

Universidad de las Ciencias Informáticas  
Facultad 4



Desarrollo de los juegos Rompecabezas, Puzle,  
Tangram y Tetravex para la colección de juegos  
educativos MundoClick

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

Autores

Arianni Pérez Mendoza

Dannier Salas Garcé

Tutores

Ing. Nersa Doraines Acosta Labrada

Ing. Arisbel Hechavarría Rojas



“El aspecto fundamental en el cual la juventud debe señalar el camino es precisamente en el aspecto de ser vanguardia en cada uno de los trabajos que le compete.”

Ernesto Guevara de la Serna

### **Declaración de autoría**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2014.

---

Arianni Pérez Mendoza  
Firma del Autor

---

Dannier Salas Garcé  
Firma del Autor

---

Ing. Nersa Doraines Acosta Labrada  
Firma del Tutor

---

Ing. Arisbel Hechavarría Rojas  
Firma del Tutor

## **Dedicatoria**

De Arianni Pérez Mendoza:

A mis padres Erminia y Eduardo, por su apoyo y comprensión todos estos años. Todos y cada uno de mis logros como estudiante se los debo a ustedes.

A mi hermana, por ser ese rayito de felicidad que nos ilumina a todos con su amor y cariño.

De Dannier Salas Garcé:

A Dios, por todo lo que ha hecho en mi vida.

A mis padres, por su dedicación, apoyo y ejemplo.

A mis hermanos Dilver y Daniel.

## **Agradecimientos**

De Arianni Pérez Mendoza:

A mis padres Eduardo y Erminia, por depositar toda su confianza en mí.

A mi hermana Lismary, por darle tanta alegría a mi vida.

A mi novio, por su ayuda incondicional y por soportarme tanto tiempo.

A mis tutores, por su apoyo.

A todas mis amistades, por darme ánimo para continuar y ser capaces de sacar una sonrisa en los malos momentos.

De Dannier Salas Garcé:

A Dios, por permitirme llegar hasta aquí y darme la fuerza para seguir adelante.

A mi familia, por apoyarme siempre.

A mis hermanos, por el ánimo que me transmitieron.

A todos mis compañeros de universidad, los de mi grupo y del apartamento, en especial a Oscar, Abel, Yunior, Sandy y Ariel.

A mis tutores y a los profesores del proyecto MundoClick por su ayuda.

A mi compañera de tesis, por su comprensión y paciencia.

A todas las personas que de una u otra forma aportaron algo para que fuera posible la realización de esta investigación.

## Resumen

Actualmente el Centro de Tecnologías para la Formación (FORTES) de la Universidad de las Ciencias Informáticas se encuentra inmerso en la construcción de una colección de juegos educativos denominada MundoClick. En el presente documento se abordan los elementos esenciales de los flujos de trabajo: requisitos, diseño, implementación y pruebas correspondientes a los juegos educativos Rompecabezas, Puzle, Tangram y Tetravex para la colección MundoClick. Además se realiza un estudio del estado del arte de la propuesta de solución y se utiliza el conjunto de herramientas y tecnologías definidas por el proyecto MundoClick.

La investigación fue guiada por la metodología tradicional RUP y arrojó como principales resultados la documentación y los artefactos de ingeniería de software generados durante el desarrollo del sistema, así como la realización de cuatro juegos educativos que fomentan habilidades espaciales y de pensamiento lógico-matemático.

**Palabras claves:** juegos educativos, MundoClick.

## Índice general

Introducción .....	1
Capítulo 1: Fundamentación teórica .....	5
Introducción .....	5
1.1 Software educativo .....	5
1.1.1 Clasificación de software educativo .....	5
1.2 Juegos educativos de ordenador .....	6
1.2.1 Tipos de juegos de ordenador .....	8
1.3 Habilidades espaciales y de pensamiento lógico .....	8
1.4 Análisis de soluciones similares .....	11
1.4.1 Colección de software educativo GCompris .....	11
1.4.2 Puzle–Números .....	12
1.4.3 GNOME Tetravex .....	12
1.4.4 Colección de software educativo La caja mágica .....	12
1.5 Plataforma Pointer .....	13
1.6 Metodología de desarrollo .....	14
1.6.1 Metodología RUP .....	14
1.7 Lenguaje Unificado de Modelado (UML) .....	14
1.8 Herramienta CASE de modelado con UML .....	14
1.8.1 Visual Paradigm v8.0 .....	14
1.9 Lenguajes de desarrollo .....	15
1.9.1 HTML5 .....	15
1.9.2 CSS3 .....	15
1.9.3 JavaScript .....	16

1.10 Framework de desarrollo .....	16
1.10.1 Bootstrap v2.3.2 .....	17
1.10.2 JQuery v1.10.1 .....	17
1.10.3 NodeJS v0.8.11 .....	17
1.10.4 AppJS v0.0.20 .....	18
1.10.5 CreateJS .....	18
1.11 Entorno de desarrollo integrado .....	18
1.11.1 Netbeans v7.3 .....	19
1.12 Herramienta de control de versiones (Subversion).....	19
Conclusiones del capítulo .....	19
Capítulo 2: Características del sistema .....	20
Introducción .....	20
2.1 Descripción general de la propuesta del sistema .....	20
2.2 Especificación de los requisitos de software .....	21
2.2.1 Requisitos funcionales.....	22
2.2.2 Requisitos no funcionales.....	25
2.3 Modelo conceptual.....	26
2.4 Modelo de casos de uso del sistema .....	31
Conclusiones del capítulo .....	51
Capítulo 3: Diseño del sistema .....	52
Introducción .....	52
3.1 Patrón arquitectónico .....	52
3.1.1 Modelo Vista Controlador (MVC) .....	52
3.2 Modelo de diseño.....	53
3.2.1 Diagramas de clases del diseño .....	53

3.2.2 Diagramas de interacción .....	62
3.3 Patrones GRASP .....	65
3.4 Estilos de codificación.....	65
Conclusiones .....	67
Capítulo 4: Implementación y pruebas .....	68
Introducción .....	68
4.1 Modelo de implementación .....	68
4.1.1 Diagrama de componentes.....	68
4.2 Prueba .....	71
4.2.1 Niveles de prueba.....	71
4.2.2 Tipos de pruebas.....	72
4.2.3 Métodos de pruebas.....	72
4.2.4 Casos de prueba .....	72
4.2.5 Resultados de las pruebas .....	74
Conclusiones .....	75
Conclusiones generales .....	76
Recomendaciones.....	77
Referencias .....	78
Glosario de términos .....	83
Anexos .....	84

## Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) desde su surgimiento en los años ochenta provocaron cambios en los sectores económicos, sociales y culturales hasta el punto que en la actualidad no se concibe ningún proceso alejado de las mismas. La educación es una rama que está estrechamente vinculada a los adelantos tecnológicos. Las oportunidades que abren en el ámbito formativo posibilitan la búsqueda de nuevas soluciones que apoyen la calidad del proceso de enseñanza-aprendizaje.

De esta manera, uno de los medios que más se aprovecha por las instituciones escolares en el mundo como parte de las nuevas tecnologías es el software educativo. La utilización de este tipo de programa desarrolla la imaginación, la creatividad, la memoria y facilitan la transmisión de conocimientos de una forma más amena, integradora, diferenciada y activa que el resto de los medios de enseñanza, constituyendo un efectivo instrumento para el crecimiento educacional del hombre. (1)

El gobierno cubano enfrenta el desafío de utilizar las TIC para elevar la calidad de vida de los ciudadanos, priorizando la formación de los mismos a todos los niveles. Una alternativa clave para la ejecución de esta tarea lo constituye la Universidad de las Ciencias Informáticas (UCI), centro productor de tecnologías de información, que tiene entre sus misiones el desarrollo de productos de software educativo tanto para el entorno nacional como para el exterior. (2)

Con el objetivo de organizar la labor productiva que efectúa la UCI se dividió el trabajo en varios proyectos, quedando como encargado de realizar productos educativos el Centro de Tecnologías para la Formación (FORTES). En este sentido, el centro se encuentra inmerso en la construcción de una colección de juegos educativos denominada MundoClick, la cual está dirigida a niños de 6 a 12 años de edad y enfocada a potenciar en ellos diversas habilidades cognitivas que mejoren su desempeño en las asignaturas que reciben.

Hasta el momento, de los veintisiete juegos que deben integrarse a la colección, el equipo de trabajo que se ocupa de la realización del proyecto ha logrado incorporar diecinueve. Entre los que aparecen: el Elevador, el Parchís, Buscando Parejas, Escalera de Palabras, Memoria de Trayectorias y Siguiendo la Pista, los cuales motivan el estudio de los alumnos y fortalecen los contenidos de variadas temáticas mediante una serie de habilidades referidas a la agilidad mental, el cálculo, la memoria, la asociación de conceptos y el uso del teclado y mouse.

A pesar de contar con una parte importante del sistema propuesto es necesario que la aplicación logre potenciar en los niños el mayor número de destrezas posibles. En este aspecto, se deben considerar las habilidades espaciales y de pensamiento lógico-matemático pues estimulan en el estudiante su comprensión de los números, de los conceptos espaciales y del pensamiento matemático en general. Según lo planteado, MundoClick no dispone de una solución destinada especialmente a desarrollar las habilidades mencionadas anteriormente en los usuarios que interactúan con ella.

La situación descrita deriva en el siguiente **problema de investigación**: ¿Cómo contribuir al desarrollo de habilidades espaciales y de pensamiento lógico-matemático en los niños de la enseñanza primaria desde la colección de juegos educativos MundoClick?

Para dar respuesta al problema planteado, se define como **objetivo general** desarrollar los juegos Rompecabezas, Puzle, Tangram y Tetravex para contribuir al desarrollo de habilidades espaciales y de pensamiento lógico-matemático en los niños de la enseñanza primaria desde la colección de juegos educativos MundoClick.

El **objeto de estudio** se centra en el desarrollo de juegos educativos para niños y el **campo de acción** se enfoca en el desarrollo de juegos educativos que contribuyan a desarrollar las habilidades espaciales y de pensamiento lógico-matemático en los niños de la enseñanza primaria.

Del objetivo general se desprenden los objetivos específicos siguientes:

- Elaborar la fundamentación teórica relacionada con el desarrollo de juegos educativos para niños de la enseñanza primaria.
- Realizar el diseño de los juegos educativos propuestos para la colección MundoClick.
- Implementar los juegos educativos propuestos para la colección MundoClick.
- Probar los juegos desarrollados para garantizar su correcto funcionamiento.

En correspondencia con los objetivos elaborados se plantean las tareas de investigación:

- Valoración de los conceptos, tecnologías, metodologías y herramientas relacionadas con el desarrollo de juegos educativos para niños.
- Análisis de soluciones similares a los juegos educativos de la propuesta de solución.

- Definición de los requisitos funcionales y no funcionales de acuerdo con las características de la colección MundoClick.
- Realización del diseño de los juegos Rompecabezas, Puzle, Tangram y Tetravex.
- Implementación de los juegos Rompecabezas, Puzle, Tangram y Tetravex.
- Realización de pruebas a los juegos educativos.

## **Métodos científicos de la investigación**

El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. Se clasifican en teóricos y empíricos, los cuales están dialécticamente relacionados. (3) En el desarrollo de la presente investigación se hace uso de los siguientes métodos científicos:

### **Métodos Teóricos**

**Analítico-Sintético:** El uso de este método posibilitó que a través de la bibliografía consultada se definieran los conceptos más importantes relacionados con el desarrollo de los juegos educativos así como el análisis de las herramientas y tecnologías seleccionadas por el proyecto MundoClick para luego identificar la propuesta de solución adecuada al problema planteado.

**Histórico-Lógico:** Este método fue muy útil en la elaboración de la fundamentación teórica de la investigación ya que mediante su uso se realizó un análisis de la evolución de soluciones similares, enfocándose en las habilidades desarrolladas por los juegos educativos en los niños y las ventajas de los mismos en el proceso de enseñanza-aprendizaje.

**Modelación:** Se aplicó para representar mediante modelos las características y funcionalidades de los juegos durante el desarrollo de la investigación.

El presente documento está conformado por cuatro capítulos, estructurados de la siguiente manera:

**Capítulo 1:** Fundamentación teórica. Aborda los principales conceptos relacionados con la propuesta de solución. Se realiza un análisis de las soluciones similares en Cuba y en el mundo demostrando la necesidad de desarrollar los juegos Rompecabezas, Puzle, Tangram y Tetravex y se describen las metodologías, herramientas y tecnologías a utilizar para la implementación de los juegos educativos.

**Capítulo 2:** Características del sistema. Se describen las características de la propuesta de solución. Se identifican los requisitos funcionales y no funcionales, se elaboran los casos de uso del sistema y la descripción de cada uno de ellos.

**Capítulo 3:** Diseño del sistema. Se elabora el diseño de la solución construyendo los diagramas de clases y de secuencia. Además se realiza una selección del patrón arquitectónico, los patrones de diseño y los estilos de código a utilizar como parte de la estructuración y organización del sistema.

**Capítulo 4:** Implementación y pruebas. Se describen los principales elementos que caracterizan la implementación del sistema utilizando los diagramas de componentes. Además se definen las pruebas necesarias para garantizar la calidad de los juegos a desarrollar.

## Capítulo 1: Fundamentación teórica

### Introducción

En el presente capítulo se presentan los diferentes conceptos relacionados con el proceso de desarrollo de juegos educativos para niños con el objetivo de fortalecer el proceso de enseñanza aprendizaje. A su vez se realiza un estudio de las soluciones similares existentes a nivel nacional e internacional que pueden aportar una base sólida para el sistema propuesto y se describen las tecnologías, metodologías y herramientas a emplear durante su implementación.

### 1.1 Software educativo

El ordenador como herramienta de trabajo ha transformado muchos ambientes laborales, entre ellos el educativo. (4) En esta área es cada vez mayor el número de programas que se elaboran para facilitar la interacción de maestros y estudiantes. El doctor en pedagogía y profesor de la Universidad Autónoma de Barcelona Pere Marqués sostiene que se pueden emplear los términos "*programas didácticos*" y "*programas educativos*" para referirse a "*software educativo*", definiéndolo como programas para ordenador, creados con la finalidad específica de ser utilizados como medio didáctico para facilitar los procesos de enseñanza y de aprendizaje. (5)

#### 1.1.1 Clasificación de software educativo

Los programas educativos a pesar de tener rasgos esenciales básicos y una estructura general común se presentan con características muy diversas. (5) Laura Liguories es una de las especialistas de esta área que ha aportado su clasificación para agruparlos de manera general. Ella plantea que estos pueden ser de tipo tutorial, ejercitación o práctica, demostración, simulación y juegos. Sin embargo, a su entender dicha clasificación es arbitraria pues muchas veces un mismo programa puede combinar varias de estas modalidades. (6)

Atendiendo a la denominación anterior, la investigación en cuestión se centra en los juegos debido a que, como la propia autora los cataloga, son herramientas con las que la mayoría de los alumnos han interactuado. Además su alto valor pedagógico no se aparta de los intereses recreativos.

## 1.2 Juegos educativos de ordenador

El juego es una actividad indispensable para las personas, principalmente en edades tempranas de la vida. El niño a través del mismo aprende mucho más que por medio de cualquier otra vía, puesto que el infante se compromete personalmente; así el conocimiento que adquiere es muy valioso toda vez que se obtiene a partir de su propia experiencia. (7)

Hoy más que nunca el juego es aprovechado para los fines del conocimiento, pero el amplio uso de la computadora y el desarrollo de la tecnología han provocado que los más jóvenes se orienten a un nuevo entorno: los juegos de ordenador, por lo que su utilización en la escuela ha cambiado.

Para muchos estudiantes, el colegio es un lugar frío, difícil y muy poco llama la atención la educación como tal. La mejor forma de aprendizaje sucede cuando el alumno está motivado, comprometido y un ejemplo de cómo lograr la motivación es mediante los juegos electrónicos. (8)

Como bien expresan Patricia Mendoza Barros y Álvaro Galvis Panqueva los juegos de ordenador en la educación son una herramienta que permite a los estudiantes lograr sus objetivos trazados de una manera más entretenida. Los juegos son motivantes, lo cual es necesario para el aprendizaje a largo plazo, en otras palabras, para toda la vida; permiten descubrir conocimiento, se participa activamente en su aprendizaje y se retroalimenta. (8)

Por su parte, para Cristina Carmona y Eva Millán los juegos electrónicos educativos son sistemas de enseñanza en los que las actividades pedagógicas se mezclan con entornos “divertidos” que pretenden aumentar la motivación del alumno a la hora de enfrentarse a la tarea. Con esta actividad se pretende que un usuario, normalmente un niño, aprenda algo sobre determinada materia o que practique una determinada habilidad. (9)

Consuelo Giménez Parde los define como juegos orientados al entretenimiento (como todos y cada uno de los juegos) pero consiguiendo este entretenimiento por medio del aprendizaje. (10)

De acuerdo con el análisis de las definiciones aportadas por cada uno de los especialistas se toma como guía para la investigación la realizada por Cristina Carmona y Eva Millán pues es la que más abarca los aspectos de la solución que se quiere incorporar a la colección MundoClick.

## *Capítulo 1: Fundamentación teórica*

---

Las autoras Cristina Carmona y Eva Millán plantean que la calidad de un juego educativo está determinado por las siguientes características: (9)

- Como en todos los materiales para la instrucción, los juegos educativos deben ser apropiados, es decir, accesibles para el nivel de desarrollo del usuario al que está dirigido.
- Los estudiantes deben superar muchas barreras para realmente divertirse, como puede ser el miedo al fallo, el miedo al desconcierto y el temor a perder el control.
- Los juegos son divertidos gracias a la interacción. Los jugadores son los responsables de sus decisiones porque estas decisiones influyen en el resultado. El éxito en un juego se puede reflejar en el esfuerzo y la habilidad del usuario.
- Si se utilizan refuerzos, el usuario repetirá aquellos comportamientos que son recompensados y abandonará aquellos que son ignorados o penalizados.
- Un buen juego es aquél que los niños quieran jugar una y otra vez. Para ello deben tener unas reglas y objetivos claros; debe ser fácil, que los usuarios sepan cuál es su progreso a medida que van jugando; se debe poder utilizar varias estrategias; el juego debe ser tan motivador que haga que los niños mantengan el interés al enfrentarse a nuevos retos y a mejorar sus estrategias para ser mejores jugadores. Aunque muchos de los juegos actuales cuentan con sonidos, gráficos y otros efectos especiales que captan la atención del jugador, si el juego no está bien estructurado, estos elementos pueden resultar molestos. Para que el juego sea educativo, además de todo lo anterior, debe proporcionar un aprendizaje.
- En el caso de que el niño se quede estancado, un buen juego debe proporcionarle ayuda inmediata para evitar la frustración (normalmente a través de pistas).

En la construcción de la propuesta de solución es importante tener en cuenta las características mencionadas para obtener un producto que llegue de forma efectiva a los niños. Es necesario que los juegos, como bien se explica, sean apropiados para los estudiantes de primaria, divertidos, con reglas y objetivos bien definidos, que incluyan recursos multimedia correctamente estructurados, además que proporcionen la ayuda necesaria para que el usuario sepa cómo proceder en cada momento y muchos otros aspectos relevantes para lograr materiales con buena aceptación.

## 1.2.1 Tipos de juegos de ordenador

En las dos últimas décadas el desarrollo de juegos de video ha evolucionado, dando lugar a una gran variedad de estos. El doctor Marqués sostiene que aplicados al ámbito educativo se clasifican en puzzles y juegos de lógica, juegos de acción, de aventura y rol, simuladores, de estrategia y juegos de preguntas, además señala las habilidades que son capaces de fomentar cada uno: (11)

**Juegos de acción:** pueden contribuir al desarrollo psicomotor y de la orientación espacial de los estudiantes, aspecto especialmente útil en el caso de los más pequeños.

**Juegos de deportes:** permiten la ejercitación de diversas habilidades de coordinación psicomotora y profundizar en el conocimiento de las reglas y estrategias de los deportes.

**Juegos de aventura y rol:** promueve el desarrollo del conocimiento de diferentes temáticas, aportando valores y contravalores.

**Simuladores:** permite aprender a controlar la tensión y desarrollar la imaginación.

**Juegos de estrategia:** exigen administrar unos recursos escasos, prever los comportamientos de los rivales y trazar estrategias de actuación para lograr unos objetivos.

**Puzzles y juegos de lógica:** desarrollan la percepción espacial, la lógica, la imaginación y la creatividad.

**Juegos de preguntas:** pueden servir para repasar determinados conocimientos de todo tipo.

De todos los juegos descritos y que pueden ser vistos en el ámbito escolar, los autores consideran que los puzzles y los juegos de lógica se ajustan de mejor manera a las expectativas del trabajo pues son los que posibilitan el desarrollo de las habilidades requeridas en los pequeños.

## 1.3 Habilidades espaciales y de pensamiento lógico

A lo largo de la investigación se menciona que los juegos educativos tienen la capacidad de activar distintas habilidades cognitivas, para comprender el tema es necesario conocer bien este término. El doctor Carlos Álvarez de Zayas define habilidad como la dimensión del contenido que muestra el comportamiento del hombre en una rama del saber propio de la cultura de la humanidad. Es, desde el punto de vista psicológico, el sistema de acciones y

## *Capítulo 1: Fundamentación teórica*

---

operaciones dominado por el sujeto que responde a un objetivo. (12) Ernan Santisteban expresa que es una formación psicológica individual que gracias a la actividad y a la comunicación en un proceso de socialización, el sujeto cognoscente expresa un conocimiento en la práctica; dicho conocimiento se concreta en un sistema de acciones y operaciones dominadas por el sujeto que le permiten alcanzar un objetivo. (13)

Los autores anteriores no divergen mucho en sus definiciones, manteniendo que la habilidad es un sistema de acciones y operaciones dominadas por el sujeto que le permiten alcanzar un objetivo. En correspondencia con la investigación se toma la expresada por Ernan Santisteban pues plantea además que la habilidad se alcanza mediante la actividad.

A lo largo de la historia del hombre en lo que concierne al proceso educativo, se ha podido apreciar que las matemáticas, han sido una de las disciplinas más complejas para los niños. (14) Las dificultades de aprendizaje que con mayor frecuencia se presenta en esta rama del saber están relacionadas con la adquisición de saberes geométricos y el bajo rendimiento desde el punto de vista lógico que presentan los estudiantes. (15) Con motivo de reducir tales deficiencias es de interés abordar las habilidades espaciales y de pensamiento lógico-matemático.

En la escuela primaria la geometría permite lograr en el alumno el conocimiento del mundo, la interpretación de las formas y el espacio, mismas que le ayuden a resolver y transformar las situaciones problemáticas que se le presentan. (16)

En un estudio realizado por Luis Francisco García reconoce que las habilidades geométricas vienen condicionadas por las habilidades espaciales, entendiéndose éstas como las habilidades de generar, retener y manipular imágenes espaciales abstractas. (16)

El desarrollo de estos aspectos es fundamental para facilitar que los niños se apropien de contenidos geométricos como la ubicación espacial y la clasificación de cuerpos y figuras.

En otro caso, se entiende como lógico el pensamiento que es correcto, es decir, el pensamiento que garantiza que el conocimiento mediato que proporciona se ajusta a lo real. (17)

El profesor Víctor Royo Molina sostiene que cuando se utilizan las formas lógicas del pensamiento dentro la rama de las matemáticas para resolver ejercicios y problemas de una forma correcta, entonces se habla de un pensamiento lógico-matemático. (18)

## *Capítulo 1: Fundamentación teórica*

---

La inteligencia lógica matemática es la capacidad de razonamiento lógico, que se utiliza para resolver problemas de lógica y matemáticas. Está asociada a las habilidades de comprender y resolver cálculos numéricos, problemas de lógica y conceptos abstractos. (14)

Mediante el desarrollo de la inteligencia lógico matemática los niños y niñas logran: (14)

- Estimular sus capacidades intelectuales, se propicia el ordenamiento lógico de su pensamiento.
- Ejercitar la percepción y discriminación.
- Descubrir las nociones con referencia a objetos concretos.
- Adquirir un conocimiento adecuado y progresivo de las nociones tempo-espacial.
- Agrupar los objetos por sus características.

Un estudio del Departamento de Psicología de la Universidad de Chicago demuestra que las habilidades espaciales en los niños no son únicamente adquiridas de forma natural sino que se pueden potenciar, especialmente mediante los rompecabezas. (19)

Además los rompecabezas pueden ser elementos estratégicos para aprender y enseñar la matemática pues desarrollan habilidades que el estudiante necesita precisamente para entender los conceptos abstractos que maneja esta área. Todos los juegos de la propuesta de solución pertenecen a la clasificación de rompecabezas, específicamente de tipo lógico-matemático y de manera general fomentan: (15)

- La orientación espacial.
- El razonamiento lógico espacial.
- La memoria visual.
- La percepción de figura y fondo.
- La destreza mental.
- La destreza visual.
- La relación de ordenamiento lógico.
- La relación de asociación por números y colores.

## 1.4 Análisis de soluciones similares

Tanto en el ámbito internacional como en el nacional existen diversos juegos similares a la propuesta de solución, por lo que se hace necesario un estudio de los mismos para determinar si hay alguno que cumpla con los requisitos de la colección MundoClick.

### 1.4.1 Colección de software educativo GCompris

GCompris es un programa educativo especialmente diseñado para niños de entre dos y diez años. Es software libre y está disponible para Windows, Mac y Linux. Contiene una colección formada por más de cincuenta actividades y juegos dedicados a tratar todo tipo de temas: (20)

- Uso del ordenador y sus periféricos: acostumbrarse al teclado, ratón.
- Álgebra: memoria, números, tablas, imágenes.
- Ciencia: el ciclo del agua, electricidad, canales, el submarino.
- Geografía: países y mapas.
- Juegos: ajedrez, memoria, fútbol, rompecabezas.
- Lectura: diferentes ejercicios de lectura.
- Otros: las horas, dibujos vectoriales.

