínima redundancia.

A continuación se analizarán dos de las herramientas de modelado más usadas a nivel mundial.

### **1.6.1 RATIONAL ROSE**

Rarional Rose se ha convertido en una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Con el uso del lenguaje común de modelado UML, facilita el trabajo para el equipo de desarrollo y garantiza mayor eficiencia y calidad del trabajo. Permite generar código en diferentes lenguajes de programación partiendo de modelado UML. Además, hace

posible efectuar la ingeniería inversa, es decir, obtener información del diseño de un software partiendo de su código. La interfaz de Rational Rose facilita mucho el uso de la herramienta, está formada por los siguientes elementos principales:

**Navegador:** permite navegar rápidamente a través de las distintas vistas del modelo.

**Ventana de documentación:** para manejar los documentos del elemento seleccionado en cualquiera de los diagramas.

**Barra de herramientas Standard:** para acceder rápidamente a las acciones comunes a ejecutar para cada uno de los diagramas del modelo.

**Barra de herramientas Diagrama:** muestra el conjunto de herramientas disponibles para el diagrama activo.

**Ventana de Diagrama:** permite desplegar y editar cualquiera de los diagramas UML.

**Ventana Registro:** registra todas las órdenes ejecutadas y los errores que se producen durante su ejecución.

**Barra de Estado:** muestra el programa de la carga del modelo, el estado de lectura/escritura del elemento seleccionado, y otros datos de utilidad.

Esta herramienta enmarca los requerimientos y variables del diseño, permitiendo crear un ciclo de desarrollo más pequeño, robusto y adaptable. Evidenciando así las características: (SAÚCO 2008)

**Reusabilidad:** Mediante el modelado visual se pueden crear componentes (modelos), que al salvarlos pueden ser compartidos y reutilizados por varios proyectos, permitiendo que los cambios sean fácilmente incorporados a proyectos existentes.

**Desarrollo basado en componentes:** El desarrollo basado en componentes se ha convertido en el proceso de diseño más efectivo, los usuarios de Rational pueden modelar sus componentes e interfaces sólo haciendo un “arrastrar y soltar” de los componentes del sistema hacia el diagrama de componentes.

Rational Rose permite mantener la consistencia de los modelos del sistema software, realiza chequeo de la sintaxis UML y genera documentos automáticamente. Otra ventaja de Rose es que los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con la función de integración de modelos con otros componentes del proyecto.

### 1.6.2 VISUAL PARADIGM

Visual Paradigm es una herramienta UML profesional fácil de usar. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Dentro de las principales características de la herramienta están:

- Soporte de UML versión 2.1.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio.
- Diagramas de flujo de datos.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas.
- Importación y exportación de ficheros.
- Editor de figuras.

Visual Paradigm ofrece un entorno amigable para el usuario, sugiere nuevos posibles componentes a utilizar, por lo que ya no es necesario localizarlos en la barra y así se crea fácilmente cualquier tipo de diagrama. Incluye gran variedad de estereotipos para la creación de diagramas de fácil entendimiento, los que organiza automáticamente.

### 1.6.3 HERRAMIENTA SELECCIONADA

Anteriormente se analizaron las principales características de las herramientas de modelado de software Rational Rose y Visual Paradigm. Ambas se encargan de llevar a cabo tanto la automatización de los sistemas para la posterior generación de código, como para labores de ingeniería inversa. Esta última es muy importante para obtener la documentación de proyectos que prácticamente ya están realizados, pero que necesitan ser documentados, puesto que la comprensión del código suele ser realmente compleja y se necesita de los diagramas correspondientes debido a que en algún momento el software puede estar sujeto a cambios y es necesario conocer su diseño para efectuar los mismos.

Finalmente se decidió escoger a Visual Paradigm como herramienta modelado debido a que tiene disponibilidad en múltiples plataformas, ofrece apoyo adicional en cuanto a generación de artefactos automáticamente, permite dibujar todos los tipos de diagramas usados en el proceso de desarrollo de software y genera documentación. Además, debido a que es una herramienta libre, permite la modelación de software usando las versiones más actualizadas de UML, lenguaje que constituye la base arquitectónica de ApEM-L.

## **1.7 HERRAMIENTAS DE DESARROLLO DE APLICACIONES CON TECNOLOGÍA MULTIMEDIA.**

Existen diversas herramientas para el desarrollo de aplicaciones con tecnología multimedia, dependiendo de qué es lo que se desea hacer. Pero todas estas tienen algo en común, y es que manejan elementos de media asociados a la programación con el objetivo de lograr una funcionalidad interactiva. Además, permiten el diseño de las interfaces de usuario, posibilitando que se ajuste a las características de la aplicación que se esté desarrollando. Unido a la posibilidad de proporcionar un método para que los usuarios interactúen con el proyecto, la mayoría de las herramientas de desarrollo de multimedia ofrecen también facilidades para crear y editar texto e imágenes, y tienen extensiones para controlar los reproductores de sonido, vídeo y otros periféricos relacionados. Seguidamente se hará un análisis más detallado de algunas de las herramientas más potentes y utilizadas a nivel mundial para el desarrollo de aplicaciones con tecnología multimedia.

### **1.7.1 TOOLBOOK**

ToolBook es una herramienta que permite realizar aplicaciones con tecnología multimedia. Su interfaz aparenta a un programa de dibujo. Posee un lenguaje de programación orientado a objetos llamado OpenScript, el que enriquece extraordinariamente sus posibilidades en la generación de aplicaciones multimedia.

Dentro de las principales características de ToolBook están:

- Interfaces amigables y fáciles de usar.
- Asistentes, plantillas y un catálogo de objetos reusables que ayuda a crear una aplicación en poco tiempo.
- Soporta un amplio rango de elementos multimedia.
- Se pueden incorporar scripts, simulaciones, software de entrenamiento y más.

Las aplicaciones creadas con ToolBook se asemejan a la estructura de un libro, donde cada una de sus pantallas se corresponde con las diferentes páginas del mismo. Además, ToolBook permite asignar un mismo fondo en la aplicación (mediante una imagen, color o conjunto de objetos) para optimizar los recursos tanto de la aplicación como del propio ordenador. (PEÑA 2008)

En fin, ToolBook es una herramienta muy poderosa, especialmente útil en el campo educativo por su interactividad y amplio rango de posibilidades.

### 1.7.2 MACROMEDIA DIRECTOR MX 2004

Macromedia Director MX 2004 es una herramienta diseñada con el fin de desarrollar rápidamente aplicaciones y presentaciones multimedia a través de interfaz basada en una metáfora visual muy fácil de usar. Los archivos de Director, conocidos como películas, permiten combinar varios medios como animación, sonido, video y gráficos para crear espectaculares producciones multimedia. La filosofía seguida por este programa es la de una línea de tiempo durante la cual irán sucediendo diferentes acontecimientos según se vayan necesitando.

Lingo es el lenguaje de programación propio de Director, con una sintaxis relativamente fácil. Se trata de un lenguaje orientado a objetos, de modo que la aplicación correrá cuando se produzca un evento en un objeto dado; pero es que además, cada uno de los objetos de la aplicación puede tener asociados diversos scripts que se ejecutarán con cada evento.

El poderoso lenguaje de programación de Director agiliza los tiempos de desarrollo y ayuda a integrar a las producciones una interactividad de alto nivel. Este lenguaje de programación ofrece posibilidades como: el inspector centralizado de propiedades, el administrador visual del reparto y los comportamientos de tipo “arrastrar y soltar”, para realizar las producciones rápidamente y con el menor esfuerzo.

Una de las principales desventajas de Director, es que los editores que presenta son muy limitados y principalmente sirven para crear objetos muy sencillos, o para retocar los creados externamente. Director no cuenta con las mejores herramientas de dibujo ni de animación, y no permite el trabajo con los Clips de Películas que tanto ayudan a organizar el trabajo, como la herramienta que se tratará a continuación.

### 1.7.3 MACROMEDIA FLASH 8

La herramienta Flash ha evolucionado mucho desde sus primeras versiones hasta llegar a la número ocho, su última versión junto a Macromedia, sus nuevas versiones ahora han estado bajo las licencias de Adobe.

Macromedia Flash 8 cuenta con un conjunto bastante completo de herramientas de dibujos que permiten hacer realidad casi todas las ideas que estén en la mente de un dibujante o diseñador gráfico. Es por ello que a menudo se puede ver en la Web, archivos desarrollados en Flash, ya que estos le reportan dinamismo e interactividad, sobre todo con el uso de las animaciones.

Esta herramienta posibilita la animación de los objetos por medio de fotogramas. Permite realizar dibujos en cada uno de los fotogramas que se desee. Pero existen elementos en una animación que no son necesarios volver a dibujar, para esto Flash ofrece el trabajo por capas, que también es de gran utilidad para separar objetos que se encuentran en diferentes niveles de profundidad en un ambiente de dos dimensiones. Las interpolaciones son otras de las ventajas que proporciona esta herramienta para hacer mucho más fácil el trabajo con las animaciones. Otro de los elementos que simplifica y organiza mucho el trabajo a la hora de animar y diseñar con Flash, es el uso de los símbolos, dentro de los que se destaca el Clip de Película y el Gráfico.

Pero esta herramienta no solo permite el trabajo con imágenes y animaciones, sino que brinda la posibilidad de hacer uso de sonidos de manera muy eficiente así como el trabajo con videos y textos.

Flash posee un reproductor propio que permite mostrar con alta calidad todo lo que está contenido en un archivo hecho con esta herramienta.

Todas estas características unidas su lenguaje de programación orientado a objetos: ActionScript (2.0 para Flash 8) y a su capacidad de interactuar con numerosas bases de datos externas, son las que le dan esa potencialidad a Flash para desarrollar atractivos software con tecnología multimedia.

### 1.7.4 HERRAMIENTA SELECCIONADA

Macromedia Flash 8 es una herramienta orientada a crear aplicaciones y contenidos dinámicos para la Internet, es decir, utilidades interactivas y multimedia con una amplia posibilidad de animación. El resultado de las películas Flash, además de brindar una gran calidad visual, está asegurado en la mayoría de las plataformas con la amplia distribución de su reproductor, Macromedia Flash Player. A continuación se muestra algunas otras potencialidades de esta herramienta:

- **Facilidad de uso:** Flash 8 ofrece un entorno de trabajo amigable para sus usuarios. Cuenta con una ayuda bastante completa, con casi toda la información que se necesita para el trabajo con la herramienta, incluye ejemplos que facilitan su comprensión.
- **Filtros:** Esta versión de Flash incluye una colección de filtros que con su uso se puede obtener resultados fascinantes en el aspecto visual de los objetos como sombra paralela, desenfoque, resplandor, bisel y ajuste de color. Estos son aplicables fundamentalmente a textos y símbolos.
- **Trabajo con símbolos:** Dentro de los símbolos que se pueden usar en Flash están los Clips de Película, los Gráficos y los Botones, este último se usa mucho a la hora de hacer aplicaciones programables como sitios Web, juegos y multimedia. Los dos primeros son más usados para las animaciones y fundamentales también para la creación de estas aplicaciones. Los Clips de Película al igual que los Gráficos son muy importantes para repetir ciertos movimientos dentro de una animación. Esto favorece enormemente el trabajo al animar ciertos objetos.
- **Calidad de los textos:** A través del motor de procesamiento de fuentes FlashType, se obtienen textos claros y de alta calidad, sobre todo para el desarrollo de aplicaciones que van a ser utilizadas en displays pequeños como el de teléfonos celulares.
- **Compatibilidad:** Las aplicaciones desarrolladas con Flash 8 soportan prácticamente cualquier tipo de plataforma Windows, Macintosh, Unix, PDAs y teléfonos móviles lo que aumenta aún más su compatibilidad y difusión.
- **Importación de videos:** Flash posibilita la importación de videos de diferentes formatos, esto es de gran utilidad si se quiere agregar a una aplicación algún video externo.
- **Reproductor propio:** El reproductor de los archivos hechos con Flash se nombra Flash Player. Una de las principales ventajas de este reproductor es su portabilidad, ya que puede ser incluido en un archivo ejecutable (.exe) y este podrá ser reproducido sin depender de si está instalado o no el Flash Player en una computadora. El rendimiento de este reproductor desde sus primeras versiones se ha elevado mucho, se estima que de dos a cinco veces. Mediante código se puede manejar muchas de las propiedades de este reproductor como configurar la calidad de los gráficos, ponerlo en pantalla completa, entre otros.
- **Compatible con XML:** XML se ha convertido con éxito en el estándar de las comunicaciones por Internet; lo que antes se transmitía en formatos propietarios ahora es fácilmente

interoperable gracias al lenguaje XML. Flash usa dentro de ActionScript el objeto XML para la carga de este tipo de archivos.

- **ActionScript 2.0:** Es el lenguaje de programación que usa Flash 8 para sus aplicaciones. Está orientado a objetos, permitiendo la interpretación de la programación como situaciones del mundo real. Soporta el trabajo con herencia y eventos.

Finalmente se decide seleccionar a Macromedia Flash 8 como herramienta de desarrollo, debido a las potencialidades que ofrece para la creación de aplicaciones con tecnología multimedia, y unido a esto, al previo conocimiento y experiencia del equipo de desarrollo sobre esta octava versión de Flash.

## **1.8 CONCLUSIONES DEL CAPÍTULO**

En este capítulo se hizo un análisis del estado del arte del software educativo con tecnología multimedia en el ámbito nacional e internacional. Se hizo un estudio de la tecnología para el desarrollo de este tipo de software. Para la solución propuesta se decidió el uso de RUP como metodología de desarrollo. Como lenguaje de modelado se escogió a ApEM-L, una extensión de UML para aplicaciones educativas multimedia. Como herramienta de modelado se decidió usar Visual Paradigm. Y como herramienta de programación y diseño a Macromedia Flash 8 con el lenguaje de programación orientado a objetos ActionScript. También se hizo referencia a la arquitectura de software y a diferentes patrones de diseño de software.

## **CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA**

En este capítulo además de identificar la audiencia, se hace una descripción de la solución que se propone para el problema identificado. Se modela el entorno donde se usará el software y se identifican y describen sus principales conceptos asociados, reflejados mediante la realización de un Modelo del Dominio. Se definen los requerimientos funcionales y no funcionales por los que se regirá el software propuesto. Se representa además, la Vista de Gestión del Modelo y sus subsistemas. Se hace una descripción textual de las principales Vistas de Presentación y se muestran los Diagramas de Estructura de Presentación y los Diagramas de Estructura de Navegación como artefactos propuestos por el lenguaje de modelado utilizado: ApEM-L.

### **2.1 LA AUDIENCIA**

Al dar inicio a un proceso de desarrollo de software es de vital importancia identificar a quién va dirigido este producto. A partir de aquí se puede definir con mayor exactitud, el diseño de las interfaces de la aplicación, los colores, la tipografía y otros elementos importantes de presentación.

El software educativo a desarrollar va dirigido a niños cubanos de escolaridad primaria con edades comprendidas entre los 7 y 12 años. Sabido esto, junto a la historia que encierra el juego, se puede definir las características de los escenarios, los personajes, los complementos de ambientación y hasta de los sonidos a emplear. Y se hace notorio el uso de animaciones debido al alto poder atractivo y comunicativo de estas, así como al grado de aceptación que generan las mismas en este tipo de usuarios.

Para hacer uso del software en cuestión, el usuario debe tener conocimientos básicos del movimiento del mouse y del trabajo con el teclado, además, debe saber leer y poseer previos conocimientos de la vida y obra de José Martí.

### **2.2 SOLUCIÓN PROPUESTA.**

“Pico Maestro” es el título del software educativo con tecnología multimedia que se propuso desarrollar como solución al problema planteado. Se presenta en forma de juego educativo, cuyas características permiten su adecuación al entorno identificado.

Cuando el usuario accede al juego se muestra un video que introduce a este último, donde se plantea la misión a cumplir, esta consiste en escalar la montaña más alta de la geografía cubana, el Pico Turquino ubicado en oriente cubano, con el objetivo de homenajear el busto de José Martí que se encuentra en su cima y clavar una bandera que identificará a los niños como “Martianitos”, calificativo que coincide con el nombre de un paquete de juegos educativos que se detallará más adelante.

La escalada estará dividida en diez niveles, en los que el personaje, guiado por el usuario, debe ir saltando sobre plataformas de diferentes alturas evitando obstáculos enemigos, precipicios y respondiendo todas las incógnitas que propone el nivel en que se esté jugando. Cada pregunta, en dependencia de la respuesta dada por el usuario, acumulará o disminuirá determinada cantidad de puntos, los que colocan al jugador en cierta posición del ranking de partidas ganadas. El contenido de las preguntas a responder está acorde con el desarrollo psicológico de niños de esas edades y en correspondencia con los conocimientos que se imparten en la enseñanza primaria sobre la vida y obra de José Martí.

Como toda escalada al Pico Turquino debe estar acompañada de un guía, pues en esta no podía faltar. “Lapiolo” es el nombre del personaje que guiará al usuario, al que se podrá consultar ante determinadas situaciones que se presentan en el juego, así como a la hora de responder una pregunta.

Debido a que cumplir la misión que propone el juego no es tarea de poco tiempo, el niño podrá guardar su partida para poder continuar jugándola en otro momento. También se le brindan otras opciones al usuario como configurar el volumen del sonido de fondo y ajustar los gráficos del juego.

Pico Maestro formará parte de un paquete de juegos educativos que se viene desarrollando con un mismo objetivo: contribuir con el estudio de la vida y obra de José Martí. El nombre del paquete es “Martianitos” y ya contiene juegos que han tenido gran aceptación por los pioneros de las escuelas primarias en los que han sido probados, dentro de las que se encuentran: Francisco Cruz Bourzac, ubicada en el municipio capitalino La Lisa, Víctor Mestres, ubicada en municipio habanero San José de las Lajas, y Manuel “Piti” Fajardo Rivero ubicada la provincia Ciego de Ávila en el municipio Bolivia.

Entre los juegos que incluye Martianitos hasta el momento están: “Versos contra Nudo” y “Bosque en Piezas”.

## 2.3 MODELO DEL DOMINIO

Por pequeño que sea un sistema, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales.

Debido a que para la modelación del software educativo propuesto no se lograron definir procesos específicos en el entorno, solamente se pudo identificar conceptos y objetos relacionados con el mismo, se decide representar los conceptos que se manejan en la situación existente a través de un Modelo del Dominio, el cual es una de las alternativas que brinda RUP para la identificación de requisitos y la comprensión del contexto cuando existe poca estructuración en los procesos de negocio.

El Modelo del Dominio (o Modelo Conceptual) es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema. Representa conceptos del mundo real, no de los componentes de software. (JACOBSON, I. *et al.* 2004)

### 2.3.1 CONCEPTOS DEL DOMINIO

**Maestro:** Persona que se encarga de la educar y enseñar a los alumnos.

**Alumno:** Niños de entre 7 y 12 años de edad.

**Medio de Enseñanza:** Materiales que se utilizan para apoyar el estudio de la vida y obra de José Martí.

**Cuaderno Martiano:** Selección que reúne diferentes textos de José Martí.

**Lámina:** Rectángulo de cartón que contiene alguna imagen.

**Libro:** Conjunto de tópicos o epígrafes relacionados con un tema en general y que aparecen como un todo, unificados bajo un mismo título y autor.

**Mural Martiano:** Medio donde se expone contenido relacionado con José Martí. Generalmente soportado sobre una plancha de cartón, forrado con papel y adornado con imágenes y colores.

**Material Audiovisual:** Producto digital de comunicación compuesto por imágenes y sonidos. Son proyectados fundamentalmente en los televisores y computadoras.

**Documental:** Material audiovisual que narra algún acontecimiento histórico.

**Película:** Espectáculo cinematográfico que narra una historia.

**Video:** Material audiovisual que hace referencia a imágenes acompañada fundamentalmente por los versos de José Martí musicalizados. Aunque pueden incluir otras canciones relacionadas con Martí.

### 2.3.2 DIAGRAMA DE CLASES DEL MODELO DEL DOMINIO

El Modelo del Dominio se representa a través de un Diagrama de Clases en el que se muestran los conceptos u objetos del dominio del problema, las asociaciones entre ellas y los atributos que poseen. Su objetivo principal es brindar una mayor comprensión de los eventos que suceden en el entorno que se desea modelar. A continuación se representan los conceptos que anteriormente fueron descritos y la interacción entre ellos.

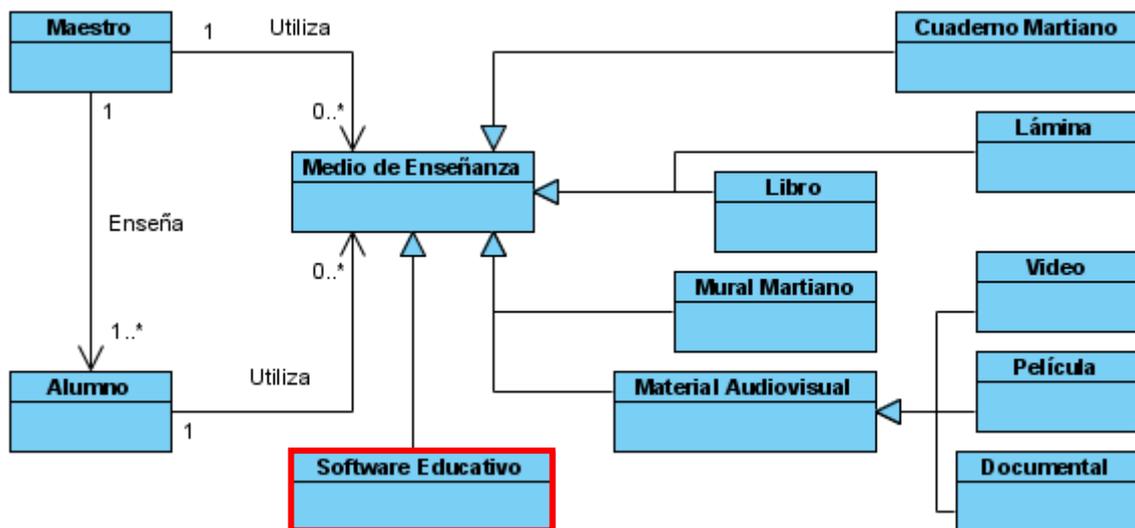


Figura 1. Diagrama de Clases del Modelo del Dominio

## 2.4 DESCRIPCIÓN DE LA FUNCIONALIDAD

Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requerimientos. Estos cumplen un papel primordial en el proceso de desarrollo del software, ya que marcan el punto de partida para actividades como la planeación, contribuyen a la identificación de las funcionalidades, el comportamiento de entrada y salida del sistema. Los requerimientos generan especificaciones correctas que describen con claridad, sin ambigüedades y en forma consistente las características o cualidades que debe poseer la futura aplicación. Se pueden clasificar en: funcionales y no funcionales.

### 2.4.1 REQUERIMIENTOS FUNCIONALES

Los Requerimientos Funcionales de un sistema especifican acciones que este debe ser capaz de realizar, sin considerar restricciones físicas. El sistema propuesto debe ser capaz de:

1. Mostrar presentación del juego.
  - 1.1. Saltar presentación del juego.
2. Mostrar pantalla principal del juego.
3. Mostrar la Ayuda.
4. Mostrar los Créditos.
5. Cargar una partida guardada.
6. Permitir la selección de un personaje.
  - 6.1. Asignar un nombre al personaje.
7. Mostrar introducción de la sección Juego.
8. Mostrar la sección Juego.
  - 8.1. Permitir el control de los movimientos del personaje.
  - 8.2. Permitir la interacción del personaje con los elementos del nivel.
9. Mostrar el mapa.
10. Mostrar una pista solicitada.
11. Permitir la configuración del juego.

- 11.1. Ajustar el volumen del sonido de fondo.
- 11.2. Ajustar los gráficos.
12. Guardar la partida.
13. Mostrar pregunta seleccionada.
  - 13.1. Mostrar mensaje de error o felicitación ante una respuesta dada.
14. Mostrar sección de recuperación del personaje.
  - 14.1. Continuar la partida.
15. Permitir el paso al siguiente nivel.
  - 15.1. Mostrar presentación del nuevo nivel alcanzado.
16. Mostrar sección de Perder Juego.
17. Mostrar la sección Ganar Juego.
  - 17.1. Mostrar ranking por puntos.
18. Cerrar la aplicación.
  - 18.1. Confirmar salida.

## 2.4.2 REQUERIMIENTOS NO FUNCIONALES

Los Requerimientos no Funcionales de un sistema expresan las características o propiedades que de una u otra forma puedan limitar al mismo, es decir, son las características que permiten obtener un producto atractivo, usable, rápido o confiable.

### Requerimientos de apariencia o interfaz externa:

- La aplicación se ejecutará a pantalla completa.
- La aplicación tendrá una interfaz sencilla, intuitiva y amigable para sus usuarios.
- Se deben usar colores llamativos, ordenados en una gama coherente.
- El diseño de la interfaz gráfica deberá garantizar la distinción visual entre los elementos del sistema.
- La entrada de cada una de las pantallas debe estar acompañada alguna animación que muestre dinamismo.

**Requerimientos Políticos - Culturales:**

- Se deben mostrar elementos característicos del lugar que se simula, y también símbolos patrios, como la Palma Real, la flor Mariposa y el Toco-ro-ro.
- En determinadas pantallas se debe mostrar el rostro de José Martí.

**Requerimientos de Ayuda del sistema:**

- La aplicación contará con una ayuda que contendrá indicaciones básicas para usar el software.

**Requerimientos de Navegación:**

- Desde cualquier pantalla se podrá acceder a la pantalla Principal (exceptuando las pantallas de transición PerderVida y CambiarNivel).
- Solo se puede salir de la aplicación desde la pantalla Principal, con una previa confirmación del usuario.

**Requerimientos de Usabilidad**

- La aplicación deberá poseer una interfaz asequible y funcional, tanto para usuarios expertos como para los que no tienen conocimientos profundos de informática.

**Requerimientos de Software:**

- Flash Player 7 o superior.

**Requerimientos de Portabilidad:**

Multiplataforma. La aplicación podrá ser ejecutada en cualquier sistema operativo que soporte a Flash Player, para esto se dispone de un archivo swf. Para el caso de las plataformas Windows, por ser las más común en las escuelas primarias cubanas, se dispondrá como otra opción, de un archivo .exe para ejecutar la aplicación, evitando con este la dependencia del Flash Player.

Entre los sistemas operativos identificados están:

- Microsoft Windows 98, ME, NT4, 2000, XP, Vista o superior.
- Macintosh OS X versión 10.1 o superior.
- GNU/Linux.

### Requerimientos de Hardware

- Procesador Pentium II o superior.
- 256 MB de RAM (Mínimo).
- 16 MB de espacio libre de disco (Mínimo).
- Tarjeta de sonido y video.
- Kit de Multimedia, Mouse y Teclado.
- Unidad de lectura CD.

### Restricciones en el diseño y la implementación:

- La herramienta de desarrollo de la aplicación será Macromedia Flash 8.

## 2.5 VISTA DE GESTIÓN DEL MODELO Y SUS SUBSISTEMAS

El principio de organización que propone ApEM-L para la construcción de un sistema se basa en la división de la aplicación por subsistemas. Cada subsistema va a estar compuesto por una o más Vistas de Presentación. Las relaciones entre los subsistemas van a ser representados mediante la Vista de Gestión del Modelo (VGM). Para la construcción de juego educativo Pico Maestro se han identificado cinco subsistemas, definidos por las características y funcionalidades de las vistas que lo conforman.

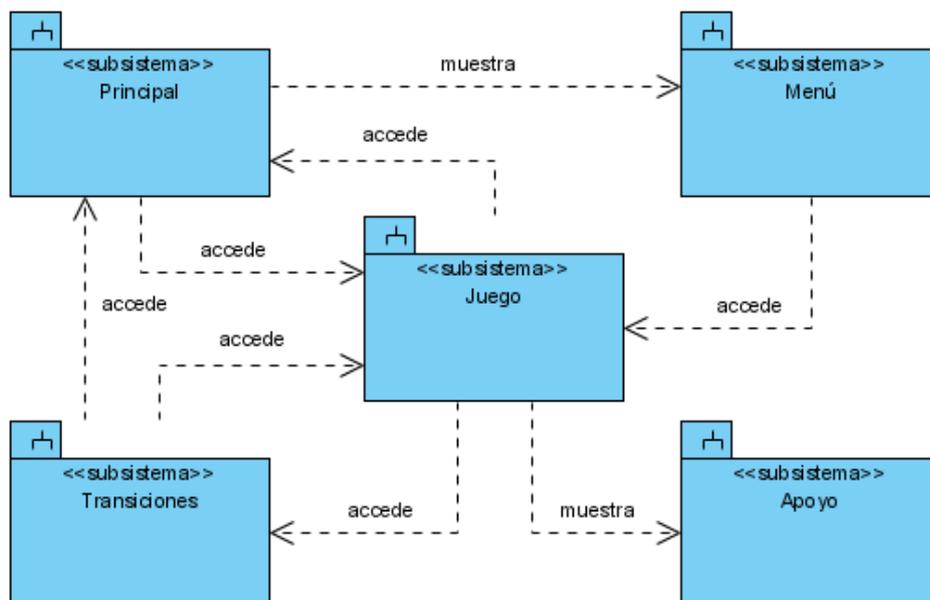


Figura 2. Representación de la Vista de Gestión del Modelo

**Subsistema Principal:** Está compuesto por la vista donde se muestra la animación que da presentación al juego y la vista principal, a través de esta última se puede mostrar todas las vistas del subsistema Menú y acceder a la vista de selección del jugador perteneciente al subsistema Juego.

Vistas que contiene:

- Vista Presentacion
- Vista Principal

**Subsistema Menú:** Está compuesto por las vistas que permiten al usuario realizar diferentes funciones antes de comenzar a jugar. Desde cualquiera de estas vistas se podrá acceder a la vista Principal. Desde la vista Ayuda se puede acceder a la vista de selección del jugador perteneciente al subsistema Juego. Desde la vista donde se cargan las partidas guardadas se puede acceder directamente a la vista EntradaJuego perteneciente al subsistema Juego.

Vistas que contiene:

- Ayuda
- Creditos
- CargarPartida
- ConfirmarSalida

**Subsistema Juego:** Está compuesto por las vistas que permiten el desarrollo del juego, como la de selección del personaje y la vista donde se desenvuelve el juego. Desde estas vistas se puede acceder a la vista Principal perteneciente al subsistema Principal excepto de la vista EntradaJuego que puede acceder solo a la vista Juego. Desde esta última se puede acceder a las vistas del subsistema de apoyo y a las del subsistema Transiciones.

Vistas que contiene:

- SeleccionarJugador
- EntradaJuego
- Juego

**Subsistema Apoyo:** Está compuesto por las vistas que permiten al usuario obtener información útil para su desenvolvimiento en el juego y la vista que permite la configuración de ciertos elementos de la

aplicación. Desde cada una de estas vistas se puede acceder a la vista Juego perteneciente al subsistema Juego.

Vistas que contiene:

- Mapa
- Configuracion
- ContestarPregunta
- Pista
- Guardando

**Subsistema Transiciones:** Está compuesto por las vistas que marcan un cambio sustancial en el desenvolvimiento del juego.

Vistas que contiene:

- PerderVida
- CambiarNivel
- PerderJuego
- GanarJuego

### 2.5.1 VISTAS DE PRESENTACIÓN

La Vista de Presentación es el artefacto que recoge la descripción de los elementos y su relación para el cumplimiento de una determinada función del sistema. 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.

### 2.5.2 DESCRIPCIÓN TEXTUAL DE LAS VISTAS DE PRESENTACIÓN

La Descripción Textual de las Vistas de Presentación surge para una mejor descripción del contexto productivo del software educativo, bien explícito, incluyendo objetivos pedagógicos, elementos asociados a las vistas y medias a utilizar. Su objetivo principal es dejar claro en detalle todo el flujo de evento así como todas las características esenciales y necesarias de la vista de presentación.

**Tabla 1.** Descripción textual de la Vista de Presentación Principal

<b>Descripción Textual de la Vista de Presentación <i>Principal</i></b>		
<b>Actores de la Vista de Presentación</b>	Usuario	
<b>Propósito</b>	Permitir al usuario de la aplicación seleccionar entre las opciones de trabajo dentro de la aplicación en su pantalla principal.	
<b>Objetivos pedagógicos</b>	-	
<b>Resumen</b>	La vista se inicia cuando culmina la animación de presentación o cuando el usuario salta la presentación. Muestra al usuario vínculos a la Ayuda, los Créditos, Cargar Partida, Jugar y posibilita la salida de la aplicación.	
<b>Vistas asociadas</b>	Ayuda, Créditos, CargarPartida, SeleccionarJugador, ConfirmarSalida.	
<b>Referencias</b>	R2	
<b>Precondiciones</b>	Finalizada la animación de la vista Presentacion ó el usuario salta la presentación.	
<b>Poscondiciones</b>	Se muestra la vista Principal.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
<p>2. El usuario hace clic sobre el botón Ayuda.</p> <p>4. El usuario hace clic sobre el botón Créditos.</p> <p>6. El usuario hace clic sobre el botón Cargar Partida.</p> <p>8. El usuario hace clic sobre el botón Salir.</p>	<p>1. Se muestran los elementos de la vista Principal.</p> <p>3. Se muestra la vista Ayuda.</p> <p>5. Se muestra la vista Créditos.</p> <p>7. Se muestra la vista CargarPartida.</p> <p>9. Se muestra la vista ConfirmarSalida.</p>	<p>1.1. Entrada con efecto de transparencia de los elementos implicados en la vista. Estas entradas estarán basadas en movimientos de desaceleración. Las ubicaciones serán:</p> <p>1.1.1. El logotipo del juego estará ubicado en el centro superior de la pantalla.</p> <p>1.1.2. Los vínculos a las diferentes secciones estarán</p>

		<p>situadas debajo del logotipo y en el centro de la pantalla.</p> <p>Un poco más abajo la opción de salir.</p> <p>1.1.3. A ambos extremos de la pantalla se mostrará una animación de los personajes principales (hembra y varón).</p> <p>1.1.4. En la esquina superior izquierda se mostrará una animación de la mascota.</p>	
<b>Cursos Alternos de los Eventos</b>			
<b>Acción</b>		<b>Curso Alterno</b>	
-		-	
<b>Prioridad</b>		Crítica	
<b>Mejoras</b>		-	
<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>
<b>Imagen</b>	I001	Imagen tenue de José Martí en el nivel más bajo de profundidad.	Existente.
	I002	Se mostrará la base de una montaña que indica el comienzo del camino por recorrer.	Existente.
	I003	Imagen de la palma real ( la que tiene dos racimos de palmiche rojo)	Existente.
	I 004	Imagen de la flor Mariposa.	Existente.
	I 005	Borde azul degradado y cuadriculado por líneas semitransparentes.	Existente.
<b>Animación</b>	A002	Animación en donde la mascota (Lapiolo) está recostada.	Existente.
	A003	Animación del logotipo de Pico Maestro, donde vate	Existente.

		la bandera con el isotipo del paquete Martianitos.	
	A004	Animación que muestra el tocororo posado en una rama seca.	Existente.
	A005	Animación que muestra al personaje principal en su versión femenina, de frente y solo pestaña, con cierta expresión de alegría.	Existente.
	A006	Animación que muestra al personaje principal en su versión masculina, de frente y solo pestaña, con cierta expresión de alegría.	Existente.
<b>Sonido</b>	S021	Sonido muy breve para cuando se pase el cursor sobre un botón.	Existente.
	S022	Sonido muy breve para cuando se haga clic sobre un botón.	Existente.
<b>Texto</b>	T001	Advierte sobre el mal uso del producto.	Existente.
<b>Reglas pedagógicas</b>			
-			

**Tabla 2.** Descripción textual de la Vista de Presentación Juego

<b>Descripción Textual de la Vista de Presentación <i>Juego</i></b>	
<b>Actores de la Vista de Presentación</b>	Usuario
<b>Propósito</b>	Permitir al usuario de la aplicación mover al personaje por las plataformas del nivel interactuando con los elementos del mismo incluyendo los precipicios y las preguntas.
<b>Objetivos pedagógicos</b>	Permite al usuario responder deferentes preguntas propuestas sobre la vida y obra de José Martí. Permitir al usuario desarrollar habilidades con el mouse y el teclado, estimular al uso de la lógica ante diferentes situaciones, ejercitar la coordinación de los movimientos al vincular actividades como caminar, saltar, retroceder, tomar impulso para lograr un salto mayor, entre otros movimientos

	del mundo real.	
<b>Resumen</b>	La vista muestra una serie de plataformas ubicadas estratégicamente para que usuario pueda guiar los movimientos del personaje sobre todas estas. Incluye elementos con los que tendrá que interactuar o evitar el usuario. Algunos de estos elementos favorecen atributos del personaje como los puntos y la energía, otros afectan a esta última y a las vidas del personaje. Otros elementos del nivel son las preguntas, con las que tendrá que interactuar el personaje. En esta vista aparecen otras opciones para el usuario.	
<b>Vistas asociadas</b>	Principal, Mapa, Pista, Configuración, ContestarPregunta, PerderVida, CambiarNivel, PerderJuego, GanarJuego.	
<b>Referencias</b>	R8	
<b>Precondiciones</b>	-	
<b>Poscondiciones</b>	Se muestran todos los elementos del nivel.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
2. El usuario oprime las teclas direccionales del teclado.	<p>1. Se muestran todos los elementos del nivel.</p> <p>3. Se desplaza el personaje en la dirección indicada y este interactúa con elementos con los que haga contacto obteniendo de los mismos resultados favorables o desfavorables en dependencia de la naturaleza de estos.</p> <p>Elementos de Interacción:</p> <ul style="list-style-type: none"> <li>Plataformas: El personaje camina sobre ellas. Si este cae por un precipicio se muestra la vista PerderVida y se penaliza con la</li> </ul>	<p>1.1. Si es una nueva partida se accede al primer nivel.</p> <p>Si es una partida cargada se accede al nivel de la partida.</p> <p>Los elementos a mostrar son:</p> <ul style="list-style-type: none"> <li>Elementos de Interacción.</li> <li>Complementos de ambientación: <ul style="list-style-type: none"> <li>Montaña de fondo.</li> <li>Cielo.</li> <li>Nubes.</li> </ul> </li> </ul>

	<p>perdida de una vida.</p> <ul style="list-style-type: none"> <li>• Objetos de puntos: Son esferas con aspecto de cristal que tienen una imagen de Martí y pueden ser de tres variedades:             <ul style="list-style-type: none"> <li>▪ Tonalidad verde: 10 puntos.</li> <li>▪ Tonalidad naranjada: 15 puntos.</li> <li>▪ Tonalidad morada: 20 puntos.</li> </ul> </li> <li>• Vidas: Suman una vida si le quedan menos de cinco.</li> <li>• Frutas: Proporcionan energía siempre que esta esté por debajo del 100%. Las frutas son:             <ul style="list-style-type: none"> <li>▪ Mango: Aumenta 30%.</li> <li>▪ Guayaba: Aumenta 25%.</li> <li>▪ Piña: Aumenta 23%.</li> <li>▪ Plátano: Aumenta 20%.</li> <li>▪ Mamey: Aumenta 18%.</li> <li>▪ Melón: Aumenta 16%.</li> <li>▪ Uvas: Aumenta 15%.</li> </ul> </li> <li>• Obstáculos: Afectan el paso por determinado lugar y disminuyen la vida y la energía del personaje.</li> <li>• Preguntas: Al seleccionar una pregunta se muestra la vista ContestarPregunta.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Mata de coco.</li> <li>▪ Palma Real.</li> <li>▪ Flor Mariposa.</li> <li>▪ Tocaroro.</li> <li>• Elementos de Información:             <ul style="list-style-type: none"> <li>▪ Mapa: Muestra la vista Mapa</li> <li>▪ Número del nivel.</li> <li>▪ Cantidad de puntos.</li> <li>▪ Barra de energía.</li> <li>▪ Barra de vida.</li> <li>▪ Cantidad de vidas.</li> <li>▪ Mascota Lapiolo: Muestra la vista Pista.</li> <li>▪ Vínculo a la vista Configuración: Muestra la vista Configuracion.</li> <li>▪ Icono de Guardar Partida: Se guarda la partida y se muestra una barra que simula el guardado de la partida.</li> <li>▪ Vínculo a la vista Principal: Muestra la</li> </ul> </li> </ul>
--	--	--

			vista Principal.
<b>Cursos Alternos de los Eventos</b>			
<b>Acción</b>		<b>Curso Alterno</b>	
-		-	
<b>Prioridad</b>		Crítica	
<b>Mejoras</b>		-	
<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>
<b>Imagen</b>	I013	Imagen de color azul degradado que representa el cielo.	Existente.
	I014	Imagen de fondo transparente de una nube.	Existente.
	I015	Imagen de la montaña para los fondos.	Existente.
	I016	Imagen de las piedras grandes de tonalidad carmelita.	Existente.
	I017	Imagen de las piedras de tonalidad gris.	Existente.
	I018	Imagen de la palma Real.	Existente.
	I019	Imagen de la Flor Mariposa.	Existente.
	I020	Imagen de José Martí donde aparece solo su parte de su rostro: de los hombros para arriba.	Existente.
<b>Animación</b>	A016	Animación donde personaje principal aparece detenido, se muestra de perfil por un tiempo y después voltea la cara mirando al usuario con un rostro amistoso.	Existente.
	A017	Animación donde personaje principal aparece detenido con un rostro que refleja cansancio.	Existente.
	A018	Animación donde personaje principal aparece desmayado cayendo en una camilla.	Existente.
	A019	Animación donde personaje principal aparece caminando con su mochila en la espalda.	Existente.

