PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN VIDEOJUEGO EN 2D ORIENTADO A LA EJERCITACIÓN DE LA MEMORIA Y EL DESARROLLO DE LA INTELIGENCIA ESPACIAL

Tesis para optar por el Título de **Ingeniero Informático**, que presenta el bachiller:

Luis Alberto Quispe Gonzales

ASESOR: Ing. Raúl Valencia Ibarra

Lima, junio de 2013

RESUMEN

Hoy en día, el aspecto educacional es uno de los más importantes en todo país debido a que una nación logra desarrollarse plenamente gracias al aporte de sus ciudadanos y profesionales. Es aquí en donde surge el principal objetivo a conseguir con la presente tesis, promover el desarrollo de las habilidades cognitivas de las personas.

La actual tesis presentará un videojuego que permitirá a toda persona que lo utilice poder mejorar sus habilidades y destrezas a nivel cerebral debido a que se ha logrado realizar un trabajo minucioso y exhaustivo sobre las bondades que debe poseer una aplicación para poder cumplir este fin con gran eficacia.

Se ha decidido optar por esta solución debido a que, tras realizar un estudio del mercado actual, no se ha identificado un videojuego a nivel local que permita cumplir con estas necesidades. Si bien es cierto, a nivel global existen algunos videojuegos que buscan acercarse al objetivo expuesto en la presente tesis aún no se ha logrado uno que se consolide en cuanto al incentivo del desarrollo de las destrezas de los video-jugadores.

Se propone el videojuego como una solución recreativa, académica en incluso médica a la vez ya que permitirá a todo tipo de personas (niños, adultos, estudiantes, profesionales, pacientes, entre otros) poder utilizarlo sin problemas debido a su intuitiva y sencilla jugabilidad. Adicionalmente, las ventajas que ofrecerá el videojuego a desarrollar, así como las habilidades que ayudarán son múltiples y variadas y abarca aspectos desde la mejora de la memoria, pasando por la atención, velocidad, flexibilidad y hasta incluso la mejora en la destreza de la solución de problemas.

El proyecto se desarrolla en un escenario real en el cual se pueden aplicar las mejores prácticas de gestión y desarrollo de proyectos.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. CAPÍTULO 1: GENERALIDADES

En este capítulo se expondrá el problema que aborda el proyecto de tesis, así como las definiciones y conceptos básicos como parte del contexto del problema y el estado del arte. Adicionalmente, se presenta el plan del proyecto, donde se detallan las actividades a realizar en el transcurso del proyecto de tesis; conjuntamente con la definición y sustentación de la solución brindada.

1.1. Definición del problema

La importancia de la educación radica principalmente en el desarrollo de las personas, así como también en el de una sociedad. Es por ello, que se tiene como finalidad formar personas capaces de lograr su realización ética, intelectual, artística y cultural, para su integración a la sociedad, el ejercicio de su ciudadanía y el desarrollo de sus habilidades para ingresar al entorno laboral. Una educación de calidad permite que las personas se encuentren preparadas para enfrentar a un mundo altamente competitivo, globalizado y tecnológico. Por otro lado, contribuye a la formación de una sociedad democrática, solidaria, justa, próspera, que supere la pobreza e impulse el desarrollo del país [UNE001].

Dada la importancia de la educación, la Organización para la Cooperación y el Desarrollo Económico (OCDE) realiza cada 3 años un análisis del rendimiento de estudiantes a través de un proyecto denominado Programa para la Evaluación Internacional de Alumnos - PISA *(Programme for International Student Assessment*), que tiene por objetivo evaluar la formación de los alumnos cuando llegan al final de educación obligatoria, hacia los 15 años. La evaluación cubre las áreas de comprensión de lectura, matemáticas y competencia científica [OCD001]. En la Evaluación PISA 2009, participaron 65 países, de los cuales 30 son miembros de la OCDE y 8 son de Latinoamérica, entre ellos el Perú [EDU001].

Basado en el informe emitido por la OCDE, el Ministerio de Educación del Perú presentó los "Resultados de la Evaluación PISA 2009". A continuación, desde la **Figura 1.1** hasta la **1.4**, se muestran los puntajes promedios obtenidos en cada una de las áreas, y cuadros comparativos respecto a otros países.

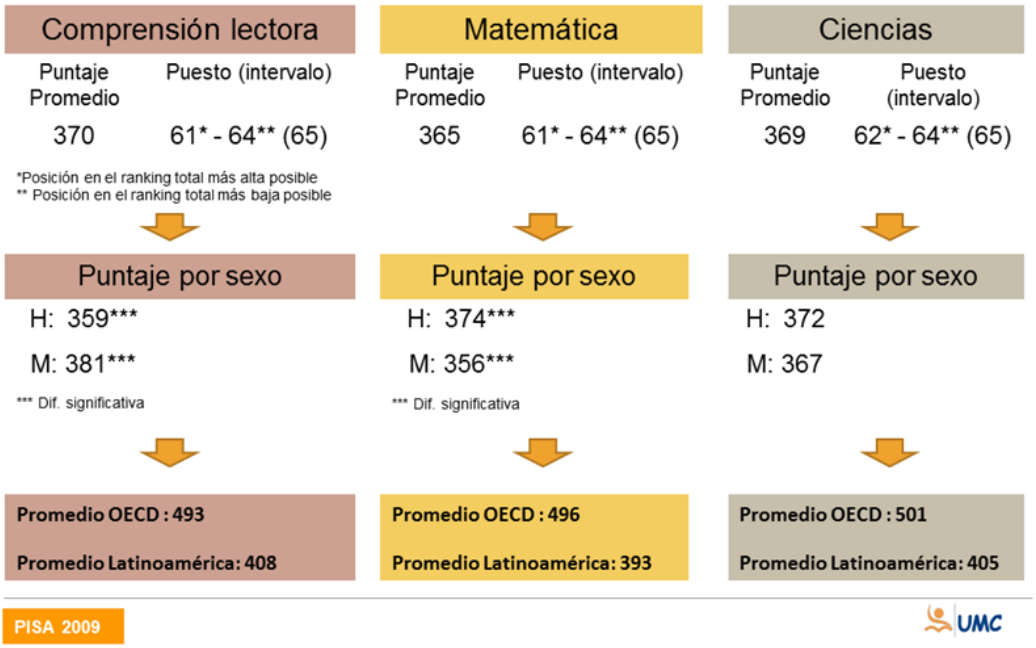


Figura 1.1: Perú: Resultados PISA 2009 según puntaje promedio

Tomado de: Resultados de la Evaluación PISA 2009

Escala en Comprensión lectora							
Descripción	País	Puntaje	S.E.	Rango del ranking			
				Países OECD		Todos los países	
				Posición en el ranking más alta posible	Posición en el ranking más baja posible	Posición en el ranking más alta posible	Posición en el ranking más baja posible
Países con los puntajes más altos	Shanghai-China	556	(2,4)	-	-	1	1
	Corea del Sur	539	(3,5)	1	2	2	4
	Finlandia	536	(2,3)	1	2	2	4
	Hong Kong-China	533	(2,1)	-	-	3	4
	Singapur	526	(1,1)	-	-	5	6
Países de América Latina	Chile	449	(3,1)	33	33	44	44
	Uruguay	426	(2,6)	-	-	46	50
	México	425	(2,0)	34	34	46	49
	Colombia	413	(3,7)	-	-	50	55
	Brasil	412	(2,7)	-	-	51	54
	Argentina	398	(4,6)	-	-	55	59
	Panamá	371	(6,5)	-	-	61	64
	Perú	370	(4,0)	-	-	61	64

Figura 1.2: PISA 2009: Puntaje promedio en comprensión lectora para algunos países seleccionados

Tomado de: Resultados de la Evaluación PISA 2009

Escala en Matemática							
Descripción	País	Puntaje	S.E.	Rango del ranking			
				Países OECD		Todos los países	
				Posición en el ranking más alta posible	Posición en el ranking más baja posible	Posición en el ranking más alta posible	Posición en el ranking más baja posible
Países con los puntajes más altos	Shanghai-China	600	(2,8)	-	-	1	1
	Singapur	562	(1,4)	-	-	2	2
	Hong Kong-China	555	(2,7)	-	-	3	4
	Corea del Sur	546	(4,0)	1	2	3	6
	Taiwán	543	(3,4)	-	-	4	7
Países de América Latina	Uruguay	427	(2,6)	-	-	45	49
	Chile	421	(3,1)	33	34	47	51
	México	419	(1,8)	33	34	49	51
	Argentina	388	(4,1)	-	-	55	58
	Brasil	386	(2,4)	-	-	55	58
	Colombia	381	(3,2)	-	-	56	59
	Perú	365	(4,0)	-	-	61	64
	Panamá	360	(5,2)	-	-	62	64

Figura 1.3: PISA 2009: Puntaje promedio en matemática para algunos países seleccionados

Tomado de: Resultados de la Evaluación PISA 2009

Escala en Ciencias							
Descripción	País	Puntaje	S.E.	Rango del ranking			
				Países OECD		Todos los países	
				Posición en el ranking más alta posible	Posición en el ranking más baja posible	Posición en el ranking más alta posible	Posición en el ranking más baja posible
Países con los puntajes más altos	Shanghai-China	575	(2,3)	-	-	1	1
	Finlandia	554	(2,3)	1	1	2	3
	Hong Kong-China	549	(2,8)	-	-	2	3
	Singapur	542	(1,4)	-	-	4	6
	Japón	539	(3,4)	2	3	4	6
Países de América Latina	Chile	447	(2,9)	32	33	43	45
	Uruguay	427	(2,6)	-	-	47	49
	México	416	(1,8)	34	34	50	51
	Brasil	405	(2,4)	-	-	52	56
	Colombia	402	(3,6)	-	-	53	58
	Argentina	401	(4,6)	-	-	53	59
	Panamá	376	(5,7)	-	-	60	64
	Perú	369	(3,5)	-	-	62	64

Figura 1.4: PISA 2009: Puntaje promedio en ciencias para algunos países seleccionados

Tomado de: Resultados de la Evaluación PISA 2009

Según las estadísticas mostradas anteriormente, el rendimiento estudiantil en el Perú es bajo respecto a otros países, y esto se ve evidenciado en las dificultades en cuanto a las competencias cognitivas pues no logran comprender completamente lo que leen, no son capaces de aplicar sus conocimientos y destrezas en matemáticas para enfrentar situaciones novedosas y encuentran muy difícil la toma de decisiones basadas en pensamiento y razonamiento científico.

Ante la situación descrita, los especialistas en neurología, psicología y docencia han explorado diversas alternativas con el fin de conseguir el potenciamiento y mejora de las habilidades cognitivas de sus pacientes o alumnos, tratando al mismo tiempo de ahorrar costos y mejorar la efectividad de los tratamientos realizados.

En la actualidad, se han empezado a usar los videojuegos con fines de aprendizaje, tal como se comenta en la *Serious Games Association* (SGA) [SGA001]:

"*James Paul Gee, profesor de lectura en la Universidad de Wisconsin-Madison, fue uno de los primeros en documentar, desde su propia experiencia al jugar videojuegos con sus hijos, la manera diferente que las herramientas de aprendizaje digitales conducen no sólo a la exploración y el descubrimiento, sino también a nuevos tipos de desafíos. Las investigaciones muestran que los juegos son una forma más interactiva y*

participativa para ayudar a personas de todas las edades a entender casi cualquier cosa."

Asimismo, el Diario Médico de Madrid informa los progresos logrados haciendo uso de los videojuegos para la estimulación cognitiva en pacientes que padecen la enfermedad de Alzheimer [SAL001]:

"*La Asociación de Familiares de Enfermos de Alzheimer de Salamanca (AFA) y la Universidad de Salamanca han elaborado un estudio acerca de los beneficios sobre la estimulación cognitiva que producen los ejercicios realizados con la consola Wii en Enfermedad de Alzheimer (…). La AFA y la Universidad de Salamanca han demostrado que el programa de estimulación cognitiva mediante el juego Big Brain Academy ha resultado ser más efectivo que un programa de psicoestimulación tradicional o actividades en formato lápiz y papel, ya que los pacientes del grupo del videojuego manifestaron una reducción significativa del declive cognitivo y de la sintomatología depresiva en relación con el grupo de estimulación tradicional, según ha manifestado a DM Roberto Rodríguez Pérez, director médico de la AFA Salamanca".*

En el Perú, actualmente el uso de videojuegos con fines educativos o como parte de tratamientos médicos no se encuentra muy difundido, por lo que se tiene una buena oportunidad para presentar una alternativa económica y viable de poder estimular y mejorar habilidades cognitivas a través de videojuegos. Asimismo, la propuesta contiene elementos lúdico-recreativos que permite una mayor aceptación por parte de las personas que emplearían un software de este tipo.

Por lo expuesto se puede ver la necesidad de una herramienta, con un enfoque lúdico (un videojuego) para apoyar la ejercitación del razonamiento lógico, la memoria de corto y mediano plazo, así como el desarrollo de habilidades cognitivas referentes a la inteligencia espacial: reconocimiento de formas y figuras, capacidad para procesar tamaños, direcciones y relaciones espaciales, entre otros.

1.2. Marco conceptual

Para entender el problema planteado es necesario definir todos los elementos involucrados que permitirán comprender mejor la propuesta del tema y su implementación.

1.2.1. Definiciones

En este punto se encuentran las definiciones formales de los principales términos utilizados en el proyecto para una mejor comprensión del mismo conforme se vaya avanzando a las siguientes secciones.

- **Memoria**

La memoria [HIP001] es la capacidad mental que posibilita a un sujeto registrar, conservar y evocar las experiencias (ideas, imágenes, acontecimientos, sentimientos, etc.). A continuación, se pasa a clasificar los tipos de memoria:

Según su duración:

- **Memoria sensorial:** Puede ser visual (de escasa duración, menos de medio segundo) o auditiva (entre uno y dos segundos de duración).
- **Memoria inmediata** (memoria a corto plazo)**:** Duración de menos de un minuto y limitada a unos pocos objetos.
- **Memoria reciente:** Su duración oscila entre unos minutos y varias semanas, y su capacidad de almacenamiento es mayor que la de la memoria inmediata.
- **Memoria remota:** Mantiene la información desde semanas hasta toda la vida.

Según su contenido o utilización:

- **Memoria de referencia:** Contiene la información reciente y remota obtenida por experiencias previas.
- **Memoria de trabajo:** Se aplica a un proceso activo que está siendo actualizado de manera continua por la experiencia de un momento determinado.

- **Memoria episódica:** Contiene la información relativa a sucesos acontecidos en un momento y lugar determinados.

- **Memoria semántica:** Contiene información que no varía, como por ejemplo el número de horas que tiene el día o las capitales de los departamentos de Perú. Los elementos pertenecen habitualmente a categorías determinadas, llamadas categorías semánticas: nombre de animales, reyes godos, instrumentos musicales, etc.

- **Memoria declarativa** (o explícita)**:** Contiene los hechos del mundo y los acontecimientos personales del pasado que es necesario recuperar de manera consciente para recordarlos.

- **Memoria de procedimiento** (o implícita): Aprendizaje y conservación de destrezas y habilidades, como peinarse o montar en bicicleta. Estos procedimientos se automatizan y no precisan de una ejecución consciente.

Durante el desarrollo del proyecto se hace énfasis en la ejercitación de la memoria inmediata y reciente (según su duración), así como en el uso de la memoria de trabajo (según su utilización).

- **Habilidades cognitivas**

Las habilidades cognitivas son las facilitadoras del conocimiento, aquellas que operan directamente sobre la información: recogiendo, analizando, comprendiendo, procesando y guardando información en la memoria para, posteriormente, poder recuperarla y utilizarla dónde, cuándo y cómo convenga.

En general, son las siguientes [COG001]:

- **Atención:** Exploración, fragmentación, selección y contra-distractoras.
- **Comprensión** (habilidades de trabajo intelectual)**:** Captación de ideas, subrayado, traducción a lenguaje propio y resumen, gráficos, redes, esquemas y mapas conceptuales. A través del manejo del lenguaje oral y escrito (velocidad, exactitud, comprensión).
- **Elaboración:** Preguntas, metáforas, analogías, organizadores, apuntes y mnemotecnias.
- **Memorización/Recuperación** (habilidades de estudio)**:** Codificación y generación de respuestas. Como ejemplo clásico y básico, el método 3R: Leer, recitar y revisar (del inglés: *read*, *recite*, *review*).

Las habilidades cognitivas aluden directamente a las distintas capacidades intelectuales que resultan de la disposición o capacidad que demuestran los individuos al hacer algo. Estas habilidades son, como indican Hartman y Sternberg, los obreros del conocimiento. Pueden ser numerosas, variadas y de gran utilidad, a la hora de trabajar en las distintas áreas de conocimientos y cuya actividad específica se ve afectada por multitud de factores que dependen de la materia, de la tarea, de las actitudes y de las variables del contexto donde tienen lugar. Precisamente, la actuación estratégica se refiere a la selección, organización y disposición de las habilidades que caracterizan el sistema cognitivo del individuo.

- **Teoría de las inteligencias múltiples**

Teoría propuesta en 1993 por el doctor Howard Gardner, director del Proyecto Zero y profesor de Psicología y Ciencias de la educación en la Universidad de Harvard [IMU001]. A través de esta teoría, el Dr. Gardner llegó a la conclusión de que la inteligencia no es algo innato y fijo que domina todas las destrezas y habilidades de resolución de problemas que posee el ser humano, y estableció que la inteligencia está localizada en diferentes áreas del cerebro, interconectadas entre sí y que pueden también trabajar en forma individual, teniendo la propiedad de desarrollarse ampliamente si encuentran un ambiente que ofrezca las condiciones necesarias para ello. En dicha teoría se menciona la existencia de siete inteligencias, las cuales se describen a continuación:

  - **La inteligencia lingüística-verbal:** Es la capacidad de emplear de manera eficaz las palabras, manipulando la estructura o sintaxis del lenguaje, la fonética, la semántica, y sus dimensiones prácticas.
  - **La inteligencia física-cinestésica:** Es la habilidad para usar el propio cuerpo para expresar ideas y sentimientos, y sus particularidades de coordinación, equilibrio, destreza, fuerza, flexibilidad y velocidad, así como propioceptivas y táctiles.
  - **La inteligencia lógica-matemática:** Es la capacidad de manejar números, relaciones y patrones lógicos de manera eficaz, así como otras funciones y abstracciones de este tipo.
  - **La inteligencia espacial:** Es la habilidad de apreciar con certeza la imagen visual y espacial, de representarse gráficamente las ideas, y de sensibilizar el color, la línea, la forma, la figura, el espacio y sus interrelaciones.

- **La inteligencia musical:** Es la capacidad de percibir, distinguir, transformar y expresar el ritmo, timbre y tono de los sonidos musicales.
- **La inteligencia interpersonal:** Es la posibilidad de distinguir y percibir los estados emocionales y signos interpersonales de los demás, y responder de manera efectiva a dichas acciones de forma práctica.
- **La inteligencia intrapersonal:** Es la habilidad de la auto-introspección, y de actuar consecuentemente sobre la base de este conocimiento, de tener una autoimagen acertada, y capacidad de autodisciplina, comprensión y amor propio.