#### El juego de los quince

El juego de los quince es un juego educativo que forma parte de la colección GCompris y se encuentra dentro del tema Puzles. Tiene como objetivo mover cada elemento para construir series incrementales. Este juego posee varios niveles de dificultad, los que se podrán jugar de forma secuencial o el usuario podrá escoger el nivel de dificultad a jugar. Además presenta un elemento de ayuda en la que se brindará información sobre cómo jugar, y un elemento para regresar al menú principal.

#### El juego de puzle de tangram

El juego de puzle de tangram es un juego educativo que forma parte de la colección GCompris y se encuentra dentro del tema Puzles. Presenta un elemento de ayuda y un elemento para regresar al menú principal.

Al interactuar con el juego se selecciona el tangram a formar, las piezas se mueven arrastrándolas y se pueden rotar con ayuda de los botones que se encuentran en la parte

## *Capítulo 1: Fundamentación teórica*

---

inferior izquierda. Si se quiere dibujar el borde de la forma pulsamos el botón de forma. El juego tiene como objetivo construir una forma seleccionada con siete piezas.

### **Monta el puzle**

Monta el puzle es un juego educativo que forma parte de la colección GCompris y se encuentra dentro del tema Puzles. Tiene como objetivo seleccionar y mover los elementos para crear las pinturas originales. Este juego posee varios niveles de dificultad, los que se podrán jugar de forma secuencial o el usuario podrá escoger el nivel de dificultad a jugar. Presenta un elemento de ayuda, un elemento para regresar al menú principal y otro elemento con las opciones del juego.

### **1.4.2 Puzle–Números**

El juego educativo Puzle-Números es una nueva aplicación gratuita desarrollada para el sistema operativo Android la cual incluye rompecabezas de lógica. Tiene como tarea enseñarles a los niños en edad preescolar el mundo de los números. Consiste en escoger entre cinco imágenes diferentes el puzle de números a resolver, las piezas se arrastran hasta la zona marcada. El juego brinda además un elemento ayuda para observar la imagen completa y un sistema de premios que se le otorga al jugador que complete el puzle.

### **1.4.3 GNOME Tetravex**

GNOME Tetravex se distribuye bajo los términos de la Licencia Pública General GNU. La ventana del juego se divide en dos cajas; la parte derecha de la ventana contiene las piezas que se necesita organizar; la parte izquierda de la ventana es donde las piezas de la caja de la derecha se organizarán y moverán. Cada pieza se divide en cuatro pequeños triángulos que contienen un número. La finalidad del juego es mover las piezas de una parte del tablero a otra en el menor tiempo posible hasta posicionarlas de tal forma que solo los números iguales estén juntos. El usuario puede seleccionar cinco dimensiones diferentes desde 2x2 hasta 6x6. (21)

### **1.4.4 Colección de software educativo La caja mágica**

“La caja mágica” se crea para dotar al Sistema Educativo Venezolano de catorce software educativos, promoviendo así la formación integral de los estudiantes de la enseñanza primaria. (22)

## *Capítulo 1: Fundamentación teórica*

---

Esta colección dispone de varios juegos entre los que se encuentra el Rompecabezas y el Tangram los cuales pertenecen al producto educativo Las formas que nos rodean 1. El juego Rompecabezas consiste en seleccionar el rompecabeza que se desea armar, así como la cantidad de piezas y el tipo de corte. Las piezas se arrastran hacia el área en blanco y para girarlas se pulsa el botón derecho. El juego Tangram se juega seleccionando el tipo de tangram a formar. Las piezas se arrastran y se giran utilizando los botones de girar y se pueden colorear pulsando sobre el color. Al terminar cada uno de los juegos descritos anteriormente se le otorga un premio al usuario que haya completado correctamente algún rompecabeza o tangram y se le da a conocer el nivel de efectividad con que realizó el juego.

Al analizar las diferentes soluciones anteriores existentes a nivel nacional e internacional se destacan varios elementos positivos como la capacidad de despertar en el niño el desarrollo de la visión espacial, la lógica, la fantasía y la imaginación, permitiéndole divertirse y aprender al mismo tiempo. Los juegos presentan un alto grado de motivación ya que disponen de premios a los jugadores y de varios niveles de dificultad para garantizar la complejidad de los mismos.

Dichas soluciones cuentan con un conjunto de dificultades, entre las cuales se pueden encontrar que están desarrolladas para un país o un sistema operativo en específico, carecen de una historia por lo que no motiva al jugador a llegar al final del juego, no existe tiempo límite en cada nivel y no se guarda el estado del juego.

Estos juegos no pueden ser incorporados a MundoClick porque las tecnologías sobre las que han sido desarrollados no se corresponden con las requeridas por la plataforma de dicha colección. Además el código fuente de muchos de éstos no se encuentra disponible, es por eso que se decide desarrollar desde cero los juegos educativos Rompecabezas, Puzle, Tangram y Tetravex teniendo en cuenta las ventajas y desventajas que ofrecen las soluciones similares.

### **1.5 Plataforma Pointer**

La plataforma Pointer se encuentra orientada a la implementación y despliegue de juegos basados en HTML5, CCS3 y JavaScript. Su objetivo principal es apoyar el desarrollo de juegos creando una abstracción de varias funcionalidades comunes entre ellos; esto permitirá acortar el tiempo de implementación, estandarizar gran parte del código y ofrecer un ambiente controlado para el funcionamiento de los mismos. Pointer utiliza como tecnologías AppJS, NodeJS, JQuery, HTML5, CCS3 y JavaScript.

## 1.6 Metodología de desarrollo

En el proceso de desarrollo de los juegos Rompecabezas, Puzle, Tangram y Tetravex se utilizará la Metodología Tradicional RUP (*Rational Unified Process*). Dicha metodología se encuentra definida por la colección de juegos educativos MundoClick a la cual pertenece la propuesta de solución, debido a que brinda un seguimiento detallado de cada una de sus fases y permite generar una amplia documentación.

### 1.6.1 Metodología RUP

RUP es un proceso formal: provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Es guiado por casos de uso, centrado en la arquitectura, y propone UML como lenguaje de notación. Las cuatro fases del ciclo de vida son: Inicio, Elaboración, Construcción y Transición. (23)

## 1.7 Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado para la Construcción de Modelos (UML) se define como un lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software. (24)

Este lenguaje proporciona como ventaja en la representación del ciclo de vida del software una comunicación sencilla y rápida entre desarrolladores y clientes de la aplicación que se desarrolla.

## 1.8 Herramienta CASE de modelado con UML

Consiste en un conjunto de programas que ayudan y dan asistencia en la labor de los ingenieros, analistas y programadores en los pasos del ciclo de vida del desarrollo del software. Tiene como objetivo mejorar la calidad y minimizar el costo del sistema al ser el principal soporte para dicho trabajo. (25)

### 1.8.1 Visual Paradigm v8.0

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo del software: análisis y diseño orientados a objetos, construcción, pruebas y

# Capítulo 1: Fundamentación teórica

---

despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (26)

## 1.9 Lenguajes de desarrollo

Un lenguaje de programación es una técnica estándar de comunicación que puede ser utilizada para controlar el comportamiento de una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen la estructura de un lenguaje informático. (27)

### 1.9.1 HTML5

*Hyper Text Markup Language*, traducido al español como: lenguaje de marcado de hipertexto (HTML) se utiliza para presentar información en la *World Wide Web*. (28)

HTML5 es la quinta versión, la cual pretende remplazar al actual (X)HTML, corrigiendo problemas con los que los desarrolladores web se encuentran, así como rediseñar el código actualizándolo a nuevas necesidades que demanda la web de hoy en día. (29) Esta tecnología proporciona nuevas características: audio y video, canvas, almacenamiento local, web sockets, semántica, compatibilidad hacia atrás y eficacia en el renderizado. (30)

La utilización de esta tecnología permitirá la manipulación y maquetación de los contenidos web dentro de los juegos y se emplearán nuevas características que se mencionan anteriormente como audio y video y canvas. HTML5 garantiza la compatibilidad con la mayoría de los navegadores.

### 1.9.2 CSS3

*Cascade Style Sheets*, traducido al español como: hojas de estilo en cascada (CSS) es un lenguaje creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación, es imprescindible para la creación de páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto obliga a crear documentos HTML/XHTML bien definidos y con un significado completo. (31)

# Capítulo 1: Fundamentación teórica

---

La novedad más importante que aporta CSS3 consiste en la incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas, sin tener que recurrir a trucos, que a menudo complicaban el código. (32)

Según Antonio Ojeda las ventajas de usar CSS3 son: (33)

- Código más simple para muchas tareas.
- Mayores opciones de gráficas.

En la implementación de los juegos propuestos se empleará CSS3 para darle estilo y posicionamiento a los contenidos web que poseen los mismos.

## 1.9.3 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. (34) Es un lenguaje multiplataforma, pequeño y ligero; que no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores web. Dentro de un entorno anfitrión, puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos. (35)

Jerry Bradenbaugh plantea que algunas de las ventajas que presenta JavaScript son: (36)

- Fácil de aprender, rápido y potente. Una vez que se conocen las bases del lenguaje, no hay que esforzarse mucho para crear grandes aplicaciones.
- Usabilidad. JavaScript es el lenguaje de programación que más se utiliza en la Web.

JavaScript se empleará para la implementación de los juegos pues brinda gran dinamismo a los juegos propuestos. Además propicia mayor interactividad al sistema. Este lenguaje de programación posee numerosos frameworks que facilitan el desarrollo web.

## 1.10 Framework de desarrollo

Un framework constituye un conjunto de componentes que integran un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. (37)

La plataforma Pointer para desarrollar los juegos establece el uso de determinados frameworks los cuales se describen a continuación.

## 1.10.1 Bootstrap v2.3.2

Bootstrap es un framework que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. La mayor ventaja es que se pueden crear interfaces que se adapten a los distintos navegadores. (38)

Esta potente herramienta posee numerosos componentes webs ahorrando mucho esfuerzo y tiempo al programador. Se empleará dentro de la propuesta de solución para mostrar los mensajes al usuario.

## 1.10.2 JQuery v1.10.1

JQuery es una biblioteca JavaScript rápida, pequeña y rica en funciones, permite simplificar la manera de interactuar con los documentos HTML, manejo de eventos, desarrollar animaciones y agregar interacción con la técnica AJAX en páginas web. (39) Este framework dispone de una serie de funcionalidades implementadas, puestas al servicio del programador web y listas para ser usadas.

Es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de su mejora y actualización. Ad

emás tiene licencia para uso en cualquier tipo de plataforma y es gratuita. (40)

Esta librería se utilizará para manipular los elementos del DOM de HTML y para la comunicación con AJAX entre el cliente y el servidor.

## 1.10.3 NodeJS v0.8.11

NodeJS es un entorno de desarrollo en JavaScript del lado de servidor, totalmente asíncrono y orientado a eventos. Usa el motor de JavaScript V8 de Google, dotando a este entorno de una gran velocidad y agilidad, y sumando además una capacidad de Entrada/Salida realmente ligera y potente. Esto permite construir aplicaciones altamente escalables y escribir código que maneje miles de conexiones simultáneas en un solo equipo. (41)

NodeJS implementa los protocolos de comunicaciones en redes más habituales de los usados en Internet, como puede ser el HTTP, DNS, TLS, SSL. (42) También es de código abierto y multiplataforma.

# Capítulo 1: Fundamentación teórica

---

NodeJS es la base donde se implementa la plataforma, el servidor completo está montado sobre este entorno de desarrollo. En los juegos Rompecabezas, Puzle, Tangram y Tetravex se evidenciará en los controladores de cada uno, que devuelven información del servidor. Por ejemplo, a la hora de acceder a los datos de los ficheros .json.

## 1.10.4 AppJS v0.0.20

AppJS es una herramienta que permite crear aplicaciones atractivas de escritorio con HTML, CSS y JavaScript de una manera muy sencilla. Se puede decorar los widgets, haciendo uso de sombras, animación de elementos y transformación de objetos en el espacio 3D. La parte interesante de AppJS es que utiliza Node.js como la columna vertebral, siendo construida para facilitar el proceso de desarrollo de aplicaciones escalables de redes. Una de las características a destacar de AppJS es que es multiplataforma. (43)

## 1.10.5 CreateJS

CreateJS es un conjunto de librerías JavaScript con las que mejorar la experiencia de los usuarios. Este recurso de código abierto se compone de diferentes librerías especializadas para trabajar con efectos de movimiento, trabajo con el canvas de HTML5, carga elementos de forma asíncrona o extracción de animaciones en formato swf. Sin requerimiento previo y compatible con los principales navegadores web, todas las librerías de createJS pueden ser usadas de forma individual o en conjunto. (44)

CreateJS representa un recurso de mucha importancia para el desarrollo de los juegos de la propuesta de solución pues mediante las librerías que contiene se dibujarán los objetos de los juegos, permitirá trabajar las animaciones, las transiciones de imágenes y los eventos del sistema.

## 1.11 Entorno de desarrollo integrado

El entorno de desarrollo integrado es un programa que proporciona herramientas muy útiles para los programadores, tales como un editor de código, un compilador y un depurador para facilitar el desarrollo de otras aplicaciones. (45)

## 1.11.1 Netbeans v7.3

NetBeans IDE es un entorno de desarrollo para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java y soporta varios lenguajes de programación. Existe además un número importante de módulos para extenderlo. Es un producto libre y gratuito sin restricciones de uso. (46)

## 1.12 Herramienta de control de versiones (Subversion)

Es una herramienta de código abierto, multiplataforma, para el control de versiones de ficheros electrónicos, como son el software o la documentación. Se basa en un repositorio central que actúa como un servidor de ficheros, con la capacidad de recordar todos los cambios que se hacen tanto en sus directorios como en sus ficheros. El repositorio incrementa un número global de revisión con cada conjunto de cambios enviados al mismo. Es posible copiar y renombrar ficheros; crear una rama del proyecto es tan fácil como copiar un directorio. (47)

## Conclusiones del capítulo

Con la búsqueda de elementos teóricos referentes a los juegos educativos se llegó a la conclusión que el uso de estos programas en las escuelas es una alternativa que mejora las posibilidades de aprendizaje de los estudiantes, sobre todo al inicio de su vida escolar, no solo en cuanto a determinadas temáticas sino también con respecto a muchas habilidades que son relevantes para su crecimiento intelectual. El estudio de sistemas similares demostró que los mismos no cumplen con las necesidades requeridas para ser integrados a la colección MundoClick, pero aportaron un conjunto de elementos que serán vinculados a los nuevos juegos que se implementarán. Por último, el análisis realizado por los autores de la investigación reafirmó que la selección de las tecnologías, metodologías y herramientas por el grupo de trabajo del proyecto es la más indicada para el desarrollo de los juegos propuestos debido a las características y ventajas que presentan.

## **Capítulo 2: Características del sistema**

### **Introducción**

En el presente capítulo se describen brevemente las características de los juegos: Rompecabezas, Puzle, Tangram y Tetravex. Se definen los requisitos funcionales y no funcionales con el objetivo de establecer las cualidades y condiciones que debe cumplir el sistema. Se elabora los diagramas de casos de uso y las descripciones de los mismos con el propósito de facilitar la implementación de los juegos. Además se realiza un modelo conceptual para lograr un mayor entendimiento de los principales conceptos asociados a los juegos y las relaciones entre los mismos.

### **2.1 Descripción general de la propuesta del sistema**

Se propone la implementación de cuatro juegos para la colección MundoClick los cuales desarrollan habilidades espaciales y de pensamiento lógico-matemático en los niños de 6 a 12 años de edad. A continuación se realiza la descripción del funcionamiento de cada uno de los juegos a desarrollar.

#### **Juego Rompecabezas**

El juego tiene como finalidad ayudar a un niño y su familia a reconstruir un álbum de dibujos animados que ha sido destruido por su hermano pequeño. Está estructurado por seis niveles y en cada uno se debe resolver un rompecabeza. La dificultad de cada nivel estará vinculada con el tamaño de la matriz y el tipo de corte de las piezas. El juego brinda al jugador la bonificación “lupa” la cual señala la ubicación de una pieza dentro del rompecabeza y el bono pincel mágico en cada nivel el cual muestra la imagen original que se debe completar. Además, el juego muestra la cantidad de puntos acumulados y el tiempo que lleva el jugador en la partida. (Ver anexo 23)

#### **Juego Puzle**

El juego consiste en completar el mapa del tesoro del pirata Barba Negra el cual dividió el mismo en varias partes. Cada pedazo puede ser encontrado resolviendo el puzle asociado a este. Las piezas del puzle se mueven a través de un espacio en blanco hasta ordenarlas y conformar la imagen original. Cuenta con seis niveles y la complejidad radica en el tamaño de la

## *Capítulo 2: Características del sistema*

---

matriz del puzle. Cuando el jugador inicia el juego dispone de la bonificación “catalejo” que coloca sobre cada pieza el número con la posición que le corresponde dentro del puzle. Además, el juego muestra la cantidad de puntos acumulados y el tiempo que lleva el jugador en la partida. (Ver anexo 24)

### **Juego Tangram**

El juego tiene como objetivo llevar a la princesa hasta el final del pasadizo secreto del castillo. El pasadizo contiene diez secciones y para lograr avanzar de una sección a otra es necesario cruzar la puerta que las separa. La misma posee una imagen que se debe construir utilizando siete figuras geométricas las cuales se pueden rotar y también arrastrar hasta la posición deseada. A medida que avance el juego disminuye el tiempo en que se debe resolver el tangram. Durante el juego se le concederá al jugador el bono “lupa” el cual señala una figura y la posición que ocupa dentro de la imagen que se debe armar. Además, el juego muestra la cantidad de puntos acumulados y el tiempo que demora el jugador en la partida. (Ver anexo 25)

### **Juego Tetravex**

El juego consiste en devolverle los colores al arcoíris del pueblo Colorín el cual fue hechizado por un malvado brujo. Dispone de siete niveles y en cada uno de estos el jugador deberá resolver un rompecabeza de colores (Tetravex) para poder deshacer el hechizo. Las figuras del Tetravex son cuadrados que se forman con la unión de cuatro triángulos de diferentes colores que se deben posicionar de manera que solo los colores y números iguales coincidan. La complejidad de los niveles radica en el tamaño de la matriz. El juego brinda al jugador la bonificación “lupa” que señala una figura y la posición que ocupa dentro de la zona establecida para armar el tetravex. Además, se muestra la cantidad de puntos acumulados y el tiempo que lleva el jugador en la partida. (Ver anexo 26)

## **2.2 Especificación de los requisitos de software**

Un requisito del software es una característica que se debe exhibir por el software desarrollado o adaptado para solucionar un problema particular. Son típicamente una combinación compleja de requisitos de diversa personas en diversos niveles de una organización y del ambiente en el cual el software funcionará. (48) Se pueden clasificar en requisitos funcionales y no funcionales.

### 2.2.1 Requisitos funcionales

Cuando se habla de una característica requerida de la cual se sabe que va a ser satisfecha por medio de la adición de un subsistema o bloque de código en el software, entonces se dice que estamos ante un requisito funcional, por cuanto es un requisito que denota una funcionalidad del sistema. (49)

#### Requisitos funcionales del juego Rompecabezas:

**RF1:** Comenzar juego: El sistema debe mostrar la historia del juego y permitirle al jugador comenzar a jugar.

**RF2:** Mostrar rompecabeza: El sistema debe mostrar el rompecabeza a resolver.

**RF3:** Mover pieza: El sistema debe permitirle al jugador mover las piezas del rompecabeza.

**RF4:** Evaluar respuesta: El sistema debe evaluar la respuesta del jugador, notificándole la evaluación.

**RF5:** Registrar puntuación: El sistema debe registrar los puntos del jugador en cada nivel.

**RF6:** Obtener bonos: El sistema debe permitirle al jugador obtener bonos.

**RF7:** Usar bonos: El sistema debe permitirle al jugador hacer uso de los bonos activados en el momento en que lo desee.

**RF8:** Mostrar tiempo: El sistema debe mostrarle el tiempo al jugador en el transcurso del juego.

**RF9:** Mostrar ayuda: El sistema le permitirá al jugador consultar la ayuda cada vez que lo desee.

**RF10:** Cambiar de nivel: El sistema debe permitirle al jugador pasar al próximo nivel una vez vencido el actual.

**RF11:** Mostrar los resultados del juego. El sistema debe mostrarle al jugador al finalizar el juego la puntuación del mismo.

**RF12:** Terminar juego: El sistema debe permitirle al jugador terminar el juego en el momento que lo desee.

**RF13:** Guardar el estado del juego. El sistema debe permitirle al jugador que su juego se guarde una vez terminado.

## *Capítulo 2: Características del sistema*

---

**RF14:** Jugar varios usuarios: El sistema debe permitir la participación de hasta cuatro jugadores uno detrás de otro en la misma computadora.

### **Requisitos funcionales del juego Puzle:**

**RF1:** Comenzar juego: El sistema debe mostrar la historia del juego y permitirle al jugador comenzar a jugar.

**RF2:** Mostrar puzle: El sistema debe mostrar el puzle a resolver.

**RF3:** Mover pieza: El sistema debe permitirle al jugador mover las piezas del puzle.

**RF4:** Evaluar respuesta: El sistema debe evaluar la respuesta del jugador, notificándole la evaluación.

**RF5:** Registrar puntuación: El sistema debe registrar los puntos del jugador en cada nivel.

**RF6:** Obtener bonos: El sistema debe permitirle al jugador obtener bonos.

**RF7:** Usar bonos: El sistema debe permitirle al jugador hacer uso de los bonos activados en el momento en que lo desee.

**RF8:** Mostrar tiempo: El sistema debe mostrarle el tiempo al jugador en el transcurso del juego.

**RF9:** Mostrar ayuda: El sistema le permitirá al jugador consultar la ayuda cada vez que lo desee.

**RF10:** Cambiar de nivel: El sistema debe permitirle al jugador pasar al próximo nivel una vez vencido el actual.

**RF11:** Mostrar los resultados del juego. El sistema debe mostrarle al jugador al finalizar la puntuación del mismo.

**RF12:** Terminar juego: El sistema debe permitirle al jugador terminar el juego en el momento que lo desee.

**RF13:** Guardar el estado del juego. El sistema debe permitirle al jugador que su juego se guarde una vez terminado.

**RF14:** Jugar varios usuarios: El sistema debe permitir la participación de hasta cuatro jugadores uno detrás de otro en la misma computadora.

## *Capítulo 2: Características del sistema*

---

### **Requisitos funcionales del juego Tangram:**

**RF1:** Comenzar juego: El sistema debe mostrar la historia del juego y permitirle al jugador comenzar a jugar.

**RF2:** Mostrar tangram: El sistema debe mostrar el tangram a resolver.

**RF3:** Mover figura: El sistema debe permitirle al jugador mover las figuras del tangram.

**RF4:** Rotar figura: El sistema debe permitirle al jugador rotar las figuras del tangram.

**RF5:** Evaluar respuesta: El sistema debe evaluar la respuesta del jugador, notificándole la evaluación.

**RF6:** Registrar puntuación: El sistema debe registrar los puntos del jugador en cada nivel.

**RF7:** Obtener bonos: El sistema debe permitirle al jugador obtener bonos.

**RF8:** Usar bonos: El sistema debe permitirle al jugador hacer uso de los bonos activados en el momento en que lo desee.

**RF9:** Mostrar tiempo: El sistema debe mostrarle el tiempo al jugador en el transcurso del juego.

**RF10:** Mostrar ayuda: El sistema le permitirá al jugador consultar la ayuda cada vez que lo desee.

**RF11:** Cambiar de nivel: El sistema debe permitirle al jugador pasar al próximo nivel una vez vencido el actual.

**RF12:** Mostrar los resultados del juego. El sistema debe mostrarle al jugador al finalizar el juego la puntuación del mismo.

**RF13:** Terminar juego: El sistema debe permitirle al jugador terminar el juego en el momento que lo desee.

**RF14:** Guardar el estado del juego. El sistema debe permitirle al jugador que su juego se guarde una vez terminado.

**RF15:** Jugar varios usuarios: El sistema debe permitir la participación de hasta cuatro jugadores uno detrás de otro en la misma computadora.

### **Requisitos funcionales del juego Tetravex:**

## *Capítulo 2: Características del sistema*

---

**RF1:** Comenzar juego: El sistema debe mostrar la historia del juego y permitirle al jugador comenzar a jugar.

**RF2:** Mostrar tetravex: El sistema debe mostrar el tetravex a resolver.

**RF3:** Mover figura: El sistema debe permitir que el jugador mueva las figuras del tetravex.

**RF4:** Evaluar respuesta: El sistema debe evaluar la respuesta del jugador, notificándole la evaluación.

**RF5:** Registrar puntuación: El sistema debe registrar los puntos del jugador en cada nivel.

**RF6:** Obtener bonos: El sistema debe permitirle al jugador obtener bonos.

**RF7:** Usar bonos: El sistema debe permitirle al jugador hacer uso de los bonos activados en el momento en que lo desee.

**RF8:** Mostrar tiempo: El sistema debe mostrarle el tiempo al jugador en el transcurso del juego.

**RF9:** Mostrar ayuda: El sistema le permitirá al jugador consultar la ayuda cada vez que lo desee.

**RF10:** Cambiar de nivel: El sistema debe permitirle al jugador pasar al próximo nivel una vez vencido el actual.

**RF11:** Mostrar los resultados del juego. El sistema debe mostrarle al jugador al finalizar el juego la puntuación del mismo.

**RF12:** Terminar juego: El sistema debe permitirle al jugador terminar el juego en el momento que lo desee.

**RF13:** Guardar el estado del juego. El sistema debe permitirle al jugador que su juego se guarde una vez terminado.

**RF14:** Jugar varios usuarios: El sistema debe permitir la participación de hasta cuatro jugadores uno detrás de otro en la misma computadora.