A020	Animación donde personaje principal aparece con la mirada hacia arriba manteniendo una postura de salto.	Existente.
A021	Animación donde personaje principal aparece con la mirada hacia abajo manteniendo una postura de bajad de un salto.	Existente.
A022	Animación donde personaje principal aparece bocarriba sufriendo de un impacto, con las estrellitas dándole vueltas encima.	Existente.
A023	Animación donde personaje principal aparece con una expresión de susto como sintiendo los efectos de caída al vacío.	Existente.
A024	Animación donde personaje principal aparece lamentando el contacto con algún objeto. Su tono es rojo al principio y después se va retomando sus colores originales.	Existente.
A025	Animación donde personaje principal aparece indicando la imposibilidad de seguir avanzando ya que aún quedan preguntas por contestar.	Existente.
A026	Animación donde personaje principal aparece indicando la imposibilidad de seguir avanzando en sentido inverso.	Existente.
A027	Animación donde personaje principal aparece enredado en las telas de una araña, muestra una expresión de susto y un tono rojo en su rostro.	Existente.
A028	Animación donde personaje principal aparece con una mano extendida hacia arriba, como suspendido de algo que asciende.	Existente.
A029	Animación donde personaje principal aparece comiendo.	Existente.
A030	Animación de la flecha verde con efecto cristal que	Existente.

	indica la ubicación del personaje.	
A031	Animación donde aparece un panal con abejas volando a su alrededor constantemente.	Existente.
A032	Animación donde aparece el Tocatoro posado en una rama seca, donde hace movimientos leves.	Existente.
A033	Animación donde aparece un alacrán caminando de un lado para otro.	Existente.
A034	Animación donde aparece una piedra que cae dando vueltas.	Existente.
A035	Animación donde aparece la candela ardiendo.	Existente.
A036	Animación donde aparece la candela apagándose.	Existente.
A037	Animación donde aparece una esfera con pinchos subiendo y bajando con giros.	Existente.
A038	Animación donde aparece el murciélago volando de arriba abajo y viceversa comiendo de un mango.	Existente.
A039	Animación donde aparece el murciélago volando detrás del mango que cae.	Existente.
A040	Animación donde aparece un péndulo con una esfera de pinchos.	Existente.
A041	Animación donde aparece un majá que se desenrosca repentinamente y vuelve a su posición inicial.	Existente.
A042	Animación donde aparece una lagartija sacando la lengua.	Existente.
A043	Animación donde aparece una araña tranquila en su tela.	Existente.
A044	Animación donde aparece una araña recorriendo su tela.	Existente.
A045	Animación donde aparece un caracol con aspecto natural desplazándose de un lado a otro.	Existente.

	A046	Animación donde aparece un caracol con aspecto de dibujo estacionado.	Existente.
	A047	Animación donde aparece un caracol con aspecto de dibujo ascendiendo.	Existente.
	A048	Animación donde aparece un resorte (con un cojín en un extremo) que se dispara repentinamente.	Existente.
	A049	Animación donde aparecen las estrellas dando vueltas rodeando los puntos.	Existente.
	A050	Animación donde aparece la mascota Lapiolo observando lo que sucede en el juego.	Existente.
	A051	Animación donde desaparece el corazón al variar su tamaño hasta su mínima expresión, esto representa una vida que se pierde.	Existente.
	A052	Animación donde el corazón varía su tamaño desde pequeño a grande y viceversa hasta alcanzar el tamaño normal, esto representa una vida que se gana.	Existente.
	A053	Animación donde aparece la mascota Lapiolo mirando para el usuario como esperando alguna orden de este.	Existente.
<b>Sonido</b>	S002	Sonido de fondo para esta vista.	Existente.
	S003	Sonido para cuando el personaje salta por voluntad del usuario.	Existente.
	S004	Sonido para cuando el personaje salta por impulso del majá.	Existente.
	S005	Sonido para cuando el personaje come.	Existente.
	S006	Sonido para cuando el personaje cae a un precipicio.	Existente.
	S007	Sonido de la candela ardiendo.	Existente.
	S008	Sonido del agua arrojada a la candela.	Existente.
	S009	Sonido de las ramas al tumbar el mango.	Existente.
	S010	Sonido para cuando el personaje interactúa con un	

		objeto de puntos.	
	S011	Sonido para cuando se aumentan los puntos.	Existente.
	S012	Sonido para cuando se disminuyen los puntos.	Existente.
	S013	Sonido donde se escuchan pájaros cantando, para simular los efectos de un impacto en el personaje principal.	Existente.
<b>Texto</b>	T010	Texto que indica el número del nivel que se está jugando.	Existente.
	T011	Texto que indica la cantidad de puntos obtenidos.	Existente.
	T012	Texto compuesto solo por la palabra: Nivel.	Existente.
	T013	Texto compuesto solo por la palabra: Puntos.	Existente.
	T014	Texto compuesto solo por la palabra: Energía.	Existente.
	T015	Texto compuesto solo por la palabra: Vidas.	Existente.
<b>Reglas pedagógicas</b>			
-			

Las descripciones textuales del resto de las vistas de la aplicación pueden ser consultadas en los Anexos 6-21.

### 2.5.3 DIAGRAMAS DE ESTRUCTURA DE PRESENTACIÓN

Los Diagramas de Estructura de Presentación (DEP) permiten representar la estructura que tendrán las interfaces de usuario, estableciendo una organización lógica de los elementos conformantes de las mismas y dejándoles a los diseñadores gráficos la función de decidir dónde y cómo serán en términos visuales dichos elementos.

Para la mejor estructuración del modelo ApEM-L define dos nuevos tipos de clases: 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. La clase estáticos agrupará los componentes que solo tiene como función visualizar información, pero que no permiten interacción con el usuario. Todo lo contrario con los agrupados bajo la clase interacción, los cuales serán los elementos de la vista que permiten interacción del usuario con el sistema informático modelado. Para cada una de las vistas de

presentación se realiza DEP. A continuación se muestran los DEP de las diferentes vistas de presentación de Pico Maestro:

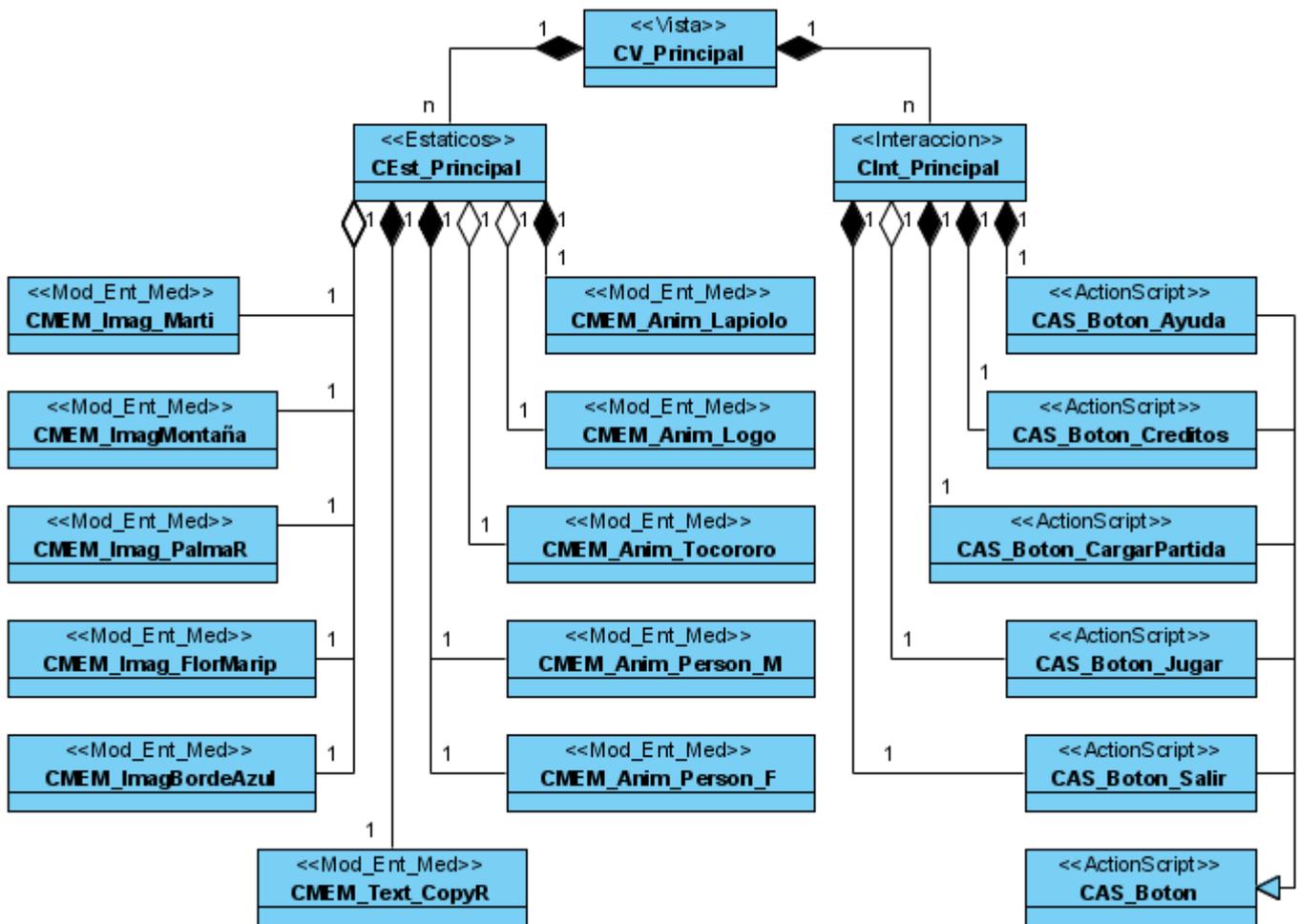


Figura 3. DEP Vista de Presentación Principal

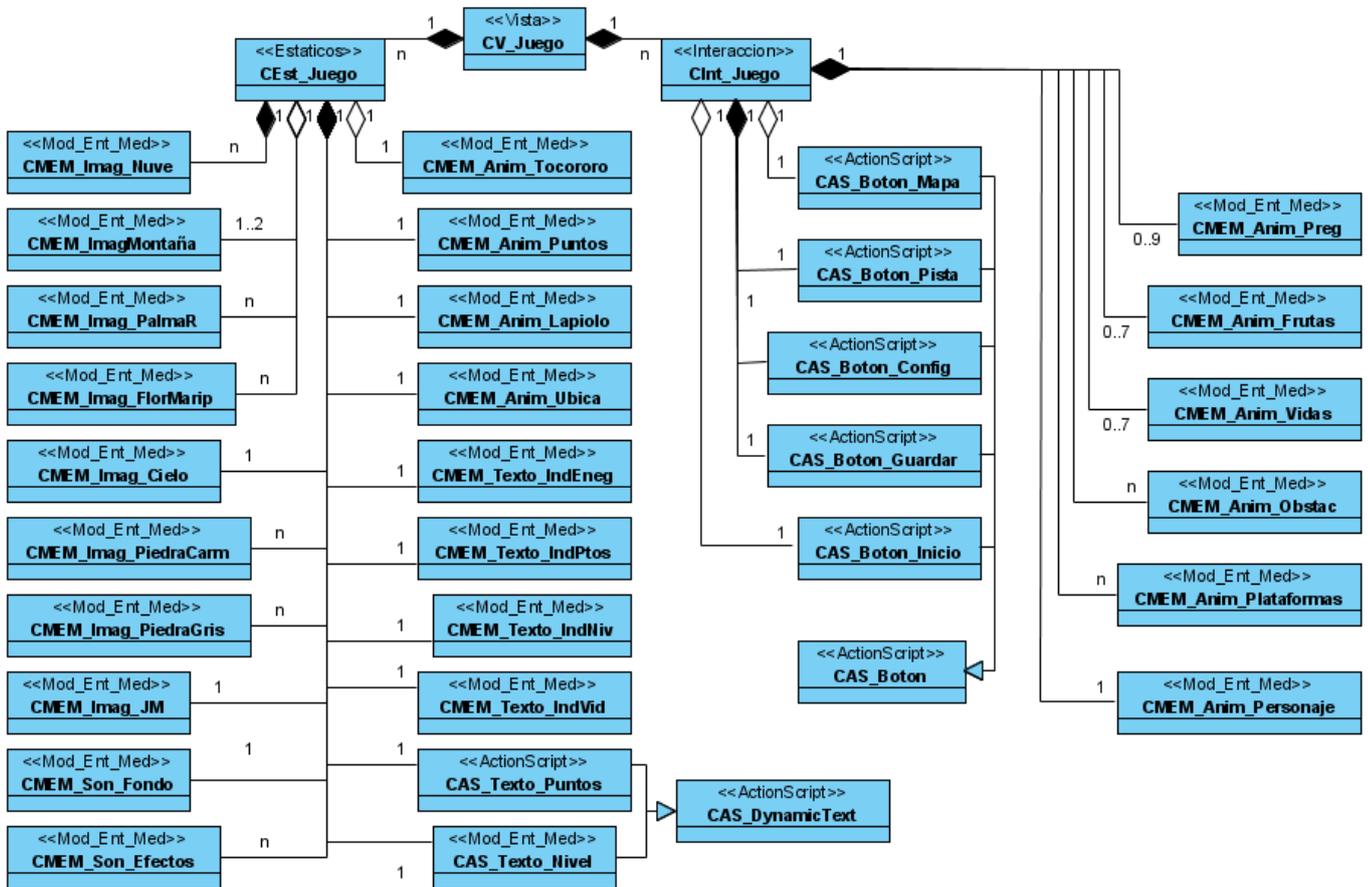


Figura 4. DEP Vista de Presentación Juego

Los Diagramas de Estructura de Presentación del resto de las vistas de la aplicación pueden ser consultados en los Anexos 22-37.

### 2.5.4 DIAGRAMAS DE ESTRUCTURA DE NAVEGACIÓN

Debido a la ineficiencia del lenguaje base UML para modelar la navegación en el software educativo se ha definido el Diagrama de Estructura de Navegación (DEN) como un artefacto de ApEM-L.

El DEN representa la estructura y organización de los elementos de navegación de sistemas multimedia incorporando nuevos estereotipos restrictivos que facilitan la comprensión del modelo como son: la clase botón, la clase consulta y la clase menú. Estos estereotipos son elementos de composición de la clase vista que permiten el enlace con otras clases vistas del modelo. (FEBE and HERRERA 2008)

Para lograr un mayor entendimiento y organización se propondrá realizar un DEN para cada subsistema. A continuación se muestran los DEN para cada subsistema identificado.

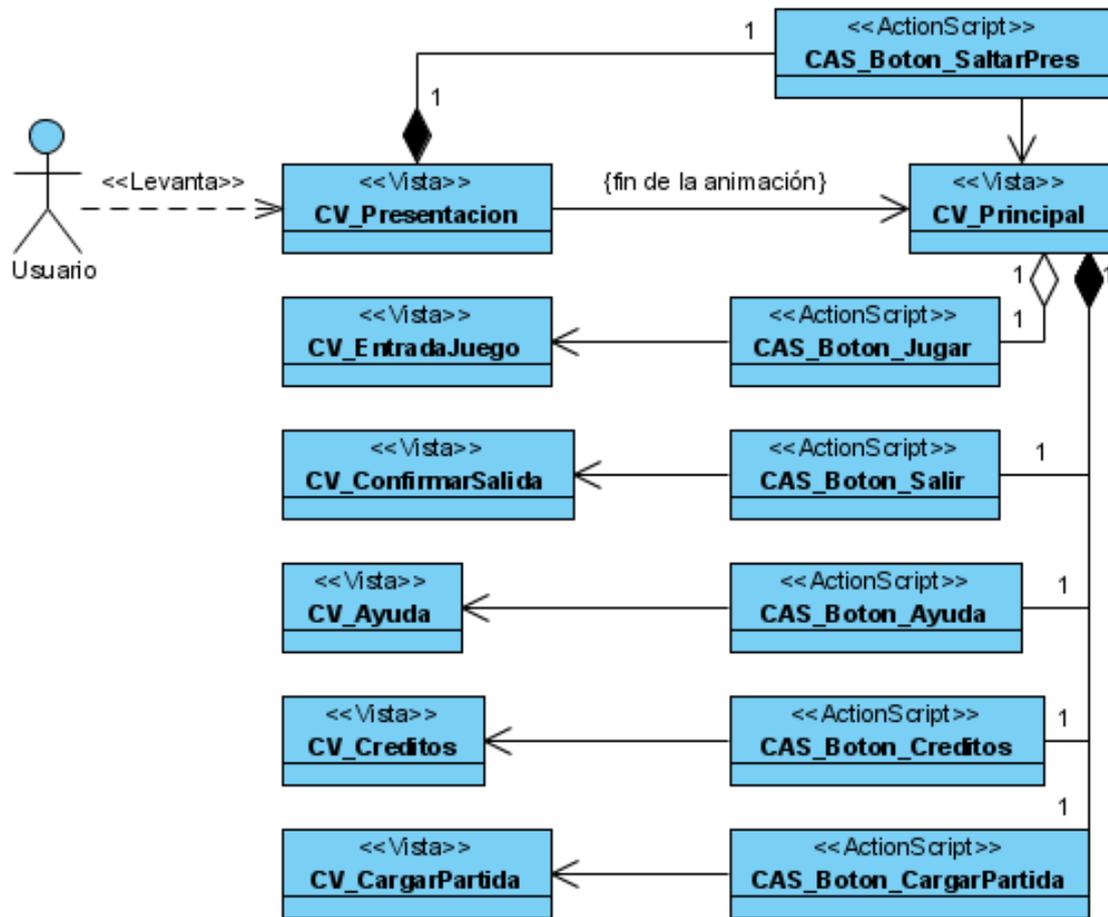


Figura 5. DEN Subsistema Principal

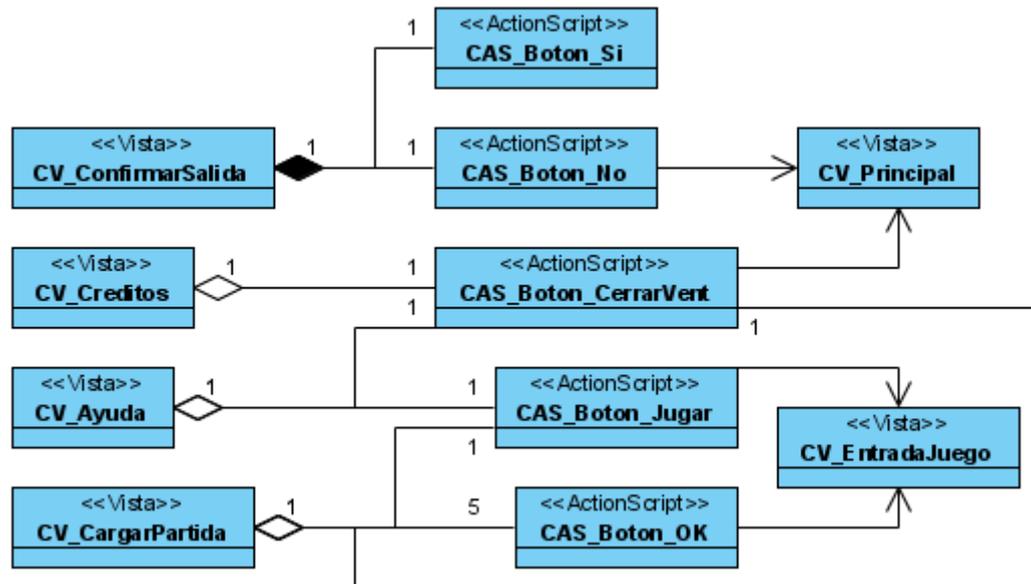


Figura 6. DEN Subsistema Menú

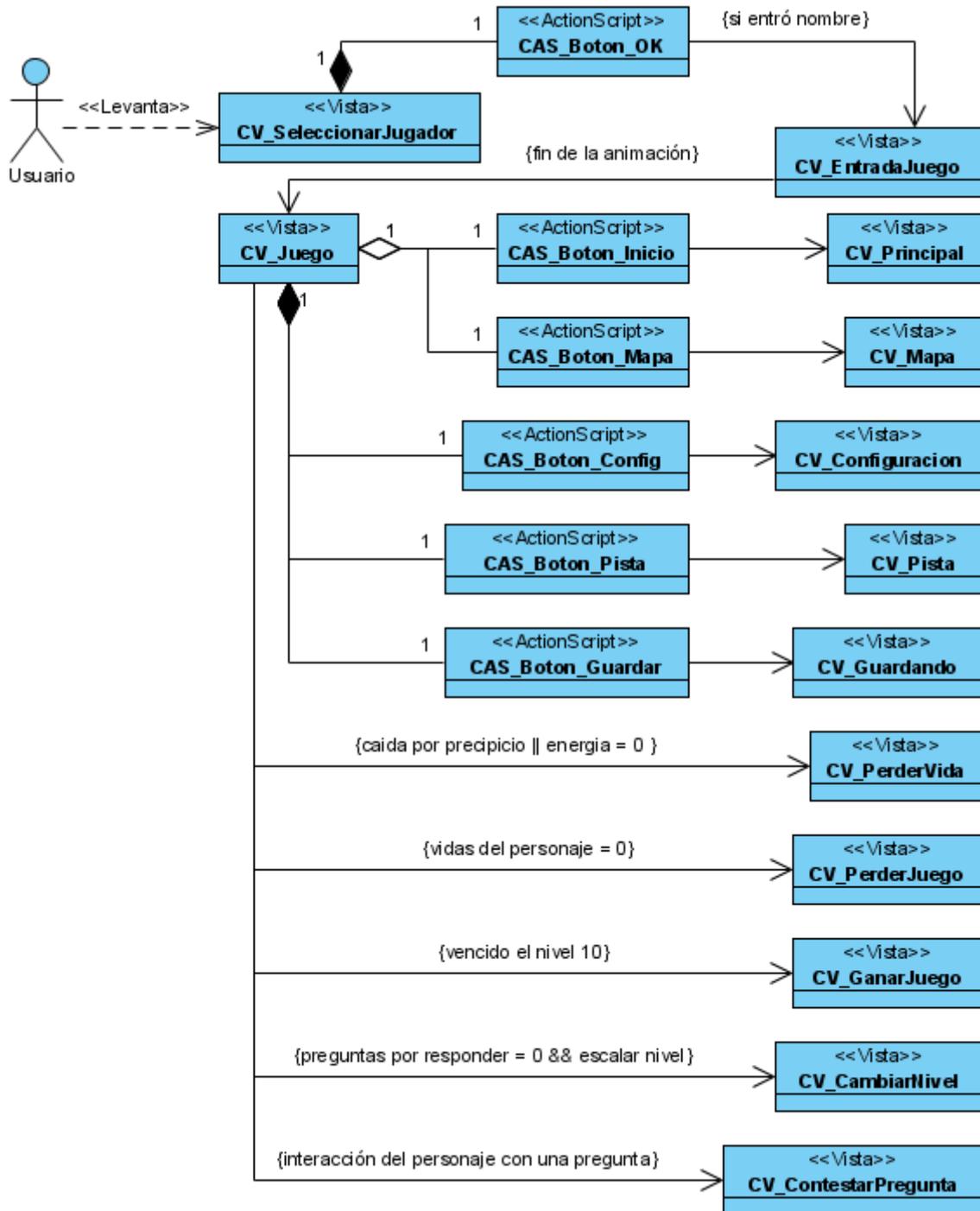


Figura 7. DEN Subsistema Juego

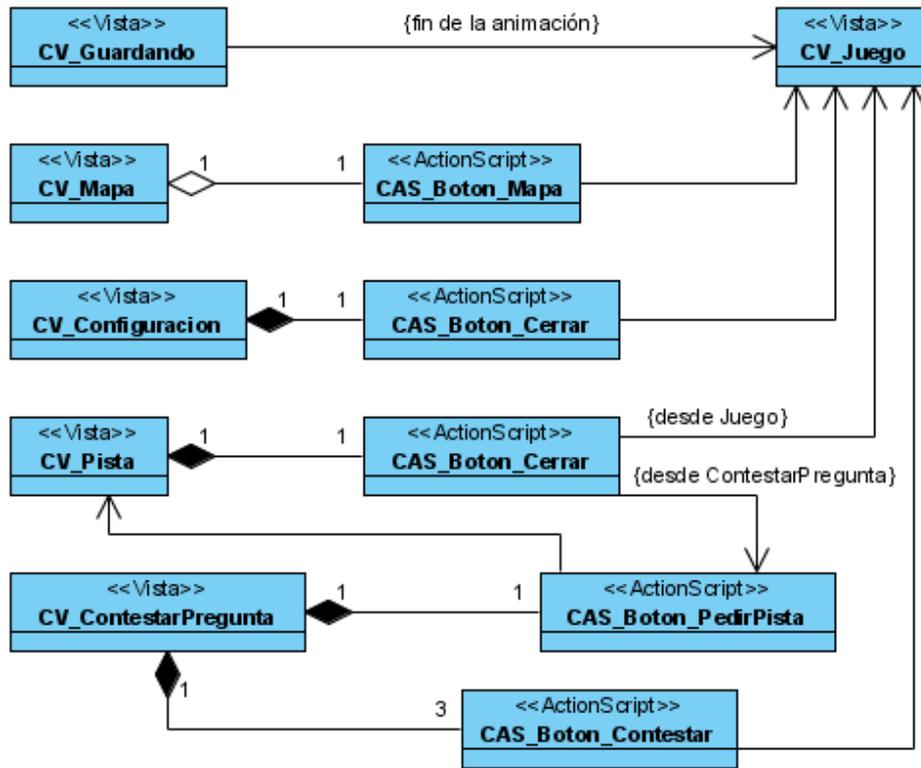


Figura 8. DEN Subsistema Apoyo

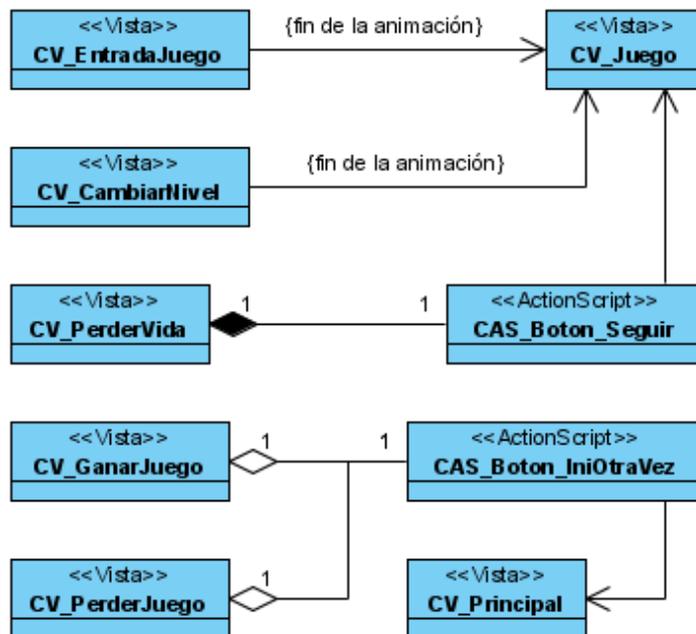


Figura 9. DEN Subsistema Transiciones

## **2.6 CONCLUSIONES DEL CAPÍTULO**

En este capítulo realizó la identificación de la audiencia y se describió la solución propuesta. Se identificaron y describieron los principales conceptos del entorno donde se usará el software, se realizó el modelamiento del dominio y se especificaron dieciocho requerimientos funcionales y veintiuno no funcionales que debe cumplir el software a desarrollar. Además se representó la Vista de Gestión del Modelo, compuesta por cinco subsistemas. También se hizo una descripción textual de dos de las principales Vistas de Presentación y se mostraron los Diagramas de Estructura de Presentación de estas vistas. Y se mostraron los Diagramas de Estructura de Navegación correspondientes a cada uno de los subsistemas identificados.

## CAPÍTULO 3. DISEÑO DE LA SOLUCIÓN PROPUESTA

En este capítulo se hace una descripción del patrón arquitectónico y de los patrones de diseño a utilizar. Además se presentan los artefactos fundamentales que guiarán el trabajo correspondiente a la implementación de la solución propuesta. Dentro de los que se muestran los diagramas de clases del diseño y los diagramas de secuencia, con ciertas modificaciones propuestas por el lenguaje de modelado utilizado para aplicaciones educativas multimedia: ApEM-L, y se presenta el diagrama de despliegue.

### 3.1 PATRÓN ARQUITECTÓNICO Y PATRONES DE DISEÑO UTILIZADOS

El patrón arquitectónico utilizado para diseñar el software educativo con tecnología multimedia Pico Maestro fue Modelo Vista Controlador Entidad (MVC-E). Con el cual se descarga las responsabilidades de la clase modelo concernientes a la gestión de la información persistente de la aplicación en las clases *Modelo-Entidad-Media* y *Modelo-Entidad-Persistente*.

Uno de los patrones de diseño utilizados fue el Experto, el que permitió que se asignaran responsabilidades a las clases que contaran con la información necesaria para cumplir tal responsabilidad. Se evidencia su aplicación en la clase *Personaje*, que como es la que cuenta con toda la información referente al personaje del juego pues es la responsable de hacer que el mismo camine, salte, gane vidas y energía, entre otras acciones capaz de realizar el personaje.

Otro patrón de diseño utilizado fue el Creador con el que se pudo determinar qué clase sería la responsable de crear alguna instancia de otra clase. Se aplica por ejemplo en la clase *Juego*, la que es responsable de crear una instancia de la clase *Personaje*, debido a que un personaje tendrá vida solo en el juego. Otro ejemplo del uso de este patrón está en las clases vistas, las que son responsables de crear las instancias de las clases del lenguaje de alto nivel ActionScript, como los clips de película, botones, textos dinámicos y otros objetos que componen la vista.

Y el otro patrón de diseño que se aplicó fue Singleton con el que se garantizó que solo se creara una instancia de la clase *Personaje* durante la ejecución de una partida en el juego, creando para este un punto de acceso global, garantizando que cuando se necesitara referirse al personaje de juego se tratara del mismo objeto en todo momento, debido a que en una partida solo interviene un personaje.

## 3.2 MODELO DEL DISEÑO

En el modelo de diseño es un modelo de objetos que se centra en cómo los requerimientos funcionales y no funcionales, junto con otras restricciones del entorno de implementación, tienen impacto en el sistema que se desarrolla.

El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. Entre sus principales propósitos están:

- Adquirir una comprensión de los aspectos relacionados con los requerimientos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requerimientos o subsistemas individuales, interfaces y clases.
- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable y sólida, y crear un plano del modelo de implementación.

### 3.2.1 DIAGRAMAS DE CLASES

El diagrama de clases es el modelado de determinados conceptos en forma de clases que describen un conjunto de objetos que almacenan información y se comunican para implementar su comportamiento. La información almacenada se representa como atributos de estas clases y las operaciones a través de los métodos de las mismas. Los diagramas de clase son un tipo de diagrama estático que constituyen el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer, como para mostrar cómo puede ser construido. Son los diagramas más comunes en el modelado de sistemas orientados a objetos.

ApEM-L propone algunas modificaciones para este tipo de diagrama, que consisten en dividirlo en dos grandes zonas, la de la izquierda dedicada al árbol jerárquico de las clases modelo entidad medias que representan los recursos mediáticos de la aplicación y en la zona de la derecha del diagrama las clases que controlan la lógica del negocio de la aplicación propiamente dicha. Esta zona de la derecha

vuelve a subdividirse en cuatro zonas. La primera dedicada a las clases vista, la contigua a esta y en el extremo superior derecho dedicada a las clases controladoras, inmediatamente debajo de esta sección, la destinada a las clases modelo, quedando una banda inferior derecha dedicada en su extremo derecho a las clases modelo entidad persistentes para el tratamiento de la información persistente de la aplicación; y en el extremo izquierdo las clases correspondientes al Lenguaje de Alto Nivel (HLL) con el que se programe. (FEBE 2007) (Consular Anexo 38)

Este tipo de diagrama se representa para cada una de las vistas del producto de software. Seguidamente se muestran los diagramas de clases de las vistas más importantes del software a construir.

