



**Universidad de las Ciencias  
Informáticas**

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

**Título: Análisis, diseño e implementación de la  
versión 2.0 de los juegos del primer ciclo de  
la Colección Multisaber**

**Autores: Ramiro Agras Hechavarría  
Danay Delgado Del Rio**

**Tutores: Ing. Ismael Armando Nodarse Mora  
Ing. Ana Margarita Rojas Riverón**

**Ciudad de la Habana, junio de 2010  
“Año 52 de la Revolución”**

# DECLARACIÓN DE AUTORÍA

---

## Declaración de Autoría

Declaramos ser los únicos autores del presente trabajo "Análisis, diseño e implementación de la versión 2.0 de los juegos del primer ciclo de la Colección Multisaber" y autorizamos a la Facultad 8 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autores:

\_\_\_\_\_

Ramiro Agras Hechavarría

\_\_\_\_\_

Danay Delgado Del Rio

Tutor:

\_\_\_\_\_

Ing. Ismael Armando Nodarse Mora

\_\_\_\_\_

Ing. Ana Margarita Rojas Riverón

*“La educación empieza con la vida, y no acaba sino con la muerte”*

*José Martí*

### Agradecimientos

*A nuestros padres y abuelos por confiar y brindarnos todo su amor y apoyo incondicional, por estar junto a nosotros en todos los momentos difíciles y de tensión, por guiarnos en situaciones e indicarnos los pasos a seguir en cada momento, a ellos muchas gracias por existir.*

*A nuestra familia en general tíos, primos y otros familiares por ayudarnos a construir este sueño en realidad.*

*A nuestros tutores por su paciencia y esfuerzo incondicional, y gracias por confiar en nosotros.*

*A todos nuestros amigos y compañeros de grupo por compartir con nosotros momentos de alegría y de tristeza, nunca los olvidaremos.*

*A todos en general que se preocuparon y ayudaron de una forma u otra a la realización de este trabajo.*

### Dedicatoria

*A nuestra familia y en especial a nuestros padres que siempre han sido nuestra inspiración y apoyo para hacer este sueño una realidad, brindándonos su cariño y confianza incondicional.*

## Resumen

Es incalculable la velocidad a la que avanza la investigación y el desarrollo en la rama de la computación y la informática, tanto en niveles como gobierno, empresas, escuelas y en la misma población en general. Este avance ha significado un salto fundamental en la productividad de la sociedad cubana, principalmente en el desarrollo de software, es por ello que el presente trabajo está enmarcado en desarrollar una nueva versión de los juegos del primer ciclo de la Colección Multisaber para la Enseñanza Primaria. Con este fin se realiza el levantamiento de requerimientos, el análisis, diseño e implementación de la solución propuesta siguiendo la metodología de RUP, modelando con UML y apoyándose en Visual Paradigm. Para su desarrollo se hace un análisis de la versión actual de los juegos del primer ciclo de la Colección Multisaber así como el estudio de otros software educativos, permitiendo incorporar nuevas funcionalidades. La importancia del desarrollo de esta aplicación radica en la obtención de una nueva versión de los juegos del primer ciclo de la Colección Multisaber completamente orientada a objetos, garantizando la estabilidad del software y facilitando la satisfacción de todos aquellos que hagan uso de la aplicación.

# TABLA DE CONTENIDOS

Tabla de contenidos

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1. Fundamentación Teórica</b> .....	<b>6</b>
1.1 Introducción .....	6
1.2 Las Tecnologías de la Información y la Comunicación en la educación. ....	6
1.2.1 Los juegos educativos como materiales didácticos digitales en la Enseñanza Primaria. ....	8
1.3 Audiencia .....	9
1.4 Análisis de otras soluciones existentes.....	9
1.5 Herramientas y tecnologías utilizadas .....	11
1.5.1 Metodologías de desarrollo de software.....	11
1.5.2 Herramientas CASE .....	15
1.5.3 Lenguajes de programación .....	16
1.5.4 Servidor Web.....	19
1.5.5 Gestor de Base de Datos .....	20
1.5.6 Entorno de Desarrollo Integrado .....	21
1.6 Conclusiones .....	22
<b>CAPÍTULO 2. Descripción de la solución propuesta</b> .....	<b>23</b>
2.1 Introducción .....	23
2.2 Modelo de Dominio .....	23
2.2.1 Identificación de los conceptos del dominio .....	23
2.3 Solución propuesta .....	25
2.4 Requerimientos del Sistema .....	26
2.4.1 Definición de los requisitos funcionales.....	26
2.4.2 Definición de los requisitos no funcionales.....	28
2.5 Modelo de Casos de Uso del Sistema .....	30
2.5.1 Determinación y justificación de los actores del sistema.....	30
2.5.2 Diagrama de Casos de Uso del Sistema.....	31
2.5.3 Descripción de los Casos de Uso del Sistema.....	31
2.6 Conclusiones .....	43
<b>CAPÍTULO 3. Análisis y Diseño del Sistema</b> .....	<b>¡Error! Marcador no definido.</b>
3.1 Introducción .....	44
3.2 Modelo del Análisis .....	44
3.2.1 Diagrama de Clases del Análisis.....	44

# TABLA DE CONTENIDOS

---

3.3 Modelo del Diseño .....	50
3.3.1 Diagrama de Clases del Diseño .....	50
3.4 Diagrama de Clases Persistentes.....	57
3.5 Estándar de Codificación .....	57
3.6 Conclusiones .....	59
<b>CAPÍTULO 4. Implementación del Sistema .....</b>	<b>60</b>
4.1 Introducción .....	60
4.2 Modelo de Implementación .....	60
4.2.1 Diagramas de Componentes.....	60
4.2.2 Diagrama de Despliegue .....	65
4.3 Conclusiones .....	66
<b>Conclusiones Generales.....</b>	<b>67</b>
<b>Recomendaciones.....</b>	<b>68</b>
<b>Referencias Bibliográficas .....</b>	<b>69</b>
<b>Bibliografía.....</b>	<b>72</b>
<b>Glosario de Términos.....</b>	<b>75</b>
<b>Anexos .....</b>	<b>76</b>



## INTRODUCCIÓN

El desarrollo acelerado de las Tecnologías de la Información y la Comunicación (TIC), ha provocado un impacto significativo en la sociedad actual. Su carácter transformador y creador, tienen día a día una mayor presencia en todos los aspectos de la vida laboral y personal, ofreciendo un nuevo espacio de innovación en ámbitos como la industria, los servicios, la salud, la administración, el comercio y la educación.

Cuba también se integra a esta revolución tecnológica y como parte del proceso de transformaciones educacionales y sociales, se crean un conjunto de medidas encaminadas a desarrollar la industria de software e informatizar el país. Esta idea trajo consigo que en el año 2002 se creara la Universidad de las Ciencias Informáticas (UCI) como respuesta inmediata a esta serie de medidas.

Actualmente la UCI trabaja en numerosos proyectos encaminados a la producción de software. Estos se encuentran distribuidos por cada una de las facultades existentes, de acuerdo con la especialidad a la que pertenezcan (Software Educativo, Gestión); representando una estrategia trazada por la universidad con el objetivo de organizar los proyectos productivos y centralizar la producción.

La facultad 8 es conocida en la UCI por su experiencia en el desarrollo de software educativo. Uno de los proyectos más importantes asignado a esta facultad es Multisaber. Este se encuentra inmerso en el desarrollo de una colección de software educativo integrada por varios productos, cada uno compuesto por seis módulos fundamentales; el módulo Temas, módulo Ejercicios, módulo Juegos, módulo Mediateca, módulo Registros y módulo Maestro. Estos tienen el propósito de facilitar y complementar el proceso de enseñanza-aprendizaje en los niños de la enseñanza primaria.

El módulo Juegos lo componen 19 juegos, diseñados para niños de edades comprendidas entre primero y sexto grado. Estos están clasificados en tres grupos; los de primer ciclo para niños de primero hasta cuarto grado, los del segundo ciclo para niños de quinto y sexto grado y los de informática para niños de todos los grados escolares, desde primero a sexto.

Los juegos del primer ciclo de la Colección Multisaber no brindan la posibilidad de gestionar los contenidos educativos que presentan, lo que impide que los profesores desempeñen su rol como asesor y guía en el ambiente de aprendizaje de los estudiantes. Además, tienen la desventaja de presentar comportamientos comunes para alumnos con diferentes características, es decir; el nivel de complejidad que exigen es el mismo para un alumno aventajado que para uno menos aventajado, lo que provoca que la enseñanza no se adapte a las características individuales del alumno, viéndose afectado en múltiples ocasiones el proceso de enseñanza-aprendizaje.

Otros de los principales problemas que presentan actualmente los juegos, es que los mismos fueron implementados sin antes realizar un análisis y un diseño; lo que provocó que se violaran algunos preceptos definidos en la metodología de desarrollo de software empleada en el proyecto. Todo esto trajo consigo el uso incorrecto de buenas prácticas para el desarrollo de un producto con calidad como son: el mal uso del principio de Programación Orientada a Objetos, la no utilización de estándares de codificación y la no reutilización de código. Estos errores provocaron que muchas de sus funcionalidades fueran implementadas de forma diferente, poniendo en riesgo la estabilidad del software.

Ante tal situación el **Problema científico** radica en: ¿Cómo facilitar la gestión de contenidos y configuración de los juegos del primer ciclo de la Colección Multisaber?

El **Objetivo general** de esta investigación está centrado en: desarrollar una nueva versión de los juegos del primer ciclo de la Colección Multisaber que soporte la gestión de los contenidos educativos así como su configuración.

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

1. Analizar y diseñar la nueva versión de los juegos del primer ciclo de la Colección Multisaber de forma dinámica, flexible y escalable.
2. Incorporar un panel de configuración que permita la gestión de los contenidos educativos que presentan los juegos, así como su configuración.
3. Implementar la nueva versión de los juegos del primer ciclo de la Colección Multisaber.

Este problema se enmarca en el **Objeto de estudio**: proceso de producción de software educativo, teniendo como **Campo de acción** el proceso de desarrollo de juegos educativos de la Colección Multisaber.

La **Idea a defender** del presente trabajo es: si se desarrolla una nueva versión de los juegos del primer ciclo de la Colección Multisaber que le facilite a los profesores gestionar los contenidos educativos y configurar los juegos según las características individuales de los alumnos, permitirá que el proceso de enseñanza-aprendizaje sea más dirigido, organizado y flexible.

Los **métodos de investigación científica** que se utilizaron son:

### **Métodos Teóricos**

- Análisis histórico lógico: Este método tiene gran importancia porque mediante su estudio se logró hacer un análisis profundo sobre el surgimiento y desarrollo del software educativo en el mundo y en Cuba. En este contexto se indagó principalmente en el desarrollo de juegos educativos, analizando las tendencias actuales así como sus potencialidades de enseñanza-aprendizaje en la educación primaria.
- Analítico-sintético: La utilidad del método se centra en el estudio de una serie de documentos relacionados al proceso de desarrollo de software educativo, permitiendo realizar un análisis de la metodología utilizada en la versión actual de los juegos, así como otras metodologías utilizadas en el desarrollo de software educativo, además de las herramientas que se utilizarán en el desarrollo de la nueva versión.

### **Métodos Empíricos:**

- Entrevistas: Las entrevistas han sido importantes para la obtención de información e ideas para el desarrollo de la nueva versión de los juegos del primer ciclo de la Colección Multisaber. Entre las personas entrevistadas se encuentran los profesores y especialistas del Ministerio de Educación que trabajan en el proyecto Multisaber.

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

1. Estudiar los requisitos funcionales de la versión actual de los juegos del primer ciclo de la Colección Multisaber.
2. Definir las nuevas funcionalidades que serán incorporadas a los juegos del primer ciclo de la Colección Multisaber.
3. Identificar los juegos que admiten la gestión de los contenidos educativos.
4. Identificar los juegos que permiten la configuración de sus funcionalidades.
5. Estudiar la estructura interna de la versión actual de los juegos del primer ciclo de la Colección Multisaber.
6. Estudiar y analizar las principales características que propone la metodología RUP.
7. Realizar un estudio sobre las diferentes formas de programar orientado a objetos en Java Script.
8. Realizar un estudio de las herramientas que se utilizarán para el desarrollo de la nueva versión de los juegos del primer ciclo de la Colección Multisaber.
9. Diseñar el panel de configuración de los juegos.
10. Realizar el análisis y diseño a la nueva versión de los juegos del primer ciclo de la Colección Multisaber.
11. Realizar la implementación a la nueva versión de los juegos del primer ciclo de la Colección Multisaber.

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

**Capítulo 1: Fundamentación Teórica.** Este capítulo trata acerca de algunos elementos teóricos necesarios que sirven de soporte para la realización de todo el trabajo en general, tales como: estado del arte, análisis de la solución existente, identificación de la audiencia así como la descripción y selección de las herramientas y metodologías a utilizar.

**Capítulo 2: Descripción de la solución propuesta.** En dicho capítulo se identifican y describen los conceptos asociados al dominio del problema así como su modelo correspondiente (Modelo de Dominio). Por otro lado, se definen cuáles son los requerimientos funcionales, no funcionales, actores, casos de uso del sistema y Diagrama de Casos de Uso del Sistema.

**Capítulo 3: Análisis y Diseño del sistema.** En este capítulo se presentan los Diagramas de Clases del Análisis, lo cual influye notablemente a la hora de concebir el diseño del sistema, además de los Diagramas de Clases del Diseño, que reflejan de una forma más clara cómo va a funcionar dicho sistema y qué clases están presentes en el mismo, acompañado también de la descripción de cada una de estas.

**Capítulo 4: Implementación.** En este capítulo se describe como está implementado el sistema, a través de los Diagramas de Componentes y el Diagrama de Despliegue.



### 1.1 Introducción

En el presente capítulo se describen los elementos fundamentales que constituyen la base teórica de la investigación. En él se abordan aspectos relacionados con la aplicación de las TICs en el ámbito educativo así como el impacto de los juegos educativos en la enseñanza primaria. También se hace un análisis de la solución existente así como una descripción de las herramientas y metodologías usadas para la creación de la nueva versión de los juegos del primer ciclo de la Colección Multisaber.

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

La incorporación de las TICs en el ámbito educativo, se convierte día a día en una estrategia creativa y flexible más cercana a la diversidad y a las necesidades educativas. Las nuevas herramientas tecnológicas, junto a los nuevos materiales educativos, constituyen hoy en día recursos indispensables tanto para profesores como para alumnos.

Los profesores representan un papel fundamental en el proceso educativo actual. Estos desempeñan funciones como orientador, asesor, facilitador de recursos y herramientas de aprendizaje capaces de:

- Guiar a los alumnos en el uso de las bases de información y conocimiento.
- Potenciar la actividad de los alumnos en el aprendizaje autodirigido.
- Asesorar y gestionar el ambiente de aprendizaje de los alumnos.
- Brindar los recursos necesarios para lograr que los estudiantes incrementen y desarrollen su creatividad e imaginación.

Para los alumnos, la aplicación de las TICs en la educación ha traído consigo múltiples ventajas, entre las que se encuentran:

- Aumento del interés por la materia estudiada.
- Mejora la capacidad para resolver problemas.
- Aprenden a trabajar en grupo y a comunicar sus ideas.

## CAPÍTULO 1. Fundamentación Teórica

---

- Adquieren mayor confianza en sí mismos.
- Incrementan su creatividad e imaginación.
- Favorecen el espíritu de búsqueda.
- Estimulan el desarrollo de ciertas habilidades intelectuales tales como el razonamiento, la resolución de problemas y la creatividad.
- Mejoran los resultados académicos. (1)

Actualmente las TICs pueden emplearse en el sistema educativo moderno de tres formas distintas: como objeto de aprendizaje, como medio de enseñanza o como apoyo al aprendizaje.

El uso de las nuevas tecnologías como objeto de aprendizaje, permiten que los alumnos se familiaricen con el ordenador y adquieran las destrezas necesarias para hacer del mismo un instrumento útil a lo largo de su vida. Sin embargo, cuando son utilizadas como medio de enseñanza se convierten en una herramienta al servicio de la formación de los estudiantes, ya sea de forma presencial o no presencial. Ejemplo de esto pueden ser: una multimedia educativa, cursos en línea a través de Internet, videoconferencias, programas de simulación, de ejercicios, entre otros.

Las TICs también reflejan un impacto significativo como apoyo al aprendizaje. Estas se encuentran pedagógicamente integradas al proceso de enseñanza-aprendizaje, responden a diferentes necesidades de formación y son empleadas de forma cotidiana en las escuelas, en los hogares y en los centros de trabajos; enmarcándose en una perspectiva de formación continua de los estudiantes, ya sea como complemento o enriquecimiento de los contenidos presentados. (2)

El proceso de enseñanza-aprendizaje mediante las TICs no solo trata de ofrecer más oportunidades de formación, sino también de generar una conciencia y motivación para aprender. Requiere que el estudiante tome parte activa en este proceso, que sepa aprender en multiplicidad de entornos, que sepa personalizar el aprendizaje y que sepa utilizar las nuevas herramientas tecnológicas de acuerdo con sus necesidades. Por esta razón los sistemas educativos a nivel mundial se enfrentan al desafío de utilizarlas con el fin de brindarle a sus alumnos las herramientas y los conocimientos necesarios que se requieren en la sociedad del siglo XXI.

## 1.2.1 Los juegos educativos como materiales didácticos digitales en la enseñanza primaria

El desarrollo de las TICs, ha hecho posible la aparición de nuevos recursos y herramientas en el ámbito educativo. Los juegos educativos como materiales didácticos digitales, constituyen una parte fundamental de la revolución educacional que hoy se lleva a cabo en la enseñanza primaria.

Los materiales didácticos digitales son recursos que se crean específicamente para ser usados en la computadora, con la finalidad de complementar el proceso de enseñanza-aprendizaje mediante la incorporación de elementos multimedia (textos, imágenes, sonidos, gráficos, vídeo y voz). Estas características permiten presentar la información de forma variada e interactiva, haciendo que el aprendizaje sea más flexible, abierto y motivador. (3)

En la actualidad muchos de los juegos educativos que se utilizan en las escuelas, son materiales didácticos digitales diseñados para formar y educar a los alumnos en tiempo reducido; convirtiéndose en actividades fundamentales de enseñanza-aprendizaje en la educación primaria.

El impacto que ha provocado la incorporación de los juegos educativos en los alumnos de la enseñanza primaria, ha traído consigo múltiples ventajas que permiten:

- Potenciar el desarrollo de la actividad mental y física.
- Fomentar normas de relaciones sociales y morales.
- Descubrir valores y aptitudes.
- Entender diversos conceptos mediante representaciones simbólicas.
- Fomentar la interacción entre los jugadores, que expresan y confrontan sus puntos de vista y opiniones.
- Utilizar y aplicar conocimientos y habilidades en diversos contextos y situaciones.
- Ganar en confianza y vencer dificultades.

Sin embargo, los juegos educativos pueden considerarse una actividad de enseñanza-aprendizaje efectiva si:

- Requiere la participación activa de los jugadores.



- Exige el uso de diversas habilidades, destrezas y conocimientos.
- Su objetivo radica en que los niños se diviertan y aprendan simultáneamente.
- Estimula el razonamiento de los niños.
- Fomenta la comunicación.
- Es susceptible de progresar en él, de jugar cada vez mejor. (4)

La aplicación provechosa de los juegos educativos en la enseñanza primaria no solo posibilita el desarrollo biológico, psicológico, social y espiritual del alumno como eje de la acción educativa; también promueven la diversión y el entretenimiento mientras adquieren cultura y conocimiento.

### 1.3 Audiencia

Los juegos del primer ciclo de la Colección Multisaber están dirigidos directamente a los niños de la enseñanza primaria. Su uso no exige tener ciertas habilidades con la computadora, pero si deben tener al menos un mínimo conocimiento de los contenidos que se presentan.

### 1.4 Análisis de otras soluciones existentes

#### Ardora

Ardora es una aplicación informática para docentes que permite crear actividades educativas, en formato html, para sus alumnos y alumnas de un modo muy sencillo. Este permite crear más de 45 tipos distintos de actividades (crucigramas, sopas de letras, completar, paneles gráficos, entre otros), así como más de 10 tipos distintos de páginas multimedia (galerías de imágenes, reproductores, entre otros). (5)

Características:

- Capacidad de brindarles la posibilidad a los profesores de configurar las actividades como ellos estimen conveniente.
- La forma de configurar las actividades se realizan mediante formularios muy sencillos.
- Permite a los profesores centrar su esfuerzo en los elementos de la actividad no en su tratamiento informático.
- Fácil de usar.
- Rapidez a la hora de crear actividades docentes educativas.

- Permite guardar, publicar y hacer uso previo de actividades educativas.
- Permite guardar la actividad en formato html.

Aun cuando Ardora no cuenta con un diseño atractivo, hay que destacar que sus características en función de la creación de actividades docentes educativas la convierten en una potente herramienta.

## El Navegante

El Navegante constituye una colección de software educativo dirigido a los alumnos de la enseñanza secundaria de Cuba. Esta colección se encuentra compuesta por un conjunto de software educativos, los cuales representan a las asignaturas que se imparten en esta enseñanza. Su objetivo radica en complementar el proceso de enseñanza-aprendizaje en los alumnos, así como servir de apoyo para los profesores. Estos software educativos que conforman esta colección, están basados en la tecnología multimedia, lo que permite enriquecer los contenidos presentados mediante imágenes, videos, sonidos, entre otros.

Los software educativos que integran la colección son:

- **“Elementos matemáticos”** que representa la asignatura de Matemática.
- **“El fabuloso mundo de las palabras”** que representa la asignatura de Literatura y Español.
- **“La naturaleza y el hombre”** que representa la asignatura de Geografía, Biología, Física y Química.
- **“Rainbow”** que representa la asignatura de Inglés.
- **“Informática Básica”** que representa la asignatura de Informática Básica.
- **“Encuentro con el pasado”** que representa la asignatura de Historia Antigua y Medieval.
- **“EduArte”** que representa la asignatura de Educación Artística.
- **“GeoClío”** que representa la asignatura de Historia Moderna y Contemporánea.
- **“Por los senderos de mi patria”** que representa la asignatura de Historia de Cuba y Arte cubano.
- **“Aprende construyendo”** que representa la asignatura de Educación Laboral y Dibujo Básico.

Las características de este software y su capacidad de abarcar todas las asignaturas que se imparten en la enseñanza secundaria, convierten a esta colección en una potente herramienta que ayuda tanto a los estudiantes como a los profesores a guiarse y aprender de una forma variada y dinámica.

## **Multisaber**

Multisaber es una colección de software educativo compuesta por varios productos donde cada uno se encuentra integrado por diversos módulos. El módulo Juegos está compuesto por 19 juegos, los cuales fueron diseñados específicamente para apoyar la enseñanza y el aprendizaje en los alumnos de la enseñanza primaria de Cuba. Sin embargo, hay que señalar que aun cuando resultan atractivos para aquellos a quienes están dirigidos, los mismos presentan varias deficiencias que le restan dinamismo y flexibilidad.

Actualmente los juegos no les brindan la posibilidad a los profesores de ejercer una labor más activa sobre el proceso de enseñanza-aprendizaje de los alumnos. Su diseño está enfocado específicamente en los estudiantes, sin tener en cuenta las ventajas que puede traer la participación del profesor en todo este ambiente de aprendizaje.

Otras de las dificultades, es que los juegos presentan comportamientos comunes para todos los alumnos, es decir; no se tiene implementado una forma que permita flexibilizar los juegos haciendo que estos aumenten o disminuyan su complejidad.

El estudio realizado a estas soluciones, ha permitido analizar con un mayor nivel de complejidad sus características, ventajas y desventajas; con el objetivo de obtener un conjunto de ideas para la realización de la nueva versión de los juegos del primer ciclo de la Colección Multisaber que se está desarrollando en la UCI.

## **1.5 Herramientas y tecnologías utilizadas**

A continuación se describen todas las herramientas y tecnologías que serán utilizadas para darle solución al problema en general, teniendo en cuenta las necesidades existentes y el entorno donde se aplicará el sistema que se va a construir.

### **1.5.1 Metodologías de desarrollo de software**

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar un software con calidad.

Define las etapas para el desarrollo de un software, las tareas que se deben llevar a cabo en cada etapa, las restricciones que deben aplicarse, las técnicas y herramientas que deben emplearse, y la forma de controlar y gestionar un proyecto. (6)

Algunas de las metodologías de desarrollo de software más utilizadas en la actualidad son:

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

## **Programación Extrema**

XP representa una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos de corto plazo y equipo de desarrollo pequeño. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Esta define cuatro fases para llevar a cabo el desarrollo del proyecto, Planificación, Diseño, Desarrollo y Prueba. (7)

## **Proceso Unificado de Rational**

El Proceso Unificado de Rational (RUP) constituye una de las metodologías más utilizada para el desarrollo de sistemas de software orientados a objetos.

La metodología RUP define como sus principales elementos:

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

RUP divide el proceso de desarrollo de un software en cuatro fases, dentro de las cuales se realizan varias iteraciones según el tamaño del proyecto y la prioridad que tengan las actividades. Esta metodología divide las actividades en grupos lógicos definiendo 9 flujos de trabajo, los 6 primeros conocidos como los de ingeniería y los 3 últimos de apoyo (Ver Anexo 1).