### **2.2.2 Requisitos no funcionales**

Característica requerida del sistema, del proceso de desarrollo o del servicio prestado, que señala una restricción del software. (49)

## Capítulo 2: Características del sistema

---

### RNF 1: Requisito de portabilidad

El sistema podrá ser utilizado bajo los sistemas operativos Windows XP o superior y Linux distribución Ubuntu 12.04 o superior.

### RNF 2: Requisito de hardware

- Requiere 500MB de espacio libre en el disco duro.
- Como mínimo 512MB de RAM (recomendado 1GB de RAM o mayor).
- Soporte de video que admita resolución de al menos 1024x780px y 24 bits.

### RNF 3: Requisito de interfaz

La interfaz del sistema debe presentar rasgos de diseño infantiles, predominando el uso de colores atractivos para captar la atención del niño e imágenes en lugar de textos para sugerir las acciones.

### RNF 4: Requisitos de soporte

Se realizará la transferencia tecnológica de la colección a los clientes.

### RNF 5: Requisitos de usabilidad

El software será usado por los niños en las edades de 6 a 12 años.

### RNF 6: Restricciones de diseño e implementación

- Lenguaje de desarrollo: JavaScript
- Lenguaje de marcado: HTML5
- Hoja de estilo: CSS3
- Herramientas de desarrollo: jQuery v1.10.1, NodeJs v0.8.11, AppJS v0.0.20, Bootstrap v2.3.2 y CreateJS
- Entorno de Desarrollo Integrado: NetBeans v7.3
- Herramienta de modelado UML: Visual Paradigm v8.0

## 2.3 Modelo conceptual

Un modelo conceptual es una representación de conceptos en un dominio del problema; es el artefacto más importante a crear durante el análisis orientado a objetos. Además muestra conceptos, asociaciones entre ellos y atributos de los mismos. (24)

## *Capítulo 2: Características del sistema*

---

A continuación se definen los elementos principales que intervienen en el modelo conceptual de cada uno de los juegos a desarrollar.

### **Juego Rompecabezas**

**Juego:** Conjunto de niveles en los que se muestran rompecabezas correspondiente a dibujos animados que el jugador debe armar.

**Jugador:** Usuario que interactúa con el juego Rompecabezas.

**Bonificación:** Premio que recibe el jugador.

**Nivel:** Cada nivel representa un desafío diferente donde aumenta la complejidad del juego a medida que el jugador va avanzando.

**Pieza:** Piezas que serán usadas por los jugadores para completar los rompecabezas.

**Tipo de corte:** Tipos de cortes de las piezas del rompecabezas que serán utilizadas en el juego.

**Cuadrado, Irregular:** Tipos de cortes de las piezas.

**Complejidad:** Grado de dificultad a jugar. La complejidad de los niveles pueden ser de tres tipos: Fácil, Medio o Difícil.

**Fácil:** Primer nivel de complejidad.

**Medio:** Segundo nivel de complejidad.

**Difícil:** Último nivel de complejidad.

## Capítulo 2: Características del sistema

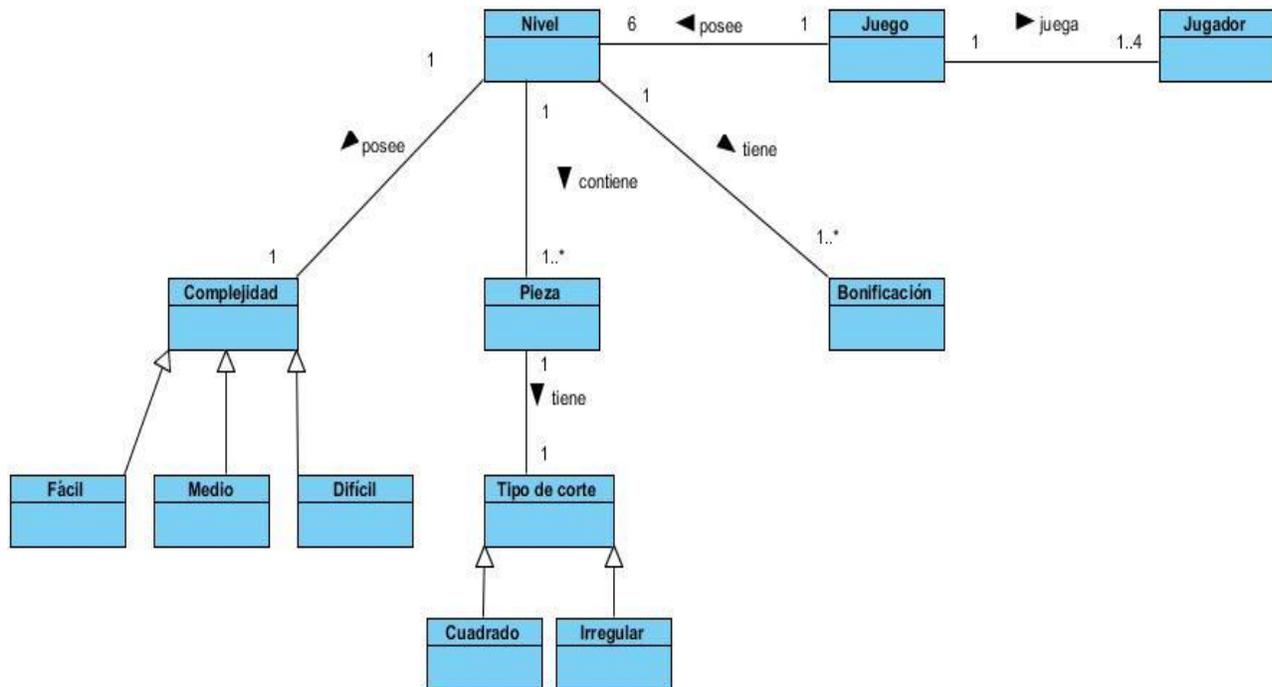


Figura 1: Modelo conceptual del juego Rompecabezas

### Juego Puzle

**Juego:** Conjunto de niveles en los que se muestran puzles que el jugador debe armar para completar el mapa del tesoro.

**Jugador:** Usuario que interactúa con el juego Puzle.

**Bonificación:** Premio que recibe el jugador.

**Nivel:** Cada nivel representa un desafío diferente donde aumenta la complejidad del juego a medida que el jugador va avanzando.

**Pieza:** Piezas que serán usadas por los jugadores para completar los puzles.

**Complejidad:** Grado de dificultad a jugar. La complejidad de los niveles pueden ser de tres tipos: Fácil, Medio o Difícil.

**Fácil:** Primer nivel de complejidad.

**Medio:** Nivel intermedio de complejidad.

## Capítulo 2: Características del sistema

**Difícil:** Último nivel de complejidad.

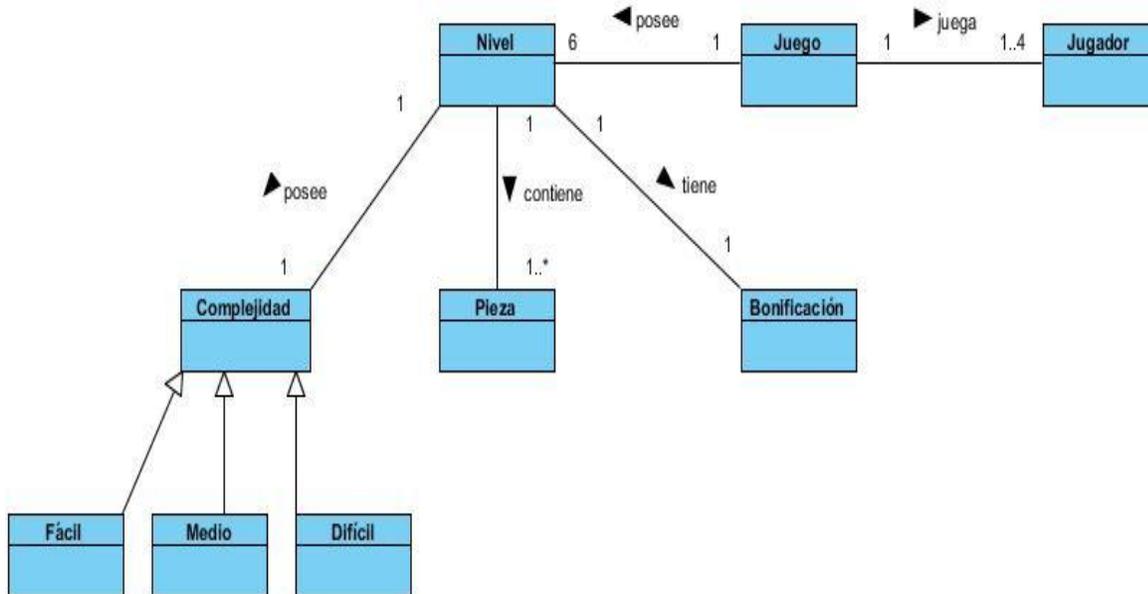


Figura 2: Modelo conceptual del juego Puzle

### Juego Tangram

**Juego:** Cada nivel constituye una puerta de un pasadizo secreto y para atravesarlas el jugador debe resolver un tangram.

**Jugador:** Usuario que interactúa con el juego Tangram.

**Bonificación:** Premio que recibe el jugador.

**Nivel:** Cada nivel representa un desafío diferente donde aumenta la complejidad del juego a medida que el jugador va avanzando.

**Figura:** Figuras que serán usadas por los jugadores para completar los tangram.

**Triángulo, Paralelogramo, Cuadrado:** Tipos de figuras que serán utilizadas en el juego.

**Tiempo:** Delimita un intervalo de tiempo durante el cual el jugador debe resolver un tangram en cada nivel.

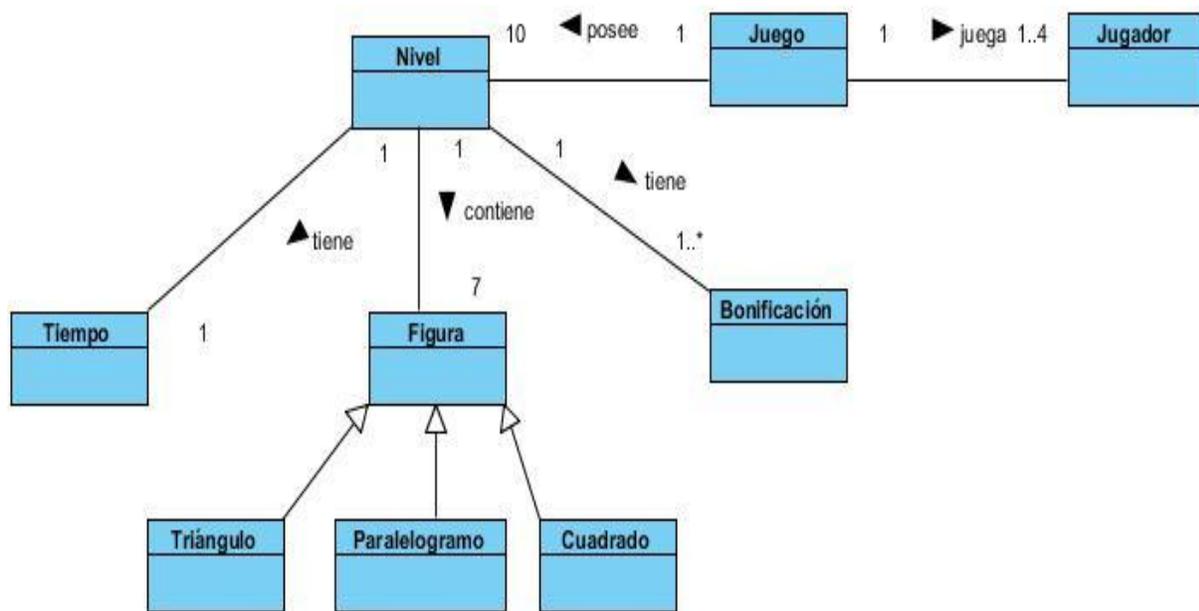


Figura 3: Modelo conceptual del juego Tangram

### Juego Tetravex

**Juego:** En cada nivel el jugador debe resolver un tetravex para romper el hechizo y devolverle los colores del arcoíris del pueblo Colorín.

**Jugador:** Usuario que interactúa con el juego Tetravex.

**Bonificación:** Premio que recibe el jugador.

**Nivel:** Cada nivel representa un desafío diferente donde aumenta la complejidad del juego a medida que el jugador va avanzando.

**Figura:** Figuras que serán usadas por los jugadores para completar los tetravex.

**Complejidad:** Grado de dificultad a jugar. La complejidad de los niveles pueden ser de cuatro tipos: Fácil, Medio, Díficil o Avanzado.

**Fácil:** Primer nivel de complejidad.

**Medio:** Segundo nivel de complejidad.

**Díficil:** Tercer nivel de complejidad.

**Avanzado:** Último nivel de complejidad.

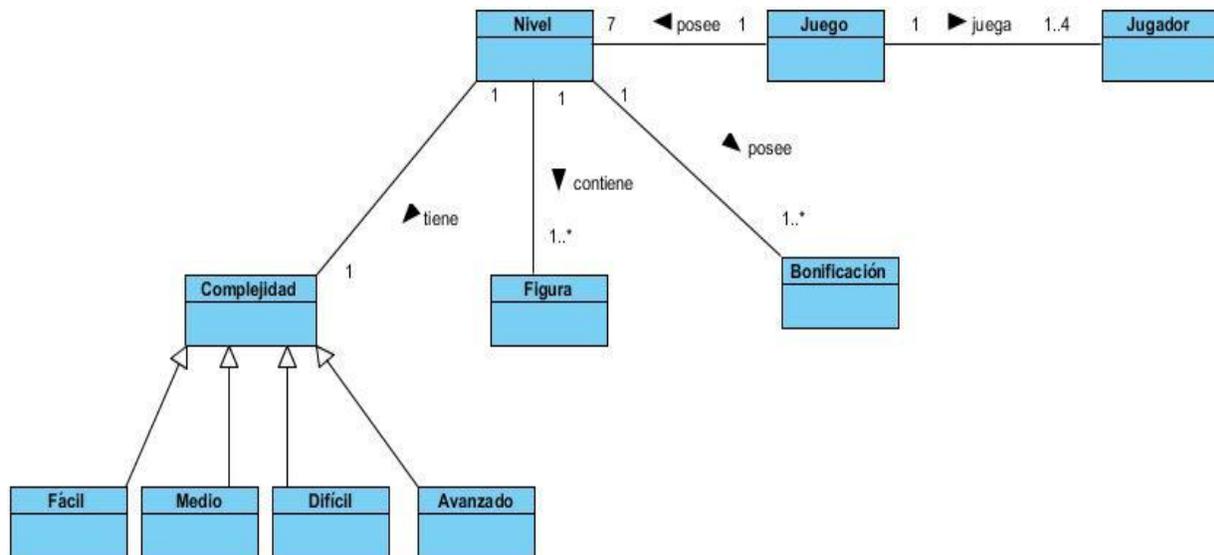


Figura 4: Modelo conceptual del juego Tetravex

### 2.4 Modelo de casos de uso del sistema

El modelo de casos de uso representa los requisitos del sistema. El mismo ayuda al cliente, a los usuarios y desarrolladores a llegar a un acuerdo sobre cómo utilizar la aplicación. Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo y que producen un resultado observable de valor para un actor concreto. (50)

A continuación se presentan los diagramas de casos de uso de los juegos Rompecabezas, Puzle, Tangram y Tetravex, así como la descripción de los casos de uso más significativos de los mismos. Las restantes descripciones se pueden encontrar en los anexos del presente documento.

## Capítulo 2: Características del sistema

### Juego Rompecabezas

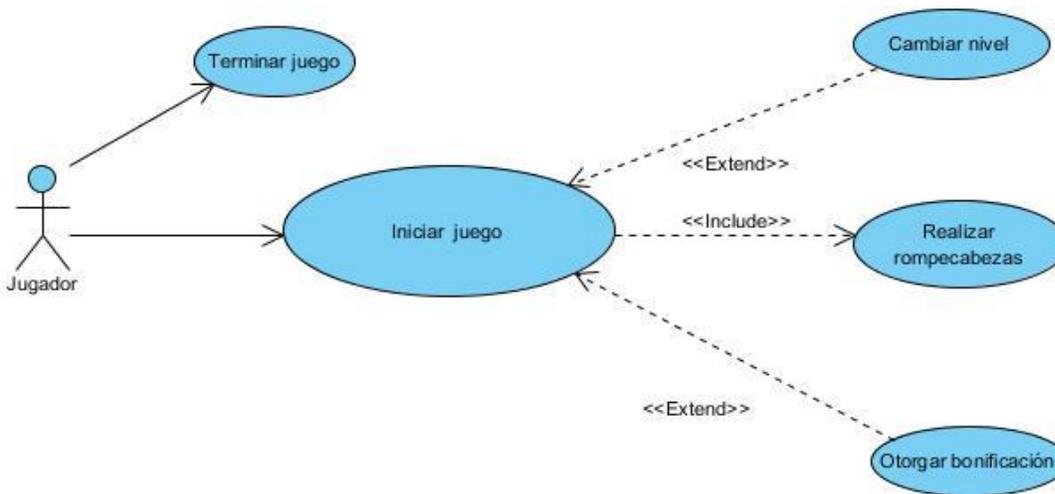


Figura 5 : Diagrama de casos de uso del juego Rompecabezas

Tabla 1: Descripción del caso de uso Iniciar juego del juego Rompecabeza

<b>Caso de Uso</b>	Iniciar juego	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia una vez que el jugador selecciona el juego. El sistema carga el juego, muestra el comic inicial y la pantalla principal del juego. Si el jugador selecciona la opción salir el sistema salva los datos y termina el juego, finalizando así el caso de uso.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El jugador se autenticado en la plataforma.	
<b>Postcondiciones</b>	Se inició el juego.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona el juego.	1.1 El sistema verifica el jugador (es) y brinda las siguientes opciones: <ul style="list-style-type: none"> <li>• Si el jugador ha jugado anteriormente el</li> </ul>

## *Capítulo 2: Características del sistema*

		<p>juego y no ha culminado todos los niveles muestra una interfaz con el nivel actual en que se encuentra y las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• “Continuar”: ir al paso 1.1 de la sección 2 “Iniciar juego”.</li> <li>• “Comenzar desde el principio” ir al paso 2 del flujo básico.</li> <li>• Si hay más de un jugador autenticados en el sistema habilita el nivel multijugador: ir a la sección 3 “Multijugador”.</li> <li>• Si el jugador no ha jugado el sistema permite comenzar desde el principio.</li> </ul>
2.	Selecciona la opción “Comenzar desde el principio”.	<p>2.1 Muestra la historia del juego en forma de comic.</p> <p>2.2 Muestra una interfaz que permite al jugador seleccionar una de las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Jugar el nivel de entrenamiento: ir a la sección “Nivel de entrenamiento”.</li> <li>• Comenzar a jugar: ir a la sección “Iniciar juego”.</li> </ul>
<b>Sección 1 “Nivel de entrenamiento”: Flujo Normal de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción “Jugar nivel de entrenamiento”.	<p>1.1 Muestra un mensaje de bienvenida al nivel y la siguiente opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
2.	Selecciona la opción “Continuar”.	<p>2.1 Muestra una interfaz donde se explica en que consiste el juego y la opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
3.	Selecciona la opción “Continuar”.	<p>3.1 Muestra un interfaz donde explica cómo mover</p>

## *Capítulo 2: Características del sistema*

		<p>las piezas y la opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
4.	Selecciona la opción "Continuar".	<p>4.1 Muestra una interfaz donde explica en que consiste las bonificaciones del juego y la opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
5.	Selecciona la opción "Continuar".	5.1 Muestra el rompecabezas a armar.
6.	Mueve una pieza.	<p>6.1 Comprueba que la posición donde se ubica la pieza dentro de la matriz esté vacía.</p> <p>6.2 Si la pieza es ubicada en una posición que se encuentra ocupada por otra pieza, regresa a su posición anterior.</p> <p>6.3 Verifica que cada pieza ocupe el lugar que le corresponde dentro de la matriz.</p>
7.	Completa el rompecabezas.	<p>7.1 Cuando todas las piezas estén ubicadas correctamente se muestra una interfaz con la siguiente opción:</p> <ul style="list-style-type: none"> <li>• Ir al juego.</li> </ul>
8.	Selecciona la opción "Ir al juego".	<p>8.1 Muestra el primer nivel y se comenzará a ejecutar el flujo normal de eventos en la sección 2 "Iniciar juego" en el paso 1.1.</p> <p>8.2 Termina el caso de uso.</p>
<b>Sección 2 "Iniciar juego": Flujo Normal de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción "Comenzar a jugar".	<p>1.1 Muestra el rompecabeza que debe completar el jugador. (Ver CU Realizar rompecabezas)</p> <p>1.2 Muestra al jugador el momento en que pasa de</p>

## *Capítulo 2: Características del sistema*

		<p>nivel. (Ver CU Cambiar nivel)</p> <p>1.3 Muestra cuando un jugador obtiene una bonificación. (Ver CU Otorgar bonificación)</p> <p>1.4 Muestra el contador de tiempo.</p> <p>1.5 Muestra la puntuación del jugador.</p>
2.		2.1 Termina el caso de uso.
<b>Sección 3 “Multijugador”: Flujo Normal de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		<p>1.1 Muestra el rompecabeza a armar.</p> <p>1.2 Muestra la imagen que se debe completar.</p> <p>1.3 Muestra el jugador en turno.</p> <p>1.4 Muestra el contador de tiempo.</p> <p>1.5 Muestra el contador de puntos.</p>
2.	Mueve una pieza.	<p>2.1 Comprueba que la posición donde se ubica la pieza dentro de la matriz esté vacía.</p> <p>2.2 Si la pieza se ubica en una posición donde existe otra pieza, regresa a su posición anterior.</p> <p>2.3 Verifica que cada pieza ocupe el lugar que le corresponde dentro de la matriz.</p>
3.	Completa el rompecabezas.	<p>3.1 Cuando todas las piezas estén ubicadas correctamente muestra el rompecabeza y se comienza a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1.</p> <p>3.2 Cuando terminen de jugar todos los jugadores se mostrará los resultados siguientes:</p> <ul style="list-style-type: none"> <li>• Nombre de cada jugador.</li> <li>• Puntos obtenidos.</li> </ul>

## Capítulo 2: Características del sistema

		<ul style="list-style-type: none"> <li>Lugar alcanzado.</li> </ul> <p>3.3 Muestra las opciones siguientes:</p> <ul style="list-style-type: none"> <li>Reiniciar.</li> <li>Salir.</li> </ul>
4.	Selecciona la opción "Reiniciar".	<p>4.1 Se ejecuta el flujo básico de eventos en el paso 1.1.</p> <p>4.2 Termina el caso de uso.</p>
<b>Flujo alternativo: "Salir"</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción "Salir".	<p>1.1 Sale del juego.</p> <p>1.2 Termina el caso de uso.</p>
<b>Relaciones</b>	<b>CU Incluidos</b>	Realizar rompecabezas
	<b>CU Extendidos</b>	Otorgar bonificación Cambiar nivel

### Juego Puzle

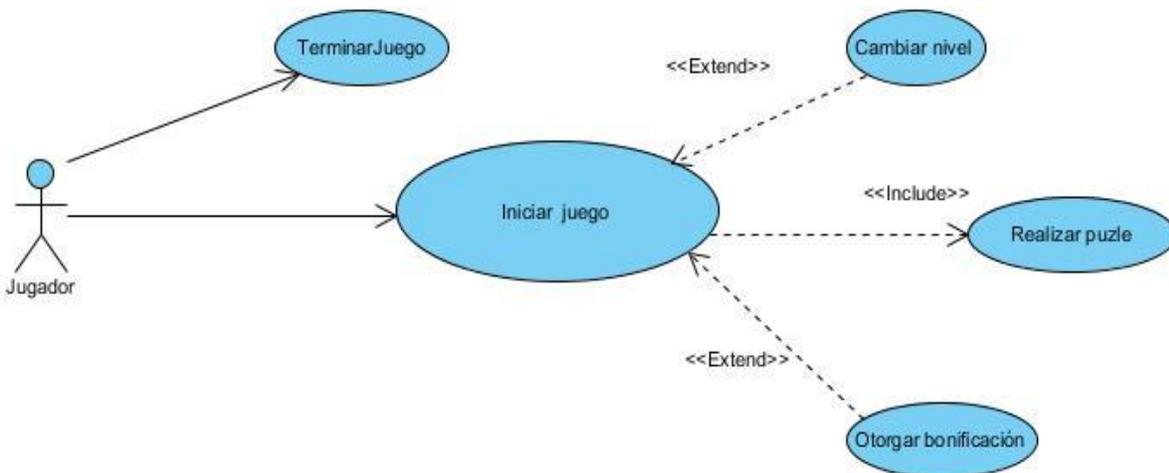


Figura 6: Diagrama de casos de uso del juego Puzle

## Capítulo 2: Características del sistema

Tabla 2: Descripción del caso de uso Iniciar juego del juego Puzle

<b>Caso de uso</b>	Iniciar juego	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia una vez que el jugador selecciona el juego. El sistema carga el juego, muestra el comic inicial y la pantalla principal del juego. Si el jugador selecciona la opción salir el sistema salva los datos y termina el juego, finalizando así el caso de uso.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El jugador debe estar registrado en el sistema.	
<b>Postcondiciones</b>	Se inició el juego.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona el juego.	<p>1.1 El sistema verifica el jugador (es) y brinda las siguientes opciones:</p> <ul style="list-style-type: none"><li>• Si el jugador ha jugado anteriormente el juego y no ha culminado todos los niveles muestra una interfaz con el nivel actual en que se encuentra y las siguientes opciones: “Continuar”: ir al paso 1.1 de la sección 2 “Iniciar juego”.</li><li>• “Comenzar desde el principio” ir al paso 2 del flujo básico.</li><li>• Si hay más de un jugador autenticados en el sistema habilita el nivel multijugador: ir a la sección 3 “Multijugador”.</li></ul>

## *Capítulo 2: Características del sistema*

		<ul style="list-style-type: none"> <li>• Si el jugador no ha jugado el sistema permite comenzar desde el principio.</li> </ul>
2.	Selecciona la opción “Comenzar desde el principio”.	<p>2.1 Muestra la historia del juego en forma de comic.</p> <p>2.2 Muestra una interfaz que permite al jugador seleccionar una de las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Jugar nivel de entrenamiento: ir a la sección: “Nivel de entrenamiento”.</li> <li>• Comenzar a jugar: ir a la sección: “iniciar juego”.</li> </ul>
<b>Sección 1 “Nivel de entrenamiento”: Flujo Normal de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción “Jugar nivel de entrenamiento”.	<p>1.1 Muestra un mensaje de bienvenida al nivel y la siguiente opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
2.	Selecciona la opción “Continuar”.	<p>2.1 Muestra una interfaz donde se explica en que consiste el juego y la opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
3.	Selecciona la opción “Continuar”.	<p>3.1 Muestra un interfaz donde explica cómo mover las piezas y la opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
4.	Selecciona la opción “Continuar”.	<p>4.1 Muestra una interfaz donde explica en que consiste la bonificación del juego y la opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
5.	Selecciona la opción “Continuar”.	<p>5.1 Muestra la imagen que se debe completar.</p> <p>5.2 Muestra el puzle a armar.</p>

## *Capítulo 2: Características del sistema*

6.	Mueve una pieza.	<p>6.1 Verifica que exista un espacio vacío al lado de la pieza.</p> <p>6.2 Verifica que la pieza ocupe la posición que le corresponde dentro de la matriz.</p>
7.	Completa el puzle.	<p>7.1 Cuando todas las piezas estén ubicadas correctamente muestra una interfaz indicando que el nivel culminó y la siguiente opción:</p> <ul style="list-style-type: none"> <li>• Ir al juego.</li> </ul>
8.	Selecciona la opción “Ir al juego”.	<p>8.1 Muestra el primer nivel y se comenzará a ejecutar el flujo normal de eventos en la sección 2 “Iniciar juego” en el paso 1.1.</p> <p>8.2 Termina el caso de uso.</p>

### Sección 2 “Iniciar juego”: Flujo Normal de Eventos

	Actor	Sistema
1.	Selecciona la opción “Comenzar a jugar”.	<p>1.1 Muestra el puzle que debe completar el jugador. (Ver CU Realizar puzle)</p> <p>1.2 Muestra al jugador el momento en que pasa de nivel. (Ver CU Cambiar nivel)</p> <p>1.3 Muestra cuando un jugador obtiene una bonificación. (Ver CU Otorgar bonificación)</p> <p>1.4 Muestra el contador de tiempo.</p>
2.		2.1 Termina el caso de uso.