Durante la implementación del proyecto se hace énfasis en el desarrollo de la inteligencia espacial, lo que se traduce, en la capacidad de establecer interrelaciones entre tamaños, direcciones y figuras.

- **Leguaje de Programación Python**

Python es un lenguaje de programación multiparadigma ya que soporta la programación orientada a objetos, programación imperativa, y en menor medida, programación funcional. Es un lenguaje interpretado, lo que significa, que está diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados. Entre las principales características del lenguaje Python se encuentran las siguientes [PYT001]:

- Capacidad de ejecución en diversos sistemas operativos: Windows, Linux/Unix, OS/2, Mac, Amiga, entre otros.
- Fácil integración con las plataformas Java, COM y .NET.
- Posee licencia de código abierto.

- **Videojuegos**

Es un software creado para el entretenimiento en general y basado en la interacción de una a varias personas por medio de un controlador y un aparato electrónico que ejecuta dicho videojuego. En muchos casos, los videojuegos recrean entornos y situaciones virtuales en los que el jugador puede controlar uno o varios personajes (o cualquier otro elemento de dicho entorno) para conseguir objetivos por medio de reglas determinadas. Entre los principales géneros de videojuegos se encuentran los siguientes [VID001]:

- **Aventura:** Son videojuegos en los que el protagonista debe avanzar en la trama interactuando con diversos personajes y objetos.

- **Disparos:** En estos videojuegos el protagonista tiene que abrirse camino a base de disparos.
- **Educativos:** Videojuegos cuyo objetivo es dar a conocer al usuario algún tipo de conocimiento.
- **Estrategia:** Se caracterizan por la necesidad de manipular a un numeroso grupo de personajes, objetos o datos para lograr los objetivos. Según su temática los hay de gestión (ya sea esta económica o social) y bélicos.
- **Lucha:** Son videojuegos basados en combate cuerpo a cuerpo. Se dividen en videojuegos de uno contra uno (o *versus*), videojuegos de avanzar y pegar (o *beat'em up*) y el híbrido de ambos: *free-for-all* o todo contra todo.
- **Survival horror:** Correspondientes al género de terror. Los protagonistas viven aventuras donde salen airosos de situaciones típicas de una película de terror. Un factor importante es el terror psicológico, ayudado de una buena ambientación y apartado sonoro.
- **Plataformas:** Los videojuegos de plataformas se caracterizan por tener que recorrer, saltar o escalar una serie de plataformas y acantilados con enemigos, mientras se recogen objetos para poder completar el videojuego. Suelen usar desplazamiento lateral hacia la izquierda o hacia la derecha.
- **Rol:** Se inspiran en los juegos de rol clásicos, donde el protagonista interpreta un papel y ha de mejorar sus habilidades mientras interactúa con el entorno y otros personajes. Se suelen también caracterizar por que las decisiones de tu personaje influyen en su futuro próximo, dando la posibilidad de ir por el bien o el mal.
- **Musicales:** En este tipo de videojuegos, su desarrollo gira en torno a la música.
- **Agilidad mental/rompecabezas:** Son videojuegos donde el jugador debe resolver problemas que pueden ser de índole matemático, espacial o lógico, en estos juegos puede haber un cronómetro convirtiéndolos en una competencia a contrarreloj. Es frecuente encontrarlos en dispositivos pórtatiles.
- **Party games:** En este género los jugadores habrán de ir avanzando por turnos en un tablero virtual e ir superando diversas pruebas de tipos muy diversos en los que compiten entre sí por llegar lo antes posible a la meta, o conseguir la máxima cantidad posible de puntos.
- **Simulación:** Involucran al jugador en una situación simulada determinada, ya sea de gestor de un zoológico, una ciudad o una vida propia virtual.
- **Deportivo:** Son videojuegos basados en el mundo del deporte.
- **No lineal:** También conocidos como *sandbox*, se caracterizan por ser videojuegos en los que el jugador puede elegir el orden de las misiones o viajar libremente por el

mapa del videojuego, e interactuar con casi todo lo que este a su disposición. Estos juegos suelen ser una mezcla de disparos, luchas y carreras.

- **Cocos2D**

Cocos2D es un framework (estructura conceptual y tecnológica de soporte definido, con base a la cual otros proyectos de software pueden ser más fácilmente organizados y desarrollados) utilizado para construir juegos en 2D, demos y otras aplicaciones gráficas/interactivas. Este framework es utilizado para desarrollos en Python, C++ y Objective-C (este último para desarrollo en iPhone). Entre las principales características del framework se encuentran las siguientes [C2D001]:

- Fácil manejo del control del flujo entre diferentes escenas
- Manejo eficiente de sprites
- Transiciones elegantes (con estilo) entre escenas
- Intérprete Python incluido
- Licencia BSD
- Basado en OpenGL

Para el 12 de agosto de 2012 se lanzó la versión Cocos2D 0.5.5 (para el desarrollo en Python), que será la utilizada en la implementación del videojuego propuesto en el presente proyecto.

- **Algoritmo**

Los algoritmos son procedimientos bien definidos que toman un valor o conjunto de valores como entrada y producen un valor, o conjunto de valores como salida. Un algoritmo es, en consecuencia, un conjunto de pasos que transforman la entrada en una salida [ALG001].

- **Inteligencia Artificial**

La Inteligencia Artificial es la rama de la Ciencia de la Computación que estudia la resolución de problemas empleando algoritmos que incluyan la aplicación de cierta forma de razonamiento humano o comportamiento inteligente [IAR001].

1.2.2. Alcance

El software a desarrollar será un demo de un videojuego en dos dimensiones de *scroll* multidireccional del género plataformas, cuya historia se desarrollará a lo largo de dos niveles. El demo incluirá un módulo de configuración donde se podrá establecer los botones (teclas) que se utilizarán durante el juego, el control del sonido y el nivel de dificultad del juego. El videojuego se podrá ejecutar en un entorno Windows (a partir de Windows XP) y corresponde a un esquema *stand-alone*.

El proyecto incluirá la implementación de algoritmos basados en Inteligencia Artificial para manejar el movimiento y toma de decisiones de los personajes en el videojuego, dependiendo del nivel de dificultad seleccionado.

Adicionalmente, el videojuego utilizará la información almacenada sobre el puntaje obtenido cada vez que alguien juega, con lo que se podrá generar clasificaciones de los más altos puntajes alcanzados de acuerdo al nivel de dificultad.

1.3. Plan de proyecto

La Gestión de Proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del mismo.

Para la realización del presente proyecto se considera la participación de un solo tesista dentro del ciclo de vida del proyecto.

Los formatos que se presentarán para la planificación realizada en el proyecto se mencionan a continuación.

1.3.1. Estructura de descomposición de trabajo

En la **Figura 1.5** se muestra el EDT del trabajo a desarrollar en el proyecto.

1.3.2. Diagrama de Gantt

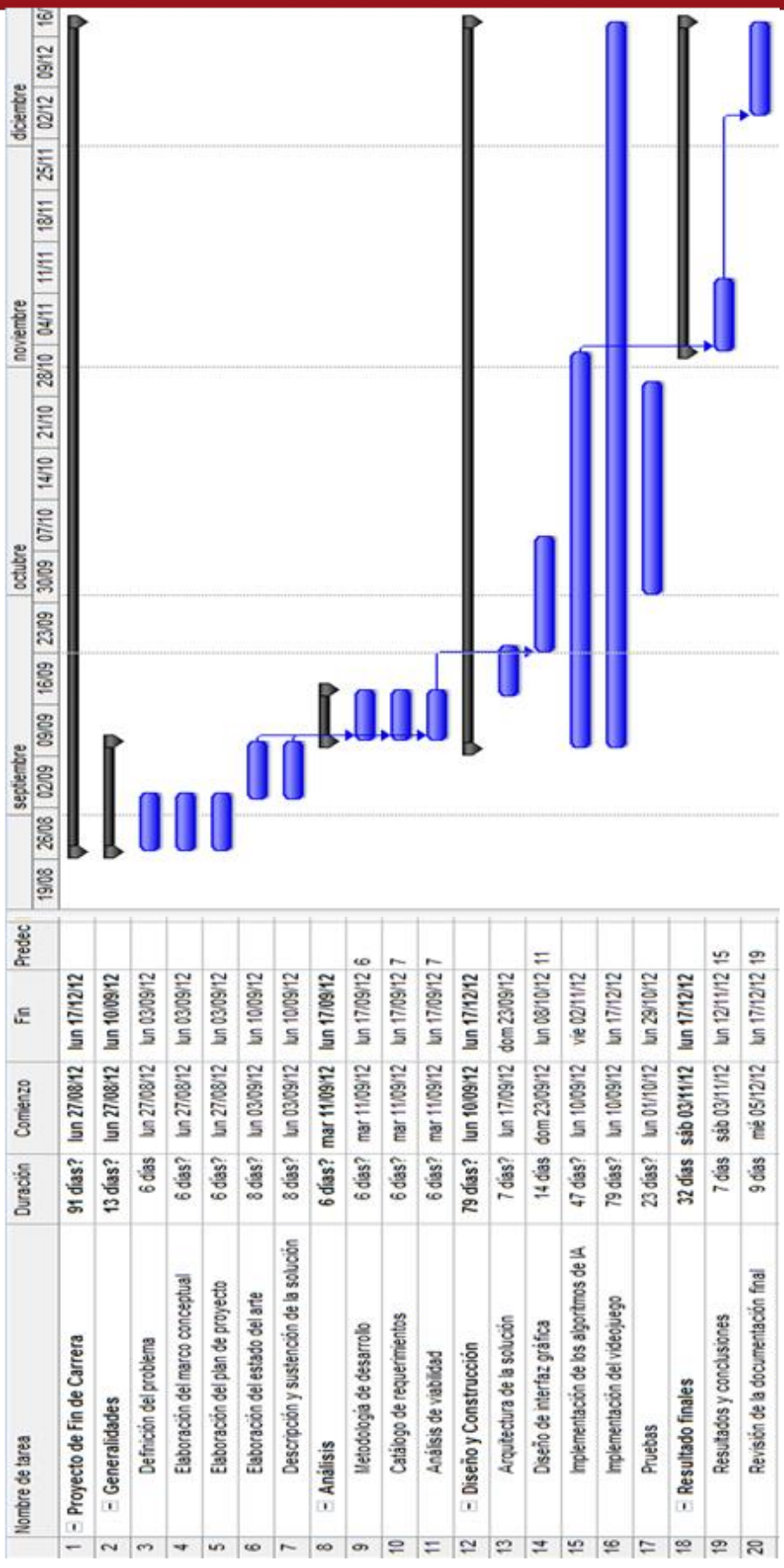
En la **Figura 1.6** se muestra el diagrama de Gantt a utilizar en el presente proyecto con las tareas predecesoras definidas, los recursos utilizados y las fechas asignadas. (Ver **Anexo A**).

1.3.3. Identificación de riesgos del proyecto

Como se explicará en el capítulo 2.3 se ha definido.

Nivel 1	Nivel 2
1 Generalidades	
	1.1 Definición del problema
	1.2 Elaboración del marco conceptual
	1.3 Elaboración del plan de proyecto
	1.4 Elaboración del estado del arte
	1.5 Descripción y sustentación de la solución
2 Análisis	
	2.1 Metodología de desarrollo
	2.2 Catálogo de requerimientos
	2.3 Análisis de viabilidad
3 Diseño y Construcción	
	3.1 Arquitectura de la solución
	3.2 Diseño de interfaz gráfica
	3.3 Implementación de los algoritmos de IA
	3.4 Implementación del videojuego
	3.5 Pruebas
4 Resultados finales	
	4.1 Resultados y conclusiones
	4.2 Revisión de la documentación final

Figura 1.5: Estructura de descomposición de trabajo



	Nombre de tarea	Duración	Comienzo	Fin	Predec
1	**Proyecto de Fin de Carrera**	**91 días?**	**lun 27/08/12**	**lun 17/12/12**	
2	**Generalidades**	**13 días?**	**lun 27/08/12**	**lun 10/09/12**	
3	Definición del problema	6 días	lun 27/08/12	lun 03/09/12	
4	Elaboración del marco conceptual	6 días?	lun 27/08/12	lun 03/09/12	
5	Elaboración del plan de proyecto	6 días?	lun 27/08/12	lun 03/09/12	
6	Elaboración del estado del arte	8 días?	lun 03/09/12	lun 10/09/12	
7	Descripción y sustención de la solución	8 días?	lun 03/09/12	lun 10/09/12	
8	**Análisis**	**6 días?**	**mar 11/09/12**	**lun 17/09/12**	
9	Metodología de desarrollo	6 días?	mar 11/09/12	lun 17/09/12	6
10	Catálogo de requerimientos	6 días?	mar 11/09/12	lun 17/09/12	7
11	Análisis de viabilidad	6 días?	mar 11/09/12	lun 17/09/12	7
12	**Diseño y Construcción**	**79 días?**	**lun 10/09/12**	**lun 17/12/12**	
13	Arquitectura de la solución	7 días?	lun 17/09/12	dom 23/09/12	
14	Diseño de interfaz gráfica	14 días	dom 23/09/12	lun 08/10/12	11
15	Implementación de los algoritmos de IA	47 días?	lun 10/09/12	vie 02/11/12	
16	Implementación del videojuego	79 días?	lun 10/09/12	lun 17/12/12	
17	Pruebas	23 días?	lun 01/10/12	lun 29/10/12	
18	**Resultado finales**	**32 días**	**sáb 03/11/12**	**lun 17/12/12**	
19	Resultados y conclusiones	7 días	sáb 03/11/12	lun 12/11/12	15
20	Revisión de la documentación final	9 días	mié 05/12/12	lun 17/12/12	19

Figura 1.6: Diagrama de Gantt

En cuanto a la definición de riesgos que puedan ocurrir durante el transcurso del desarrollo del presente proyecto se procede a realizar un análisis de diversos aspectos que puedan impactar en la continuidad de la implementación presentada.

Específicamente se han definido 4 rubros que pueden impactar al proyecto los cuales son: Falta de experiencia con las herramientas de desarrollo, Falta de tiempo para probar el producto, Fallas de programación e Incumplimiento de objetivos del software.

A continuación se los riesgos mencionados y sus posibles soluciones:

- **Falta de experiencia con las herramientas de desarrollo (Tabla 1.1)**

En la siguiente tabla se describe la estrategia aplicada a este tipo de riesgo:

Ítem	**Descripción**
Magnitud	**Medio**
Descripción	Falta de experiencia del desarrollador en las herramientas de necesarias para el desarrollo del proyecto.
Impacto	Falta de calidad del producto final si no se utilizan adecuadamente las herramientas.
Indicadores	Necesidad de uso de la herramienta para culminación del proyecto.
Estrategia de mitigación	Planificar cursos de aprendizaje de las herramientas en las etapas iniciales del proyecto.
Plan de contingencia	Entrenarse con manuales de las herramientas disponibles en Internet.

Tabla 1.1: Falta de experiencia con las herramientas de desarrollo

- **Fallas de programación (Tabla 1.2)**

En la siguiente tabla se describe la estrategia aplicada a este tipo de riesgo:

Ítem	Descripción
Magnitud	**Medio**
Descripción	Fallas en el sistema por programación que causa que el usuario obtenga errores o no obtenga lo que está solicitando.
Impacto	Baja calidad del producto al no cumplir con funcionalidad.
Indicadores	Retrasos en la construcción o entrega del código.
Estrategia de mitigación	Incluir pruebas de software en las fases iniciales de programación.
Plan de contingencia	En caso de presentarse, realizar una inversión de tiempo en los errores de programación.

Tabla 1.2: Fallas de programación

- **Falta de tiempo para probar el producto (Tabla 1.3)**

En la siguiente tabla se describe la estrategia aplicada a este tipo de riesgo:

Ítem	Descripción
Magnitud	**Medio**
Descripción	Significa que no se dispone del tiempo que se requiere para probar el software.
Impacto	El producto no posee la garantía de su funcionalidad si no es posible probarlo.
Indicadores	Falta de cumplimiento del plan del proyecto en las fechas indicadas. Si el proyecto se va retrasando, la etapa de pruebas se va postergando también.
Estrategia de mitigación	Planificar la etapa de validación dentro del tiempo que se tiene previsto para la elaboración de este proyecto.
Plan de contingencia	De llegar a ocurrir este riesgo, se debe realizar una inversión de tiempo extra en realizar las pruebas.

Tabla 1.3: Falta de tiempo para probar el producto

- **Incumplimiento de objetivos del software (Tabla 1.4)**

En la siguiente tabla se describe la estrategia aplicada a este tipo de riesgo:

Ítem	Descripción
Magnitud	**Alto**
Descripción	El videojuego está diseñado para que pueda servir como ayuda al desarrollo intelectual de los jugadores. Si no se logra un incremento de las habilidades cognitivas acordadas podría decirse que el programa no logra los objetivos para los que fue diseñado.
Impacto	Puede producir la no aprobación del producto del tema de tesis.
Indicadores	Observaciones del asesor de tesis al probar los prototipos.
Estrategia de mitigación	Las actividades a realizar con respecto a las funcionalidades y al diseño del software están avaladas por el asesor de tesis desde el momento en que se inicia el proyecto. Otra estrategia es utilizar herramientas de programación que impliquen la incorporación de cambios fácilmente.
Plan de contingencia	Cambio de las actividades parcial o totalmente, involucrando una inversión de tiempo considerable dependiendo del caso.

Tabla 1.4: Incumplimiento de objetivos del software

1.4. Estado del arte

En la actualidad existen juegos que son utilizados como ejercicios recreativos enfocados al apoyo del a ejercitación del razonamiento lógico, la memoria, así como el desarrollo del reconocimiento de formas y figuras, la capacidad para procesar tamaños, direcciones y relaciones espaciales, entre otros. Entre los principales juegos podemos mencionar los siguientes [LUM001]:

1.4.1. Speed Match

Consiste en un juego de ritmo rápido que se enfoca en ejercitar la habilidad del cerebro para procesar la información y por lo tanto mejora una amplia variedad de habilidades cognitivas. Se sabe que el cerebro sigue creando nuevas neuronas durante toda la vida. El ejercicio y la estimulación cognitiva pueden afectar el número de nuevas células creadas en el cerebro. El juego consiste en los siguientes pasos:

- Recordar el símbolo que aparece en pantalla.
- Comparar cada símbolo con el que apareció inmediatamente anterior a éste e identificar si son iguales.
- Se tiene un período limitado de tiempo para conseguir la mayor cantidad de respuestas correctas posibles.
- A mayor cantidad de respuestas correctas seguidas, mayor cantidad de puntos se obtendrán.