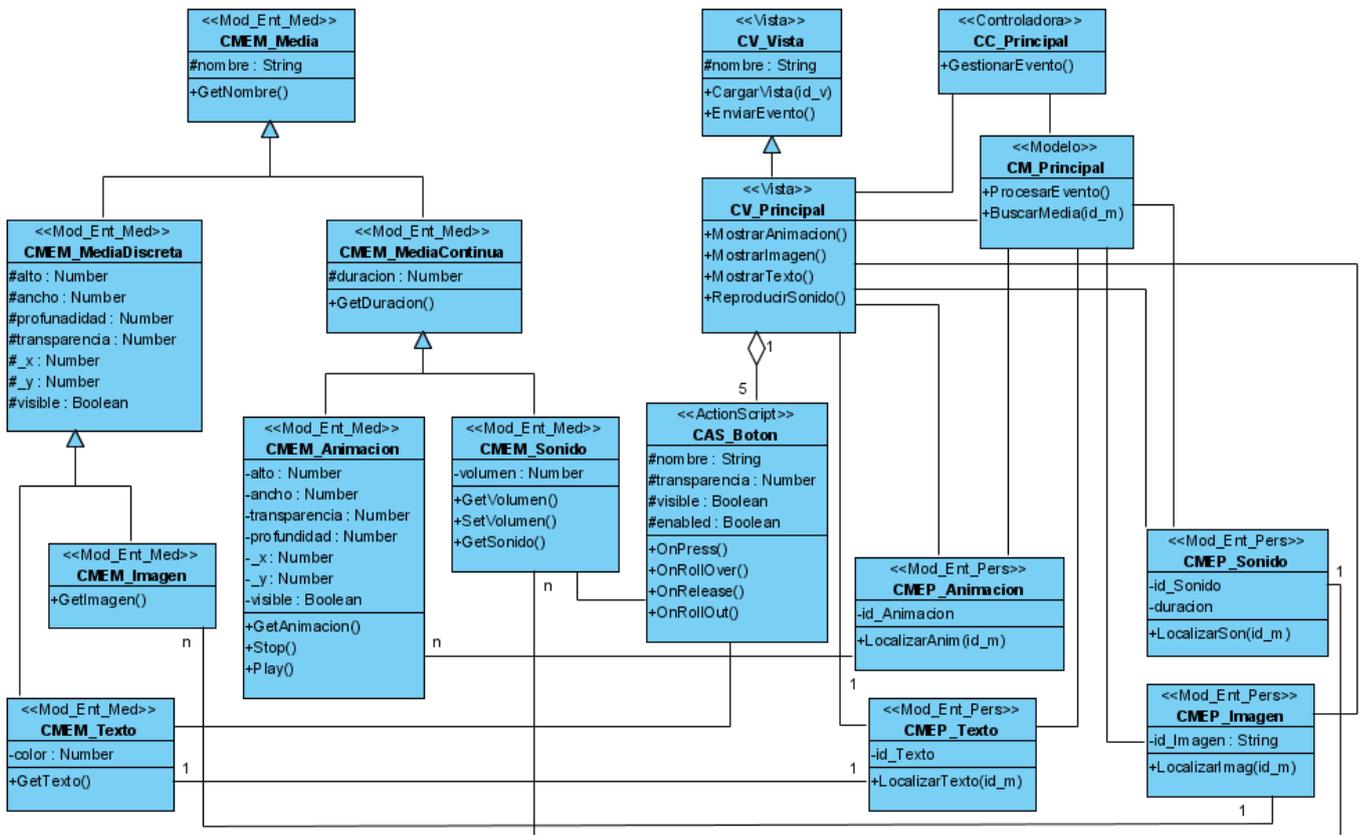


Figura 10. Diagrama de Clases de la Vista de Presentación Principal

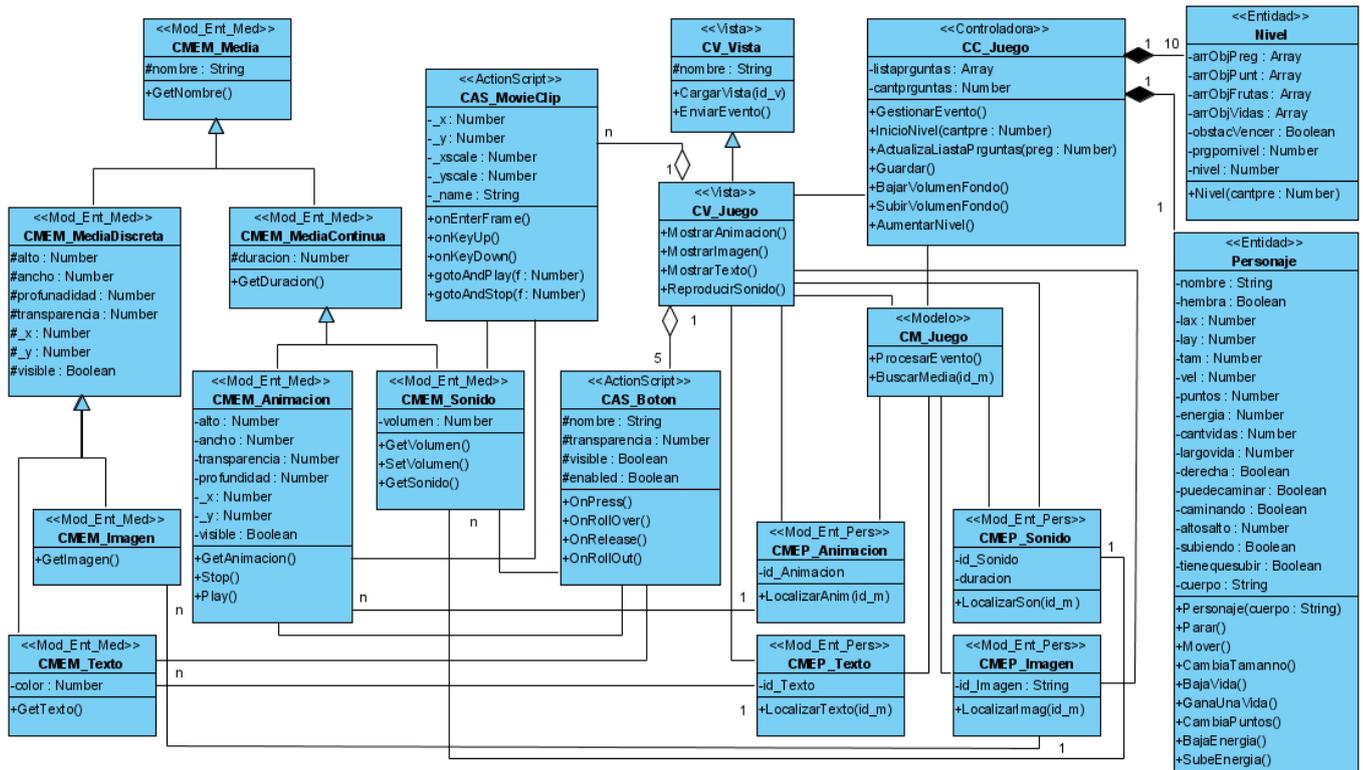


Figura 11. Diagrama de Clases de la Vista de Presentación Juego.

Los Diagramas de Clases del resto de las vistas de la aplicación pueden ser consultados en los Anexos 39-54.

### 3.2.2 DIAGRAMAS DE SECUENCIAS

Un Diagrama de Secuencias es un tipo de diagrama de interacción que muestra los objetos con sus interacciones en el tiempo representadas como mensajes.

ApEM-L propone una modificación para este tipo de diagrama, que consisten en la representación del tiempo cuando su incorporación sea necesaria, por la importancia del control de esta variable en las aplicaciones educativas y en especial en el trabajo con las medias continuas como sonido, video y animaciones. (FEBE 2007)

Este tipo de diagrama se representa para cada una de las vistas del producto de software. Seguidamente se muestran los diagramas de secuencias de las vistas más importantes del software a construir.



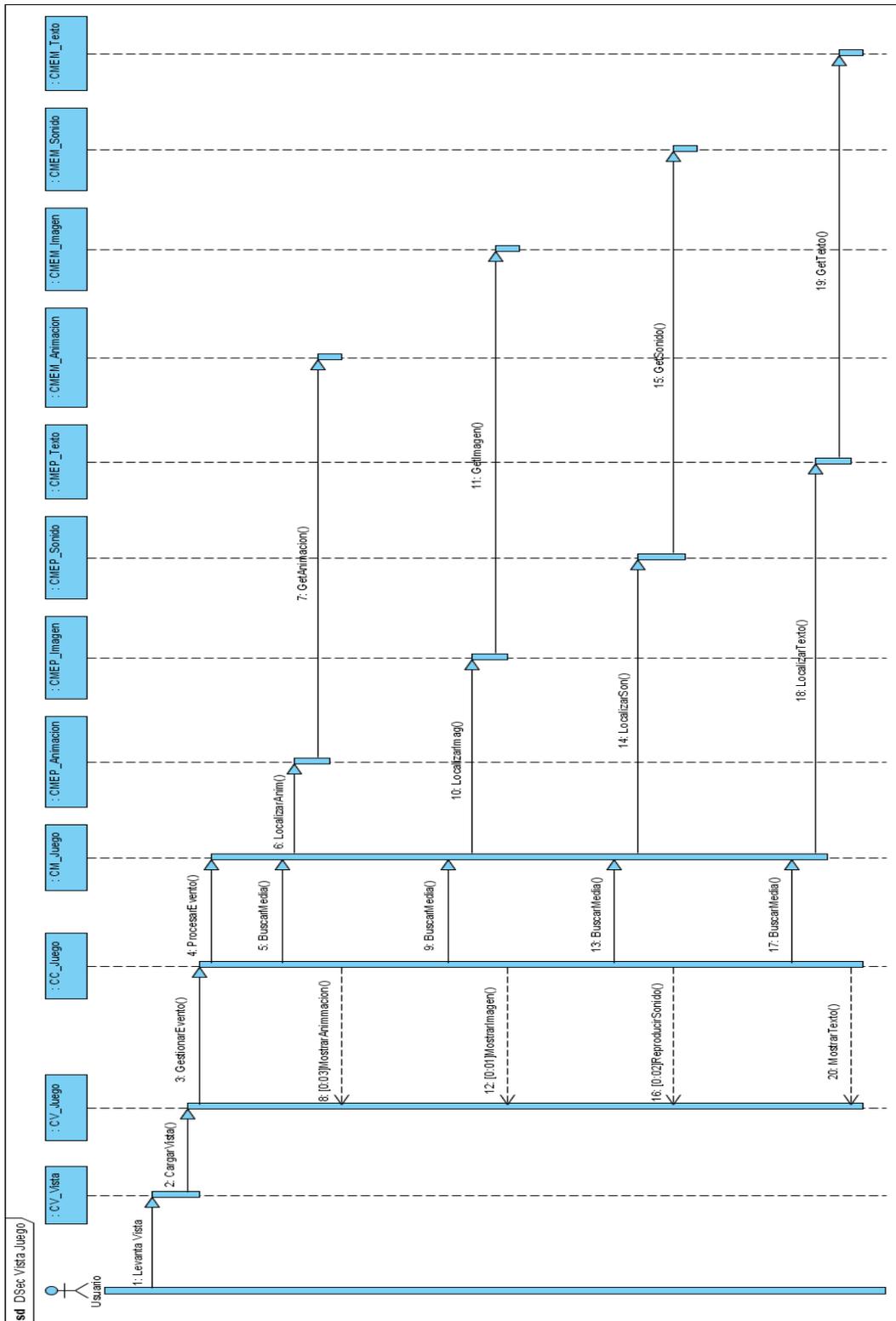


Figura 13. Diagrama de Secuencias de la Vista de Presentación Juego

Los Diagramas de Secuencias del resto de las vistas de la aplicación pueden ser consultados en los Anexos 55-70.

### 3.2.3 DIAGRAMA DE DESPLIEGUE

EL Diagrama de Despliegue describe la distribución física del sistema. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware.

Para el funcionamiento del software educativo con tecnología multimedia a desarrollar, solo se necesita de una computadora personal con las prestaciones mínimas de hardware expuestas como requerimientos no funcionales.

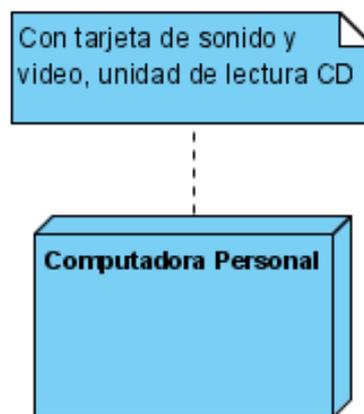


Figura 14. Diagrama de Despliegue

## 3.3 CONCLUSIONES DEL CAPÍTULO

En este capítulo se hizo una descripción del patrón arquitectónico y de los patrones de diseño utilizados para el diseño del software educativo con tecnología multimedia Pico Maestro. Se obtuvieron los artefactos fundamentales correspondientes al diseño de la solución propuesta, se representaron dos diagramas de clases del diseño de los dieciocho modelados y dos de los dieciocho diagramas de secuencia que se obtuvieron. Y finalmente se mostró el diagrama de despliegue de la aplicación.

## **CAPÍTULO 4. IMPLEMENTACIÓN DE LA SOLUCIÓN PROPUESTA**

En este capítulo se muestra el modelo de Implementación para la solución propuesta. Se presenta el diagrama de componentes. Se describen de los archivos XML que contendrán los datos persistentes y se muestran las pantallas principales de la aplicación.

### **4.1 MODELO DE IMPLEMENTACIÓN**

El Modelo de implementación representa la composición física de la implementación en términos de subsistemas de implementación y elementos de implementación. Describe cómo los elementos de diseño se implementan en componentes. (JACOBSON *et al.* 2000)

Se considera el artefacto más significativo del flujo de trabajo de Implementación, debido a la importancia que tiene para los desarrolladores comprender el funcionamiento del sistema desde el punto de vista de componentes y sus relaciones. Este modelo está conformado por el diagrama de componentes.

#### **4.1.1 DIAGRAMAS DE COMPONENTES**

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

ApEM-L extiende la semántica de UML para este tipo de diagrama al incorporar los elementos de organización en paquetes asociados al patrón arquitectónico MVC-E y sus relaciones de funcionamiento.

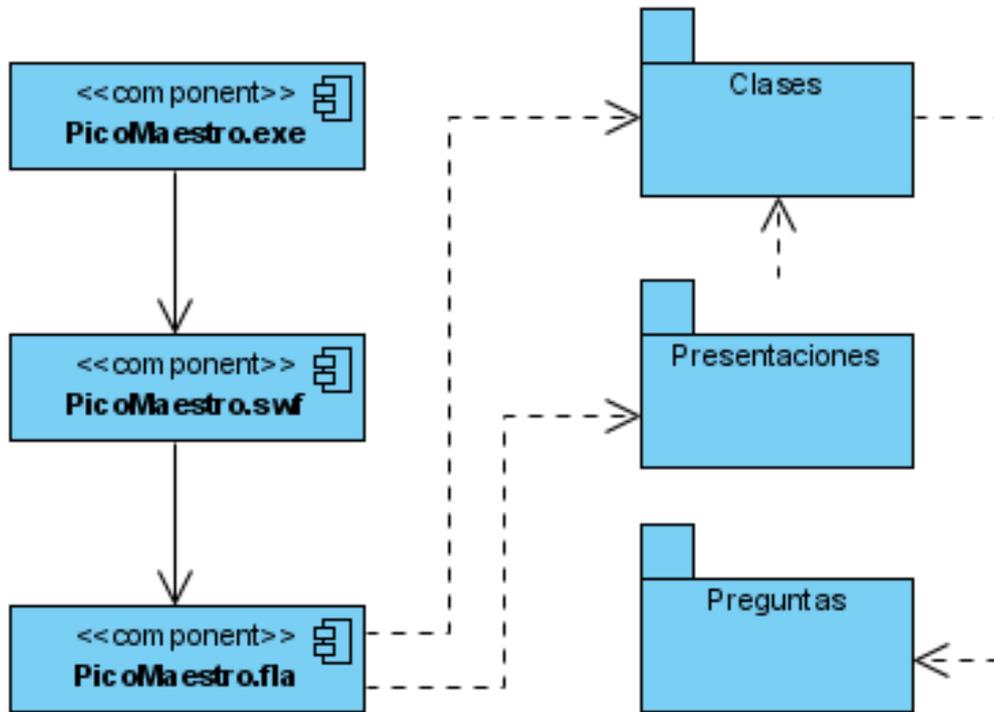


Figura 15. Diagrama General de Componentes

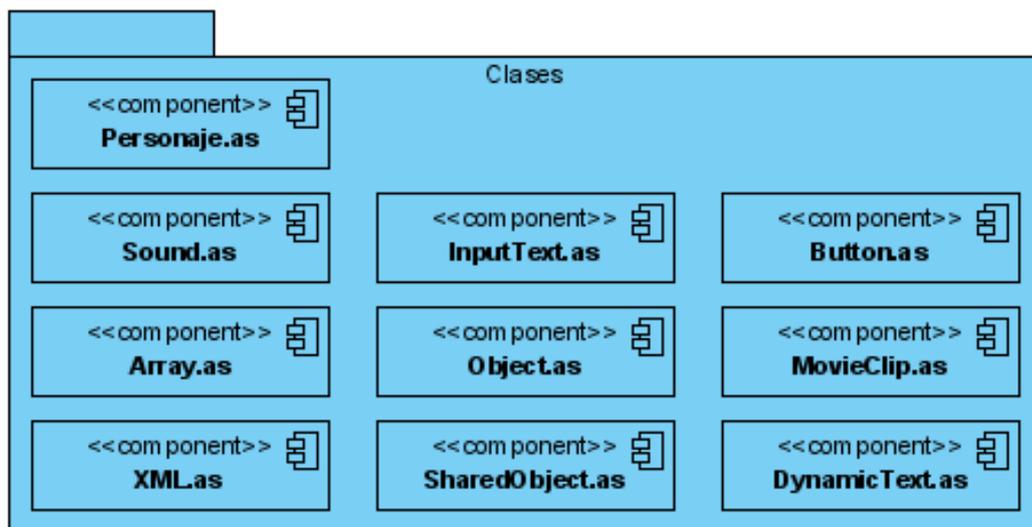


Figura 16. Diagrama de Componentes del Paquete Clases

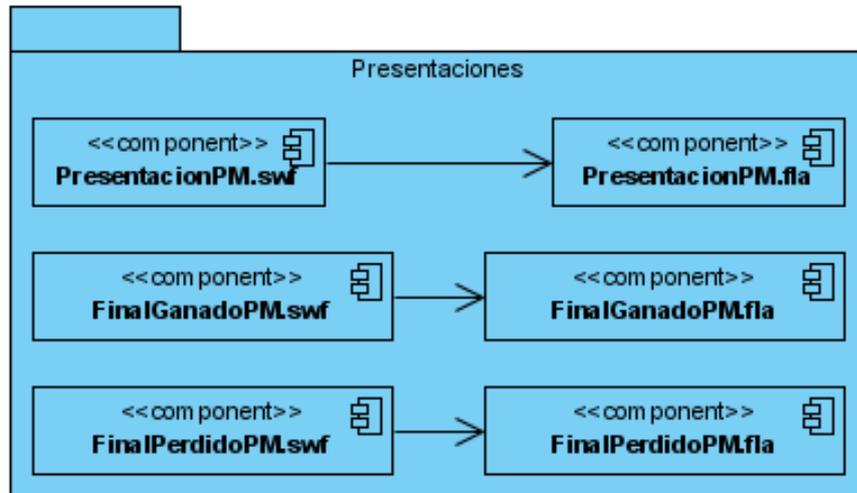


Figura 17. Diagrama de Componentes del Paquete Presentaciones

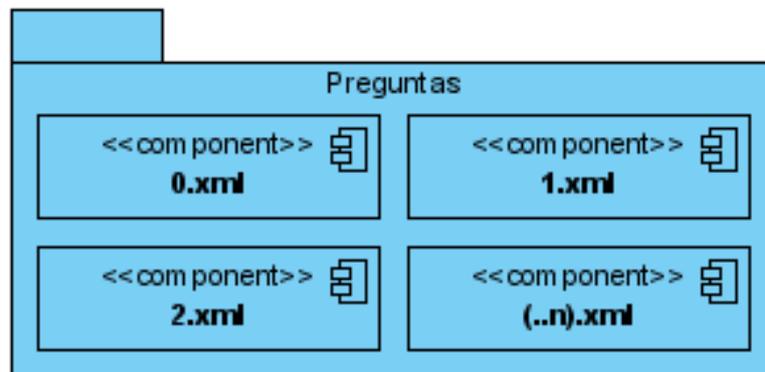


Figura 18. Diagrama de Componentes del Paquete Preguntas

## 4.2 DESCRIPCIÓN DE ARCHIVOS XML

XML es el lenguaje de marcas estándar para el intercambio de información entre aplicaciones, destinado exclusivamente a la gestión de archivos y contenidos formateados en este estándar. Es perfectamente soportado por ActionScript, permitiendo a las aplicaciones desarrolladas en Flash la gestión de este tipo de archivos (XML), que son archivos de texto en los que se almacena información que puede ser interpretada por determinada aplicación.

El contenido de cada una de las preguntas mostradas en el software educativo con tecnología multimedia Pico Maestro está almacenado en archivos XML independientes. Es decir, existe un archivo

XML por pregunta. El nombre de estos archivos tendrá el formato: (número).xml, permitiendo que se disponga un conjunto de preguntas cuyos nombres representen una secuencia numérica, donde el primero de ellos es "0.xml", esto es de gran importancia, debido a que las preguntas serán mostradas de manera aleatoria y se hace necesario que sus nombres lo permitan. Aunque se dispondrá de cientos de estos archivos la estructura de todos será la misma.

#### 4.2.1 ARCHIVO XML PREGUNTA

```
<pregunta>
  <textopregunta>Texto de la pregunta. No debe superar los 40 caracteres</textopregunta>
  <posibleresp1>Posible respuesta 1. No debe superar los 16 caracteres</posibleresp1>
  <posibleresp2>Posible respuesta 2. No debe superar los 16 caracteres</posibleresp2>
  <posibleresp3>Posible respuesta 3. No debe superar los 16 caracteres</posibleresp3>
  <verdaderaresp>número de la respuesta verdadera (1, 2 ó 3)</verdaderaresp>
  <valorpreg>valor de la pregunta (número entre 0 y 100)</valorpreg>
  <pista>Texto de una pista para responder la pregunta. No debe superar los 64 caracteres</pista>
</pregunta>
```

#### 4.2.2 ARCHIVO XML CANTIDADTOTALPREGUNTAS

Debido a que es posible incorporar todas las preguntas que sean necesarias dentro de la carpeta que contienen estos archivos, se requiere conocer cuántos son los disponibles para ser cargados por la aplicación. Esta información estará contenida en otro archivo nombrado "CantidadTotalPreguntas.xml", cuya estructura es diferente a los mostrados anteriormente.

```
<cantpreg>
  <cantpreguntas>número</cantpreguntas>
</cantpreg>
```

### 4.3 PANTALLAS PRINCIPALES

A continuación se muestran las principales pantallas mostradas en la aplicación:



Figura 19. Pantalla Principal



Figura 20. Selección del personaje



Figura 21. Primer nivel del juego



Figura 22. Respondiendo una pregunta



Figura 23. Sección de recuperación



Figura 24. Transición entre niveles

#### **4.4 CONCLUSIONES DEL CAPÍTULO**

En este capítulo se hizo una representación del diagrama de componentes para la implementación del software educativo con tecnología multimedia Pico Maestro y se describieron los archivos XML que contendrán los datos persistentes. También se presentaron las pantallas principales de la aplicación.

## CONCLUSIONES

Mediante el presente trabajo se ha realizado un estudio de las tendencias y tecnologías actuales para el desarrollo de software educativos con tecnología multimedia. Lo que permitió que se escogiera como metodología de desarrollo a RUP. Además, se seleccionó a Macromedia Flash 8 como herramienta de programación y diseño gráfico debido a las potencialidades que ofrece para este tipo de trabajo. La especificación de dieciocho requerimientos funcionales y veintiuno no funcionales identificados, permitió definir las funcionalidades de la aplicación. La utilización del estilo arquitectónico MVC-E y de los patrones de diseño Experto, Creador y Singleton, unido al uso de ApEM-L como lenguaje de modelado y asistido por Visual Paradigm como herramienta de modelado, permitió diseñar el software orientado a la tecnología multimedia, obteniéndose dieciocho diagramas de clases del diseño y dieciocho diagramas de secuencia. Una vez obtenido el diseño se pudo implementar el software educativo con tecnología multimedia Pico Maestro, el que mediante la actividad lúdica contribuye al estudio la vida y obra de José Martí en niños de entre 7 y 12 años de edad.

## RECOMENDACIONES

Teniendo en cuenta que el objetivo fundamental del software educativo con tecnología multimedia Pico Maestro es contribuir al estudio de la vida y obra de José Martí en niños de entre 7 y 12 años de edad mediante la actividad lúdica, se recomienda lo siguiente:

- Desplegar el software en todas las escuelas primarias y los Joven Club de Computación de Cuba.
- Realizar nuevas versiones del software donde intervengan personajes de los cuentos escritos por José Martí.
- Desarrollar otros software educativos que contribuyan con el estudio de la vida y obra de José Martí, e integrarlos al paquete de juegos educativos Martianitos.
- Incorporar en ApEM-L el uso de estereotipos decorativos para lograr mayor entendimiento de la modelación con este lenguaje.

## REFERENCIAS BIBLIOGRÁFICAS

- ANTÓN, M. G. D.; M. A. PÉREZ, *et al.* *PROPUESTA DE UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE EDUCATIVO BAJO UN ENFOQUE DE CALIDAD SISTÉMICA*. Caracas, Venezuela, Universidad Simón Bolívar (USB), 2002.
- BOOCH, G.; J. RUMBAUGH, *et al.* *El lenguaje unificado de modelado*, Addison Wesley, 2000. [Disponible en: <http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf>]
- BOTANA, L. M. S. *¿EVALUAMOS EL SOFTWARE EDUCATIVO?*, 2007.
- CURBELO, F. H.; R. B. GARCÍA, *et al.* *LA CLASE DE SOFTWARE EDUCATIVO EN LA ENSEÑANZA SECUNDARIA BASICA*
- CUBANA. *RESULTADOS DE UN PROYECTO EDUCATIVO.*, Universidad Pedagógica “José Martí”, 2007. [Disponible en: [http://www.ispcmw.rimed.cu/sitios/pedag2007/trabajo/simposio7/Hurtado\\_Fermin.pdf](http://www.ispcmw.rimed.cu/sitios/pedag2007/trabajo/simposio7/Hurtado_Fermin.pdf)]
- FEBE, A. *ApEM – L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI*. Ciudad de la Habana, Universidad de las Ciencias Informáticas, 2007. p.
- FEBE, A. *Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos.*, 2006.
- FEBE, A. and Y. HERRERA. *DISEÑO DE APLICACIONES EDUCATIVAS MULTIMEDIAS UTILIZANDO UNA NUEVA VERSIÓN DEL LENGUAJE DE MODELACIÓN ApEM – L*. INFORMÁTICAS, U. D. L. C. Ciudad de La Habana, 2008.
- GALLARDO, L. *Juguetes educativos: una elección acertada en Navidad*, 2008. [Disponible en: <http://www.aprendemas.com/Reportajes/P1.asp?Reportaje=1345>]
- HEREDIA, Y. P. *METODOLOGÍAS PARA EL DESARROLLO DE SOFTWARE EDUCATIVO: UN ESTUDIO COMPARATIVO*. Ciudad de La Habana,, Universidad de las Ciencias Informáticas, 2007. p.
- JACOBSON; BOOCH, *et al.* *El Proceso Unificado de Desarrollo*, 2000.
- JACOBSON, I.; G. BOOCH, *et al.* *El proceso unificado de desarrollo de software*. La Habana, Ed. Félix Varela, 2004. p.
- LORENTE, A. E.; Y. M. PÉREZ, *et al.* *PLANTILLA PARA EL MONTAJE DINÁMICO DE LOS PRODUCTOS DE LA COLECCIÓN MULTISABER*, 2006. [Disponible en: [http://www.informaticahabana.com/evento\\_virtual/files/MUL067.pdf](http://www.informaticahabana.com/evento_virtual/files/MUL067.pdf)]
- MARQUÉS, P. *VENTAJAS E INCONVENIENTES DEL MULTIMEDIA EDUCATIVO*, 2000. [Disponible en: <http://www.peremarques.net/ventajas.htm>]

- ORALLO, E. H. *El Lenguaje Unificado de Modelado (UML)*, 2001. [Disponible en: <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>]
- PEÑA, A. M. S. *Análisis y diseño del juego interactivo Energía para Aprender*. Ciudad de la Habana, Universidad de las Ciencias Informáticas, 2008. p.
- REYNOSO, C. B. *Introducción a la Arquitectura de Software*, UNIVERSIDAD DE BUENOS AIRES, 2004.
- SÁNCHEZ. *Construyendo y Aprendiendo con el Computador*. Santiago de Chile, Universidad de Chile, 1999. p.
- SAÚCO, A. M. *Análisis y diseño del juego interactivo Energía para Aprender*. Ciudad de la Habana, Universidad de las Ciencias Informáticas, 2008. p.
- TORRES, M. C. and H. M. ESPINO. *Colección Antonio Guiteras y Carlos Aponte: Seis libros para la formación de conciencia política*. Ciudad de la Habana, Universidad de las Ciencias Informáticas, 2008. p.
- VISPO, Y. *El desarrollo de valores morales a través de cuentos en el proceso de formación de profesores generales integrales* 2006. [Disponible en: [http://www.sepbcs.gob.mx/comunicacion/Noticias%20educacion/Noticias%202008/VALORES\\_MORALES.htm](http://www.sepbcs.gob.mx/comunicacion/Noticias%20educacion/Noticias%202008/VALORES_MORALES.htm)]

## BIBLIOGRAFÍA

Ayuda de Macromedia Flash 8.

BOOCH, G.; J. RUMBAUGH, *et al.* *El lenguaje unificado de modelado*, Addison Wesley, 2000.  
[Disponible en: <http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf>

CORREDERA, A. R. M. *SOFTWARE PARA FACILITAR EL TRATAMIENTO DE LA ÉTICA MARTIANA EN LA ENSEÑANZA TÉCNICA Y PROFESIONAL*. Disponible en:  
<http://rvarela.ispvc.rimed.cu/articulos/rv1805.pdf>

*Curso Online Programación con ActionScript 2.0*. Disponible en: [http://www.virtual-formac.com/informatica/programacion/curso\\_programacion\\_con\\_actionscript\\_2\\_0-c6408.html](http://www.virtual-formac.com/informatica/programacion/curso_programacion_con_actionscript_2_0-c6408.html).

FEBE, A. *ApEM – L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI*. Ciudad de la Habana, Universidad de las Ciencias Informáticas, 2007.

FEBE, A. *Utilización del Patrón Modelo – Vista – Controlador (MVC) en el diseño de software educativos.*, 2006.

FEBE, A. and Y. HERRERA. *DISEÑO DE APLICACIONES EDUCATIVAS MULTIMEDIAS UTILIZANDO UNA NUEVA VERSIÓN DEL LENGUAJE DE MODELACIÓN ApEM – L*. INFORMÁTICAS, U. D. L. C. Ciudad de La Habana, 2008.

HEREDIA, Y. P. *METODOLOGÍAS PARA EL DESARROLLO DE SOFTWARE EDUCATIVO: UN ESTUDIO COMPARATIVO*. Ciudad de La Habana,, Universidad de las Ciencias Informáticas, 2007.

<http://www.mujeres.cubaweb.cu/Marti/HTML/BIOGRAFIA.htm>.

JACOBSON; BOOCH, *et al.* *El Proceso Unificado de Desarrollo*, 2000.

JACOBSON, I.; G. BOOCH, *et al.* *El proceso unificado de desarrollo de software*. La Habana, Ed. Félix Varela, 2004.

LORENTE, A. E.; Y. M. PÉREZ, *et al.* *PLANTILLA PARA EL MONTAJE DINÁMICO DE LOS PRODUCTOS DE LA COLECCIÓN MULTISABER*, 2006. [Disponible en:  
[http://www.informaticahabana.com/evento\\_virtual/files/MUL067.pdf](http://www.informaticahabana.com/evento_virtual/files/MUL067.pdf)

*Portal José Martí*. Oficina del Programa Martiano, 2008. [Disponible en:  
<http://www.josemarti.cu/?q=lcursos&nid=1585>

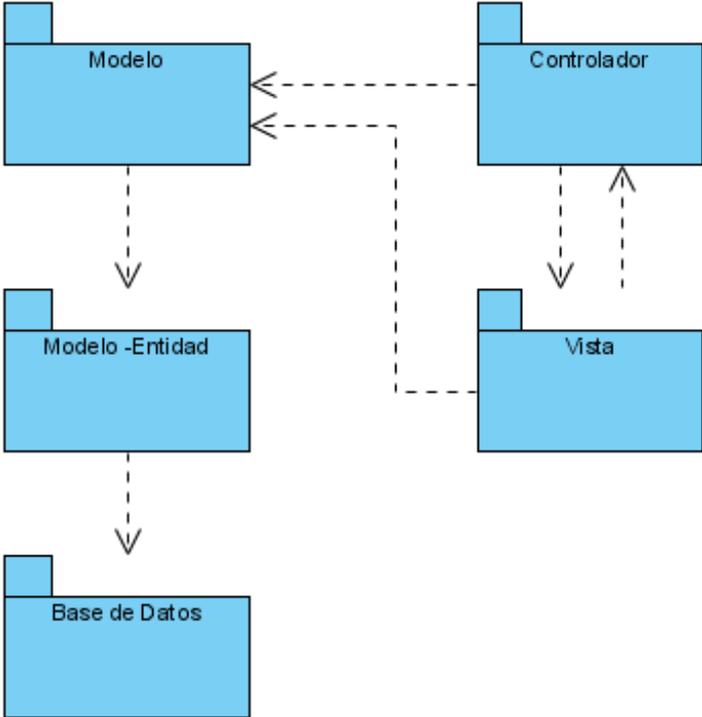
QUESADA, M., GONZALO. *Así fue Martí*. La Habana, Editorial Gente Nueva, 1976. p.  
RODRÍGUEZ, L. A. *SOFTWARE EDUCATIVO. HACIA UNA NUEVA PEDAGOGÍA BASADA EN LAS TICS*. Disponible en: <http://rvarela.ispvc.rimed.cu/articulos/rv1801.pdf>

*Sitio Web Feature Driven Development.* Disponible en: <http://www.featuredrivendevelopment.com/>

*Sitio Web La Demajagua.* Disponible en: <http://www.lademajagua.co.cu/>

*Tutorial UML 2.* Disponible en: [http://www.sparxsystems.com.ar/resources/uml2\\_tutorial](http://www.sparxsystems.com.ar/resources/uml2_tutorial)

# ANEXOS



Anexo 1. Patrón Modelo Vista Controlador Entidad



Anexo 2. Bases de ApEM-L

**Anexo 3.** Comparación de UML con ApEM-L en sus vistas estática y de arquitectura

<b>Aspecto o Elemento</b>	<b>Lenguaje Unificado de Modelado (UML)</b>	<b>Lenguaje para la Modelación de Aplicaciones Educativas (ApEM-L)</b>
Diagrama de clases	<ul style="list-style-type: none"> <li>• No establece un patrón arquitectónico que rija la concepción del diseño.</li> <li>• No contiene la semántica de tratamiento de las clases asociadas a las tecnologías multimedia e hipermedia.</li> </ul>	<ul style="list-style-type: none"> <li>• Establece el patrón arquitectónico MVC-E para la concepción del diseño de las aplicaciones educativas.</li> <li>• Plantea la semántica y los estereotipos restrictivos y descriptivos para las clases asociadas a las tecnologías multimedia e hipermedia.</li> <li>• Organiza la estructura del diagrama en secciones para la representación lógica de los distintos tipos de clases, incorporando las clases abstractas del modelo conceptual.</li> </ul>
Diagrama de casos de uso	• No se establecen modificaciones a lo planteado por el lenguaje base UML.	
Diagrama de componentes	• No se modifica la semántica del lenguaje base para este tipo de diagrama, sino que se extiende esta al incorporar los elementos de organización en paquetes asociados al patrón arquitectónico MVC-E y sus relaciones de funcionamiento.	
Diagrama de despliegue	• No se establecen modificaciones a lo planteado por el lenguaje base UML.	

**Anexo 4.** Comparación de UML con ApEM-L en su vista de comportamiento

<b>Aspecto o Elemento</b>	<b>Lenguaje Unificado de Modelado (UML)</b>	<b>Lenguaje para la Modelación de Aplicaciones Educativas (ApEM-L)</b>
Diagrama de	No se modifica la semántica del lenguaje base para este tipo de diagrama, sino que se extiende esta al incorporar el estereotipo restrictivo asociado al tiempo y	

actividad	su representación siempre que sea necesario enriqueciendo la descripción funcional de este tipo de aplicaciones.
Diagrama de secuencia	• No se establecen modificaciones a lo planteado por el lenguaje base UML.
Diagrama de colaboración	• No se establecen modificaciones a lo planteado por el lenguaje base UML.
Diagrama de estado	• No se establecen modificaciones a lo planteado por el lenguaje base UML.

**Anexo 5.** Comparación de UML con ApEM-L en su vista de presentación

<b>Aspecto o Elemento</b>	<b>Lenguaje Unificado de Modelado (UML)</b>	<b>Lenguaje para la Modelación de Aplicaciones Educativas (ApEM-L)</b>
Diagrama de estructura de navegación	• No existe este diagrama, lo que dificulta la representación de la estructura o mapa de navegación dentro de la aplicación.	• Extiende la semántica de los diagramas de clases para poder utilizarlos en este tipo de representación, incorporando los estereotipos restrictivos de clases: menú, índice, consulta y botón, además de utilizar las ya definidas: clases modelo entidad media texto y modelo entidad media imagen.
Diagrama de estructura de presentación	• No existe este diagrama, lo que dificulta la representación de la estructura o elementos de composición visual de las interfaces de comunicación con el usuario, o lo que es lo mismo las clases vista.	• Extiende la semántica de los diagramas de clases para poder utilizarlos en la representación de la estructura de las presentaciones, incorporando los estereotipos restrictivos de clases: estáticos e interacción y un árbol jerárquico a partir de estos estereotipos que agrupa los componentes visuales de acuerdo a su función en la interfaz a la que pertenece.

**Anexo 6.** Descripción textual de la Vista de Presentación Presentacion

<b>Descripción Textual de la Vista de Presentación <i>Presentacion</i></b>		
<b>Actores de la Vista de Presentación</b>	Usuario	
<b>Propósito</b>	Mostrar al usuario una presentación del juego Pico Maestro.	
<b>Objetivos pedagógicos</b>	Mostrar al usuario elementos históricos sobre el busto de José Martí que se encuentra en el Pico Turquino.	
<b>Resumen</b>	La vista se inicia cuando el usuario levanta la aplicación, se muestran una animación que incluye el nombre del juego y narra la historia del juego.	
<b>Vistas asociadas</b>	Principal.	
<b>Referencias</b>	R1	
<b>Precondiciones</b>	Se comprueba la existencia de los ficheros XML.	
<b>Poscondiciones</b>	De no existir los ficheros XML se muestra un mensaje de error y se cierra la aplicación.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
1. El usuario levanta la aplicación.  3. Saltar presentación.	2. Se muestra la animación A001 que presenta al juego.  4. Se muestra la Vista Principal.	2.1. La animación durará 68 segundos. La animación incluye imágenes y videos del Pico Turquino sincronizados con animaciones de los personajes del juego así como con una historia narrada.
<b>Cursos Alternos de los Eventos</b>		
<b>Acción</b>	<b>Curso Alterno</b>	
3.1. Finaliza el tiempo de la animación	3.1.a. Se muestra la Vista Principal.	
<b>Prioridad</b>	Opcional	
<b>Mejoras</b>	-	

<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>
<b>Animación</b>	A001	Animación de presentación del juego Pico Maestro. Narra la historia que trata el juego y muestra imágenes del Pico Turquino y de los personajes del juego invitando al usuario a jugar.	Existente.
<b>Sonido</b>	S021	Sonido muy breve para cuando se pase el cursor sobre un botón.	Existente.
	S022	Sonido muy breve para cuando se haga clic sobre un botón.	Existente.
<b>Reglas pedagógicas</b>			
-			

#### Anexo 7. Descripción textual de la Vista de Presentación Ayuda

<b>Descripción Textual de la Vista de Presentación Ayuda</b>	
<b>Actores de la Vista de Presentación</b>	Usuario
<b>Propósito</b>	Brindar la información necesaria al usuario sobre el funcionamiento y control del juego.
<b>Objetivos pedagógicos</b>	Permitir al usuario dominar el trabajo con la aplicación.
<b>Resumen</b>	La vista se inicia cuando el usuario selecciona la opción Ayuda perteneciente a la vista principal. Termina cuando el usuario desee volver a la pantalla principal o cuando el usuario opte por empezar a jugar. Da la oportunidad al usuario empezar un juego nuevo.
<b>Vistas asociadas</b>	Principal, SeleccionarJugador.
<b>Referencias</b>	R3
<b>Precondiciones</b>	-
<b>Poscondiciones</b>	Se muestra la vista Ayuda.
<b>Curso Normal de los Eventos</b>	

Acciones del Actor	Respuesta del Sistema	Elementos de la Vista
<p>1. El usuario hace clic sobre el botón Ayuda de la Vista Principal.</p> <p>3. El usuario cierra la ventana de contenido de la Ayuda.</p> <p>5. El usuario hace clic sobre el botón Jugar.</p>	<p>2. Se muestra la vista Ayuda.</p> <p>4. Se muestra la Vista Principal.</p> <p>6. Se muestra la Vista SeleccionarJugador.</p>	<p>2.1. Todos los elementos de la vista Principal se tornan oscuros y se bloquea la interacción del usuario con los mismos. Hace entrada desde la izquierda el contenido de la ayuda.</p> <p>4.1. El contenido hace su salida hacia la derecha. Todos los elementos de la vista Principal retoman su color y se permite la interacción del usuario con los mismos.</p>

**Cursos Alternos de los Eventos**

Acción	Curso Alternativo
-	-
Prioridad	Crítica
Mejoras	-

**Medias a utilizar**

Tipo de Media	Nombre	Descripción	Estado
Imagen	I006	Imagen sobre la que mostrará el contenido.	Existente.
Animación	A007	Animación de las teclas direccionales de teclado y del mouse.	Existente.
	S021	Sonido muy breve para cuando se pase el cursor sobre un botón.	Existente.
	S022	Sonido muy breve para cuando se haga clic sobre un botón.	Existente.
Texto	T002	Nombre de la Ventana	Existente.
	T003	Texto de la ayuda	Existente.