### Sección 3 “Multijugador”: Flujo Normal de Eventos

	Actor	Sistema
1.		<p>1.1 Muestra el puzle a armar.</p> <p>1.2 Muestra la imagen que se debe completar.</p>

## *Capítulo 2: Características del sistema*

		<p>1.3 Muestra el jugador en turno.</p> <p>1.4 Muestra el contador de tiempo.</p> <p>1.5 Muestra el contador de puntos.</p>
2.	Mueve una pieza.	<p>2.1 Verifica que exista un espacio vacío al lado de la pieza.</p> <p>2.2 Verifica que cada pieza ocupe la posición que le corresponde dentro de la matriz.</p>
3.	Completa el puzle.	<p>3.1 Cuando todas las piezas estén ubicadas correctamente muestra el puzle y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1.</p> <p>3.2 Cuando terminen de jugar todos los jugadores se muestran los resultados siguientes:</p> <ul style="list-style-type: none"> <li>• Nombre de cada jugador.</li> <li>• Puntos obtenidos.</li> <li>• Lugar alcanzado.</li> </ul> <p>3.3 Muestra las opciones siguientes:</p> <ul style="list-style-type: none"> <li>• Reiniciar.</li> <li>• Salir.</li> </ul>
4.	Selecciona la opción "Reiniciar".	<p>4.1 Se ejecuta el flujo básico de eventos en el paso 1.1.</p> <p>4.2 Termina el caso de uso.</p>
<b>Flujo alterno: "Salir"</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción "Salir".	<p>1.1 Sale del juego.</p> <p>1.2 Termina el caso de uso.</p>

## *Capítulo 2: Características del sistema*

<b>Relaciones</b>	<b>CU Incluidos</b>	Realizar puzle
	<b>CU Extendidos</b>	Otorgar bonificación Cambiar nivel

### Juego Tetravex

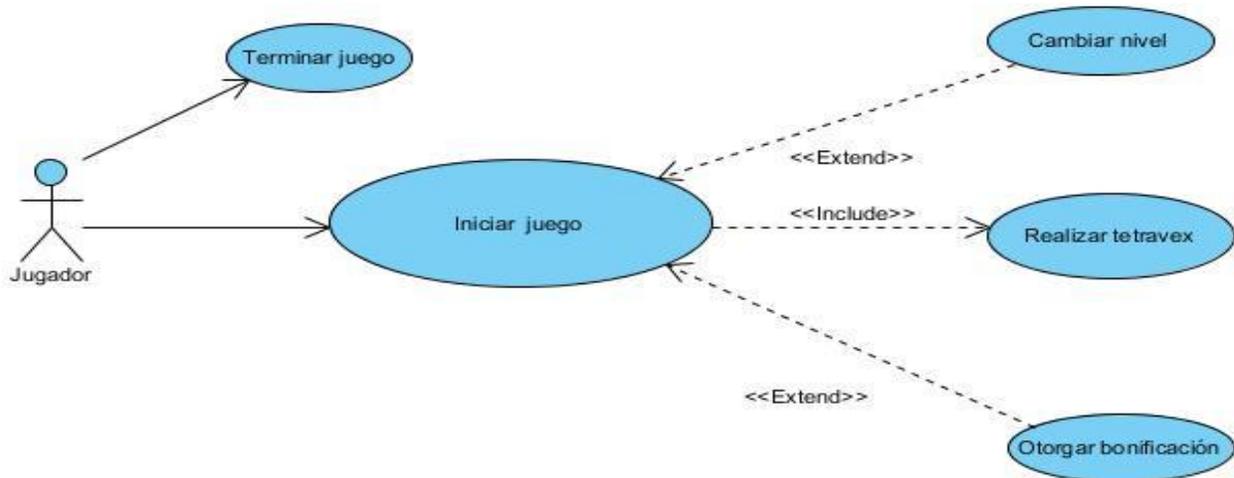


Figura 7: Diagrama de casos de uso del juego Tetravex

Tabla 3: Descripción del caso de uso Iniciar juego del juego Tetravex

<b>Caso de uso</b>	Iniciar juego
<b>Actores</b>	Jugador
<b>Resumen</b>	El caso de uso inicia una vez que el jugador selecciona el juego. El sistema carga el juego, muestra el comic inicial y la pantalla principal del juego. Si el jugador selecciona la opción salir el sistema salva los datos y termina el juego, finalizando así el caso de uso.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	El jugador se encuentre autenticado en el sistema.
<b>Postcondiciones</b>	Se realizó el juego.

## *Capítulo 2: Características del sistema*

Flujo de eventos		
	Actor	Sistema
1.	Selecciona el juego.	<p>1.1 El sistema verifica el jugador (es) y brinda las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Si el jugador ha jugado anteriormente el juego y no ha culminado todos los niveles muestra una interfaz con el nivel actual en que se encuentra y las siguientes opciones:  “Continuar”: ir al paso 1.1 de la sección 2  “Iniciar juego”.</li> <li>• “Comenzar desde el principio” ir al paso 2 del flujo básico.</li> <li>• Si hay más de un jugador autenticados en el sistema habilita el nivel multijugador: ir a la sección “Multijugador”.</li> </ul> <p>Si el jugador no ha jugado el sistema permite comenzar desde el principio.</p>
2.	Selecciona la opción “Comenzar desde el principio”.	<p>2.1 Muestra la historia del juego en forma de comic.</p> <p>2.2 Muestra una interfaz que permite al jugador seleccionar una de las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Jugar nivel de entrenamiento: ir a la sección: “Nivel de entrenamiento”.</li> <li>• Comenzar a jugar: ir a la sección: “iniciar juego”.</li> </ul>
<b>Sección 1 “Nivel de entrenamiento”: Flujo Normal de Eventos</b>		
	Actor	Sistema

## Capítulo 2: Características del sistema

1.	Selecciona la opción "Jugar nivel de entrenamiento".	1.1 Muestra un mensaje de bienvenida al nivel y la siguiente opción: <ul style="list-style-type: none"><li>• Continuar.</li></ul>
2.	Selecciona la opción "Continuar".	2.1 Muestra el tetravex a armar, un mensaje explicando en que consiste el juego y la siguiente opción: <ul style="list-style-type: none"><li>• Continuar.</li></ul>
3.	Selecciona la opción "Continuar".	3.1 Muestra un mensaje donde orienta al jugador que mueva una figura hasta la casilla que le corresponde y la opción: <ul style="list-style-type: none"><li>• Continuar.</li></ul>
4.	Selecciona la opción "Continuar".	4.1 Señala una figura de la derecha y la posición que le corresponde en el espacio de solución.
5.	Mueve la figura indicada.	5.1 Comprueba que la posición donde se ubica la figura dentro del espacio de solución esté vacía. 5.2 Si la figura se ubica en una posición que se encuentra ocupada por otra figura, regresa a su posición anterior. 5.3 Verifica que la figura se posicione de forma tal que solo los colores y los números iguales coincidan, en caso contrario la figura regresa a su posición anterior. 5.4 Muestra un mensaje indicando que se puede usar la bonificación lupa y la siguiente opción: <ul style="list-style-type: none"><li>• Continuar.</li></ul>
6.	Selecciona la opción "Continuar".	6.1 Activa la lupa.
7.	Hace click en el icono lupa.	7.1 Realiza el flujo normal de eventos en el paso 4.1.

## *Capítulo 2: Características del sistema*

8.	Mueve la figura indicada.	<p>8.1 Realiza el flujo normal de eventos en el paso 5.1, 5.2 y 5.3.</p> <p>8.2 Muestra un mensaje indicando que se debe mover la última figura y la opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
9.	Selecciona la opción "Continuar".	<p>9.1 Cuando termina de mover todas las figuras muestra un mensaje anunciando que se completó el nivel y la opción:</p> <ul style="list-style-type: none"> <li>• Ir al juego.</li> </ul>
10.	Selecciona la opción "Ir al juego".	<p>10.1 Muestra el primer nivel y se comenzará a ejecutar el flujo normal de eventos en la sección 2 "Iniciar juego" en el paso 1.1.</p> <p>10.2 Termina el caso de uso.</p>

### **Sección 2 "Iniciar juego": Flujo Normal de Eventos**

	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción "Comenzar a jugar".	<p>1.1 Muestra el tetravex que debe completar el jugador. (Ver CU Realizar tetravex)</p> <p>1.2 Muestra al jugador el momento en que pasa de nivel. (Ver CU Cambiar nivel)</p> <p>1.3 Muestra cuando un jugador obtiene una bonificación. (Ver CU Otorgar bonificación)</p> <p>1.4 Muestra el contador de tiempo.</p> <p>1.5 Muestra la puntuación del jugador.</p>
2.	Selecciona la opción "Salir del juego".	<p>2.1 Almacena los datos del jugador una vez concluido el juego.</p> <p>2.2 Termina el caso de uso.</p>

## Capítulo 2: Características del sistema

Sección 3 “Multijugador”: Flujo Normal de Eventos		
	Actor	Sistema
1.		1.1 Muestra un mensaje de bienvenida al modo retador, las reglas del nivel y la opción: <ul style="list-style-type: none"><li>• Jugar.</li></ul>
2	Selecciona la opción “Jugar”.	2.1 Muestra el tetravex a armar. 2.2 Muestra el jugador en turno. 2.3 Muestra el contador de tiempo.
3.	Mueve una figura.	3.1 Comprueba que la posición donde se ubica la figura dentro del espacio de solución esté vacía. 3.2 Si la figura se ubica en una posición que se encuentra ocupada por otra figura, regresa a su posición anterior. 3.3 Verifica que la figura se posicione de forma tal que solo los colores y los números iguales coincidan, en caso contrario la figura regresa a su posición anterior. 3.4 Muestra el resultado de la verificación aumentando o disminuyendo la puntuación.
4.	Completa el tetravex.	4.1 Cuando todas las figuras estén ubicadas correctamente muestra el tetravex y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 2.1. 4.2 Cuando terminen de jugar todos los jugadores se mostrará los resultados siguientes: <ul style="list-style-type: none"><li>• Nombre de cada jugador.</li><li>• Puntos obtenidos.</li><li>• Lugar alcanzado.</li></ul>

## Capítulo 2: Características del sistema

		4.3 Muestra las opciones siguientes: <ul style="list-style-type: none"> <li>• Reiniciar.</li> <li>• Salir.</li> </ul>
5.	Selecciona la opción "Reiniciar".	5.1 Se ejecuta el flujo básico de eventos en el paso 1.1. 5.2 Termina el caso de uso.
<b>Flujo alterno: "Salir"</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción "Salir".	1.1 Sale del juego. 1.2 Termina el caso de uso.
<b>Relaciones</b>	<b>CU Incluidos</b>	Realizar tetravex
	<b>CU Extendidos</b>	Otorgar bonificación Cambiar nivel

### Juego Tangram

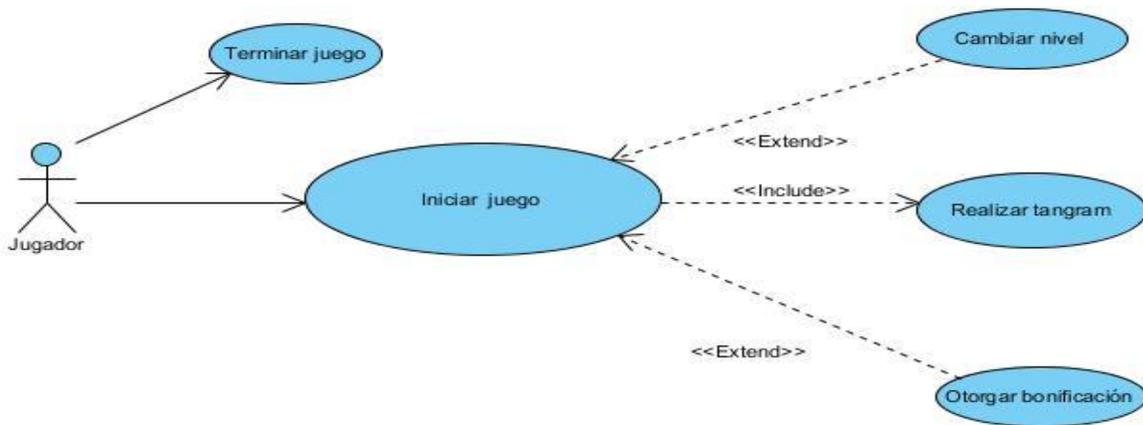


Figura 8 : Diagrama de casos de uso del juego Tangram

## *Capítulo 2: Características del sistema*

**Tabla 4: Descripción del caso de uso Iniciar juego del juego Tangram**

<b>Caso de uso</b>	Iniciar juego	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia una vez que el jugador selecciona el juego. El sistema carga el juego, muestra el comic inicial y la pantalla principal del juego. Si el jugador selecciona la opción salir el sistema salva los datos y termina el juego, finalizando así el caso de uso.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El jugador se encuentre autenticado en el sistema.	
<b>Postcondiciones</b>	Se realizó el juego.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona el juego.	<p>1.1 El sistema verifica el jugador (es) y brinda las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Si el jugador ha jugado anteriormente el juego y no ha culminado todos los niveles muestra una interfaz con el nivel actual en que se encuentra y las siguientes opciones:  “Continuar”: ir al paso 1.1 de la sección 2  “Iniciar juego”.</li> <li>• Si hay más de un jugador autenticados en el sistema habilita el nivel multijugador: ir a la sección “Multijugador”.</li> </ul> <p>Si el jugador no ha jugado el sistema permite</p>

## *Capítulo 2: Características del sistema*

		comenzar desde el principio.
2.	Selecciona la opción “Comenzar desde el principio”.	<p>2.1 Muestra la historia del juego en forma de comic.</p> <p>2.2 Muestra una interfaz que permite al jugador seleccionar una de las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Jugar nivel de entrenamiento: ir a la sección: “Nivel de entrenamiento”.</li> <li>• Comenzar a jugar: ir a la sección: “iniciar juego”.</li> </ul>
<b>Sección 1 “Nivel de entrenamiento”: Flujo Normal de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción “Jugar nivel de entrenamiento”.	<p>1.1 Muestra un mensaje de bienvenida al nivel y la siguiente opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
2.	Selecciona la opción “Continuar”.	<p>2.1 Muestra el tangram a armar, la lupa y un mensaje explicando en que consiste el juego y la siguiente opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
3.	Selecciona la opción “Continuar”.	<p>3.1 Muestra un mensaje donde orienta al jugador que rote y luego mueva una figura hasta la imagen sombreada y la opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
4.	Selecciona la opción “Continuar”.	<p>4.1 Señala una figura y la posición que le corresponde en la imagen sombreada.</p>
5.	Rota y mueve la figura indicada.	<p>5.1 Comprueba que la posición donde se ubica la figura esté vacía.</p> <p>5.2 Verifica que la figura se posiciona en el lugar</p>

## *Capítulo 2: Características del sistema*

		<p>que le corresponde con el ángulo correcto.</p> <p>5.3 Muestra un mensaje que indica que se puede usar la bonificación lupa y la siguiente opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
6.	Selecciona la opción "Continuar".	6.1 Activa la lupa.
7.	Hace click en el icono lupa.	7.1 Realiza el flujo normal de eventos en el paso 5.3.
8.	Mueve la figura indicada.	<p>8.1 Realiza el flujo normal de eventos en el paso 5.1 y 5.2.</p> <p>8.2 Muestra un mensaje indicando que se debe mover la última figura y la opción:</p> <ul style="list-style-type: none"> <li>• Continuar.</li> </ul>
9.	Selecciona la opción "Continuar".	<p>9.1 Cuando termina de mover todas las figuras muestra un mensaje anunciando que se completó el nivel y la opción:</p> <ul style="list-style-type: none"> <li>• Ir al juego.</li> </ul>
10.	Selecciona la opción "Ir al juego".	<p>10.1 Muestra el primer nivel y se comenzará a ejecutar el flujo normal de eventos en la sección 2 "Iniciar juego" en el paso 1.1.</p> <p>10.2 Termina el caso de uso.</p>
<b>Sección 2 "Iniciar juego": Flujo Normal de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción "Comenzar a jugar".	<p>1.1 Muestra el tangram que debe completar el jugador. (Ver CU Realizar tangram)</p> <p>1.2 Muestra al jugador el momento en que pasa de nivel. (Ver CU Cambiar nivel)</p> <p>1.3 Muestra cuando un jugador obtiene una</p>

## *Capítulo 2: Características del sistema*

		<p>bonificación. (Ver CU Otorgar bonificación)</p> <p>1.4 Muestra el tiempo límite de cada nivel.</p> <p>1.5 Muestra la puntuación del jugador.</p>
2.	Selecciona la opción "Salir del juego".	<p>2.1 Almacena los datos del jugador una vez concluido el juego.</p> <p>2.2 Termina el caso de uso.</p>
<b>Sección 3 "Multijugador": Flujo Normal de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		<p>1.1 Muestra un mensaje de bienvenida al modo retador y la opción:</p> <ul style="list-style-type: none"> <li>• Jugar.</li> </ul>
2	Selecciona la opción Jugar.	<p>2.1 Muestra el tangram a armar.</p> <p>2.2 Muestra el jugador en turno.</p> <p>2.3 Muestra el contador de tiempo.</p>
3.	Mueve una figura.	<p>3.1 Comprueba que la posición donde se ubica la figura esté vacía.</p> <p>3.2 Verifica que la figura se posicione en el lugar que le corresponde con el ángulo correcto.</p> <p>3.3 Si la figura es colocada en una parte del espacio sombreado que no le corresponde regresa a su posición anterior.</p>
4.	Rota una figura.	<p>4.1 Permite rotar la figura con un ángulo de 45 grados utilizando el doble click.</p>
5.	Completa el tangram.	<p>5.1 Cuando todas las figuras estén ubicadas correctamente se comenzará a ejecutar nuevamente el flujo básico de eventos desde el paso 2.1.</p> <p>5.2 Cuando terminen de jugar todos los jugadores</p>

## *Capítulo 2: Características del sistema*

		<p>se mostrará los resultados siguientes:</p> <ul style="list-style-type: none"> <li>• Nombre de cada jugador.</li> <li>• Puntos obtenidos.</li> <li>• Lugar alcanzado.</li> </ul> <p>5.3 Muestra las opciones siguientes:</p> <ul style="list-style-type: none"> <li>• Reiniciar.</li> <li>• Salir.</li> </ul>
6.	Selecciona la opción "Reiniciar".	<p>6.1 Se ejecuta el flujo básico de eventos en el paso 1.1.</p> <p>6.2 Termina el caso de uso.</p>
<b>Flujo alterno "Salir"</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción "Salir".	<p>1.1 Sale del juego.</p> <p>1.2 Termina el caso de uso.</p>
<b>Relaciones</b>	<b>CU Incluidos</b>	Realizar tangram
	<b>CU Extendidos</b>	<p>Otorgar bonificación</p> <p>Cambiar nivel</p>

### **Conclusiones del capítulo**

A partir de la descripción de las características de los juegos Rompecabezas, Puzle, Tangram y Tetravex se definieron los principales conceptos del dominio del problema brindando una mejor comprensión de la propuesta de solución a implementar. El levantamiento de requisitos funcionales y no funcionales propició la obtención de los diagramas de caso de uso y estableció las restricciones que se deben tener en cuenta a la hora de desarrollar los juegos. Las descripciones de los casos de uso sentaron las bases para realizar el modelo de diseño del sistema.

### Capítulo 3: Diseño del sistema

#### Introducción

En el presente capítulo se realizan los diagramas de interacción los cuales muestran gráficamente como los objetos se comunican entre ellos. Se describen las características y ventajas del patrón arquitectónico utilizado y se define el estándar de codificación a tener en cuenta en la implementación de los juegos con el propósito de obtener un estilo de programación uniforme y comprensible. Además se representa una vista interna de los juegos a desarrollar a partir de los diagramas de clases del diseño.

#### 3.1 Patrón arquitectónico

Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor. (51)

##### 3.1.1 Modelo Vista Controlador (MVC)

El MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (Modelo, Vista y Controlador). El modelo es el responsable de definir las reglas del negocio y llevar un registro de las vistas y controladores del sistema. La vista es la encargada de recibir datos del modelo y mostrarlos al usuario. El controlador es la capa que controla todo lo que se puede realizar en una aplicación. Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. (52)

Ventajas del patrón arquitectónico MVC:

- Clara separación entre interfaz, lógica de negocio y de presentación.
- Sencillez para crear distintas representaciones de los mismos datos.
- Reutilización de los componentes.
- Simplicidad en el mantenimiento de los sistemas.
- Facilidad para desarrollar prototipos rápidos.
- Los desarrollos suelen ser más escalables.

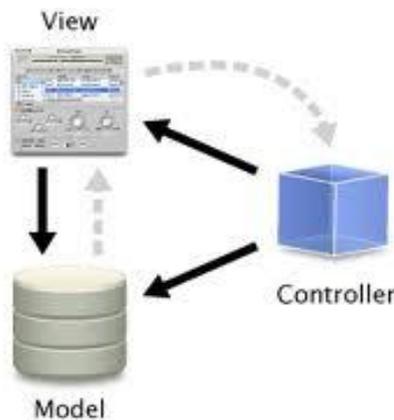


Figura 9: Patrón arquitectónico Modelo Vista Controlador

### 3.2 Modelo de diseño

Antes de pasar a las particularidades del diseño de la presente investigación, se hace necesario justificar por qué no se realiza el análisis. Primeramente, se tiene el hecho de que en el proyecto MundoClick donde se desarrolla la propuesta de solución dispone de un equipo de trabajo pequeño y con conocimientos de informática por lo que el proceso de captura de requisitos fue explícito. En la metodología RUP durante el análisis se obtiene una comprensión más precisa de dichos requisitos para poder estructurar el software entero; (50) por lo que se decidió ofrecer una visión general de los juegos a través de los resultados del diseño y la implementación.

El modelo de diseño, es una abstracción de la implementación del sistema que describe la realización física de los casos de uso, centrado en cómo los requisitos funcionales y no funcionales junto con otras restricciones relacionadas con el entorno de desarrollo, tienen impacto en las funcionalidades que se desean implementar. (50)

#### 3.2.1 Diagramas de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. (24)

Normalmente contienen la siguiente información: (24)

- Clases, asociaciones y atributos.

## *Capítulo 3: Diseño del sistema*

---

- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de los atributos.
- Navegabilidad.
- Dependencias.

A continuación se muestran los diagramas de clases del diseño de los juegos Rompecabezas y Puzle. Los restantes diagramas de clases se encuentran en los anexos.