A continuación en la F**igura 1.7** se presentan imágenes del juego Speed Match:

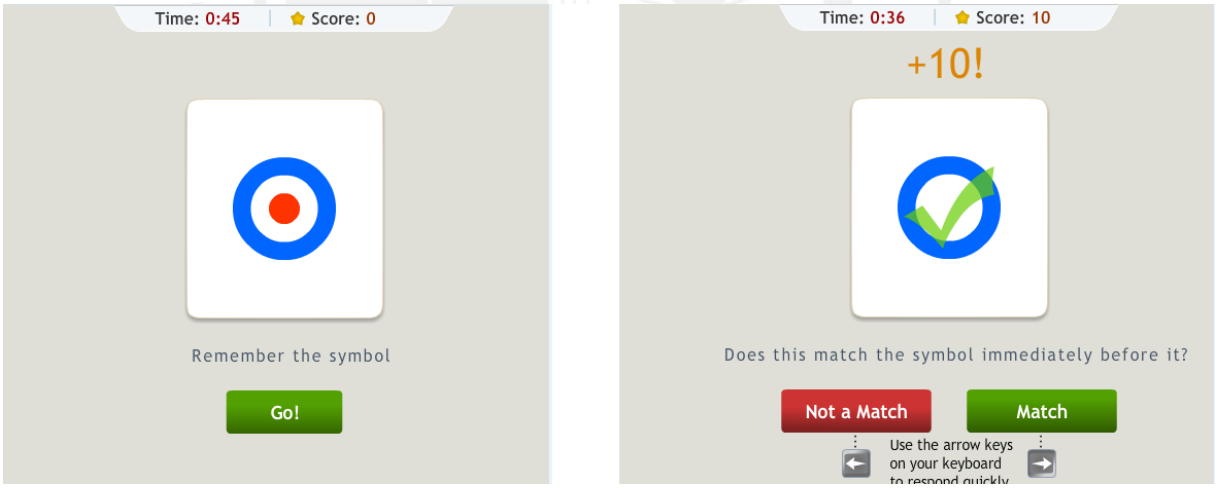


Figura 1.7: Speed Match
Tomado de: www.lumosity.com

1.4.2. Memory Matrix

Es un juego que ejercita el recuerdo espacial, es decir, la habilidad de recordar la localización exacta de un objeto.

Las investigaciones han demostrado que el entrenamiento con tareas de recuerdo espaciales puede mejorar la memoria de trabajo y atención. El juego consiste en los siguientes pasos:

- Un patrón aparece en pantalla.
- Repetir el patrón indicando los espacios correctos.

A continuación en la **Figura 1.8** se presentan imágenes del juego Memory Matrix:

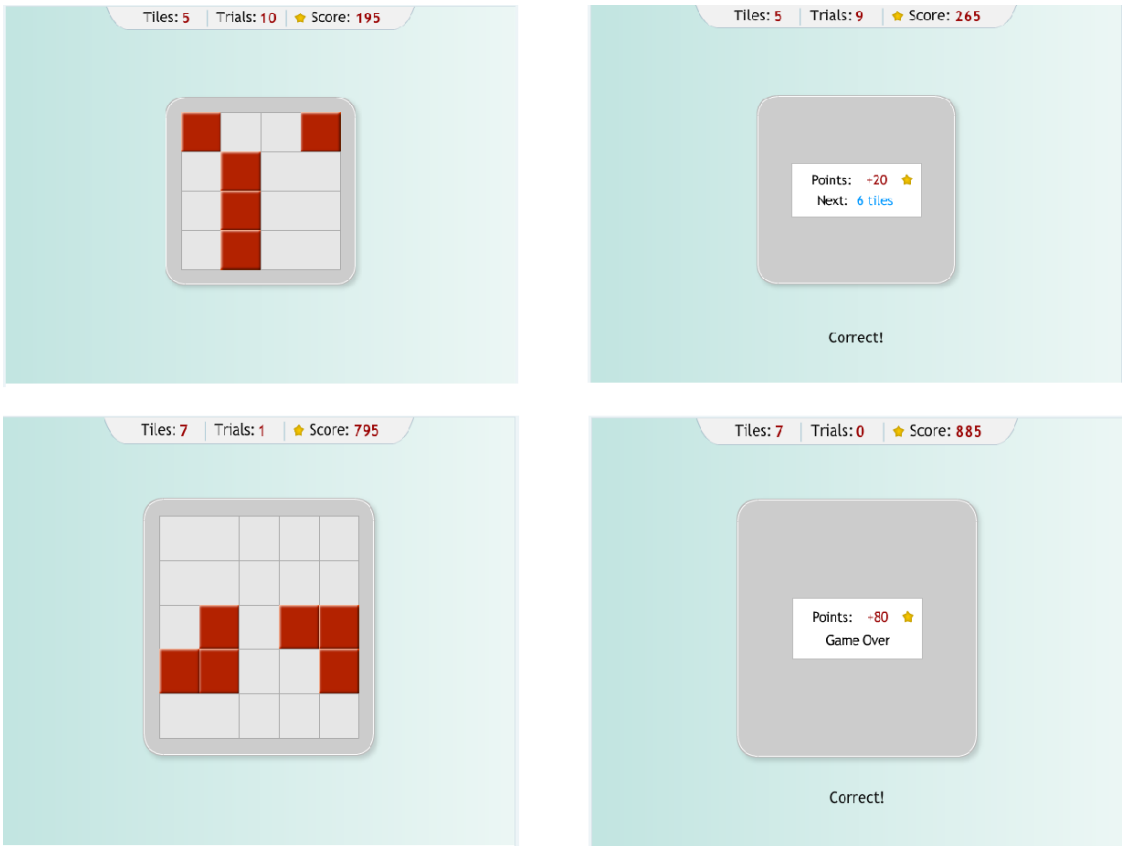


Figura 1.8: Memory Matrix
Tomado de: www.lumosity.com

1.4.3. Eagle Eye

Es un juego de atención visual que puede ayudar a percibir elementos de nuestro entorno de una forma más rápida y precisa.

- Un ave y un número aparecen en pantalla.

- Indicar la ubicación exacta donde apareció el ave.
- Indicar el número aparecido junto al ave.

A continuación en la **Figura 1.9** se presentan imágenes del juego Eagle Eye:

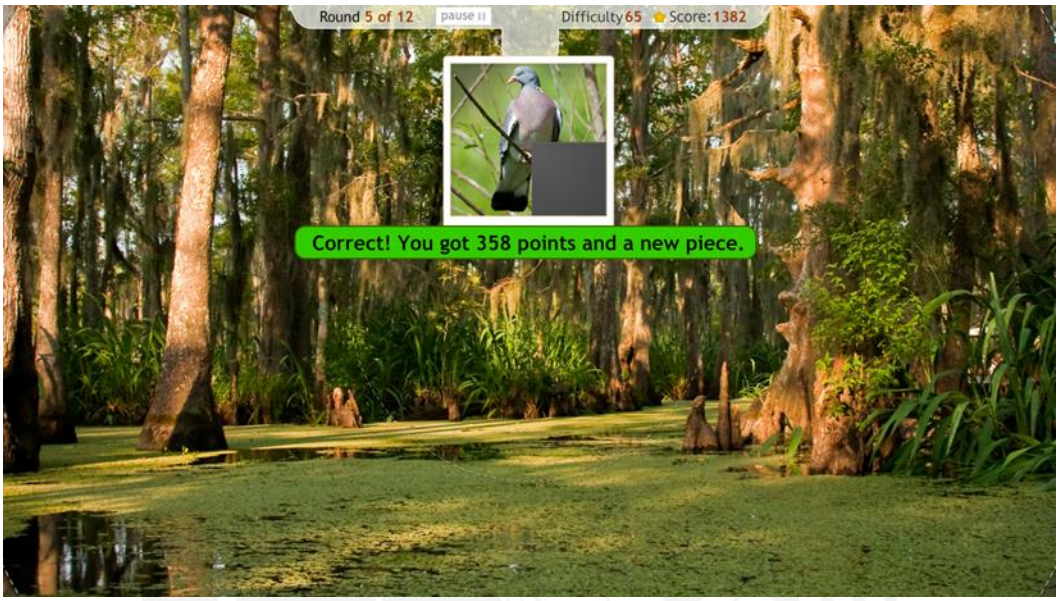


Figura 1.9: Eagle Eye
Tomado de: www.lumosity.com

1.4.4. Big Brain Academy: Wii Degree

Es un videojuego lanzado para la consola Wii de Nintendo. En Big Brain Academy el peso del cerebro es medido mediante una serie de pruebas. Cuanto más pese el cerebro, más inteligente es, o mejor es el tiempo de reacción. No hay un mecanismo de juego, puesto que es un conjunto de puzles sin que ninguno tenga mayor prioridad sobre los otros. Hay tres modos de juego: Prueba, Práctica y Competición (multijugador con WiFi), los cuales se describirán a continuación.

- **Prueba**: Aquí es donde se pone a prueba la habilidad del jugador en diferentes etapas del juego, y en este modo de juego es donde se muestra el peso cerebral dependiendo de las condiciones que el juego va poniendo.
- **Práctica**: Para pasar el tiempo, o practicar en aquellas pruebas en las que uno tiene dificultades. En este modo no se evalúa nada, pero las actividades tienen una duración de sólo 60 segundos.

- **Competición**: Se puede competir hasta con 7 jugadores más. Se compite por ver quién es evaluado con la mejor capacidad cerebral de todos.

Tenemos cinco áreas de conocimiento dentro del modo Repaso, y dentro de cara área encontramos tres pruebas. Cada prueba dura un minuto y tendremos diferentes objetivos a completar, según nuestros aciertos, fallos y velocidad, se nos otorgarán al final un peso cerebral, y cuanto mayor sea más posibilidades tendremos de obtener las medallas más valiosas (bronce, plata, oro).

Los bloques de pruebas serán Asociación, Lógica, Cálculo, Análisis y Memoria como puede observarse en la F**igura 1.10**:

Figura 1.10: Big Brain Academy
Tomado de: www.meristation.com

1.4.5. Brain Age: Train your brain in minutes a day

Es un videojuego de lógica y puzles desarrollado y distribuido por Nintendo para la plataforma Nintendo DS. Brain Training engloba una serie de ejercicios de diversa dinámica, desde ejercicios de cálculo y retención de datos hasta ejercicios de lectura y

ortografía, pasando por los muy conocidos sudokus, todo ello con el fin de ejercitar la mente.

Al principio únicamente es posible realizar unos pocos ejercicios. No obstante, poco a poco se van desbloqueando nuevos ejercicios de forma paulatina según se va jugando con el tiempo. Junto a los ejercicios también es posible realizar una prueba o examen, el cual está compuesto por varios ejercicios disponibles y será el que determine la "edad cerebral" del jugador. También están disponibles unos gráficos donde es posible consultar los datos de los resultados de los distintos ejercicios y de los exámenes realizados, pudiendo el jugador apreciar así su evolución en cada uno de los distintos ejercicios y en los exámenes generales. Ver **Figura 1.11**:

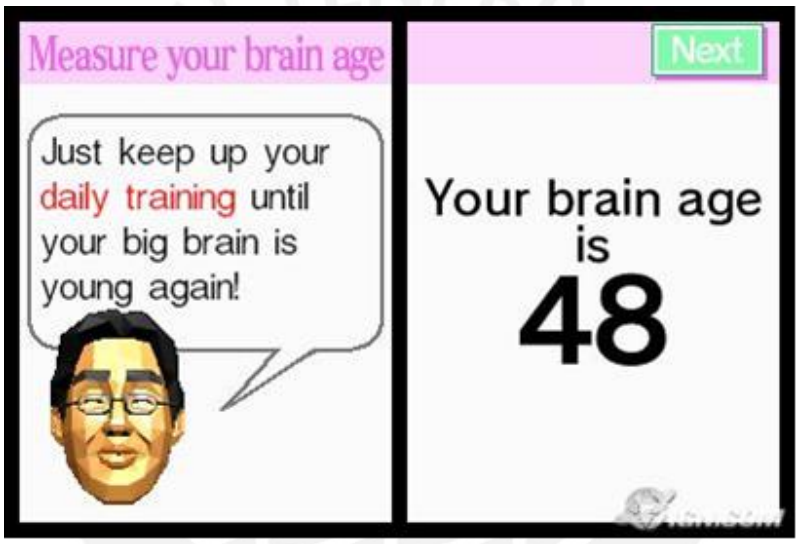


Figura 1.11: Brain Age

Tomado de: www.meristation.com

1.4.6. Raindrops

Es un juego que utiliza la aritmética para mejorar las habilidades de resolución de problemas. Ver **Figura 1.12**:

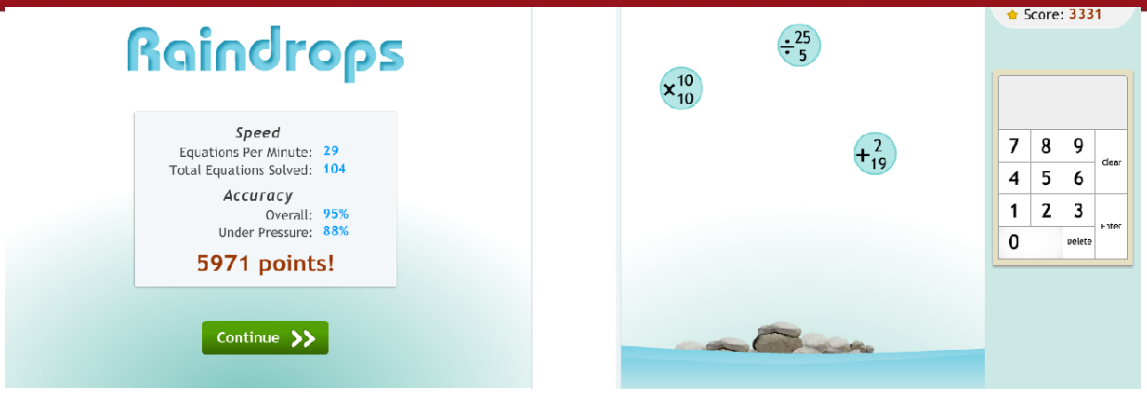


Figura 1.12: Raindrops

Tomado de: www.lumosity.com

1.4.7. Lost in Migration

Este juego ejercita las habilidades de atención ya que permite mejorar la concentración ayudando a evitar las distracciones y desarrollar la productividad en el trabajo.

Las investigaciones demuestran que luego de 33 sesiones de cinco minutos de entrenamiento de atención visual, se incrementa la habilidad para detectar objetos en la visión periférica. Ver **Figura 1.13**:

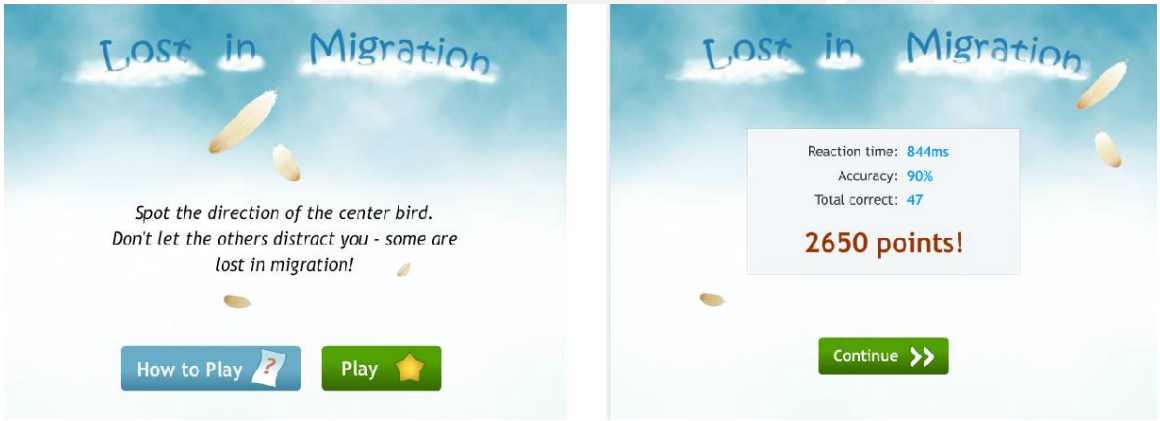


Figura 1.13: Lost in Migration

Tomado de: www.lumosity.com

1.5. Descripción y sustentación de la solución

En esta sección del documento se explicará la solución que se propone para el problema planteado, así como las ventajas que ofrece.

Como se mencionó anteriormente en la sección 1.1 Definición del problema, se considera que es recomendable involucrar a la tecnología en un proyecto que ayude a estimular y mejorar las habilidades cognitivas de las personas. Es por esto que en la actualidad se han comenzado a utilizar los videojuegos como una herramienta que estimule el desarrollo cognitivo y el aprendizaje en las personas.

En el Perú, actualmente el uso de videojuegos con fines educativos o como parte de tratamientos médicos no se encuentra muy difundido, por lo que el uso de estos representa una alternativa económica y viable de poder estimular y mejorar habilidades cognitivas. Asimismo, al contener esta propuesta elementos lúdico-recreativos, permite una mayor aceptación por parte de las personas que lo utilizan.
A continuación en la **Tabla 1.5** se realiza una comparación con otros videojuegos que existen en el mercado actual:

Calificación	Videojuego propuesto en la tesis	Speed Match	Memory Matrix	Big Brain Academy
Jugabilidad	Bueno	Regular	Regular	Bueno
Gráficos	Bueno	Regular	Regular	Bueno
Sonido	Regular	Regular	Regular	Regular
Innovación	Bueno	Bueno	Regular	Bueno
Aporte educacional	Muy bueno	Regular	Regular	Bueno

Tabla 1.5: Análisis comparativo con otros videojuegos del Mercado

En cuanto a la categoría de aporte educacional se considera que el videojuego propuesto en la presente tesis abarcará muchos aspectos que otros no toman en cuenta por lo que ofrece mayores ventajas sobre los existentes en la actualidad.

Las ventajas que ofrecerá el videojuego a desarrollar, así como las habilidades que ayudarán a desarrollar se pueden separar en los siguientes aspectos:

- **Memoria:**

  - Recordar la ubicación exacta de diversos objetos.
  - Retener y comprender varias ideas al mismo tiempo.
  - Aprender nuevas estrategias de una manera más rápida y precisa.

- **Atención:**

  - Mantenerse enfocado en las tareas más importantes.
  - Evitar distracciones.
  - Mayor concentración mientras se aprende algo nuevo.
  - Mejorar la productividad y precisión en todo momento.

- **Velocidad:**

  - Toma de decisión en situaciones de tiempo críticas.
  - Rapidez en la comprensión del proceso cognitivo.
  - Adaptación a los ambientes de cambios.
  - Reacción rápida y mejora en los reflejos.

- **Flexibilidad:**

  - Realizar múltiples tareas rápida y eficientemente.
  - Saber enfrentar nuevos tipos de desafíos más complicados.
  - Predisposición al descubrimiento de nuevas dificultades.