Reglas pedagógicas	
-	

**Anexo 8.** Descripción textual de la Vista de Presentación Creditos

Descripción Textual de la Vista de Presentación <i>Creditos</i>		
<b>Actores de la Vista de Presentación</b>	Usuario	
<b>Propósito</b>	Mostrar al usuario los datos de contactos del equipo de desarrollo.	
<b>Objetivos pedagógicos</b>	-	
<b>Resumen</b>	La vista se inicia cuando el usuario selecciona la opción Créditos perteneciente a la vista principal. Termina cuando el usuario desee volver a la pantalla principal.	
<b>Vistas asociadas</b>	Principal.	
<b>Referencias</b>	R4	
<b>Precondiciones</b>	-	
<b>Poscondiciones</b>	Se muestra la vista Creditos.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	Elementos de la Vista
1. El usuario hace clic sobre el botón Créditos de la Vista Principal. 3. El usuario cierra la ventana de contenido de los Créditos.	2. Se muestra la Vista Creditos. 4. Se muestra la Vista Principal.	2.1. Todos los elementos de la vista Principal se tornan oscuros y se bloquea la interacción del usuario con los mismos. Hace entrada desde la izquierda el contenido de los créditos. 4.1. El contenido hace su salida hacia la derecha. Todos los elementos de la vista Principal retoman su color y se permite la interacción del

		usuario con los mismos.
<b>Cursos Alternos de los Eventos</b>		
<b>Acción</b>	<b>Curso Alterno</b>	
-	-	
<b>Prioridad</b>	Opcional	
<b>Mejoras</b>	-	
<b>Medias a utilizar</b>		
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>
<b>Imagen</b>	I007	Imagen sobre la que mostrará el contenido.
	I008	Logotipo del productor.
<b>Sonido</b>	S021	Sonido muy breve para cuando se pase el cursor sobre un botón.
	S022	Sonido muy breve para cuando se haga clic sobre un botón.
<b>Texto</b>	T004	Nombre de la Ventana
	T005	Nombre de las personas implicadas en el desarrollo de la aplicación. Datos de contactos.
<b>Reglas pedagógicas</b>		
-		

**Anexo 9.** Descripción textual de la Vista de Presentación CargarPartida

<b>Descripción Textual de la Vista de Presentación <i>CargarPartida</i></b>	
<b>Actores de la Vista de Presentación</b>	Usuario
<b>Propósito</b>	Permitir al usuario de la aplicación cargar una partida que se haya guardado con anterioridad en el disco duro de la computadora.
<b>Objetivos pedagógicos</b>	-
<b>Resumen</b>	La vista se inicia cuando el usuario selecciona la opción Cargar Partida perteneciente a la vista principal. Muestra

	una lista con el nombre de las últimas cinco partidas guardadas y permite retomar una de estas partida para seguir jugándola. Termina cuando el usuario desee volver a la pantalla principal.	
<b>Vistas asociadas</b>	Principal, EntradaJuego, SeleccionarJugador.	
<b>Referencias</b>	R5	
<b>Precondiciones</b>	-	
<b>Poscondiciones</b>	Se muestra la vista CargarPartida.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
<p>1. El usuario hace clic sobre el botón Cargar Partida de la Vista Principal.</p> <p>3. El usuario selecciona la jugada que desea cargar.</p> <p>5. El usuario hace clic sobre el botón Jugar.</p> <p>7. El usuario cierra la ventana de las partidas guardadas.</p>	<p>2. Se muestra la vista CargarPartida.</p> <p>4. Se muestra la vista EntradaJuego.</p> <p>6. Se muestra la Vista SeleccionarJugador.</p> <p>8. Se muestra la Vista Principal.</p>	<p>2.1. Todos los elementos de la vista Principal se tornan oscuros y se bloquea la interacción del usuario con los mismos. Hace entrada desde la izquierda la tableta que contiene el nombre de las jugadas guardadas y entra con ella la mascota.</p> <p>8.1. La tableta que contiene el nombre de las jugadas guardadas hace su salida hacia la derecha. La mascota se retira con ella. Todos los elementos de la vista Principal retoman su color y se permite la interacción del usuario con los mismos.</p>
<b>Cursos Alternos de los Eventos</b>		
<b>Acción</b>	<b>Curso Alterno</b>	
3.1. No existen jugadas	3.1.a. Se muestra un texto que informa que no existen jugadas	

guardadas.	guardadas. Se muestra el botón Jugar. Va 5 o a 7.		
<b>Prioridad</b>	Crítica		
<b>Mejoras</b>	-		
Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
<b>Imagen</b>	I009	Imagen sobre la que mostrará el contenido.	Existente.
<b>Animación</b>	A008	Animación donde se muestra la mascota señalando el contenido de la tableta.	Existente.
<b>Sonido</b>	S021	Sonido muy breve para cuando se pase el cursor sobre un botón.	Existente.
	S022	Sonido muy breve para cuando se haga clic sobre un botón.	Existente.
<b>Texto</b>	T006	Nombre de la Ventana	Existente.
	T007	Nombre de las partidas guardadas	Existente.
Reglas pedagógicas			
-			

#### Anexo 10. Descripción textual de la Vista de Presentación ConfirmarSalida

Descripción Textual de la Vista de Presentación <i>ConfirmarSalida</i>	
<b>Actores de la Vista de Presentación</b>	Usuario
<b>Propósito</b>	Permitir al usuario confirmar la salida de la aplicación.
<b>Objetivos pedagógicos</b>	-
<b>Resumen</b>	La vista se inicia cuando el usuario hace clic en el botón Salir de la vista Principal.
<b>Vistas asociadas</b>	Principal.
<b>Referencias</b>	R18
<b>Precondiciones</b>	-
<b>Poscondiciones</b>	Se muestra la vista ConfirmarSalida.

<b>Curso Normal de los Eventos</b>			
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
1. El usuario hace clic en el botón Salir de la vista Principal. 3. El usuario hace clic en el botón Sí.		2. Se muestra la vista ConfirmarSalida.  4. Se cierra la aplicación.	2.1. Se muestra una pequeña ventana sobre el botón Salir de la vista Principal con un texto para verificar la voluntad de usuario de salir de la aplicación, con dos opciones: Sí (para salir) y No (para cerrar).
<b>Cursos Alternos de los Eventos</b>			
<b>Acción</b>		<b>Curso Alterno</b>	
3.1 El usuario hace clic en el botón No.		3.1.a. Se cierra la vista, se retorna a la vista Principal.	
<b>Prioridad</b>		Auxiliar	
<b>Mejoras</b>		-	
<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>
<b>Imagen</b>	I028	Imagen de fondo de la vista.	Existente.
<b>Video o Animación</b>	A066	Animación donde aparece una flanja que se sobrepone sobre una respuesta incorrecta.	Existente.
<b>Sonido</b>	S021	Sonido muy breve para cuando se pase el cursor sobre un botón.	Existente.
	S021	Sonido muy breve para cuando se haga clic sobre un botón.	Existente.
<b>Texto</b>	T024	Texto que solicita la voluntad del usuario de salir de la aplicación.	Existente.
<b>Reglas pedagógicas</b>			
-			

**Anexo 11.** Descripción textual de la Vista de Presentación SeleccionarJugador

<b>Descripción Textual de la Vista de Presentación <i>SeleccionarJugador</i></b>		
<b>Actores de la Vista de Presentación</b>	Usuario	
<b>Propósito</b>	Permitir al usuario de la aplicación seleccionar el personaje con el que desea jugar y ponerle un nombre.	
<b>Objetivos pedagógicos</b>	Hacer que el usuario se identifique con un personaje.	
<b>Resumen</b>	La vista muestra los personajes (hembra o varón) posibles a seleccionar por el usuario. Y posibilita que el mismo ponga un nombre al personaje seleccionado, este nombre se usará también como nombre para la partida.	
<b>Vistas asociadas</b>	Principal, EntradaJuego.	
<b>Referencias</b>	R6	
<b>Precondiciones</b>	-	
<b>Poscondiciones</b>	Se muestra la vista SeleccionarJugador.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
<p>1. El usuario hace clic sobre el botón Jugar de la Vista Principal.</p> <p>3. El usuario selecciona un personaje.</p> <p>5. El usuario acepta después de haber seleccionado un personaje y haberle puesto un nombre.</p>	<p>2. Se muestra la vista SeleccionarJugador.</p> <p>4. Se muestra el personaje seleccionado.</p> <p>6. Se retiran los elementos de la vista.</p> <p>7. Se muestra la vista EntradaJuego.</p>	<p>2.1. Entrada le los objetos de la vista con efecto de transparencia. Estas entradas estarán basadas en movimientos de desaceleración. Las ubicaciones serán:</p> <p>2.1.1. Ambos personaje en el centro de la pantalla.</p> <p>2.1.2. Debajo de los personajes estará el campo de entrada del nombre. A la misma altura pero a la derecha, un botón para aceptar.</p> <p>4.1. Se muestra una animación donde se entrecruzan los personajes y se queda solo el seleccionado.</p> <p>6.1. El personaje ascenderá hasta desaparecer de la pantalla, el campo de</p>

		entra de texto se retirará por la izquierda y el botón aceptar lo hará hacia adentro.	
<b>Cursos Alternos de los Eventos</b>			
<b>Acción</b>		<b>Curso Alterno</b>	
5.1. El usuario acepta sin poner un nombre al personaje. 5.2. El usuario acepta sin haber seleccionado un personaje.	5.1.a. Se muestra una animación sobre el campo de entrada del nombre que indica la necesidad de llenar este campo. 5.2.a. Se toma por defecto al personaje masculino. Va a 6.		
<b>Prioridad</b>	Crítica		
<b>Mejoras</b>	-		
<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>
<b>Imagen</b>	I010	Imagen de fondo con color degradado de anaranjado a amarillo, manchada con círculos.	Existente.
	I011	Imagen sobre la que se posicionará el componente de entrada del nombre del usuario.	
<b>Animación</b>	A009	Animación donde aparece el personaje principal en su versión femenina de perfil y sin mochila.	Existente.
	A010	Animación donde aparece el personaje principal en su versión masculina de perfil y sin mochila.	Existente.
	A011	Animación donde aparece el personaje principal en su versión femenina corriendo y con la mochila en la espalda.	Existente.
	A012	Animación donde aparece el personaje principal en su versión masculina corriendo y con la mochila en la espalda.	Existente.
	A013	Animación de las barras en movimiento circular, para	

		resaltar el espacio de personaje seleccionado.	
	A014	Animación de parpadeo amarillo alrededor del campo de entrada de texto para indicar la necesidad de llenar este campo.	Existente.
<b>Sonido</b>	S001	Efecto de sonido para mostrar la animación A_ donde se entrecruzan los personajes.	Existente.
	S021	Sonido muy breve para cuando se pase el cursor sobre un botón.	Existente.
	S022	Sonido muy breve para cuando se haga clic sobre un botón.	Existente.
<b>Texto</b>	T008	Texto que indica al usuario la selección del personaje.	Existente.
	T009	Texto que indica al usuario la entrada de un nombre para el personaje.	Existente.
<b>Reglas pedagógicas</b>			
-			

**Anexo 12.** Descripción textual de la Vista de Presentación EntradaJuego

<b>Descripción Textual de la Vista de Presentación <i>EntradaJuego</i></b>	
<b>Actores de la Vista de Presentación</b>	Usuario
<b>Propósito</b>	Mostrar al usuario una transición antes de entrar a jugar, para evitar un salto brusco hacia la vista Juego.
<b>Objetivos pedagógicos</b>	-
<b>Resumen</b>	La vista se inicia cuando se hace entrada a la vista Juego.
<b>Vistas asociadas</b>	Juego.
<b>Referencias</b>	R7
<b>Precondiciones</b>	-
<b>Poscondiciones</b>	-
<b>Curso Normal de los Eventos</b>	

Acciones del Actor		Respuesta del Sistema	Elementos de la Vista
		1. Se muestra la vista EntradaJuego cuando desde la vista SeleccionarJugador o CargarPartida se desea acceder a la vista Juego. 2. Al finalizar la animación de la vista se accede a la vista Juego.	1.1. Se muestra la animación de una barra de carga. Mientras la barra se llena, la imagen de fondo se aclara y se va mostrando los elementos de la vista Juego.
<b>Cursos Alternos de los Eventos</b>			
<b>Acción</b>		<b>Curso Alterno</b>	
-		-	
<b>Prioridad</b>		Crítica	
<b>Mejoras</b>		-	
<b>Medias a utilizar</b>			
Tipo de Media	Nombre	Descripción	Estado
Imagen	I012	Imagen de fondo, de tonalidad amarilla y naranja.	Existente.
Animación	A015	Animación de la barra que simula la carga del juego.	Existente.
<b>Reglas pedagógicas</b>			
-			

**Anexo 13.** Descripción textual de la Vista de Presentación Mapa

<b>Descripción Textual de la Vista de Presentación <i>Mapa</i></b>	
<b>Actores de la Vista de Presentación</b>	Usuario
<b>Propósito</b>	Mostrar al usuario información de su ubicación en el terreno.
<b>Objetivos pedagógicos</b>	-
<b>Resumen</b>	La vista se muestra cuando en usuario hace clic en al botón Mapa.
<b>Vistas asociadas</b>	Juego.

<b>Referencias</b>		R9	
<b>Precondiciones</b>		-	
<b>Poscondiciones</b>		Se muestra el mapa de la escalada.	
<b>Curso Normal de los Eventos</b>			
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
1. El usuario hace clic en el botón Mapa no estando mostrada la Vista Mapa.		2. Se muestra la Vista Mapa.	2.1. Se despliega el mapa que contiene la información en cuanto a posicionamiento del personaje. Indica la altura y muestra una leyenda.
<b>Cursos Alternos de los Eventos</b>			
<b>Acción</b>		<b>Curso Alterno</b>	
1.1. El usuario hace clic en el botón Mapa estando mostrada la Vista Mapa.		1.1.a. Se oculta la Vista Mapa.	
<b>Prioridad</b>		Secundaria	
<b>Mejoras</b>		-	
<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>
<b>Imagen</b>	I021	Imagen de fondo de la vista, de tonalidad verde, con líneas que conforman el mapa.	Existente.
<b>Animación</b>	A054	Animación de las barras en movimiento circular.	Existente.
	A055	Animación de un punto que parpadea, que representa la ubicación del personaje en el mapa.	Existente.
<b>Reglas pedagógicas</b>			
-			

**Anexo 14.** Descripción textual de la Vista de Presentación Configuración

<b>Descripción Textual de la Vista de Presentación Configuración</b>			
<b>Actores de la Vista de Presentación</b>		Usuario	
<b>Propósito</b>		Permitir al usuario configurar los gráficos de la aplicación y el volumen del sonido de fondo.	
<b>Objetivos pedagógicos</b>		-	
<b>Resumen</b>		La vista se inicia cuando el usuario hace clic sobre el botón Configuración del la vista Juego. Posibilita que el usuario configure los gráficos y el sonido de fondo.	
<b>Vistas asociadas</b>		Juego	
<b>Referencias</b>		R11	
<b>Precondiciones</b>		-	
<b>Poscondiciones</b>		Se muestra la vista Configuración.	
<b>Curso Normal de los Eventos</b>			
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
1. El usuario hace clic en el botón Configuración. 3. El usuario hace clic en el botón Cerrar de la vista.		2. Se muestra la Vista Configuración. 4. Se cierra la vista.	2.1. Los elementos de la vista se muestran mediante una entrada desde una posición fuera de la imagen de fondo.
<b>Cursos Alternos de los Eventos</b>			
<b>Acción</b>		<b>Curso Alterno</b>	
-		-	
<b>Prioridad</b>		Crítica	
<b>Mejoras</b>		-	
<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>
<b>Imagen</b>	I022	Imagen de fondo para la vista.	Existente.
<b>Animación</b>	A057	Animación de un círculo amarillo apareciendo.	Existente.

	A058	Animación televisor sin señal.	Existente.
	A059	Animación una bocina funcionando.	Existente.
<b>Texto</b>	T017	Texto con el nombre de la vista.	Existente.
	T018	Texto con la palabra: Volumen	Existente.
	T019	Texto con la palabra: Gráficos	Existente.
<b>Reglas pedagógicas</b>			
-			

**Anexo 15.** Descripción textual de la Vista de Presentación ContestarPregunta

<b>Descripción Textual de la Vista de Presentación <i>ContestarPregunta</i></b>		
<b>Actores de la Vista de Presentación</b>	Usuario	
<b>Propósito</b>	Permitir al usuario responder una pregunta propuesta por el sistema.	
<b>Objetivos pedagógicos</b>	Permitir al usuario demostrar dominio sobre el tema de la vida y obra de José Martí.	
<b>Resumen</b>	La vista se inicia cuando el personaje, conducido por el usuario, selecciona una pregunta del nivel en que se está jugando. La vista muestra una pregunta con tres posibles respuestas y el valor de la misma en puntos.	
<b>Vistas asociadas</b>	Pista, Juego.	
<b>Referencias</b>	R13	
<b>Precondiciones</b>	-	
<b>Poscondiciones</b>	Si la pregunta fue respondida correctamente se suman los puntos obtenidos al total acumulado hasta el momento. Si no, se restan.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
1. El personaje conducido por el usuario hace contacto con	2. Se muestra la vista ContestarPregunta. Se limitan los	2.1. La ventana hace entrada desde la parte superior de la

una pregunta del nivel en que se esté jugando.	permisos del personaje para seguir avanzando así como de su interacción con otros elementos del nivel.	pantalla. La vista contiene el texto de la pregunta y de tres posibles respuestas, así como el valor de la pregunta. La mascota Lapiolo está ubicada en el lateral izquierdo de la vista.
3. El usuario hace clic en la respuesta correcta.	4. Se suman los puntos a sumar por la pregunta. 5. Se notifica al usuario el éxito en su respuesta. 6. Se cierra la vista.	6.1. Todos los elementos de la vista hace salida por el lateral derecho de la pantalla.
7. El usuario pide una pista para poder responder la pregunta.	8. Se muestra la vista Pista y se disminuye el valor de la pregunta.	8.1. El valor de la pregunta disminuye en un 75%.

**Cursos Alternos de los Eventos**

Acción	Curso Alterno
3.1. El usuario hace clic en una respuesta incorrecta por primera vez. 3.2. El usuario hace clic en una respuesta incorrecta por segunda vez.	3.1.a. Disminuye el valor de la pregunta. Va a 3. 3.2.a. Se resta todos los puntos que valía la pregunta en un inicio. 3.2.b. Se notifica al usuario el fracaso en su respuesta. Va a 6.
<b>Prioridad</b>	Crítica
<b>Mejoras</b>	-

**Medias a utilizar**

Tipo de Media	Nombre	Descripción	Estado
<b>Imagen</b>	I023	Imagen de fondo de la vista.	Existente.
<b>Animación</b>	A061	Animación donde aparece la mascota Lapiolo indicando al usuario que piense para responder.	Existente.
	A062	Animación donde aparece la mascota Lapiolo	Existente.

		notificando el éxito de la respuesta.	
	A063	Animación donde aparece la mascota Lapiolo indicando que esa no es la respuesta correcta.	Existente.
	A064	Animación donde aparece la mascota Lapiolo notificando el fracaso de la respuesta.	Existente.
	A065	Animación donde aparece una paloma de marcado, para indicar la respuesta correcta.	Existente.
	A066	Animación donde aparece una franja que se superpone sobre una respuesta incorrecta.	Existente.
	A067	Animación con el texto que notifica si fue correcta o no la respuesta.	Existente.
	A068	Franja de subrayado que varía su color entre rojo y blanco.	Existente.
<b>Sonido</b>	S014	Sonido que simula la voz de la mascota cuando la respuesta es correcta.	Existente.
	S015	Sonido que simula la voz de la mascota cuando la respuesta es incorrecta.	Existente.
	S016	Sonido que simula la voz de la mascota cuando las oportunidades de responder se agotan.	Existente.
	S021	Sonido muy breve para cuando se pase el cursor sobre un botón.	Existente.
	S022	Sonido muy breve para cuando se haga clic sobre un botón.	Existente.
<b>Texto</b>	T020	Texto donde se muestra el contenido de una pregunta.	Existente.
<b>Reglas pedagógicas</b>			
-			

**Anexo 16.** Descripción textual de la Vista de Presentación Pista

<b>Descripción Textual de la Vista de Presentación Pista</b>			
<b>Actores de la Vista de Presentación</b>		Usuario	
<b>Propósito</b>		Permitir que usuario obtenga información referente a la situación en que se encuentre en el momento que la solicite.	
<b>Objetivos pedagógicos</b>		Permitir al usuario ejercitar la lectura y la comprensión de la misma. Mostrar al usuario información para que este la convierta en conocimientos.	
<b>Resumen</b>		La vista se inicia cuando el usuario hace clic sobre la mascota Lapiolo.	
<b>Vistas asociadas</b>		Juego, ContestarPregunta.	
<b>Referencias</b>		R10	
<b>Precondiciones</b>		-	
<b>Poscondiciones</b>		Se muestra al usuario la información requerida.	
<b>Curso Normal de los Eventos</b>			
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>		<b>Elementos de la Vista</b>
1. El usuario hace clic sobre la mascota Lapiolo. 3. El usuario hace clic en el botón Cerrar de la vista.	2. Se muestra un texto al usuario con la información requerida. 4. Se cierra la vista Pista.		2.1. El texto estará contenido dentro de una nube que simboliza el pensamiento de la mascota. 4.1. Desaparece el texto y se disuelve la nube.
<b>Cursos Alternos de los Eventos</b>			
<b>Acción</b>	<b>Curso Alterno</b>		
-	-		
<b>Prioridad</b>	Crítica		
<b>Mejoras</b>	-		
<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>

<b>Animación</b>	A056	Animación donde aparece la nube, y donde se disuelve.	Existente.
<b>Texto</b>	T016	Texto que muestra la información para orientar al usuario.	Existente.
<b>Reglas pedagógicas</b>			
-			

**Anexo 17.** Descripción textual de la Vista de Presentación Guardando

<b>Descripción Textual de la Vista de Presentación <i>Guardando</i></b>		
<b>Actores de la Vista de Presentación</b>	Usuario	
<b>Propósito</b>	Permitir al usuario guardar la partida.	
<b>Objetivos pedagógicos</b>	-	
<b>Resumen</b>	La vista se inicia cuando el usuario hace clic sobre el botón Guardar del la vista Juego.	
<b>Vistas asociadas</b>	Juego.	
<b>Referencias</b>	R12	
<b>Precondiciones</b>	El personaje debe tener permisos para trasladarse.	
<b>Poscondiciones</b>	Se muestra la vista Guardando.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
1. El usuario hace clic en el botón Guardar.	2. Se muestra la Vista Guardando.	2.1. La vista está compuesta únicamente por una animación. Al culminar la animación se cierra la vista.
<b>Cursos Alternos de los Eventos</b>		
<b>Acción</b>	<b>Curso Alternativo</b>	
-	-	
<b>Prioridad</b>	Crítica	
<b>Mejoras</b>	-	

Medias a utilizar			
Tipo de Media	Nombre	Descripción	Estado
Animación	A060	Animación donde aparece el disquete con la barra de carga debajo.	Existente.
Reglas pedagógicas			
-			

#### Anexo 18. Descripción textual de la Vista de Presentación PerderVida

Descripción Textual de la Vista de Presentación <i>PerderVida</i>		
<b>Actores de la Vista de Presentación</b>	Usuario	
<b>Propósito</b>	Permitir al usuario recuperar su personaje y obtener consejos dados por la mascota.	
<b>Objetivos pedagógicos</b>	Animar y aconsejar al usuario tras la pérdida de una vida del personaje.	
<b>Resumen</b>	La vista se muestra cuando el personaje conducido por el usuario cae por un precipicio o cuando el personaje pierde toda su energía.	
<b>Vistas asociadas</b>	Juego.	
<b>Referencias</b>	R14	
<b>Precondiciones</b>	El personaje cae por un precipicio ó su energía toma valor cero.	
<b>Poscondiciones</b>	El personaje se incorpora al juego con una vida menos.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	Elementos de la Vista
1. El personaje guiado por el usuario cae por un precipicio o su energía toma valor cero.	2. Se muestra la vista PerderVida.	2.1. Aparece el personaje sentado en una camilla y la mascota Lapiolo en un extremo mostrando un mensaje.
3. El usuario hace clic en el	4. Se muestra la vista Juego.	4.1. Se retira la camilla con el

botón Seguir Jugando.			personaje y la mascota, quedando en pantalla solo la imagen de fondo de la vista. Se muestra una barra de carga y a medida que se llena la barra se aclara la imagen de fondo, mostrando los elementos del nivel que se esté jugando.
<b>Cursos Alternos de los Eventos</b>			
<b>Acción</b>		<b>Curso Alterno</b>	
-		-	
<b>Prioridad</b>		Crítica	
<b>Mejoras</b>		-	
<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>
<b>Imagen</b>	I024	Imagen de fondo de azul.	Existente.
	I025	Imagen de tonalidad amarilla y naranja.	Existente.
	I026	Imagen de una camilla sobre unos troncos.	Existente.
<b>Animación</b>	A069	Animación de la barra de carga.	Existente.
	A070	Animación que representa el pensamiento de la mascota, donde aparece un consejo para el usuario.	Existente.
	A071	Animación que muestra al personaje golpeado.	Existente.
	A072	Animación que muestra a la mascota aconsejando al personaje.	Existente.
<b>Sonido</b>	S017	Sonido que simula la voz de la mascota.	Existente.
	S021	Sonido muy breve para cuando se pase el cursor sobre un botón.	Existente.
	S022	Sonido muy breve para cuando se haga clic sobre un botón.	Existente.

<b>Texto</b>	T021	Texto de un consejo para el usuario.	Existente.
<b>Reglas pedagógicas</b>			
-			

**Anexo 19.** Descripción textual de la Vista de Presentación CambiarNivel

<b>Descripción Textual de la Vista de Presentación <i>CambiarNivel</i></b>		
<b>Actores de la Vista de Presentación</b>	Usuario	
<b>Propósito</b>	Informar al usuario de que acaba de vencer un nivel.	
<b>Objetivos pedagógicos</b>	Permitir al usuario prepararse psicológicamente para jugar el próximo nivel.	
<b>Resumen</b>	La vista se inicia cuando el usuario logra responder todas las preguntas del nivel y guía al personaje a escalar la cima del nivel: plataforma más cercana a la esquina superior derecha de la pantalla.	
<b>Vistas asociadas</b>	Juego.	
<b>Referencias</b>	R15	
<b>Precondiciones</b>	Todas las preguntas del nivel deben ser respondidas.	
<b>Poscondiciones</b>	Se muestra un nuevo nivel a jugar.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
1. El usuario conduce al personaje hasta la plataforma más cercana a la esquina superior derecha de la pantalla, después de haber contestado todas las preguntas propuestas en el nivel.	2. Se muestra la vista CambiarNivel.	2.1. Se muestra el logo de Pico Maestro animado, así como un texto que informa al usuario de la nueva altura alcanza.
	3. Al culminar la animación se muestra un nuevo nivel de la vista Juego.	3.1. Los elementos implicados en la vista se van retirando hasta que queda solo la

		imagen de fondo y una barra de carga, a medida que se llena esta se aclara la imagen de fondo y se van mostrando los elementos del nuevo nivel alcanzado.	
<b>Cursos Alternos de los Eventos</b>			
<b>Acción</b>	<b>Curso Alterno</b>		
1.1. El usuario no ha contestado todas las preguntas propuestas en el nivel.	1.1.a. El personaje indica la imposibilidad de continuar hacia el próximo nivel.		
<b>Prioridad</b>	Crítica		
<b>Mejoras</b>	-		
<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>
<b>Imagen</b>	I027	Imagen de fondo de tonalidad azul.	Existente.
<b>Animación</b>	A003	Animación del logotipo de Pico Maestro.	Existente.
	A069	Animación de la barra de carga.	Existente.
	A073	Animación de la bandera de Martianitos.	Existente.
	A002	Animación en donde la mascota (Lapiolo) está recostado	Existente.
	A011	Animación donde aparece el personaje principal en su versión femenina corriendo y con la mochila en la espalda.	Existente.
	A012	Animación donde aparece el personaje principal en su versión masculina corriendo y con la mochila en la espalda.	Existente.
<b>Sonido</b>	S018	Sonido que simboliza una victoria.	Existente.
	S019	Sonido leve y rápido que acompaña la entrada del	Existente.

		texto	
	S020	Sonido leve y rápido que simboliza la retirada de los elementos de la vista.	Existente.
<b>Texto</b>	T022	Texto que informa al usuario de la nueva altura alcanza.	Existente.
<b>Reglas pedagógicas</b>			
-			

**Anexo 20.** Descripción textual de la Vista de Presentación PerderJuego

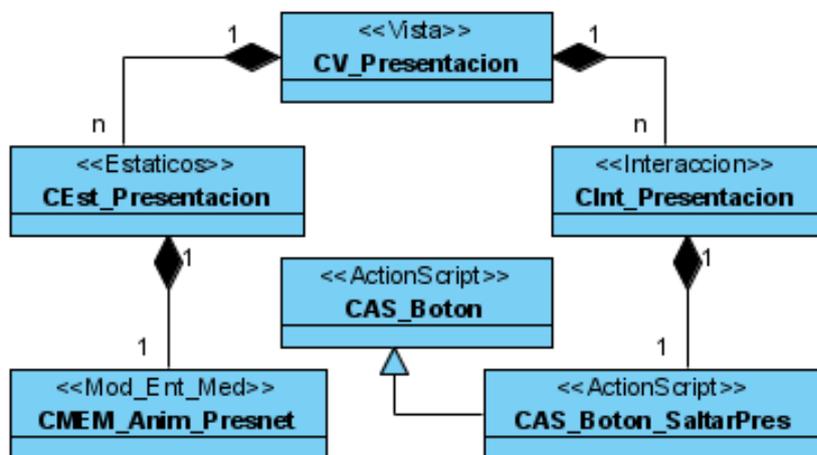
<b>Descripción Textual de la Vista de Presentación <i>PerderJuego</i></b>		
<b>Actores de la Vista de Presentación</b>	Usuario	
<b>Propósito</b>	Informar al usuario de su fracaso en el cumplimiento de la misión.	
<b>Objetivos pedagógicos</b>	-	
<b>Resumen</b>	La vista se inicia cuando el personaje pierde todas sus vidas. La vista se mantiene fija hasta que el usuario decida retornar a la vista Principal.	
<b>Vistas asociadas</b>	Principal.	
<b>Referencias</b>	R16	
<b>Precondiciones</b>	Vidas del personaje en cero.	
<b>Poscondiciones</b>	Se muestra la vista PerderJuego.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
1. El personaje pierde todas sus vidas.	2. Se muestra la vista PerderJuego.	2.1. Se muestra la animación referente a esta vista.
<b>Cursos Alternos de los Eventos</b>		
<b>Acción</b>	<b>Curso Alternativo</b>	
-	-	
<b>Prioridad</b>	Crítica	

<b>Mejoras</b>		-	
<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>
<b>Animación</b>	A074	Animación para cuando se pierda el juego.	Existente.
<b>Sonido</b>	S021	Sonido muy breve para cuando se pase el cursor sobre un botón.	Existente.
	S022	Sonido muy breve para cuando se haga clic sobre un botón.	Existente.
<b>Reglas pedagógicas</b>			
-			

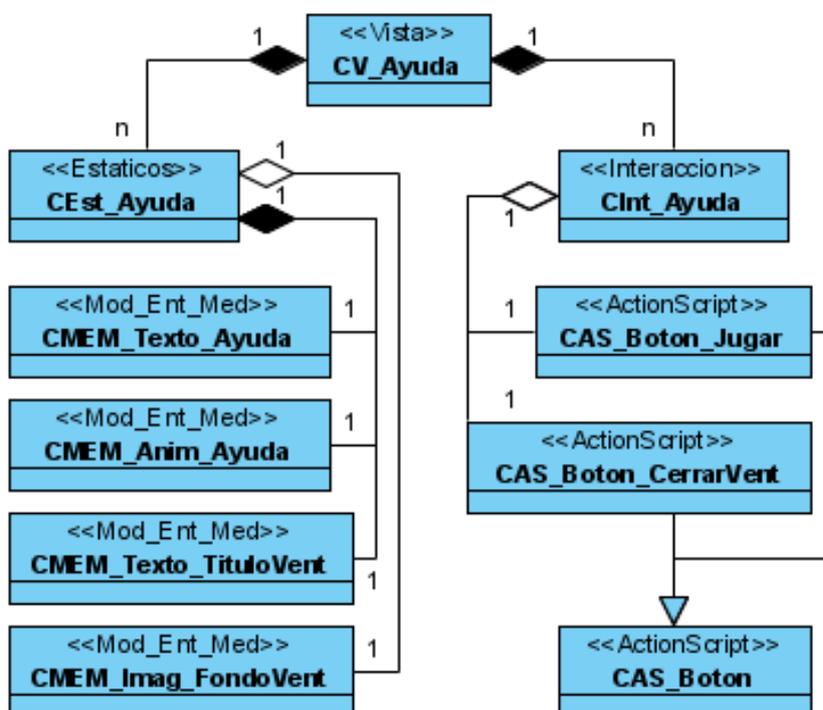
**Anexo 21.** Descripción textual de la Vista de Presentación GanarJuego

<b>Descripción Textual de la Vista de Presentación <i>GanarJuego</i></b>		
<b>Actores de la Vista de Presentación</b>	Usuario	
<b>Propósito</b>	Informar al usuario de su éxito en el cumplimiento de la misión.	
<b>Objetivos pedagógicos</b>	-	
<b>Resumen</b>	La vista se inicia cuando el personaje logra escalar el nivel diez. Muestra una animación alegórica al éxito de la misión y un ranking con el nombre de los cinco usuarios que con mayor puntuación han llegado a la cima.	
<b>Vistas asociadas</b>	Principal.	
<b>Referencias</b>	R17	
<b>Precondiciones</b>	Nivel 10 escalado.	
<b>Poscondiciones</b>	Se muestra la vista GanarJuego.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	<b>Elementos de la Vista</b>
1. El usuario logra que el	2. Se muestra la vista GanarJuego.	2.1. Se muestra la animación

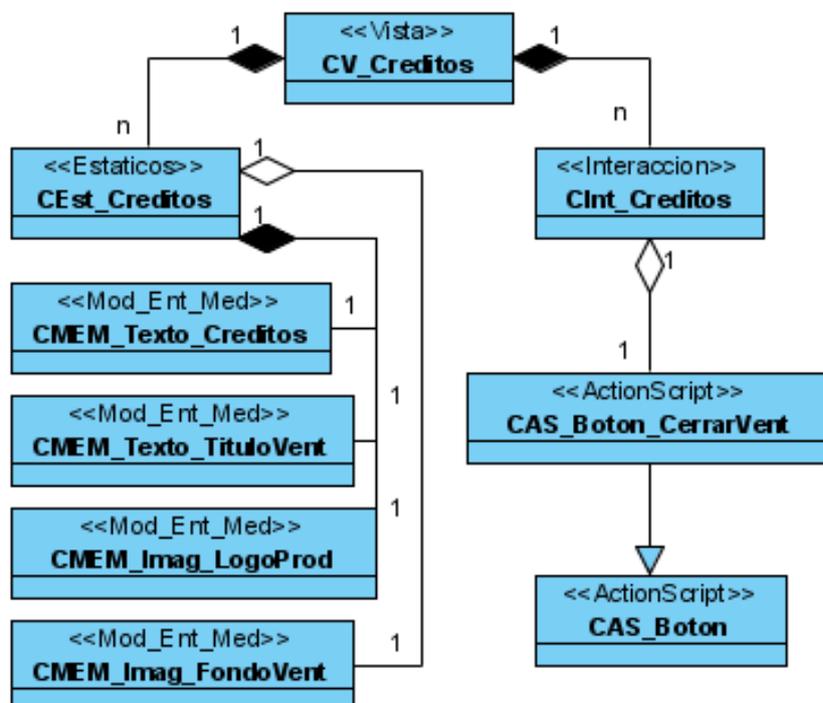
personaje escale el decimo nivel.		referente a esta vista.	
<b>Cursos Alternos de los Eventos</b>			
<b>Acción</b>	<b>Curso Alterno</b>		
-	-		
<b>Prioridad</b>	Crítica		
<b>Mejoras</b>	-		
<b>Medias a utilizar</b>			
<b>Tipo de Media</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Estado</b>
<b>Animación</b>	A075	Animación para cuando se gana el juego.	Existente.
<b>Sonido</b>	S021	Sonido muy breve para cuando se pase el cursor sobre un botón.	Existente.
	S022	Sonido muy breve para cuando se haga clic sobre un botón.	Existente.
<b>Texto</b>	T023	Texto que muestra el ranking de las mejores jugadas.	Existente.
<b>Reglas pedagógicas</b>			
-			