# Capítulo 3: Diseño del sistema

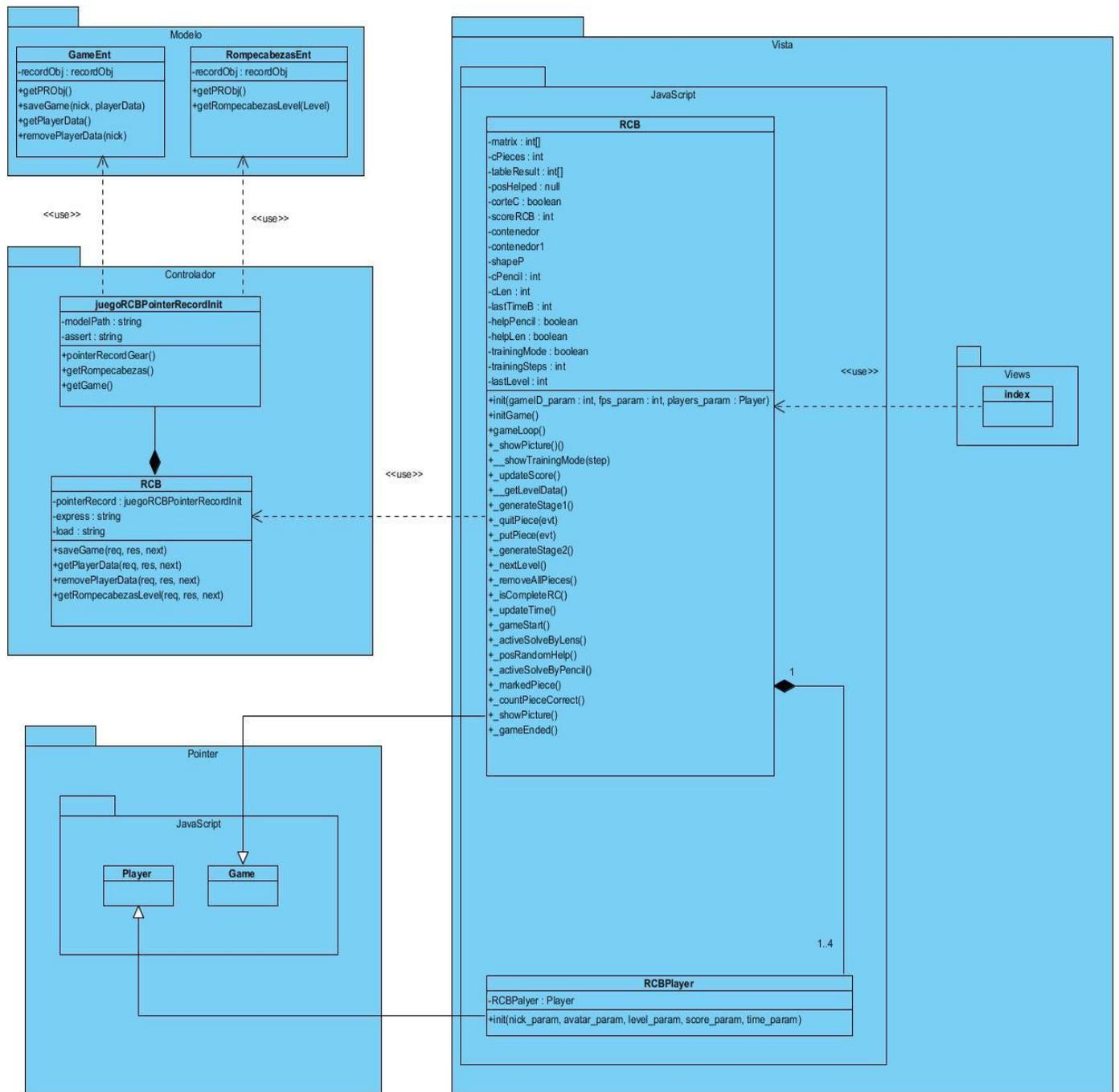


Figura 10: Diagrama de clases del juego Rompecabezas

## Capítulo 3: Diseño del sistema

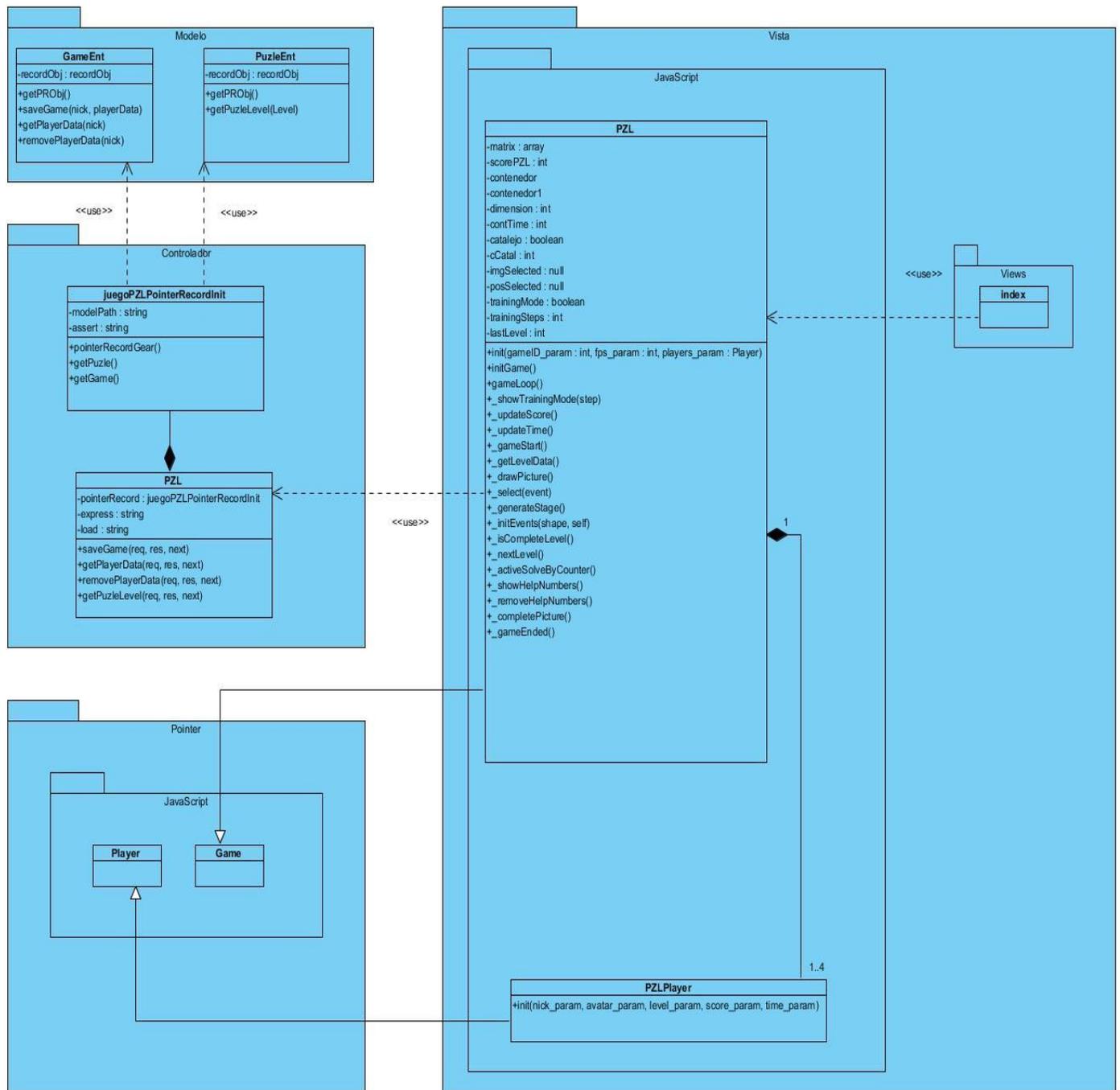


Figura 10: Diagrama de clases del juego Puzle

A continuación se describen las clases encargadas de manejar las principales funcionalidades de los juegos Rompecabezas y Puzle. Además se especifican los atributos y el tipo de dato

## Capítulo 3: Diseño del sistema

para cada uno de ellos y el nombre de cada uno de los métodos con sus parámetros correspondientes. Las restantes descripciones de los diagramas de clases se encuentran en los anexos.

**Tabla 5: Descripción de la clase RCB del juego Rompecabezas**

<b>Nombre</b>	RCB
<b>Tipo de clase</b>	Vista
<b>Atributo</b>	<b>Tipo de dato</b>
matrix	int[]
tableResult	int[]
cPieces	int
posHelped	null
corteC	boolean
scoreRCB	int
lastTimeB	int
contenedor	–
contenedor1	–
shapeP	–
helpPencil	boolean
cPencil	int
helpLen	boolean
cLen	int
asigToolLen	boolean
trainingMode	boolean
trainingSteps	int

## *Capítulo 3: Diseño del sistema*

---

lastLevel	int
Funcionalidades	
Nombre	Descripción
init(gameID_param: int, fps_param :int, players_param: Player)	Inicializa los atributos.
initGame()	Inicializa el juego.
gameLoop()	Actualiza el escenario, los puntos y el tiempo.
_showTrainingMode (step)	Se encarga del nivel de entrenamiento. Se le pasa por parámetro el paso del tutorial.
_updateScore()	Actualiza los puntos.
_updateTime()	Actualiza el tiempo.
_gameStart()	Inicia el nivel.
_getLevelData()	Extrae los datos del json.
_generateStage2()	Muestra las piezas del rompecabeza y se añaden los eventos de presionar, arrastrar y soltar las mismas.
_putPiece(evt)	Permite mover una pieza desde el lugar que se encuentra hasta la zona establecida.
_quitPiece(evt)	Permite mover una pieza ubicada dentro de la zona marcada a cualquier lugar dentro de la misma.
_generateStage1()	Muestra la zona marcada para armar el rompecabeza.
_nextLevel ()	Cambia el nivel.
_removeAllPieces()	Elimina todas las piezas de la zona marcada para

## Capítulo 3: Diseño del sistema

---

	armar el rompecabeza.
_activeSolveByLens ()	Permite utilizar el bono lupa.
_posRandomHelp()	Selecciona una pieza que no ha sido colocada en la zona marcada.
_markedPiece()	Comprueba que se ha colocado una pieza correcta en la zona marcada para asignar la cantidad de bonificaciones de lupa.
_countPieceCorrect()	Retorna el número de piezas correctas ubicadas en la zona marcada.
_activeSolveByPencil ()	Permite utilizar el bono pincel.
_showPicture()	Muestra la imagen original del rompecabeza.
_isCompleteRC()	Verifica si el rompecabeza está completo.
_gameEnded()	Finaliza el juego.

## Capítulo 3: Diseño del sistema

Tabla 6: Descripción de la clase PZL del juego Puzle

<b>Nombre</b>	PZL
<b>Tipo de clase</b>	Vista
<b>Atributo</b>	<b>Tipo de dato</b>
matrix	int[]
scorePZL	int
contenedor	int
contenedor1	int
dimesion	int
contTime	int
catalejo	boolean
cCatal	int
helpCatal	boolean
imgSelected	null
posSelected	null
trainingMode	boolean
trainingSteps	int
lastLevel	int
<b>Funcionalidades</b>	
<b>Nombre</b>	<b>Descripción</b>
init(gameID_param: int, fps_param :int, players_param: Player)	Inicializa los atributos.
initGame()	Inicializa el juego.
gameLoop()	Actualiza el escenario, los puntos y el tiempo.

## Capítulo 3: Diseño del sistema

---

_trainingLevel(step)	Se encarga del nivel de entrenamiento. Se le pasa por parámetro el paso del tutorial.
_updateScore()	Actualiza los puntos.
_updateTime()	Actualiza el tiempo.
_gameLevel()	Controla el juego.
_getLevelData ()	Extrae los datos del json.
_drawPicture()	Pinta la imagen original que se debe completar.
_generateStage ()	Pinta el área del puzle, las piezas sobre esa área y adiciona los eventos de ambas.
_initEvents (shape, self)	Inicializa los eventos de las piezas y de la zona donde aparecen las mismas.
_select(event)	Comprueba que al dar click sobre la pieza que desea mover exista un espacio en blanco adyacente a ésta para que pueda desplazarse hacia el mismo.
_nextLevel ()	Cambia el nivel.
_activeSolveByCounter()	Permite utilizar el bono catalejo.
_showHelpNumbers()	Enumera las piezas con la posición que ocupa en el puzle de izquierda a derecha y de arriba hacia abajo.
_removeHelpNumbers()	Elimina los números de la posición donde se encontraba la pieza antes de moverla hacia el espacio en blanco.
_completePicture()	Muestra la última pieza del puzle cuando se terminan de ordenar.
_isCompleteLevel()	Verifica si el puzle está ordenado.

_gameEnded()	Finaliza el juego.
--------------	--------------------

### 3.2.2 Diagramas de interacción

Un diagrama de interacción explica gráficamente las interacciones existentes entre las instancias de las clases del modelo. Constituye uno de los artefactos más importantes que se generan en el análisis y diseño orientados a objetos. El UML define dos tipos de estos diagramas; ambos sirven para expresar interacciones semejantes o idénticas de mensaje: (24)

- Diagramas de colaboración: describen las interacciones entre los objetos en un formato de grafo o red.
- Diagramas de secuencia: describen las interacciones en una especie de formato de cerca o muro.

En el diseño, es preferible representar la secuencia de acciones con diagramas de secuencia ya que el centro de atención principal es encontrar secuencias de interacciones detalladas y ordenadas en el tiempo. (50)

A continuación se muestran los diagramas de secuencia de los juegos Rompecabezas y Puzle. Los restantes diagramas de secuencia se encuentran en los anexos.

# Capítulo 3: Diseño del sistema

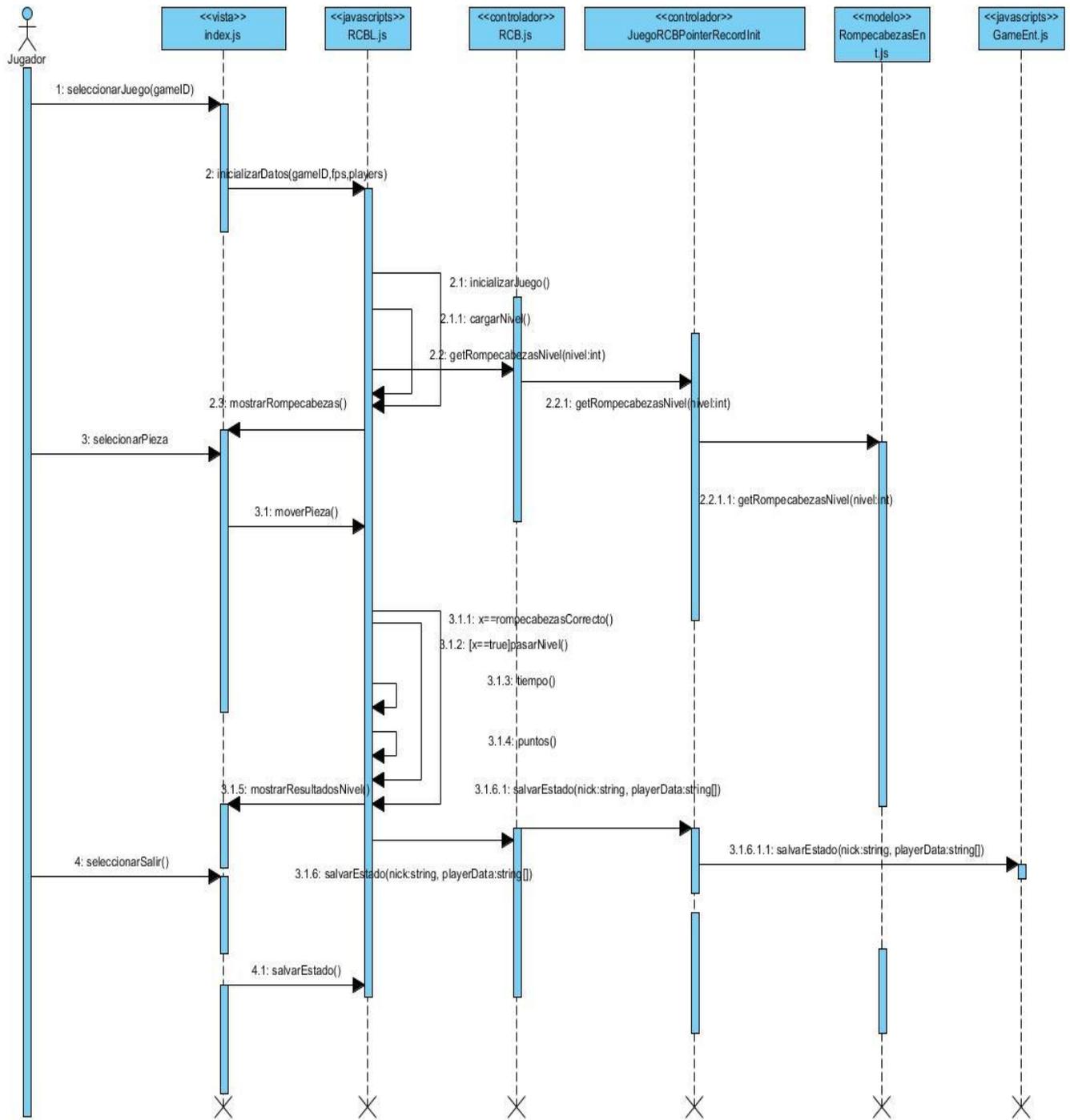


Figura 11 Diagrama de secuencia del juego Rompecabezas

# Capítulo 3: Diseño del sistema

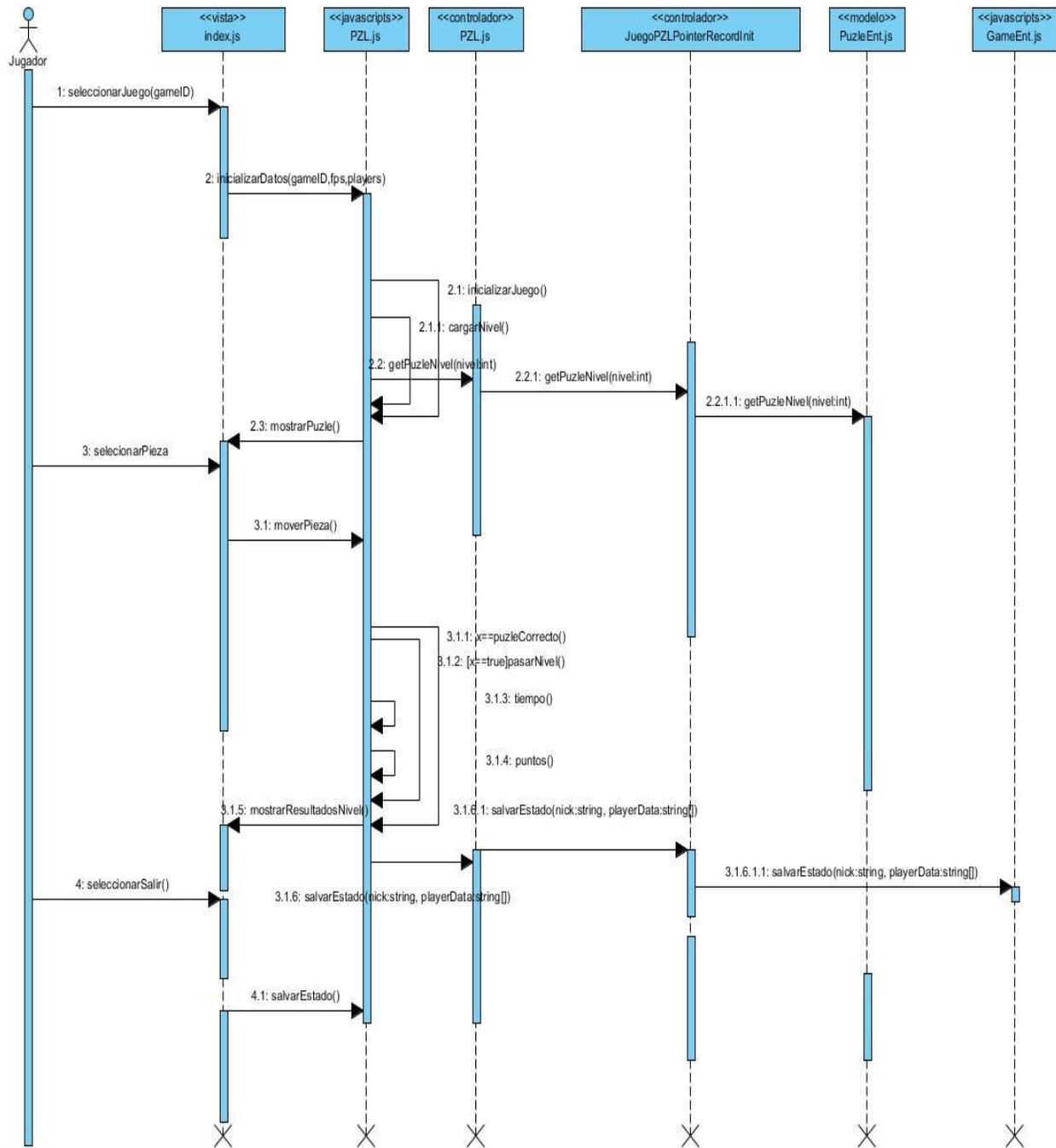


Figura 12: Diagrama de secuencia del juego Puzzle

### 3.3 Patrones GRASP

Un patrón es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. (24)

Los patrones *General Responsibility Assignment Software Patterns* traducido al español como: Patrones Generales de Software para Asignar Responsabilidades (GRASP) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. (24)

**Experto:** Es un patrón que se usa al asignar responsabilidades. Una ventaja de este patrón consiste en conservar el encapsulamiento ya que los objetos se valen de su propia información para hacer lo que se les pide. (24) Se manifiesta en los todos los juegos, por ejemplo en el Puzle y el Rompecabezas las clases PZL y RCB son las que se especializan en realizar las funcionalidades del juego; contienen la información necesaria para llevar a cabo sus tareas.

**Creador:** Es un patrón que guía la asignación de responsabilidades relacionadas con la creación de objetos (24). Se utiliza en los juegos Tetravex y Tangram en las clases TVX y TGM que se encargan de crear una instancia de las clases TVXFigure y TGMFigure para manipular sus datos.

**Controlador:** Es el responsable de controlar o manejar los eventos del sistema. (24). Se evidencia en las clases controladoras PZL, RCB, TVX y TGM que funcionan como intermediarias entre la vista y el modelo.

### 3.4 Estilos de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código rutinarias. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación posibilitan que el software se convierta en un sistema fácil de comprender y de mantener. (53)

## Capítulo 3: Diseño del sistema

Teniendo en cuenta lo importante que es utilizar un mismo estándar de codificación entre un equipo de desarrollo, el proyecto MundoClick al cual pertenece la solución propuesta acordó como estándares de codificación CamelCase e idiomatic.js.

CamelCase es la práctica de escribir frases o palabras compuestas eliminando los espacios y poniendo en mayúscula la primera letra de cada palabra. Existen dos tipos de CamelCase: (54)

- *LowerCamelCase*: la primera letra de la palabra está en minúscula. Por ejemplo, la frase "generar datos aleatorios" en *lowerCamelCase* sería "generarDatosAleatorios".
- *UpperCamelCase*: la primera letra de la palabra en mayúscula. Por ejemplo, la frase "generar datos aleatorios" en *upperCamelCase* sería "GenerarDatosAleatorios".

Para los nombres de las clases de los juegos a implementar se utilizará *upperCamelCase* y para los nombres de las variables y los métodos *lowerCamelCase*.

```
var RCB;  
RCB = ring.create([Game], {  
  /** xxx
```

Figura 13: Ejemplo sobre uso de *upperCamelCase*

```
  _generateStage: function() {  
    var self = this;  
    var conti = 0;  
    var dx = 600;  
    var dy = 50;  
  
    for (var i = 0; i < self.matrix.length; i++) {  
      var shape = new createjs.Shape();  
      shape.graphics.beginFill("rgba(0,0,0,0.3)").drawRect(0, 0, 60, 60);  
      shape.name = "b" + i;  
      shape.x = dx;  
      shape.y = dy;  
      shape.figure = null;  
  
      self.contenedor1.addChild(shape);  
  
      conti++;  
      dx += 60;  
      if (conti == self.dimension) {  
        conti = 0;  
        dy += 60;  
        dx = 600;  
      }  
    }  
    self.stage.addChild(self.contenedor1);  
    self.stage.update();  
  },  
  _nextLevel: function() { ■  
  },  
  _removeAllPieces: function() { ■  
  },  
}
```

Figura 14: Ejemplo del uso de *lowerCamelCase*

idiomatic.js es un manual que recoge la opinión de varios autores. En el mismo se mencionan una serie de consejos para lograr buenas prácticas de programación específicamente en el lenguaje JavaScript. Algunas de las recomendaciones son las siguientes: (55)

- Nunca mezclar espacios y tabulaciones.

## *Capítulo 3: Diseño del sistema*

---

- Eliminar el espacio en blanco del fin de línea.
- Eliminar el espacio en blanco del fin de las líneas vacías.

### **Conclusiones**

En el capítulo fueron descritos los elementos principales del diseño de los juegos: Rompecabezas, Puzle, Tangram y Tetravex. Además se obtuvieron los diagramas de clases del diseño y los diagramas de secuencia pertenecientes a la propuesta de solución los cuales constituyen una entrada apropiada y un punto de partida para la implementación. La utilización de los patrones arquitectónicos Modelo Vista Controlador y de diseño GRASP y los estilos de codificación garantizaron las buenas prácticas de programación.

## Capítulo 4: Implementación y pruebas

### Introducción

Los artefactos generados en el diseño constituyen la entrada principal en el flujo de trabajo de implementación. En el presente capítulo se realiza el diagrama de componentes con el objetivo de visualizar la estructura general del sistema. Posteriormente se realizan las pruebas de software con el propósito de detectar errores en la aplicación y verificar que los requisitos han sido implementados correctamente.

### 4.1 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. El mismo representa como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en los lenguajes de programación utilizados y como dependen los componentes uno de otros. (50)

#### 4.1.1 Diagrama de componentes

Los diagramas de componentes muestran los elementos de diseño de un sistema de software. Un diagrama de componentes permite visualizar con facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. (56)

A continuación se muestran los diagramas de componentes de los juegos: Rompecabezas, Puzle, Tangram y TetraVex y la descripción de los principales elementos que los conforman.