- **Solución de problemas:**

  - Determinar el mejor camino de acción.
  - Realizar estimaciones rápida y eficientemente.
  - Diseccionar argumentos complejos.

Ante la importancia que tiene la educación en el desarrollo de las personas y por lo tanto de la sociedad se considera necesario el desarrollo de este videojuego pues permitirá de una manera recreativa mejorar sustancialmente las habilidades en los diferentes aspectos mencionados anteriormente.

2. CAPÍTULO 2: ANÁLISIS

En el presente capítulo se detalla la metodología a utilizar para el desarrollo del producto tales como la metodología de gestión de proyectos, la metodología de desarrollo y los métodos algorítmicos utilizados para la presente solución.

También se presenta el listado de los requerimientos funcionales del proyecto, los cuales están vinculados estrechamente con la funcionalidad que tendrá el sistema, y los requerimientos no funcionales, los cuales hacen mención a aspectos técnicos como tiempo de respuesta y tecnologías a utilizar.

Adicionalmente, se define la viabilidad del sistema en términos económicos y técnicos para posteriormente definir la asignación de funciones y las restricciones de tiempo para el actual proyecto.

2.1. Metodología aplicada para el desarrollo de la solución

Para lograr la realización del proyecto de fin de carrera se ha visto necesario contemplar dos enfoques, uno orientado a la Gestión del Proyecto y otro a la Implementación del

Software, de modo que ambos se complementen y permitan llevar la planificación, organización, dirección y control del proyecto e implementación del software con éxito.

2.1.1. Metodología de gestión del proyecto

Para cubrir el frente de la gestión del proyecto, se usará PMBOK (Project Management Body of Knowledge), el cual es una guía para la administración y gestión de proyectos, y fue desarrollada por el Project Management Institute (PMI). La razón por la que se ha escogido PMBOK es porque contiene el conjunto de las mejores prácticas para la Gestión de Proyectos que permitirán tener un mayor control acerca de la calidad de los entregables durante el proyecto, así como seguir oportunamente el cronograma de los entregables. Como punto adicional se tiene que el tesista se encuentra bastante familiarizado con PMBOK.

PMBOK reconoce cinco grupos de procesos básicos los cuales se mencionan a continuación [PMI001]: Iniciación, Planificación, Ejecución, Control Reporte y Cierre.

Para la realización del presente proyecto se ha planteado seguir estos 5 pasos definidos por el PMBOK y desarrollar las tareas de acuerdo a la guía de gestión lo cual permite tener un mayor control acerca de la calidad de los entregables durante el proyecto, así como seguir oportunamente el cronograma de los mismos. En la **Figura 2.1** se muestran los principales procesos del PMBOK:

Figura 2.1: Procesos del PMBOK

Tomado de: www.pmi.org

Adicionalmente se definen entregables para cada una de las áreas de conocimiento de la dirección de proyectos según los lineamientos indicados en el PMBOK [PMI002], y para el presente proyecto se han considerado los que se listan en la T**abla 2.1** que se presenta a continuación.

Área de conocimiento	Actividades	Entregables
Gestión del Alcance del Proyecto	✓ Definición del alcance ✓ Crear la estructura de desglose de trabajo (EDT) ✓ Verificación del alcance ✓ Control del alcance	✓ Alcance del proyecto (Ver **capítulo 1.2.2**) ✓ Estructure de desglose de trabajo (EDT) (Ver **capítulo 1.3**)
Gestión del Tiempo del Proyecto	✓ Definición de las actividades involucradas para el desarrollo del videojuego ✓ Estimación de la duración de las actividades ✓ Desarrollo del cronograma ✓ Control del cronograma	✓ Plan de proyecto (Ver **Anexo A**)
Gestión de los Costos del Proyecto	✓ Estimación de costos ✓ Preparación del presupuesto de costos ✓ Control de costos	✓ Documento de análisis económico del videojuego (Ver **capítulo 2.3.3**)
Gestión de los Riesgos del Proyecto	✓ Planificación de la gestión de riesgos ✓ Identificación de los riesgos ✓ Análisis cualitativo de los riesgos ✓ Análisis cuantitativo de los riesgos ✓ Seguimiento y control de riesgos	✓ Documento del análisis de riesgos del videojuego (Ver **capítulo 1.3**)

Tabla 2.1: Entregables por cada área de conocimiento PMBOK

Como puede verse en la **Figura 2.2** se cubrirán 4 de las 7 áreas de conocimiento que presente el PMBOK. Estas áreas cubiertas son: Gestión del Alcance del Proyecto, Gestión del Tiempo del Proyecto, Gestión de los Riesgos del Proyecto y Gestión de los Costos del Proyecto.

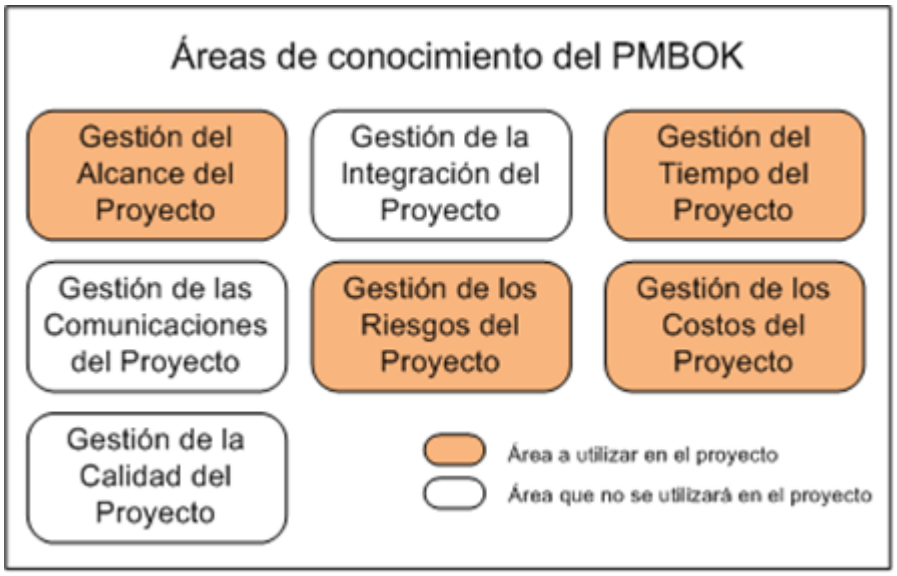


Figura 2.2: Áreas de conocimiento del PMBOK

2.1.2. Metodología de desarrollo de software

Para atender el frente correspondiente a la implementación del software la metodología de desarrollo a utilizar será RUP (Rational Unified Process) debido a su difusión y reconocimiento, junto con UML (Unified Modeling Language) que constituye la propuesta para el modelamiento para el análisis y diseño, y en conjunto para la implementación y documentación del sistema. [RUP001]

Se establecerán las etapas de Análisis, Diseño y Construcción como las más importantes a considerar, y se tomará como prioridad los documentos de Especificación de Requerimientos, Análisis, Diseño, Arquitectura del sistema para iniciar con el desarrollo de la solución.

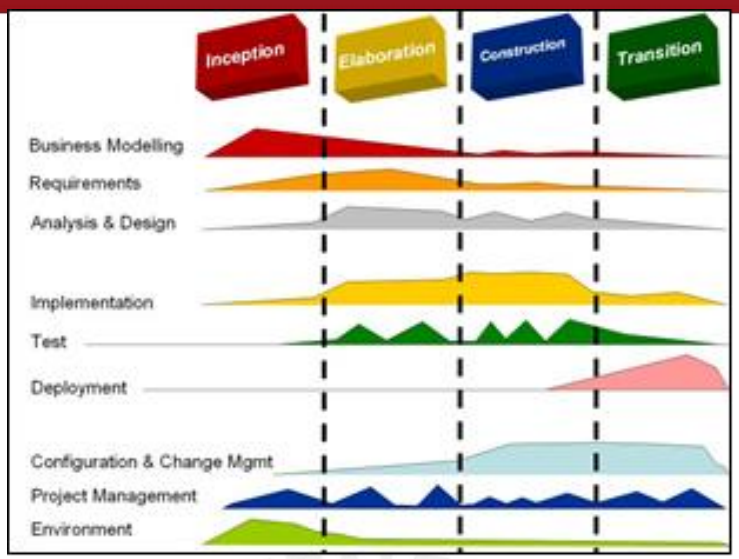


Figura 2.3: Fases del RUP
Tomado de: www.ibm.com

Otra razón por la cual se utilizará RUP para la presente solución es que se trata de un proyecto complejo, por lo cual la documentación que se elabore bajo este enfoque será de gran ayuda para facilitar el entendimiento sobre el concepto y funcionamiento de dichos elementos.

RUP constituye una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo. La ventaja principal de RUP está en que se basa todo en las mejores prácticas que se han intentado y probado en el campo.

RUP se divide en las 4 fases (**Figura 2.3**) indicadas a continuación:

- **Inicio:** define el alcance del proyecto.
- **Elaboración:** definición, análisis, diseño.
- **Construcción:** implementación.
- **Transición:** fin del proyecto y puesta en producción.

A continuación en la **Tabla 2.2** se presenta la lista de entregables que se tendrán en cada una de las fases de la metodología RUP aplicada al proyecto de fin de carrera:

Fase del RUP	Entregables
Fase de Inicio	✓ Se establece la oportunidad y alcance del videojuego. ✓ Se identifican todas las entidades externas con las que se trata y se define la interacción a un alto nivel de abstracción: - Identificar todos los casos de uso - Describir algunos en detalle ✓ Entregables: - Un documento de visión general (Ver **Anexo B**) - Requerimientos generales del proyecto - Características principales del videojuego - Restricciones del videojuego - Modelo inicial de casos de uso - Glosario (Ver **Anexo C**) - Identificación inicial de riesgos - Definir el plan de proyecto - Uno o más prototipos del videojuego
Fase de Elaboración	✓ Analizar el dominio del problema que el videojuego busca solucionar ✓ Establecer una arquitectura base sólida basada en capas ✓ Desarrollar el plan de proyecto ✓ Eliminar los elementos de mayor riesgo para el desarrollo exitoso del videojuego ✓ Entregables: - Modelo de casos de uso (Ver **Anexo D**) - Otros requerimientos no funcionales o no asociados a casos de uso - Descripción de la Arquitectura del videojuego - Un prototipo ejecutable de la arquitectura. - Lista revisada de riesgos y del caso de negocio - Plan de desarrollo para el resto del proyecto - Un manual del videojuego preliminar para el jugador (Ver **Anexo E**)

Fase del RUP	Entregables
Fase de Construcción	✓ En esta fase todos los componentes restantes se desarrollan e incorporan al producto ✓ Entregables: - El producto de videojuego integrado y corriendo en la plataforma adecuada ✓ Manuales finales del videojuego a ser incluidos con la demo
Fase de Transición	✓ Finalmente se produce la distribución del videojuego definitivo ✓ Incluye: - Distribución del videojuego en el mercado

Tabla 2.2: Entregables por cada fase del RUP

2.1.3. Métodos algorítmicos

Adicionalmente a las metodologías de gestión de proyectos y desarrollo de software, se presenta a continuación una investigación del método algorítmico utilizado para el desarrollo de juegos enfocados a la solución de la problemática planteada en el presente proyecto.

El proyecto incluirá la implementación de algoritmos basados en Inteligencia Artificial para manejar el movimiento y toma de decisiones de los personajes en el videojuego, dependiendo del nivel de dificultad seleccionado. En vista de esto se procedió a investigar acerca del modelamiento de juegos basados en Inteligencia Artificial.

El modelo presentado para el presente proyecto divide la tarea de la Inteligencia Artificial básicamente en 3 secciones: movimiento, toma de decisión y estrategia como puede verse en la **Figura 2.4**.

Las dos primeras secciones contienen algoritmos que trabajan en la base de un solo jugador y la última sección opera en un conjunto o equipo.

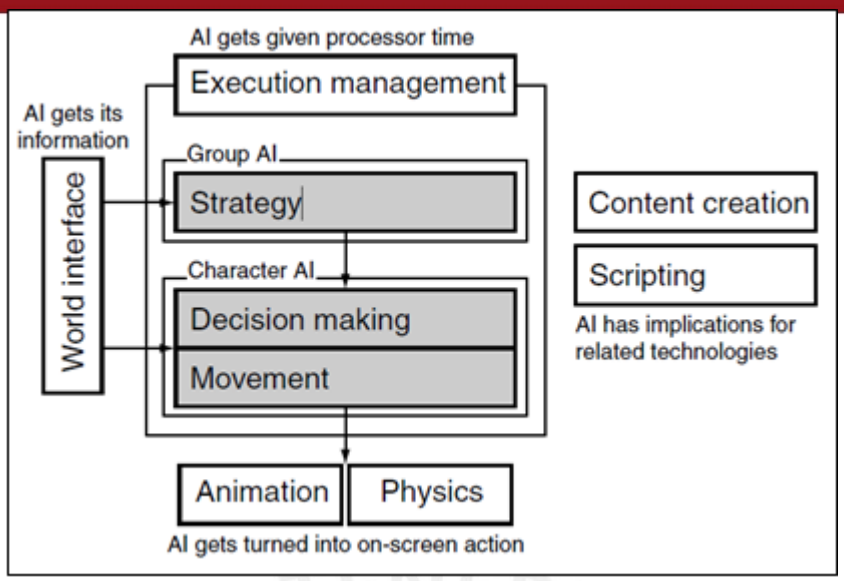


Figura 2.4: El modelo de Inteligencia Artificial

Tomado de: www.ai4g.com

No todos los juegos requieren todos los niveles del modelo de Inteligencia Artificial (IA). Los juegos de mesa como “Ajedrez” requieren solo del nivel de estrategia ya que los personajes en este juego individualmente (las fichas) no toman sus propias decisiones y no necesitan preocuparse en cómo moverse.

Por otro lado, en otros juegos la estrategia no es un factor indispensable. Los personajes en un juego de plataformas son puramente reactivos realizando sus propias decisiones simples y actuando tan solo en base a ellas, es por esto que no hay plena confianza de que los personajes enemigos del juego realicen su mejor trabajo posible para frustrar al jugador. A continuación se presenta algunos elementos relacionados del modelo de IA.

- **Movimiento**

El movimiento refiere a los algoritmos que toman decisiones en algún tipo de movimiento. Por ejemplo, cuando un personaje enemigo no armado necesita atacar a nuestro jugador, éste primero se acerca al jugador directamente. Cuando esté lo suficientemente cerca, éste puede realizar el ataque. La decisión de atacar es llevada a cabo por una serie de algoritmos de movimiento que conducen a la posición del jugador. Solo recién allí puede ser reproducida la animación del ataque y la salud del jugador puede ser reducida.

Pero los algoritmos de movimiento pueden llegar a ser más complejos. Un personaje puede necesitar evadir obstáculos en su camino o incluso trabajar el camino a través de una serie de habitaciones. Por ejemplo, un guardia enemigo, ante la presencia de nuestro jugador, responderá mediante la activación de alguna alarma. Esto puede requerir un recorrido hasta el punto de alarma más cercano al guardia, lo cual puede ser una larga distancia y puede involucrar una navegación compleja alrededor de obstáculos o a través de corredores.

Muchas acciones son llevadas a cabo usando directamente la animación. Por ejemplo si en un juego de simulación social, nuestro jugador está sentado en una mesa con comida al frente y necesita ejecutar la acción de comer, entonces simplemente esta acción es ejecutada y no se necesita inteligencia artificial. En cambio si el personaje se encuentra detrás de una puerta y necesita realizar la acción de comer entonces los movimientos basados en Inteligencia Artificial requieren primero guiarlo a la silla para recién poder comer.

Para el presente proyecto se utilizará el algoritmo de búsqueda A* el cual es un algoritmo de búsqueda en grafos. Él método busca el camino en un grafo de un vértice inicial hasta un vértice final. Él es la combinación de aproximaciones heurísticas como del algoritmo de búsqueda Best First (primero el mejor). La razón por la que se opta por este método es que ha sido muy utilizado con éxito en diversos juegos.

- **Toma de decisión**

La toma de decisión involucra al personaje a trabajar en lo siguiente que debe realizar. Generalmente, cada personaje tiene un rango de distintos comportamientos que ellos pueden escoger a realizar: atacar, mantener su posición, esconderse, explorar, patrullar, etc. El sistema de toma de decisión necesita trabajar en decidir cuál de estos comportamientos es el más adecuado a realizar en determinado momento del juego. El comportamiento escogido puede llegar a ser ejecutado usando movimiento de Inteligencia Artificial y tecnología de animación.

Un personaje puede tener reglas simples para seleccionar un comportamiento. Por ejemplo, en algún juego de aventura algunos animales de granja mantendrán su posición calmada mientras el jugador permanezca alejado de ellos, sin embargo ellos huirán en caso el jugador se acerque demasiado.

En el otro extremo, los enemigos de un juego de disparos en primera persona despliegan decisiones complejas donde ellos ejecutarán una serie de diversas estrategias para alcanzar al jugador tales como lanzar granadas y ocultarse para evitar el fuego del jugador.

Para el presente proyecto se utilizará las Máquinas de Estado Finitos para la toma de decisiones las cuales son modelos computacionales que realizan cómputos en forma automática sobre una entrada para producir una salida.

- **Estrategia**

Muchos juegos se basan solo en las dos primeras secciones mencionadas. Sin embargo para coordinar un equipo la estrategia de Inteligencia Artificial es requerida.

En esta categoría se encuentran los algoritmos de IA que no controlan un solo jugador sino que influye en el comportamiento de un conjunto de jugadores. Cada personaje del grupo puede tener sus propios algoritmos de toma de decisiones y de movimiento pero toda su decisión se verá influenciada por una estrategia de grupo.

En un juego de guerra en tercera persona, los enemigos trabajan en equipo para contrarrestar y eliminar al jugador. Esto también es seguido en juegos de acción tácticos más recientes con incrementadas y sofisticadas acciones.

Para el presente proyecto se utilizarán Algoritmos Evolutivos para lo referente a las estrategias de juego debido a que en estos algoritmos se mantiene un conjunto de entidades que representan posibles soluciones, las cuales se mezclan, y compiten entre sí, de tal manera que las más aptas son capaces de prevalecer a lo largo del tiempo, evolucionando hacia mejores soluciones cada vez.

2.2. Identificación de requerimientos

Basado en la definición del problema al cual se quiere dar solución y al alcance definido para el presente proyecto se procede a definir los requerimientos tanto funcionales como no funcionales para el presente videojuego. Los requerimientos que se han identificado son los necesarios para cubrir todas las características del proyecto mencionadas en el capítulo 1.

Se han identificado los siguientes requerimientos para la solución planteada:

2.2.1. Requerimientos Funcionales

Los requerimientos funcionales se han organizado según los módulos con los que contará el videojuego a desarrollarse. El videojuego estará compuesto de 2 módulos los cuales son los siguientes: Módulo de Juego (**Tabla 2.3**) y Módulo de Configuración (**Tabla 2.4**).