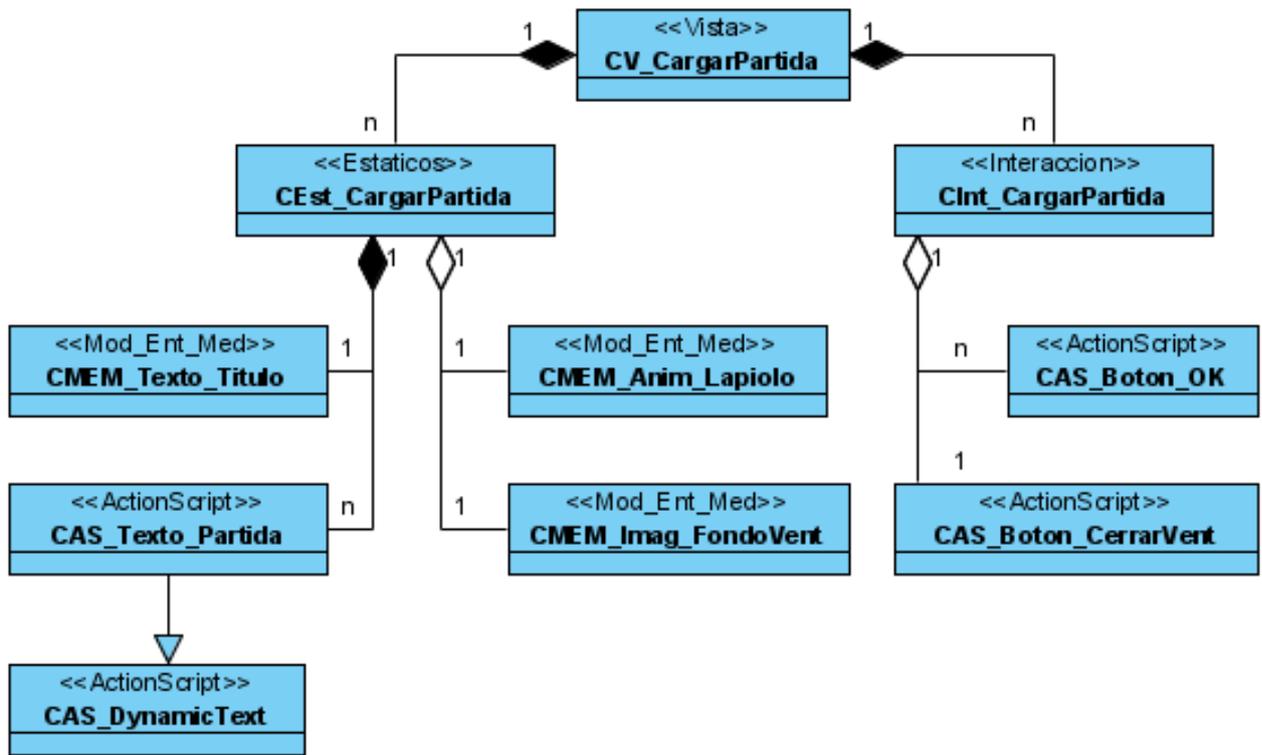
Anexo 22. DEP Vista de Presentación Presentacion



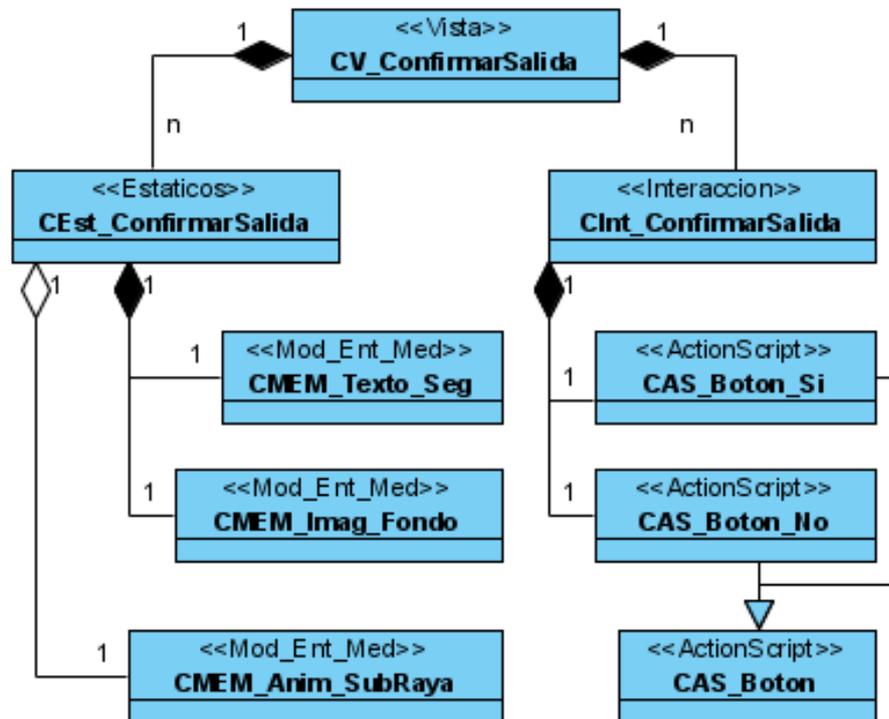
Anexo 23. DEP Vista de Presentación Ayuda



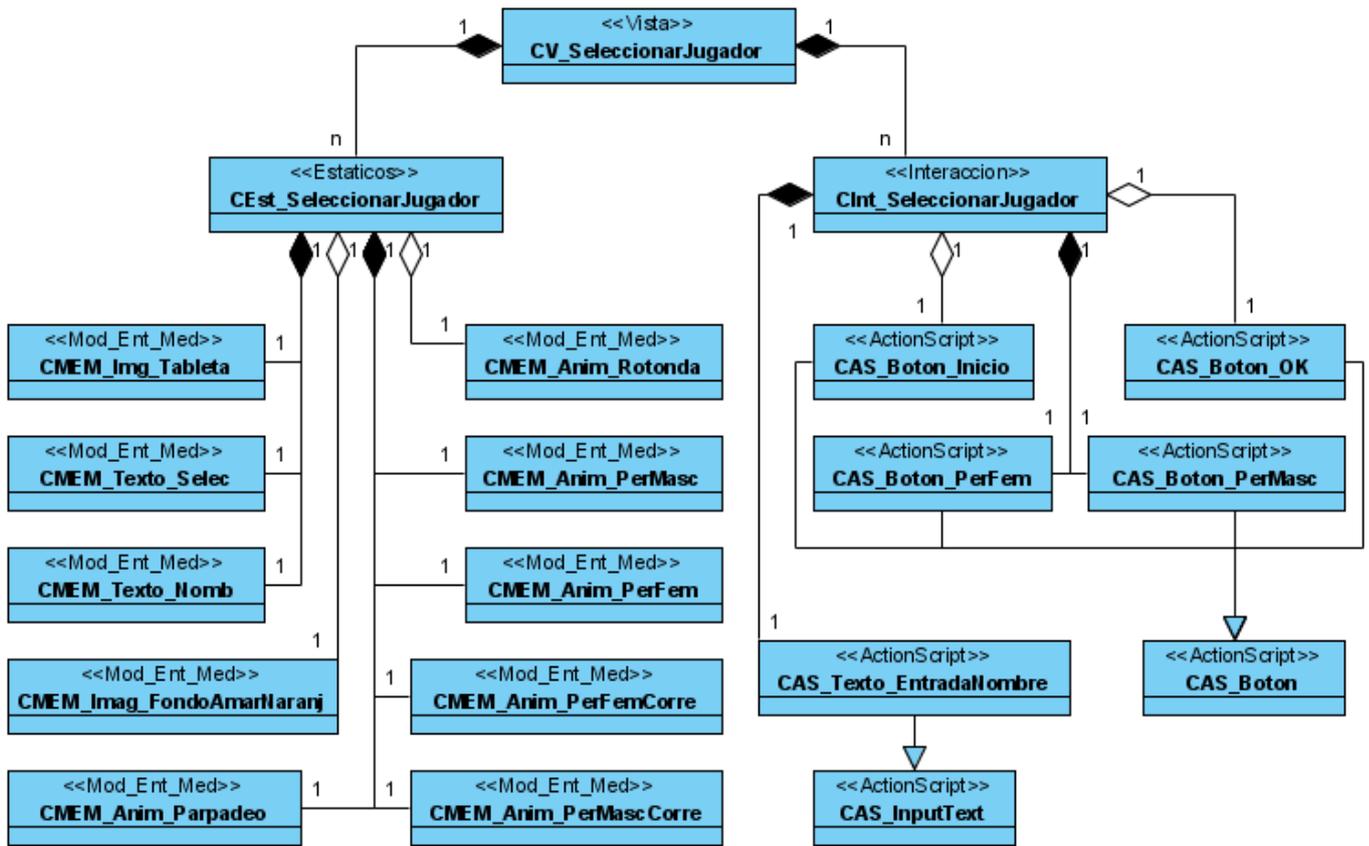
Anexo 24. DEP Vista de Presentación Creditos



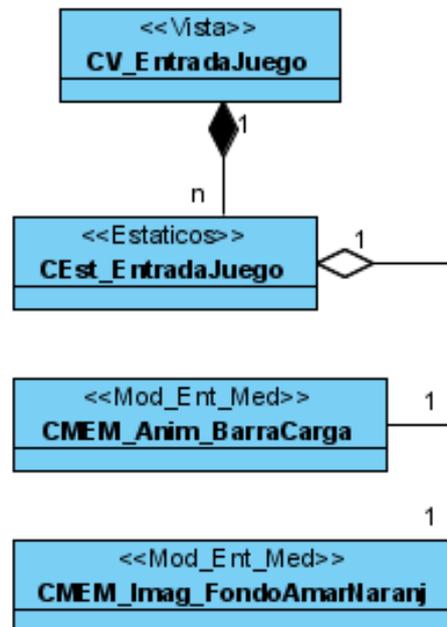
Anexo 25. DEP Vista de Presentación CargarPartida



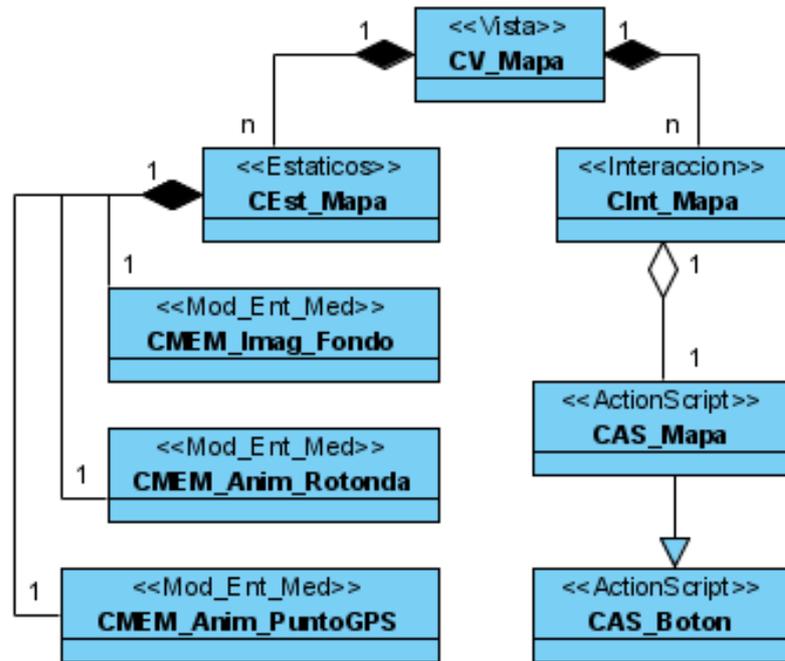
Anexo 26. DEP Vista de Presentación ConfirmarSalida



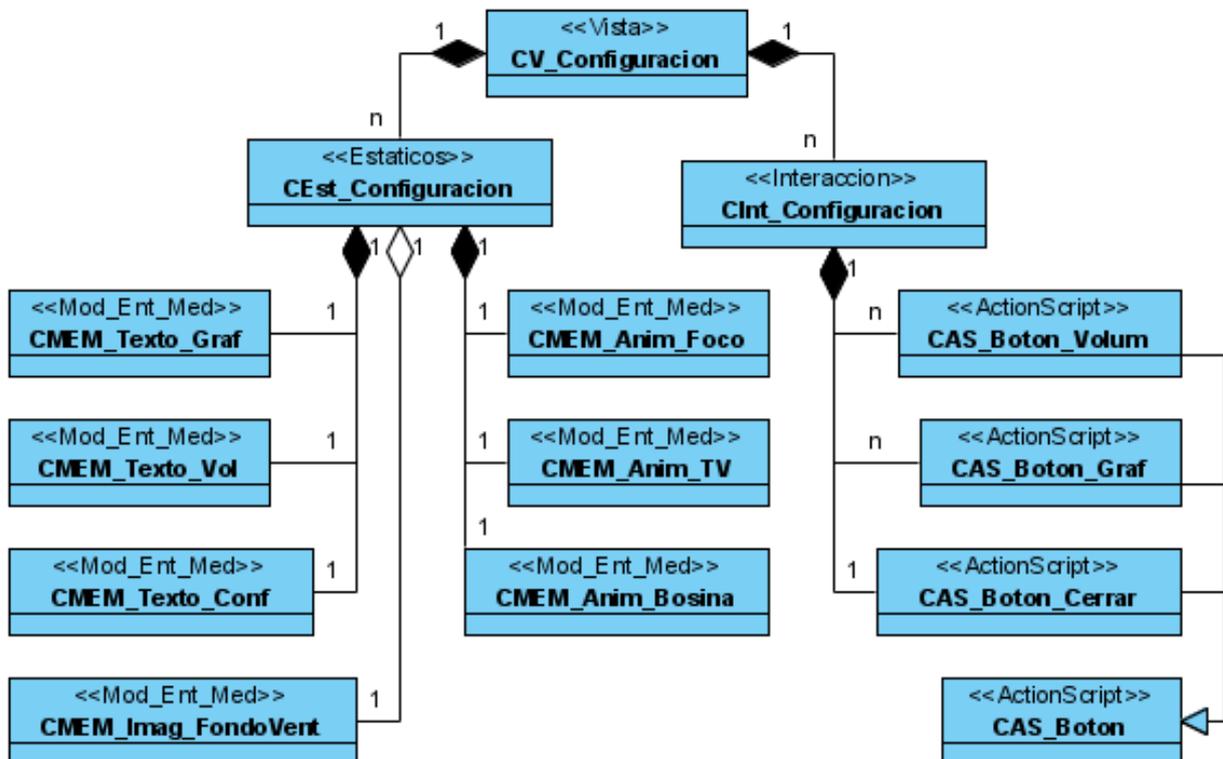
Anexo 27. DEP Vista de Presentación SeleccionarJugador



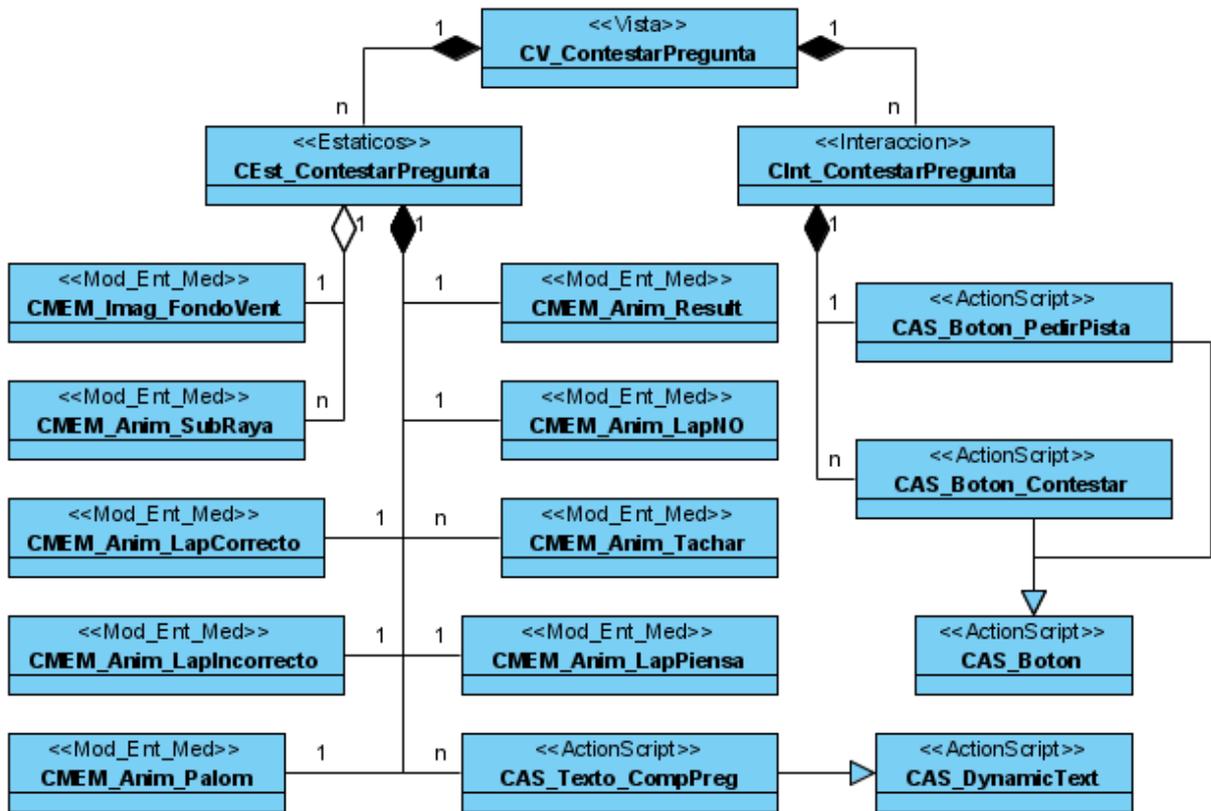
Anexo 28. DEP Vista de Presentación EntradaJuego



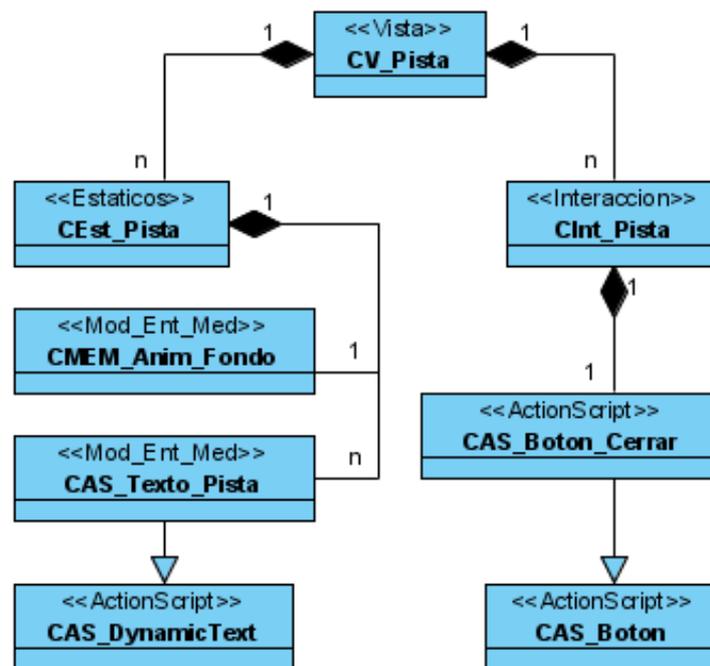
Anexo 29. DEP Vista de Presentación Mapa



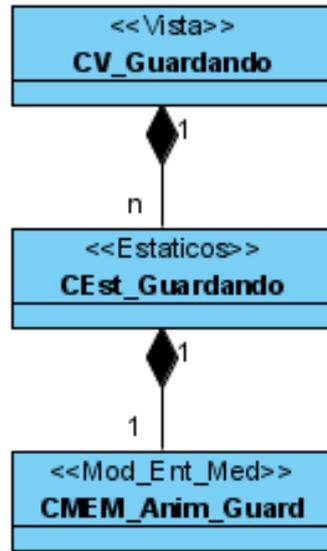
Anexo 30. DEP Vista de Presentación Configuración



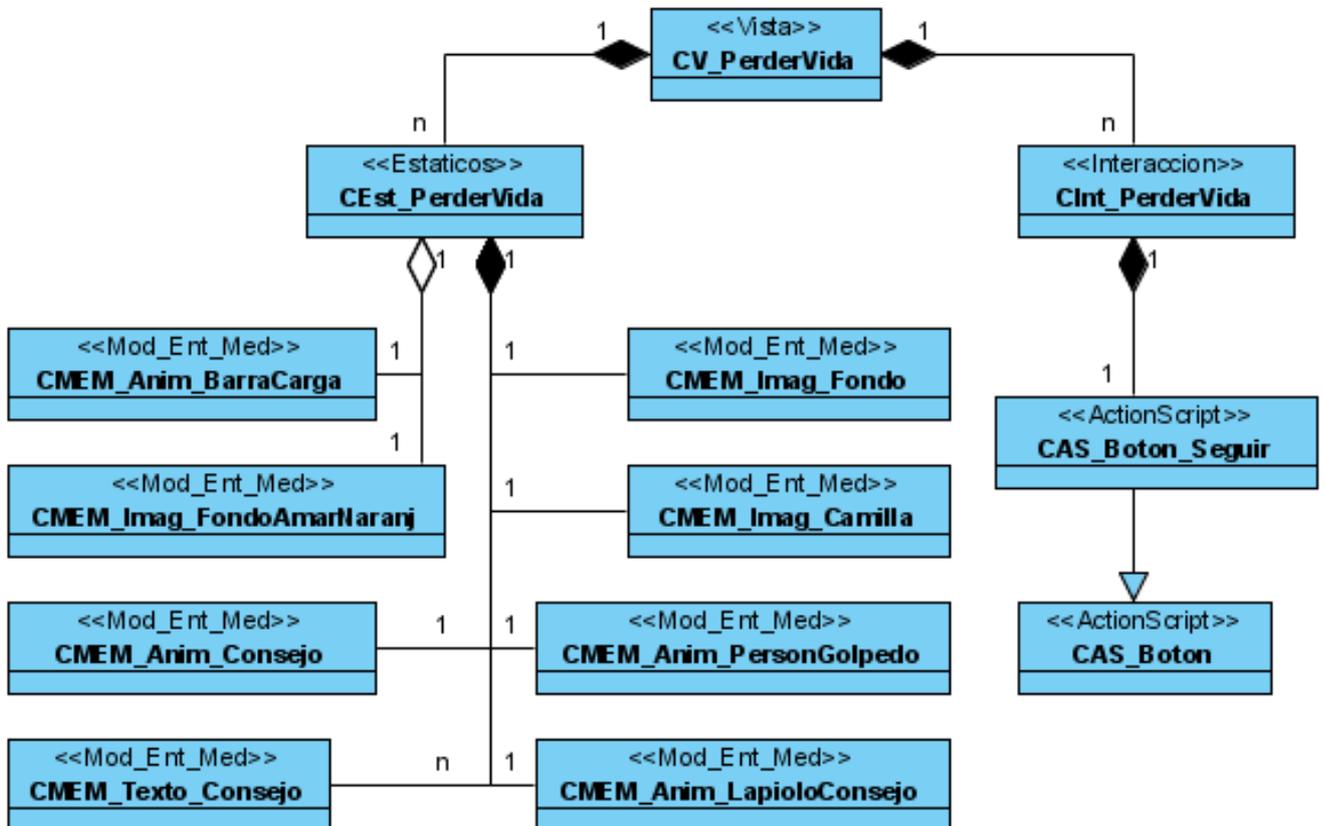
Anexo 31. DEP Vista de Presentación ContestarPregunta



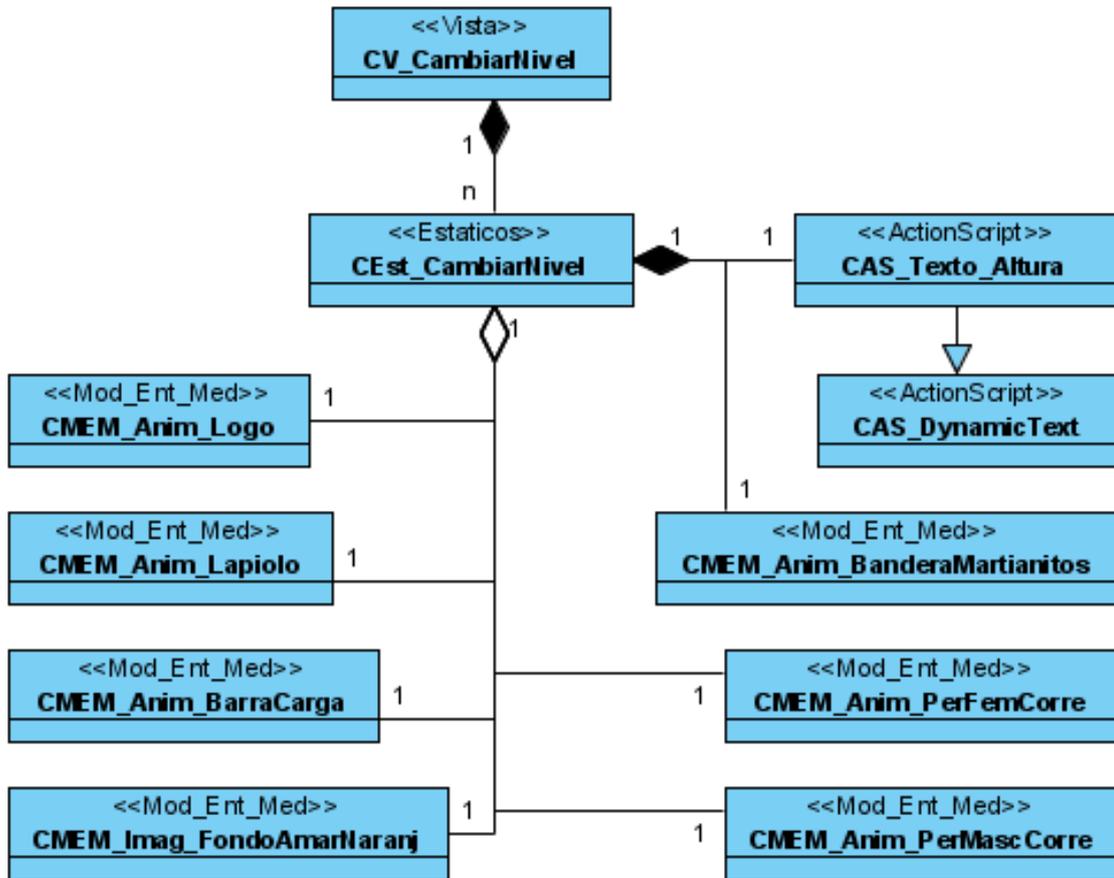
Anexo 32. DEP Vista de Presentación Pista



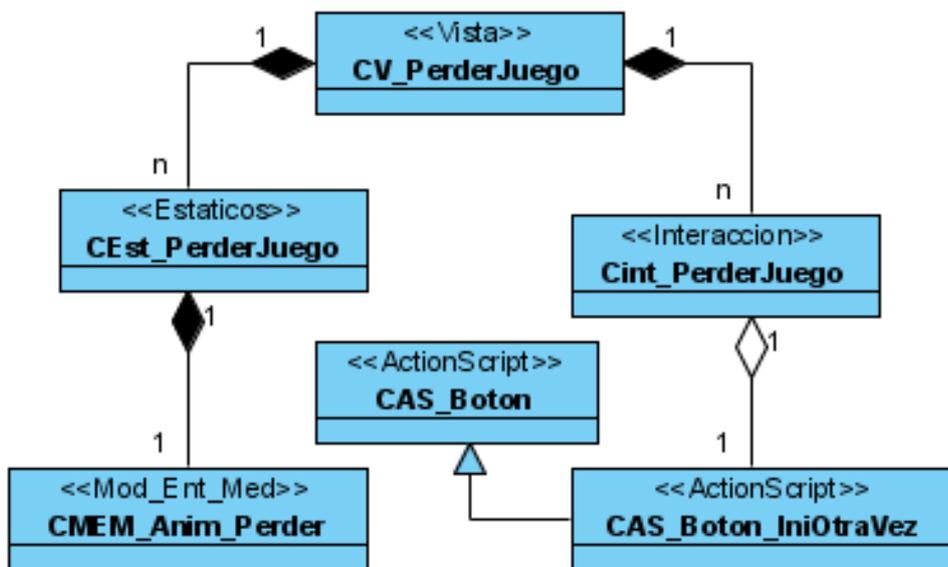
Anexo 33. DEP Vista de Presentación Guardando



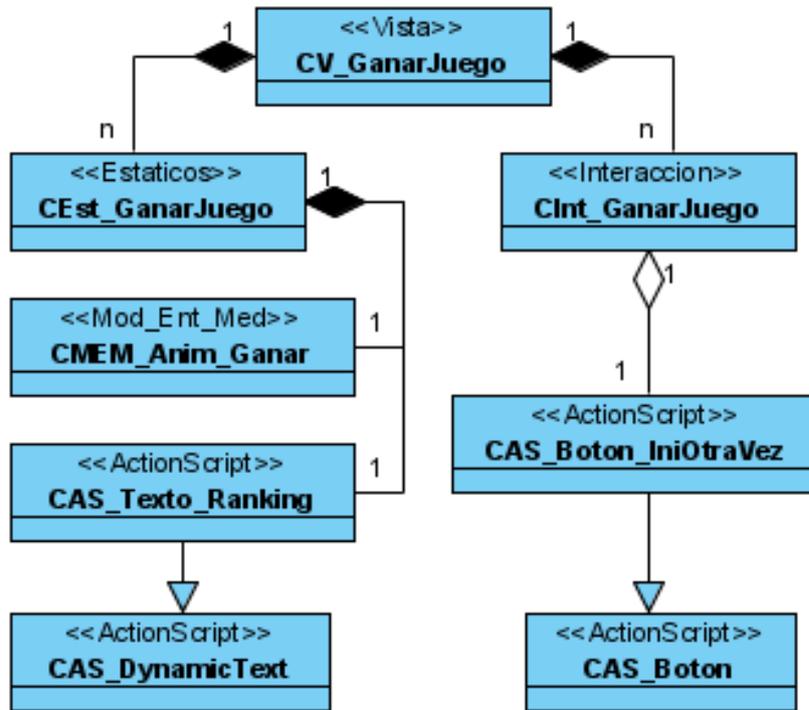
Anexo 34. DEP Vista de Presentación PerderVida



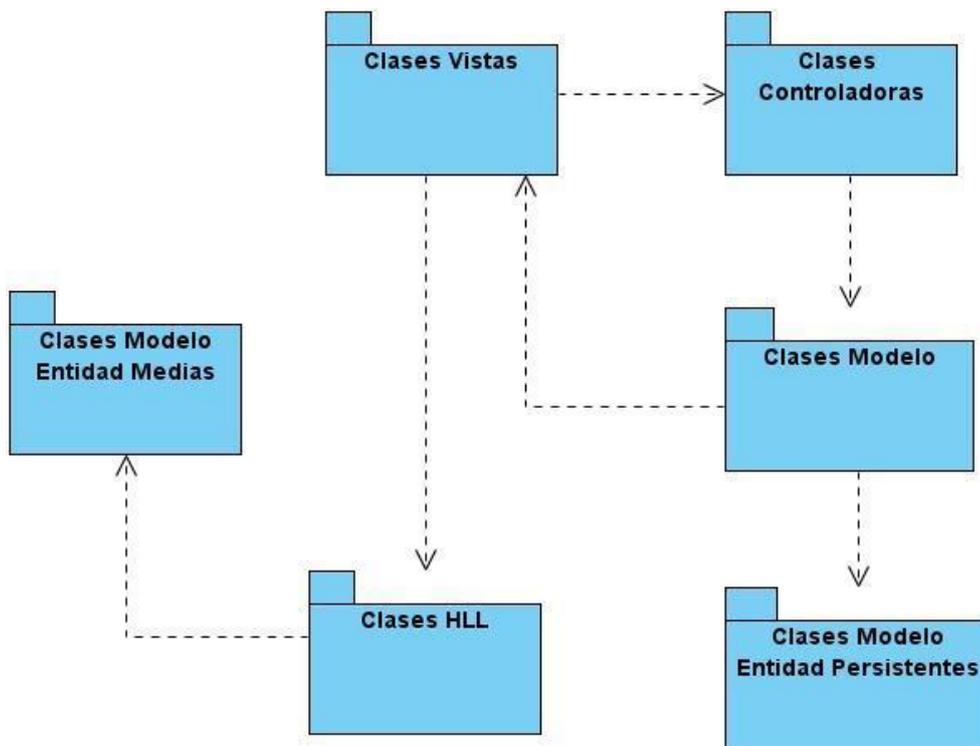
Anexo 35. DEP Vista de Presentación CambiarNivel



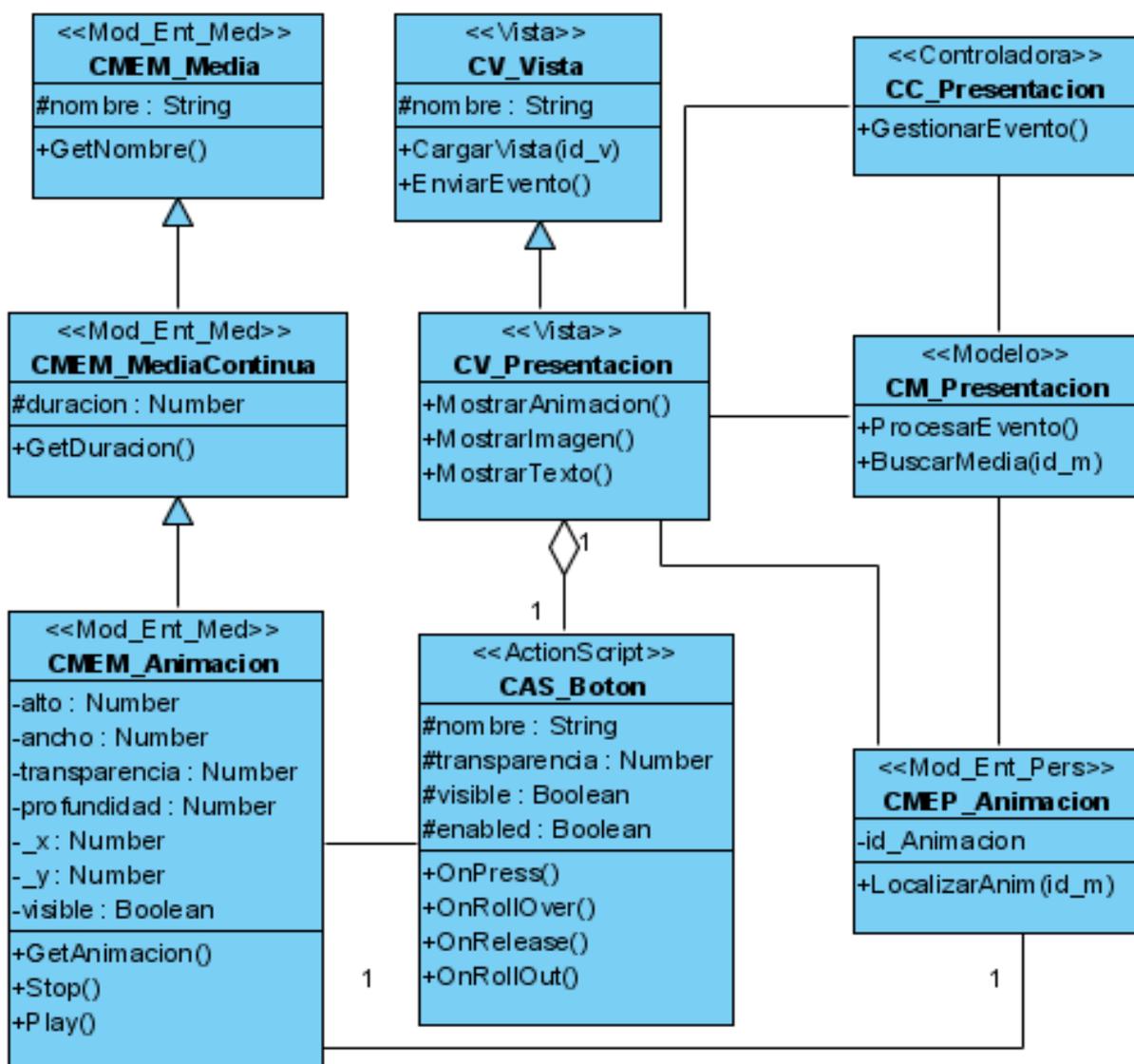
Anexo 36. DEP Vista de Presentación PerderJuego