## Fases de desarrollo:

- **Conceptualización (Concepción o Inicio):** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene una o varias versiones del producto que han pasado las pruebas.
- **Transición:** Se obtiene un producto listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

## Flujos de trabajo:

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba:** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce versiones del producto y realiza actividades (empaquete, instalación, asistencia a usuarios) para entregar el software a los usuarios finales.

- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización, actualización concurrente de elementos y control de versiones.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

El ciclo de vida de RUP se caracteriza por:

1. **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
2. **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
3. **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. (8)

RUP es una metodología que tiene bien definido y organizado el proceso de desarrollo de software, lo que permite que los involucrados estén claro de los pasos a seguir para obtener un producto con calidad. Además, provee un entorno de proceso de desarrollo configurable, basado en estándares; que garantiza que esta metodología sea adaptable a las necesidades de la organización y del proyecto. Estas ventajas, junto a sus características, representan algunas de las razones que justifican la selección de esta metodología para guiar el proceso de desarrollo de la nueva versión de los juegos del primer ciclo de la Colección Multisaber.

## 1.5.2 Herramientas CASE

La misión de cualquier herramienta CASE que utiliza UML como notación para elaborar los modelos, es comunicar, de la manera más eficiente posible a todos los agentes del proyecto, todas aquellas decisiones que se toman con respecto a la arquitectura del sistema en discusión y que son determinantes para cumplir con los objetivos del proyecto.

Establecer una arquitectura racional y eficiente, acorde a las necesidades de los actores, es la razón fundamental para invertir tiempo y esfuerzos en una herramienta CASE que utiliza la notación UML. Escoger los modelos apropiados en cada fase del proyecto, usar convenciones para nombrar y situar correctamente los artefactos en cada modelo, saber aplicar un vocabulario controlado que permita conocer la trazabilidad de los artefactos a través de las perspectivas de los distintos modelos; está dirigido a extraer el máximo rendimiento de la herramienta para comunicar los puntos clave del diseño de una arquitectura, desde la especificación, hasta el entorno de ejecución.

A través de la notación UML y con el soporte de una herramienta CASE se puede comunicar y compartir el conocimiento de una arquitectura (conceptual, de diseño, y de implementación), gracias a la combinación simultánea de distintas perspectivas como:

1. Definir conceptos a través de sus atributos distintivos y trazar sus límites.
2. Formalizar unas reglas de relación y actuación.
3. Hacer visible su granularidad y entrelazamiento.
4. Mantener la trazabilidad entre la concepción de un problema y su solución.
5. Tomar decisiones y evaluar su impacto de manera rápida.
6. Organizar la experiencia de los usuarios en patrones de conocimiento.
7. Comprobar la coherencia, completitud y usabilidad de los entregables.
8. Usar un vocabulario controlado dentro de un espacio de colaboración. (9)

Existen varias herramientas CASE para el modelado visual mediante UML de sistemas software, entre estas se encuentran Enterprise Architect, Rational Rose, Visual Paradigm.

**Enterprise Architect** es una herramienta UML que abarca el ciclo de vida completo del desarrollo de software que provee una estructura competitiva en el modelado de negocio, diseño de software, ingeniería de sistemas, arquitectura de empresas, gestión de requisitos y prueba.

# CAPÍTULO 1. Fundamentación Teórica

---

Permite integrar al equipo de desarrollo proporcionando una visión de trabajo compartido. Enterprise Architect es una herramienta multiusuario, basada en Windows, diseñada para ayudar a construir software robusto, fácil de mantener, fácil de usar, rápido y flexible. Ofrece salida de documentación flexible y de alta calidad. Provee una trazabilidad completa desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue. Las bases de Enterprise Architect están construidas sobre la especificación de UML 2.0. (10)

**Rational Rose** proporciona un desarrollo iterativo que permite cubrir todo el ciclo de vida de un proyecto, en el que se puede especificar, analizar, diseñar un sistema antes de codificarlo; ayudando a establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Además, permite generar código a partir de los modelos y realizar ingeniería inversa (crear modelo a partir código). (11)

Sin embargo, cuando estas características la convierten en una herramienta muy utilizada actualmente, su particularidad de ser propietaria (no gratuito, restricciones de licencia, código fuente) la hacen ser rechazada por muchas empresas y desarrolladores de software.

**Visual Paradigm** es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas que sigue el estándar de UML (Diagramas de Clase, Casos de Uso, Comunicación, Secuencia, Estado, Actividad, Componentes). También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial, es fácil de instalar y actualizar, compatible entre ediciones y admite ingeniería de ida y vuelta. Además, permite generación de código, documentación y de bases de datos. (12)

Estas características convierten a esta herramienta en unas de las más potentes y utilizadas a nivel mundial, debido a la cantidad de funcionalidades y ventajas sobre otras herramientas CASE. Estas razones hacen de Visual Paradigm la herramienta seleccionada para llevar a cabo el modelado visual de la nueva versión de los juegos del primer ciclo de la Colección Multisaber.

## 1.5.3 Lenguajes de programación

Un lenguaje de programación es un lenguaje artificial que permite establecer una comunicación con una máquina, especialmente una computadora.



# CAPÍTULO 1. Fundamentación Teórica

---

Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

Son ejemplos de lenguajes de programación: ASP, ActionScript, Python, Pascal, JAVA, PHP, JavaScript, entre otros. (13)

**PHP** (Hypertext Preprocessor) es un lenguaje de código abierto interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. (14)

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Características:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF y analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas en la web mediante una fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede procesar la información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Ventajas:

- Es un lenguaje multiplataforma.
- Principalmente orientado a la web.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
- Posee una amplia documentación en la que se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones.
- Rapidez. PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz y se ejecuta rápidamente utilizando poca memoria. (15)

**JavaScript** es un lenguaje interpretado en el cliente por el navegador al momento de cargarse la página, es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento HTML. (16)

JavaScript proporciona los medios para:

- Controlar las ventanas del navegador y el contenido que muestran.
- Evitar depender del servidor web para cálculos sencillos.
- Capturar los eventos generados por el usuario y responder a ellos sin salir a Internet.
- Comprobar los datos que el usuario introduce en un formulario antes de enviarlos.
- Comunicarse con el usuario mediante diversos métodos. (17)

Características:

- Es simple, no hace falta tener muchos conocimientos de programación para poder hacer un programa en JavaScript.
- Maneja objetos dentro de una página web y sobre ese objeto se pueden definir diferentes eventos, los cuales facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades y modificaciones de archivos.

- Es dinámico, responde a eventos en tiempo real.
- Soporta cuatro tipos de datos, aunque no es necesario declarar el tipo de las variables, argumentos de funciones, ni valores de retorno de las funciones. (16)

El análisis exhaustivo sobre los posibles lenguajes de programación a utilizar en el desarrollo de la versión 2.0 de los juegos del primer ciclo de la Colección Multisaber, permitió la selección de los lenguajes de programación anteriormente justificados. Sin embargo, no se hace un estudio profundo sobre otros lenguajes, debido a que el proyecto Multisaber decidió escoger PHP como lenguaje del lado del servidor y JavaScript del lado del cliente.

## 1.5.4 Servidor web

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP (HyperText Transfer Protocol) o el protocolo HTTPS (versión cifrada y autenticada).

Un servidor web puede definirse como un programa que escucha las peticiones de los usuarios o navegantes y las atiende o satisface. Por medio de la especificación de la búsqueda el servidor web buscará una página específica o ejecutará un programa, pero, necesariamente, enviará algún resultado sobre la búsqueda recibida. Entre los servidores web más utilizados se encuentran el Internet Information Server (IIS), el Xitami y el Apache. (18)

**Apache** es uno de los servidores web más reconocidos y utilizados a nivel mundial. Este constituye una muestra, al igual que el sistema operativo Linux, de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difícil de igualar. Su gran popularidad se debe a múltiples características que brindan una serie de ventajas como:

- Corre en una multitud de sistemas operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código abierto.
- Apache es un servidor altamente configurable de diseño modular. Actualmente existen muchos módulos para Apache que son adaptables y permiten su instalación facilitando el trabajo con este servidor web.

- Apache trabaja con gran cantidad de lenguajes como Perl, PHP, Java, JavaScript, entre otros; teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta capacidad de configuración en la creación y gestión de log. Apache permite la creación de ficheros log que facilita tener un mayor control sobre lo que sucede en el servidor. (19)

Estas razones hacen de Apache un servidor web por excelencia, que por su capacidad de configuración, robustez y estabilidad haya sido seleccionado como el servidor web soporte de la aplicación.

## 1.5.5 Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de Datos, asegurando su integridad, confidencialidad y seguridad. Existen diferentes tipos de SGBD, pero los más conocidos y utilizados actualmente son PostgreSQL, Oracle, MySQL, entre otros. (20)

**PostgreSQL** está ampliamente considerado como uno de los SGBD de código abierto más avanzado del mundo. Este proporciona un gran número de capacidades como: consultas complejas, manejo de vistas, integridad referencial, bloqueo de tabla y filas, entre otras.

Originalmente, al igual que otros proyectos de software libre, se desarrolló para operar dentro del sistema operativo Linux, pero las mismas exigencias de los usuarios han contribuido a que en la actualidad se disponga de versiones para Windows. (21)

**Oracle** es uno de los SGBD más utilizados en la gestión de grandes bases de datos debido a su alto rendimiento en transacciones y disponibilidad. Además, presenta otras ventajas como portabilidad, compatibilidad y gestión de seguridad, pero su uso exige ser pagado a un elevado precio, lo que provoca que sea rechazado por muchas empresas que se dedican al desarrollo de software de gestión. (22)

**MySQL** es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU, creada por la empresa sueca MySQL AB. Las principales características de este SGBD son las siguientes:

- Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo.
- Rápido y fácil de usar.
- Fácil de instalar y configurar.
- Cuenta con una infinidad de librerías y herramientas que facilita su uso.
- Capacidad de gestionar y almacenar grandes cantidades de datos.
- Gran portabilidad entre sistemas.
- Gestión de usuarios y contraseñas, manteniendo un buen nivel de seguridad en los datos. (23)

Este SGBD es uno de los más usados en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Estas ventajas hacen de MySQL el SGBD seleccionado para almacenar y gestionar los datos que necesitan ser almacenados.

## 1.5.6 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE) es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes. (24)

Algunos ejemplos de IDE son el Eclipse, Aptana Studio, Zend Studio, NetBeans, entre otros.

**Eclipse** es un (IDE), para todo tipo de aplicaciones libres, inicialmente desarrollado por IBM, y actualmente gestionado por Fundación Eclipse. Es una herramienta desarrollada principalmente para el desarrollo de aplicaciones Java, aunque se le puede integrar nuevos módulos ('plugins'), para programar en otros lenguajes de programación como C, C++, PHP, Python, Ruby, entre otros. (25)

**Aptana Studio** es un IDE para la elaboración de aplicaciones web dinámicas. Este posee un asistente de código que ayuda al programador en la escritura de lenguajes como CSS y JavaScript entre ellos. En el caso del HTML, puede mostrar todos los elementos pertenecientes a este lenguaje y sus propiedades. (26)

# CAPÍTULO 1. Fundamentación Teórica

---

**Zend Studio** es un IDE creado para el desarrollo de aplicaciones altamente fiables, proporciona una facilidad de uso inigualable, escalabilidad, fiabilidad, y la extensión que los programadores profesionales y de empresas requieren para desarrollar, distribuir, depurar y administrar aplicaciones PHP. (27)

**NetBeans** es uno de los IDE más utilizados en la actualidad. Este es un producto de código abierto, con todos los beneficios del software disponible en forma gratuita, el cual ha sido examinado por una comunidad de desarrolladores.

Este enfoque de bienes comunes ha permitido una mayor capacidad de uso, con cada nueva versión, y ha proporcionado a los desarrolladores mayor flexibilidad gracias a su integración con múltiples herramientas y lenguajes de programación (Java, C, C++, Ruby, Python, PHP, JavaScript).

NetBeans ofrece gran flexibilidad entre plataformas, el cumplimiento de UML y la capacidad de administrar la complejidad, ayudan a garantizar que las aplicaciones cumplan con los requerimientos definidos. (28)

Este IDE garantiza que los desarrolladores puedan confiar en una plataforma de desarrollo que integra todas las piezas críticas necesarias para llevar a cabo el desarrollo de aplicaciones. En este contexto, ventajas como su capacidad de integrar los lenguajes de programación PHP y JavaScript, y disponibilidad gratuita; hacen de este IDE el seleccionado para desarrollar la nueva versión de los juegos del primer ciclo de la Colección Multisaber.

## 1.6 Conclusiones

En este capítulo quedó plasmado todo lo correspondiente a la base teórica de la investigación, donde se logró realizar una profunda valoración del estado del arte mediante el análisis de algunos temas como las TICs en la educación, Los juegos educativos como materiales didácticos digitales en la enseñanza primaria, así como el estudio de otras soluciones existentes en la actualidad. Todo esto permitió demostrar la importancia de los juegos educativos en la enseñanza primaria y la posibilidad de incorporar a los juegos de la Colección Multisaber un panel de configuración sencillo. También se realizó un estudio bien detallado sobre las herramientas y tecnologías a utilizar, donde queda definido RUP como metodología de desarrollo de software, Visual Paradigm como herramienta CASE para el modelado UML, PHP y JavaScript como lenguajes de programación, Apache como servidor web, MySQL como gestor de base de datos y NetBeans como IDE de programación.

### Descripción de la solución propuesta



#### 2.1 Introducción

El presente capítulo trata algunos de los elementos fundamentales que forman parte de la solución, según la metodología utilizada. Se identifican los conceptos asociados al dominio del problema, requisitos funcionales, no funcionales, actores y los casos de uso del sistema. Además, se realiza el Modelo de Dominio según los conceptos identificados, así como el Modelo de Casos de Uso del Sistema.

#### 2.2 Modelo de Dominio

RUP como metodología de desarrollo de software propone realizar un Modelo de Dominio (modelo conceptual) para cuando los procesos de negocio no están claramente definidos.

El Modelo de Dominio es una representación visual de los conceptos u objetos del mundo real, significativos para un problema o área de interés. Captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Representa conceptos del mundo real, no de los componentes de software. (1)

##### 2.2.1 Identificación de los conceptos del dominio

- **Colección Multisaber:** Es una colección de software educativo destinada a apoyar el proceso de enseñanza-aprendizaje en los niños de las escuelas primarias de Venezuela.
- **Escuela primaria:** Institución educativa centrada en la formación de menores, que tiene como objetivo fundamental organizar la enseñanza y el aprendizaje de contenidos básicos.
- **Estudiante de primaria:** Persona que cursa estudios en las escuelas primarias.
- **Maestro de primaria:** Persona responsable de educar, guiar y supervisar a los estudiantes en el proceso de enseñanza-aprendizaje.
- **Producto Multimedia:** Producto que combina diferentes tipos de medias.
- **Módulo:** Se le denomina al conjunto de elementos fundamentales que componen el producto.
- **Juegos:** Es aquel módulo encargado de que los estudiantes pueden divertirse mientras emplean sus habilidades y conocimientos.

## CAPÍTULO 2. Descripción de la solución propuesta

- **Mediateca:** Es aquel módulo encargado de representar la información gráfica (galería de imágenes, videos).
- **Ejercicios:** Es aquel módulo encargado de brindarle a los estudiantes actividades como baterías de preguntas o cuestionarios que emiten un criterio evaluativo a cada una de las respuestas.
- **Resultados:** Módulo encargado de almacenar y visualizar el comportamiento del estudiante durante su interacción con la colección.
- **Temas:** Módulo en el que se presentan los contenidos de las asignaturas o temas al cual corresponde el software.
- **Maestro:** Es el módulo que le permite al profesor configurar algunas opciones del producto en cuestión.
- **Media:** Se utiliza para referirse tanto a texto, imágenes, sonidos o videos.
- **Texto:** Es una composición de signos codificado en un sistema de escritura (como un alfabeto) que forma una unidad de sentido.
- **Video:** Su objetivo radica en brindarle al usuario una información audiovisual, para que perciba imágenes y sonidos.
- **Sonido:** Sensación producida en el órgano del oído por el movimiento vibratorio de los cuerpos, transmitido por un medio elástico, como el aire.
- **Imagen:** Representación gráfica de una expresión.

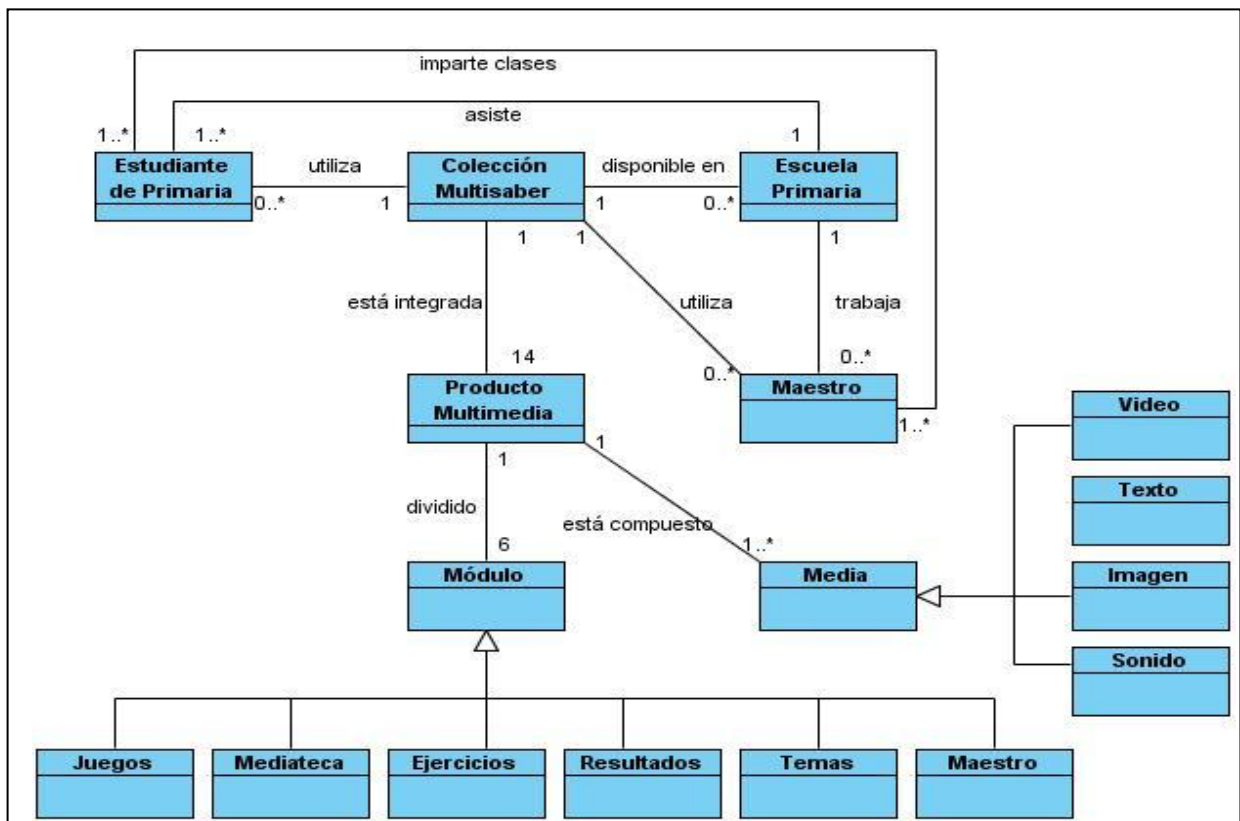


Figura 1. Modelo de Dominio



## CAPÍTULO 2. Descripción de la solución propuesta

---

### 2.3 Solución propuesta

Para darle solución a la problemática planteada se determinó realizar una nueva versión de los juegos del primer ciclo de la Colección Multisaber, con el objetivo de hacerlos más dinámicos, flexibles y escalables.

Para lograr este objetivo, primeramente se realizará el análisis y diseño de los mismos, para luego pasar a la fase de Implementación. En este contexto se decidió reprogramar estos juegos haciendo un buen uso del principio de la Programación Orientada a Objetos y utilizando un estándar de codificación apropiado con el propósito de que su código sea más organizado y legible; listo para próximas mejoras.

Las características anteriormente mencionadas garantizarán que los juegos del primer ciclo de la Colección Multisaber presenten una gran estabilidad en sus funcionalidades; además de que con la incorporación de nuevas funcionalidades, el trabajo del profesor en el ambiente de aprendizaje de los estudiantes será beneficiado.

Algunas de las ventajas que caracterizarán a la nueva versión de los juegos del primer ciclo de la Colección Multisaber son:

- **Dinamismo:** La información podrá ser presentada de manera diferente, ofreciendo caminos de interacción en función de los intereses del profesor.
- **Flexibilidad:** Capacidad de aumentar o disminuir su nivel de complejidad, permitiendo adaptarse a las diferentes características de los usuarios.
- **Escalabilidad:** Es la capacidad que tienen estos juegos de ir incrementando su complejidad en función del proceso de enseñanza-aprendizaje a medida que los alumnos vayan venciendo los objetivos que se deseen.
- **Configurables:** Los profesores pueden configurar las principales funcionalidades de los juegos.
- **Estabilidad:** Las características comunes existentes en los juegos serán implementadas y mostradas de forma semejante, permitiendo que los usuarios no puedan identificar alteraciones en las funcionalidades de los juegos.
- **Organización y legibilidad en el código:** El código de los juegos estará bien organizado y estructurado; facilitando que en un futuro cuando se deseen realizar otras mejoras, los desarrolladores sean capaces de entenderlo fácilmente.

### 2.4 Requerimientos del sistema

Para llevar a cabo el diseño y la implementación de un software, es necesario saber qué debe hacer dicho sistema y cómo debe hacerlo, además de tener bien claro cuáles son las expectativas del cliente con respecto al sistema que se va a desarrollar. Para realizar los requisitos de forma exitosa y obtener con buena calidad un producto, RUP propone como proceso de desarrollo de software el flujo de trabajo de Requerimientos. Los requisitos pueden clasificarse en: funcionales y no funcionales.

#### 2.4.1 Definición de los requisitos funcionales

Los requerimientos o requisitos funcionales, son capacidades o condiciones que el sistema debe cumplir, define lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas.

A continuación se presentan los juegos del primer ciclo de la Colección Multisaber con sus correspondientes requisitos funcionales:

#### Requisitos Generales

**RF 1** - Cargar pantalla principal de los juegos.

- **R 1.1** - Cargar imagen de identificación del juego.
- **R 1.2** - Permitir seleccionar el juego.
- **R 1.3** - Mostrar el juego.

**RF 2** - Cargar información inicial.

- **R 2.1** - Cargar todas las imágenes del juego.
- **R 2.2** - Cargar datos de los jugadores.
- **R 2.3** - Mostrar área de estadísticas.
- **R 2.4** - Mostrar área de jugador en turno.
- **R 2.5** - Mostrar mensajes de orientación.

**RF 3** - Controlar eventos.

- **R 3.1** - Controlar evento Clic.
- **R 3.2** - Controlar evento Mouse Over.
- **R 3.3** - Controlar evento Mouse Out.

**RF 4** - Controlar tiempo.

## CAPÍTULO 2. Descripción de la solución propuesta

---

- **R 4.1** - Iniciar conteo del tiempo.
- **R 4.2** - Detener tiempo.
- **R 4.3** - Calcular tiempo en segundos.
- **R 4.4** - Mostrar el tooltip.

**RF 5** - Realizar actividad.

- **R 5.1** - Mostrar preguntas.
- **R 5.2** - Seleccionar respuestas.
- **R 5.3** - Evaluar respuestas.

**RF 6** - Evaluar jugador.

- **R 6.1** - Otorgar calificación.
- **R 6.2** - Asignar efectividad.
- **R 6.3** - Guardar el tiempo transcurrido.

**RF 7** - Permitir configurar los juegos.

- **R 7.1** - Permitir ver, insertar, eliminar o modificar el contenido de los juegos.
- **R 7.2** - Permitir que los juegos se configuren dinámicamente.

**RF 8** - Mostrar resultado del juego.

- **R 8.1** - Cargar podio.
- **R 8.2** - Mostrar el resultado final de los jugadores.

### **Rescatando al Turpial**

**RF 9** - Controlar funcionalidades de Rescatando al Turpial.

- **R 9.1** - Clausurar puerta.
- **R 9.2** - Avanzar hacia la siguiente sala.
- **R 9.3** - Mostrar el localizador de la sala.
- **R 9.4** - Liberar turpial.

### **Descubre la Imagen 1**

**RF 10** - Controlar funcionalidades de Descubre la Imagen 1.

- **R 10.1** - Cargar mascota aleatoriamente.
- **R 10.2** - Mostrar mascota inicial.
- **R 10.3** - Mover mascota.
- **R 10.4** - Mostrar imagen descubierta.