Adicionalmente para cada requerimiento se procede a colocarle un código descriptivo único el cual consiste en las 3 iniciales del módulo (“JUE” para el módulo de Juego y “CON” para el módulo de Configuración) seguido de un número correlativo de 3 dígitos. También por cada requerimiento se define la prioridad que éste tendrá en el proyecto (Alta, Media, Baja), la Dificultad en cuanto a su desarrollo (Alta, Media, Baja) y su tipo dependiendo si el requerimiento es Exigible (RE) o Deseable (RD).

A continuación se presentan los requerimientos funcionales que tendrá el presente videojuego.

- **Módulo de Juego**

Código	Descripción	Prioridad	Dificultad	Tipo
JUE001	El videojuego permitirá iniciar una nueva partida.	Alta	Alta	RE
JUE002	El videojuego permitirá pausar una partida.	Alta	Media	RE
JUE003	El videojuego permitirá grabar una partida.	Media	Alta	RD
JUE004	El videojuego permitirá abandonar una partida.	Alta	Baja	RE
JUE005	El videojuego permitirá cargar una partida grabada.	Media	Alta	RE

Código	Descripción	Prioridad	Dificultad	Tipo
JUE006	El videojuego permitirá enviar un e-mail al jugador con los resultados del mismo.	Alta	Baja	RD
JUE007	El videojuego permitirá ir acumulando puntos por cada objetivo alcanzado dentro de cada nivel del juego.	Alta	Media	RD
JUE008	El videojuego permitirá guardar el nombre y puntuación total obtenida por el jugador al final del juego.	Alta	Media	RD
JUE009	El videojuego permitirá visualizar un video introductorio inicial sobre la historia del juego.	Media	Alta	RE

Tabla 2.3: Requerimientos del Módulo de Juego

- **Módulo de Configuración**

Código	Descripción	Prioridad	Dificultad	Tipo
CON001	El videojuego permitirá configurar los botones a utilizar en el juego.	Alta	Alta	RE
CON002	El videojuego permitirá configurar la dificultad del juego.	Alta	Alta	RD
CON003	El videojuego permitirá mostrar las mejores puntuaciones obtenidas (High-Score).	Baja	Alta	RD
CON004	El videojuego permitirá cambiar el idioma del juego.	Baja	Alta	RE
CON005	El videojuego permitirá mostrar los créditos del juego.	Baja	Baja	RD

Código	Descripción	Prioridad	Dificultad	Tipo
CON006	El videojuego permitirá ajustar el brillo y contraste del juego.	Baja	Media	RD
CON007	El videojuego permitirá ajustar si jugar en modo Pantalla completa o no.	Media	Media	RE
CON008	El videojuego permitirá activar y desactivar el audio del juego.	Media	Media	RE

Tabla 2.4: Requerimientos del Módulo de Configuración

2.2.2. Requerimientos No Funcionales

Los requerimientos no funcionales se han organizado en una única tabla en donde cada requerimiento cuenta con un código único que viene dado por las siglas “RNF” seguido de un número correlativo de 3 dígitos. Entre los requerimientos no funcionales del sistema se tienen los siguientes (**Tabla 2.5**):

Código	Descripción	Prioridad
RNF001	El videojuego será soportado a partir del sistema operativo Windows XP en adelante.	Alta
RNF002	Se utilizará RUP como metodología de desarrollo.	Alta
RNF003	El PC cliente debe tener memoria de 128MB como mínimo para ejecutar el videojuego.	Alta
RNF004	El PC cliente deberá tener espacio disponible de 100MB en su disco duro para almacenar el videojuego.	Alta
RNF005	El videojuego será desarrollado en el lenguaje Python	Alta
RNF006	El videojuego será intuitivo y de fácil aprendizaje.	Alta
RNF007	Se presentará una documentación de guía para el jugador.	Alta

Tabla 2.5: Requerimientos No Funcionales

2.2.3. Catálogo de Actores

En cuanto al actor que interactúa con el presente videojuego se puede mencionar al "Usuario del videojuego" el cual dependiendo del contexto en el cual este software sea utilizado podrá ser:

- **Un estudiante:** que desee desarrollar sus aptitudes cognitivas en mediante este juego.
- **Paciente:** que se encuentre utilizando este juego como parte de su tratamiento.
- **Jugador casual:** que simplemente tenga un objetivo recreativo.

A continuación, en la **Figura 2.5** se presenta el actor que interactuará con el videojuego.

Figura 2.5: Catálogo de actores

2.2.4. Casos de Uso

En esta sección se mencionarán los principales Casos de Uso que tiene el videojuego agrupados por paquetes. Se han definido 2 paquetes, un paquete por cada Módulo que manejará el sistema: Módulo de Juego, Módulo de Configuración.

- **Casos de Uso del Módulo de Juego**

  - CU_JUE001
    Iniciar Partida: este caso de uso permite al jugador comenzar una nueva partida sin datos.

  - CU_JUE002
    Pausar Partida: este caso de uso permite al jugador pausar la partida en cualquier momento del juego.

- CU_JUE003

  Grabar Partida: este caso de uso permite al jugador grabar una partida al finalizar un nivel.

- CU_JUE004

  Abandonar Partida: este caso de uso permite al jugador finalizar la partida en cualquier momento.

- CU_JUE005

  Cargar Partida: este caso de uso permite al jugador cargar una partida grabada con la posibilidad de escoger un nivel ya superado para volverlo a jugar o escoger el nivel inmediato superior último alcanzado.

- CU_JUE006

  Visualizar intro del juego: El videojuego permitirá visualizar un video introductorio inicial sobre la historia del juego lo que permitirá al jugador conocer el contexto de los personajes del videojuego y adentrarse en el juego desde el principio.

A continuación en la **Figura 2.6** se muestra el Diagrama de Paquetes para el módulo de Juego:

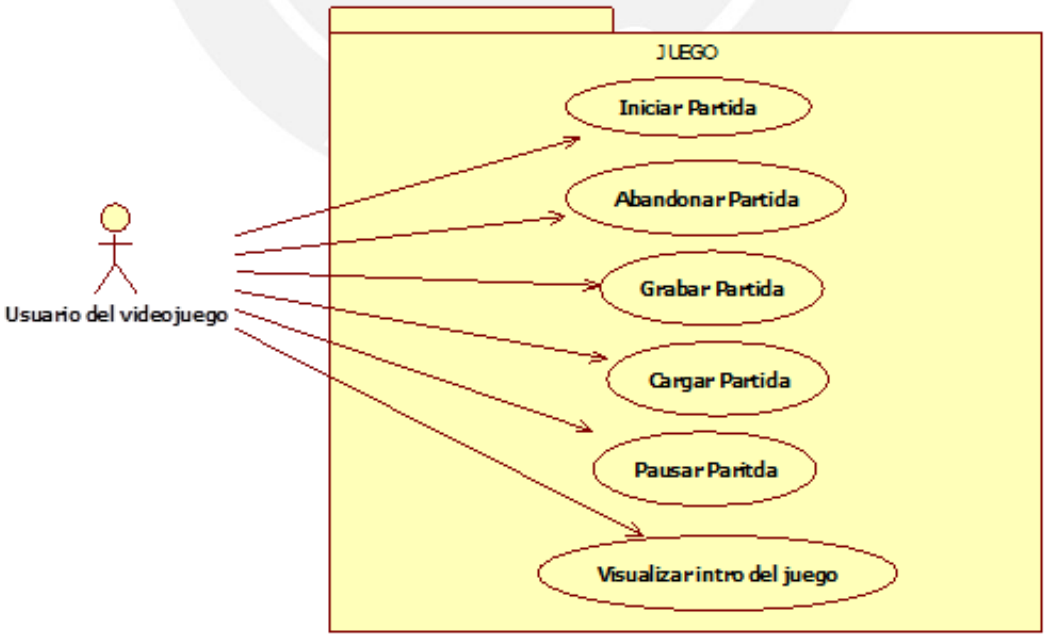


Figura 2.6: Paquete de Juego – Casos de Uso

- **Casos de Uso del Módulo de Configuración**

  - CU_CON001

    Configurar botones del juego: este caso de uso permite al jugador configurar los botones a utilizar en el juego.

  - CU_CON002

    Configurar modo de Pantalla Completa: este caso de uso permite al jugador configurar si para jugar se utilizará el modo de pantalla completa o no.

  - CU_CON003

    Ajustes de audio: este caso de uso permite al jugador activar o desactivar el audio del juego.

  - CU_CON004

    Cambiar idioma: este caso de uso permite al jugador cambiar el idioma del juego.

A continuación en la **Figura 2.7** se muestra el Diagrama de Paquetes para el módulo de Configuración:

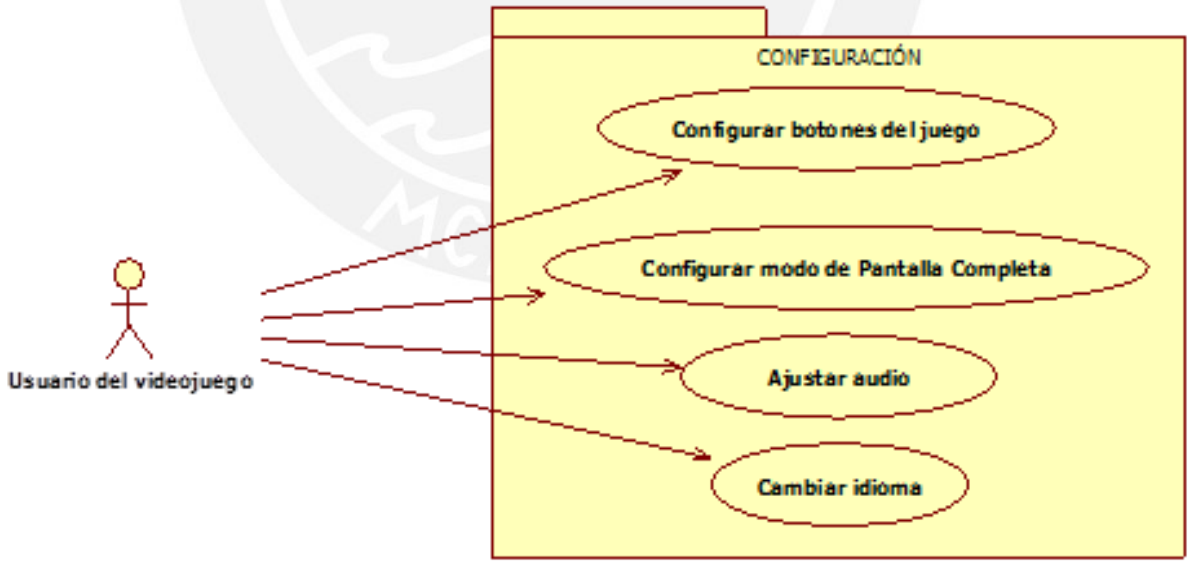


Figura 2.7: Paquete de Configuración – Casos de Uso

2.2.5. Diagrama de Clases

Para el presente proyecto se han definido seis clases: Juego, Nivel, Personaje, Obstáculo, Puntuación y Enemigo las cuales están relacionadas según lo muestra la F**igura 2.8**. A continuación se presenta un diagrama de clases preliminar generado de un análisis de la solución:

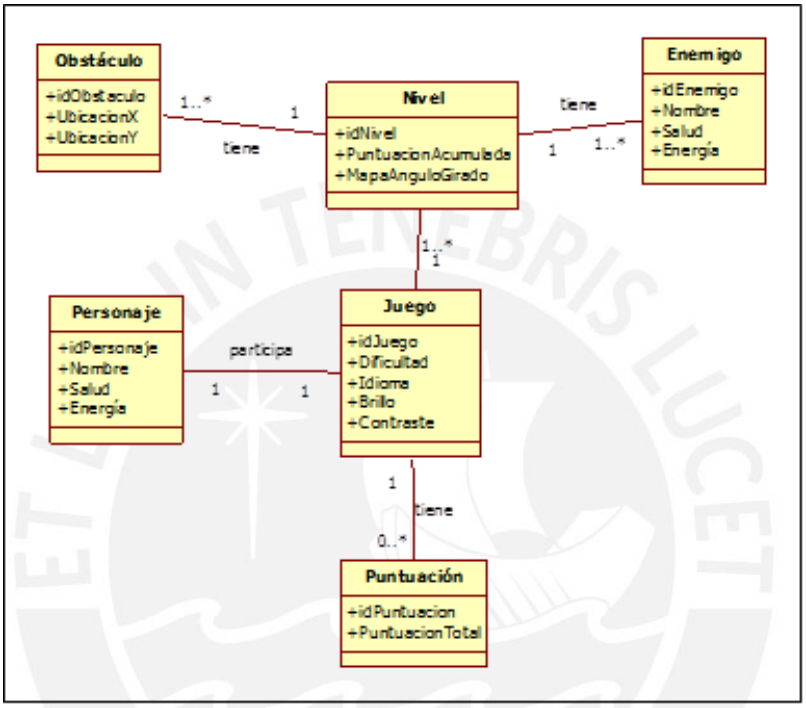


Figura 2.8: Diagrama de Clases

2.3. Análisis de viabilidad

En esta sección se presentará un análisis de la viabilidad del sistema el cual considera un análisis a nivel técnico y económico del videojuego. En cuanto al análisis técnico se tomarán en cuenta las diversas tecnologías a utilizar en el presente proyecto y se realizará una comparación entre los diversos lenguajes de programación y entornos de desarrollo integrados candidatos a usar en la presente solución.

Adicionalmente se realiza un análisis de la asignación de funciones y de las restricciones de costo y tiempo para el actual proyecto.

2.3.1. Viabilidad del sistema

El videojuego que se está planteando recoge los requerimientos necesarios para dar solución al problema originado por la baja estimulación para el desarrollo y aprendizaje en las personas.

Uno de los aspectos más importantes en la construcción del presente videojuego es el conocimiento global del negocio y público para el cual se desea implementar el juego. Este aspecto es vital pues de esto dependerá la real utilidad y ayuda que el videojuego pueda proporcionar.

Para el problema y el entorno del problema identificado en este proyecto se posee alguna experiencia del tesista en la participación de proyectos relacionados a videojuegos por lo que el tiempo que dura el entendimiento y análisis del negocio no será muy excesivo para este proyecto sino que se ahorrará tiempo de investigación acerca de los detalles del negocio de los videojuegos y se aprovechará este tiempo en realizar una investigación para una mejor elección de las herramientas para la construcción de la solución.

Según la investigación realizada sobre el entorno en el cual el problema identificado se hace presente, se obtuvo, que no existe una alta competencia en cuanto al desarrollo de videojuegos educativos en el Perú. Tal y como se describe en el estado del arte del presente documento y en la comparación realizada en la sustentación del proyecto, el videojuego presentado en el actual proyecto presenta cualidades y beneficios que no poseen otros juegos en el mercado.

Por lo expuesto se puede concluir que es necesaria una herramienta, con un enfoque recreativo que apoye el ejercicio del razonamiento lógico y el desarrollo de habilidades cognitivas referentes a la inteligencia espacial.

2.3.2. Viabilidad técnica

En lo que respecta al análisis a nivel técnico, es necesario recalcar que en la actualidad hay varias plataformas existentes que pueden soportar la tecnología necesaria para la implementación del videojuego planteado.

Para el desarrollo del videojuego, luego de un análisis técnico de las diversas tecnologías que incluyeron Python, Java, C# y Visual Basic.NET se ha optado por utilizar Python como lenguaje de programación debido a la eficiencia que presenta utilizar este lenguaje. El detalle del análisis comparativo de los diversos lenguajes de programación se presenta en el Capítulo 3.3.1 del presente documento.

Basado en el lenguaje escogido, Python, se seleccionó el entorno de desarrollo integrado (IDE) PyDev debido a su poco consumo de memoria. El detalle del análisis comparativo de los entornos de desarrollo integrado para el lenguaje Python se presenta en el Capítulo 3.3.1.

Adicionalmente para el presente proyecto se utilizará el framework Cocos2D debido a que es bastante útil para el desarrollo de videojuegos en 2D utilizando el lenguaje de programación Python.

Cabe resaltar que no se utilizará ningún motor de base de datos en el presente proyecto.

Con lo mencionado anteriormente es justo mencionar que a nivel técnico el proyecto es viable debido a que se cuentan con varios medios o tecnologías estables para la implementación.

2.3.3. Viabilidad económica

El análisis económico nos permite obtener todos los costos en los que se incurrirían en el proyecto de implementación de este videojuego. La viabilidad del proyecto dependerá mucho de este factor debido a que el presupuesto calculado no debería superar las expectativas que tiene el cliente ya que de ser el caso el proyecto sería no viable.

Cabe destacar que para el proyecto se intentará utilizar software de desarrollo libre para minimizar los costos de sistema considerablemente y evitar gastos en licencias. Tomando esto en cuenta el costo que debemos considerar es básicamente el costo producido por el tiempo de desarrollo del sistema.

A continuación en la **Tabla 2.6** se muestra la estimación de los costos:

Concepto	Días	Horas diarias	Horas totales invertidas	Costo por hora invertida	Costo Total
Levantamiento de información	13	3	39	S/. 40	S/. 1,560
Análisis	6	4	34	S/. 40	S/. 1,360
Diseño y Construcción	79	3	237	S/. 40	S/. 9,480
Otros gastos	-	-	-	-	S/. 500
				TOTAL	S/. 12,900

Tabla 2.6: Análisis Económico

2.3.4. Asignación de funciones

En esta sección se procederá a identificar los elementos que se encuentran involucrados en el desarrollo del videojuego. Además también se listan las funciones asociadas a cada uno de ellos.

Entre los elementos más importantes se pueden mencionar: Hardware, Software y Recursos como se puede observar en la **Tabla 2.7**:

Recurso	Función
Hardware	El hardware a usar debe brindar funcionamiento durante todo el momento que el usuario juegue. Por esta razón es fundamental que el hardware no presente fallas en el transcurso de su funcionamiento. Para garantizar esto se necesitará trabajar mínimo con un procesador, que tenga 128MB de RAM y 100MB de disco duro.
Software	Para el presente proyecto, el software utilizado está conformado por el lenguaje de programación Python debido a que es

Recurso	Función
	software libre lo cual minimiza considerablemente el costo debido a que no es necesario pagar por licencias.
Recursos	Los usuarios finales que manejarán el videojuego son los propios videojugadores. Adicionalmente para el presente proyecto se cuenta con un tesista a cargo de todas las etapas del proyecto: Análisis, Diseño, Implementación y Pruebas.

Tabla 2.7: Asignación de Funciones

2.3.5. Restricciones de costo y tiempo

El presente proyecto de fin de carrera no tiene ningún tipo de restricciones en cuanto a las tecnologías a usar debido a que se realiza el desarrollo utilizando herramientas libres de licencias por lo que esto beneficia a una reducción considerable de costo para el desarrollador.

En cuanto al tiempo, cabe mencionar que el actual proyecto está ajustado a un plan de trabajo diseñado de manera que se realicen diversas entregas conforme el trabajo vaya avanzando. Dicho plan de trabajo se puede observar en el Capítulo 1 del actual documento de tesis.