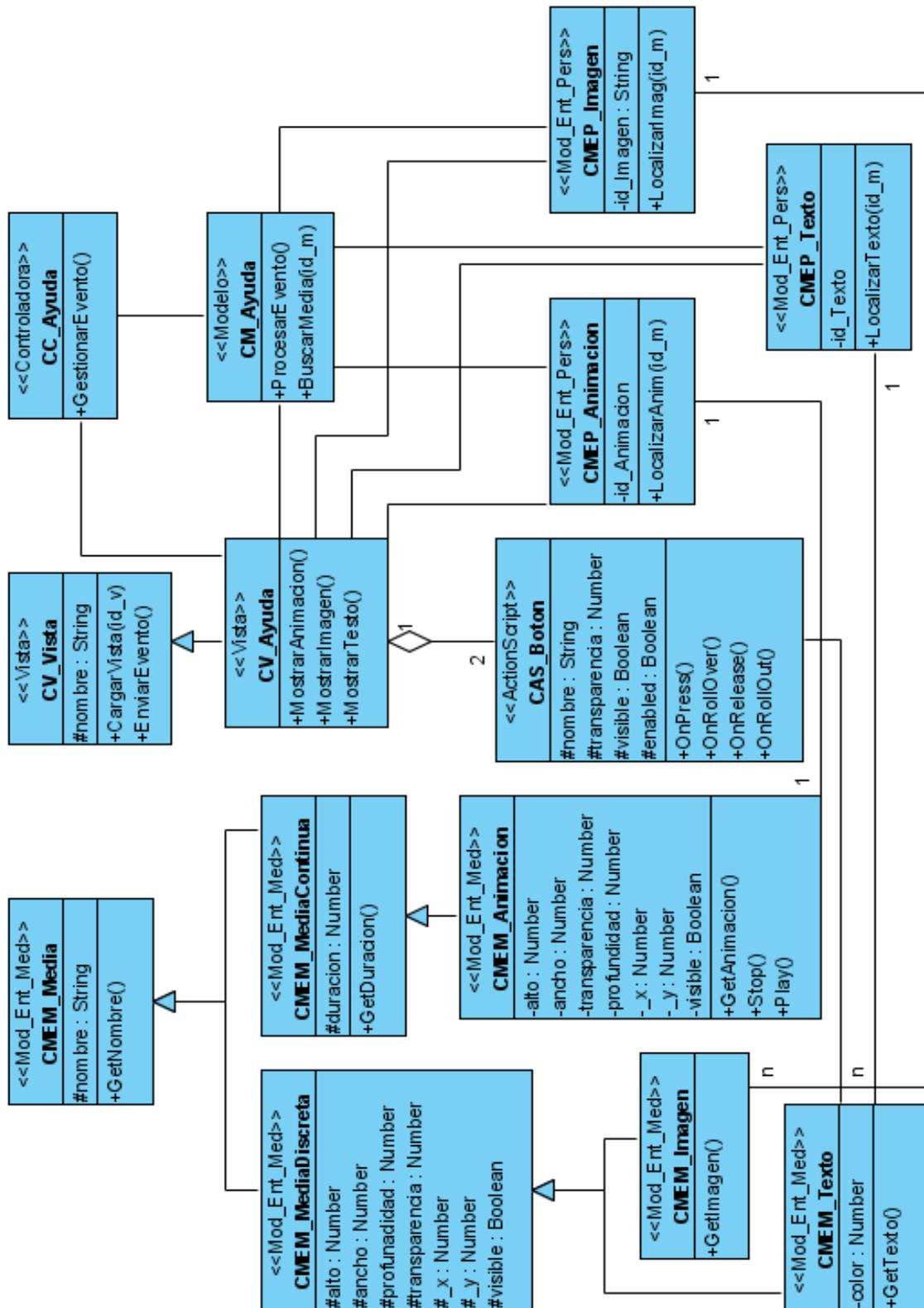
Anexo 37. DEP Vista de Presentación GanarJuego



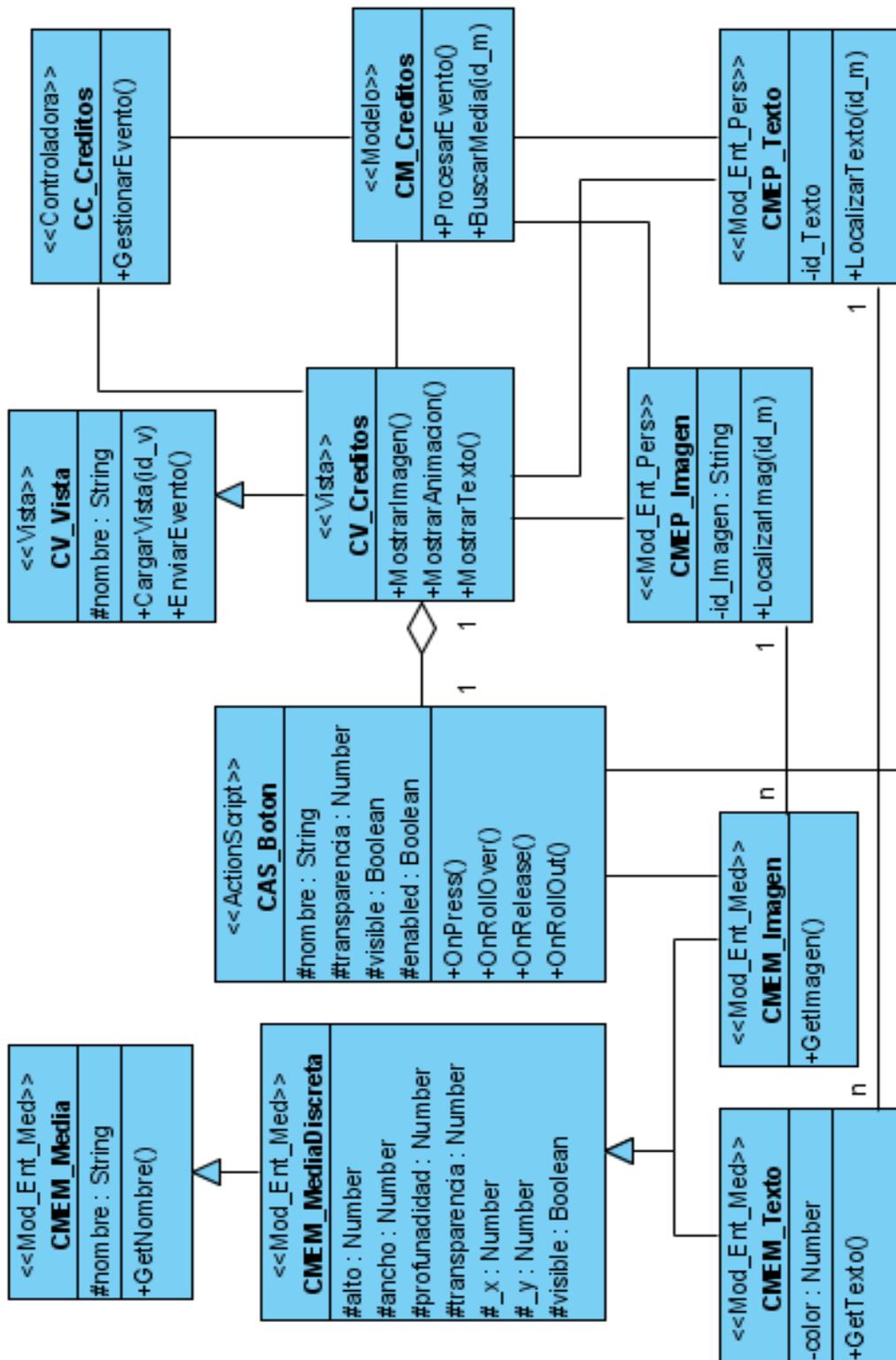
Anexo 38. Vista de Gestión de un diagrama de clases de ApEM-L



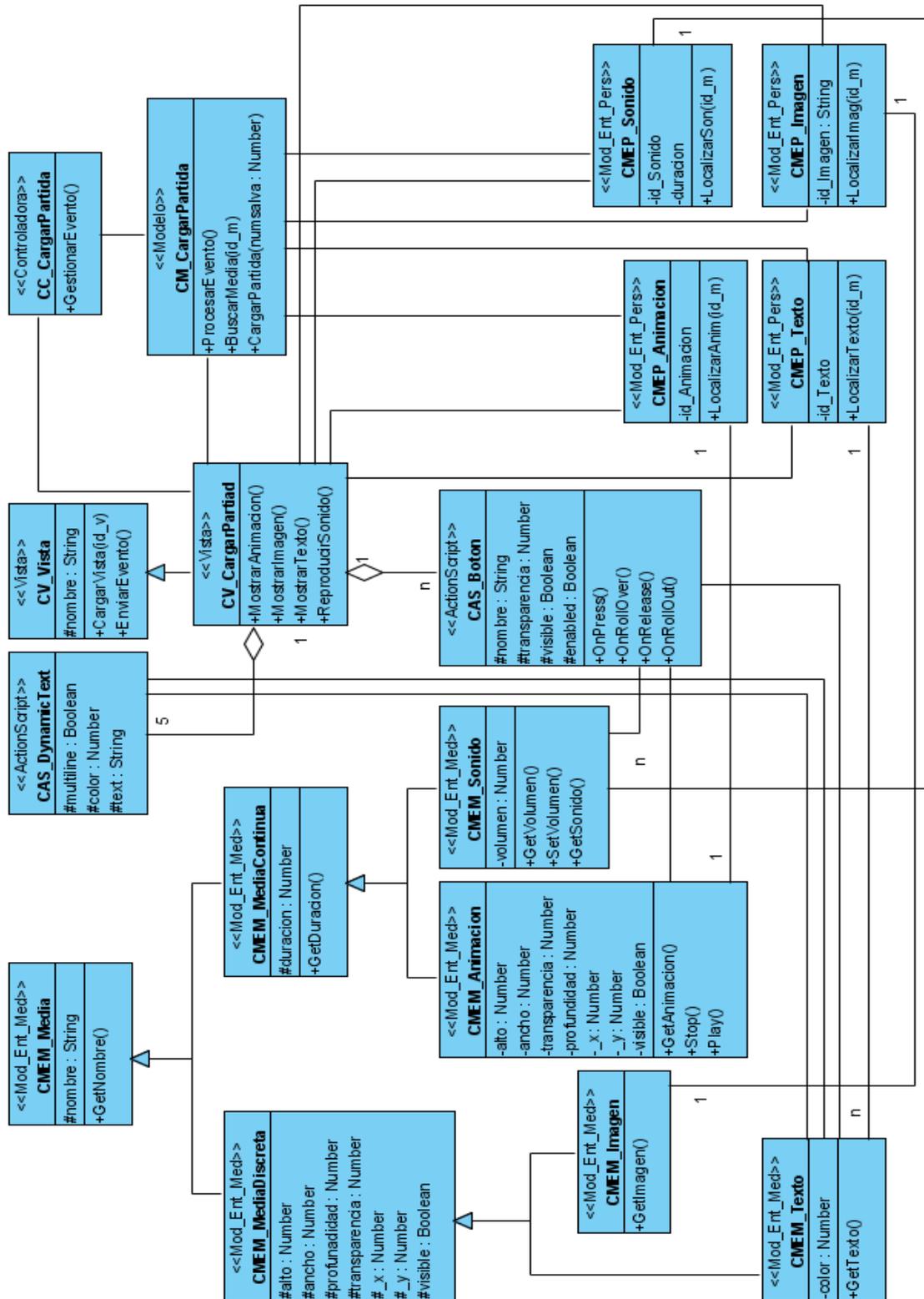
Anexo 39. Diagrama de Clases de la Vista de Presentación Presentacion



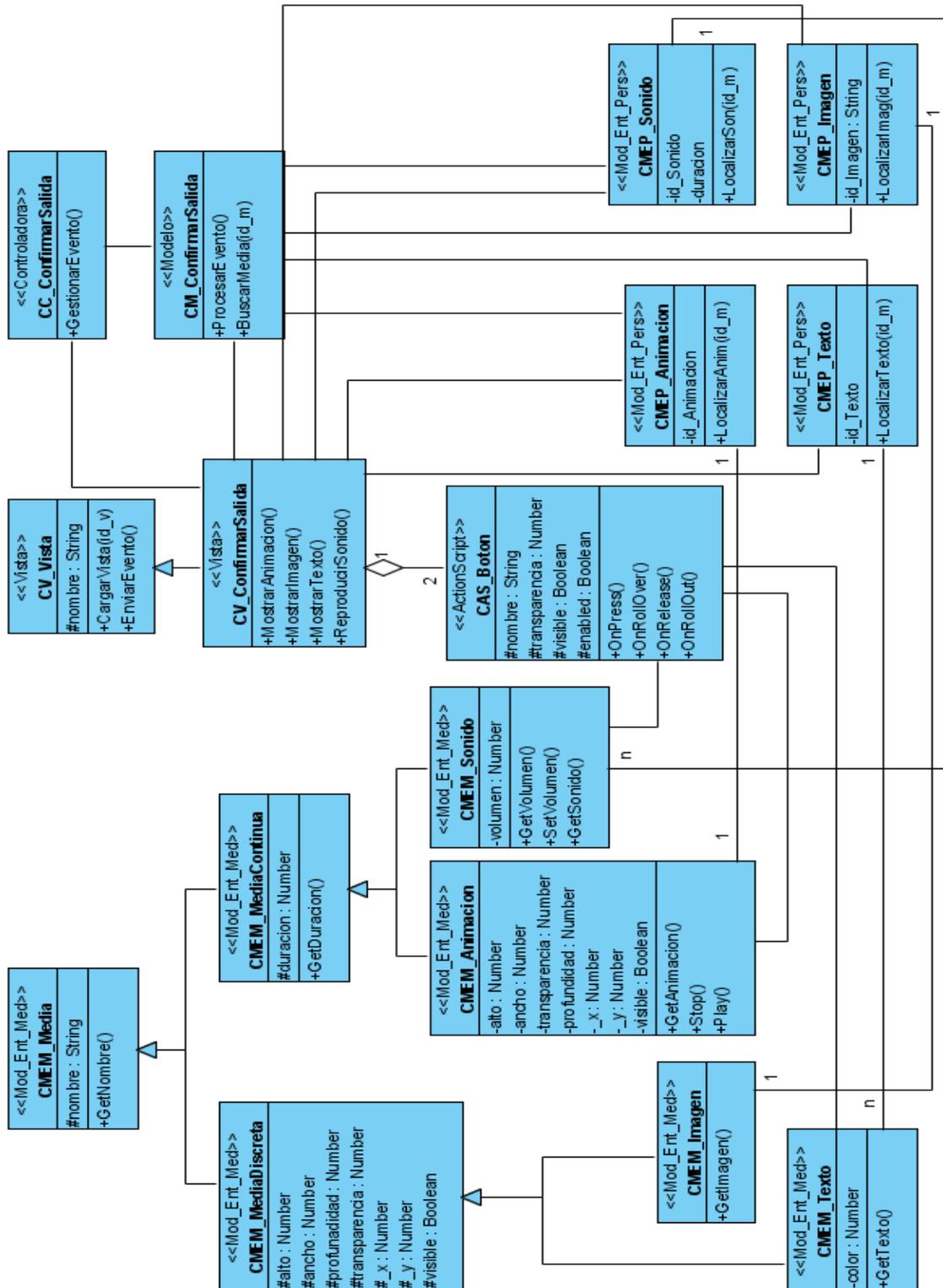
Anexo 40. Diagrama de Clases de la Vista de Presentación Ayuda



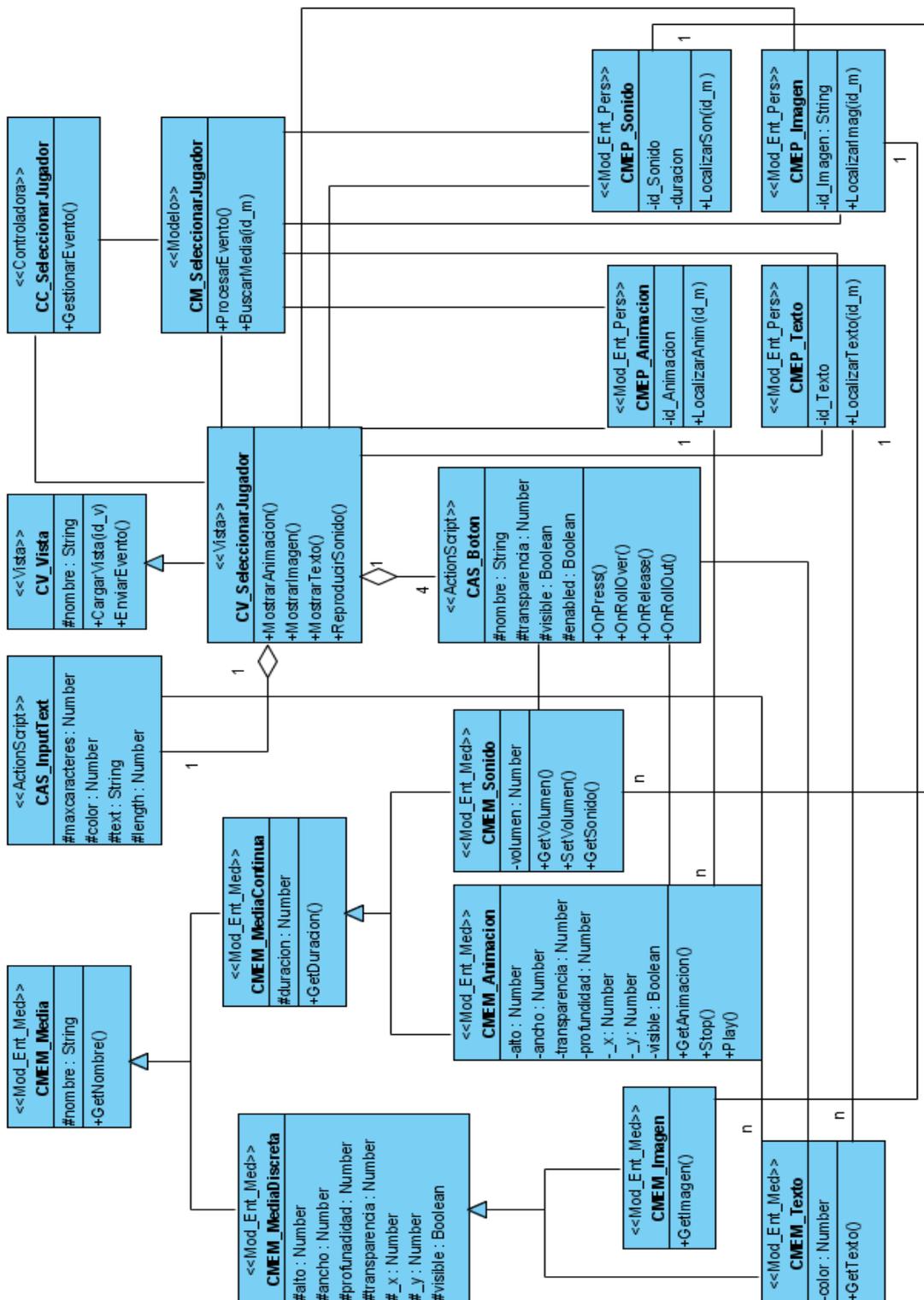
Anexo 41. Diagrama de Clases de la Vista de Presentación Creditos



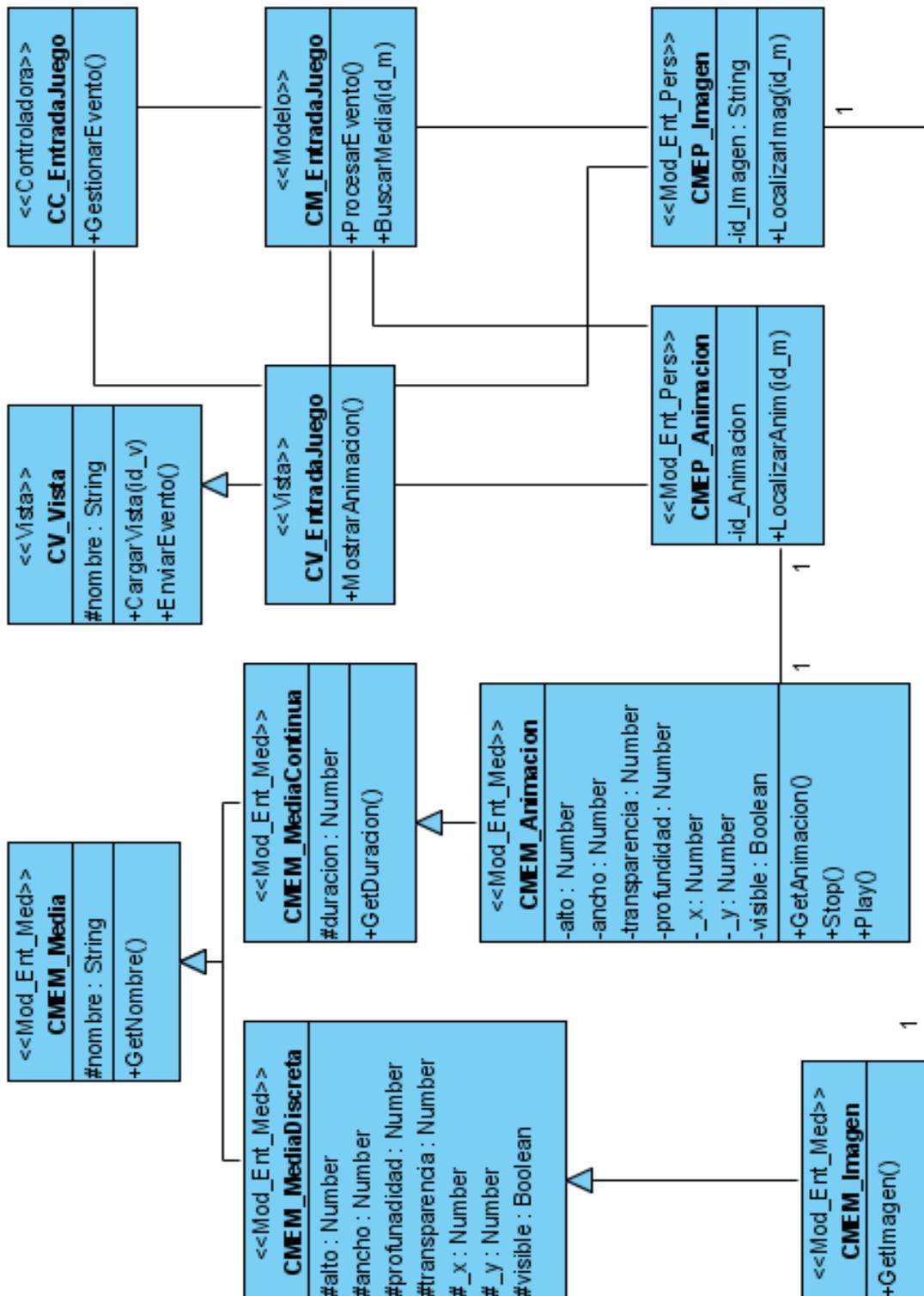
Anexo 42. Diagrama de Clases de la Vista de Presentación CargarPartida



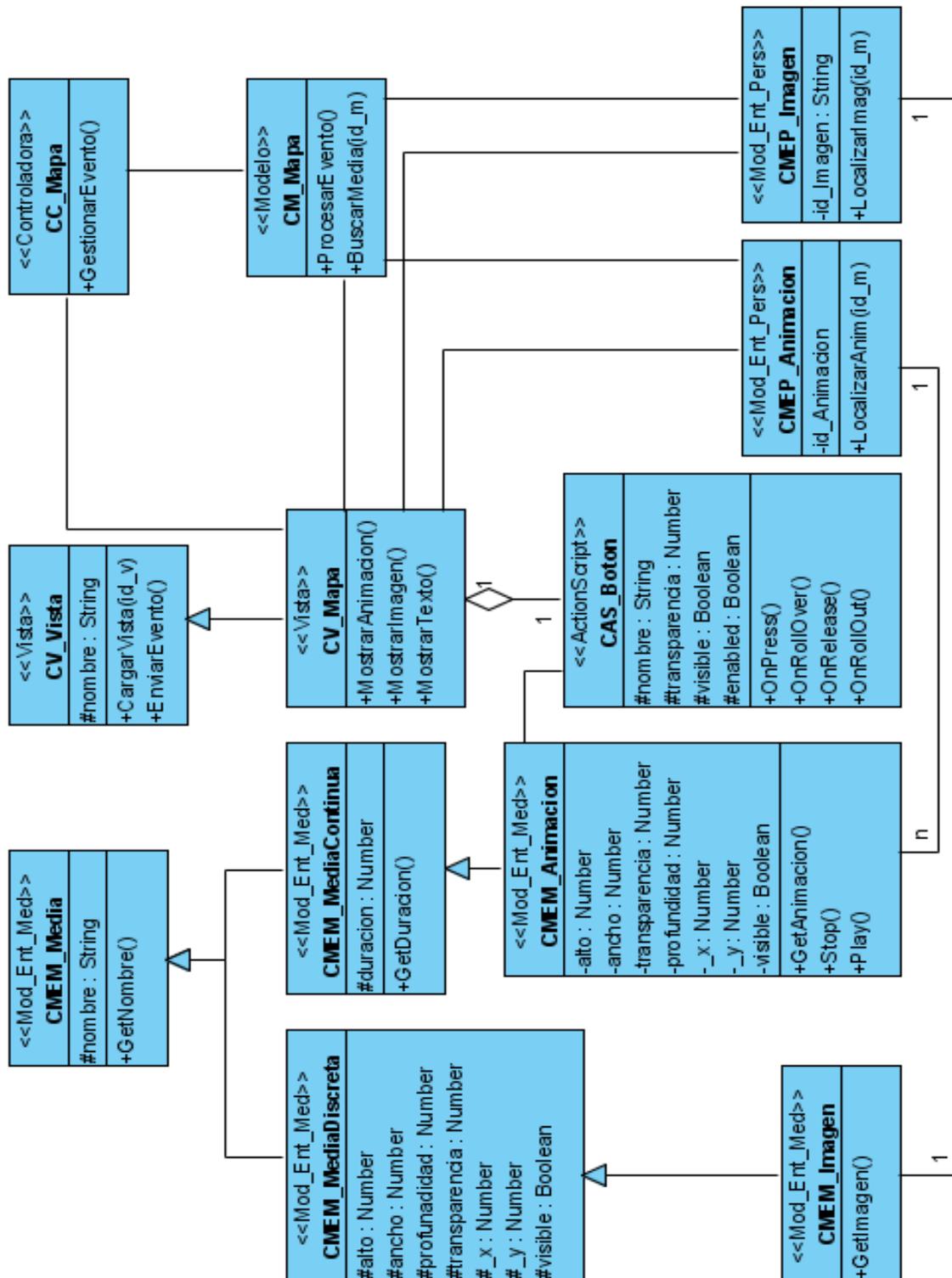
Anexo 43. Diagrama de Clases de la Vista de Presentación ConfirmarSalida



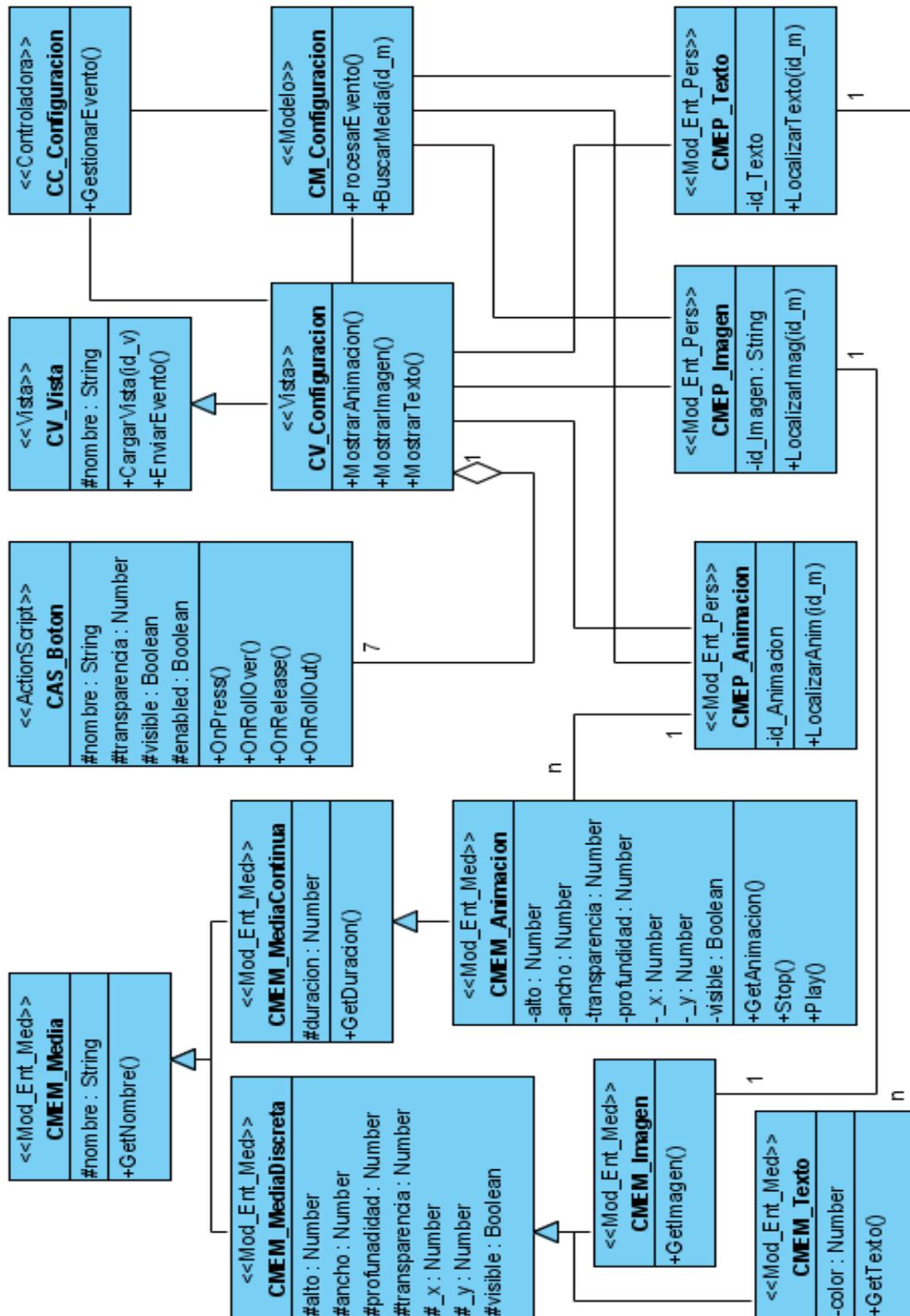
Anexo 44. Diagrama de Clases de la Vista de Presentación SeleccionarJugador



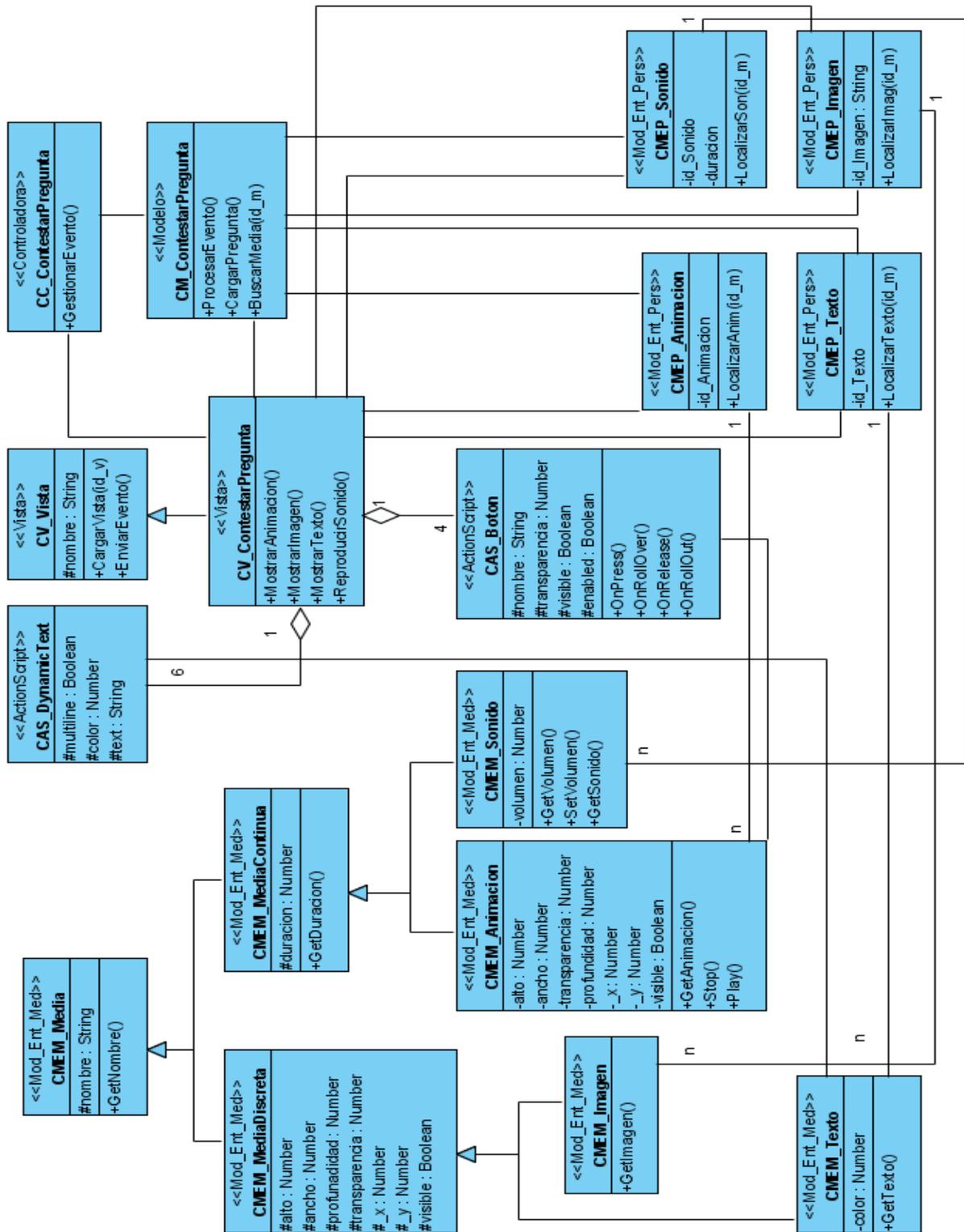
Anexo 45. Diagrama de Clases de la Vista de Presentación EntradaJuego



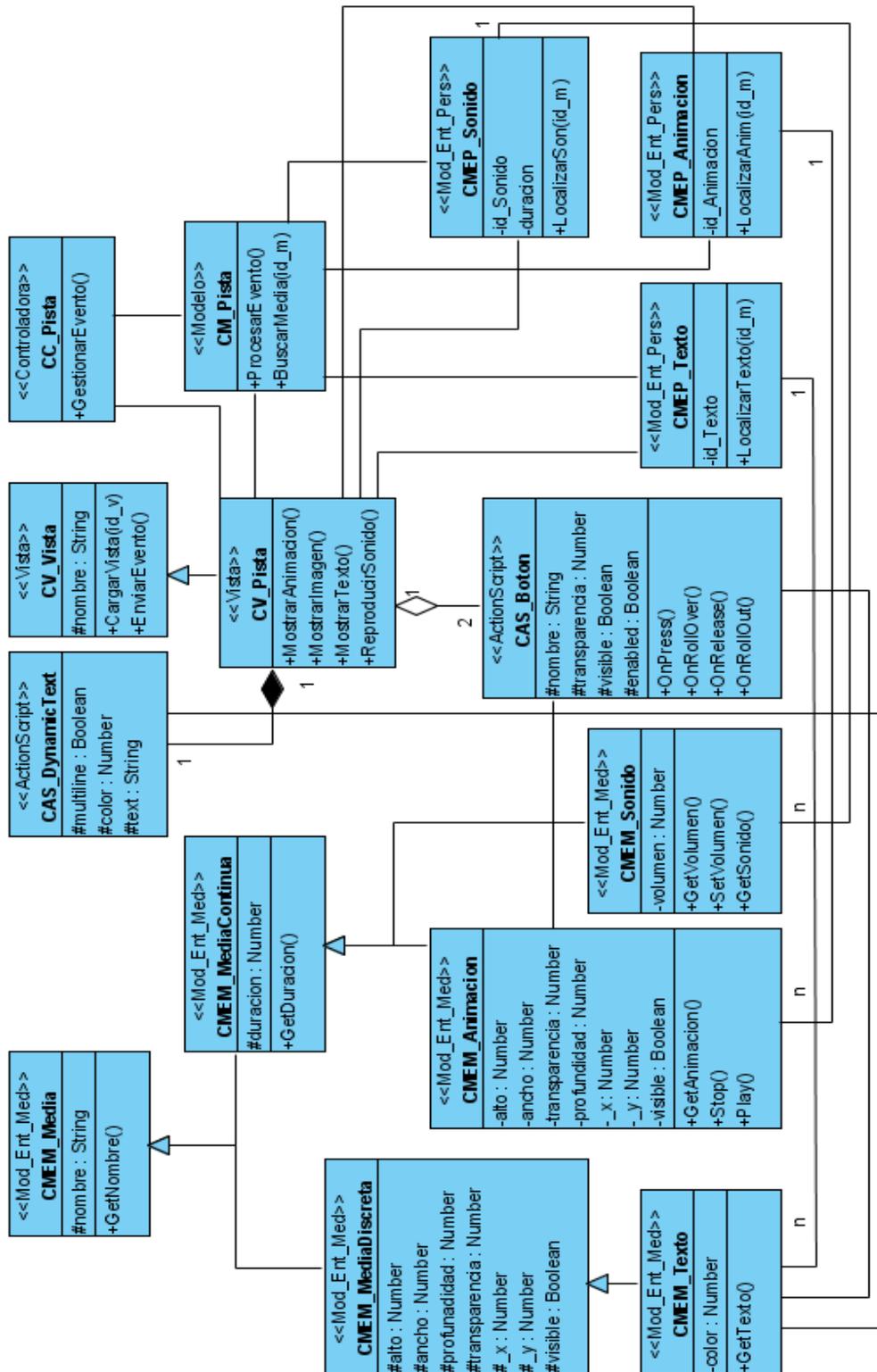
Anexo 46. Diagrama de Clases de la Vista de Presentación Mapa