## CAPÍTULO 2. Descripción de la solución propuesta

---

### Ensartando Aros

**RF 11** - Controlar funcionalidades de Ensartando Aros.

- **R 11.1** - Permitir ensartar los aros correctamente.
- **R 11.2** - Permitir separar los aros del conjunto.
- **R 11.3** - Permitir mover los aros hacia fuera.
- **R 11.4** - Permitir balancear aros.

### Rompecabezas

**RF 12** - Controlar funcionalidades del Rompecabezas.

- **R 12.1** - Seleccionar los Rompecabezas que se desean armar.
- **R 12.2** - Seleccionar la cantidad de cortes.
- **R 12.3** - Seleccionar el tipo de corte.
- **R 12.4** - Permitir rotar una pieza.
- **R 12.5** - Mover piezas.
- **R 12.6** - Mostrar jugador en turno.
- **R 12.7** - Mostrar el localizador del Rompecabezas.
- **R 12.8** - Cargar las piezas de un Rompecabezas.
- **R 12.9** - Cargar los niveles de ayuda de un Rompecabezas.
- **R 12.10** - Mostrar imagen completa.

### Escalera de Palabras

**RF 13** - Controlar funcionalidades de la Escalera de Palabras.

- **R 13.1** - Seleccionar carta.
- **R 13.2** - Virar carta.
- **R 13.3** - Mostrar palabra.
- **R 13.4** - Seleccionar respuesta.
- **R 13.5** - Mover carta.

#### 2.4.2 Definición de los requisitos no funcionales

Los requerimientos o requisitos no funcionales son propiedades o cualidades que el producto debe tener, que posibilitan que el mismo sea más atractivo, usable, confiable, entre otras. También puede ser alguna restricción que este debe tener para cumplir una determinada funcionalidad.

A continuación se presentan los requisitos no funcionales:

## CAPÍTULO 2. Descripción de la solución propuesta

---

### Requerimientos de Software:

- Los juegos deberán correr sobre el sistema operativo Linux.
- Navegador Mozilla Firefox 3.x.

### Requerimientos de Hardware:

- Procesador Pentium 233 MHz (recomendado 500 MHz o mayor).
- 64 MB de RAM (recomendado 128 MB de RAM o mayor).
- 1 GB de espacio en disco duro.
- Dispositivos de audio.
- Soporte de video que admita resolución de al menos 800x600 y 24 bits.
- Dispositivo de red de al menos 10 Mbits.
- El sistema interactuará con una impresora que permita imprimir los diferentes contenidos como respuesta a las funcionalidades del sistema.

### Restricciones en el diseño y la implementación:

- NetBeans como IDE de programación.
- El lenguaje de programación será PHP 5 y JavaScript.
- MySQL 5.0 como Sistema Gestor de Base de Datos.
- Servidor web Apache.
- Visual Paradigm como herramienta CASE para el modelado UML.

### Requerimientos de apariencia o interfaz externa:

- Los juegos tendrán una interfaz sencilla y amigable.
- La información de los juegos será presentada de forma clara y organizada, permitiendo una correcta interpretación en los usuarios.
- Los juegos se ejecutarán de manera rápida, facilitando una mejor interacción entre usuario y juego.
- El diseño de la interfaz gráfica deberá garantizar la distinción visual de los principales elementos de los juegos.

### Requerimientos de usabilidad:

- Los juegos contarán con una interfaz amigable y una fácil navegación tanto para usuarios expertos (profesores e invitados) como para los que no tienen conocimientos profundos de informática (alumnos).

## CAPÍTULO 2. Descripción de la solución propuesta

- Los contenidos que serán presentados en los juegos estarán en correspondencias con las asignaturas que se estudian en la Enseñanza Primaria, lo que permitirá que su interacción sea más asequible.

### Requerimientos de seguridad:

- **Integridad:** Solo los administradores y los profesores podrán modificar los contenidos educativos que serán presentados en los juegos.

### Requerimientos de portabilidad:

- El sistema podrá ser utilizado bajo cualquier sistema operativo.

## 2.5 Modelo de Casos de Uso del Sistema

El Modelo de Casos de Uso describe la funcionalidad propuesta del nuevo sistema. Un caso de uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema, estos se utilizan para la comunicación clara y el levantamiento de requisitos del desarrollo de un sistema, representándolos mediante un Diagrama de Casos de Uso. También se describen los actores del sistema y los casos de uso que van a representar las funcionalidades.

### 2.5.1 Determinación y justificación de los actores del sistema

Un actor del sistema define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que interactúan con el mismo intercambiando información con este.

A continuación se muestra la tabla con los actores del sistema y la justificación de los mismos.

Actor	Justificación
Usuario	Representan aquellas personas (estudiante de primaria, maestro e invitado) que de una forma u otra interactúa con los juegos de la Colección Multisaber.
Maestro	Representan todas aquellas personas encargadas de guiar, examinar y supervisar el comportamiento de los estudiantes.

## CAPÍTULO 2. Descripción de la solución propuesta

### 2.5.2 Diagrama de Casos de Uso del Sistema

Los Diagramas de Casos de Uso del sistema son un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable para un actor. Es decir, son “fragmentos” de funcionalidad que el sistema ofrece a los actores que interactúan con el mismo, brindándole algunos beneficios.

A continuación se muestra la figura correspondiente al Diagrama de Casos de Uso del Sistema.

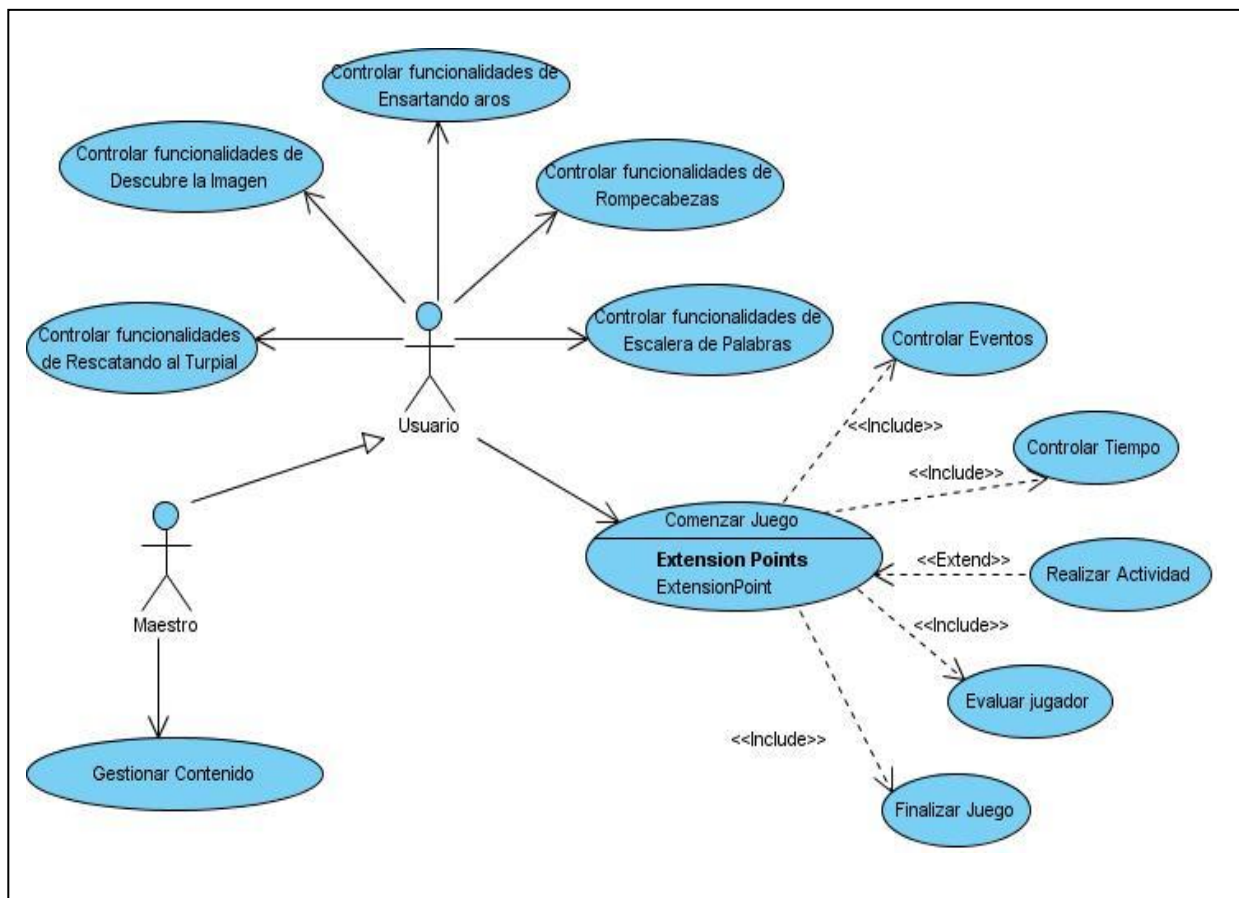


Figura 2. Diagrama de Casos de Uso del Sistema

### 2.5.3 Descripción de los Casos de Uso del Sistema

La Descripción Textual de los Casos de Uso del Sistema describe bajo la forma de acciones y reacciones el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

## CAPÍTULO 2. Descripción de la solución propuesta

Ref. #	Casos de Uso	Prioridad
CUS 1	CUS Comenzar juego	Crítico
CUS 2	CUS Controlar eventos	Secundario
CUS 3	CUS Controlar tiempo	Secundario
CUS 4	CUS Realizar actividad	Crítico
CUS 5	CUS Evaluar jugador	Crítico
CUS 6	CUS Finalizar juego	Crítico
CUS 7	CUS Gestionar contenido del juego	Crítico
CUS 8	CUS Controlar funcionalidades de Rescatando al Turpial	Crítico
CUS 9	CUS Controlar funcionalidades de Descubre la Imagen1	Crítico
CUS 10	CUS Controlar funcionalidades de Ensartando Aros	Crítico
CUS 11	CUS Controlar funcionalidades de Rompecabezas	Crítico
CUS 12	CUS Controlar funcionalidades de Escalera de Palabras	Crítico

Tabla 1. Descripción Textual del CUS Comenzar juego

<b>CUS 1 Comenzar juego</b>	
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el usuario selecciona la imagen que representa el módulo Juegos.
<b>Propósito</b>	Permitir al usuario comenzar un juego respondiendo todas las actividades correspondientes a este.
<b>Referencias</b>	RF 1, RF1.1, RF 1.2, RF 1.3, RF 2, RF 2.1, RF 2.2, RF 2.3, RF 2.4, RF 2.5
<b>Precondiciones</b>	El usuario debe entrar como invitado, estudiante o profesor para comenzar el juego.
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario selecciona la imagen de identificación del módulo Juegos.	2- El sistema muestra el patio con la imagen de identificación de todos los juegos.
3- El usuario selecciona la imagen de identificación del juego que desea jugar.	4- El sistema carga las imágenes del juego seleccionado y permite interactuar: ver Caso de Uso 2. 5- El sistema muestra un mensaje de orientación indicando los pasos a seguir para jugar correctamente.



6- El usuario presiona la opción aceptar para comenzar a jugar.	7- El sistema carga todas las imágenes del juego. 8- Carga todos los datos de los jugadores. 9- Muestra el área de estadística del jugador. 10- Muestra el área del jugador en turno. 11- Muestra el tiempo: ver Caso de Uso 3.
12- El usuario presiona la opción de comenzar las actividades del juego.	13- El sistema permite realizar las actividades: ver Caso de Uso 4. 14- El sistema permite evaluar al jugador: ver Caso de Uso 5.
15- El usuario termina las actividades del juego y presiona la opción de terminar.	16- El sistema finaliza el juego: ver el Caso de Uso 6.
<b>Prioridad</b>	Crítico

Tabla 2. Descripción Textual del CUS Controlar eventos

<b>CUS 2 Controlar eventos</b>	
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el usuario comienza interactuar con los eventos del juego.
<b>Propósito</b>	Permitir que el usuario perciba los efectos que se muestran en los juegos mientras interactúa.
<b>Referencias</b>	RF 3, RF 3.1, RF 3.2, RF 3.3
<b>Precondiciones</b>	El usuario debe hacer uso del mouse.
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario interactúa con las opciones del juego.	2- El sistema actúa en dependencia del evento ejecutado por el usuario: <ul style="list-style-type: none"> <li>- Si posiciona el mouse sobre alguna imagen (MouseOver). Ir al <b>Escenario 1</b>.</li> <li>- Si posiciona el mouse fuera de alguna imagen (MouseOut), Ir al <b>Escenario 2</b>.</li> <li>- Si da clic sobre alguna imagen. Ir al <b>Escenario 3</b>.</li> </ul>
<b>Escenario 1: Controlar evento MouseOver</b>	
	3- El sistema cambia la imagen por otra de diferente color, logrando así el efecto MouseOver.
<b>Escenario 2: Controlar evento MouseOut</b>	

	3- El sistema cambia la imagen por la que estaba cuando el evento MouseOver por la que estaba inicialmente, es decir, cuando estaba en reposo.
<b>Escenario 3:</b> Controlar evento Clic	
	3- El sistema cambia la imagen por otra permitiendo iniciar el juego. 4- Se pone visible todas las preguntas del juego. 5- Tiene lugar el caso de uso Controlar eventos.
<b>Prioridad</b>	Secundario

Tabla 3. Descripción Textual del CUS Controlar tiempo

<b>CUS 3 Controlar tiempo</b>	
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el jugador entra al juego.
<b>Propósito</b>	Permitir determinar el tiempo que demora un jugador en realizar todas las actividades de un juego.
<b>Referencias</b>	RF 4, RF 4.1, RF 4.2, RF 4.3, RF 4.4
<b>Precondiciones</b>	El usuario debe haber presionado la opción de acceso al juego.
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario interactúa con las actividades del juego.	2- El sistema comienza a contar el tiempo transcurrido en el juego por el usuario: <ul style="list-style-type: none"> <li>- Si comienza las actividades en el juego va al <b>Escenario 1.</b></li> <li>- Si termina las actividades en el juego va al <b>Escenario 2.</b></li> <li>- Mostrar el tiempo transcurrido en segundos</li> </ul>
<b>Escenario 1:</b> Contar tiempo desde que inicia el juego	
	2- El sistema inicia el conteo del tiempo desde que el usuario comienza una actividad en el juego.
<b>Escenario 2:</b> Controlar tiempo cuando termina el juego	
	2- El sistema cuenta el tiempo total cuando el usuario terminó las actividades del juego. 3- El sistema calcula el tiempo que demoró en realizar

	las actividades del juego en segundos. 4- Tiene lugar el caso de uso Controlar tiempo.
<b>Prioridad</b>	Secundario

Tabla 4. Descripción Textual del CUS Realizar actividad

<b>CUS 4 Realizar actividad</b>	
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el usuario desea avanzar en el juego.
<b>Propósito</b>	Permitir al usuario avanzar en el juego mientras responda correctamente las actividades.
<b>Referencias</b>	RF 5, RF 5.1, RF 5.2, RF 5.3
<b>Precondiciones</b>	El usuario debe haber presionado las opciones que permiten avanzar en el juego.
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario presiona algunas de las opciones que permiten avanzar en el juego.	2- El sistema muestra una ventana con la pregunta junto a sus posibles respuestas.
3- El usuario selecciona la respuesta y presiona la opción revisar.	4- El sistema valida la calidad de la respuesta y muestra una ventana de confirmación informando que la respuesta es correcta.
<b>Flujos Alternos</b>	
<b>4. Respuesta incorrecta</b>	
	4.1- Si la respuesta es incorrecta el sistema muestra una ventana de confirmación informando que la respuesta es incorrecta. Pasa a la sección 1.
<b>Prioridad</b>	Crítico

Tabla 5. Descripción Textual del CUS Evaluar jugador

<b>CUS 5 Evaluar jugador</b>	
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia una vez que el usuario realice una actividad en el juego.
<b>Propósito</b>	Permitir al usuario saber los resultados de las respuestas y los puntos en cada actividad realizada.

<b>Referencias</b>	RF 6, RF 6.1, RF 6.2, RF 6.3
<b>Precondiciones</b>	El usuario debe haber realizado algunas de las actividades en el juego.
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario realiza una de las actividades de los juegos y selecciona la opción revisar.	2- El sistema muestra un mensaje informando al usuario si la respuesta es correcta o incorrecta.
3- El usuario presiona la opción aceptar.	4- El sistema otorga la calificación en dependencia de la respuesta del usuario. 5- El sistema asigna la efectividad en relación de la respuesta del usuario. 6- El sistema guarda el tiempo transcurrido en realizar una actividad.
<b>Prioridad</b>	Crítico

Tabla 6. Descripción Textual del CUS Finalizar juego

<b>CUS 6 Finalizar juego</b>	
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el usuario termina todas las actividades del juego.
<b>Propósito</b>	Permitirle al usuario ver todos los resultados del juego con el ganador.
<b>Referencias</b>	RF 8, RF 8.1
<b>Precondiciones</b>	El usuario debe haber terminado todas las actividades del juego.
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario termina la última actividad en el juego.	2- El sistema muestra un mensaje de información diciéndole al jugador que ha terminado y que vea los resultados.
3- El usuario presiona la opción aceptar para ver los resultados del juego.	4- El sistema muestra el podio con los nombres de los jugadores ordenados en dependencia de los resultados en el juego.
<b>Prioridad</b>	Crítico

Tabla 7. Descripción Textual del CUS Gestionar contenido del juego

<b>CUS 7 Gestionar contenido del juego</b>	
<b>Actores</b>	Maestro
<b>Resumen</b>	El caso de uso se inicia cuando el maestro accede al panel de configuración de los juegos y desea insertar, eliminar o modificar el contenido del juego Escalera de Palabras.
<b>Propósito</b>	Brindar la posibilidad de insertar, eliminar o modificar el contenido de un juego.
<b>Referencias</b>	RF 7, RF 7.1
<b>Precondiciones</b>	Para que se ejecute satisfactoriamente algunas de estas funcionalidades el maestro primeramente debe llenar todos los campos que se muestran.
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El caso de uso se inicia cuando el maestro selecciona el juego que desea gestionarle el contenido.	2- El sistema muestra el contenido actual del juego seleccionado junto a las opciones de eliminar, insertar o modificar contenido.
3- El maestro selecciona una de las opciones mostradas.	4- El sistema actúa en dependencia de la opción seleccionada: -Insertar. Ver <b>Sección 1</b> . -Eliminar. Ver <b>Sección 2</b> . -Modificar. Ver <b>Sección 3</b> .
<b>Sección 1:</b> Insertar contenido en un juego.	
1- El maestro selecciona la opción de insertar contenido.	2- El sistema muestra un formulario con los campos que tiene que llenar.
3- El maestro llena los campos.	4- El sistema muestra una ventana de confirmación informando que los datos han sido insertados satisfactoriamente.
<b>Sección 2:</b> Modificar el contenido de un juego.	
1- El maestro selecciona la opción de modificar contenido.	2- El sistema brinda la posibilidad de modificar todo el contenido mostrado en el panel.
3- El maestro realiza los cambios y presiona la opción de aceptar.	4- El sistema guarda los cambios y muestra una ventana de conformación informando que los cambios se han efectuado satisfactoriamente.
<b>Sección 3:</b> Eliminar el contenido de un juego.	

1- El maestro selecciona la opción de eliminar contenido.	2- El sistema muestra un formulario donde el maestro puede entrar el identificador del contenido que desea eliminar.
3- El maestro entra el dato que se le pide y presiona la opción de aceptar.	4- El sistema elimina el contenido y muestra una ventana de confirmación informando que el contenido se ha eliminado satisfactoriamente.
<b>Flujos Alternos</b>	
<b>4 Entradas de datos incorrectos</b>	
<b>Sección 1:</b> Insertar contenido en un juego.	
	4.1- El sistema muestra una ventana de confirmación informando los datos no son válidos y muestra nuevamente el formulario para que este sea llenado correctamente.
<b>Sección 3:</b> Eliminar el contenido de un juego.	
	4.1- El sistema muestra una ventana de confirmación informando que olvidó llenar los datos, o el dato que desea eliminar no existe en la base de datos.
<b>Prioridad</b>	Crítico

Tabla 8. Descripción Textual del CUS Controlar funcionalidades de Rescatando al Turpial.

<b>CUS 8 Controlar funcionalidades de Rescatando al Turpial</b>	
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el usuario comienza a realizar las actividades del juego Rescatando al Turpial.
<b>Propósito</b>	Permitir controlar las funcionalidades específicas del juego Rescatando al Turpial.
<b>Referencias</b>	RF 9, RF 9.1, RF 9.2, RF 9.3, RF 9.4
<b>Precondiciones</b>	El usuario debe haber comenzado las actividades en el juego.
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario selecciona una de las dos puertas que se le muestran.	2- El sistema muestra la actividad que debe realizar con las posibles respuestas.
3- El usuario selecciona una de las respuestas que se muestran	4- El sistema muestra un mensaje informando que la respuesta es correcta.

y presiona la opción revisar.	5- El sistema muestra la siguiente sala y permite que el usuario avance. 6- El sistema va mostrando el localizador de la sala. 7- El sistema después de haber pasado por las salas muestra un mensaje de felicitación.
8- El usuario selecciona la opción aceptar y presiona la imagen del Turpial en la jaula.	9- El sistema muestra un mensaje de información indicando que ha terminado el juego. 10- El sistema muestra la imagen de liberar al turpial.
11- El usuario selecciona la opción aceptar.	
<b>Flujos Alternos</b>	
<b>4 Respuesta incorrecta</b>	
	4.1- Muestra un mensaje informando que la respuesta es incorrecta.
4.2- El usuario presiona la opción aceptar.	4.3- El sistema clausura la puerta y muestra otro mensaje informando que puede intentarlo por la otra puerta.
4.4- El usuario presiona la opción aceptar y selecciona la imagen de la otra puerta.	4.5- El sistema muestra la actividad que debe realizar con las posibles respuestas.
4.6- El usuario selecciona la respuesta y oprime la opción revisar.	4.7- Si la respuesta es correcta pasa a la sección 4. 4.8- Si la respuesta es incorrecta el sistema muestra un mensaje informando que no hay más entrada, que lo intente en otra ocasión y muestra el principio del juego.
<b>Prioridad</b>	Crítico

Tabla 9. Descripción Textual del CUS Controlar funcionalidades de Descubre la Imagen 1

<b>CUS 9 Controlar funcionalidades de Descubre la Imagen 1</b>	
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el usuario comienza a realizar las actividades del juego Descubre la Imagen1.
<b>Propósito</b>	Permitir controlar las funcionalidades específicas del juego Descubre la Imagen1.
<b>Referencias</b>	RF 10, RF 10.1, RF 10.2, RF 10.3, RF 10.4
<b>Precondiciones</b>	El usuario debe comenzar las actividades del juego.
<b>Flujo normal de eventos</b>	

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario presiona sobre la imagen de identificación de las mascotas.	2- El sistema selecciona de forma aleatoria la mascota inicial. 3- Si la mascota es el cangrejo muestra un mensaje informando mala suerte inténtalo de nuevo.
4- El usuario presiona la opción aceptar.	5- El sistema retrocede la mascota un paso atrás.
6- El usuario nuevamente presiona la imagen de identificación de las mascotas.	7- El sistema muestra la actividad con las posibles respuestas.
8- El usuario selecciona la respuesta y selecciona la opción revisar.	9- El sistema comienza a completar parte de la imagen. 10- El sistema permite avanzar en el juego moviendo la mascota hacia la meta. 11- El sistema muestra la imagen descubierta completamente.
<b>Flujos Alternos</b>	
<b>5 Respuesta incorrecta</b>	
	5.1- El sistema muestra un mensaje informando que la respuesta está incorrecta.
5.2- El usuario presiona la opción aceptar.	5.3- El sistema muestra nuevamente las actividades con las posibles respuestas.
5.4- El usuario selecciona la respuesta y oprime la opción revisar.	5.5- Si la respuesta es correcta pasa a la sección 9.
<b>Prioridad</b>	Crítico

Tabla 10. Descripción Textual del CUS Controlar funcionalidades de Ensartando Aros

<b>CUS 10 Controlar funcionalidades de Ensartando Aros</b>	
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el usuario comienza a realizar las actividades del juego Ensartando Aros.
<b>Propósito</b>	Permitir controlar las funcionalidades específicas del juego Ensartando Aros.
<b>Referencias</b>	RF 11, RF 11.1
<b>Precondiciones</b>	El usuario debe comenzar las actividades del juego.
<b>Flujo normal de eventos</b>	



<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario presiona sobre los aros.	2- El sistema muestra la pregunta con las posibles respuestas.
3- El usuario selecciona la respuesta y presiona la opción revisar.	4- El sistema muestra un mensaje de felicitación que la respuesta está correcta.
5- El usuario presiona la opción aceptar.	6- El sistema lanza el aro y permite ensartarlo correctamente.
<b>Flujos Alternos</b>	
<b>4 Respuesta incorrecta</b>	
	4.1- El sistema muestra un mensaje informando que la respuesta es incorrecta.
4.2- El usuario presiona la opción aceptar.	4.3- El sistema no ensarta el aro, lo mueve fuera de lugar.
<b>Prioridad</b>	Crítico

Tabla 11. Descripción Textual del CUS Controlar funcionalidades de Rompecabezas

<b>CUS 11 Controlar funcionalidades de Rompecabezas</b>	
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el usuario entra en el juego y selecciona el Rompecabezas que desea armar.
<b>Propósito</b>	Permitir controlar las funcionalidades específicas del juego Rompecabezas.
<b>Referencias</b>	RF 12, RF 12.1, RF 12.2, RF 12.3, RF 12.4, RF 12.5, RF 12.6, RF 12.7, RF 12.8, RF 12.9, RF 12.10
<b>Precondiciones</b>	El usuario debe seleccionar el Rompecabezas que desea jugar.
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario selecciona el Rompecabezas que desea armar. 2- El usuario selecciona la cantidad de cortes y el tipo de cortes.	3- El sistema muestra el Rompecabezas seleccionado por el usuario.