3. CAPÍTULO 3: DISEÑO Y CONTRUCCIÓN

En el presente capítulo se detalla la arquitectura a utilizar para el desarrollo de la solución. Además también se presentará el diseño de la interfaz gráfica de usuario, la construcción del videojuego y las pruebas a realizar (unitarias e integrales).

3.1. Arquitectura de la solución

En esta sección se plantea una visión general de la arquitectura del motor de juegos a utilizar prestando especial importancia a los módulos más relevantes desde el punto de vista del desarrollo de videojuegos.

Como sucede con la gran mayoría de sistemas software que tienen una complejidad elevada, los motores de juegos se basan en una arquitectura estructurada en capas (ver **Figura 3.1**). De este modo, las capas de nivel superior dependen de las de nivel inferior, pero no de manera inversa. Este planteamiento permite ir añadiendo capas de manera progresiva y, lo que es más importante, modificar determinados aspectos de una capa en concreto sin que el resto de capas inferiores se vean afectadas por dicho cambio.

A continuación, se describen los principales módulos que forman parte de la arquitectura:

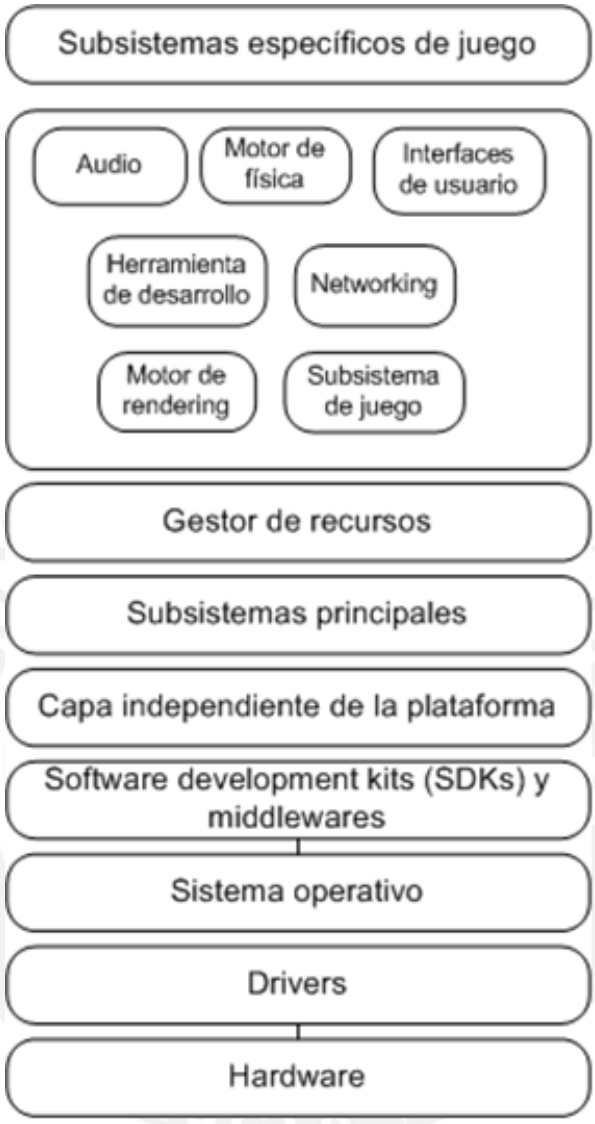


Figura 3.1: Arquitectura de videojuego basada en capas

Fuente: Elaboración propia

3.1.1. Hardware, drivers y sistema operativo

La capa referente al hardware está relacionada a la plataforma en la que se ejecutará el motor de juego. Por ejemplo, un tipo de plataforma específica podría ser una consola de juegos de sobremesa. Muchos de los principios de diseño y desarrollo son comunes a cualquier videojuego, de manera independiente a la plataforma en la que se despliegue finalmente. Sin embargo, en la práctica los programadores de videojuegos llevan a cabo

optimizaciones en el motor de juegos para mejorar la eficiencia del mismo, considerando aquellas cuestiones que son específicas de una determinada plataforma.

La capa de drivers soporta aquellos componentes software de bajo nivel que permiten la correcta gestión de determinados dispositivos, como por ejemplo las tarjetas de aceleración gráfica o las tarjetas de sonido.

La capa del sistema operativo representa la capa de comunicación entre los procesos que se ejecutan en el mismo y los recursos hardware asociados a la plataforma en cuestión. En lo referente a los videojuegos los sistemas operativos se compilan con el propio juego para producir un ejecutable. Para el presente proyecto el juego será soportado en sistemas operativos a partir de Windows XP en adelante.

3.1.2. SDKs y middlewares

Al igual que ocurre en otros proyectos de software, el desarrollo de un motor de juegos se suele apoyar en bibliotecas que ya existen y Software Development Kits (SDKs) para proporcionar una determinada funcionalidad. No obstante algunos desarrolladores prefieren personalizarlo para adaptarlo a sus necesidades particulares.

Otro ejemplo representativo de SDKs vinculados al desarrollo de videojuegos son aquellos que dan soporte a la detección y tratamiento de colisiones y a la gestión de la física de las distintas entidades que forman parte de un videojuego. Para el presente proyecto se utilizará Cocos2D como Software Development Kit debido a que es un framework bastante completo para el desarrollo de videojuegos.

3.1.3. Capa independiente de la plataforma

Gran parte de los juegos se desarrollan teniendo en cuenta su potencial lanzamiento en diversas plataformas. Por ejemplo, un título se puede desarrollar para diversas consolas de sobremesa y para PC al mismo tiempo. En este contexto, es bastante común encontrar una capa software que aísle al resto de capas superiores de cualquier aspecto que sea dependiente de la plataforma. Dicha capa se suele denominar capa independiente de la plataforma. Aunque sería bastante lógico suponer que la capa inmediatamente inferior, es decir, la capa del Cocos2D y middleware, ya posibilita la

independencia respecto a las plataformas subyacentes debido al uso de módulos estandarizados, la realidad es que existen diferencias incluso en bibliotecas estandarizadas para distintas plataformas.

3.1.4. Subsistemas principales

La capa de subsistemas principales está vinculada a todas aquellas utilidades o bibliotecas de utilidades que dan soporte al motor de juegos. Algunas de ellas son específicas del ámbito de los videojuegos pero otras son comunes a cualquier tipo de proyecto software que tenga una complejidad significativa.
A continuación se enumeran algunos de los subsistemas más relevantes que posee el framework Cocos2D:

Biblioteca matemática: responsable de proporcionar al desarrollador diversas utilidades que faciliten el tratamiento de operaciones relativas a vectores, matrices u operaciones vinculadas a líneas, rayos, esferas y otras figuras geométricas. Las bibliotecas matemáticas son esenciales en el desarrollo de un motor de juegos, ya que éstos tienen una naturaleza inherentemente matemática.

Estructuras de datos y algoritmos: responsable de proporcionar una implementación más personalizada y optimizada de diversas estructuras de datos, como por ejemplo listas enlazadas o árboles binarios, y algoritmos. Este subsistema resulta especialmente importante cuando la memoria de la plataforma o plataformas sobre las que se ejecutará el motor está limitada.
Gestión de memoria: responsable de garantizar la asignación y liberación de memoria de una manera eficiente.

3.1.5. Gestor de recursos

Esta capa del Cocos2D es la responsable de proporcionar una interfaz unificada para acceder a las distintas entidades software que conforman el motor de juegos.
En este contexto, existen dos aproximaciones principales respecto a dicho acceso: i) plantear el gestor de recursos mediante un enfoque centralizado y consistente o ii) dejar en manos del programador dicha interacción mediante el uso de archivos en disco.

Figura 3.2: Visión conceptual del gestor de recursos y sus entidades asociadas

La **Figura 3.2** muestra una visión general de un gestor de recursos, representando una interfaz común para la gestión de diversas entidades.

3.1.6. Motor de rendering

Debido a que el componente gráfico es una parte fundamental de cualquier juego, junto con la necesidad de mejorarlo continuamente, el motor de renderizado es una de las partes más complejas de cualquier motor de juego.
Al igual que ocurre con la propia arquitectura de un motor de juegos, el enfoque más utilizado para diseñar el motor de renderizado consiste en utilizar una arquitectura multi-capa, como se puede apreciar en la **Figura 3.3**.

Adicionalmente el Cocos2D ofrece un Text Rendering de etiqueta HTML con soporte de acción.

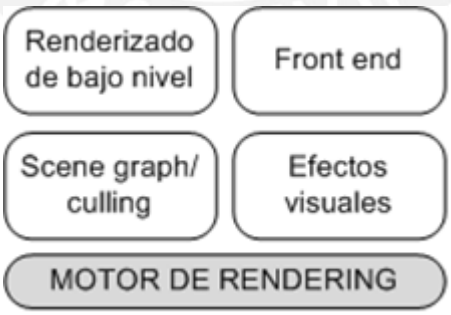


Figura 3.3: Visión conceptual de la arquitectura general de un motor de rendering

La capa de renderizado de bajo nivel aglutina las distintas utilidades de renderizado del motor, es decir, la funcionalidad asociada a la representación gráfica de las distintas entidades que participan en un determinado entorno.

El objetivo principal de esta capa reside precisamente en renderizar las distintas primitivas geométricas tan rápido como sea posible.
Esta capa también es responsable de gestionar la interacción con las APIs de programación gráficas, simplemente para poder acceder a los distintos dispositivos gráficos que estén disponibles.

La capa superior a la de renderizado de bajo nivel se denomina scene graph/culling y optimizaciones y, desde un punto de vista general, es la responsable de seleccionar qué parte o partes de la escena se enviarán a la capa de rendering. Esta selección, u optimización, permite incrementar el rendimiento del motor de rendering, debido a que se limita el número de primitivas geométricas enviadas a la capa de nivel inferior.

Sobre la capa relativa a las optimizaciones se sitúa la capa de efectos visuales, la cual proporciona soporte a distintos efectos que, posteriormente, se puedan integrar en los juegos desarrollados haciendo uso del motor.

Finalmente, la capa de front-end. Es bastante común utilizar algún tipo de módulo que permita visualizar el menú de un juego o la interfaz gráfica que permite conocer el estado del personaje principal del videojuego (inventario, armas, herramientas, etc.). En esta capa también se incluyen componentes para reproducir vídeos previamente grabados y para integrar secuencias cinemáticas, a veces interactivas, en el propio videojuego.

3.1.7. Herramienta de depuración

Debido a la naturaleza intrínseca de un videojuego, vinculada a las aplicaciones gráficas en tiempo real, resulta esencial contar con buenas herramientas que permitan depurar y optimizar el propio motor de juegos para obtener el mejor rendimiento posible. En este contexto, existe un gran número de herramientas de este tipo. Algunas de ellas son herramientas de propósito general que se pueden utilizar de manera externa al motor de juegos. Sin embargo, la práctica más habitual consiste en construir herramientas de profiling, vinculadas al análisis del rendimiento, o depuración que estén asociadas al propio motor.
Adicionalmente el framework Cocos2D ofrece un intérprete de Python incorporado para propósitos de depuración.

3.1.8. Motor de física

La detección de colisiones en un videojuego y su posterior tratamiento resultan esenciales para dotar de realismo al mismo. Sin un mecanismo de detección de colisiones, los objetos se traspasarían unos a otros y no sería posible interactuar con ellos. Desde un punto de vista general, el sistema de detección de colisiones es responsable de llevar a cabo las siguientes tareas:

- La detección de colisiones, cuya salida es un valor lógico indicando si hay o no colisión.
- La determinación de la colisión, cuya tarea consiste en calcular el punto de intersección de la colisión.
- La respuesta a la colisión, que tiene como objetivo determinar las acciones que se generarán como consecuencia de la misma.

Entre los beneficios que ofrece el framework Cocos2D se cuenta con un soporte puro basado en Python para colisiones, transiciones, efectos y diversas acciones.

3.1.9. Interfaces de usuario

En cualquier tipo de juego es necesario desarrollar un módulo que ofrezca una abstracción respecto a la interacción del usuario, es decir, un módulo que principalmente sea responsable de procesar los eventos de entrada del usuario. Típicamente, dichos eventos estarán asociados a la pulsación de una tecla, al movimiento del mouse, entre otros.

Desde un punto de vista más general, el módulo de interfaces de usuario también es responsable del tratamiento de los eventos de salida, es decir, aquellos eventos que proporcionan una retroalimentación al usuario. El módulo de interfaces de usuario actúa como un puente entre los detalles de bajo nivel del hardware utilizado para interactuar con el juego y el resto de controles de más alto nivel. Para el presente proyecto, el framework Cocos2D proporcionará funcionalidades para el adecuado manejo de eventos.

3.1.10. Subsistema de juego

El subsistema de juego, conocido por su término en inglés gameplay, integra todos aquellos módulos relativos al funcionamiento interno del juego, es decir, aglutina tanto las propiedades del mundo como la de los distintos personajes. Por una parte, este subsistema permite la definición de las reglas que gobiernan el mundo en el que se desarrolla el juego, como por ejemplo la necesidad de derrotar a un enemigo antes de enfrentarse a otro de mayor nivel. Por otra parte, este subsistema también permite la definición de la mecánica del personaje, así como sus objetivos durante el juego. (Ver **Figura 3.4**)

Referente a este punto, el framework Cocos2D ofrece varios beneficios como sprites, acciones de los personajes, propiedades de los mapas y personajes, transiciones entre mapas, efectos diversos, etc.

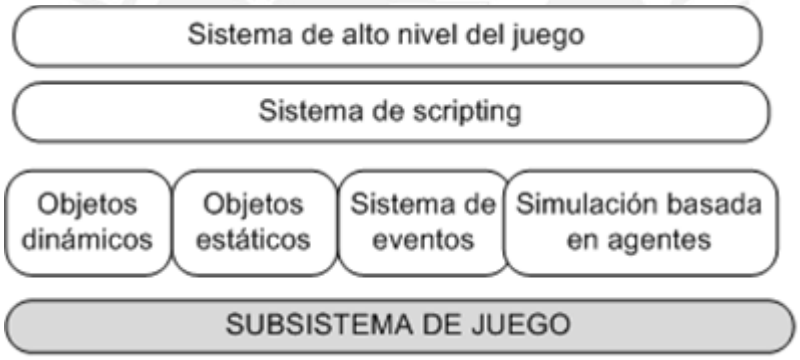


Figura 3.4: Visión conceptual de la arquitectura general del subsistema de juego

3.1.11. Audio

Tradicionalmente, el mundo del desarrollo de videojuegos siempre ha prestado más atención al componente gráfico. Sin embargo, el apartado sonoro también tiene una gran importancia para conseguir una inmersión total del usuario en el juego. Por ello, el motor de audio ha ido cobrando más relevancia.

Actualmente, al igual que ocurre con otros componentes de la arquitectura del motor de juego, es bastante común encontrar desarrollos listos para utilizarse e integrarse en el motor de juego, los cuales han sido realizados por compañías externas a la del propio motor. No obstante, el apartado sonoro también requiere modificaciones que son

específicas para el juego en cuestión, con el objetivo de obtener un alto de grado de fidelidad y garantizar una buena experiencia desde el punto de visto auditivo. El framework Cocos2D incluye un motor sonoro incorporado denominado SimpleAudioEngine con lo cual es posible incluir sonidos y música tanto solo añadiendo una línea de código. Los formatos de audio son soportados en diversas plataformas.

3.1.12. Subsistema específico de juego

Por encima de la capa de subsistema de juego y otros componentes de más bajo nivel se sitúa la capa de subsistemas específicos de juego, en la que se integran aquellos módulos responsables de ofrecer las características propias del juego. En función del tipo de juego a desarrollar, en esta capa se situarán un mayor o menor número de módulos, como por ejemplo los relativos al sistema de cámaras virtuales, mecanismos de Inteligencia Artificial específicos de los personajes no controlados por el usuario, aspectos de renderizados específicos del juego, sistemas de armas, puzzles, etc. Idealmente, la línea que separa el motor de juego y el propio juego en cuestión estaría entre la capa de subsistema de juego y la capa de subsistemas específicos de juego.

3.1.13. Jugabilidad

Un aspecto fundamental para el diseño e implementación de un juego es el uso de la Heurística como parte del proceso de creación.
La Heurística se ha convertido en un método aceptado y ampliamente utilizada para la evaluación de la usabilidad en el desarrollo de un software. La Heurística podría verse como un conjunto de directrices de diseño que sirven como una útil herramienta de evaluación para los diseñadores profesionales de videojuegos.

Las cuatro categorías heurísticas para un juego son:

- *"Gameplay":* es el conjunto de problemas y desafíos que un usuario debe enfrentar para ganar un juego.
- *"Historia del juego":* incluye todo el terreno y el desarrollo del jugador.
- *"Mecánica del juego":* involucra la programación que provee la estructura por la cual las unidades interactúan con el ambiente.

- *"Usabilidad del juego":* abarca los elementos que el usuario utiliza para interactuar con el juego (mouse, teclado, control, consola de juego, etc.).

En el desarrollo de un videojuego es necesario desarrollar un conjunto de heurísticas que permitan evaluar según cada categoría definida cada aspecto del juego mediante una métrica.
Para realizar esta medición se cuenta con:

- *La Evaluación Heurística para la Jugabilidad* (HEP – Heuristic Evaluation for Playability): está elaborada en base a la opinión de expertos en el diseño de videojuegos. El evaluador de la jugabilidad ejecuta el HEP mientras se enfoca en cómo cada heurística está cumplida o fallida y entonces define las cuestiones de la jugabilidad las cuales deben ser resultas por soluciones alternativas del diseñador del juego.
- *La Evaluación por parte del Usuario*: viene luego de la evaluación heurística realizada por el diseñador (HEP). Los usuarios son utilizados para probar el juego y brindar sus opiniones de mejora mediante un cuestionario de satisfacción. El evaluador procede a grabar un registro de las acciones del jugador, comentarios, fallas, errores, y todo aquello que haya sido considerado como parte de una buena o mala experiencia del jugador. Una experiencia positiva es considerada cuando el jugador experimenta una sensación de placer, inmersión y de desafío en el juego. En cambio una experiencia negativa es definida como la situación en la que el jugador se encuentra aburrido, frustrado y desea dejar el juego.

Luego de que ambas evaluación fueron realizadas se procede con la identificación y solución de los problemas en cuanto al diseño de la jugabilidad.
Según un estudio realizado [HEU001] las heurísticas HEP proveen mayor efectividad que incluso la Evaluación del Usuario, en todas las categorías heurísticas especialmente en los rubros de "Historia del Juego" y "Usabilidad del Juego". Esto demuestra que para el diseño inicial de un juego, la evaluación HEP es sumamente efectiva, no siéndolo tanto en los rubros de "Gameplay" y "Mecánica del juego" en donde prima la evaluación del usuario.
Por lo tanto la evaluación del usuario vendría a ser el punto de referencia para cualquier evaluación de la jugabilidad ya que un diseñador no puede predecir completamente el comportamiento que tendrá el usuario. HEP es muy útil para crear diseños acerca de la jugabilidad de los videojuegos particularmente en la fase preliminar o inicial y el estudio del usuario se ajusta mejor para encontrar problemas específicos que ya existen.