**Paquete views:** Paquete que agrupa todos los componentes relacionados con la vista del juego.

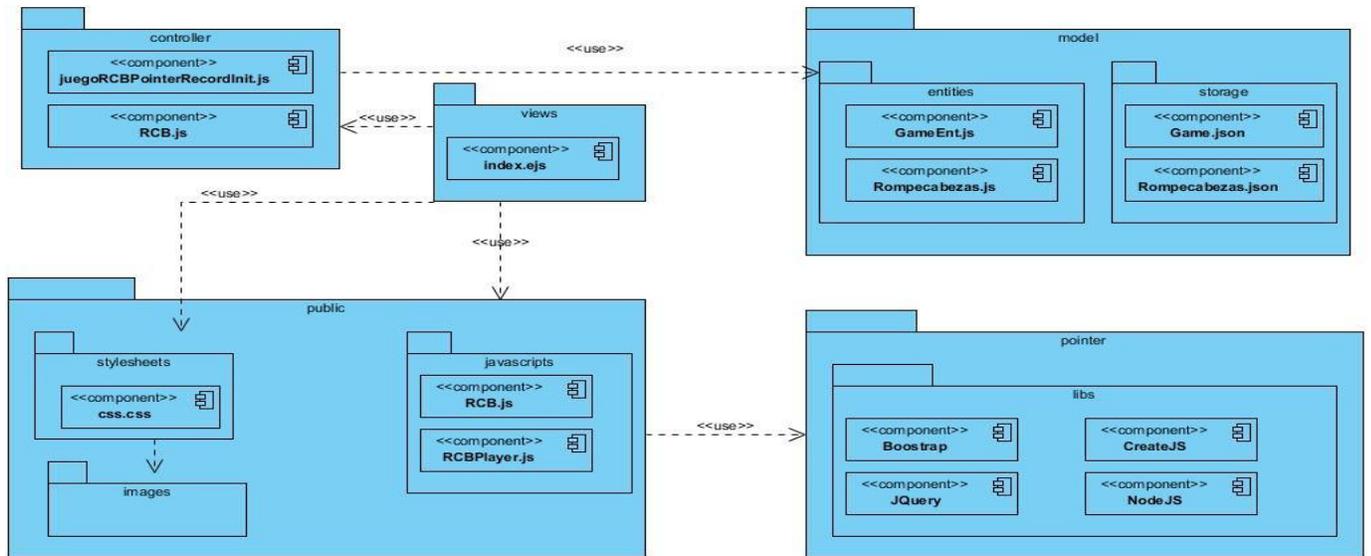
**Paquete controller:** Contiene las clases que acceden a los datos que se encuentran en el modelo y le envían la información a las vistas.

**Paquete model:** Agrupa las clases que contienen la información sobre el juego.

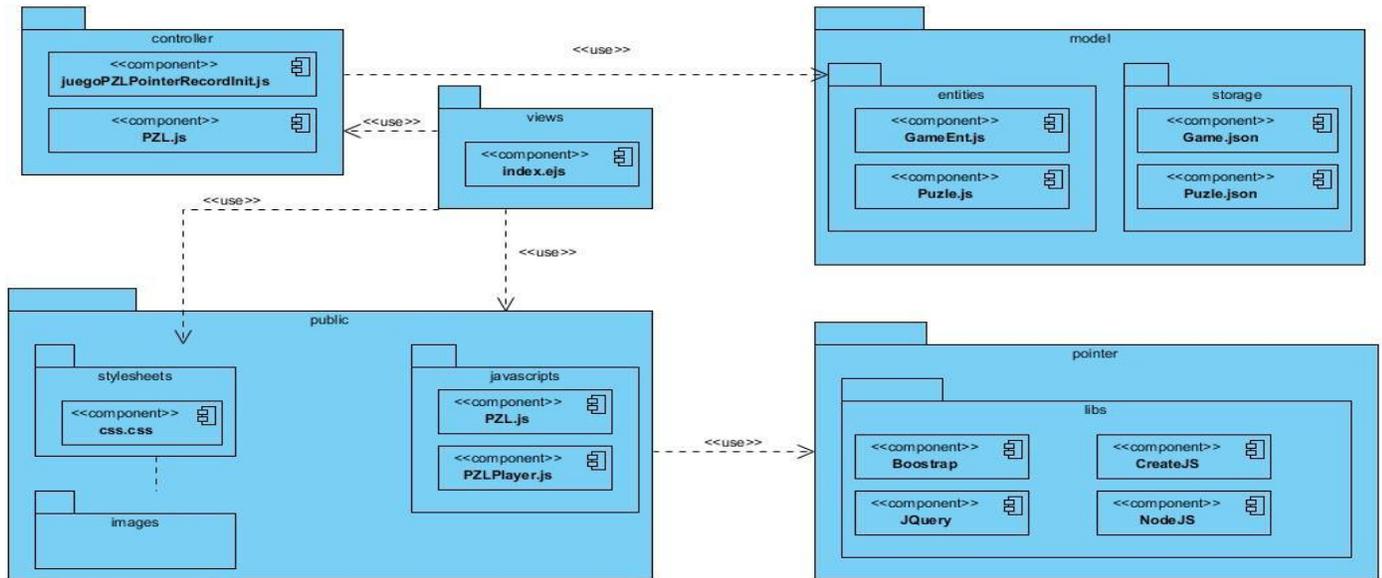
**Paquete public:** Contiene las clases que manejan la lógica del juego, almacena los recursos del mismo y la clase que se encarga de la correcta visualización del contenido de las vistas.

# Capítulo 4: Implementación y pruebas

**Paquete pointer:** Contiene las librerías utilizadas para la implementación de los juegos.



**Figura 15: Diagrama de componentes del juego Rompecabezas**



**Figura 16: Diagrama de componentes del juego Puzle**

# Capítulo 4: Implementación y pruebas

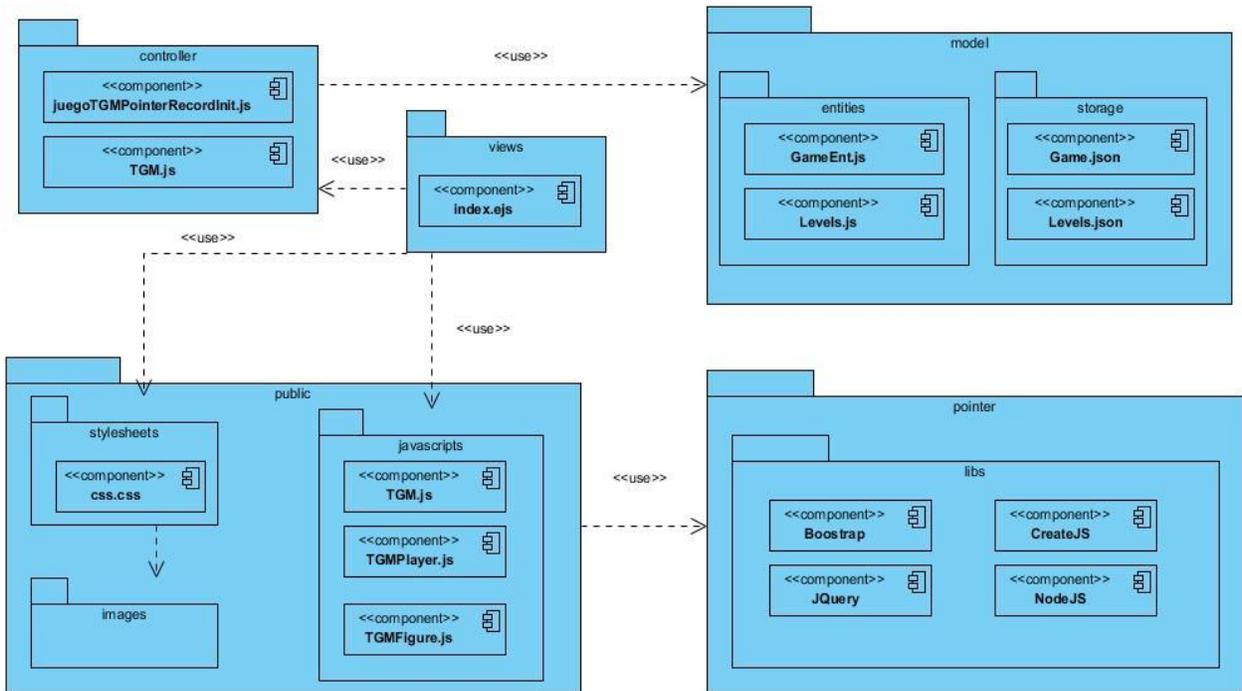


Figura 17: Diagrama de componentes del juego Tangram

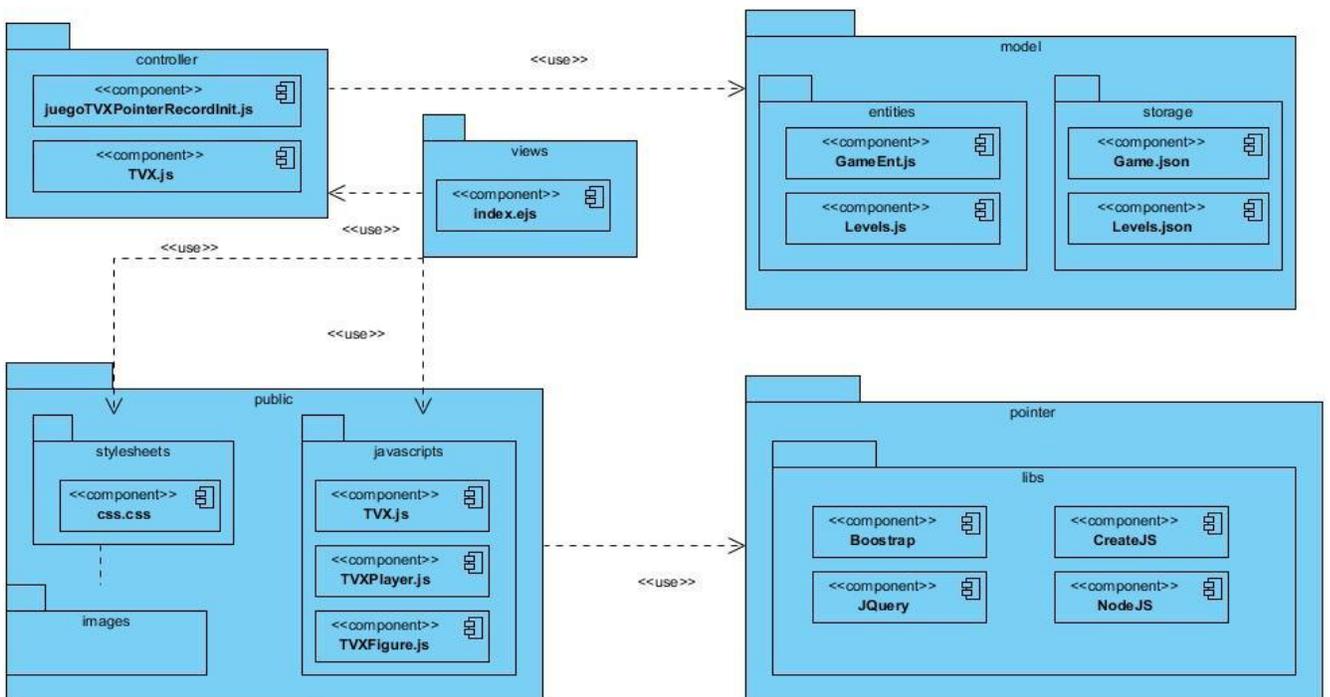


Figura 18: Diagrama de componentes del juego Tetravex

### 4.2 Prueba

Las pruebas del software constituyen un elemento crítico para la garantía de la calidad del software. Las mismas representan una revisión final de las especificaciones, el diseño y la codificación. (57)

Los objetivos de la etapa de pruebas son: (50)

- Planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de integración y las pruebas de sistema.
- Diseñar e implementar las pruebas creando los casos de pruebas que especifican qué probar.
- Realizar las diferentes pruebas y manejar los resultados de cada prueba sistemáticamente.

Una estrategia de prueba del software integra las técnicas de diseño de los casos de pruebas en una serie de pasos bien planificados que dan como resultado una correcta estructuración del sistema. (57) En la colección MundoClick se definieron niveles, tipos de pruebas, métodos y técnicas para comprobar que todos los requisitos fuesen implementados. A continuación se muestran los resultados del análisis realizado teniendo en cuenta las características específicas de la estrategia trazada.

#### 4.2.1 Niveles de prueba

**Prueba de integración:** Es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. (57)

**Prueba de sistema:** Está constituida por una serie de pruebas diferentes, las cuales verifican que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas. En este nivel existen variedades de pruebas entre las que se encuentran las de recuperación, seguridad, resistencia y rendimiento. (57)

### 4.2.2 Tipos de pruebas

**Pruebas de funcionalidad:** Se realiza con el propósito de verificar el cumplimiento de los requisitos funcionales, incluyendo el procesamiento y obtención de resultados.

**Pruebas de regresión:** Verifican que los cambios realizados no han introducido nuevos defectos y que el resto de la aplicación sigue funcionando correctamente.

### 4.2.3 Métodos de pruebas

Las pruebas de caja negra también denominada prueba de comportamiento se centran en los requisitos funcionales del software. Las mismas intentan encontrar errores de las siguientes categorías: (57)

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

La partición equivalente es una técnica de prueba de caja negra que divide el campo de entrada de un programa en clases de datos a partir de los que se pueden derivar casos de prueba. Este método se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse. (57)

### 4.2.4 Casos de prueba

Un caso de prueba específica una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de verificar y las condiciones bajo las que ha de probarse. (50)

A continuación se muestran los casos de prueba Realizar puzle del juego Puzle y Realizar tetravex del juego Tetravex.

## Capítulo 4: Implementación y pruebas

**Tabla 7: Caso de prueba Realizar puzle del juego Puzle**

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar puzle.	Selecciona la opción "Comenzar a jugar".	Muestra el puzle y la imagen que se debe armar.	Click en la opción "Comenzar a jugar".
EC 1.2 Mover pieza.	Mueve una pieza del puzle.	a) Verifica que exista un espacio vacío al lado de la pieza.  b) Verifica que cada pieza ocupe la posición que le corresponde dentro de la matriz.	Click sobre una pieza.
EC 1.3 Completar puzle.	Termina de mover todas las piezas.	Cuando todas las piezas se encuentran en el lugar que le corresponde cambia de nivel (SC Cambiar nivel).	Click sobre la última pieza.

**Tabla 8: Caso de prueba Realizar tetravex del juego Tetravex**

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar tetravex.	Selecciona la opción "Comenzar a jugar".	Muestra el tetravex.	Click en la opción "Comenzar a jugar".
EC 1.2 Mover figura.	Selecciona una figura, la arrastra y la suelta en el espacio de solución.	a) Comprueba que la posición donde se ubica la figura dentro del espacio de solución esté vacía. b) Si la figura se ubica en una posición que se encuentra ocupada por otra figura, regresa a su posición anterior.	Click sobre la figura, la arrastra y la suelta en el espacio de solución.

## Capítulo 4: Implementación y pruebas

		c) Verifica que la figura se posicione de forma tal que solo los colores y los números iguales coincidan, en caso contrario regresa a su posición anterior.	
EC 1.3 Completar tetravex.	Termina de mover todas las figuras.	Cuando todas las figuras están ubicadas correctamente incrementa la puntuación y cambia de nivel (ver SC Cambiar nivel).	Click sobre la última figura, la arrastra y la suelta en la zona marcada.

### 4.2.5 Resultados de las pruebas

Las pruebas de integración y sistema fueron realizadas a los juegos con el objetivo de verificar la integración de cada componente y el funcionamiento del sistema como un todo. Según el tipo de prueba se seleccionó las pruebas funcionales y de regresión con el propósito de comprobar el cumplimiento de los requisitos funcionales y asegurar que al detectar una no conformidad no produzca efectos adversos en otras funcionalidades del juego cuando ésta sea corregida.

Las pruebas de caja negra son las que se llevan a cabo sobre la interfaz del software. A partir de los casos de prueba generados bajo el método de caja negra se encontraron un grupo de no conformidades que estuvieron fundamentalmente relacionadas con manejo incorrecto de los puntos, tratamiento errado de las bonificaciones, posicionamiento inadecuado de las piezas y figuras en el escenario y además errores ortográficos en los mensajes.

En la tabla siguiente se muestran las no conformidades detectadas por cada una de las iteraciones realizadas. Como se evidencia, fueron detectadas un grupo importante de no conformidades, que fueron solucionándose en cada iteración de prueba hasta disminuir el número de defectos en los juegos.

## Capítulo 4: Implementación y pruebas

---

Tabla 9: Resultados de las pruebas

Juego	Iteración 1	Iteración 2	Iteración 3
Rompecabezas	6	3	0
Puzle	8	2	0
Tetravex	12	6	0
Tangram	10	5	0

### Conclusiones

Es el capítulo se obtuvo el diagrama de componentes del sistema el cual permitió una estructura robusta y organizada de los juegos Rompecabezas, Puzle, Tangram y Tetravex. Se definió además una estrategia de prueba para verificar el funcionamiento del software. Las pruebas aplicadas a los juegos dieron lugar a realizar mejoras sustanciales en el diseño y las funcionalidades de los juegos.

### **Conclusiones generales**

El presente documento recoge los resultados de la investigación realizada con el objetivo de desarrollar los juegos educativos Rompecabezas, Puzle, Tetravex y Tangram para la colección MundoClick. En el proceso de implementación de la propuesta de solución el equipo de desarrollo arribó a las siguientes conclusiones:

- El estudio de los sistemas similares aportó un conjunto de elementos que fueron vinculados a los nuevos juegos implementados.
- La obtención de los artefactos ingenieriles, definidos en cada una de los flujos de trabajo de la metodología RUP, permitió la ejecución de un proceso de desarrollo bien documentado en cada una de sus fases.
- Como principal resultado se obtuvo la realización de los juegos Rompecabezas, Puzle, Tangram y Tetravex para la colección MundoClick.
- La estrategia de pruebas diseñada para los juegos permitió mejoras sustanciales en el diseño y las funcionalidades de los juegos.

### **Recomendaciones**

Se recomienda incorporar el diseño gráfico y los elementos pedagógicos definitivos a los juegos Rompecabezas, Puzle, Tangram y Tetravex.

### Referencias

1. **Expósito , Carlos .** *Conceptos generales de software.* La Habana : Pueblo y Educación, 2005.
2. *Desarrollo de Proyectos de Software Educativo a través de esquemas de trabajo.* **Piñero Pérez, Yadenis y otros.** 5, La Habana : s.n., 2008, Vol. 1. 2306-2495.
3. **Hernández León , Rolando Alfredo y Coello González, Sayda.** *EL paradigma cuantitativo de la investigación científica.* La Habana : s.n., 2002.
4. *Ambientes de la enseñanza-aprendizaje enriquecidos por computador.* **Galvis Panqueva, Álvaro.** 2, Colombia : Ediciones Uniandes, 1988, Vol. 1.
5. **Marqués, Pere.** *El software educativo.* Barcelona : s.n., 1999.
6. **Litwin, Edith y otros.** *Tecnología educativa. Política, historias, propuestas.* Buenos Aires : Ediciones Paidós, 1995.
7. **TURNER, J.** *Juegos Creativos: Actividades educativas para niños en el hogar preescolares y escolares.* Buenos Aires : Panamericana, 1978.
8. *Juegos Multiplayer:Juegos Colaborativos para la educación.* **Mendoza Barros , Patricia y Galvis Panqueva , Álvaro.** 2, Colombia : Uniandes - Lidie, 1998, Vol. 11.
9. **Carmona , Cristina y Millán , Eva.** Departamento de Lenguajes y Ciencias de la Computación. [En línea] 2009. [Citado el: 12 de enero de 2014.] [http://www.lcc.uma.es/contenidos/informacion\\_publicacion.action?id=1236](http://www.lcc.uma.es/contenidos/informacion_publicacion.action?id=1236).
10. *Diseño y desarrollo de un juego educativo para ordenador sobre enfermedades tropicales y salud internacional.* **Giménez Pardo, Consuelo y otros.** 2, España : s.n., 2011, Vol. 8. 1697-011X.
11. **Marqués, Pere .** Los videojuegos y sus posibilidades. [En línea] 2000. [Citado el: 08 de Enero de 2014.] <http://dewey.uab.es/pmarques/pravj.htm>.
12. **Álvarez de Zayas , Carlos.** La escuela en la vida. La Habana, Cuba : Pueblo y Educación, 1999.

13. *Didáctica de la Lectura*. **Santisteban, Ernan**. Málaga, España : s.n., 2011. 13 978- 84-694-5337-7.
14. **Ramírez Ramos, Johanna Elizabet y Martínez Castro, Jesennia Zuliana**. *Rincón lógico matemático para optimizar el desarrollo del pensamiento en los niños y las niñas*. Milagro, Ecuador : s.n., 2009.
15. **Alcalá Sánchez, Ángel**. Conamat. [En línea] [Citado el: 22 de febrero de 2014.] <http://www.conamat.edu.pe/upload/topicos/juegosdetipoestrategicos.pdf>.
16. **García, Luis Francisco**. *Desarrollo de habilidades espaciales a través del uso de materiales concretos en niños de sexto grado de educación primaria*. México, D.F : s.n., 2007.
17. **Carspintrous, Luis**. *Lógica y procedimientos lógicos del pensamiento*. La Habana : s.n., 1993.
18. **Royo Molina , Víctor**. *Desarrollo del pensamiento lógico matemático a través de actividades lúdicas*. Córdoba, Colombia : s.n.
19. Reproducción Asistida. [En línea] [Citado el: 05 de junio de 2014.] <http://www.reproduccionasistida.org/los-puzzles-mejoran-las-habilidades-espaciales-en-ninos/>.
20. Genbeta. [En línea] [Citado el: 10 de diciembre de 2013.] <http://www.genbeta.com/herramientas/gcompris-software-libre-con-actividades-y-juegos-educativos-para-ninos>.
21. Gnome. [En línea] [Citado el: 23 de febrero de 2014.] <https://help.gnome.org/users/gnome-tetrevex/stable/>.
22. **Araujo Pérez, Liana Isabel y otros**. *“La caja mágica”, colección de software educativo multiplataforma*. La Habana, Cuba. : s.n.
23. **G. Figueroa, Roberth y otros**. Scribd. [En línea] [Citado el: 19 de febrero de 2014.] <http://es.scribd.com/doc/42572078/Articulo-Metodologia-de-Sw-Formato>.
24. **Larman, Graig**. *UML y Patrones. Introducción al análisis y diseño*. México : Prentice-Hall , 1999. 970-17-0261-1.
25. **Murillo, Jaibbo**. Slideshare. [En línea] 03 de Abril de 2011. [Citado el: 07 de Enero de 2014.] <http://www.slideshare.net/JaibooMurillo/herramientas-case-7495889#btnNext>.

26. Sitio de Descargas de Software. [En línea] [Citado el: 11 de diciembre de 2013.] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%28M%C3%8D%29\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/).
27. El Mundo Informático. [En línea] 2007. [Citado el: 05 de junio de 2014.] <http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>.
28. **Sanchez, Jesús** . *Lenguaje HTML*. s.l. : Grupo Eidos, 2001.
29. **Castillo Cantón, Alejandro** . Manual de HTML5 en español. [En línea] 03 de febrero de 2011. [Citado el: 20 de enero de 2014.] <http://es.scribd.com/doc/48124241/Manual-HTML-5-por-Alejandro-Castillo>.
30. **Pastorini, Andrés**. Facultad de Ingeniería. *HTML5 y CSS3*. [En línea] [Citado el: 10 de febrero de 2014.] <http://www.fing.edu.uy/tecnoinf/mvd/cursos/ria/material/teorico/ria-03-HTML5-CSS3.pdf>.
31. **Eguíluz Pérez, Javier**. *Introducción a CSS*. s.l. : Librosweb, 2008.
32. **Álvarez, Miguel Angel**. DesarrolloWeb. *Manual de CSS 3*. [En línea] 2011. [Citado el: 16 de febrero de 2014.] <http://www.desarrolloweb.com/manuales/css3.html>.
33. **Navajas Ojeda , Antonio**. *Guía completa de CSS3*. s.l. : Autoedición, 2012.
34. **Eguíluz Pérez , Javier**. *Introducción a JavaScript*. s.l. : Librosweb, 2008.
35. Developer. [En línea] [Citado el: 15 de enero de 2014.] [https://developer.mozilla.org/es/docs/Gu%C3%ADa\\_JavaScript\\_1.5/Concepto\\_de\\_JavaScript#C2.BFQu.C3.A9\\_es\\_JavaScript.3F](https://developer.mozilla.org/es/docs/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript#C2.BFQu.C3.A9_es_JavaScript.3F).
36. **Bradenbaugh, Jerry**. *Programación de Aplicaciones Web con JavaScript*. Madrid, España : ANAYA MULTIMEDIA, 1999. 84-41 5-1 070-9.
37. **Gutiérrez., Javier J**. Departamento de Lenguajes y Sistemas Informáticos. [En línea] [Citado el: 4 de junio de 2014.] [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf).
38. Genbetadev. [En línea] [Citado el: 15 de enero de 2014.] <http://www.genbetadev.com/frameworks/bootstrap>.