4- El usuario presiona la opción aceptar.	5- El sistema carga las piezas del primer Rompecabezas que seleccionó.
6- El usuario mueve y rota las piezas hacia el lugar que él estima correcto.	7- El sistema verifica la acción ejecutada y en caso de ser correcta, ubica la pieza en las coordenadas exactas.
8- El usuario selecciona la opción siguiente para acceder al próximo Rompecabezas.	9- El sistema verifica si el Rompecabezas está armado correctamente y en caso de que esto sea verdadero pasa al próximo.
10- El usuario termina de armar correctamente el Rompecabezas seleccionado.	11- El sistema muestra un mensaje de felicitación informando que el Rompecabezas ha sido armado correctamente.
12- El usuario presiona la opción terminar.	13- El sistema muestra el podio con los resultados de los usuarios en el juego.
<b>Flujos Alternos</b>	
<b>7 Acción incorrecta</b>	
	7.1- En caso de que la acción ejecutada por el usuario sea incorrecta, el sistema ubica la pieza en su posición inicial y pasa nuevamente a la sección 6.
<b>9 Acción incorrecta</b>	
	9.1- En caso de que el Rompecabezas no esté armado correctamente, se mostrará una ventana de confirmación informando que usted no ha terminado de armar el Rompecabezas.
9.2- El usuario presiona la opción aceptar.	9.3- El sistema pasa nuevamente a la sección 6.
<b>Prioridad</b>	Crítico

Tabla 12. Descripción Textual del CUS Controlar funcionalidades de Escalera de Palabras

<b>CUS 12 Controlar funcionalidades de Escalera de Palabras</b>	
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el usuario entra en el juego Escalera de Palabras.
<b>Propósito</b>	Permitir controlar todas las funcionalidades específicas de Escalera de Palabras.
<b>Referencias</b>	RF 13, RF 13.1, RF 13.2, RF 13.3, RF 13.4, RF 13.5
<b>Precondiciones</b>	El usuario debe entrar en el juego Escalera de

	Palabras.
<b>Flujo normal de eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario selecciona una carta.	2- El sistema vira la carta y muestra la palabra que debe responder el usuario, si es aguda, llana o esdrújula.
3- El usuario mueve la carta hacia la respuesta.	4- El sistema resalta la respuesta en rojo. 5- El sistema aumenta la puntuación.
<b>Flujos Alternos</b>	
<b>3 Respuesta incorrecta</b>	
	3.1- El sistema no resalta la respuesta. 3.2- El sistema disminuye la puntuación. 3.3- El sistema guarda la carta.
3.4- El usuario selecciona nuevamente una carta.	3.5- El sistema vira la carta y muestra la palabra que debe responder y pasa a la sección 3.
<b>Prioridad</b>	Crítico

## 2.6 Conclusiones

En el presente capítulo quedaron definidos los principales conceptos asociados al dominio del problema, permitiendo realizar su modelación con el propósito de lograr un mejor entendimiento entre los conceptos identificados. También se determinaron los requisitos funcionales y no funcionales que debe cumplir el sistema, y con ello fueron expuestos los casos de uso con sus correspondientes descripciones. Este capítulo como propuesta de solución provee una visión general de las principales características por la que estará compuesto el sistema.



### 3.1 Introducción

En este capítulo se expone todo lo referente al análisis y diseño del sistema. Se presentan los Diagramas de Clases del Análisis correspondientes a los casos de uso identificados, constituyendo la base fundamental para realizar un buen diseño. El proceso de diseño tiene gran importancia porque define cómo va a ser construido el software. Aquí se presentan los Diagramas de Clases del Diseño, que tienen la finalidad de describir la interacción entre las distintas páginas de la aplicación, así como ofrecer a los desarrolladores una visión exacta de lo que desea implementar. Por otro lado, se presenta el Diagrama de Clases Persistentes de la base de datos del sistema, además de los principios de diseño y programación utilizados.

### 3.2 Modelo de Análisis

El Modelo de Análisis proporciona una comprensión detallada de los requisitos, también impone una estructura del sistema que se debe esforzar por conservar lo más posible cuando se le de forma al sistema. En la construcción del Modelo de Análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas, con toda esta información se construye el Diagrama de Clases del Análisis.

#### 3.2.1 Diagrama de Clases del Análisis

El Diagrama de Clases del Análisis es uno de los principales artefactos generados en este flujo de trabajo, las clases del análisis se pueden clasificar en:

**Clase Interfaz:** Estas clases modelan la interfaz del sistema y manejan la interacción entre el sistema y sus actores. Durante el diseño estas clases son refinadas, además de facilitar la comunicación con otros sistemas.

**Clase Entidad:** Estas clases modelan información y comportamiento asociado que generalmente son de larga duración. También reflejan entidades del mundo real que resultan necesarias para realizar tareas internas del sistema.

**Clase Control:** Estas clases coordinan el trabajo para la realización y la especificación de los casos de uso, es decir, son las que ejecutan el caso de uso, además de tener un control sobre todas las acciones a realizar.

A continuación se presentan los Diagramas de Clases del Análisis para cada uno de los casos de uso del sistema:

### 3.2.1.1 Diagrama de Clases del Análisis del caso de uso “Comenzar juego”

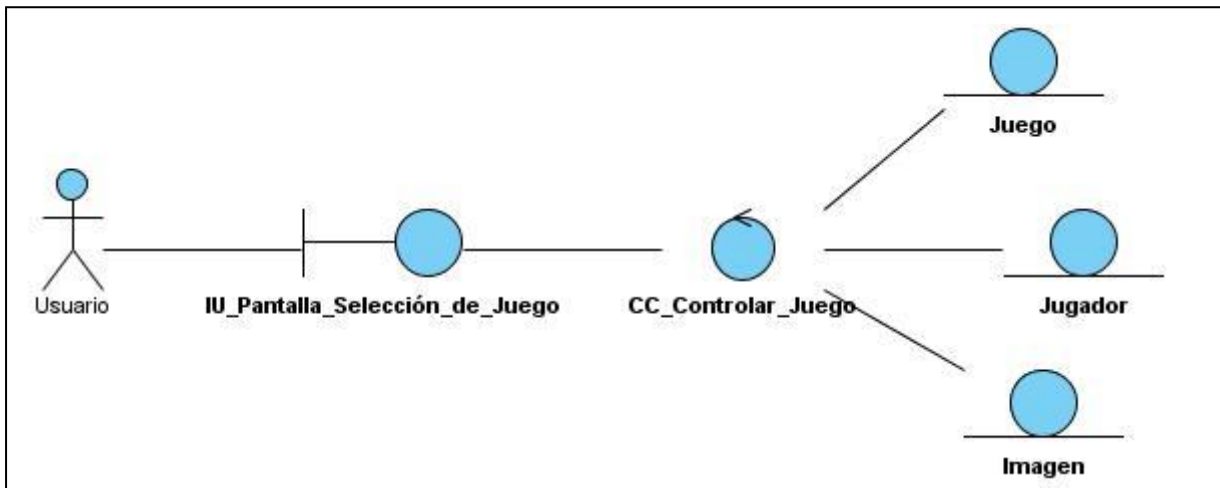


Figura 3. Diagrama de Clases del Análisis del caso de uso “Comenzar juego”

### 3.2.1.2 Diagrama de Clases del Análisis del caso de uso “Controlar eventos”

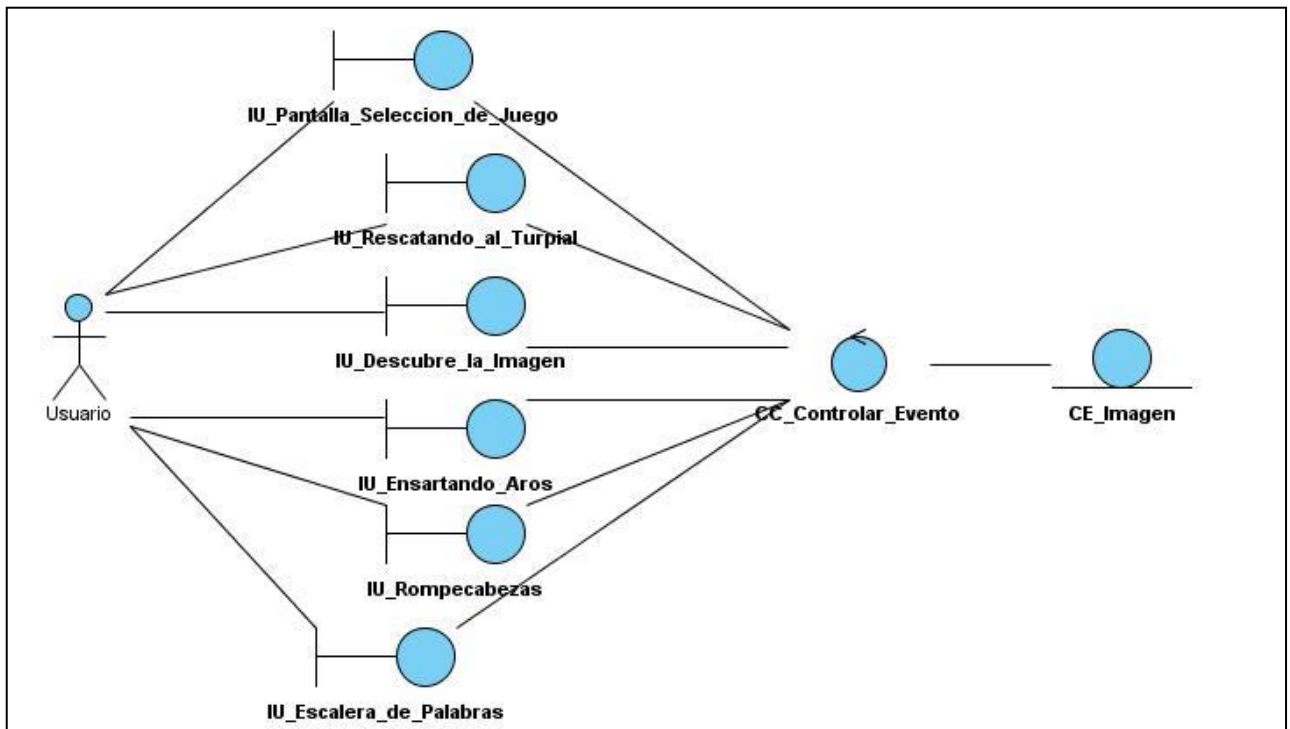


Figura 4. Diagrama de Clases del Análisis del caso de uso “Controlar eventos”

## 3.2.1.3 Diagrama de Clases del Análisis del caso de uso “Controlar tiempo”

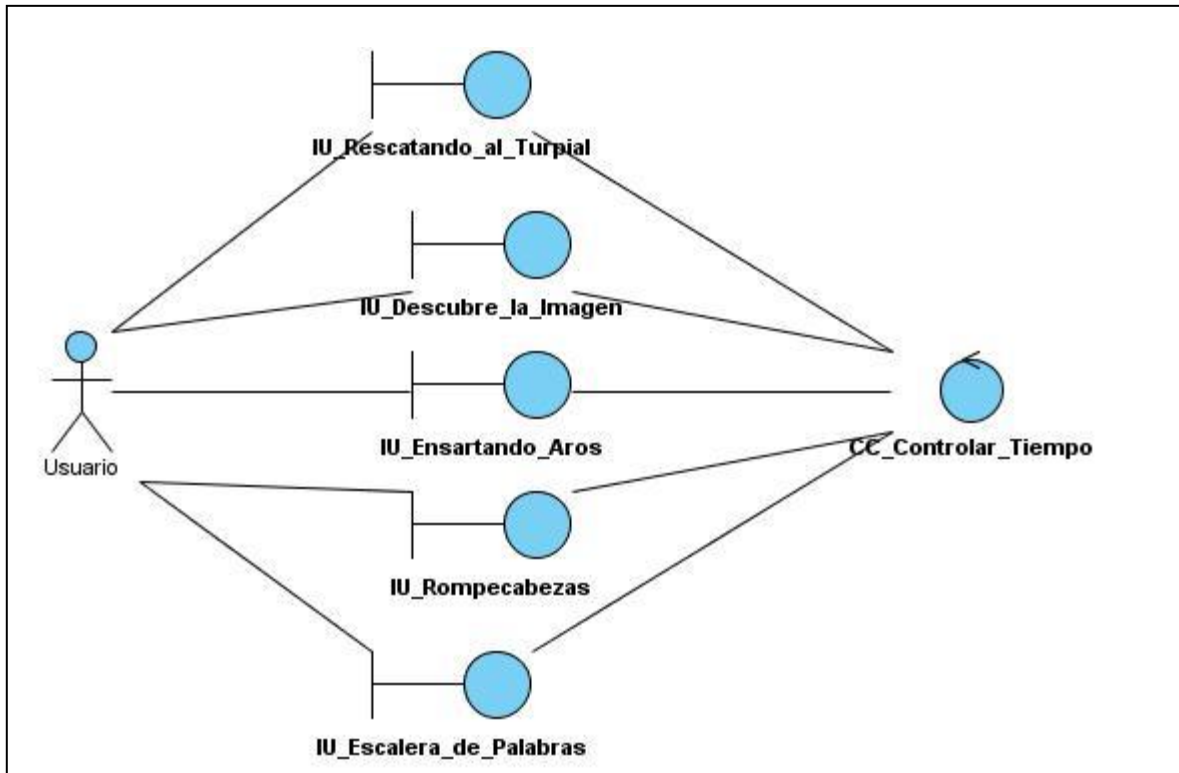


Figura 5. Diagrama de Clases del Análisis del caso de uso “Controlar tiempo”

## 3.2.1.4 Diagrama de Clases del Análisis del caso de uso “Realizar actividad”

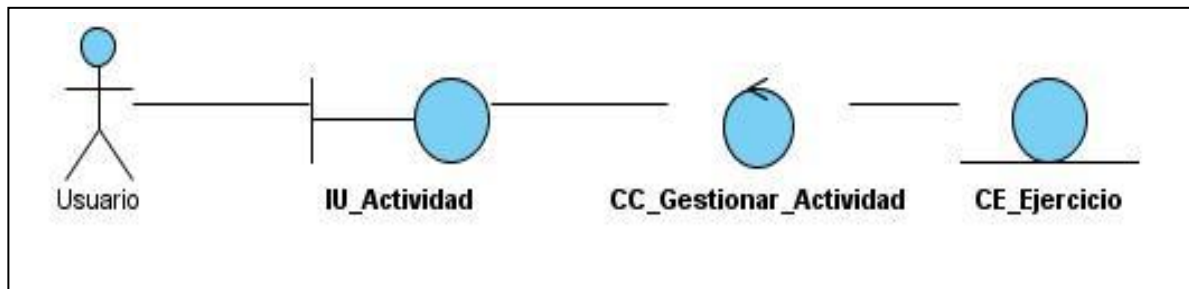


Figura 6. Diagrama de Clases del Análisis del caso de uso “Realizar actividad”

## 3.2.1.5 Diagrama de Clases del Análisis del caso de uso “Evaluar jugador”

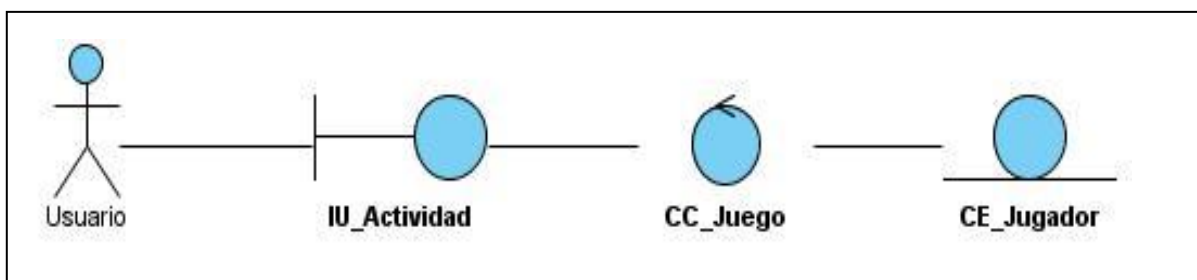


Figura 7. Diagrama de Clases del Análisis del caso de uso “Evaluar jugador”

## 3.2.1.6 Diagrama de Clases del Análisis del caso de uso “Finalizar juego”

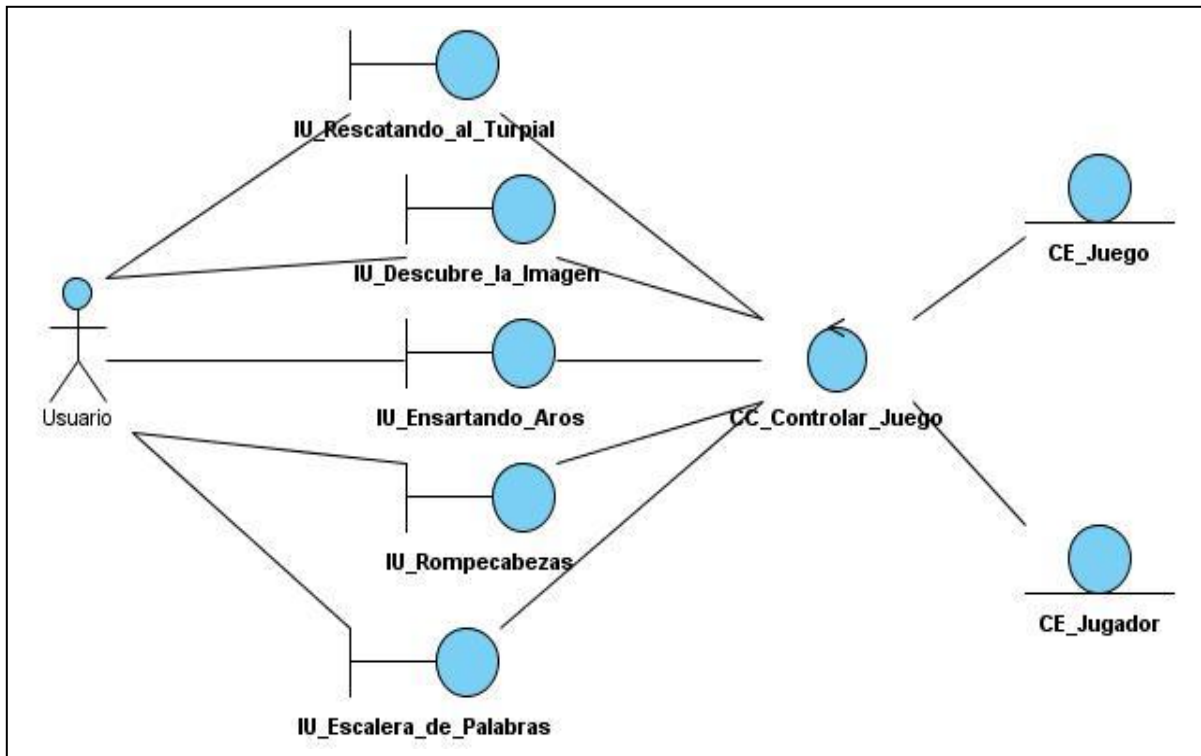


Figura 8. Diagrama de Clases del Análisis del caso de uso “Finalizar juego”

## 3.2.1.7 Diagrama de Clases del Análisis del caso de uso “Gestionar contenido del juego”

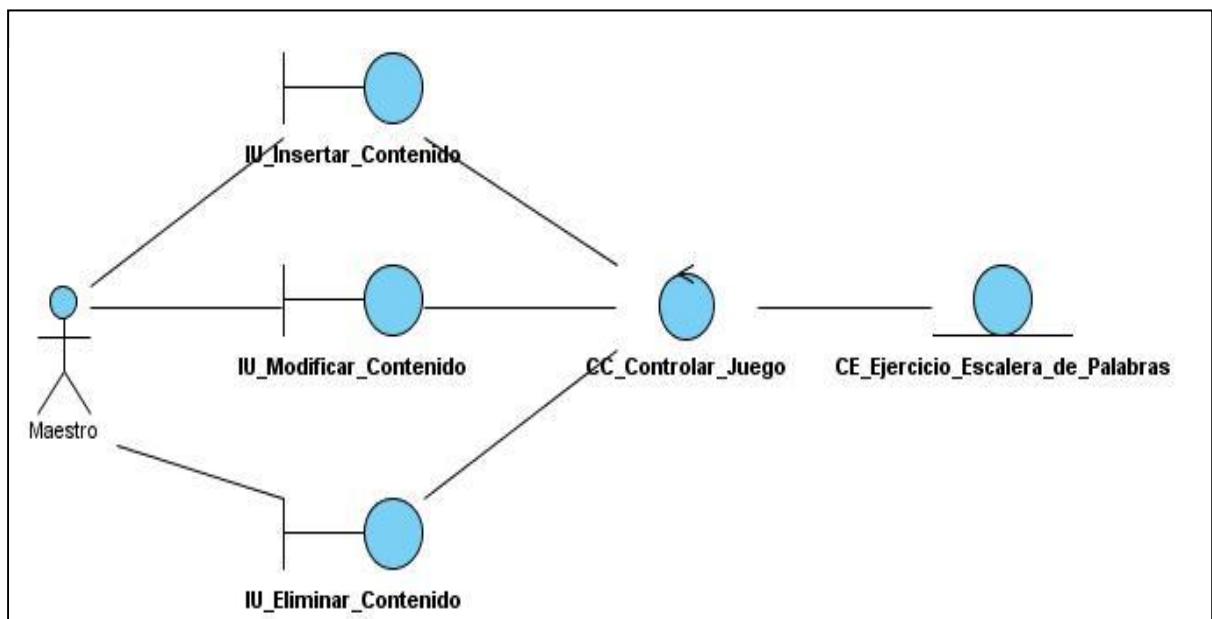


Figura 9. Diagrama de Clases del Análisis del caso de uso “Gestionar contenido del juego”

### 3.2.1.8 Diagrama de Clases del Análisis del caso de uso “Controlar funcionalidades de Rescatando al Turpial”

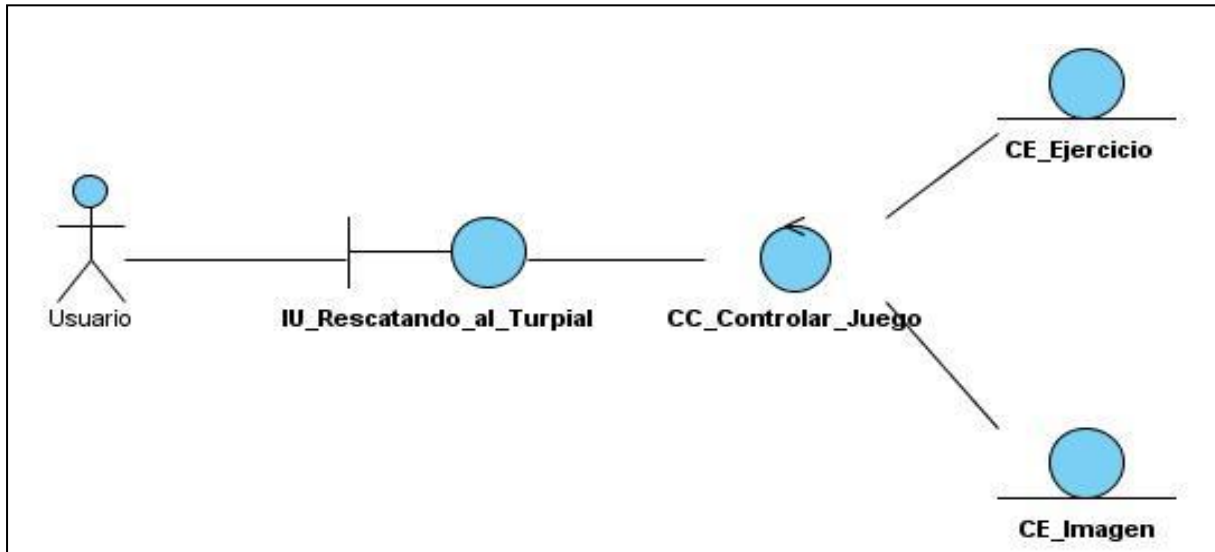


Figura 10. Diagrama de Clases del Análisis del caso de uso “Controlar funcionalidades de Rescatando al Turpial”