A continuación se indican las heurísticas definidas a tener en cuenta para el diseño del videojuego de la presente tesis. Como se describió anteriormente, se incluyen las 4 categorías heurísticas:

Gameplay

Ref.	Descripción de la heurística	Nivel esperado en el juego (1-5)
1	La fatiga del jugador es minimizada variando los niveles y las actividades a realizar durante el juego.	3
2	Proveer objetivos claros, y manifestarlos en el momento preciso durante el juego.	4
3	Hay un interesante y absorbente tutorial explicativo del juego.	3
4	El juego es muy entretenido incluso al repetirlo.	4
5	El gameplay debería ser balanceado con múltiples maneras para ganar.	3
6	El jugador aprende habilidades de manera temprana que esperarías obtener más adelante.	4
7	El jugador descubre la historia como parte del juego.	3
8	El juego es divertido en primer lugar para el jugador, en segundo lugar para el diseñador y finalmente para la computadora. Es decir, si la experiencia del jugador no experto no se pone primero, no tendría sentido una excelente mecánica de juego y unos gráficos de última generación.	5
9	El juego debería dar premios que permitan al jugador involucrarse más en el juego.	4
10	El ritmo del juego debería aplicar presión sobre el jugador pero no debería frustrarlo. Variar el nivel de dificultad para que el jugador tenga un mayor desafío a medida que avanza.	3
11	Los desafíos son experiencias positivas de juego lo cual llevará a que el usuario desee jugar más.	4

Historia del juego

Ref.	Descripción de la heurística	Nivel esperado en el juego (1-5)
1	El jugador entiende la historia del juego como una visión consistente.	3
2	La historia del juego resulta interesante para el jugador.	4
3	El jugador invierte tiempo pensando acerca de posibles historias que vengan.	3
4	El jugador tiene la sensación de control sobre el personaje del juego y es capaz de utilizar tácticas y estrategias.	5
5	El juego transporta al jugador en un nivel emocional envolvente (miedo, venganza, confort).	3

Mecánica del juego

Ref.	Descripción de la heurística	Nivel esperado en el juego (1-5)
1	El jugador siempre es capaz de visualizar el puntaje y objetivos conseguidos en cada nivel.	5
2	La curva de aprendizaje se encuentra dentro de las expectativas del jugador.	5
3	Los controles son intuitivos y son mapeados de una forma natural.	5
4	El jugador debería tomar los controles como de básico aprendizaje para ser aprendidos de manera rápida.	5

Usabilidad del juego

Ref.	Descripción de la heurística	Nivel esperado en el juego (1-5)
1	El jugador puede prender y apagar el juego de manera sencilla.	4
2	El jugador experimenta la interface de usuario como consistente (en color, diálogo, diseño).	5
3	El jugador debería experimentar el menú de juego tanto	4

	como el juego.	
4	Ni bien iniciado el juego el jugador debería tener suficiente información para poder iniciarse en la historia.	4
5	Los sonidos en el juego proveen de emociones al jugador.	3
6	El jugador no necesita de un manual para empezar el juego.	4
7	Los menús del juego son lo suficientemente intuitivos.	5

En conclusión, la evaluación heurística es una ingeniería de la usabilidad, método para la evaluación de interfaces de usuario para encontrar problemas de usabilidad. Básicamente, un conjunto de evaluadores inspeccionan la interfaz con respecto a un pequeño conjunto de principios de usabilidad bastante amplios que se conocen como "heurísticas" [HEU002].

3.2. Diseño de Interfaz gráfica

Para el diseño de la interfaz gráfica se ha buscado uniformizar todas las ventanas del videojuego para hacerlo más amigable y fácil de manejar. Se ha tomado de vital importancia el diseño gráfico del videojuego debido a que es considerado un factor importante para atraer jugadores lo cual repercute en el éxito del presente proyecto.

3.2.1. Criterios para el diseño de interfaz

Para diseñar el videojuego se han tomado las siguientes consideraciones:

- Para cada ventana se deberá identificar el elemento principal o esencial que represente a la misma para proceder a ubicarlo en una posición estratégica en la que resalte su importancia. Por ejemplo: la barra de salud de nuestro personaje tendrá una ubicación estratégica que permita al jugador tener una fácil visualización de ésta lo cual le permitirá poder tomar decisiones rápidas dependiendo de la cantidad de vitalidad que posea el personaje.
- Se utilizarán diversas imágenes que reflejen el contenido de cada ventana ya sea en los fondos como en los botones de acción. Para cada mapa o nivel de juego se presentarán imágenes representativas de ese nivel lo cual permitirá al jugador sentirse inmerso en el juego en todo momento.
- Los colores a utilizar serán escogidos de tal forma que no cansen la vista de los jugadores.

- Los nombres utilizados en las ventanas serán escogidos de manera estratégica de tal forma que el usuario sepa identificarlos inmediatamente para facilitar y agilizar la navegación por el videojuego.
- En todo momento se dispondrá de un menú de pausa que permita al jugador detener el juego momentáneamente y escoger las diversas opciones que ofrece la pausa del videojuego (reanudar juego, opciones del juego, salir del juego).

3.2.2. Tipos de pantallas

En esta sección se explicarán las distintas ventanas que se presentarán en el videojuego y cómo ha sido diseñada cada una de ellas. A continuación se expone el diseño preliminar de los prototipos del videojuego:

- **Ventana del Menú Principal**

Una vez que haya ingresado el jugador al videojuego, se mostrará la pantalla de bienvenida o Menú Principal el cual posee las distintas opciones que el jugador puede realizar en un primer instante. Cabe resaltar que el Menú Principal posee un diseño que permitirá al jugador reconocer el videojuego al instante debido a su agradable contenido visual así como la disposición de las diversas opciones intuitivas y nada complicadas de aprender a manejar.

Desde el Menú Principal se puede acceder a las diferentes opciones del juego: (ver **Figura 3.5**) tales como:

- Empezar
- Niveles
- Opciones
- Créditos
- Salir

La imagen de la ventana se muestra a continuación:

Figura 3.5: Ventana del Menú Principal

- **Ventana de Inicio de Partida**

En la ventana de Inicio de Partida se procederá a iniciar el juego con una breve historia introductoria que tendrá inmerso al jugador desde un primer instante en los objetivos a cumplir dentro de cada nivel. Se contará con un breve tutorial que explicará cómo se debe jugar y cuáles son los objetivos del juego para poder aprovechar de la mejor manera los beneficios tanto cognitivos como recreacionales que posee este proyecto en las personas.

Adicionalmente una vez empezado el juego, es posible pausarlo en el momento que se desee tras lo cual aparecerá el menú de pausado del juego con las opciones de Continuar y Salir. (Ver **Figura 3.6**).

Figura 3.6: Ventana de Pausa del Juego

- **Ventana de Carga de Partida**

Mediante la ventana de Cargar Partida se permitirá al jugador empezar el juego desde una partida ya guardada con anterioridad permitiendo al jugador la versatilidad de poder jugar cuando desee y cómo lo desee. Sin embargo cabe mencionar que la partida podrá ser guardada únicamente al finalizar un nivel por lo que la carga de partida siempre iniciará al jugador en un nivel en específico mas no en el transcurso de un mapa. También se incluirá la opción de jugar un nivel ya superado anteriormente.

En la **Figura 3.7** se muestra la ventana de Carga de partida:

Figura 3.7: Ventana de Cargar Partida

- **Ventana de Configuración**

En la ventana de Configuración se muestran las diversas opciones que posee el juego tales como: (Ver **Figura 3.8**)

- Configuración de botones: mediante esta opción se permitirá al jugador poder escoger los botones a utilizar en el videojuego para realizar las distintas acciones que tendrá nuestro personaje.
- Configuración de pantalla: mediante esta opción se permitirá al jugador especificar si se desea jugar en modo de Pantalla Completa o si desea visualizar el juego en una ventana más pequeña que la resolución que posee su equipo.

- Cambiar el idioma del juego: mediante esta opción se permitirá al jugador escoger el idioma que poseerá el juego en sus distintos niveles. En primera instancia se tiene planificado incluir los idiomas español e inglés.
- Ajuste de sonido: mediante esta opción se permitirá al jugador realizar los ajustes del sonido del juego permitiéndole activarlo o desactivarlo.

Figura 3.8: Ventana de Configuración

3.3. Construcción

En la presente sección se detallarán las tecnologías a utilizar para la construcción de la solución propuesta, haciendo una comparación de las ventajas y beneficios ofrecidos por cada una de ellas. También se presentan los estándares de programación a utilizar para el desarrollo del proyecto.

3.3.1. Tecnologías

A continuación se listan los lenguajes de programación que se analizaron como posibles para utilizar en el proyecto así como los entornos de desarrollo integrado candidatos para la presente solución y finalmente se presenta el framework a utilizar para la implementación del videojuego.

- **Lenguajes de programación**

En cuanto a los lenguajes de programación se tuvo que escoger de la lista conformada por: Visual Basic .NET, Python, C# y Java.

- **Visual Basic .NET**

A continuación se procede a listar las principales características del lenguaje VB.NET:

- Lenguaje orientado a objetos
- Potencia del compilador
- Posibilidad de implementar módulos en otros lenguajes (por ejemplo: C#), es decir, desarrollar en Visual Basic .NET y a futuro integrar un módulo en C#.
- Posee una curva de aprendizaje muy rápida
- Por tratarse de una herramienta conocida, existen muchos componentes y extensiones desarrollados para este lenguaje.
- Facilidad para la depuración del código
- Desarrollado por Microsoft que cuenta con un prestigio y liderazgo ganado en el mercado.
- Es uno de los lenguajes de uso más extendido, por lo que resulta fácil encontrar información, documentación y fuentes para los proyectos.

- **Python**

Python es un lenguaje muy expresivo, es decir, los programas son muy compactos, un programa en Python suele ser bastante más corto que su equivalente en lenguajes como C, por muchos Python es considerado un lenguaje de programación de muy alto nivel.

- Python es muy legible, la sintaxis de Python es muy elegante y permite la escritura de programas cuya lectura resulta fácil, en comparación con otros lenguajes.
- Python puede usarse como lenguaje imperativo procedimental o como lenguaje orientado a objetos.
- Python es un muy buen lenguaje para empezar a programar.
- Una ventaja fundamental de Python es la gratuidad de su intérprete.
- Es un lenguaje multiplataforma ya que funciona en Windows, Linux/Unix, Mac OS X.

- **C Sharp**

A continuación se procede a listar las principales características del lenguaje C#:

- Lenguaje orientado a objetos
- Compilación a código intermedio (CIL).
- En C# existe un rango más amplio de tipos de datos de los que hay en Java y otros.
- Desarrollado por Microsoft que cuenta con un prestigio y liderazgo ganado en el mercado.
- Recolección de basura automática.
- Eliminación del uso de punteros.
- No hay archivos de cabecera (.h).
- Definición de propiedades e inicializadores en las clases.
- Permite un mejor control de versiones, es decir, múltiples versiones de clases en forma binaria utilizando diferentes espacios de nombres.

- **Java**

Java es un lenguaje de programación orientado a objetos que posee una característica bastante importante tanto para el desarrollador del software como para el cliente que adquiera el mismo, es Open Source. El hecho de que sea un lenguaje libre lo hace una herramienta a tomar en cuenta bastante aceptable ya que la libertad de licencias es muchas veces un factor importante a tener en cuenta cuando un cliente adquiere un software.

A continuación se listan las ventajas del lenguaje Java:

- Robustez del lenguaje
- Herramienta libre de licencias
- Lenguaje orientado a objetos
- Fácil acceso a base de datos con el JDBC
- Es independiente de la plataforma de desarrollo (multiplataforma)
- Utiliza el Garbage Collector
- Existen en su librería clases gráficas como awt y swing

A continuación en la **Tabla 3.1** se presenta una comparación de los lenguajes de programación candidatos para ser utilizados en el presente proyecto:

	Python	Java	Visual Basic .NET	C#
Open Source	Sí	Sí	No	No
Multiplataforma	Sí	Sí	No	No
Experiencia con el lenguaje	Intermedio	Intermedio	Intermedio	Intermedio
Posee documentación	Sí	Sí, bastante	Regular	Regular
Eficiencia	Muy buena	Buena	Regular	Regular

Tabla 3.1: Tabla comparativa Lenguajes de Programación

Finalmente se ha decidido utilizar el lenguaje de programación Python debido a que es libre, multiplataforma y muy eficiente.

- **Entorno de desarrollo integrado**

En cuanto al entorno de desarrollo integrado se manejaron las siguientes opciones: PyDev, OpenKomodo, entre otros menos populares como PIDA, SPE, WingIDE y Eric.

- **PyDev**

Se trata de un plugin para Eclipse que tiene todo lo que pudiéramos necesitar de un IDE:

- Resaltado de código (ver **Figura 3.9**)
- Sangrado automática
- Completado automático
- Ejecución de programas
- Depurador de soluciones
- Administrador de proyectos

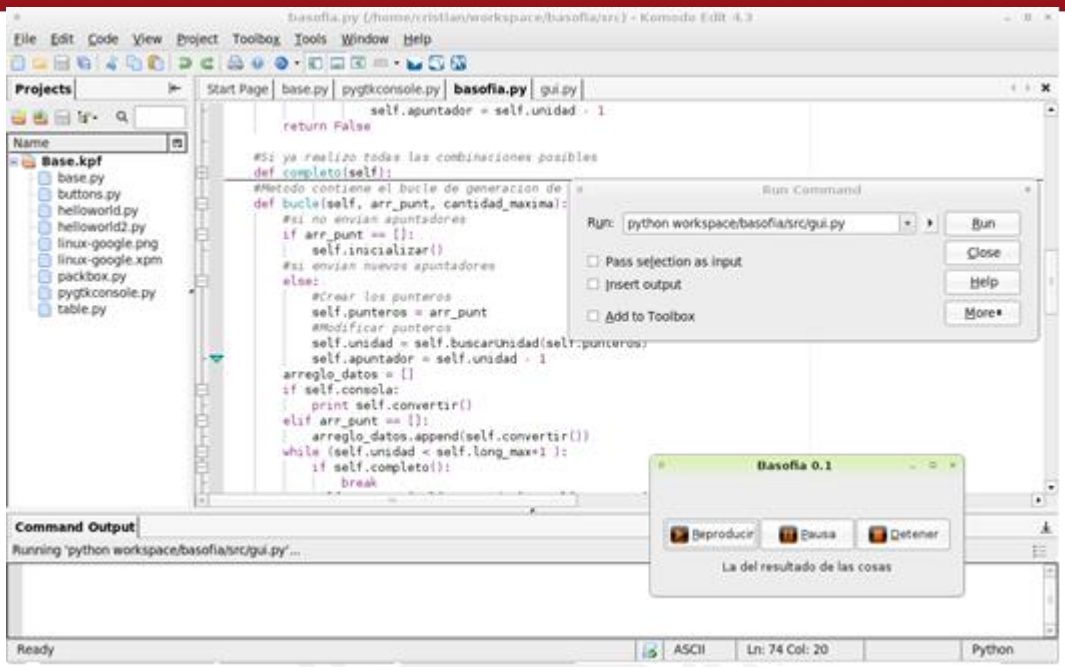


Figura 3.9: IDE PyDev

- **OpenKomodo**

Es considerado un IDE muy bueno y nace como resultado de la liberación de parte del código del editor de Komodo, pero es solo un editor, por lo que el proceso de ejecución y depuración de aplicaciones se debe realizar manualmente o usando otras herramientas. Aún así posee:

- Resaltado de código (ver **Figura 3.10**)
- Sangrado automática
- Completado automático
- Administrador de proyectos
- Y utilidades para el código (comentar bloques de código, uso de bookmarks,etc.)

En la **Tabla 3.2** se presenta una comparación de los 2 entornos de desarrollo para Python en donde se puede observar las características que presenta cada uno.

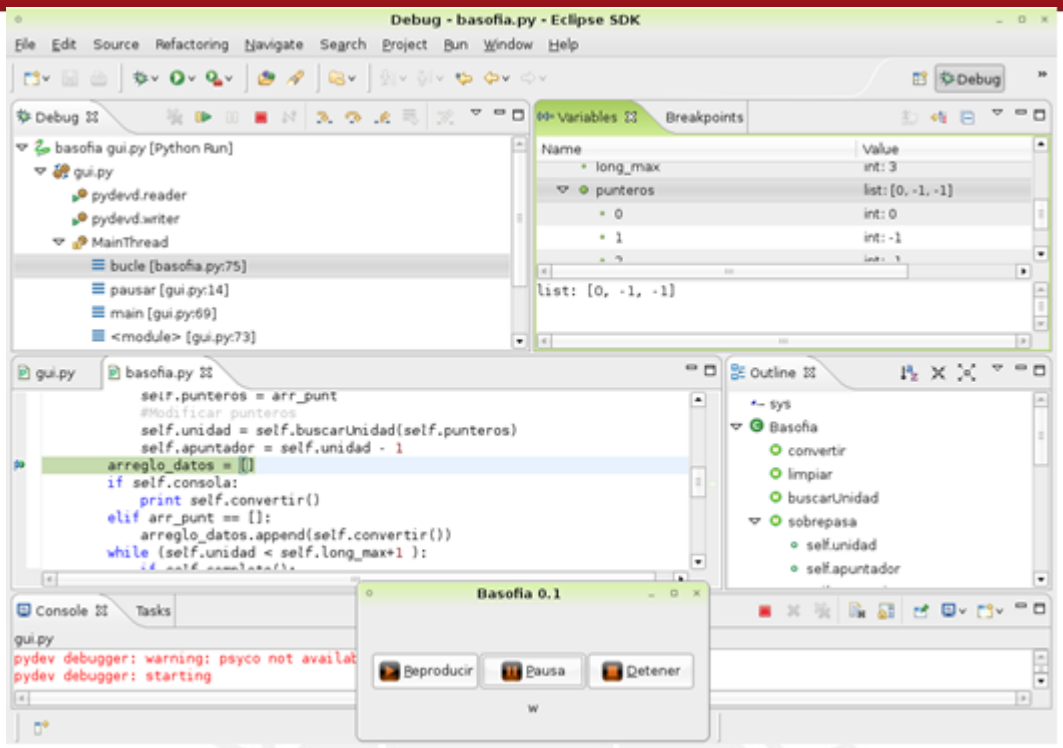


Figura 3.10: IDE OpenKomodo

	PyDev	OpenKomodo
Instalación	Regular	Regular
Experiencia con el IDE	Básico	Básico
Consumo de memoria	Regular	Regular
Apariencia del IDE	Buena	Buena

Tabla 3.2: Tabla comparativa Entorno de Desarrollo Integrado

- **Framework**

  - Cocos2D

Cabe resaltar que se utilizará para el presente proyecto el framework “Cocos2D”.