39. **Manzur, Ing. Sergio.** JQuery. [En línea] [Citado el: 24 de mayo de 2014.] <http://jquery.com/>.
40. **Álvarez, Miguel Ángel.** DesarrolloWeb. *Manual de jQuery*. [En línea] [Citado el: 7 de enero de 2014.] <http://www.desarrolloweb.com/manuales/manual-jquery.html>.
41. **Mir , Guillermo.** Securityartwork. [En línea] [Citado el: 4 de enero de 2014.] <http://www.securityartwork.es/2013/06/17/nuevas-tecnologias-viejas-vulnerabilidades-node-js-i/>.
42. **Gámez, Alejandro Morales y Álvarez, Miguel Angel .** *Introducción a NodeJS*. s.l., 2012.
43. HTML5facil. [En línea] [Citado el: 15 de enero de 2014.] <http://html5facil.com/tips/crea-aplicaciones-de-escritorio-multiplataforma-usando-html-css-y-javascript>.
44. Desarrolloweb. [En línea] [Citado el: 13 de enero de 2014.] [http://www.desarrolloweb.com/de\\_interes/createjs-herramientas-javascript-desarrolladores-7003.html](http://www.desarrolloweb.com/de_interes/createjs-herramientas-javascript-desarrolladores-7003.html).
45. antonionll. [En línea] 2013. [Citado el: 05 de junio de 2014.] <http://antonionll.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-o-ide/>.
46. Netbeans. [En línea] [Citado el: 20 de enero de 2014.] [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html).
47. **Serradilla, Juan Luis.** *Control de Versiones con Subversion y TortoiseSVN*.
48. **Henao Acosta, Carolina y Ortiz Villegaz, Juan Pablo.** *Guía de la ingeniería del software*. 2004.
49. *Tecnología y Synergix*. [En línea] [Citado el: 03 de marzo de 2014.] <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>.
50. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. Madrid : Pearson Educación, 2000. 84-7829-036-2.
51. Ingeniería del software. [En línea] [Citado el: 02 de abril de 2014.] <http://arlethparedes.wordpress.com/2012/08/27/patrones-de-arquitectura-vs-patrones-de-diseno/>.
52. Ingeniería del software. [En línea] [Citado el: 06 de abril de 2014.] <http://arlethparedes.wordpress.com/2012/09/23/el-patron-mvc-modelo-vista-controlador/>.

53. Developer Network. [En línea] 27 de abril de 2014. <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
54. [En línea] [Citado el: 27 de abril de 2014.] <http://beosman.org/docs/guia-estilo/>.
55. [En línea] [Citado el: 27 de abril de 2014.] <https://github.com/MatiasArriola/idiomatic.js>.
56. Developer Network. [En línea] [Citado el: 27 de abril de 2014.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
57. **Pressman, Roger.** *Ingeniería del software.Un enfoque práctico.* s.l. : McGraw-Hill, 2002. 8448132149 .

### **Glosario de términos**

**Metodología:** Parte de la lógica que estudia los métodos del conocimiento. Conjunto de métodos utilizados en la investigación científica.

**Multimedia:** Integración de soportes o procedimientos que emplean sonido, imágenes o textos para difundir información, especialmente si es de forma interactiva.

**Didáctico:** De la enseñanza, relacionado con ella o adecuado para ella. Área de la pedagogía que se ocupa de las técnicas y métodos de enseñanza.

**Software:** Término genérico que se aplica a los componentes no físicos de un sistema informático, como los programas y sistemas operativos.

**Arquitectura:** Término general que se aplica a la estructura de un sistema informático o de una parte del mismo.

**Multiplataforma:** Que tiene la capacidad de soportar múltiples sistemas operativos.

**Estándares:** Modelo o patrón.

**Patrones:** Modelos que sirven de muestra para obtener otros elementos iguales.

**Instancia:** Objeto referente a una clase que engloba sus atributos y operaciones.

## Anexos

### Anexo 1:

Tabla 10: Descripción del caso de uso Realizar rompecabezas del juego Rompecabezas

<b>Caso de uso</b>	Realizar rompecabezas	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador selecciona la opción “Comenzar a jugar”. Termina cuando haya completado todos los niveles del juego.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Se visualiza el rompecabezas a armar.	
<b>Postcondiciones</b>	Se completa el rompecabezas.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		1.1 Muestra el rompecabezas a armar una vez comenzado el juego.
2.	Mueve una pieza.	2.1 Comprueba que la posición donde se ubica la pieza dentro de la matriz esté vacía. 2.2 Si la pieza se ubica en una posición que se encuentra ocupada por otra pieza, regresa a su posición anterior. 2.3 Verifica que cada pieza ocupe el lugar que le corresponde dentro de la matriz.
3.	Completa el rompecabezas.	3.1 Cuando todas las piezas se encuentran en el lugar que le corresponde aumenta la puntuación y cambia el nivel (ver CU Cambiar nivel).
4.		4.1 Termina el caso de uso.

Anexo 2:

Tabla 11: Descripción del caso de uso Cambiar nivel del juego Rompecabezas

<b>Caso de uso</b>	Cambiar nivel	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador culmina un nivel satisfactoriamente. Termina cuando el jugador vence el último nivel.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Completar correctamente el rompecabezas.	
<b>Postcondiciones</b>	Cambió el nivel	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		<p>1.1 Muestra una interfaz con la opción “Aceptar” y los siguientes resultados:</p> <ul style="list-style-type: none"> <li>• Nivel culminado.</li> <li>• Puntuación obtenida.</li> <li>• Tiempo transcurrido.</li> <li>• Bonificaciones.</li> </ul>
2.	Selecciona la opción “Aceptar”.	<p>2.1 Guarda el estado del jugador:</p> <ul style="list-style-type: none"> <li>• Nivel en que se encuentra.</li> <li>• Puntuación obtenida hasta el momento.</li> </ul> <p>2.2 Muestra los datos del nuevo nivel (Ver CU Iniciar Juego, sección 2, paso 1.1).</p>
<b>Flujos alternos: “Último nivel”</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Completa el rompecabezas.	1.1 Muestra una interfaz felicitando al jugador

	por haber ganado el juego. 1.2 Termina el caso de uso.
--	---

**Anexo 3:**

**Tabla 12 : Descripción del caso de uso Otorgar bonificación del juego Rompecabezas**

<b>Caso de uso</b>	Otorgar bonificación	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador coloca la primera pieza correcta o al comenzar cada nivel. Termina cuando se otorga la bonificación.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Secundario	
<b>Precondiciones</b>	El jugador debe colocar la primera pieza correcta.	
<b>Postcondiciones</b>	Se adicionan las bonificaciones al juego.	
<b>Flujo normal de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		1.1 Premia al jugador con las siguientes bonificaciones: <ul style="list-style-type: none"> <li>• Lupa: ir a la sección 1 “Lupa”.</li> <li>• Pincel mágico: ir a la sección 2 “pincel mágico”.</li> </ul>
<b>Sección 1 “Lupa”: Flujo Normal de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Coloca la primera pieza correcta.	1.1 Bonifica al jugador con una lupa la cual señala una pieza y el lugar que ocupa dentro de la zona marcada. 1.2 Aumenta la cantidad de bonificaciones obtenidas hasta el momento.

		1.3 Muestra la bonificación obtenida en la parte inferior de la pantalla.
2.	Hace click sobre el icono "lupa".	2.1 Señala una pieza de forma aleatoria y el lugar donde debe colocarse.
3.		3.1 Termina el caso de uso.
<b>Sección 2 "Pincel mágico": Flujo Normal de Eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		1.1 Bonifica al jugador con un pincel mágico que muestra la imagen original que se debe completar. 1.2 Muestra la bonificación obtenida en la parte inferior de la pantalla.
2.	Hace click sobre el icono "pincel mágico".	2.1 Muestra la imagen.
3.		3.1 Termina el caso de uso.

**Anexo 4:**

**Tabla 13: Descripción del caso de uso Terminar juego del juego Rompecabezas**

<b>Caso de uso</b>	Terminar juego
<b>Actores</b>	Jugador
<b>Resumen</b>	El caso de uso inicia cuando el jugador selecciona la opción Salir del juego. Termina cuando se cierra el juego.
<b>Complejidad</b>	Media
<b>Prioridad</b>	Secundario
<b>Precondiciones</b>	El jugador debe encontrarse dentro del juego.
<b>Postcondiciones</b>	Se termina el juego.
<b>Flujo de eventos</b>	

	Actor	Sistema
1.	Selecciona la opción Salir del juego.	1.1 Muestra un mensaje de confirmación de la acción “¿Está seguro que desea salir?”.
2.	Selecciona la opción “Aceptar”.	2.1 Guarda los resultados del jugador.
3.		3.1 Terminar el caso de uso.
<b>Flujos alternos: “Cancelar salida del juego”</b>		
	Actor	Sistema
1.	Selecciona la opción “Cancelar” del mensaje mostrado.	1.1 Continúa en el juego.
2.		2.1 Termina el caso de uso.

**Anexo 5:**

**Tabla 14: Descripción del caso de uso Realizar puzle del juego Puzle**

<b>Caso de uso</b>	Realizar puzle	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador selecciona la opción “Comenzar a jugar”. Termina cuando haya completado todos los niveles del juego.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Se visualizan los puzles a armar.	
<b>Postcondiciones</b>	Se realizó el puzle.	
<b>Flujo básico Realizar puzle</b>		
	Actor	Sistema
1.		1.1 Muestra el puzle a armar una vez

		comenzado el juego. 1.2 Muestra la imagen que debe construir el jugador.
2.	Mueve una pieza.	2.1 Verifica que exista un espacio vacío al lado de la pieza. 2.2 Verifica que cada pieza ocupe la posición que le corresponde dentro de la matriz.
3.	Completa el puzle.	3.1 Cuando todas las piezas se encuentran en el lugar que le corresponde cambia de nivel (ver CU Cambiar nivel).
4.		4.1 Termina el caso de uso.

## Anexo 6:

Tabla 15: Descripción del caso de uso Cambiar nivel del juego Puzle

<b>Caso de uso</b>	Cambiar nivel	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador culmina un nivel satisfactoriamente. Termina cuando el jugador vence el último nivel.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Completar correctamente el puzle.	
<b>Postcondiciones</b>	Cambió el nivel.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>

1.		1.1 Muestra una interfaz con la opción "Aceptar" y los siguientes resultados: <ul style="list-style-type: none"> <li>• Nivel culminado.</li> <li>• Puntuación obtenida.</li> <li>• Tiempo transcurrido.</li> <li>• Bonificaciones.</li> </ul>
2.	Selecciona la opción "Aceptar".	2.1 Guarda el estado del jugador: <ul style="list-style-type: none"> <li>• Nivel en que se encuentra.</li> <li>• Puntuación obtenida hasta el momento.</li> </ul> 2.2 Muestra los datos del nuevo nivel (Ver CU Iniciar Juego, sección 2, paso 1.1).
<b>Flujos alternos: "Último nivel"</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Completa el puzle.	1.1 Muestra una interfaz felicitando al jugador cuando termina el juego.
2.		2.1 Termina el caso de uso.

## Anexo 7:

Tabla 16: Descripción del caso de uso Otorgar bonificación del juego Puzle

<b>Caso de uso</b>	Otorgar bonificación
<b>Actores</b>	Jugador
<b>Resumen</b>	El caso de uso inicia cuando comienza el nivel y termina cuando se enumeran las piezas.
<b>Complejidad</b>	Media
<b>Prioridad</b>	Secundario
<b>Precondiciones</b>	Iniciar nivel.
<b>Postcondiciones</b>	Se enumeran las piezas del puzle.
<b>Flujo de normal de eventos</b>	

	Actor	Sistema
1.		1.1 Bonifica al jugador con un catalejo el cual enumera las piezas del puzzle. 1.2 Muestra la bonificación obtenida en la parte inferior de la pantalla.
2.	Hace click sobre el icono "catalejo".	2.1 Muestra las piezas del puzzle enumeradas en el orden en que deben colocarse.
3		3.1 Termina el caso de uso.

**Anexo 8:**

**Tabla 17: Descripción del caso de uso Terminar juego del juego Puzzle**

<b>Caso de uso</b>	Terminar juego	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador selecciona la opción Salir del juego. Termina cuando se cierra el juego.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Secundario	
<b>Precondiciones</b>	El jugador debe encontrarse dentro del juego.	
<b>Postcondiciones</b>	Se termina el juego.	
<b>Flujo de eventos</b>		
	Actor	Sistema
1.	Selecciona la opción Salir del juego.	1.2 Muestra un mensaje de confirmación de la acción "¿Está seguro que desea salir?".
2.	Selecciona la opción "Aceptar".	2.1 Guarda los resultados del jugador.
3.		3.1 Terminar el caso de uso.
<b>Flujos alternos: "Cancelar salida del juego"</b>		

	Actor	Sistema
1.	Selecciona la opción "Cancelar" del mensaje mostrado.	1.1 Continúa en el juego.
2.		2.1 Termina el caso de uso.

**Anexo 9:**

**Tabla 18: Descripción del caso de uso Realizar tetravex del juego Tetravex**

<b>Caso de uso</b>	Realizar Tetravex	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador selecciona la opción: "Comenzar a jugar". Termina cuando haya completado todos los niveles que muestra el juego.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Se visualiza el tetravex a armar.	
<b>Postcondiciones</b>	Se completó el tetravex.	
<b>Flujo de eventos</b>		
	Actor	Sistema
1.		1.1 Muestra el tetravex a armar una vez comenzado el juego.
2.	Mueve una figura.	2.1 Comprueba que la posición donde se ubica la figura dentro del espacio de solución esté vacía. 2.2 Si la figura se ubica en una posición que se encuentra ocupada por otra figura, regresa a su posición anterior. 2.3 Verifica que la figura se posicione de

		forma tal que solo los colores y los números iguales coincidan, sino coincide la figura regresa a su posición.
3.	Completa el tetravex.	3.1 Cuando todas las figuras están ubicadas correctamente incrementa la puntuación. 3.2 Cambia de nivel (ver CU Cambiar nivel).
4.		4.1 Termina el caso de uso.

## Anexo 10:

Tabla 19: Descripción del caso de uso Cambiar de nivel del juego Tetravex

<b>Caso de uso</b>	Cambiar nivel	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador culmina un nivel satisfactoriamente. Termina cuando el jugador vence el último nivel.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Completar correctamente el tetravex.	
<b>Postcondiciones</b>	Cambió el nivel.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		1.1 Muestra una interfaz con la opción "Aceptar" y los siguientes resultados: <ul style="list-style-type: none"> <li>• Nivel culminado.</li> <li>• Puntuación obtenida.</li> <li>• Tiempo transcurrido.</li> <li>• Bonificaciones.</li> </ul>

2.	Selecciona la opción “Aceptar”.	<p>2.1 Guarda el estado del jugador:</p> <ul style="list-style-type: none"> <li>• Nivel en que se encuentra.</li> <li>• Puntuación obtenida hasta el momento.</li> </ul> <p>2.2 Muestra los datos del nuevo nivel (Ver CU Iniciar Juego, sección 2, paso 1.1).</p>
<b>Flujos alternos: “Último nivel”</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Completa el tetravex.	1.1 Muestra una interfaz felicitando al jugador por haber ganado el juego.
2.		2.1 Termina el caso de uso.

**Anexo 11:**

**Tabla 20: Descripción del caso de uso Otorgar bonificación del juego Tetravex**

<b>Caso de uso</b>	Otorgar bonificación	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador mueve la primera figura y termina cuando se le muestra al jugador la ubicación de una figura dentro del espacio de solución.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Secundario	
<b>Precondiciones</b>	Mover la primera figura.	
<b>Postcondiciones</b>	Se adiciona la bonificación “lupa” al juego.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		1.1 Bonifica al jugador con una “lupa” que le brinda una figura correcta dentro del espacio de solución.

		1.2 Aumenta la cantidad de bonificaciones obtenidas hasta el momento. 1.3 Muestra la bonificación obtenida en la parte inferior de la pantalla.
2.	Hace click sobre el icono “lupa”.	2.1 Señala una figura y el lugar que ocupa en el espacio de solución.
3.		3.1 Termina el caso de uso.

**Anexo 12:**

**Tabla 21: Descripción del caso de uso Terminar juego del juego Tetravex**

<b>Objetivo</b>	Salir del juego	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador selecciona la opción Salir del juego. Termina cuando se cierra el juego.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>		
<b>Precondiciones</b>	El jugador debe encontrarse dentro del juego.	
<b>Postcondiciones</b>	Se termina el juego.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción “Salir” del juego.	1.1 Muestra un mensaje de confirmación de la acción “¿Está seguro que desea salir?”.
2.	Selecciona la opción “Aceptar”.	2.1 Guarda los resultados del jugador.
3.		3.1 Terminar el caso de uso.
<b>Flujos alternos: “Cancelar salida del juego”</b>		
	<b>Actor</b>	<b>Sistema</b>

1.	Selecciona la opción ``Cancelar`` del mensaje mostrado.	1.1 Continúa en el juego.
2.		2.1 Termina el caso de uso.

**Anexo 13:**

**Tabla 22: Descripción del caso de uso Realizar tangram del juego Tangram**

<b>Caso de uso</b>	Realizar Tangram	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador selecciona la opción "Jugar". Termina cuando haya completado todos los niveles del juego.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Se visualiza el tangram a armar.	
<b>Postcondiciones</b>	Se completó todos los niveles.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		1.1 Muestra el tangram a armar.
2.	Mueve una figura.	2.1 Comprueba que la posición donde se ubica la figura esté vacía. 2.2 Si la figura es colocada en una parte del espacio sombreado que no le corresponde regresa a su posición anterior. 2.3 Verifica que la figura se posicione en el lugar que le corresponde con el ángulo correcto.
3.	Rota una figura.	3.1 Permite rotar la figura con un ángulo de

		45 grados utilizando el doble click.
4.	Completa el tangram de forma correcta antes del tiempo límite.	<p>4.1 Cuando todas las figuras están ubicadas correctamente incrementa la puntuación.</p> <p>4.2 Cambia de nivel (ver CU Cambiar nivel).</p> <p>4.3 Termina el caso de uso.</p>

**Anexo 14:**

**Tabla 23: Descripción del caso de uso Cambiar nivel del juego Tangram**

<b>Caso de uso</b>	Cambiar nivel	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador culmina un nivel satisfactoriamente. Termina cuando el jugador vence el último nivel.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Completar correctamente el tangram.	
<b>Postcondiciones</b>	Cambió el nivel.	
<b>Flujo de eventos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		<p>1.1 Muestra una interfaz con la opción “Aceptar” y los siguientes resultados:</p> <ul style="list-style-type: none"> <li>• Nivel culminado.</li> <li>• Puntuación obtenida.</li> <li>• Tiempo transcurrido.</li> <li>• Bonificaciones.</li> </ul>
2.	Selecciona la opción “Aceptar”.	<p>2.1 Guarda el estado del jugador:</p> <ul style="list-style-type: none"> <li>• Nivel en que se encuentra.</li> </ul>

		<ul style="list-style-type: none"> <li>Puntuación obtenida hasta el momento.</li> </ul> 2.2 Muestra los datos del nuevo nivel. (Ver CU Iniciar Juego, sección 2, paso 1.1)
<b>Flujos alternos: “Tiempo agotado”</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Llega al tiempo límite sin terminar de colocar todas las figuras.	1.1 Muestra una interfaz informando al jugador que su tiempo se ha agotado. 1.2 Brinda la opción de reiniciar el nivel.
2.	Selecciona la opción “Reiniciar”.	2.1 Muestra los datos del mismo nivel.(Ver CU Iniciar Juego, sección 2, paso 1.1) 2.2 Termina el caso de uso.
<b>Flujos alternos: “Último nivel”</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Completa el tangram.	1.1 Muestra una interfaz felicitando al jugador por haber ganado el juego.
2.		2.1 Termina el caso de uso.

**Anexo 15:**

**Tabla 24: Descripción del caso de uso Otorgar bonificación del juego Tangram**

<b>Caso de uso</b>	Otorgar bonificación
<b>Actores</b>	Jugador
<b>Resumen</b>	El caso de uso inicia cuando el jugador mueve una figura dentro de la imagen sombreada.
<b>Complejidad</b>	Media
<b>Prioridad</b>	Secundario
<b>Precondiciones</b>	Mover una figura.
<b>Postcondiciones</b>	Se adiciona la bonificación “lupa” al juego.
<b>Flujo de eventos</b>	

	Actor	Sistema
1.		1.1 Bonifica al jugador con una “lupa” la cual consiste señalar una figura y la posición que le corresponde en la imagen sombreada.  1.2 Muestra la bonificación obtenida en la parte inferior de la pantalla.
2.	Hace click sobre el icono “lupa”.	2.1 Señalar una figura y la posición que le corresponde en la imagen sombreada.
3.		3.1 Termina el caso de uso.

**Anexo 16:**

**Tabla 25: Descripción del caso de uso Terminar juego del juego Tangram**

<b>Caso de uso</b>	Salir del juego	
<b>Actores</b>	Jugador	
<b>Resumen</b>	El caso de uso inicia cuando el jugador selecciona la opción Salir del juego. Termina cuando se cierra el juego.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Secundario	
<b>Precondiciones</b>	El jugador debe encontrarse dentro del juego.	
<b>Postcondiciones</b>	Se termina el juego.	
<b>Flujo de normal eventos</b>		
	Actor	Sistema
1.	Selecciona la opción Salir del juego.	1.1 Muestra un mensaje de confirmación de la acción “¿Está seguro que desea salir?”.
2.	Selecciona la opción “Aceptar”.	2.1 Guarda los resultados del jugador.
3.		3.1 Terminar el caso de uso.

<b>Flujos alternos: "Cancelar salida del juego"</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción "Cancelar" del mensaje mostrado.	1.1 Continúa en el juego.
2.		2.1 Termina el caso de uso.

Anexo 17:

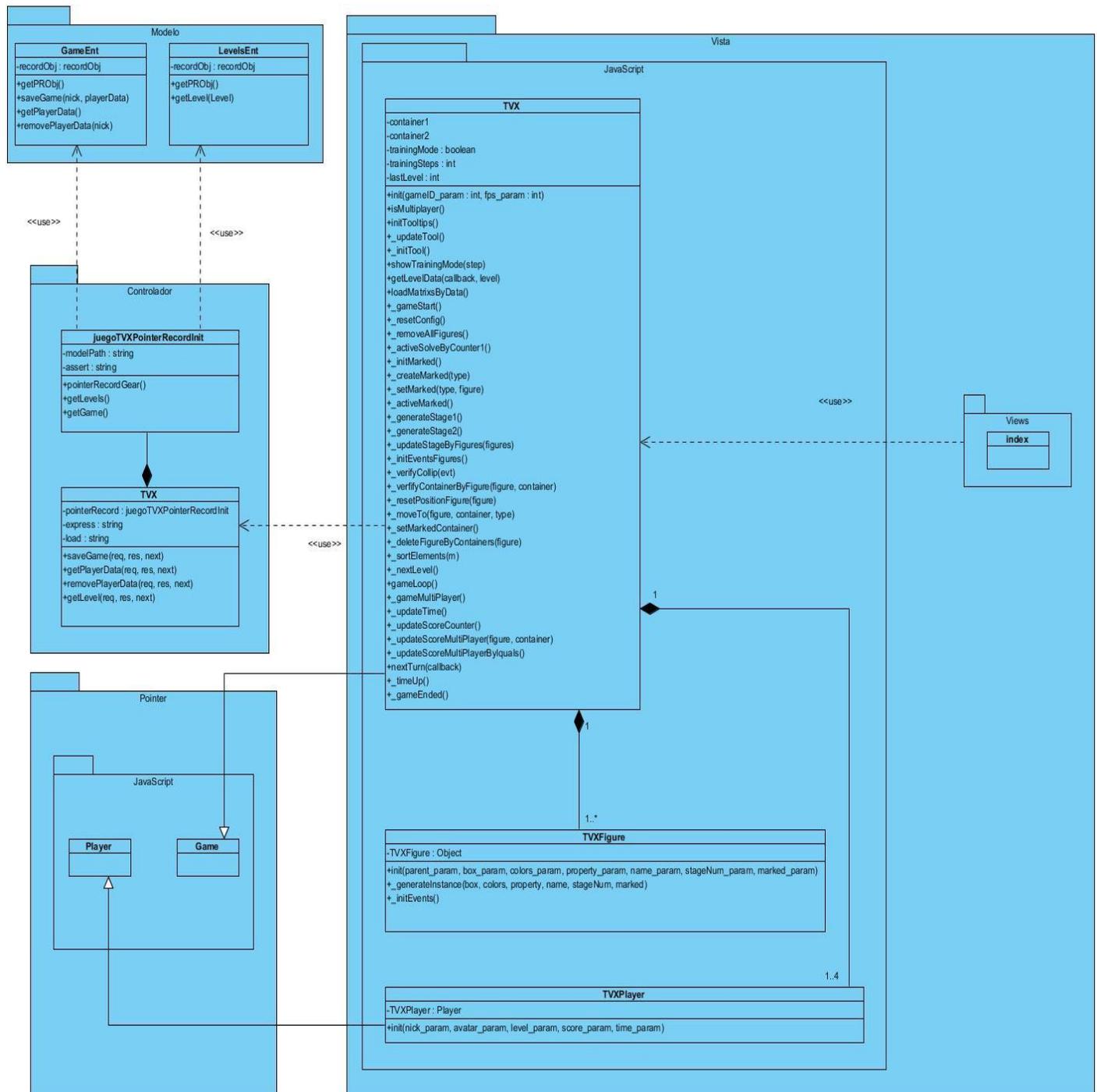


Figura 19: Diagrama de clases del juego Tetravex

Anexo 18:

Tabla 26: Descripción del diagrama de clases del juego Tetravex

Nombre	TVX
Tipo de clase	Vista
Atributo	Tipo de dato
contenedor1	–
contenedor2	–
trainingMode	boolean
trainingSteps	int
lastLevel	int
Funcionalidades	
Nombre	Descripción
init(gameID_param: int, fps_param )	Inicializa los atributos.
initGame()	Inicializa el juego.
isMultiplayer()	Verifica si el modo es multijugador.
gameLoop()	Actualiza el escenario, la puntuación y el tiempo.
_updateTool	Actualiza el contador de la lupa.
_initTool()	Permite utilizar la lupa.
loadMatrixsByData()	Actualiza las variables de configuración del nivel.
showTrainingMode(step)	Se encarga del nivel de entrenamiento. Se le pasa por parámetro el paso del tutorial.
_gameStart()	Inicia el nivel.
getLevelData(callback, level)	Extrae los datos del json.
_resetConfig()	Inicializa las variables para el próximo nivel.