### 3.2.1.9 Diagrama de Clases del Análisis del caso de uso “Controlar funcionalidades de Descubre la Imagen 1”

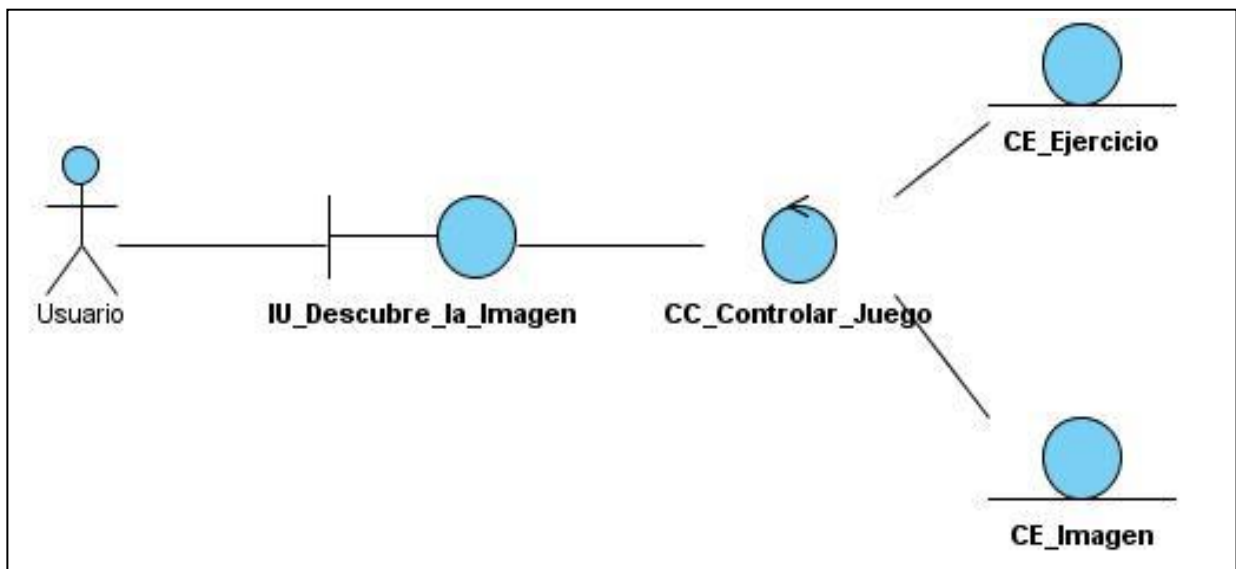


Figura 11. Diagrama de Clases del Análisis del caso de uso “Controlar funcionalidades de Descubre la Imagen 1”



### 3.2.1.10 Diagrama de Clases del Análisis del caso de uso “Controlar funcionalidades de Ensartando Aros”

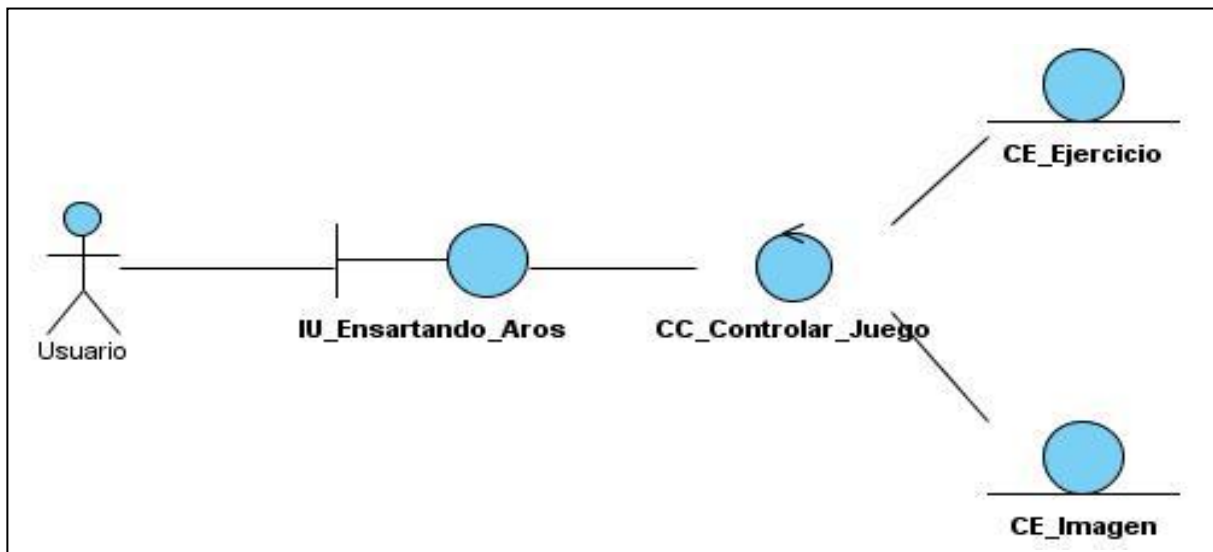


Figura 12. Diagrama de Clases del Análisis del caso de uso “Controlar funcionalidades de Ensartando Aros”

### 3.2.1.11 Diagrama de Clases del Análisis del caso de uso “Controlar funcionalidades de Rompecabezas”

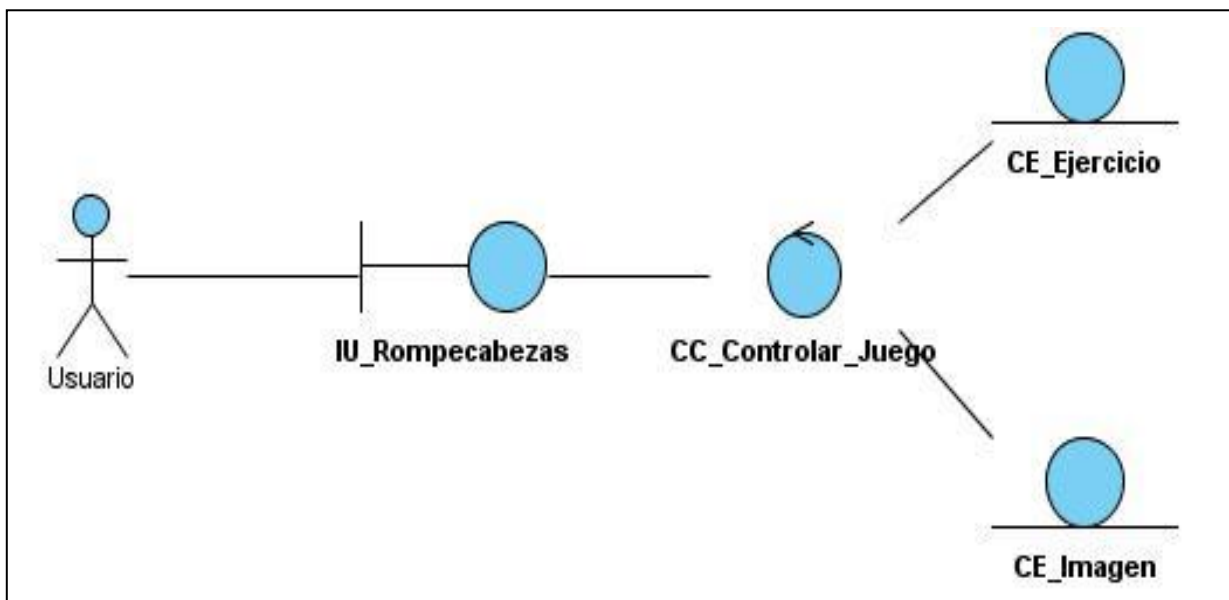


Figura 13. Diagrama de Clases del Análisis del caso de uso “Controlar funcionalidades de Rompecabezas”

### 3.2.1.12 Diagrama de Clases del Análisis del caso de uso “Controlar funcionalidades de Escalera de Palabras”

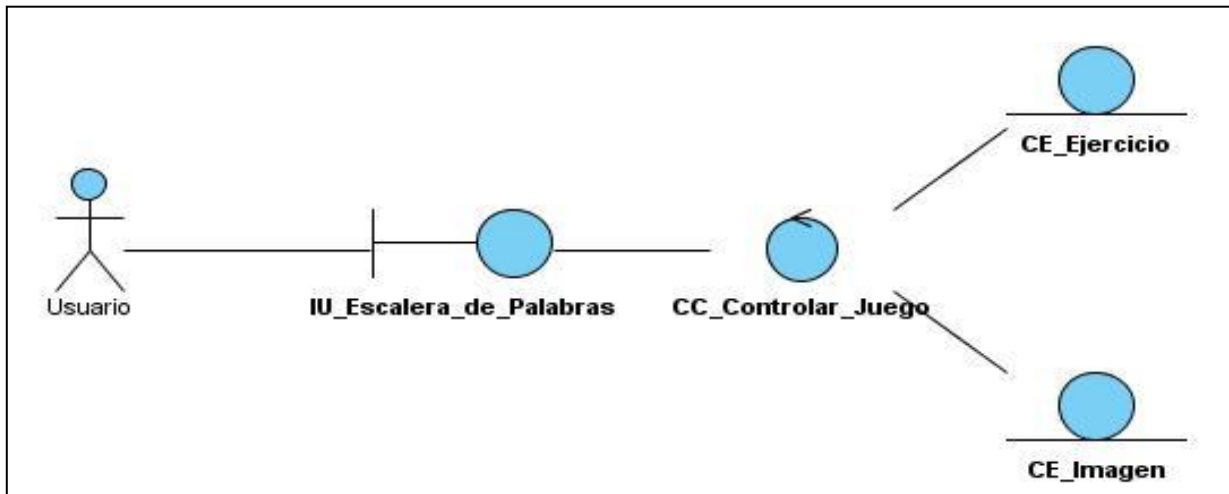


Figura 14. Diagrama de Clases de Análisis del caso de uso “Controlar funcionalidades de Escalera de Palabras”

### 3.3 Modelo del Diseño

La peculiaridad del diseño es modelar el sistema, encontrar su forma (incluida la arquitectura) para que soporte todos sus requisitos. Una entrada esencial en el diseño es el resultado del análisis, es decir, el Modelo del Análisis.

#### 3.3.1 Diagrama de Clases del Diseño

Para la modelación de la solución se proponen como clases del diseño los estándares para modelar aplicaciones web.

**Página Servidora:** Representa la página web que tiene código que se ejecuta en el servidor. Este código interactúa con recursos en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página.

**Página Cliente:** Una instancia de Página Cliente es una página web, con formato HTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del script que son accesibles para páginas cualquier función dentro de la página.

**Formulario:** Colección de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML.

# CAPÍTULO 3. Análisis y Diseño del Sistema

A continuación se presentan los Diagramas de Clases del Diseño para cada uno de los casos de uso del sistema:

## 3.3.1.1 Diagrama de Clases del Diseño del caso de uso “Comenzar juego”

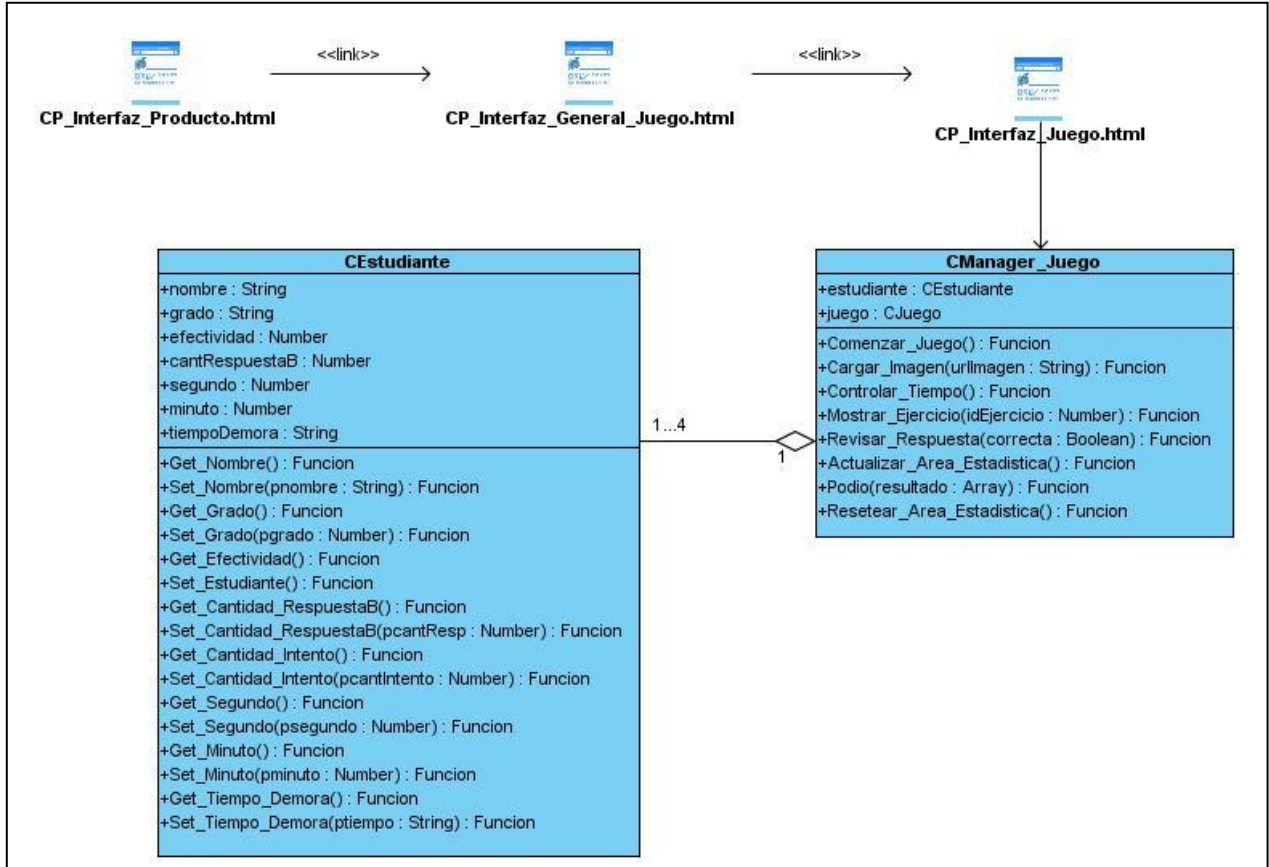


Figura 15. Diagrama de Clases del Diseño del caso de uso “Comenzar juego”

## 3.3.1.2 Diagrama de Clases del Diseño del caso de uso “Controlar eventos”



Figura 16. Diagrama de Clases del Diseño del caso de uso “Controlar eventos”

## 3.3.1.3 Diagrama de Clases del Diseño del caso de uso “Controlar tiempo”

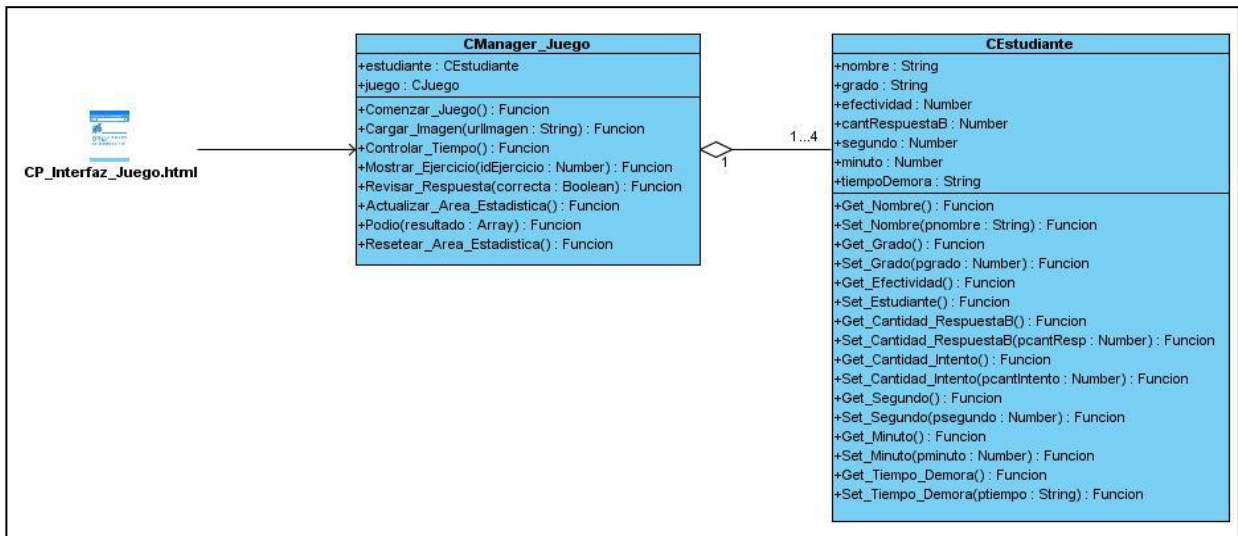


Figura 17. Diagrama de Clases del Diseño del caso de uso “Controlar tiempo”

## 3.3.1.4 Diagrama de Clases del Diseño del caso de uso “Realizar actividad”

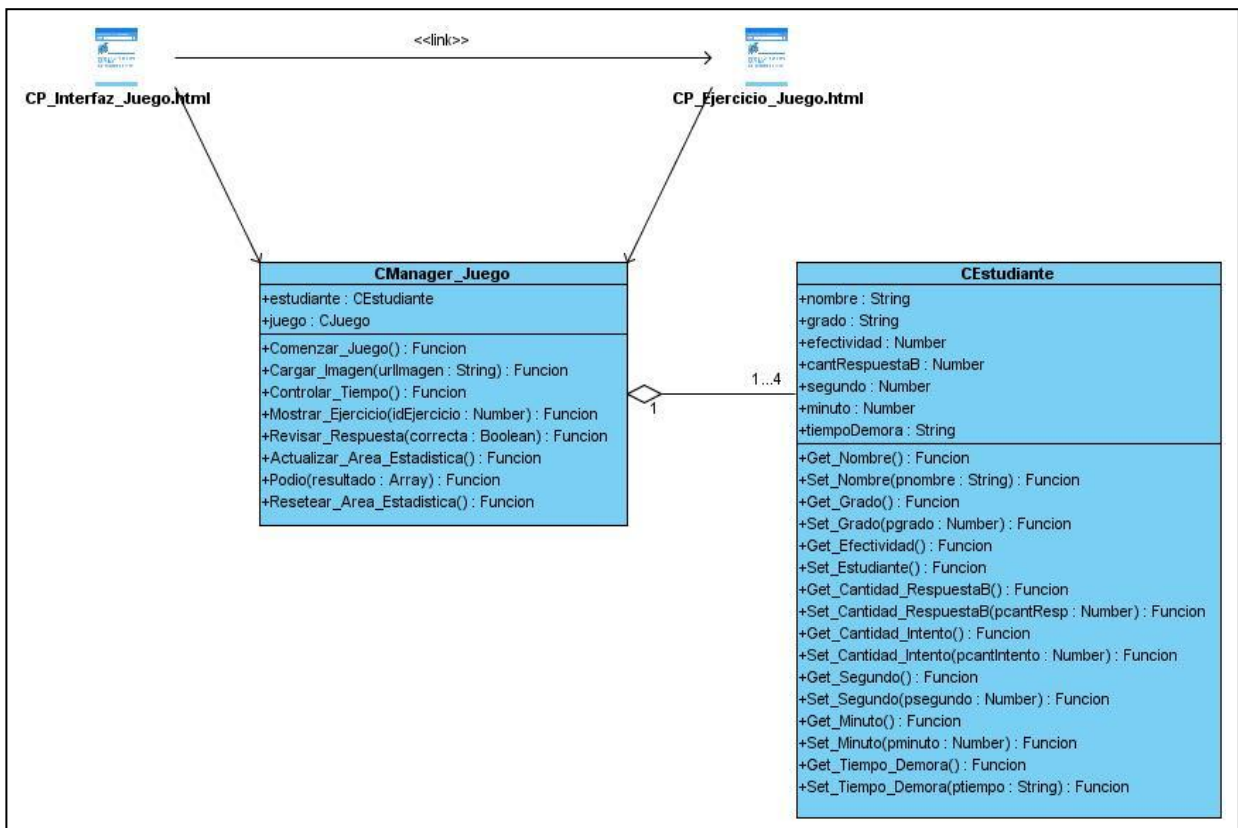


Figura 18. Diagrama de Clases del Diseño del caso de uso “Realizar actividad”

## 3.3.1.5 Diagrama de Clases del Diseño del caso de uso “Evaluar jugador”

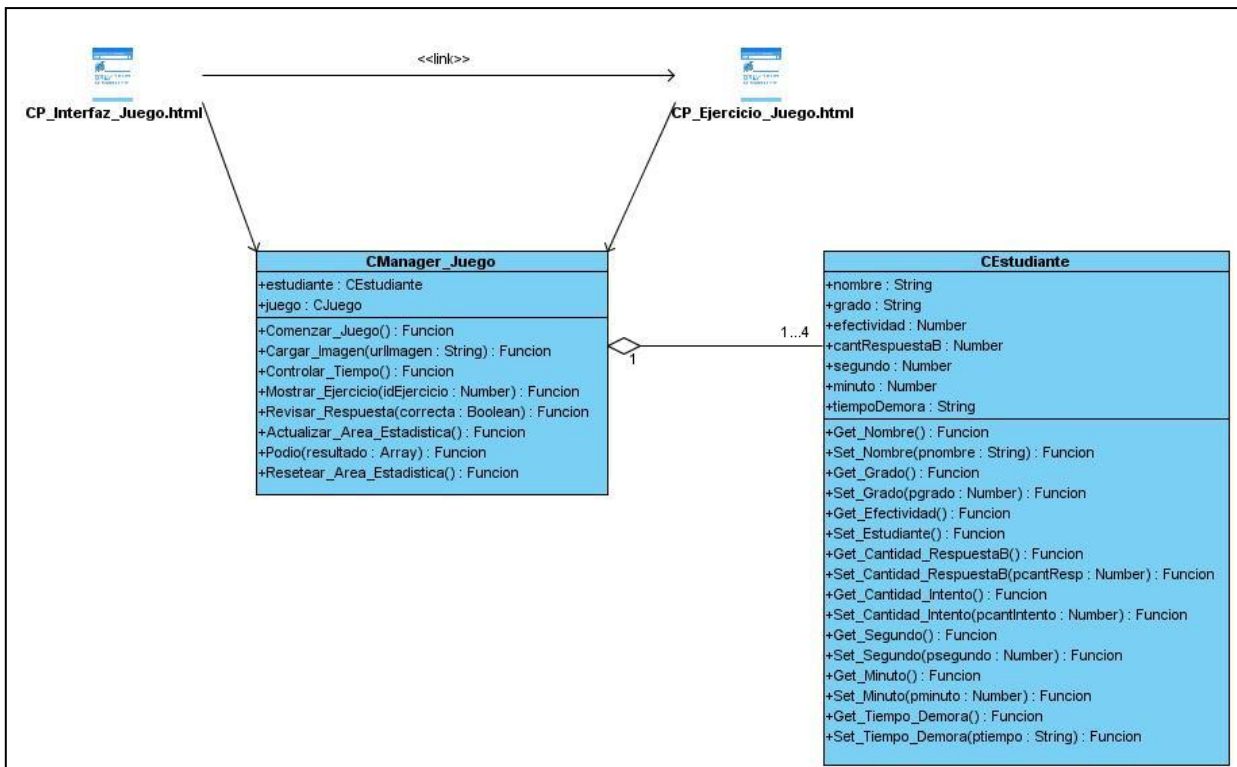


Figura 19. Diagrama de Clases del Diseño del caso de uso “Evaluar jugador”

## 3.3.1.6 Diagrama de Clases del Diseño del caso de uso “Finalizar juego”

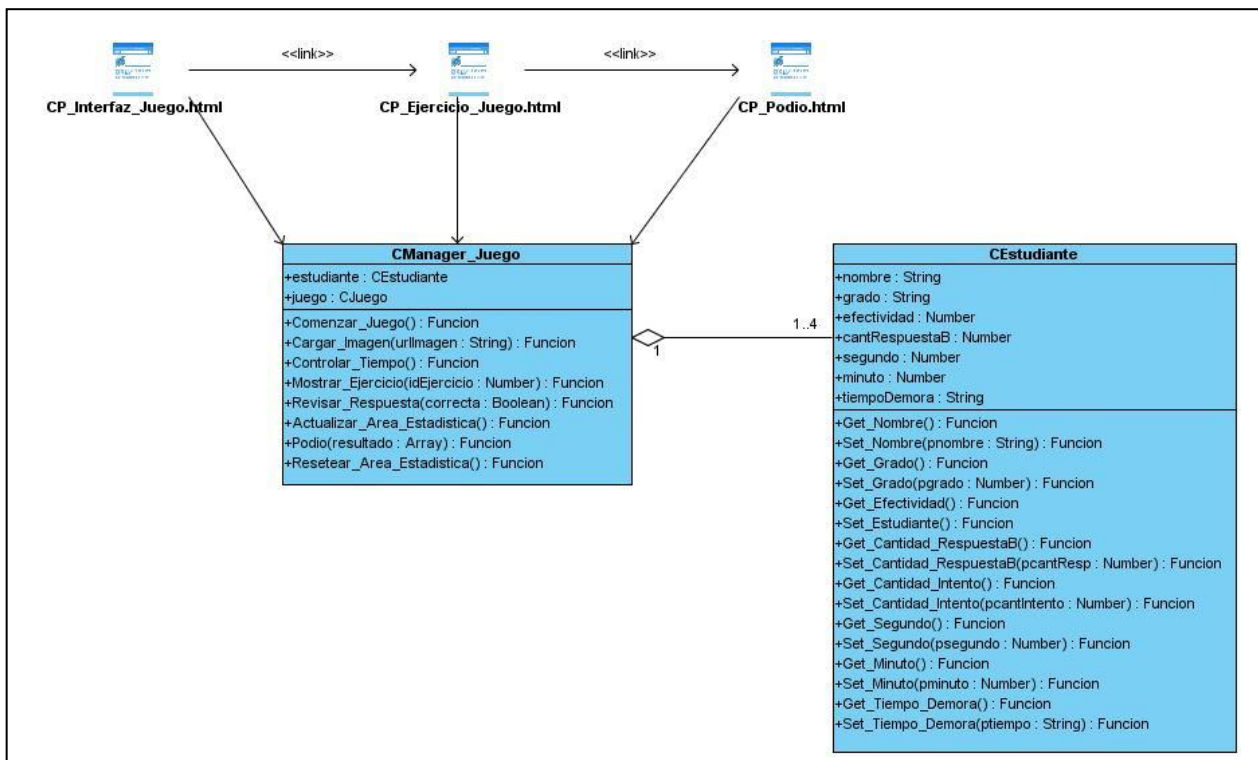


Figura 20. Diagrama de Clases del Diseño del caso de uso “Finalizar juego”