Este framework es utilizado para construir juegos en 2D, demos y otras aplicaciones gráficas/interactivas para desarrollos en el lenguaje Python.

En el presente proyecto se utilizará la versión 0.5.5 de Cocos2D la cual fue lanzada el 12 de agosto del 2012.

Entre las características que presenta este framework se puede mencionar:

- Control de flujo: Administrar el control de flujo entre diferentes escenas de una manera fácil.
- Sprites: Rápido y fácil manejo de sprites.
- Acciones: incluidas acciones como mover, rotar, escalar y mucho más.
- Efectos: Los efectos como ondas, giro, lentes y mucho más.
- Colisión: Soporte básico puro Python para las colisiones.
- Transiciones: para pasar de una escena a otra con estilo.
- Documentación: Guía de programación Referencia de la API. Tutoriales en vídeo y muchas pruebas sencillas que muestran cómo usarlo.
- Intérprete build-in Python: Para fines de depuración.
- Licencia BSD incluida.
- OpenGL: para la aceleración de hardware.

3.3.2. Estándares de programación

El presente videojuego se desarrollará utilizando el lenguaje de programación Python. A continuación se procede a indicar los estándares de programación a utilizar en el presente proyecto.

- **Tipos de dato**

Representaremos los tipos de datos con una abreviatura de 3 letras. A continuación se detallan los nemónicos (**Tabla 3.3**):

Tipo de dato	Nemónico
Boolean	Bln
Byte	Byt
Char	Chr
Float	Flt
Double	dbl
Integer	Int
Long	Lng
String	Str

Tipo de dato	Nemónico
List	Lst
Date	dat

Tabla 3.3: Estándares de Programación – Tipo de Dato

- **Definición de clases**

Para el presente proyecto se utilizarán los siguientes tipos de clases (ver **Tabla 3.4**):

- Clases del negocio: Estas clases contendrán los atributos de las entidades, además del constructor, etc. Estas son las clases que representarán las entidades del negocio.
- Clases DAO (Data Access Object): estas clases contendrán los métodos de implementación de las clases del negocio.
- Clases gestoras: Son las clases que comunican las clases del negocio con las clases DAO. Es en estas clases donde se desarrolla la lógica del negocio.
- Clases de vista: Son las clases que sirven para desplegar la interfaz gráfica del sistema.

Clases	
Declaración	Como longitud máxima del nombre de la clase se tomará 20 caracteres. El prefijo se deberá poner en mayúscula seguido del nombre de la clase con la primera letra en mayúscula. Si el nombre de la clase tiene más de una palabra, se pondrá en mayúscula la letra de inicio de cada palabra.
Ejemplos	- Personaje: Clase del negocio Personaje. - GestorPersonaje: Clase que maneja la colección de Personajes con sus respectivos métodos.

Tabla 3.4: Estándares de Programación – Definición de Clases

- **Definición de objetos**

A continuación se detallan los estándares para las instancias de las clases, es decir, los objetos (**Tabla 3.5**).

Objetos	
Formato general	[tipo de clase] + NombreObjeto
Declaración	Como longitud máxima se tomará 20 caracteres. El tipo de clase se deberá poner en minúsculas seguido del nombre del objeto con la primera letra en mayúscula. Si el nombre del objeto tiene más de una palabra, se pondrá en mayúscula la letra de inicio de cada palabra.
Ejemplos	- objPersonaje: Objeto de la clase del negocio Personaje. - gestorPersonaje: Objeto de la clase Personaje que maneja la colección de eventos con sus respectivos métodos.

Tabla 3.5: Estándares de Programación – Definición de Objetos

- **Definición de atributos**

Ningún atributo va a poder accederse sin utilizar los métodos Get y Set. Todos los atributos definidos en una clase serán privados. El nombre del atributo estará en singular.

- **Definición de métodos**

Los métodos siguen los siguientes estándares detallados en la **Tabla 3.6.**

Métodos	
Formato general	NombreMétodo
Declaración	Si el nombre de un método tiene más de una palabra, se deberá escribir con mayúscula la primera letra de cada una. No debe usarse tilde ni la letra ñ en los nombres de los métodos.
Ejemplos	- DeshabilitarSonido (); - CambiarIdioma ();

Tabla 3.6: Estándares de Programación – Definición de Métodos

- **Definición de controles**

En cuando a los estándares definidos para los controles que se utilizarán en la interfaz gráfica tenemos (**Tabla 3.7**):

Prefijos de controles		
Tipo de control	**Nombre control**	**Prefijo**
Ventana (Form)	Label	Lbl
	Button	Btn
	RadioButton	Rbtn
	CheckBox	Cbx
	ComboBox	Cmb
	ListBox	Lstb
	TextBox	Txt
Controles		
Formato general	[prefijo de control] + Nombre	
Declaración	Las letras del prefijo deberán estar en minúsculas. La primera letra del nombre deberá estar en mayúscula. Se aplican las mismas reglas que para la declaración de variables.	
Ejemplos	- TxtPuntajeObtenido - BtnIniciarPartida	

Tabla 3.7: Estándares de Programación – Definición de Controles

3.4. Pruebas

En todo proyecto de implementación de un videojuego se necesita un plan de pruebas que permita asegurar un funcionamiento correcto del mismo. Todo videojuego que sea de calidad debe pasar por diversos casos de prueba para comprobar su solidez.

Para el presente proyecto se utilizarán pruebas de caja negra pues éstas permiten centrarse en los requisitos funcionales del videojuego pues el fin es probar que las especificaciones del juego realmente cumplan su objetivo sin entrar en detalle de cómo lo hizo.

Para realizar el Plan de Pruebas se seguirán dos pasos fundamentales:

- Identificar las clases de equivalencia, tanto “válidas” como “no válidas”.
- Definir los casos de prueba. A cada clase de equivalencia se le definirá un número.

En la **Figura 3.11** se muestra como se realizarán las pruebas y registro de bugs:

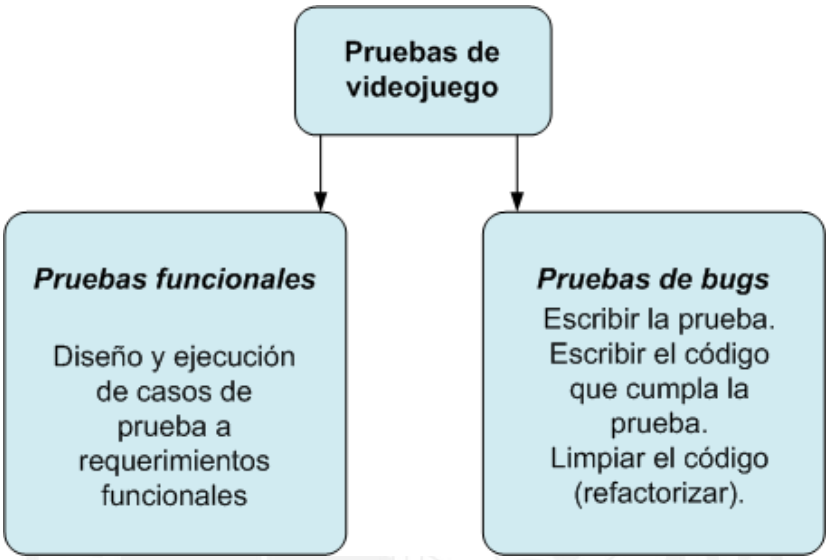


Figura 3.11: Pruebas del videojuego

3.4.1. Clases de equivalencia

Las clases de equivalencia los casos de uso más importantes del videojuego se encuentran en el “Anexo F”. (Ver **Anexo F**).

Se procede a presentar a manera de ejemplo una funcionalidad del videojuego: Cargar Partida.

- **Cargar Partida**

Se procede a definir las clases de equivalencia para el caso de uso “Cargar Partida”.

Campo	Clases Válidas	Clases No Válidas
Nivel	Opción de la lista mostrada por el videojuego: [Nivel 1 , Nivel 2 , Nivel 3]	Vacío

- **Configurar modo de Pantalla Completa**

Se procede a definir las clases de equivalencia para el caso de uso “Configurar modo de Pantalla Completa” (Ver **Tabla 3.8**).

Campo	Clases Válidas	Clases No Válidas
Modo de pantalla	Opción de la lista mostrada por el videojuego: [Normal, Completa]	Vacío

- **Caso de Uso: Opción Salto del personaje**

Se procede a definir las clases de equivalencia para el caso de uso “Opción Salto”.

Campo	Clases Válidas	Clases No Válidas
Salto	Up	Vacío o cualquier otra tecla.

Tabla 3.8: Ejemplo clases de equivalencia

3.4.2. Catálogo de Pruebas

El detalle del Catálogo de Pruebas se encuentra como anexo adjuntado con la presente tesis. (Ver **Anexo G**)

A continuación se muestra a manera de ejemplo el resultado de los casos de pruebas para 3 funcionalidades: “Cargar Partida”, “Configurar modo de Pantalla Completa” y “Opción Salto” (Ver **Tabla 3.9**).

Código	PU-01
Denominación	Cargar Partida
Propósito	Verificar un flujo de “Cargar Partida”

Código	PU-01
Precondición	El jugador ha ingresado al videojuego. El jugador ha ingresado a la opción “Seleccionar Nivel”.
Datos de entrada	En la pantalla se ingresaron los siguientes datos: **Campo** / **Valor** **Nivel** / 2
Resultado esperado	El videojuego luego de validar la opción ingresada permite el ingreso al nivel 2 del juego.

Código	PU-02
Denominación	Configurar modo de Pantalla Completa
Propósito	Verificar un flujo de “Configurar modo de Pantalla Completa”
Precondición	El jugador ha ingresado al videojuego. El jugador ha ingresado a la opción “Configuración”. El jugador ha ingresado a la sub-opción “Modo Pantalla”.
Datos de entrada	En la pantalla se ingresaron los siguientes datos: **Campo** / **Valor** **Modo Pantalla** / Completa
Resultado esperado	El videojuego luego de validar la opción ingresada permite el cambio del modo de pantalla.

Código	PU-06
Denominación	Opción Salto
Propósito	Verificar que el personaje reacciona a los controles.
Precondición	El jugador ha ingresado al videojuego. El jugador ha comenzado una partida.

<table>
<tr><td>Código</td><td>PU-06</td></tr>
<tr><td>Datos de entrada</td><td>En la pantalla se ingresaron los siguientes datos:<table><tr><th>Campo</th><th>Valor</th></tr><tr><td>Tecla</td><td>Up</td></tr></table></td></tr>
<tr><td>Resultado esperado</td><td>El personaje del videojuego tiene un impulso en contra de la gravedad y se reproduce su animación de salto.</td></tr>
</table>

Tabla 3.9: Ejemplo catálogo de pruebas

4. CAPÍTULO 4: OBSERVACIONES, CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan las observaciones, conclusiones y recomendaciones del proyecto que corresponden a la implementación del videojuego.

En la sección de Observaciones se procede a listar todos los puntos trascendentes y de importancia que ocurrieron en el proyecto del videojuego.

En la sección de Conclusiones se identifican las acciones realizadas para poder seguir con los objetivos que se han propuesto en el inicio.

En la sección de Recomendaciones se mencionan los consejos que servirán para mejorar la calidad de la solución en base a la experiencia del tesista en el presente proyecto.

4.1. Observaciones

Luego de haber implementado la solución siguiendo las metodologías propuestas, se hace necesario realizar las siguientes precisiones:

- La presente solución ha tenido como objetivo el desarrollo de un videojuego que permita estimular y mejorar las habilidades cognitivas de las personas.

- La solución desarrollada es una herramienta que estimula el aprendizaje en las personas y el desarrollo cognitivo.

- Le decisión de optar por utilizar software libre para la implementación de la solución (lenguaje Python) hizo que el proyecto sea viable debido a que para la obtención de licencias de software comercial se debe hacer un pago elevado.

- El entendimiento de la situación educativa actual en el Perú fue bastante importante por lo que el tesista debió buscar una gran cantidad de información acerca del tema para poder dominar a fondo la problemática que se está abordando.

- Para el proyecto desarrollado se utilizó una tecnología dominada por el tesista (Python), lo que facilitó el desarrollo, prescindiendo de tiempo para el aprendizaje de la tecnología.

- La presente tesis constituye la base para el desarrollo de un producto cuyo objetivo es potenciar el uso de la memoria y la inteligencia espacial; sin embargo, no profundiza en el estudio científico y/o experimental de ambas características.

4.2. Conclusiones

El presente proyecto ha sido desarrollado de manera exitosa, luego del cual se ha llegado a las siguientes conclusiones:

- La continua comunicación con el asesor y la entrega de los avances de la solución fueron muy importantes para un correcto desarrollo del proyecto. De esta forma se

obtuvo una muy buena retroalimentación que permitió realizar los ajustes tanto en el modelo de la solución, como en su implementación.

- Se han implementado exitosamente los algoritmos de Inteligencia Artificial que sirvieron para el control del movimiento y toma de decisiones de los personajes en el videojuego, verificando la factibilidad y eficiencia de este tipo de algoritmos aplicados en soluciones de esta naturaleza.

- La selección de una arquitectura adecuada permitió el desarrollo de una sistema de manera rápida y organizada.

- Culminado el proyecto de implementación, se puede concluir que gracias a la versatilidad y facilidades del lenguaje de programación Python, además de las funcionalidades ofrecidas por el framework Cocos2D, el desarrollo del videojuego se agilizó y pudo completarse oportunamente. También se puede afirmar que el uso de ambas plataformas resulta una buena combinación para el desarrollo de videojuegos.

- Según lo especificado, se elaboró un plan de pruebas detallado, el cual fue ejecutado, permitiendo identificar y subsanar los errores encontrados en el proceso de desarrollo.

4.3. Recomendaciones

La presente solución fue planificada a partir de la identificación de una oportunidad de mejora potencial en el ámbito educacional en el país. El videojuego implementado brinda la posibilidad de un desarrollo cognitivo a toda aquella persona que lo utilice. Adicionalmente, la versatilidad de la arquitectura y el diseño del videojuego permitirán incorporar características adicionales. A continuación se listan una serie de recomendaciones en relación al tema:

- Es recomendable que en caso de ajustes futuros se cuente con un ambiente de prueba. Este ambiente de prueba permitirá a los desarrolladores trabajar con mayor precisión y eficacia debido a que el juego pasará por un proceso de calidad antes de ser lanzado definitivamente.

- Para facilitar un mantenimiento de código eficiente, se recomienda realizar diversos comentarios en el código fuente. Con esto se logrará que cualquier desarrollador pueda brindarle mantenimiento al videojuego y no necesariamente dicho desarrollador debe haber formado parte del equipo original que implementó el juego.

- Se recomienda para una segunda versión del videojuego, incluir la opción de multijugador, con la cual se logrará la interacción entre varios jugadores y un mayor beneficio en algunos aspectos que no podrían cubrirse en un modo de juego individual.

- Otra recomendación para una futura versión del videojuego sería la inclusión de interfaces que permita interactuar con dispositivos móviles, además de incluir la opción que permita configurar la dificultad de los niveles.

BIBLIOGRAFÍA

[UNE001]
Datos Mundiales de Educación. VII Ed. 2010/11. Consulta: 28 de agosto de 2012.
<http://www.ibe.unesco.org/fileadmin/user_upload/Publications/WDE/2010/pdf-versions/Peru.pdf>

[OCD001] ORGANIZACIÓN PARA LA COOPERACIÓN Y EL DESARROLLO ECONÓMICO
El programa PISA de la OCDE "Qué es y para qué sirve". Consulta: 28 de agosto de 2012.
<http://www.oecd.org/pisa/39730818.pdf>

[EDU001] MINISTERIO DE EDUCACIÓN
Artículo: "OCDE presentó los resultados de la Evaluación Internacional PISA 2009". Consulta: 28 de agosto de 2012.
<http://www2.minedu.gob.pe/digesutp/formacioninicial/?p=558>

[SGA001] SERIOUS GAMES ASSOCIATION
Education. Consulta: 28 de agosto de 2012.
<http://www.seriousgamesassociation.com/games-for-learning/>

[SAL001] UNIVERSIDAD DE SALAMANCA
El uso de videojuegos retrasa el deterioro cognitivo en EA. Consulta: 28 de agosto de 2012.
<http://www.usal.es/webusal/node/7452/>

[HIP001] CIRCUNVALACIÓN DEL HIPOCAMPO
Memoria. Consulta: 28 de agosto de 2012.
<http://www.hipocampo.org/memoria.asp>

[COG001] HERRERA Clavero, Francisco
Habilidades cognitivas. Consulta: 28 de agosto de 2012.
<http://www.cprceuta.es/Asesorias/FP/Archivos/FP%20Didactica/HABILIDADES%20COGNITIVAS.pdf>

[IMU001] HERNÁNDEZ González, Eduardo
Las Inteligencias Múltiples. Consulta: 28 de agosto de 2012.
<http://www.psicologia-online.com/infantil/inteligencias_multiples.shtml>

[PYT001] PYTHON PROGRAMMING LANGUAGE – OFFICIAL WEBSITE
About Python. Consulta: 28 de agosto de 2012.
<http://www.python.org/about>

[VID001] VIDEOJUEGO
Consulta: 28 de agosto de 2012.
<http://www.3djuegos.com>

[C2D001] COCOS2D
Consulta: 28 de agosto de 2012.
<http://Cocos2D.org/index.html>

[ALG001] CORMEN, Thomas y Charles Leiserson
2009 *Introduction to Algorithms*. Tercera edición. Cambridge, MA: MIT Press.

[IAR001] TUPIA Anticona, Manuel Francisco
2009 *Fundamentos de Inteligencia Artificial.* Primera edición. Lima - Perú: Tupia Consultores y Auditores S.A.C.

[PMI001] PROJECT MANAGEMENT INSTITUTE
Consulta: 28 de agosto de 2012.
<http://www.pmi.org>

[RUP001] FASES DEL RATIONAL UNIFIED PROCESS
Consulta: 08 de setiembre de 2012.
<http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf>

[LUM001] JUEGOS DE EJERCITACIÓN DE RAZONAMIENTO Y MEMORIA
Consulta: 08 de setiembre de 2012.
<http://www.lumosity.com>

[PMI002] GUÍA DE LOS FUNDAMENTOS DE LA DIRECCIÓN DE PROYECTOS
Consulta: 08 de setiembre de 2012.
<http://gio.uniovi.es/documentos/software/GUIA_PMBok.pdf>

[HEU001] USING HEURISTICS TO EVALUATE THE PLAYABILITY OF GAMES
Consulta: 27 de febrero de 2013.
Heather Desurvire, Martin Caplan, Jozsef A. Toth (USA)

[HEU002] ENHANCING THE EXPLANATORY POWER OF USABILITY HEURISTICS
Consulta: 27 de febrero de 2013.
Jakob Nielsen (USA)