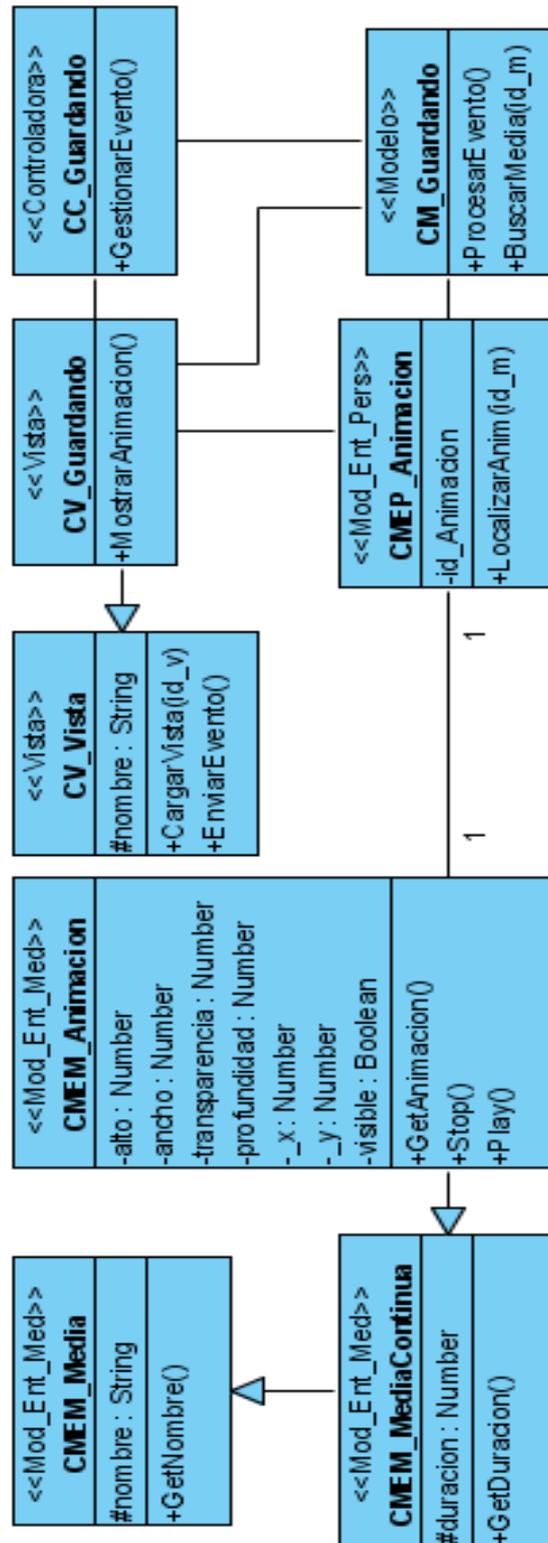
Anexo 47. Diagrama de Clases de la Vista de Presentación Configuración



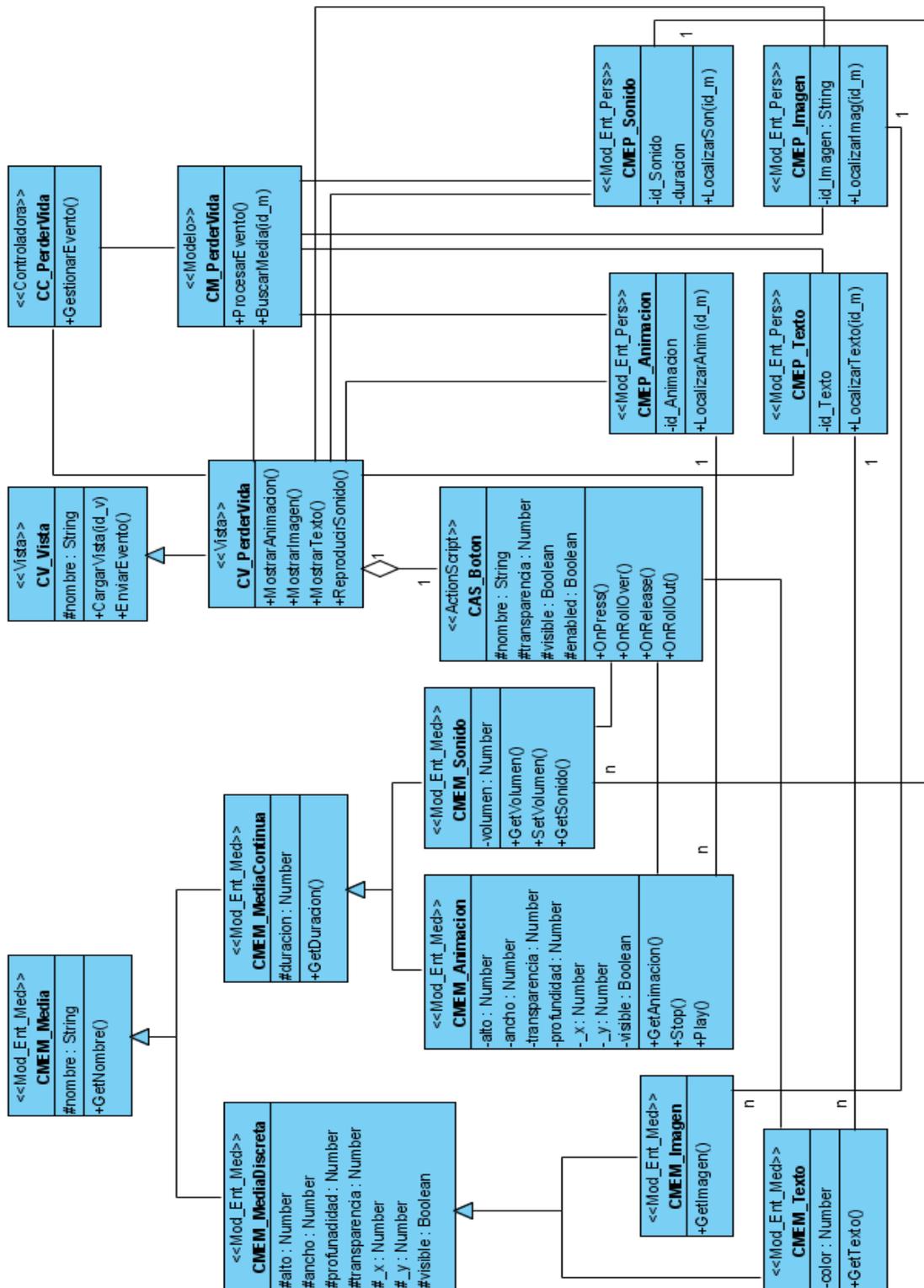
Anexo 48. Diagrama de Clases de la Vista de Presentación ContestarPregunta



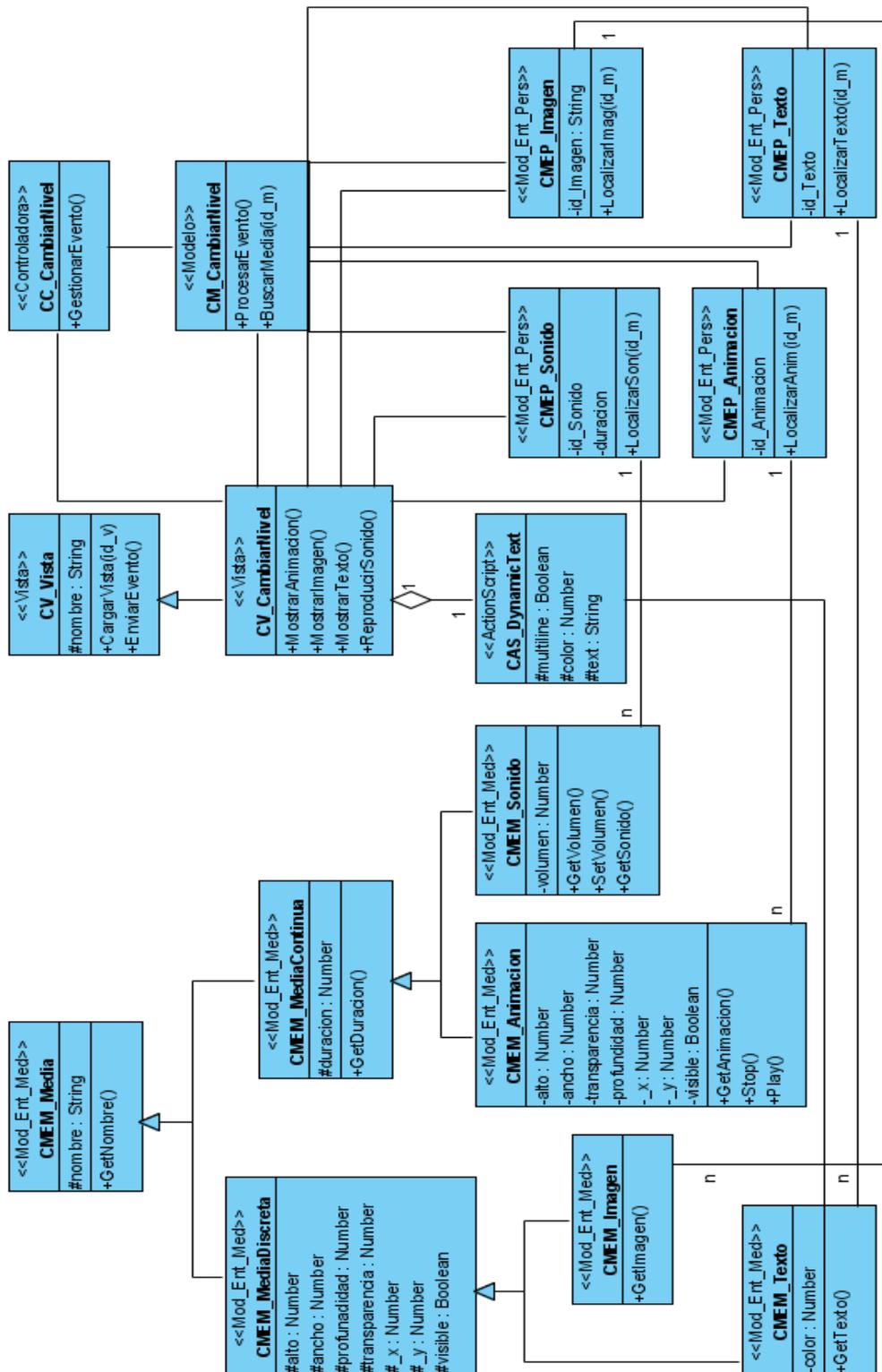
Anexo 49. Diagrama de Clases de la Vista de Presentación Pista



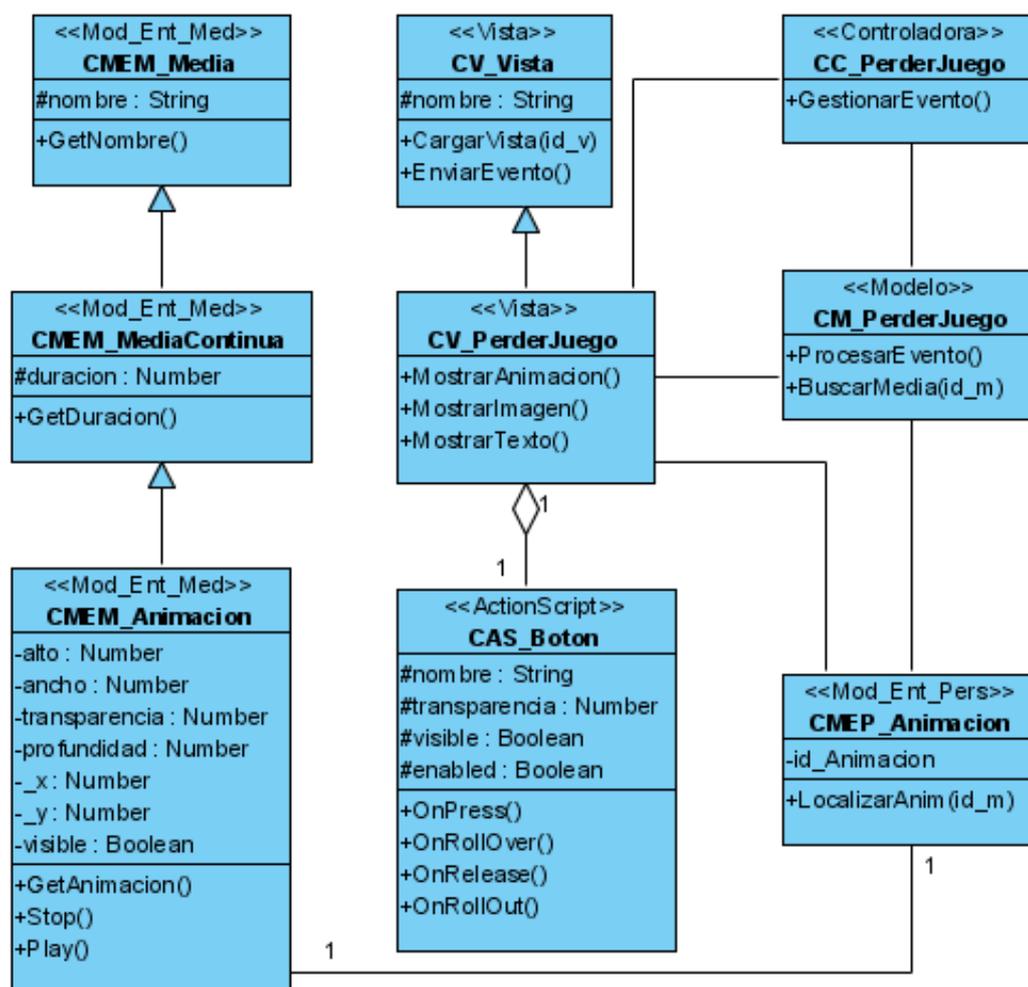
Anexo 50. Diagrama de Clases de la Vista de Presentación Guardando



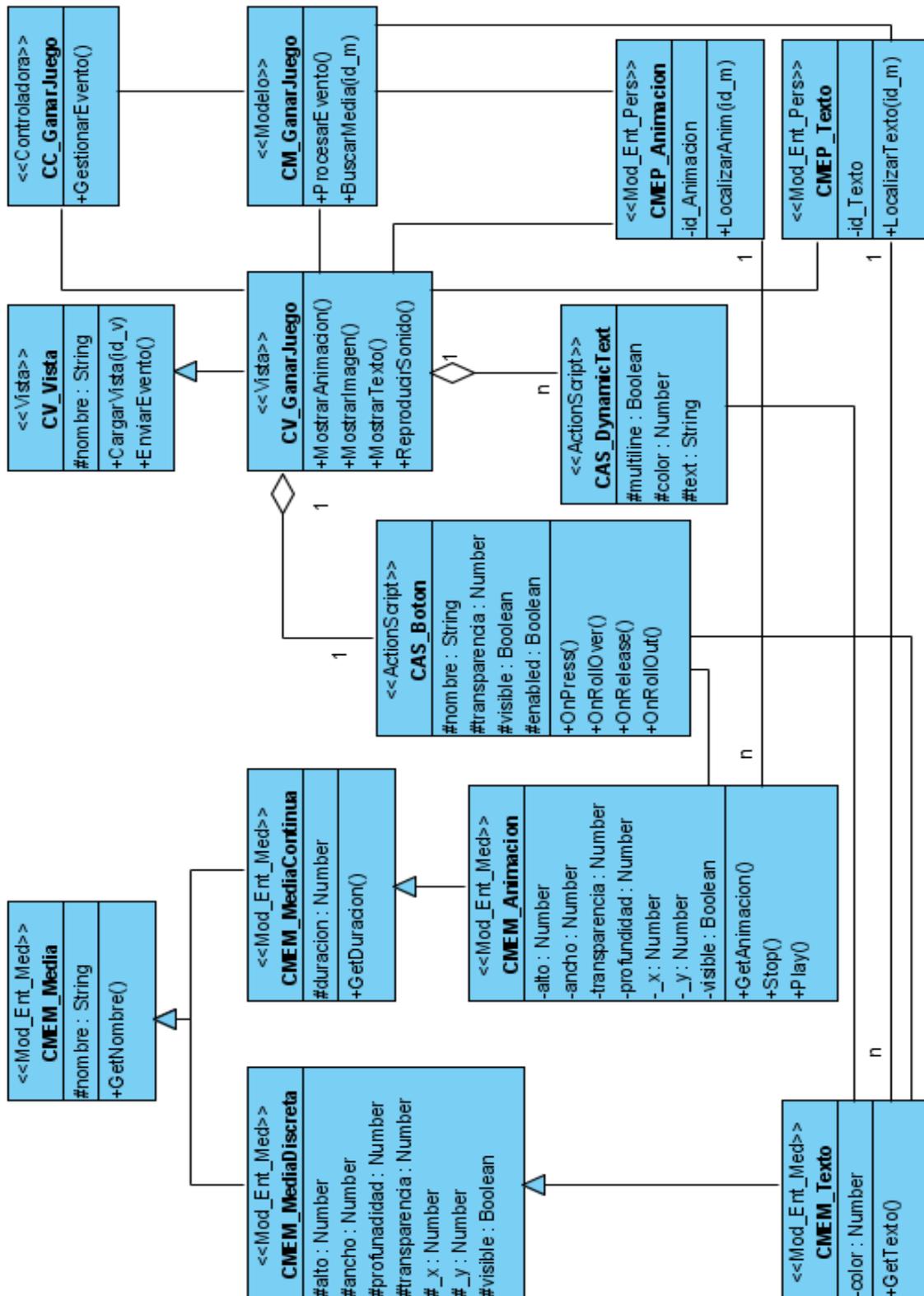
Anexo 51. Diagrama de Clases de la Vista de Presentación PerderVida



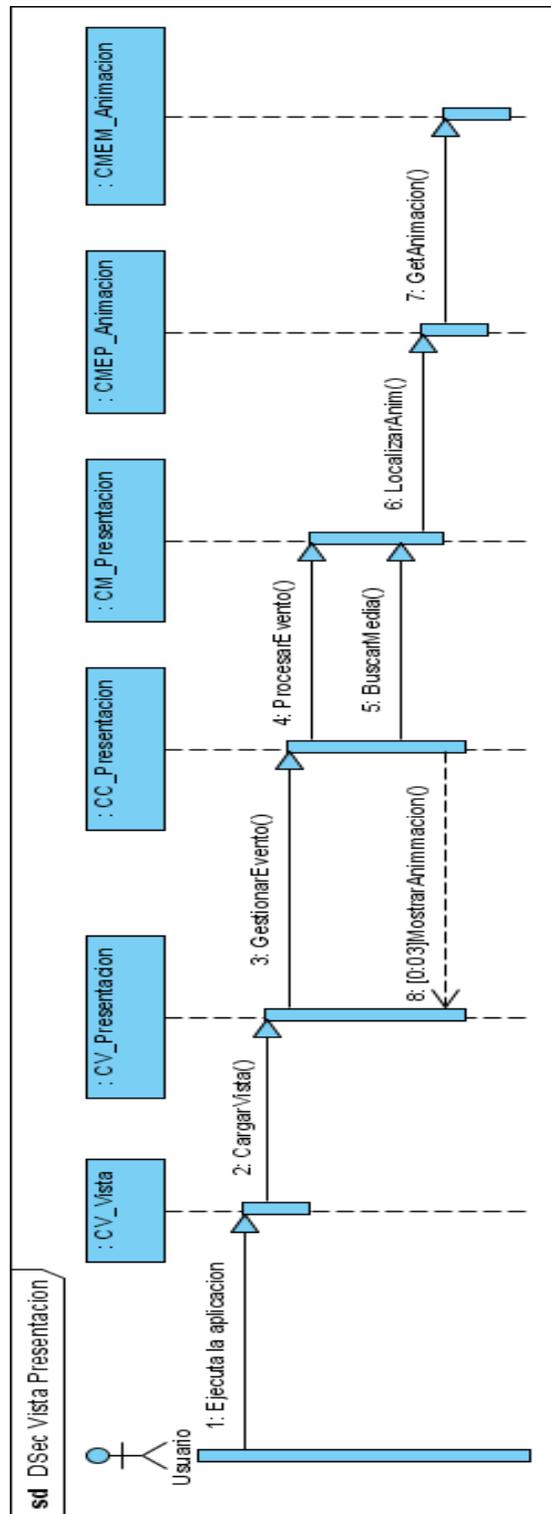
Anexo 52. Diagrama de Clases de la Vista de Presentación CambiarNivel



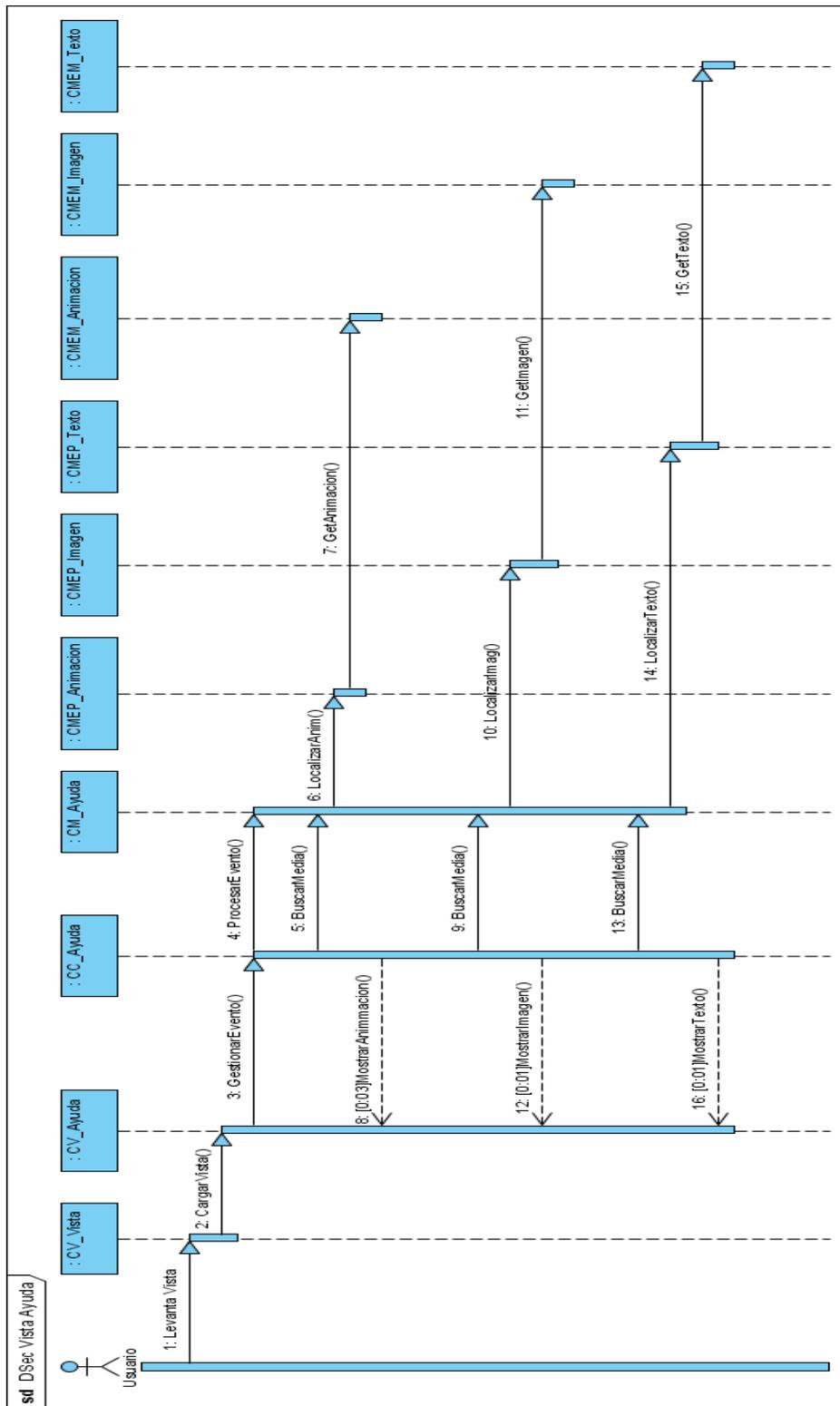
Anexo 53. Diagrama de Clases de la Vista de Presentación PerderJuego



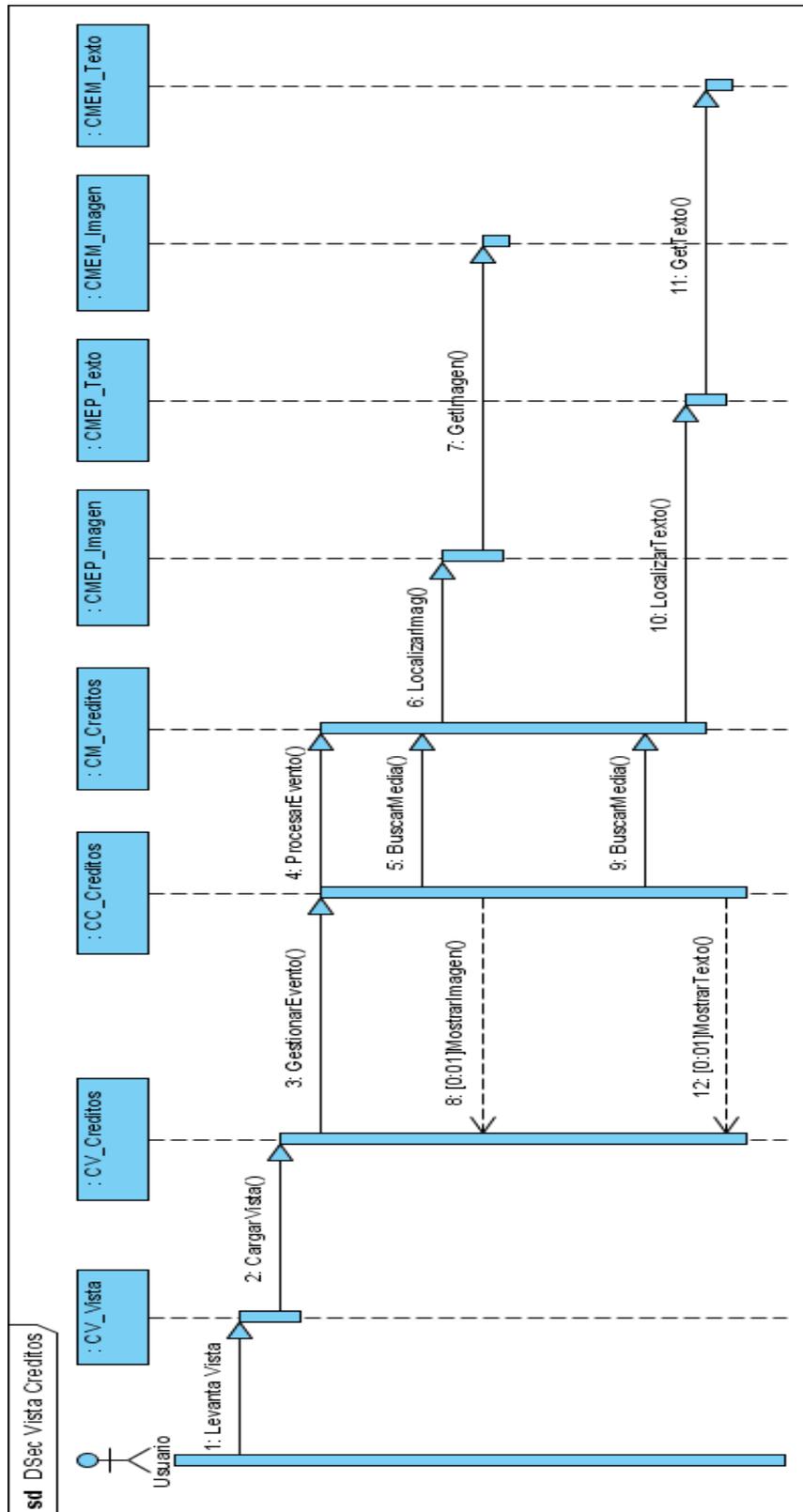
Anexo 54. Diagrama de Clases de la Vista de Presentación GanarJuego



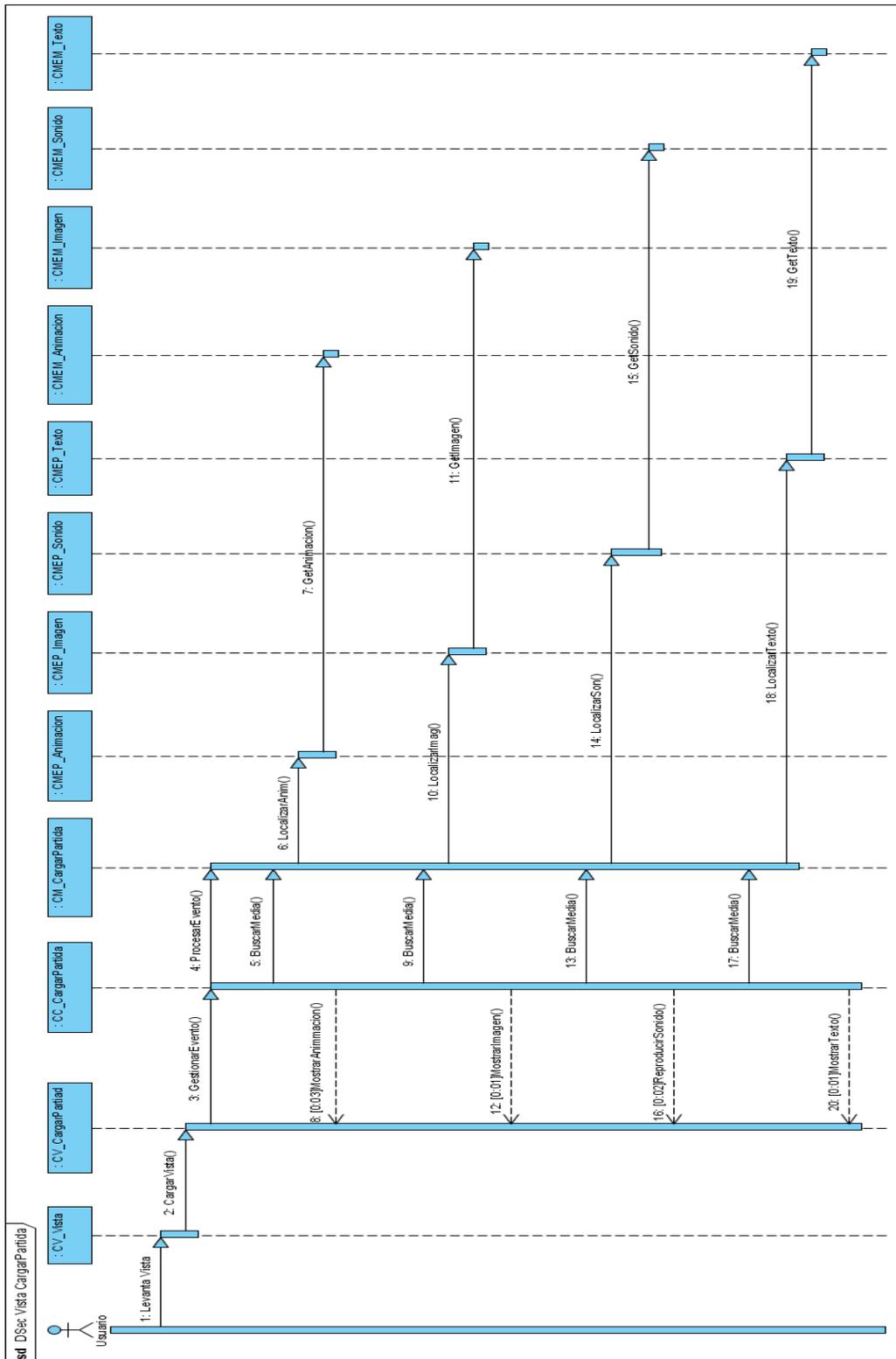
Anexo 55. Diagrama de Secuencias de la Vista de Presentación Presentacion



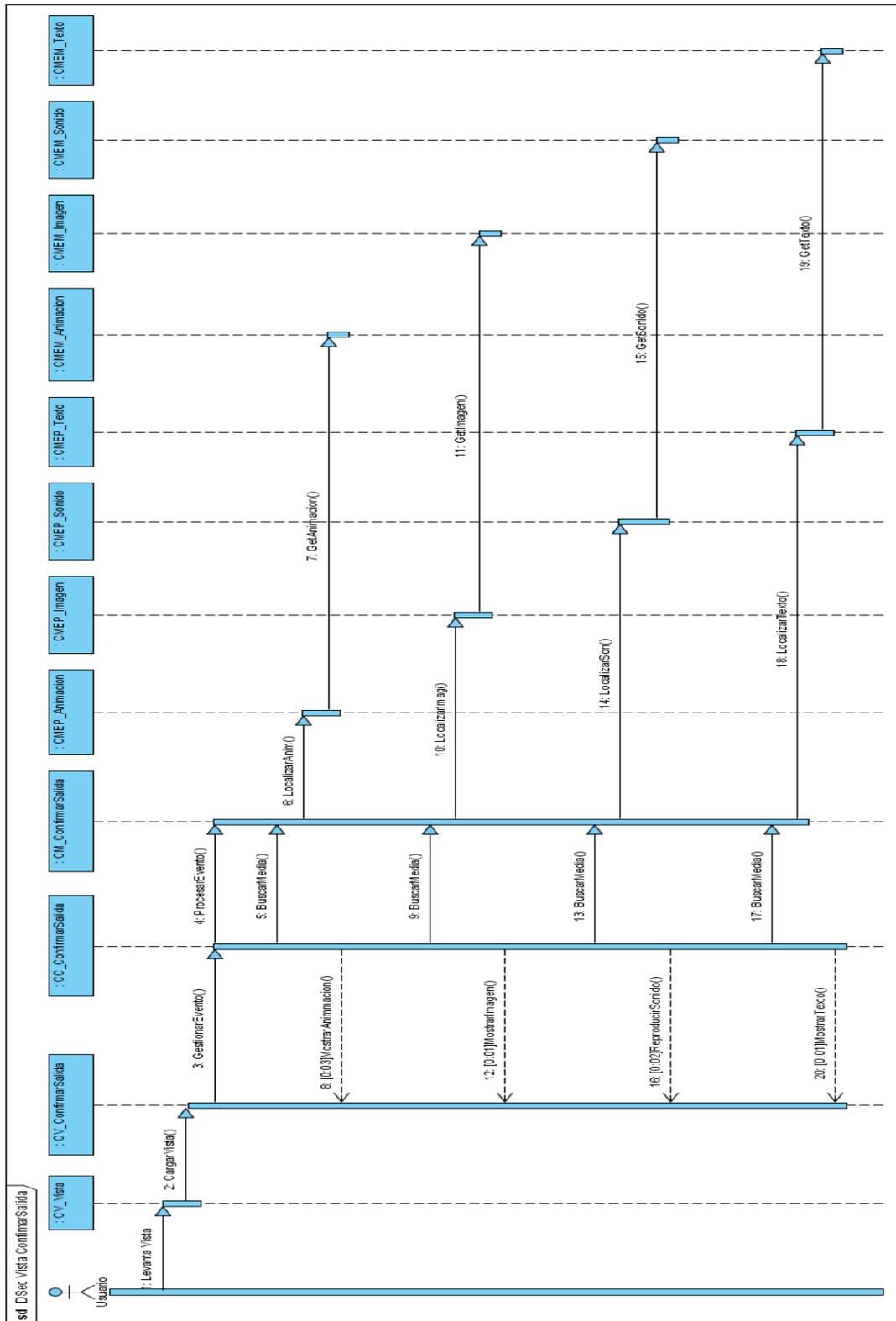
Anexo 56. Diagrama de Secuencias de la Vista de Presentación Ayuda



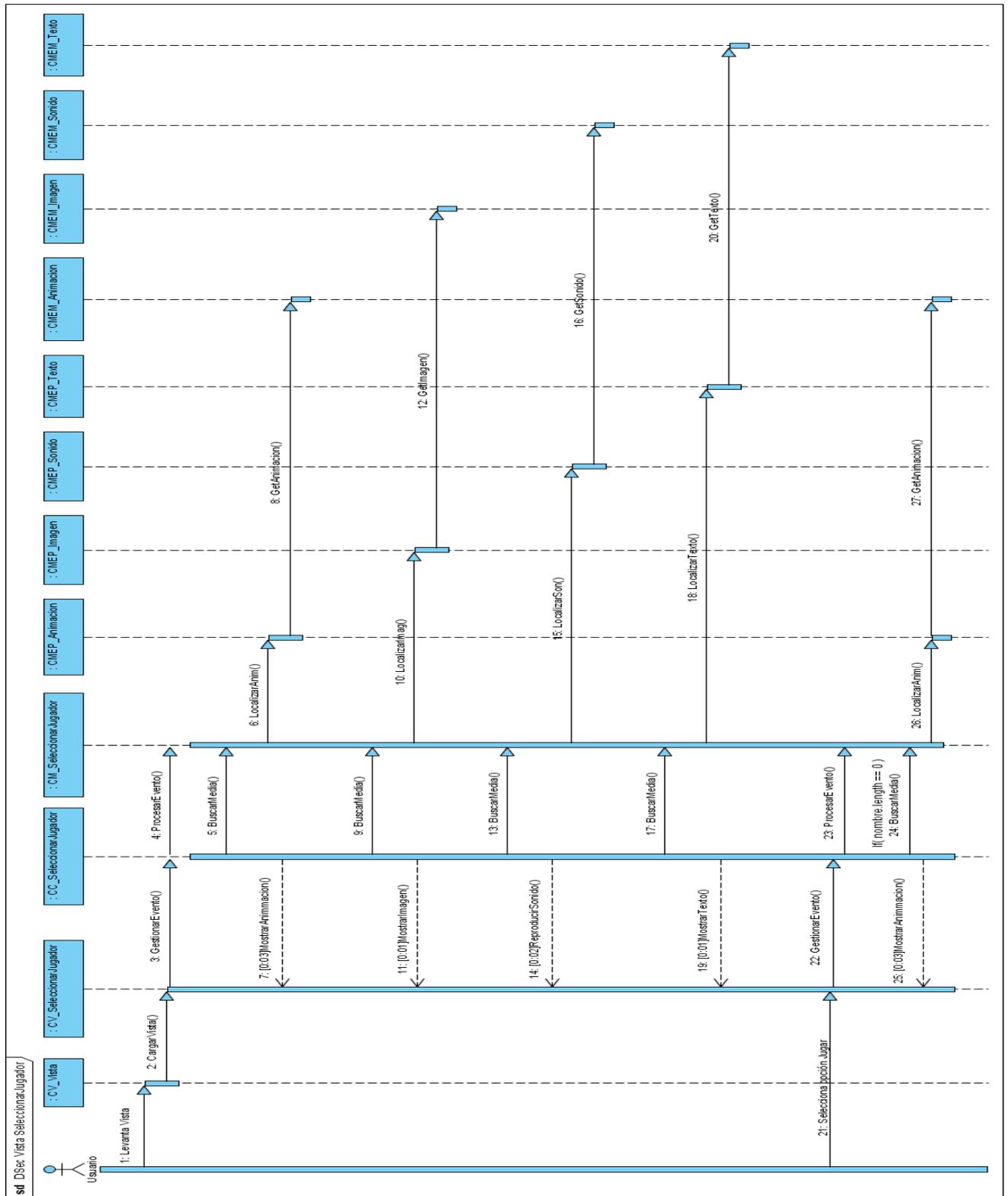
Anexo 57. Diagrama de Secuencias de la Vista de Presentación Creditos