## 3.3.1.7 Diagrama de Clases del Diseño del caso de uso “Gestionar contenido del juego”

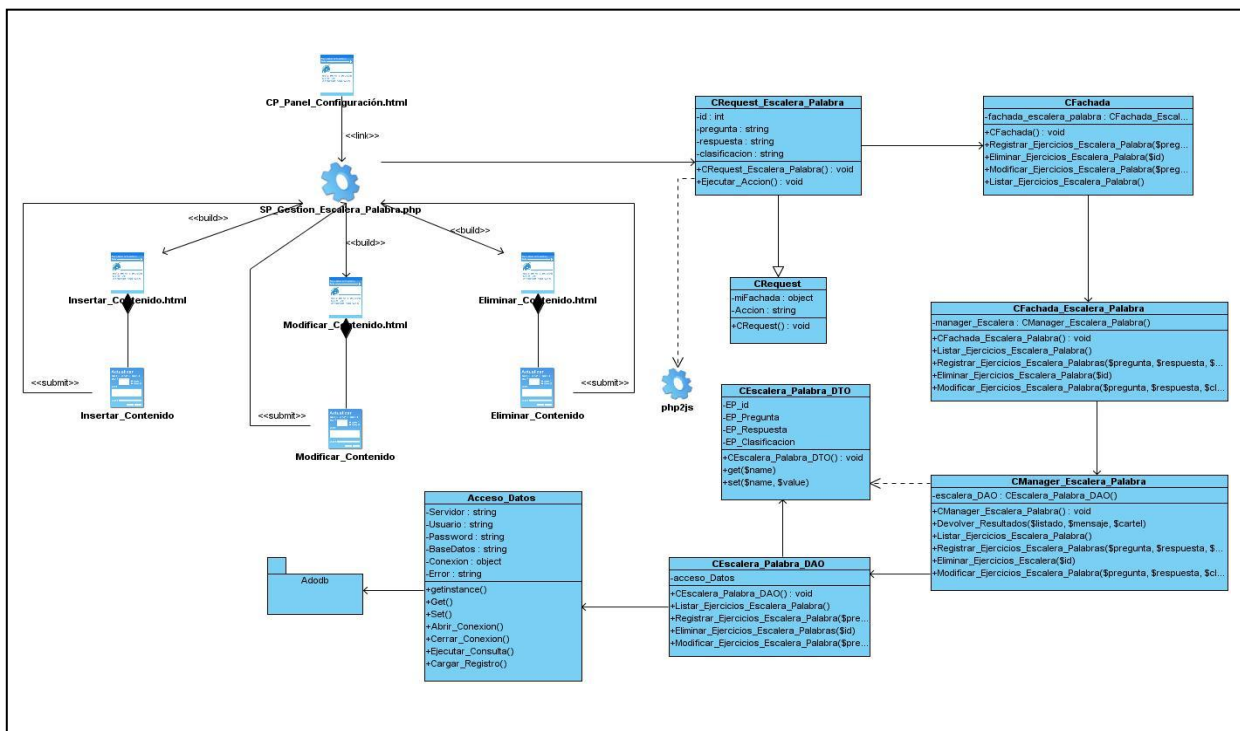


Figura 21. Diagrama de Clases del Diseño del caso de uso “Gestionar contenido del juego”

## 3.3.1.8 Diagrama de Clases del Diseño del caso de uso “Controlar funcionalidades de Rescatando al Turpial”

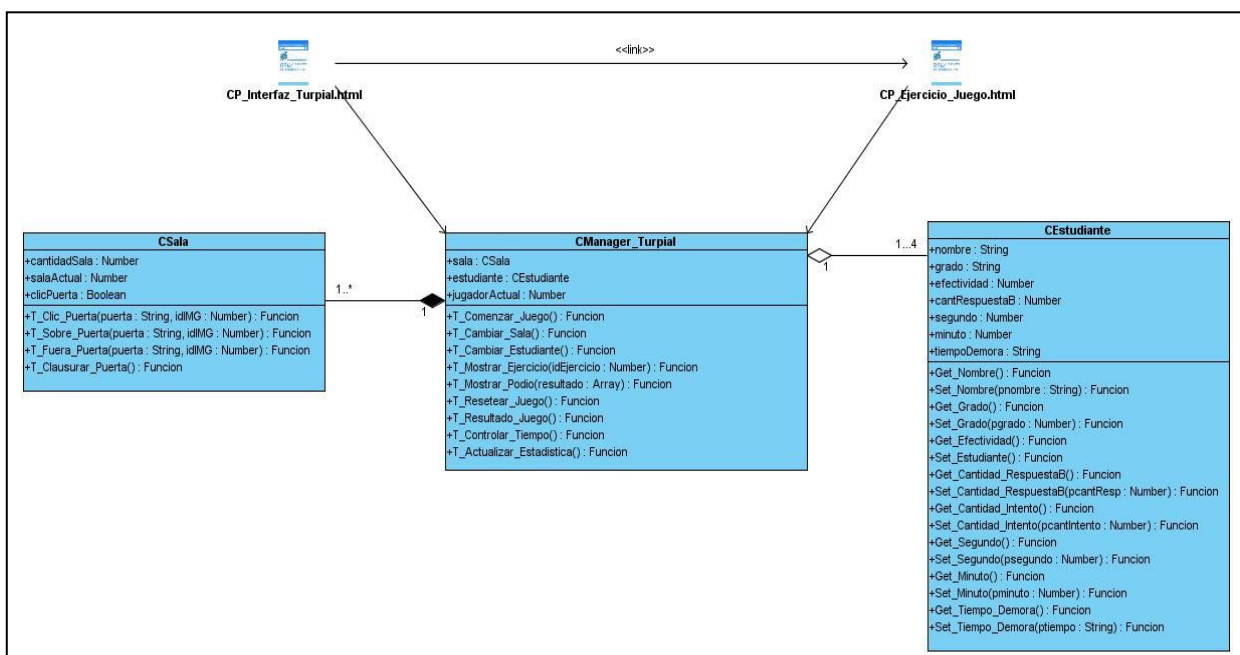


Figura 22. Diagrama de Clases del Diseño del caso de uso “Controlar funcionalidades de Rescatando al Turpial”

## 3.3.1.9 Diagrama de Clases del Diseño del caso de uso “Controlar funcionalidades Descubre la Imagen 1”

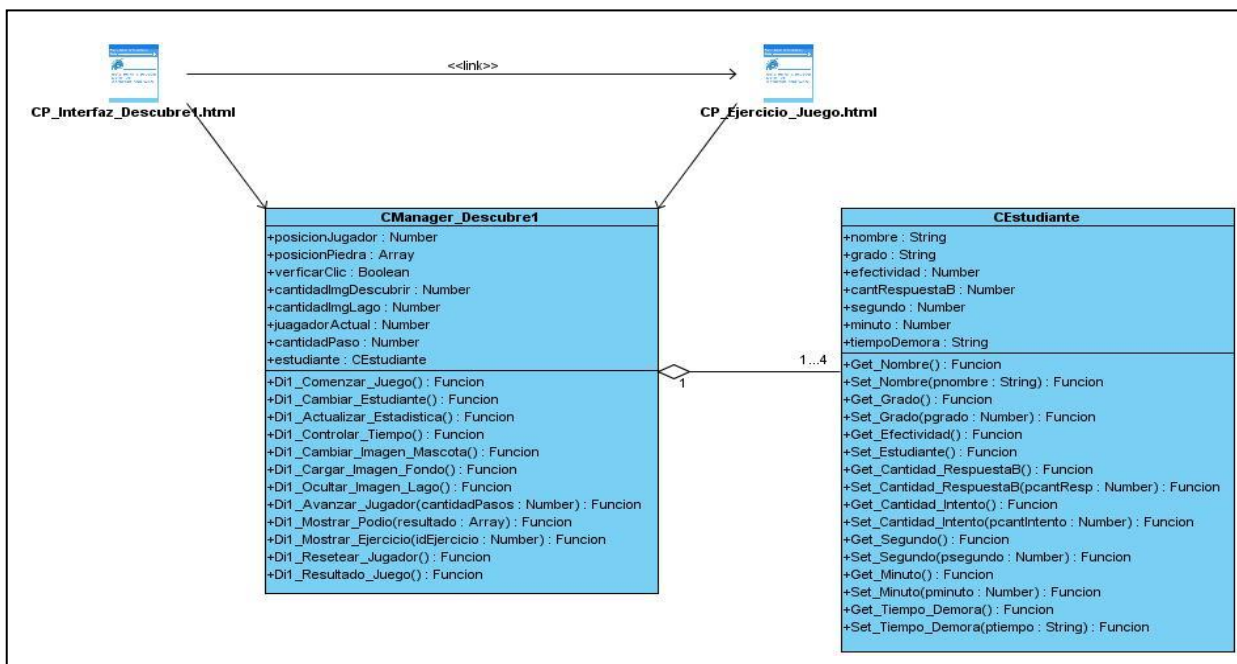


Figura 23. Diagrama de Clases del Diseño del caso de uso “Controlar funcionalidades de Descubre la Imagen 1”

## 3.3.1.10 Diagrama de Clases del Diseño del caso de uso “Controlar funcionalidades de Ensartando Aros”

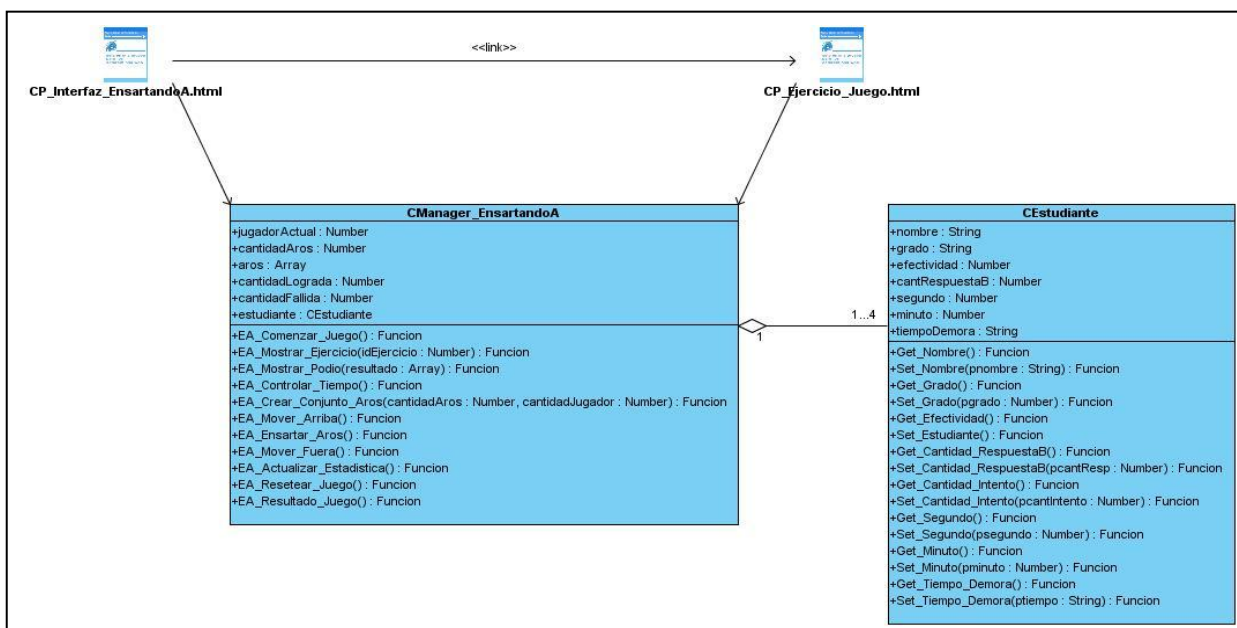


Figura 24. Diagrama de Clases del Diseño del caso de uso “Controlar funcionalidades de Ensartando Aros”

# CAPÍTULO 3. Análisis y Diseño del Sistema

## 3.3.1.11 Diagrama de Clases del Diseño del caso de uso “Controlar funcionalidades de Rompecabezas”

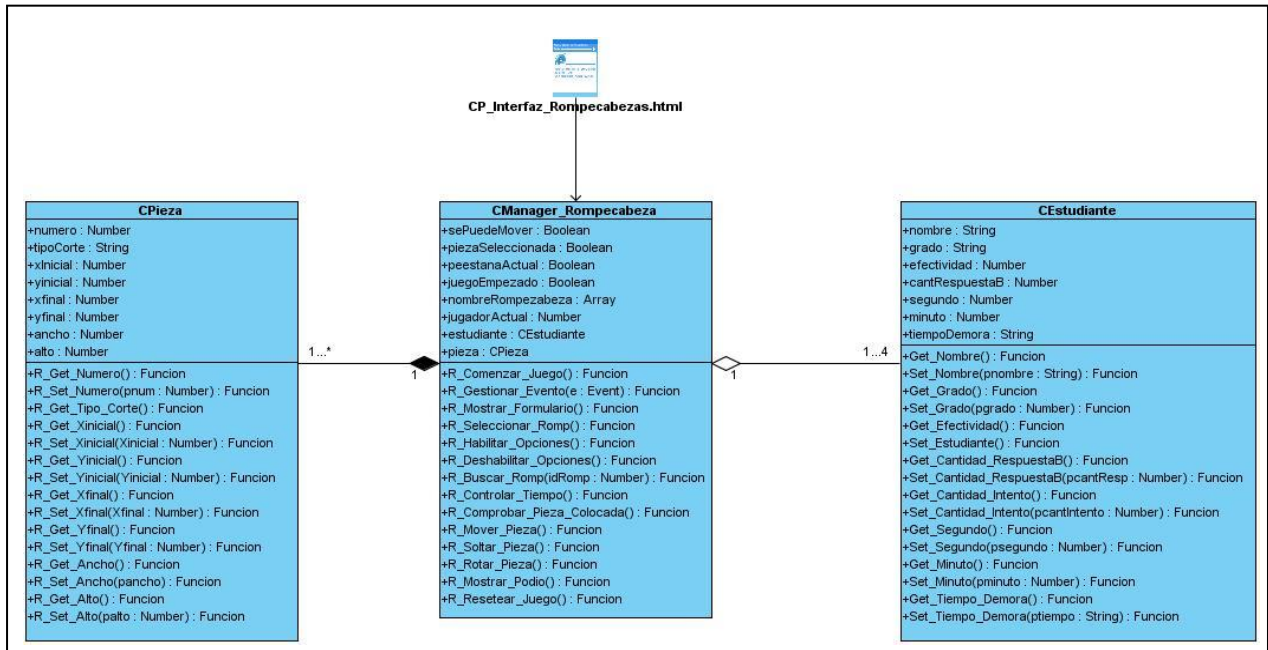


Figura 25. Diagrama de Clases del Diseño del caso de uso “Controlar funcionalidades de Rompecabezas”

## 3.3.1.12 Diagrama de Clases del Diseño del caso de uso “Controlar funcionalidades de Escalera de Palabras”

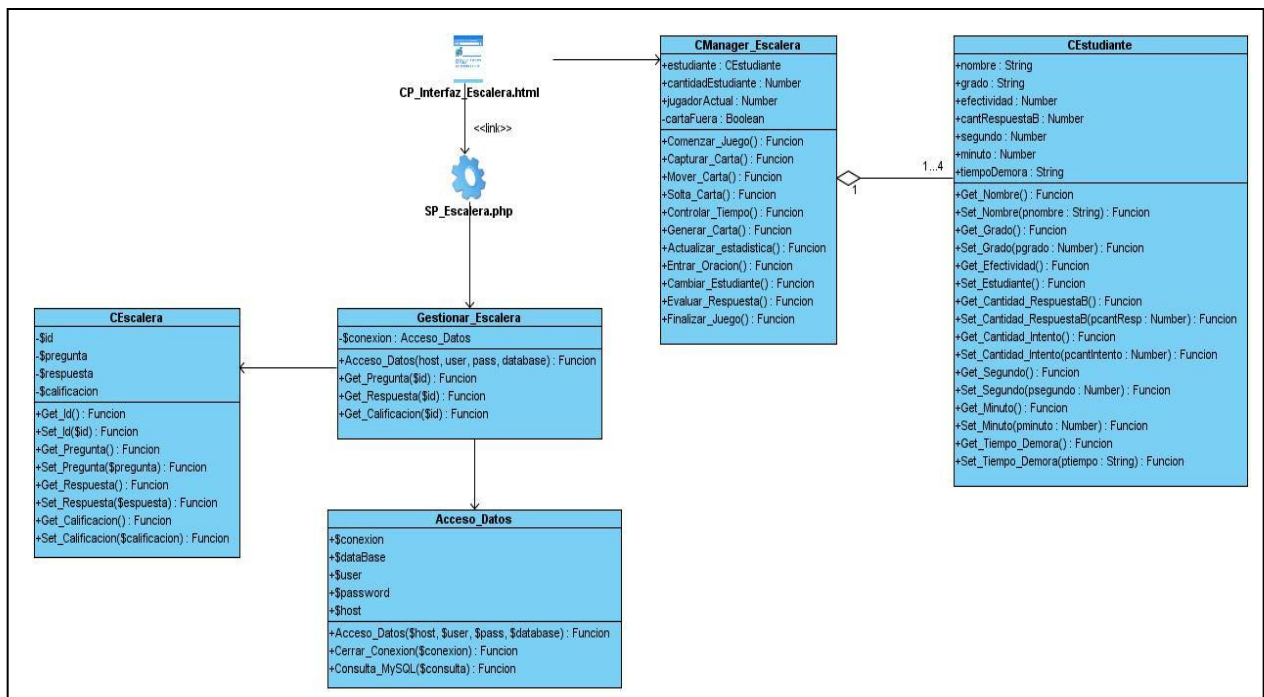


Figura 26. Diagrama de Clases del diseño del caso de uso “Controlar funcionalidades de Escalera de Palabras”



## 3.4 Diagrama de Clases Persistentes

Las clases persistentes necesitan ser capaz de guardar su estado en un medio permanente, lo cual está dado por el almacenamiento físico de la información de la clase, para la copia de seguridad en caso del fracaso del sistema, o para el intercambio de información.

A continuación se muestra el Diagrama de Clases Persistentes:

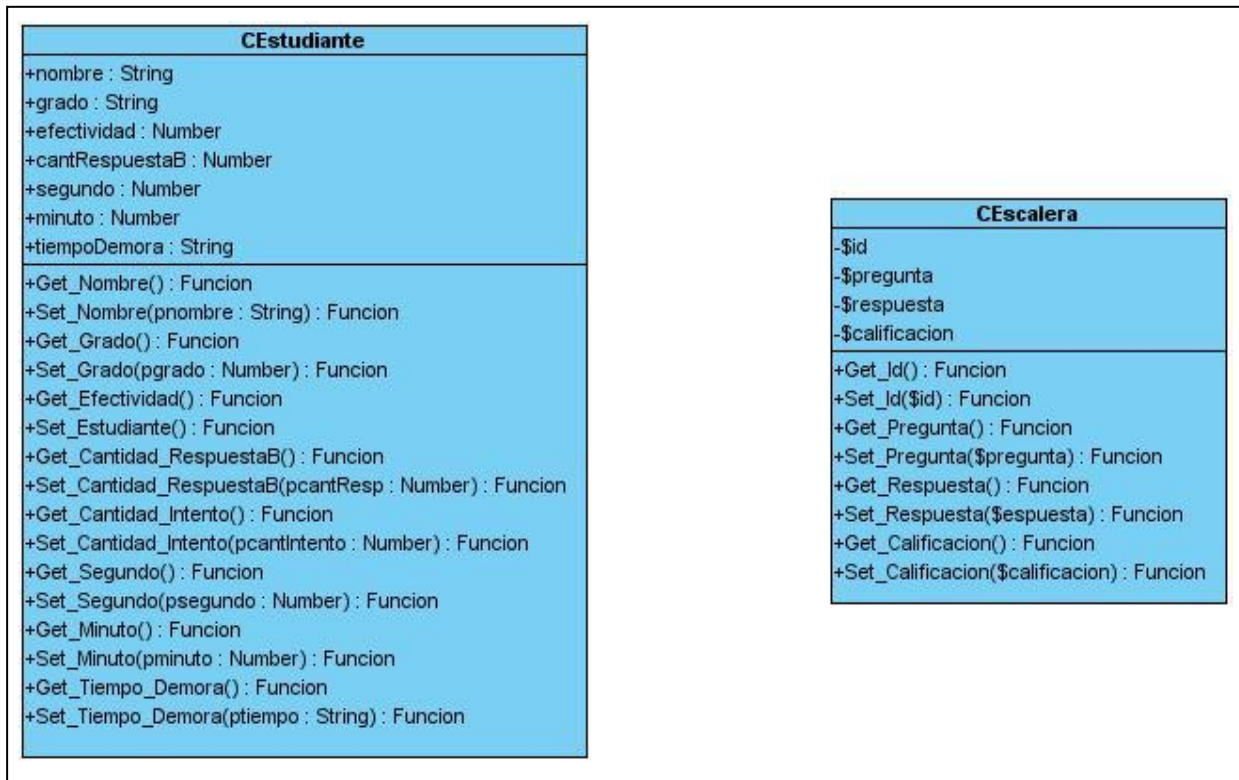


Figura 27. Diagrama de Clases Persistentes

## 3.5 Estándar de Codificación

Uno de los aspectos fundamentales que se debe tener en cuenta a la hora de implementar un software, es el uso de un estándar de codificación. Estos se definen con el objetivo de establecer un estilo de programación homogéneo que les permita a los programadores de un proyecto trabajar de forma coordinada.

Las principales ventajas de utilizar un estándar para escribir el código de las aplicaciones son:

- Reducción de los errores.
- Obtención de un código claro y comprensible.
- Garantizar una buena comunicación entre los programadores del proyecto.

## CAPÍTULO 3. Análisis y Diseño del Sistema

- Facilitar el mantenimiento del software.
- Rapidez a la hora de realizar cambios o nuevas versiones del software.

Con el objetivo de organizar y mejorar los juegos del primer ciclo de la Colección Multisaber, la dirección del proyecto Multisaber definió el siguiente estándar de codificación para ser usado durante la implementación de la nueva versión de estos juegos.

Elemento	Descripción	Ejemplo
Variables privadas	Las variables comenzarán con las letras iniciales del nombre del juego, seguidas del guión bajo, más el nombre de la variable que serán escritas con letra minúscula en el caso de ser una palabra simple, en el caso de que la variable esté compuesta por dos palabras, la primera palabra será escrita en minúscula y la segunda con letra inicial mayúscula.	<b>Variable del juego Rescatando al Turpial:</b> <code>var T_nombre</code> <b>Variable del juego Descubre la Imagen 1 :</b> <code>var Di1_nombreJuego</code>
Variables globales	Estas estarán compuestas primeramente por la palabra (g), seguida de la abreviatura perteneciente al nombre del juego, luego un guión bajo, más el nombre de la variable con letra inicial mayúscula.	<b>Variable global del Rompecabezas:</b> <code>var gRomp_Nombre</code> <code>var gRomp_Nombre_Juego</code> <b>Variable global de Rescatando al Turpial:</b> <code>var gTurp_Nombre</code> <code>var gTurp_Nombre_Juego</code>
Funciones	Estas estarán escritas primeramente con la letra inicial del juego en mayúscula unida con guión bajo con el nombre de la función con letra inicial mayúscula, en el caso de estar compuesta por dos palabras, deben estar unidas con guión bajo.	<b>Funciones que pertenecen al juego Rescatando al Turpial:</b> <code>function T_Tiempo()</code> <code>function T_Comenzar_Juego()</code> <b>Funciones que pertenecen al juego Descubre la Imagen 1:</b> <code>function Di1_Tiempo()</code>

		<code>function Di1_Comenzar_Juego()</code>
Clases	<p>Las clases tendrán la peculiaridad de tener la letra <b>C</b> en el comienzo del nombre, y su nombre variará en dependencia de su función:</p> <p>En el caso de ser Controladora la primera parte de su nombre estará compuesta por <b>CManager</b> más el nombre del juego.</p> <p>En el caso de las otras clases que conformarán los juegos, comenzarán primeramente por la letra <b>C</b>, más su nombre con letra inicial mayúscula y en el caso de estar compuesta por dos palabras deben estar unidas con guión bajo.</p>	<p><b>Clase Controladora:</b>  CManager_Turpial  CManager_Descubre_Imagen</p> <p><b>Ejemplos de otras clases utilizadas en los juegos:</b>  CEstudiante  CSala</p>
Constantes	Estas serán escritas en mayúscula y en el caso de estar compuesta por dos palabras deben estar unidas con guión bajo.	TIEMPO TIEMPO_ACTUAL
Comentarios	Estos serán usados para explicar el objetivo de determinado código.	// esto es un ejemplo de comentario  /* este es otro ejemplo */

Un ejemplo muy sencillo y claro del uso de este estándar de codificación, en la implementación de la nueva versión de los juegos del primer ciclo de la Colección Multisaber se puede ver en el **Anexo 2**.