<code>_removeAllFigures()</code>	Elimina todas las figuras de los juegos.
<code>_activeSolveByCounter()</code>	Asigna la cantidad de bonos lupa del nivel y verifica que si se está utilizando no pueda volverse a activar.
<code>_initMarked()</code>	Inicializa la animación de la lupa.
<code>_createMarked(type)</code>	Crea la animación de la lupa.
<code>_setMarked(type, figure)</code>	Cambia la posición de la animación de la figura.
<code>_activeMarked()</code>	Activa la animación de la lupa.
<code>_generateStage1()</code>	Crea el espacio de la solución.
<code>_generateStage2()</code>	Crea el espacio que contiene las figuras.
<code>_updateStageByFigures(figures)</code>	Actualiza el escenario con las figuras.
<code>_initEventsFigures()</code>	Inicia los eventos de las figuras.
<code>_verifyCollipse(evt)</code>	Verifica si una figura está sobre un contenedor determinado.
<code>_verifyContainerByFigure(figure, container)</code>	Verifica si una figura puede moverse hacia un contenedor determinado.
<code>_resetPositionFigure(figure)</code>	Retorna a una figura a su posición anterior.
<code>_moveTo(figure, container, type)</code>	Mueve una figura a un contenedor.
<code>_setMarkedContainer()</code>	Cambia la posición de la animación del contenedor.
<code>_deleteFigureByContainers(figure)</code>	Eliminar la figura del contenedor en el que se encontraba anteriormente.
<code>_sortElements(m)</code>	Intercambiar el orden de los colores.
<code>_nextLevel()</code>	Cambia el nivel.
<code>_gameMultiPlayer()</code>	Inicializa el juego para varios jugadores.

<code>_updateTime()</code>	Actualiza el tiempo.
<code>_updateScoreCounter()</code>	Actualiza la puntuación para un jugador.
<code>_updateScoreMultiPlayer(figure, container)</code>	Actualiza la puntuación cuando existen varios jugadores.
<code>_updateScoreMultiPlayerByIquals()</code>	Obtiene el ganador cuando los jugadores tengan la misma puntuación.
<code>nextTurn(callback)</code>	Pasa al siguiente jugador.
<code>_timeUp()</code>	Muestra el resultado del jugador activo al terminarse el tiempo.
<code>_gameEnded()</code>	Finaliza el juego.

Anexo 19:

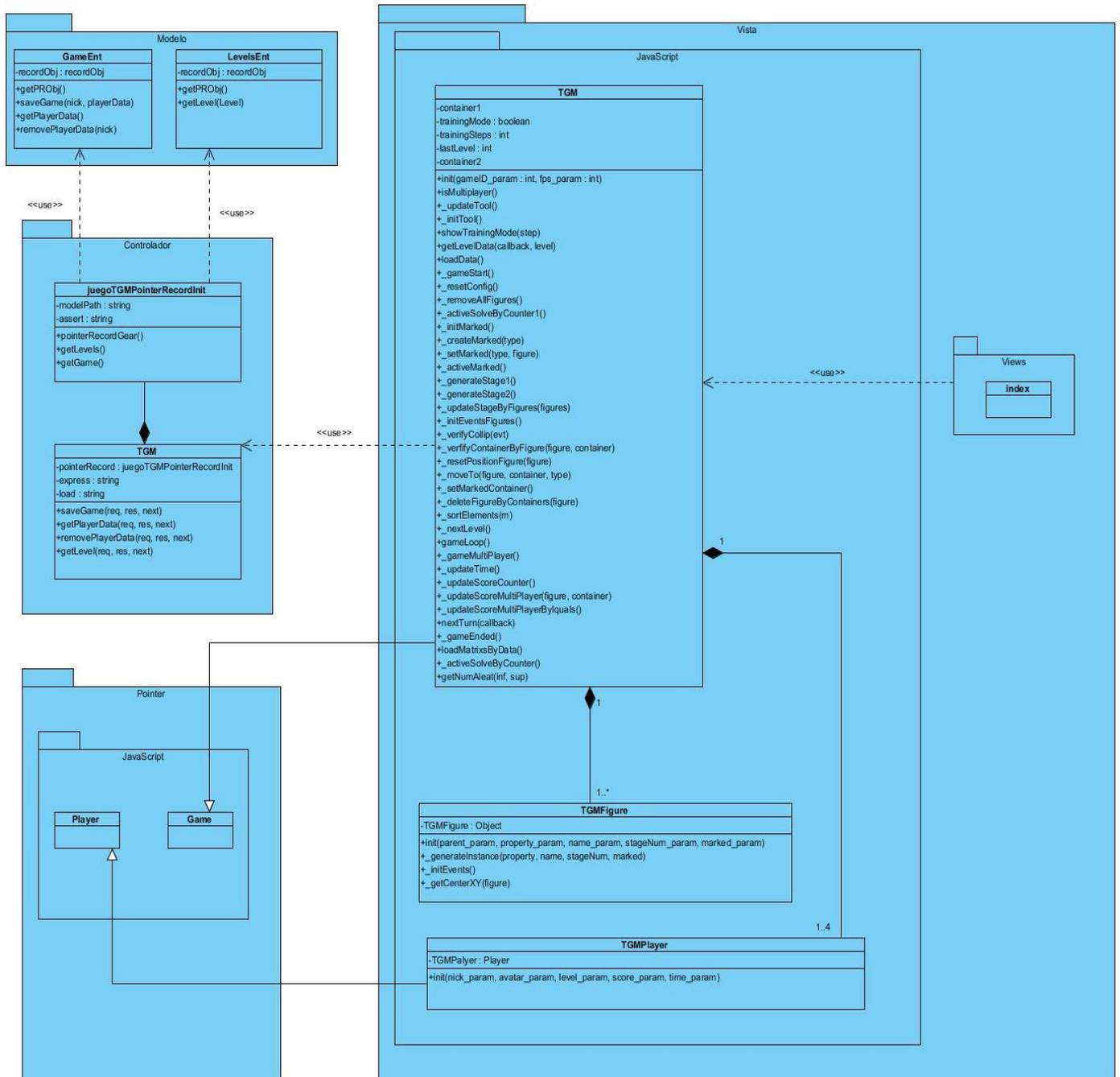


Figura 20: Diagrama de clases del juego Tangram

Anexo 20:

Tabla 27: Descripción del diagrama de clases del juego Tangram

Nombre	TGM
Tipo de clase	Vista
Atributo	Tipo de dato
contenedor1	–
contenedor2	–
trainingMode	boolean
trainingSteps	int
lastLevel	int
Funcionalidades	
Nombre	Descripción
init(gameID_param: int, fps_param )	Inicializa los atributos.
initGame()	Inicializa el juego.
isMultiplayer()	Verifica si el modo es multijugador.
gameLoop()	Actualiza el escenario, la puntuación y el tiempo.
_updateTool()	Actualiza el contador de la lupa.
_initTool	Permite utilizar la lupa.
loadData()	Actualiza las variables de configuración del nivel.
showTrainingMode(step)	Se encarga del nivel de entrenamiento. Se le pasa por parámetro el paso del tutorial.
_gameStart()	Inicia el nivel.
getLevelData(callback, level)	Extrae los datos del json.
_resetConfig()	Inicializa las variables para el próximo nivel.

<code>_removeAllFigures()</code>	Elimina todas las figuras del juego.
<code>_activeSolveByCounter()</code>	Asigna la cantidad de bonos lupa del nivel y verifica que si se está utilizando no pueda volverse a activar.
<code>_initMarked()</code>	Inicializa la animación de la lupa.
<code>_createMarked(type)</code>	Crear la animación de la lupa.
<code>_setMarked(type, figure)</code>	Cambia la posición de la animación de la figura.
<code>_activeMarked()</code>	Activa la animación de la lupa.
<code>_generateStage1()</code>	Crea el espacio de la solución.
<code>_generateStage2()</code>	Crea el espacio que contiene las figuras.
<code>_updateStageByFigures(figures)</code>	Actualiza el escenario con las figuras.
<code>_initEventsFigures()</code>	Inicia los eventos de las figuras.
<code>_getNumAleat(inf, sup)</code>	Genera una posición aleatoria que se utiliza para ubicar las figuras dentro del escenario.
<code>_verifyCollipse(evt)</code>	Verifica si una figura está sobre un contenedor determinado.
<code>_verifyContainerByFigure(figure, container)</code>	Verifica si una figura puede moverse hacia un contenedor determinado.
<code>_resetPositionFigure(figure)</code>	Retorna a una figura a su posición anterior.
<code>_moveTo(figure, container, type)</code>	Mueve una figura a un contenedor.
<code>_setMarkedContainer()</code>	Cambia la posición de la animación del contenedor.
<code>_deleteFigureByContainers(figure)</code>	Eliminar la figura del contenedor en el que se encontraba anteriormente.
<code>_sortElements(m)</code>	Intercambiar el orden de las figuras.

<code>_nextLevel()</code>	Pasa de nivel.
<code>_gameMultiPlayer()</code>	Inicializa el juego para varios jugadores.
<code>_updateTime()</code>	Actualiza el tiempo.
<code>_updateScoreCounter()</code>	Actualiza la puntuación para un jugador.
<code>_updateScoreMultiPlayer(figure, container)</code>	Actualiza la puntuación cuando existen varios jugadores.
<code>_updateScoreMultiPlayerByIquals()</code>	Obtiene el ganador cuando los jugadores tengan la misma puntuación.
<code>nextTurn(callback)</code>	Pasa al siguiente jugador.
<code>_levelUp()</code>	Muestra el resultado del jugador activo al terminarse el tiempo.
<code>_gameEnded()</code>	Finaliza el juego.

## Anexo 21:

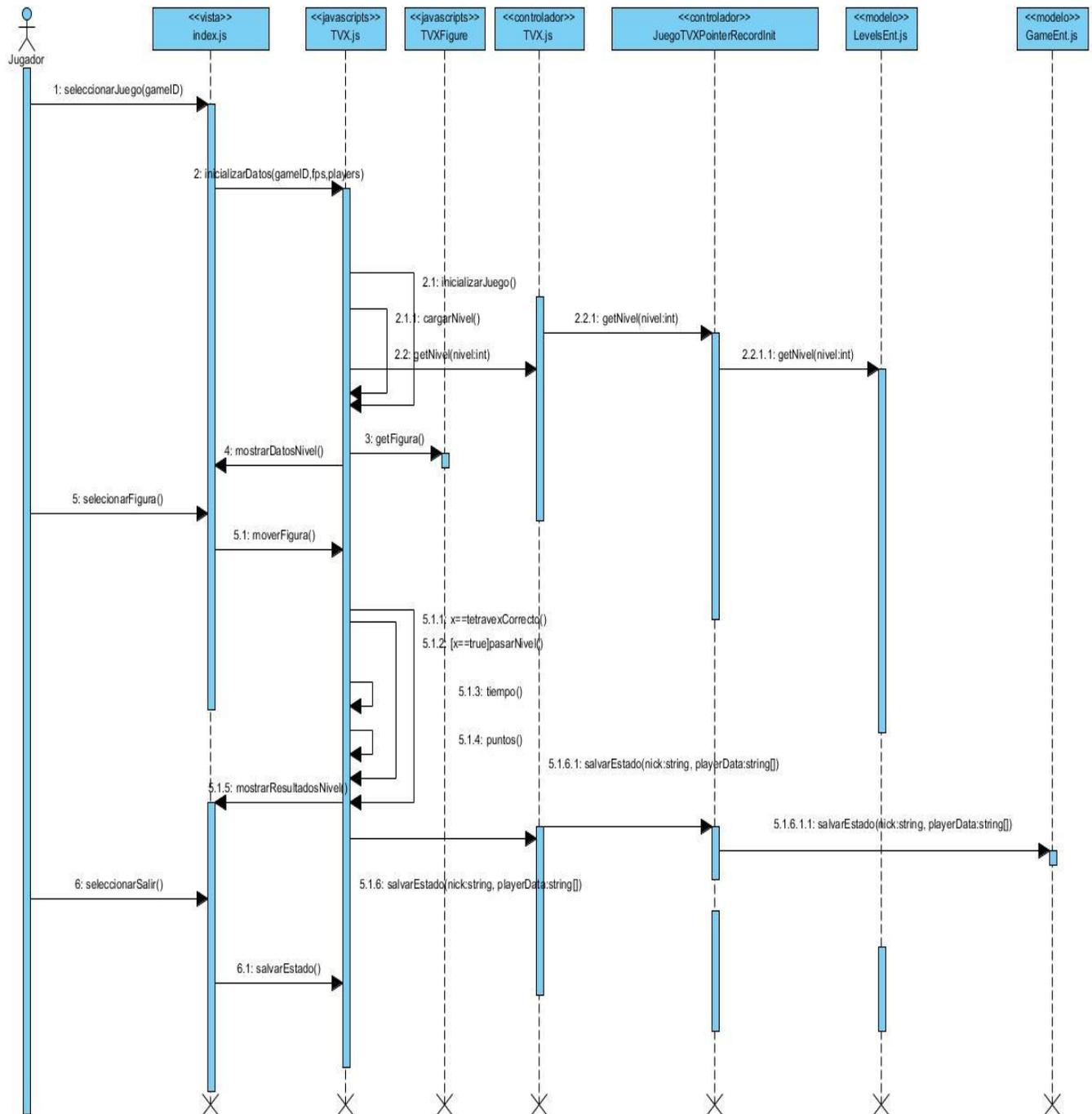


Figura 21: Diagrama de secuencia del juego Tetravex

## Anexo 22:

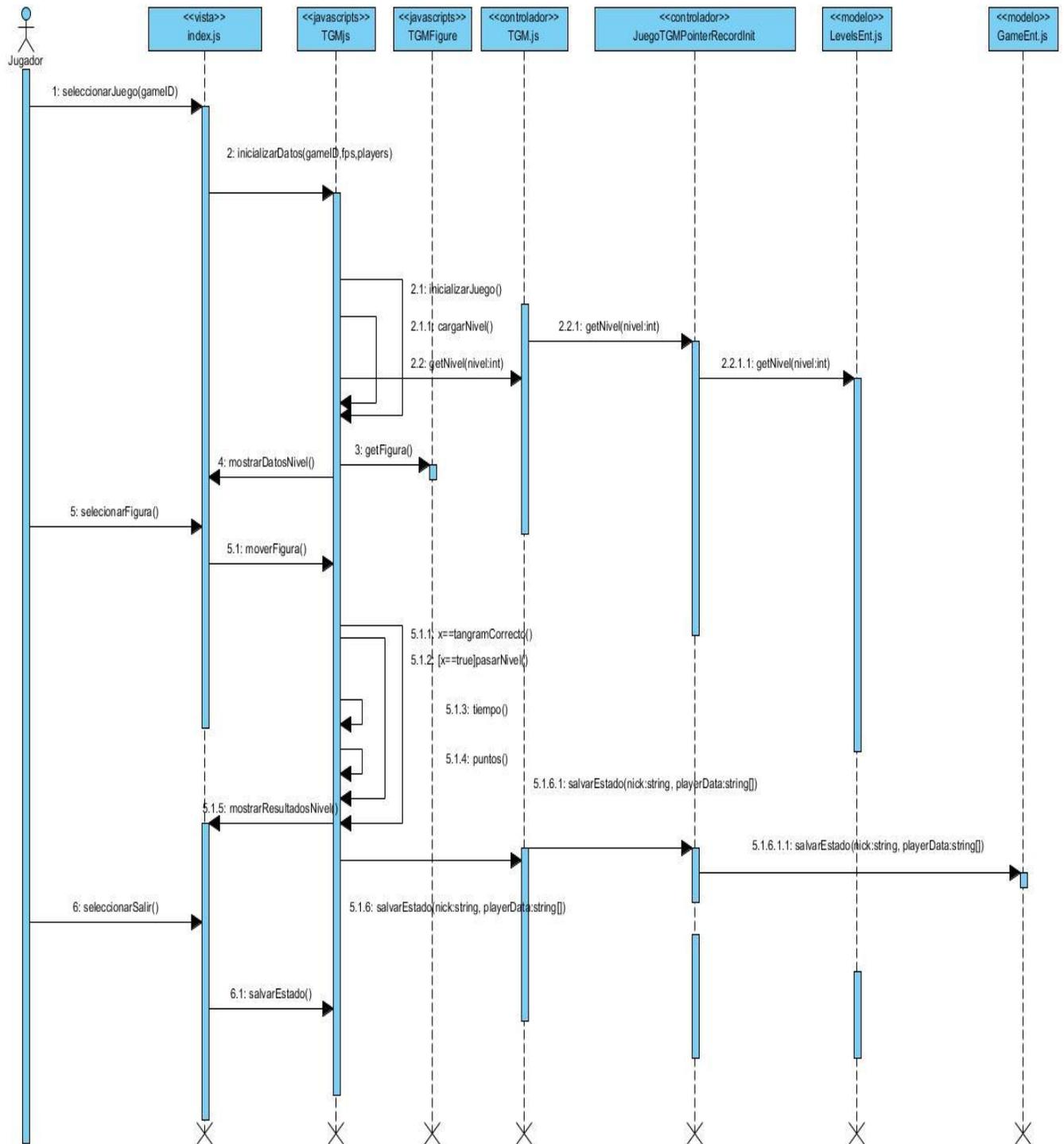


Figura 22: Diagrama de secuencia del juego Tangram

Anexo 23:

## Historia del juego Rompecabezas

Manolito se dedica a coleccionar las fotos de sus dibujos animados preferidos en un álbum. Con mucho cuidado lo guarda en su cuarto para que su hermano menor no lo encuentre. Un buen día, sin percatarse, Manolito olvidó la caja donde almacenaba su álbum encima de la cama y al regresar de la escuela se sorprendió pues el mismo estaba destruido. El inocente hermano había cortado en pequeños pedazos cada foto. La tristeza que sintió Manolito era tan grande que comenzó a llorar y a quejarse con sus padres. Ante tal situación ellos le dijeron que no se desesperara pues todavía existía la oportunidad de pegar aquellas piezas y reconstruir las fotos. Se te ha asignado la tarea de ayudar a Manolito y su familia a resolver esta situación.

El Rompecabeza dispone de tres niveles de complejidad:

- Fácil: la matriz es de 3x3.
- Medio: la matriz es de 4x4.
- Difícil: la matriz es de 5x5.

## Niveles

Nivel 1: La matriz es de 3x3 y el tipo de corte es irregular.

Nivel 2: La matriz es de 3x3 y el tipo de corte es cuadrado.

Nivel 3: La matriz es de 4x4 y el tipo de corte es irregular.

Nivel 4: La matriz es de 4x4 y el tipo de corte es cuadrado.

Nivel 5: La matriz es de 5x5 y el tipo de corte es irregular.

Nivel 6: La matriz es de 5x5 y el tipo de corte es cuadrado.

## Puntos

Cada rompecabeza resuelto aporta veinte puntos y se le resta el tiempo empleado para armarlo. Por no utilizar el bono pincel mágico el jugador obtiene cinco puntos y cada bono lupa que no use le aporta tres.

## Bonos

Forma de obtener los bonos: En cada nivel se le otorga al jugador la bonificación “lupa” que consiste en la ubicación de una pieza dentro del rompecabezas.

En cada nivel se le otorga al jugador el bono “pincel mágico” el cual muestra la imagen original que se debe completar.

Forma de activar el bono: En la parte inferior de la pantalla debe aparecer el icono del bono y al hacer click sobre el mismo se activa.

### **Anexo 24:**

#### **Historia del juego Puzle**

Carlitos paseaba por la playa y por casualidad tropezó con una botella que contenía algo. Inmediatamente pensó que solo era un trozo de papel viejo, sin embargo, descubrió que había hallado el mapa del tesoro del pirata Barba Negra el cual permanecía escondido y según se comentaba poseía grandes riquezas. Al analizarlo el niño se percató que tenía un problema pues el mapa estaba incompleto. El pirata amante de los rompecabezas, construyó varios puzles que el usuario debe resolver para encontrar cada parte del mapa. Imagina que eres el niño que llevará a cabo esta travesía y deber ingeniártelas para conformar el mapa original.

Niveles de complejidad:

- Fácil: la matriz es de 3x3.
- Medio: la matriz es de 4x4.
- Difícil: la matriz es de 5x5.

#### **Niveles**

Nivel 1: La matriz es de 3x3 y la primera fila está ordenada.

Nivel 2: La matriz es de 3x3 y las dos primeras piezas se encuentran en el lugar que le corresponde.

Nivel 3: La matriz es de 3x3.

Nivel 4: La matriz es de 4x4 y la primera fila está ordenada.

Nivel 5: La matriz es de 4x4.

Nivel 6: La matriz es de 5x5.

#### **Puntos**

Cada puzle equivale a cien puntos y al pasar diez segundos va disminuyendo en uno. Si no se utiliza la bonificación se adicionan diez puntos al final del nivel.

## **Bonos**

Forma de obtener el bono: En cada nivel se le otorgará al jugador el bono catalejo.

Forma de activar el bono: En la parte inferior de la pantalla se debe colocar el icono del bono que posee el usuario, al hacer click sobre el mismo se activa.

## **Anexo 25:**

### **Historia del juego Tangram**

En un reino muy lejano gobernaba el rey Felipe que tenía una hija llamada Diana. La princesa tenía un espíritu aventurero y siempre estaba en busca de nuevas experiencias. Un día ésta encontró un pasadizo secreto que nunca había visto dentro del castillo y sintió mucha curiosidad por saber a dónde conducía así que entró en éste. Inmediatamente después que pasó la puerta se cerró a sus espaldas sin poder abrirla. Diana caminó hasta llegar a la próxima puerta que estaba cerrada y no hallaba como atravesarla. Allí comenzó a buscar alguna pista para resolver el problema y de repente se percató que había un cofre en el suelo el cual guardaba en su interior 7 figuras y un mensaje. En el mismo venía escrito que para salir de aquel pasadizo debía cruzar todas las puertas que separaban las secciones del mismo y esto sería posible sólo armando la imagen dibujada en ellas, utilizando las figuras. Ayuda a la princesa a pasar por el pasadizo escondido dentro del castillo.

## **Niveles**

Nivel 1: Se debe armar el tangram en menos de diez minutos.

Nivel 2: Se debe armar el tangram en menos de nueve minutos y treinta segundos.

Nivel 3: Se debe armar el tangram en menos de nueve minutos.

Nivel 4: Se debe armar el tangram en menos de ocho minutos y treinta segundos.

Nivel 5: Se debe armar el tangram en menos de ocho minutos.

Nivel 6: Se debe armar el tangram en menos de siete minutos y treinta segundos.

Nivel 7: Se debe armar el tangram en menos de siete minutos.

Nivel 8: Se debe armar el tangram en menos de seis minutos y treinta segundos.

Nivel 9: Se debe armar el tangram en menos de seis minutos.

Nivel 10: Se debe armar el tangram en menos de cinco minutos y treinta segundos.

## **Puntos**

Cada tangram resuelto aporta veinte puntos, sino utiliza el bono lupa el jugador obtiene cinco puntos.

## **Bono**

Forma de obtener el bono: En cada nivel se le otorga al jugador el bono “lupa” el cual señala una figura y la posición que le corresponde en la imagen sombreada.

Forma de activar el bono: En la parte inferior de la pantalla debe aparecer el icono del bono y al hacer click sobre el mismo se activa.

## **Anexo 26:**

### **Historia del juego Tetravex**

El pueblo Colorín estaba rodeado por un hermoso arcoíris que siempre aparecía al salir el sol. Su presencia en el horizonte llenaba a los habitantes de alegría y mantenía el lugar con colores muy intensos. Aunque no todo era bueno pues allí vivía un malvado brujo llamado Baltazar que pretendía acabar con la felicidad de Colorín. Un día todos quedaron sorprendidos al ver que el arcoíris había perdido sus colores. Aquella obra sólo podía ser culpa del hechicero, que juró devolver todo a la normalidad si una persona lograba resolver el reto que él proponía.

Pasado los días ya nada era igual en aquella ciudad y ante tal situación llegaban hasta Baltazar las personas más inteligentes de la región pero nadie era capaz de completar un grupo de rompecabezas de colores. Ayuda al pueblo a destruir las intenciones del hechicero y recupera los colores del arcoíris.

El Tetravex tiene tres niveles de complejidad:

- Fácil: matriz de 2x2.
- Medio: matriz de 3x3.
- Difícil: matriz de 4x4 y 5x5.
- Avanzado: matriz de 6x6.

## Niveles

Nivel 1: La matriz es de 2x2 y se encuentra vacía.

Nivel 2: La matriz es de 3x3 y se le brinda al usuario una figura dentro de la misma.

Nivel 3: La matriz es de 3x3 y se encuentra vacía.

Nivel 4: La matriz es de 4x4 y se le brinda al usuario tres figuras dentro de la misma.

Nivel 5: La matriz es de 4x4 se le brinda al usuario una figura dentro de la misma.

Nivel 6: La matriz es de 5x5 y se encuentra vacía.

Nivel 7: La matriz es de 6x6 y se encuentra vacía.

## Puntos

En dependencia de la dimensión de la matriz es la cantidad de puntos que aporta cada tetravex resuelto como se indica a continuación:

- Si la dimensión es 2x2 el tetravex aporta veinte puntos.
- Si la dimensión es 3x3 el tetravex aporta treinta puntos.
- Si la dimensión es 4x4 el tetravex aporta cuarenta puntos.
- Si la dimensión es 5x5 el tetravex aporta cincuenta puntos.
- Si la dimensión es 6x6 el tetravex aporta sesenta puntos.

Por cada lupa que no utilice se le otorgan cinco puntos adicionales al jugador.

## Bono

Forma de obtener el bono: Cuando el jugador coloca la primera figura se le otorga el bono “lupa” el cual señala una figura y la posición que le corresponde dentro del espacio de solución. La cantidad de lupas que se brindan al jugador es la dimensión de la matriz menos uno.

Forma de activar el bono: En la parte inferior de la pantalla debe aparecer el icono del bono y al hacer click sobre este se activa.