Anexo 58. Diagrama de Secuencias de la Vista de Presentación CargarPartida

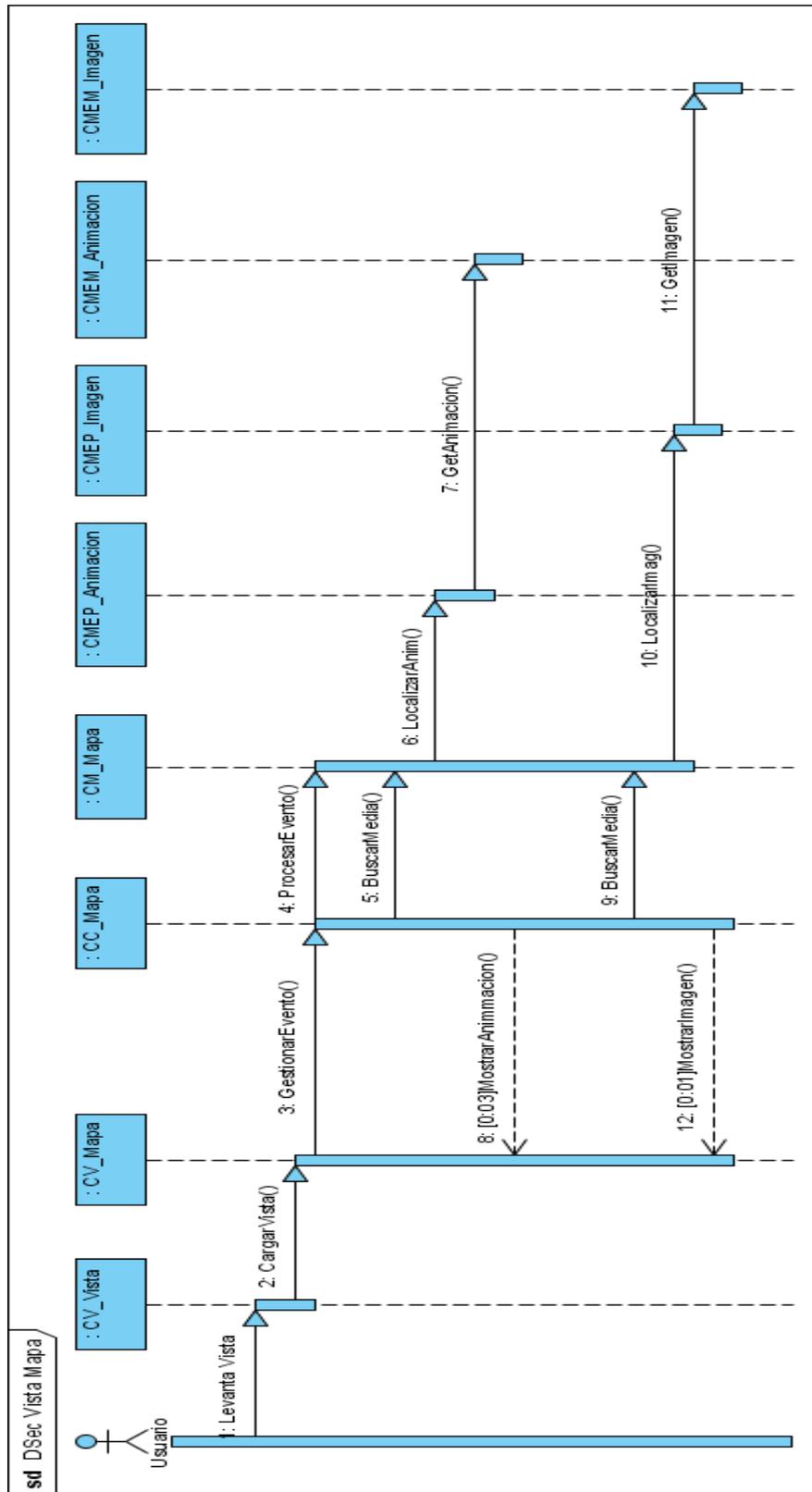


Anexo 59. Diagrama de Secuencias de la Vista de Presentación ConfirmarSalida

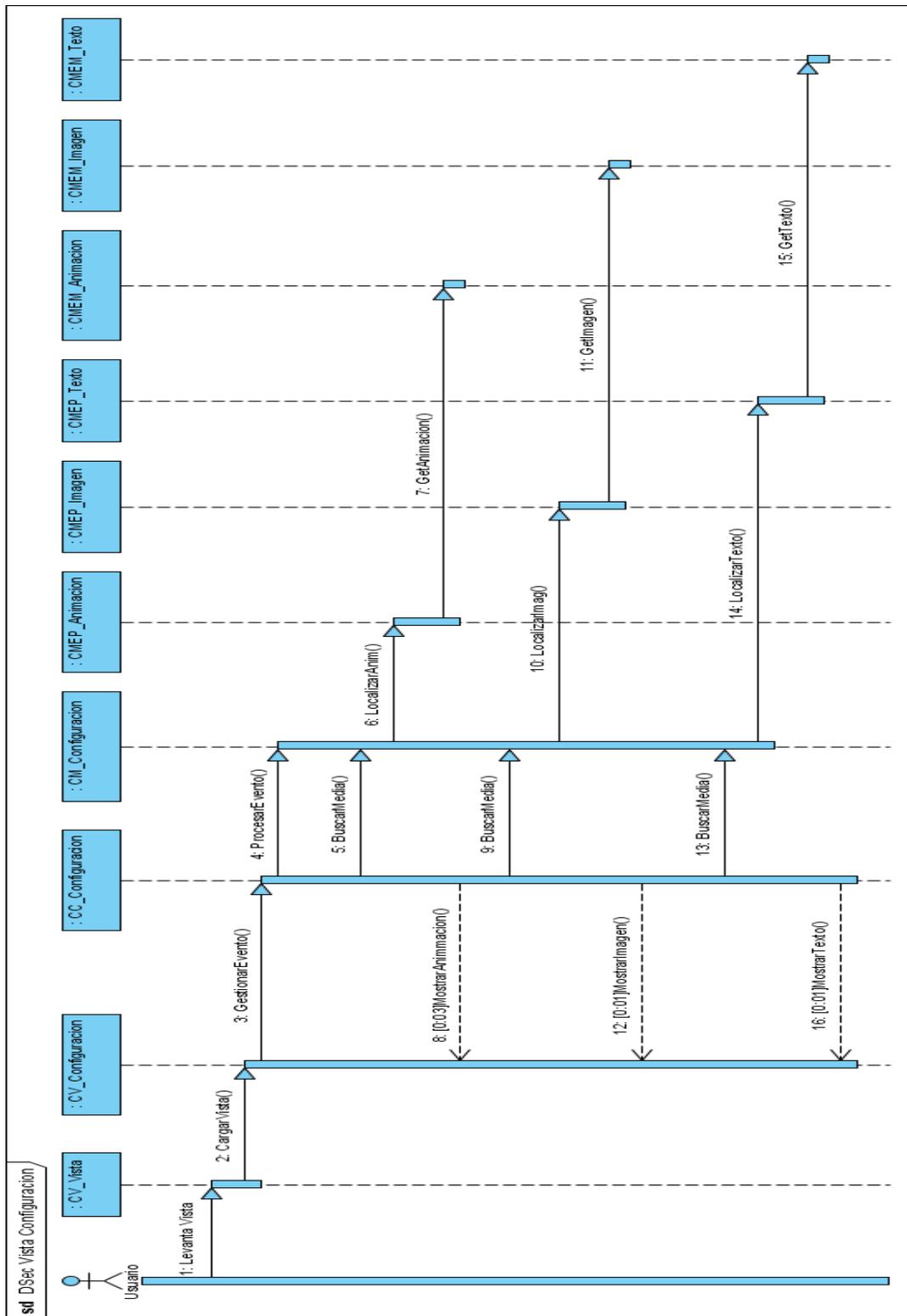


Anexo 60. Diagrama de Secuencias de la Vista de Presentación SeleccionarJugador

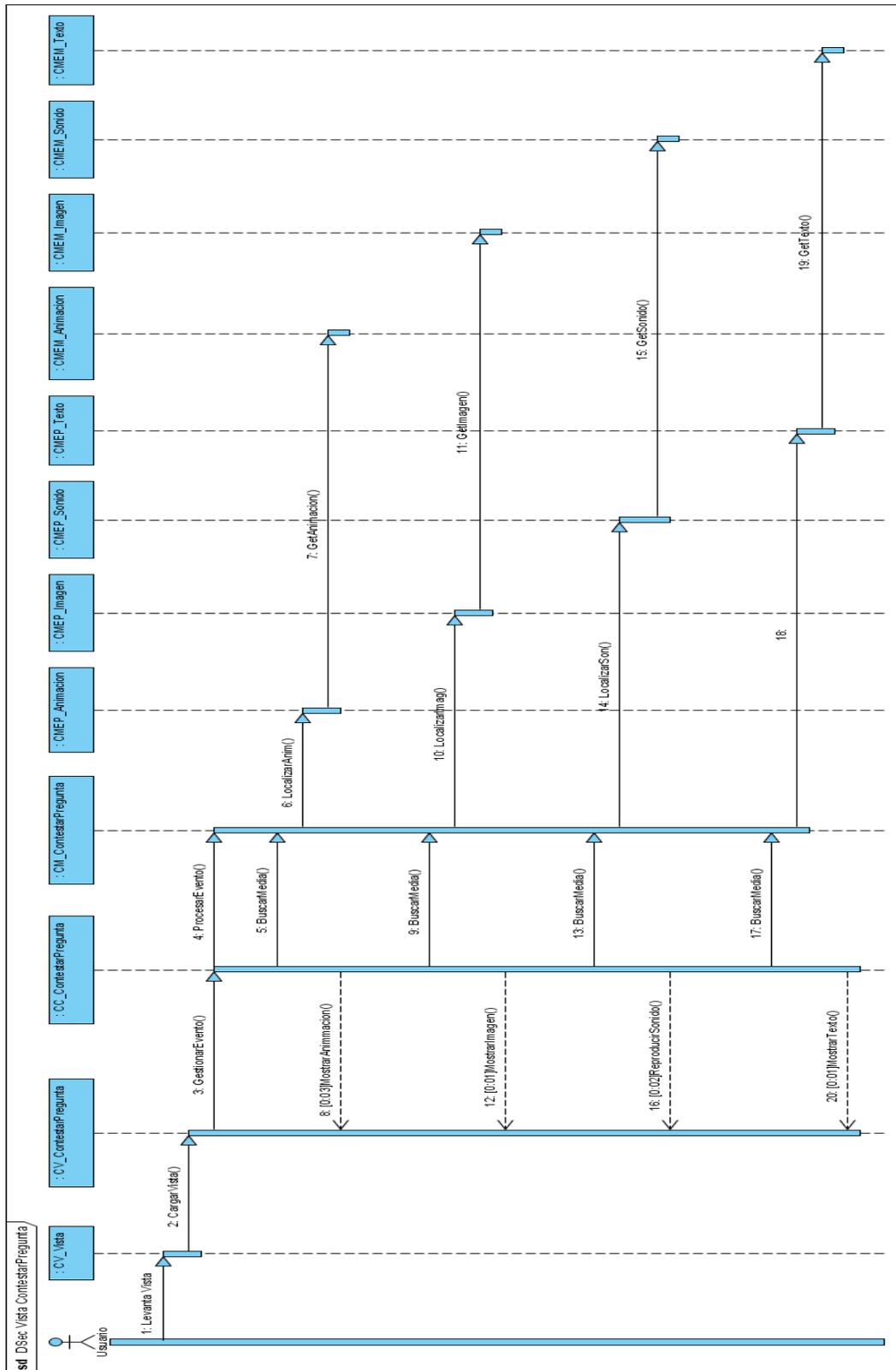




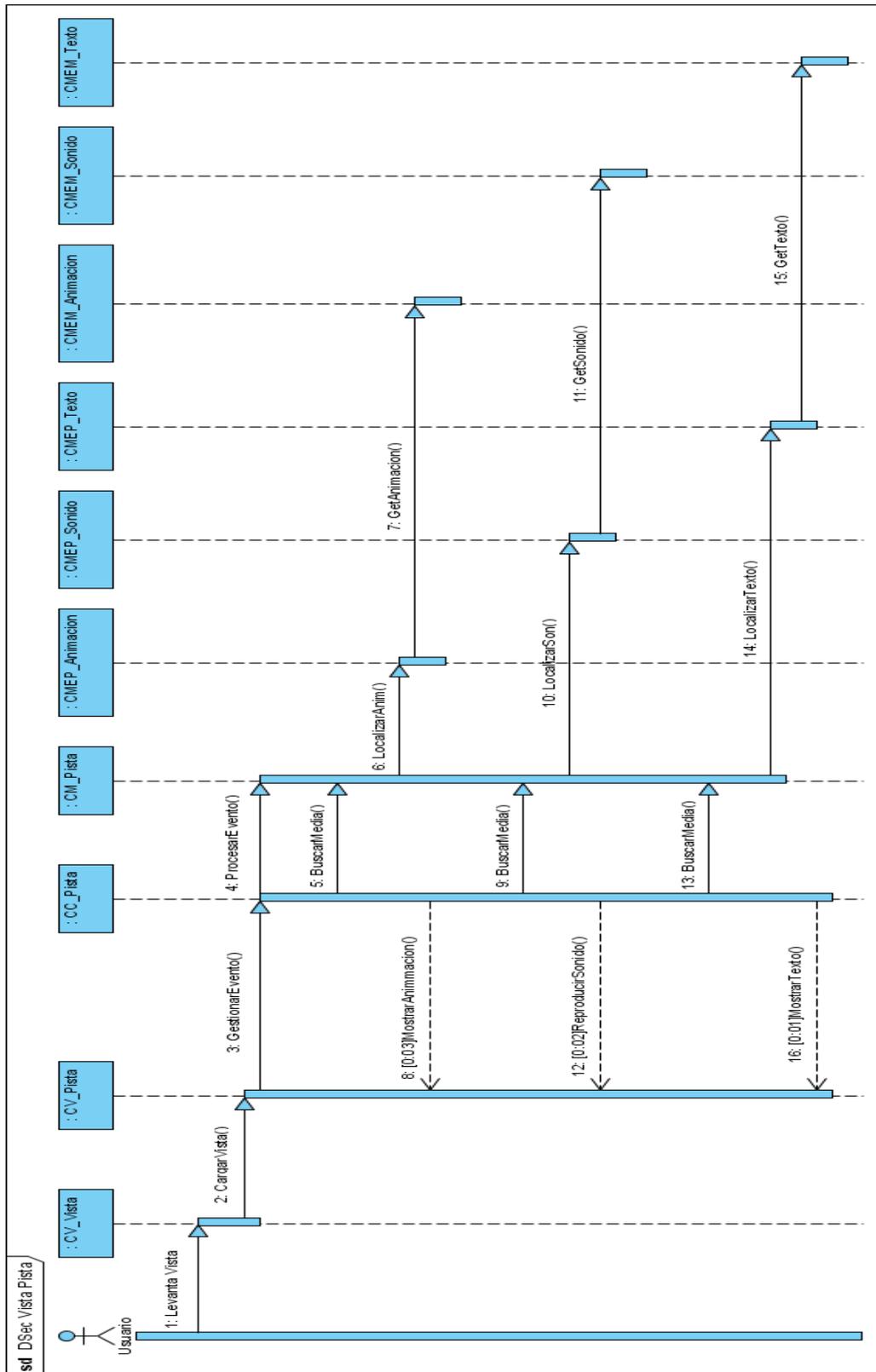
Anexo 62. Diagrama de Secuencias de la Vista de Presentación Mapa



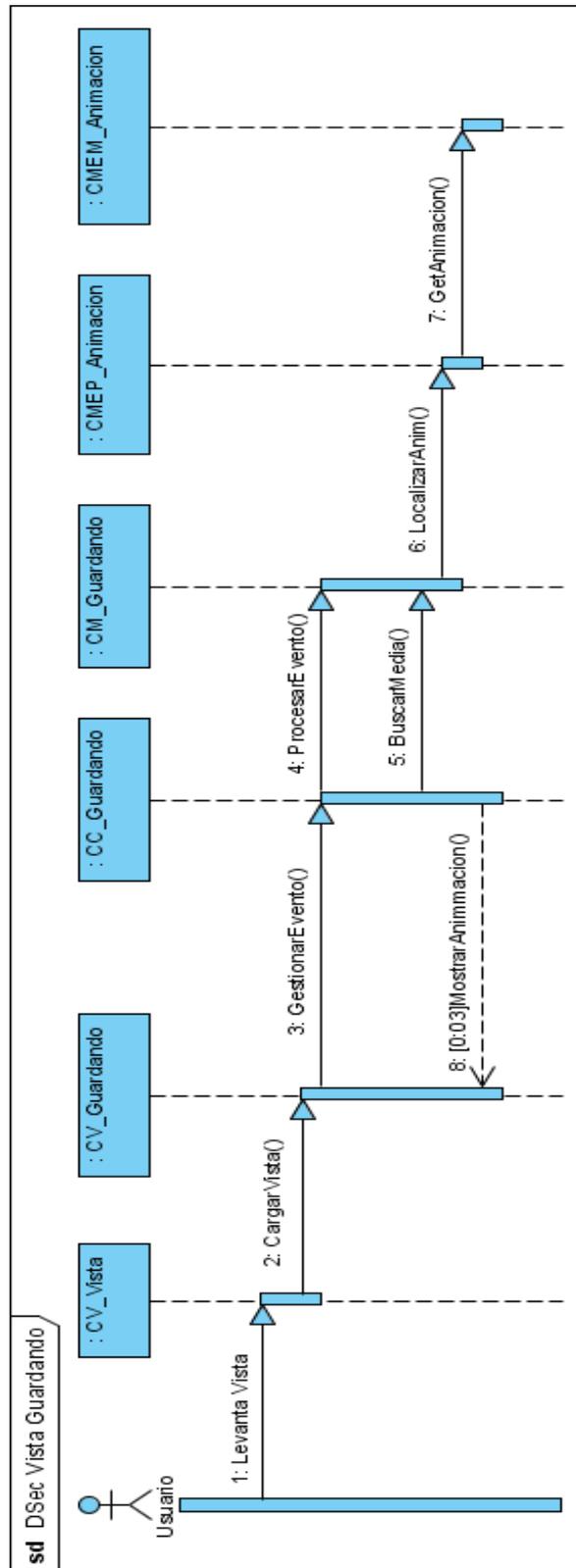
Anexo 63. Diagrama de Secuencias de la Vista de Presentación Configuración



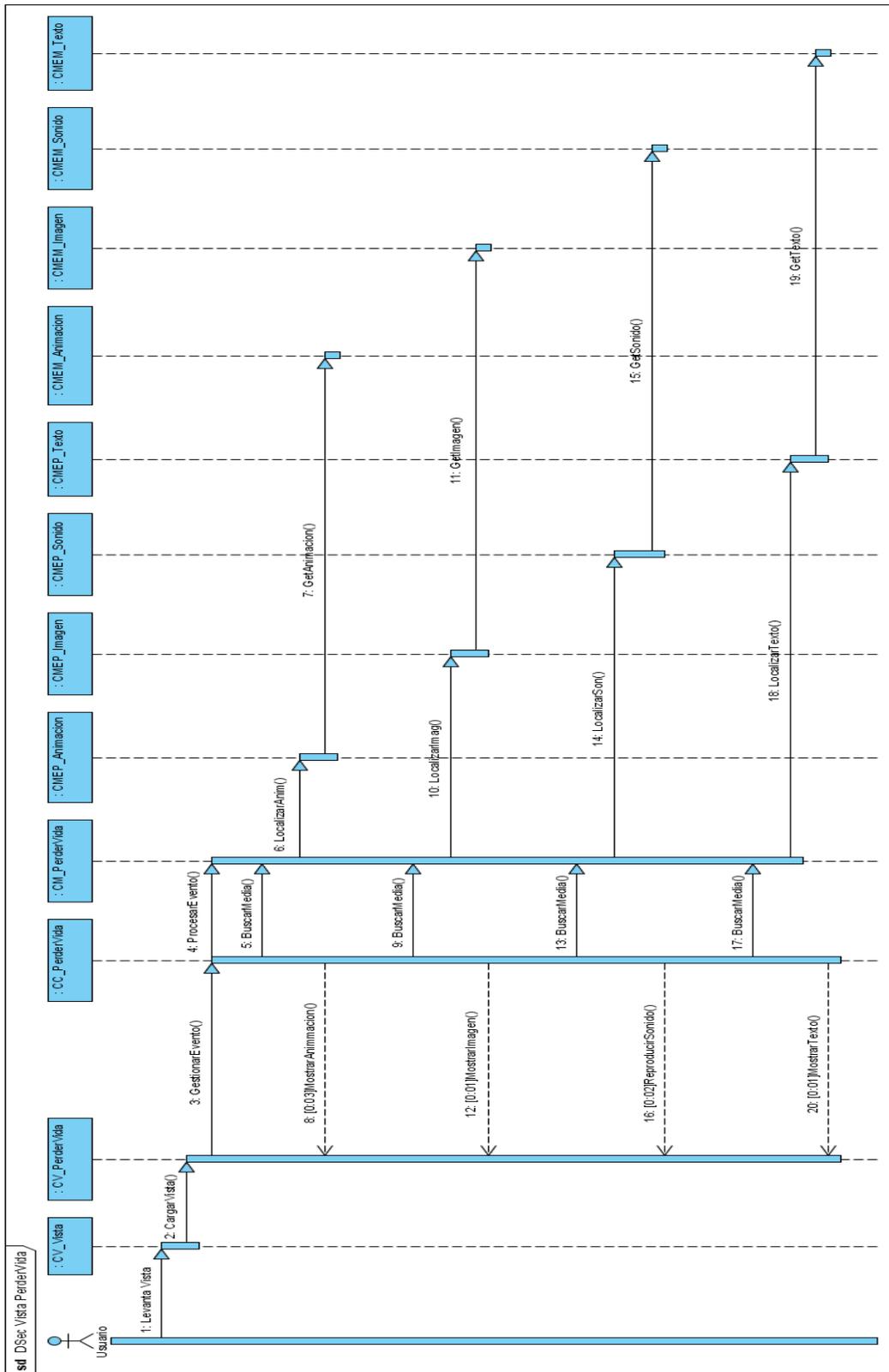
Anexo 64. Diagrama de Secuencias de la Vista de Presentación ContestarPregunta



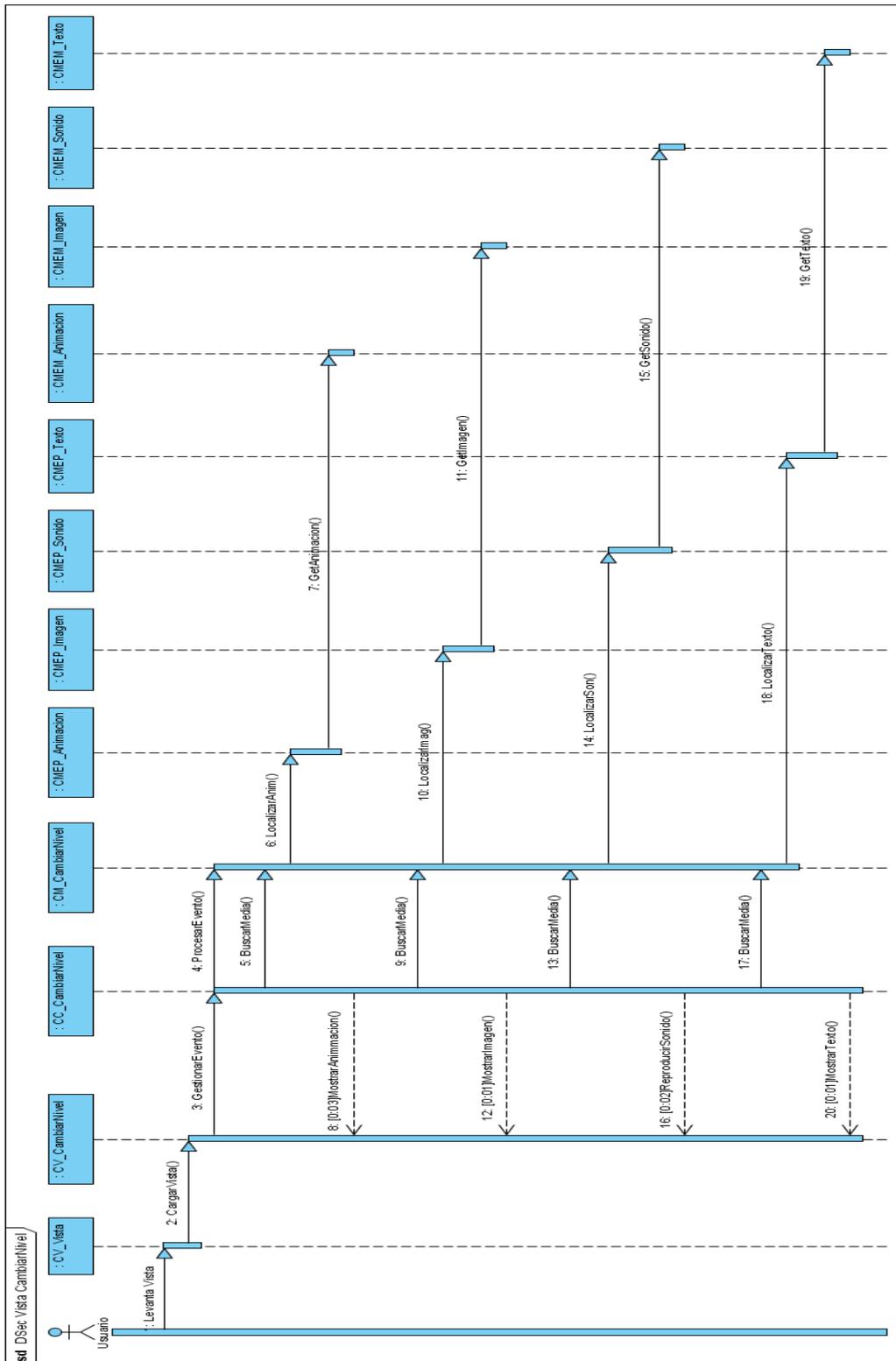
Anexo 65. Diagrama de Secuencias de la Vista de Presentación Pista



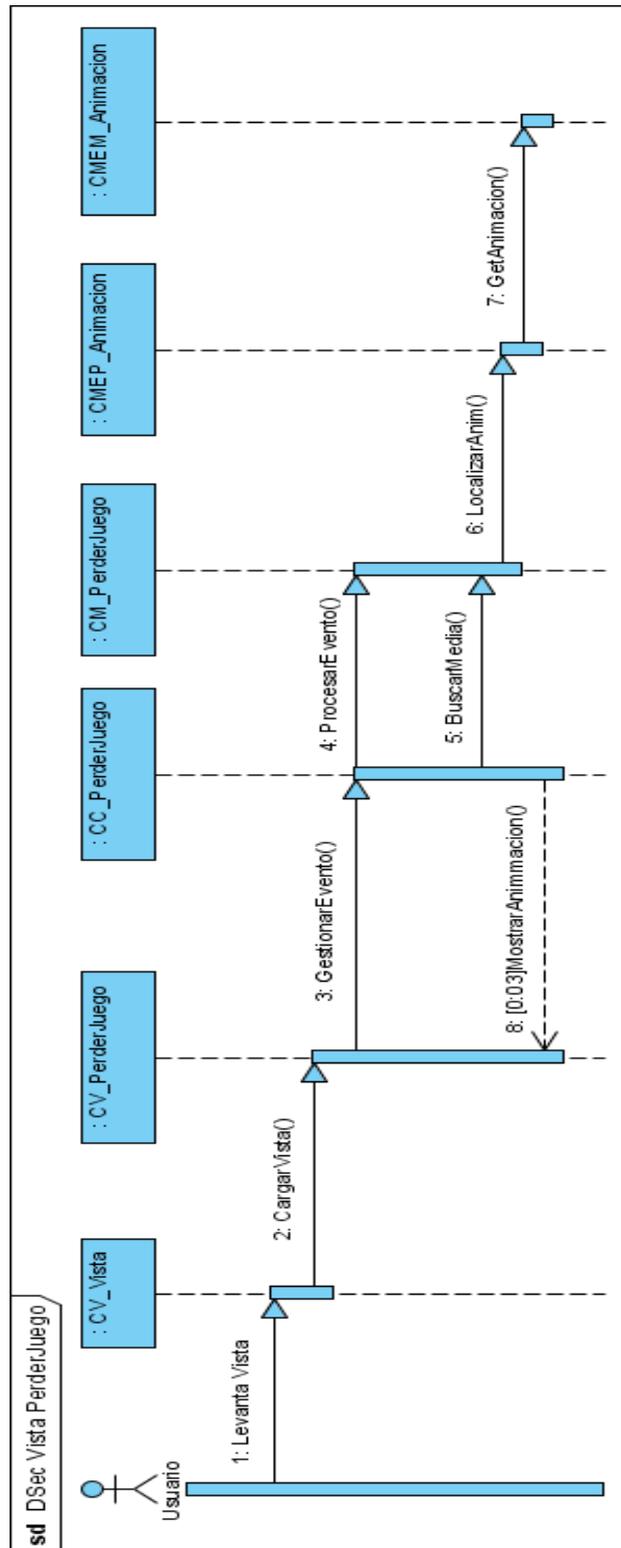
Anexo 66. Diagrama de Secuencias de la Vista de Presentación Guardando



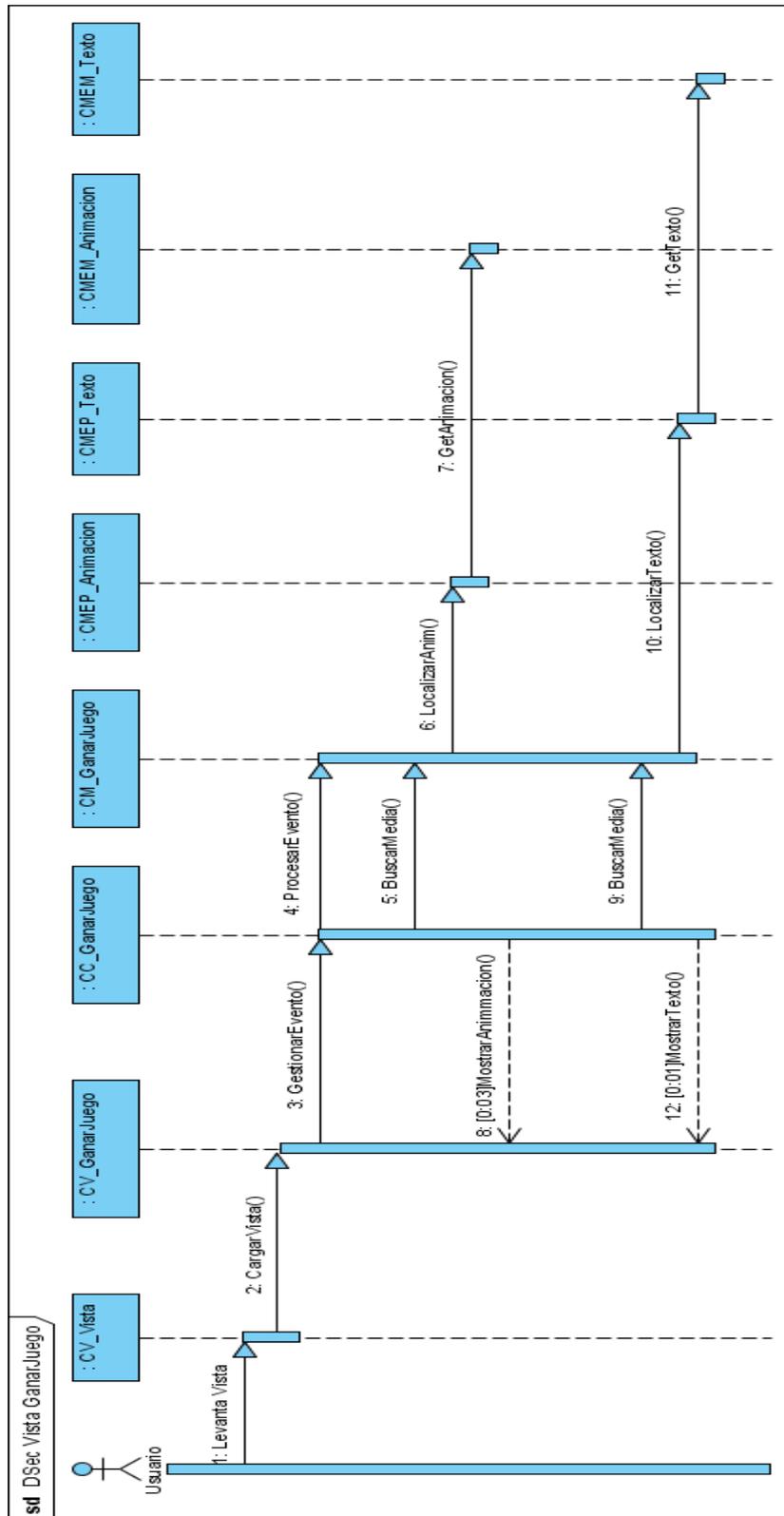
Anexo 67. Diagrama de Secuencias de la Vista de Presentación PerderVida



Anexo 68. Diagrama de Secuencias de la Vista de Presentación CambiarNivel



Anexo 69. Diagrama de Secuencias de la Vista de Presentación PerderJuego



Anexo 70. Diagrama de Secuencias de la Vista de Presentación GanarJuego

## GLOSARIO

1. **Actor:** Cualquier individuo, grupo, entidad, organización, máquina o sistema que interactúa con la aplicación.
2. **AM:** Amplitud Modulada.
3. **Animación:** Representación sucesiva de imágenes que produce sensación de movimiento.
4. **Artefacto:** Pieza de información utilizada o producida por un proceso de desarrollo de software como un documento externo o el producto de un trabajo.
5. **Botones:** Permite la interactividad en las aplicaciones informáticas, donde se pueden agregar una serie de instrucciones o acciones que el usuario podrá utilizar generando un evento.
6. **Clips de Película:** Símbolo de creado por Flash que puede contener otros clips de película, botones o gráficos, haciendo de esta manera objetos con mayor complejidad e interacción.
7. **Complemento de ambientación:** Objeto que forma parte del entorno del juego con los que no interactúa el personaje. Tiene como objetivo tributar a la ambientación de juego.
8. **Diagrama:** Representación gráfica en la que se muestran las relaciones entre las diferentes partes de un conjunto o sistema.
9. **Displays:** Dispositivo de ciertos aparatos electrónicos que permite mostrar información al usuario.
10. **Entidades:** Objetos concretos o abstractos que presentan interés para el sistema.
11. **Escenario:** Espacio destinado para la representación de elementos del juego.
12. **EXE:** Extensión que se refiere a un archivo ejecutable.
13. **Filtros:** Efectos visuales generados por un software de edición gráfica.
14. **Flash Player:** Aplicación que permite reproducir archivos SWF.
15. **Gráficos:**
  - a. Cualquier imagen generada por una computadora.
  - b. Símbolo de creado por Flash que se corresponden con imágenes o vectores estáticos.

16. **Hardware:** Componente físico tecnológico que trabaja o interactúa de algún modo con la computadora.
17. **Hipermedia:** Conjunto de métodos o procedimientos para escribir, diseñar, o componer contenidos que tengan texto, video, audio, mapas u otros medios, y que además tenga la posibilidad de interactuar con los usuarios.
18. **Hipertextos:** Texto que en la pantalla de una computadora conduce a su usuario a otro texto relacionado. La forma más habitual de hipertexto en documentos es la de hipervínculos.
19. **Interfaz:** Medio a través del cual un usuario interactúa con un sistema.
20. **Lenguaje de programación:** Lenguaje que puede ser utilizado para controlar el comportamiento de una computadora.
21. **Mascota:** Personaje que actúa como guía para la realización de las diferentes tareas del juego.
22. **Medios de enseñanza:** Herramientas que se utilizan para apoyar la enseñanza.
23. **Metodología de desarrollo:** Conjunto de normas y métodos que guían un proceso de desarrollo de software.
24. **Nivel:** Sección del juego que corresponde a una porción del trayecto total a recorrer.
25. **OCL:** Lenguaje notacional que permite escribir restricciones sobre modelos de objetos.
26. **Pantalla:** Es la agrupación visual de elementos de medias contenidas en una vista determinada.
27. **Pantalla completa:** Visualización de la aplicación ocupando todo el espacio disponible en pantalla.
28. **Personaje:** Ser simbólico con el que se identifica el usuario para realizar diferentes acciones y cumplir las misiones del juego.
29. **Plataforma:**
  - a. Principio, ya sea de hardware o software, sobre el cual un programa puede ejecutarse.
  - b. Elevación sobre el suelo, donde se pueden posicionar objetos y personajes en el juego.
30. **Símbolos:** Objetos creados en Flash que sirven para realizar un sin número de tareas, ya que poseen distintos comportamientos, los cuales ayudan a construir aplicaciones interactivas.

31. **Sistema operativo:** es el software que controla la ejecución de todas las aplicaciones y de los programas de software de sistema.
32. **Software:** Programa o aplicación para realizar tareas específicas en una computadora.
33. **Subsistema:** Es un agrupamiento semántico útil de clases u otros subsistemas.
34. **SWF:** Formato de archivo de gráficos vectoriales. Los archivos SWF pueden ser creados fundamentalmente por programas como Flash.
35. **Tecnología:** Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico.
36. **Windows, Macintosh, Unix, PDAs:** Sistemas operativos.