### 3.6 Conclusiones

En este capítulo quedaron expuestos algunos elementos que describen cómo está construido el sistema partiendo de los Diagramas de Clases del Análisis y Diseño. También se presentó el Diagrama de Clases Persistentes que contiene la información persistente que se usará para la construcción del producto. Por último, se presentó el estándar de codificación que se usará en la nueva versión de los juegos del primer ciclo de la Colección Multisaber.



### 4.1 Introducción

En el presente capítulo se describe cómo los elementos del Modelo de Diseño son implementados en términos de componentes y además cómo éstos se organizan de acuerdo con los nodos específicos en el Modelo de Despliegue.

### 4.2 Modelo de Implementación

El Modelo de Implementación es el artefacto más significativo del flujo de trabajo de Implementación, éste tiene gran importancia porque mediante su uso los desarrolladores entienden claramente el funcionamiento del sistema antes de comenzar a escribir las líneas de código. Este modelo está conformado por los Diagrama de Componentes y de Despliegue. Además de que su objetivo fundamental es desarrollar la arquitectura y el sistema como un todo, es decir, describir cómo las clases del Modelo de Diseño se implementan en términos de componentes tales como ejecutables, ficheros de código fuentes o tablas de una base de datos.

#### 4.2.1 Diagramas de Componentes

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

A continuación se muestran los diferentes Diagramas de Componentes:

## 4.2.1.1 Diagrama de Componentes General

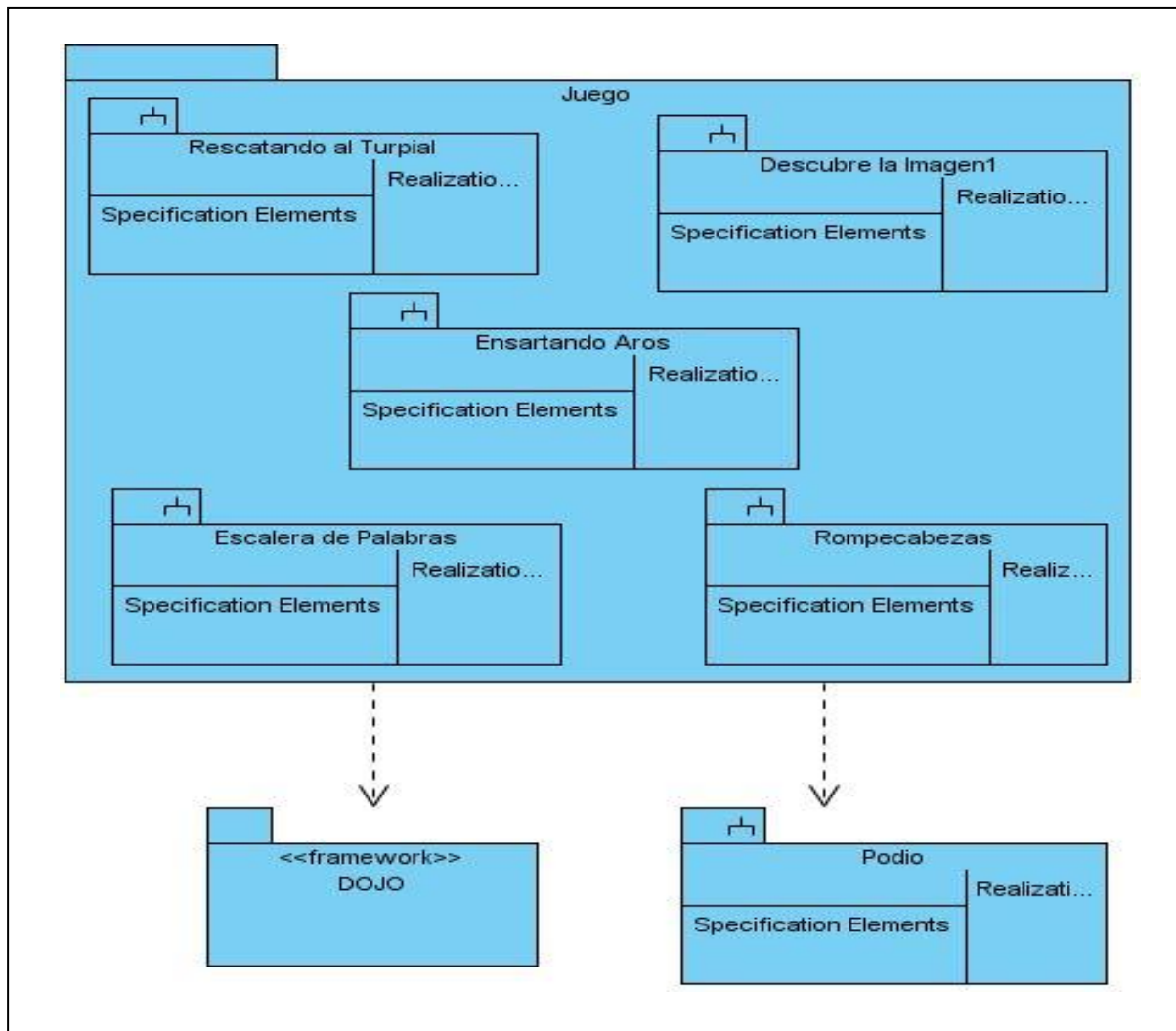


Figura 28. Diagrama de Componentes General

## 4.2.1.2 Diagrama de Componentes del Podio

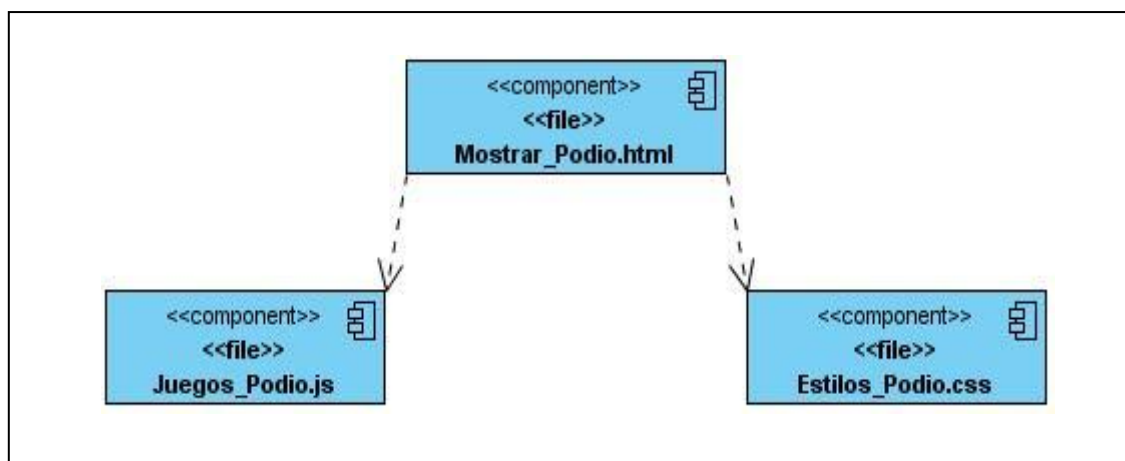


Figura 29. Diagrama de Componentes del subsistema Podio

### 4.2.1.3 Diagrama de Componentes Rescatando al Turpial

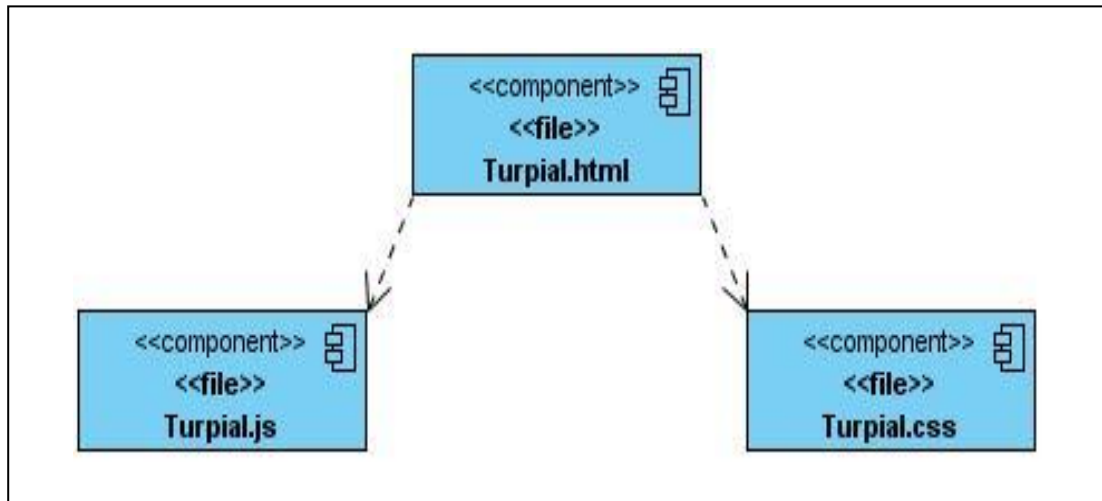


Figura 30. Diagrama de Componentes del subsistema Rescatando al Turpial

### 4.2.1.4 Diagrama de Componentes Descubre la Imagen1.

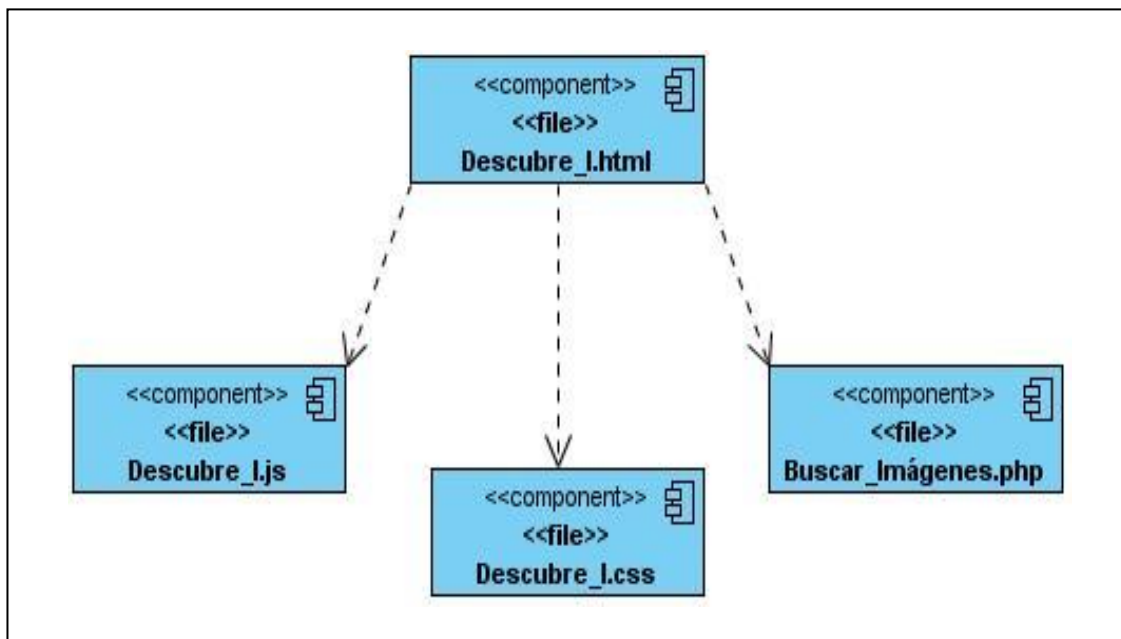


Figura 31. Diagrama de Componentes del subsistema Descubre la Imagen1

### 4.2.1.5 Diagrama de Componentes Ensartando Aros.

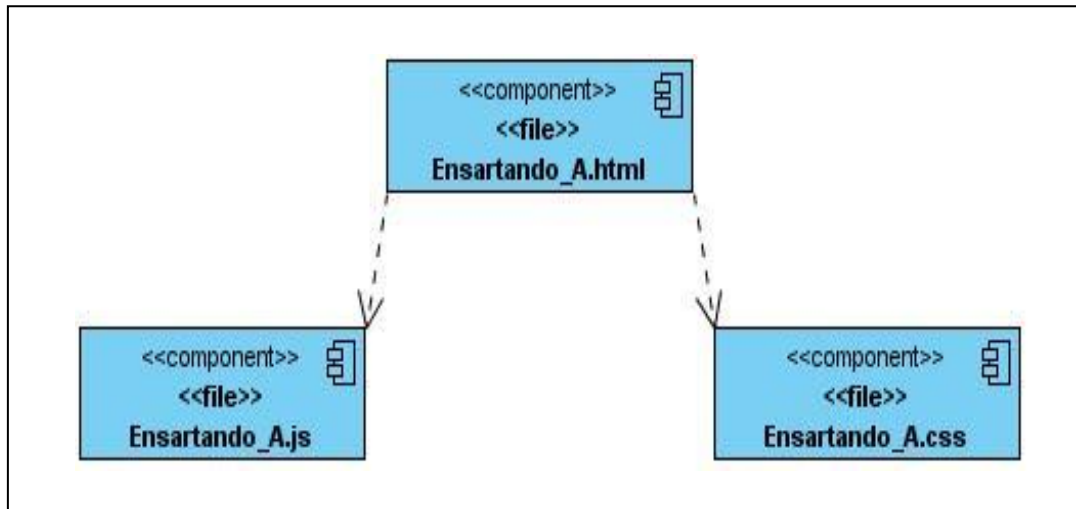


Figura 32. Diagrama de Componentes del subsistema Ensartando Aros

### 4.2.1.6 Diagrama de Componentes Rompecabezas.

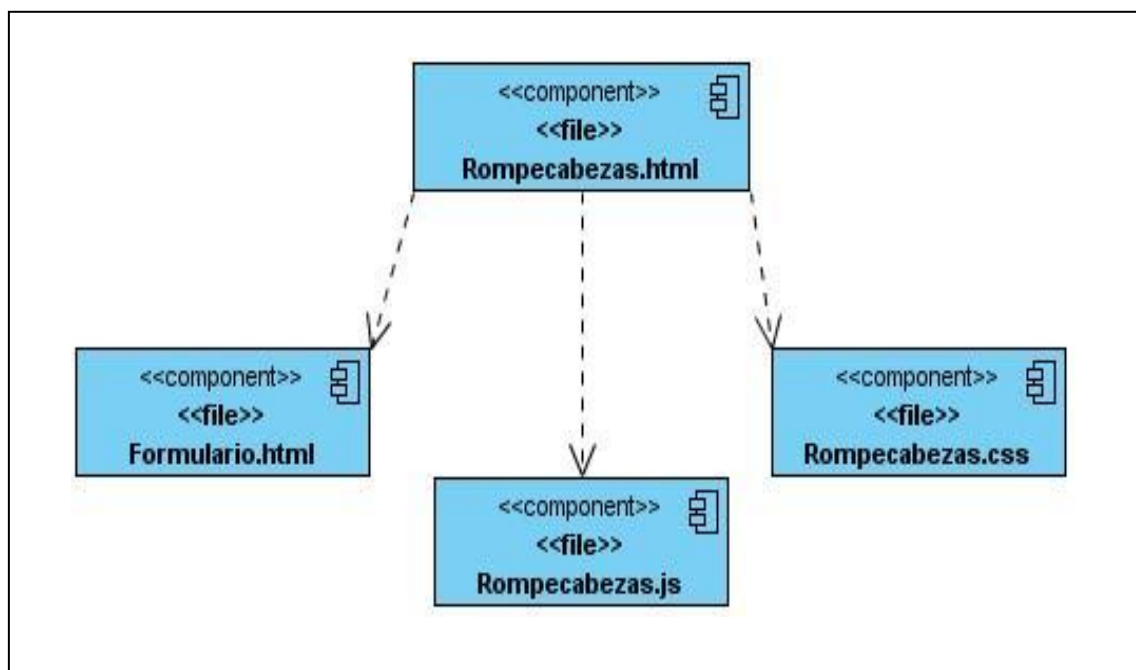


Figura 33. Diagrama de Componentes del subsistema Rompecabezas

## 4.2.1.7 Diagrama de Componentes Escalera de Palabras.

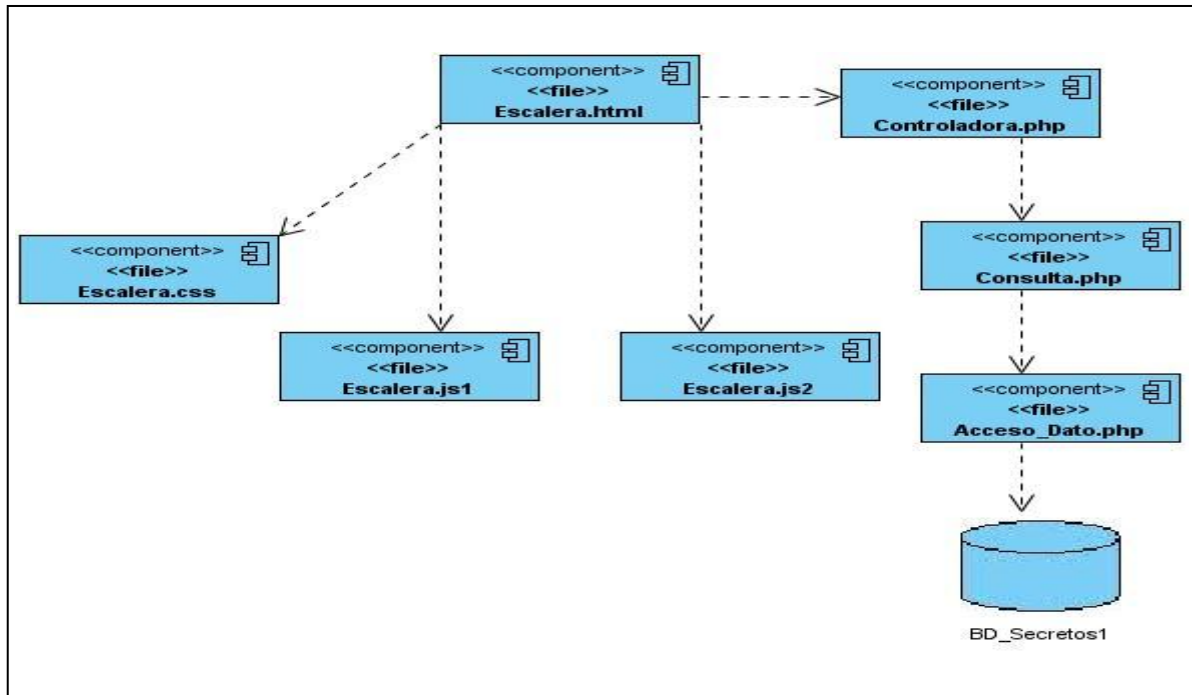


Figura 34. Diagrama de Componentes del subsistema Escalera de Palabras

## 4.2.1.8 Diagrama de Componentes de Gestión de Contenido.

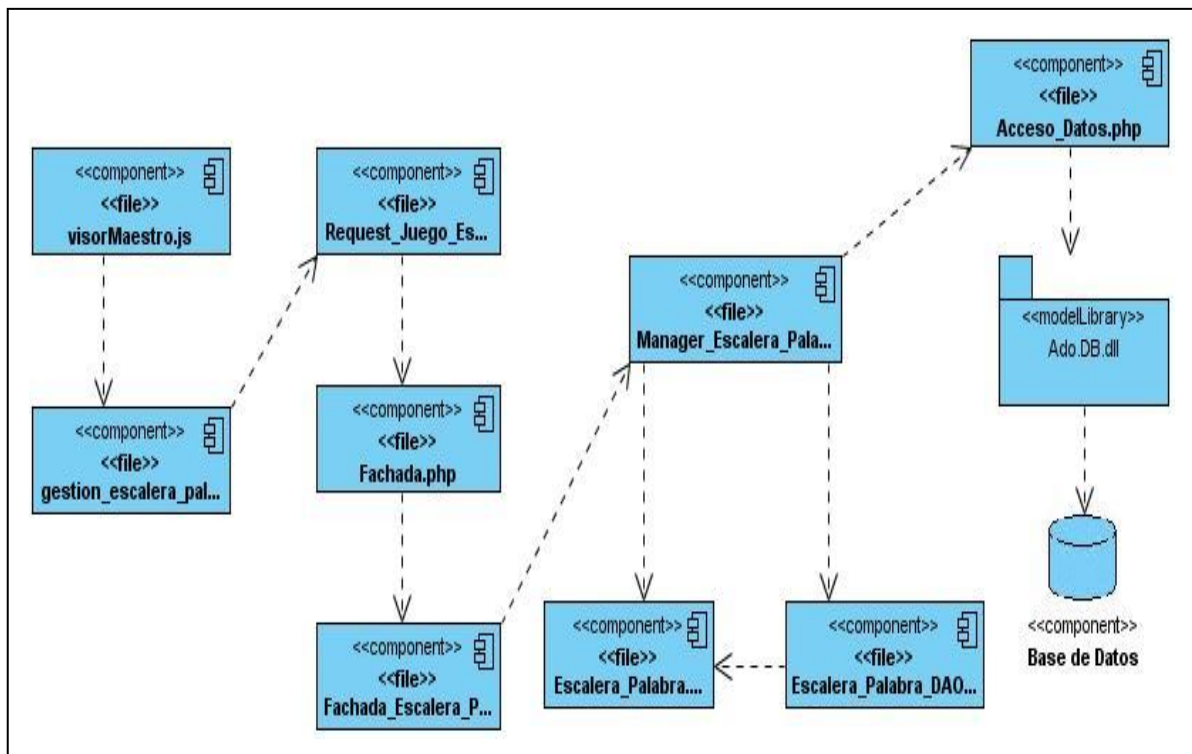


Figura 35. Diagrama de Componentes de Gestión de Contenido



### 4.2.2 Diagrama de Despliegue

Los Diagramas de Despliegue muestran las relaciones físicas entre los componentes del hardware y el software, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes del software. También un Diagrama de Despliegue es un grafo de nodos, unidos por conexiones de comunicación, donde un nodo puede contener instancias de componentes, un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

A continuación se muestran las dos formas en las que el producto puede estar distribuido:

1- En el primer caso permite instalarlo localmente en una PC, de esta forma, el software tiene que estar instalado en todas las PC de los alumnos.

2- En el segundo caso es necesario tener una PC como servidor central donde se encuentre el servidor web y el de bases de datos, y desde cualquier PC se puede hacer uso de la aplicación sin necesidad de tenerlo instalado.

#### 4.2.2.1 Modelo de Despliegue.

En este diagrama (figura 36) el nodo PC es donde se instala el producto, es decir, está instalado un servidor web, el de bases de datos y todo lo que necesita el software para su correcto funcionamiento. En el caso de la impresora es el único dispositivo que se utiliza, el sistema tiene incluido la funcionalidad de impresión de los temas y el glosario.

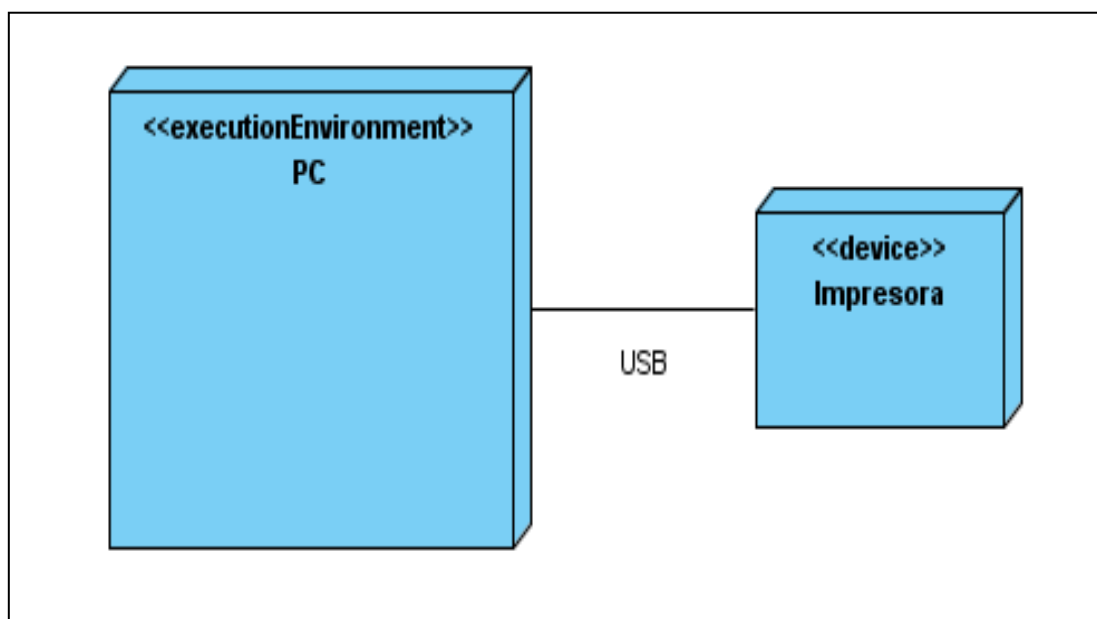


Figura 36. Modelo de Despliegue 1

## CAPÍTULO 4. Implementación del Sistema

En este diagrama (figura 37) la PC Cliente se conecta a través de un navegador al servidor central donde se encuentra el software, de esta forma no se necesita tener instalado el producto localmente. En el caso del servidor central en este se tienen instalados un servidor web y uno de bases de datos, además de los componentes necesarios para el funcionamiento del producto.

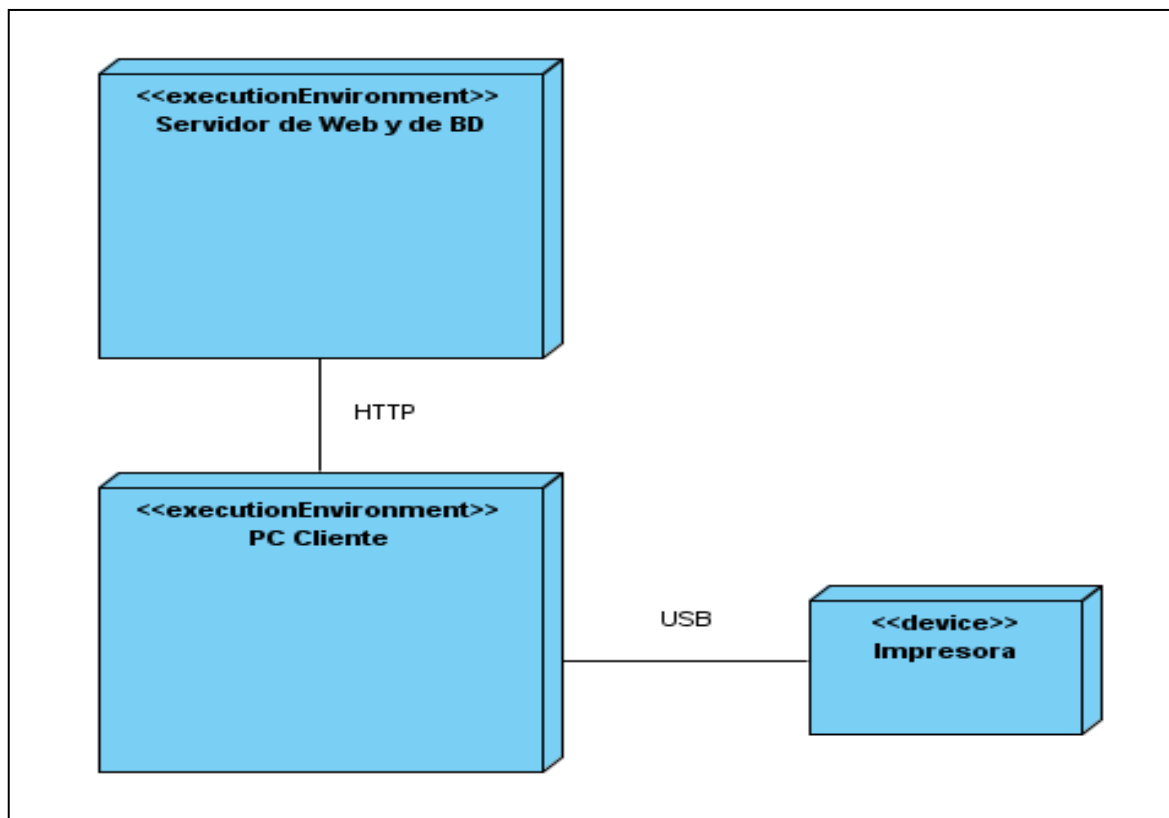


Figura 37. Modelo de Despliegue 2

### 4.3 Conclusiones

En este capítulo fue presentado como el sistema está construido a partir de los Diagramas de Componentes, tanto de todo el sistema en general como por cada uno de los paquetes de componentes. Además de mostrar el Diagrama de Despliegue que representa cuáles son los nodos que serán usados en el momento de realizar el despliegue de la aplicación.

## Conclusiones Generales

Después de realizar un análisis para el desarrollo de la aplicación y haber concluido con las fases de Concepción, Elaboración y Construcción del producto se determinó las siguientes conclusiones:

- Como resultado de esta investigación se logró la realización de una nueva versión de los juegos del primer ciclo de la Colección Multisaber siguiendo los criterios definidos por la metodología RUP y UML como notación gráfica para la representación de modelos.
- Se realizó el análisis, el diseño y la implementación de los juegos del primer ciclo de la Colección Multisaber, obteniendo toda la documentación necesaria para garantizar el futuro mantenimiento del sistema.
- Con la realización de esta aplicación el proyecto Multisaber de la facultad 8 de la UCI obtuvo un producto más dinámico, flexible y escalable; capaz de facilitar el desarrollo de diversas habilidades en los niños de la enseñanza primaria.

Todo lo planteado anteriormente justifica el cumplimiento de los objetivos propuestos y el desarrollo exitoso de la aplicación.

### Recomendaciones

Al terminar el siguiente trabajo se proponen las siguientes recomendaciones:

- Continuar con el proceso de mejoras a los juegos del primer ciclo de la Colección Multisaber.
- Realizar un estudio de los juegos del segundo ciclo de la Colección Multisaber con el propósito de incorporar nuevas funcionalidades que faciliten el proceso de enseñanza-aprendizaje en los niños de la enseñanza primaria.
- Dar a conocer el producto a otros centros de desarrollo de software educativo con el objetivo de aprovechar diversas funcionalidades ya implementadas.

## Referencias Bibliográficas

1. **Laborda, Rocio Martín.** Las nuevas tecnologías en la educación. *Cuadernos/Sociedad de la información*. España : Fundación AUNA, 2005.
2. **Pérez, José Ramón Gómez.** Las tecnologías en la educación. *Las tecnologías en la educación*. [En línea] 2 de 3 de 2007. [Citado el: 4 de 11 de 2009.] <http://boj.pntic.mec.es/jgomez46/ticedu.htm>.
3. **Kwan, Fancisco.** *Impacto de las nuevas tecnologías de la información y comunicación en la gestión del conocimiento*. Venezuela : s.n.
4. *La importancia del juego.* **Rodríguez., Esmeralda Jiménez.** 26, España : ISSN 1696-7208, 2006, Vol. III.
5. **Matanza, José Manuel Bouzán.** Ardora-Creación de actividades docentes. *Ardora-Creación de actividades docentes*. [En línea] [Citado el: 4 de 2 de 2010.] [http://webardora.net/index\\_cas.htm](http://webardora.net/index_cas.htm).
6. **Pressman, Roger S.** *Ingeniería de Software, Un enfoque práctico*. España : Darrerl Ince, 2001.
7. Metodología XP Vs. Metodología Rup . *Metodología XP Vs. Metodología Rup* . [En línea] 4 de 2008. [Citado el: 4 de 2 de 2010.] <http://metodologiavaxpvsmetodologiarup.blogspot.com/>.
8. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. España : Addison Wesley.
9. **Marzo, Josep Vilalta.** *Criterios de selección de una herramienta CASE - UML*. 2004.
10. **Sparxsystems.** Sparxsystems. *Sparxsystems*. [En línea] [Citado el: 5 de 2 de 2010.] <http://www.sparxsystems.com.ar/products/ea.html>.
11. *LESE-2 Introducción a Rational Rose.* **González Blanco, Rubén y Pérez Tobalina, Sergio.** Barcelona : s.n.
12. Visual Paradigm. *Visual Paradigm*. [En línea] [Citado el: 5 de 2 de 2010.] <http://www.visual-paradigm.com/product/vpuml/editions/modeler.jsp>.

## REFERENCIAS BIBLIOGRÁFICAS

---

13. [www.mitecnologico.com. www.mitecnologico.com.](http://www.mitecnologico.com/Main/DefinicionDeLenguajeDeProgramacion) [En línea] [Citado el: 6 de 2 de 2010.] <http://www.mitecnologico.com/Main/DefinicionDeLenguajeDeProgramacion>.
14. [www.php.net. www.php.net.](http://docs.php.net/manual/es/intro-whatism.php) [En línea] [Citado el: 6 de 2 de 2010.] <http://docs.php.net/manual/es/intro-whatism.php>.
15. [www.creargratisunapaginaweb.com. www.creargratisunapaginaweb.com.](http://www.creargratisunapaginaweb.com/PHP/Ventajas-y-desventajas-del-Personal-Home-Page-4/) [En línea] [Citado el: 7 de 2 de 2010.] <http://www.creargratisunapaginaweb.com/PHP/Ventajas-y-desventajas-del-Personal-Home-Page-4/>.
16. [www.uazuay.edu.ec. www.uazuay.edu.ec.](http://www.uazuay.edu.ec/estudios/sistemas/lenguaje_iii/MAnualJavaScript) [En línea] [Citado el: 8 de 2 de 2010.] [http://www.uazuay.edu.ec/estudios/sistemas/lenguaje\\_iii/MAnualJavaScript](http://www.uazuay.edu.ec/estudios/sistemas/lenguaje_iii/MAnualJavaScript).
17. [www.lcc.uma.es. www.lcc.uma.es.](http://www.lcc.uma.es/~eat/services/html-js/manual14.html) [En línea] [Citado el: 9 de 2 de 2010.] <http://www.lcc.uma.es/~eat/services/html-js/manual14.html>.
18. [www.cibernetia.com. www.cibernetia.com.](http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php) [En línea] [Citado el: 26 de 1 de 2010.] [http://www.cibernetia.com/manuales/instalacion\\_servidor\\_web/1\\_conceptos\\_basicos.php](http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php).
19. [www.linux.ciberaula.com. www.linux.ciberaula.com.](http://www.linux.ciberaula.com/articulo/linux_apache_intro/) [En línea] [Citado el: 20 de 1 de 2010.] [http://www.linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://www.linux.ciberaula.com/articulo/linux_apache_intro/).
20. [www.error500.net. www.error500.net.](http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php) [En línea] [Citado el: 10 de 2 de 2010.] [http://www.error500.net/garbagecollector/archives/categorias/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_datos\\_sgbd.php](http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php).
21. [www.e-socrates.org. www.e-socrates.org.](http://www.e-socrates.org/mod/resource/view.php?id=6548) [En línea] 4 de 6 de 2008. [Citado el: 10 de 2 de 2010.] <http://www.e-socrates.org/mod/resource/view.php?id=6548>.
22. **Velasco, Roberto Hernando.** *El SGBDR Oracle* .
23. **Toledo Alma, Enríquez, y otros.** [www.uaem.mx. www.uaem.mx.](http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf) [En línea] [Citado el: 11 de 2 de 2010.] <http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf>.
24. [www.petra.euitio.uniovi.es. www.petra.euitio.uniovi.es.](http://www.petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf) [En línea] [Citado el: 12 de 2 de 2010.] <http://www.petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.

## REFERENCIAS BIBLIOGRÁFICAS

---

**25. Bermejo Sanz, Laura y Gómez Monreal, Enrique.** [www.kybele.escet.urjc.es](http://www.kybele.escet.urjc.es). [www.kybele.escet.urjc.es](http://www.kybele.escet.urjc.es). [En línea] [Citado el: 12 de 2 de 2010.] <http://kybele.escet.urjc.es/documentos/HC/Exposiciones/EclipseIDE.pdf>.

**26. bitacora, Punto.** [www.puntobitacora.com](http://www.puntobitacora.com). [www.puntobitacora.com](http://www.puntobitacora.com). [En línea] [Citado el: 12 de 2 de 2010.] <http://www.puntobitacora.com/post-8-aptana-studio-2-0-un-ide-basado-en-eclipse-para-php-y-otras-yerbas>.

**27. Corporation, Zend.** [www.cvteam.es](http://www.cvteam.es). [www.cvteam.es](http://www.cvteam.es). [En línea] [Citado el: 12 de 2 de 2010.] [http://www.cvteam.es/comunes/ficheros/fileDownload.php/Zend\\_Studio.pdf](http://www.cvteam.es/comunes/ficheros/fileDownload.php/Zend_Studio.pdf).

**28. Microsystems, Sun.** [www.sun.com](http://www.sun.com). [www.sun.com](http://www.sun.com). [En línea] [Citado el: 12 de 2 de 2010.] [http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam\\_feature.html](http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam_feature.html).

## Bibliografía

1. **Laborda, Rocio Martín.** Las nuevas tecnologías en la educación. *Cuadernos/Sociedad de la información*. España : Fundación AUNA, 2005.
2. **Pérez, José Ramón Gómez.** Las tecnologías en la educación. *Las tecnologías en la educación*. [En línea] 2 de 3 de 2007. [Citado el: 4 de 11 de 2009.] <http://boj.pntic.mec.es/jgomez46/ticedu.htm>.
3. **Kwan, Fancisco.** *Impacto de las nuevas tecnologías de la información y comunicación en la gestión del conocimiento*. Venezuela : s.n.
4. *La importancia del juego.* **Rodríguez., Esmeralda Jiménez.** 26, España : ISSN 1696-7208, 2006, Vol. III.
5. **Pressman, Roger S.** *Ingeniería de Software, Un enfoque práctico*. España : Darrerl Ince, 2001.
6. Metodología XP Vs. Metodología Rup . *Metodología XP Vs. Metodología Rup* . [En línea] 4 de 2008. [Citado el: 4 de 2 de 2010.] <http://metodologiapxvsmetodologiarup.blogspot.com/>.
7. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. España : Addison Wesley.
8. **Marzo, Josep Vilalta.** *Criterios de selección de una herramienta CASE - UML*. 2004.
9. *LESE-2 Introducción a Rational Rose.* **González Blanco, Rubén y Pérez Tobalina, Sergio.** Barcelona : s.n.
10. Visual Paradigm. *Visual Paradigm*. [En línea] [Citado el: 5 de 2 de 2010.] <http://www.visual-paradigm.com/product/vpuml/editions/modeler.jsp>.
11. [www.php.net](http://www.php.net). *www.php.net*. [En línea] [Citado el: 6 de 2 de 2010.] <http://docs.php.net/manual/es/intro-what-is.php>.
12. **Velasco, Roberto Hernando.** *El SGBDR Oracle*.
13. **Toledo Alma, Enríquez, y otros.** [www.uaem.mx](http://www.uaem.mx). *www.uaem.mx*. [En línea] [Citado el: 11 de 2 de 2010.] <http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf>.



14. [www.creargratisunapaginaweb.com](http://www.creargratisunapaginaweb.com). *www.creargratisunapaginaweb.com*. [En línea] [Citado el: 7 de 2 de 2010.] <http://www.creargratisunapaginaweb.com/PHP/Ventajas-y-desventajas-del-Personal-Home-Page-4/>.
15. [www.uazuay.edu.ec](http://www.uazuay.edu.ec). *www.uazuay.edu.ec*. [En línea] [Citado el: 8 de 2 de 2010.] [http://www.uazuay.edu.ec/estudios/sistemas/lenguaje\\_iii/MANualJavaScript](http://www.uazuay.edu.ec/estudios/sistemas/lenguaje_iii/MANualJavaScript).
16. [www.lcc.uma.es](http://www.lcc.uma.es). *www.lcc.uma.es*. [En línea] [Citado el: 9 de 2 de 2010.] <http://www.lcc.uma.es/~eat/services/html-js/manual14.html>.
17. [www.cibernetia.com](http://www.cibernetia.com). *www.cibernetia.com*. [En línea] [Citado el: 26 de 1 de 2010.] [http://www.cibernetia.com/manuales/instalacion\\_servidor\\_web/1\\_conceptos\\_basicos.php](http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php).
18. [www.linux.ciberaula.com](http://www.linux.ciberaula.com). *www.linux.ciberaula.com*. [En línea] [Citado el: 20 de 1 de 2010.] [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/).
19. [www.error500.net](http://www.error500.net). *www.error500.net*. [En línea] [Citado el: 10 de 2 de 2010.] [http://www.error500.net/garbagecollector/archives/categorias/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_datos\\_sgbd.php](http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php).
20. [www.e-socrates.org](http://www.e-socrates.org). *www.e-socrates.org*. [En línea] 4 de 6 de 2008. [Citado el: 10 de 2 de 2010.] <http://www.e-socrates.org/mod/resource/view.php?id=6548>.
21. **Bermejo Sanz, Laura y Gómez Monreal, Enrique**. [www.kybele.escet.urjc.es](http://www.kybele.escet.urjc.es). *www.kybele.escet.urjc.es*. [En línea] [Citado el: 12 de 2 de 2010.] <http://kybele.escet.urjc.es/documentos/HC/Exposiciones/EclipseIDE.pdf>.
22. **bitacora, Punto**. [www.puntobitacora.com](http://www.puntobitacora.com). *www.puntobitacora.com*. [En línea] [Citado el: 12 de 2 de 2010.] <http://www.puntobitacora.com/post-8-aptana-studio-2-0-un-ide-basado-en-eclipse-para-php-y-otras-yerbas>.
23. **Corporation, Zend**. [www.cvteam.es](http://www.cvteam.es). *www.cvteam.es*. [En línea] [Citado el: 12 de 2 de 2010.] [http://www.cvteam.es/comunes/ficheros/fileDownload.php/Zend\\_Studio.pdf](http://www.cvteam.es/comunes/ficheros/fileDownload.php/Zend_Studio.pdf).
24. **Microsystems, Sun**. [www.sun.com](http://www.sun.com). *www.sun.com*. [En línea] [Citado el: 12 de 2 de 2010.] [http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam\\_feature.html](http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam_feature.html).
25. **Cabero Almenara, Julio**. *Impacto de las nuevas tecnologías de la información y la comunicación en las organizaciones*. Sevilla : Grupo Editorial Universitario, 1998. 197-206.

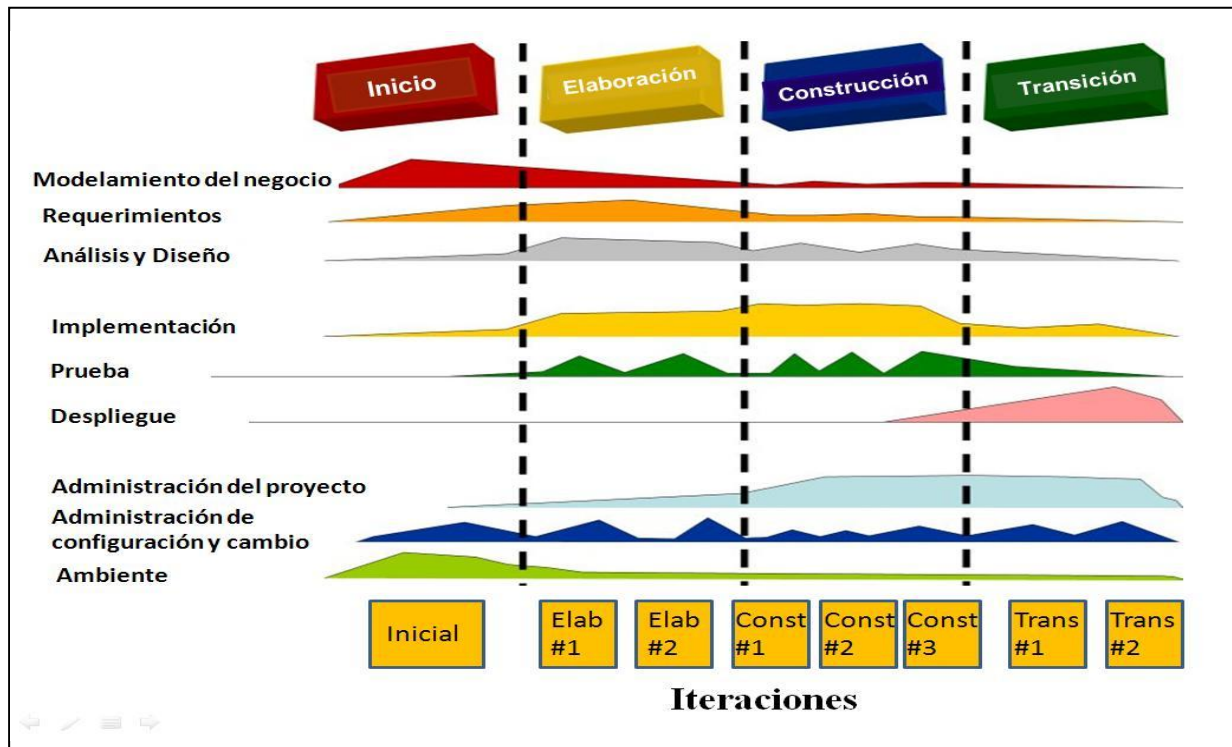
26. *Las posibilidades educativas de los videojuegos*. **Pindado, Julian**. Sevilla : Revista de Medios y Educación, 2005.
27. *AMBIENTES DE ENSEÑANZA-APRENDIZAJE ENRIQUECIDOS CON COMPUTADOR*. **Panqueva, Alvaro H. Galvis**. 2, 1998, Vol. 1.
28. *www.cristalab.com*. *www.cristalab.com*. [En línea] 9 de 10 de 2007. [Citado el: 15 de 3 de 2010.] <http://www.cristalab.com/tutoriales/programacion-orientada-a-objetos-oop-con-javascript-c232/>.
29. **Alvarez, Miguel Angel**. *www.desarrolloweb.com*. *www.desarrolloweb.com*. [En línea] 21 de 3 de 2002. [Citado el: 15 de 3 de 2010.] <http://www.desarrolloweb.com/articulos/725.php>.
30. **Gonzalez, Beatriz**. *www.ajaxhispano.com*. *www.ajaxhispano.com*. [En línea] 28 de 5 de 2006. [Citado el: 16 de 3 de 2010.] <http://www.ajaxhispano.com/curso-manual-tutorial-javascript-clases-prototype-json-programacion-orientada-objetos-ajax-espanol-castellano.html>.

## Glosario de Términos

- **Base de Datos:** Es una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los Sistemas de Información de una empresa o negocio en particular.
- **CU:** Representa la abreviatura de Caso de Uso.
- **CUS:** Representa la abreviatura de Caso de Uso del Sistema.
- **CASE:** Computer Aided Software Engineering. Ingeniería de Software Asistida por Ordenador.
- **HTML (Lenguaje de Marcado de Hipertexto):** Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.
- **HTTP (Protocolo de Transferencia de Hipertextos):** Modo de comunicación para solicitar páginas web.
- **Herramientas CASE:** Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.
- **Hardware:** Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.
- **Iteración:** Es la repetición de una serie de instrucciones dentro de cierta fase de desarrollo del software.
- **JSON:** Representa un tipo de archivo que almacena información y son muy utilizado como archivos de configuración.
- **Log:** Archivo que registra movimientos y actividades de un determinado programa. En un servidor web, se encarga de guardar todos los requerimientos y servicios entregados desde él, por lo que es la base del software de estadísticas de visitas.
- **SGBD (Sistema de Gestión de Bases de Datos):** Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.
- **Software:** Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.
- **UML (Lenguaje Unificado de Modelado):** Lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

## Anexos

## Anexo 1: Ciclo de vida de RUP.



## Anexo 2: Ejemplo que muestra el uso del estándar de codificación definido en el proyecto.

```

function CManager_DescubreI1()
{
    this.Di1_jugadores = new Array();
    this.Di1_contador=0;
    this.Di1_cantJugadores = 0;
    this.Di1_posicionJugador=0;
    this.Di1_tiempo=-1;
    this.Di1_posiciones= new Array(91,397,227,400,333,397,435,403,497,378,595,383,630,309);
    this.Di1_verificarFin=false;
    this.Di1_verificarClick=true;
    this.Di1_puntos=0;
    this.Di1_cantImágenes=16; /*cantidad de imágenes que se iran retirando*/
    this.Di1_imágenes=new Array(this.Di1_cantImágenes);
    this.Di1_imágenesFondoADescubrir = 7; /*en estos momentos solo hay 7 imágenes para des
    this.Di1_jugadorActual=0;
    this.Di1_bandera = false;
    this.Di1_numero = 0;
    this.Di1_auto = 0;
    this.Di1_fin = 0;
    this.Di1_temporal = 0;
    var Di1_http = Di1_getXMLHttpRequest();

    this.Di1_Agregar_Jugador = function(elemento){
        this.Di1_jugadores.push(elemento);
    }

    this.Di1_Listar = function(){
        return this.Di1_jugadores;
    }
}

```