



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO DE SISTEMAS E INFORMÁTICA**

AUTOR: GONZÁLEZ SOTOMAYOR ITAMAR SOLEDAD

**TEMA: DISEÑO Y DESARROLLO DE UN VIDEOJUEGO EDUCATIVO CON
TÉCNICAS DE INTELIGENCIA ARTIFICIAL PARA LA PLATAFORMA
ANDROID APLICANDO LA METODOLOGÍA OOHDM. CASO DE ESTUDIO:
LABERINTO EN 3D**

DIRECTOR: ING. MARGARITA ELIZABETH ZAMBRANO R.

CODIRECTOR: ING. CARLOS TEIRON PRÓCEL.

SANGOLQUÍ, JULIO 2014

CERTIFICADO

Ing. Margarita Zambrano

Ing. Carlos Prócel

CERTIFICAN

Que el trabajo titulado “DISEÑO Y DESARROLLO DE UN VIDEOJUEGO EDUCATIVO CON TÉCNICAS DE INTELIGENCIA ARTIFICIAL PARA LA PLATAFORMA ANDROID APLICANDO LA METODOLOGÍA OOHDM. CASO DE ESTUDIO: LABERINTO EN 3D” realizado por el Srta. ITAMAR SOLEDAD GONZÁLEZ SOTOMAYOR, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas “ESPE”.

Sangolquí, julio del 2014

Ing. Margarita Zambrano

DIRECTOR

Ing. Carlos Prócel

CODIRECTOR

DECLARACIÓN DE RESPONSABILIDAD

GONZÁLEZ SOTOMAYOR ITAMAR SOLEDAD

DECLARO QUE:

El proyecto denominado "DISEÑO Y DESARROLLO DE UN VIDEOJUEGO EDUCATIVO CON TÉCNICAS DE INTELIGENCIA ARTIFICIAL PARA LA PLATAFORMA ANDROID APLICANDO LA METODOLOGÍA OOHDM. CASO DE ESTUDIO: LABERINTO EN 3D", ha sido desarrollado con base a una investigación exhaustiva respetando los derechos intelectuales de terceros, conforme a las fuentes que se incorpora en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, julio del 2014

González Sotomayor Itamar Soledad

AUTORIZACIÓN

Yo, GONZÁLEZ SOTOMAYOR ITAMAR SOLEDAD

Autorizo a la UNIVERSIDAD DE LA FUERZAS ARMADAS “ESPE”, la publicación, en la biblioteca virtual de la Institución del trabajo “DISEÑO Y DESARROLLO DE UN VIDEOJUEGO EDUCATIVO CON TÉCNICAS DE INTELIGENCIA ARTIFICIAL PARA LA PLATAFORMA ANDROID APLICANDO LA METODOLOGÍA OOHDM. CASO DE ESTUDIO: LABERINTO EN 3D”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, julio del 2014

González Sotomayor Itamar Soledad

DEDICATORIA

Dedico este trabajo de tesis a Dios, quien en su infinita misericordia me ha permitido llegar hasta este momento tan importante en mi formación profesional.

A mi hija Camila ya que su amor es la razón que me motiva a continuar día a día sin desfallecer, convertirme en su mejor ejemplo, ha forjado en mí un espíritu de fortaleza y perseverancia para cumplir mis objetivos y metas.

“El camino más difícil, nos puede fortalecer para cumplir una meta cuando esta es parte de nuestro destino”

Soledad González

AGRADECIMIENTOS

Le doy gracias a mi Dios por permitirme vivir esta alegría, disfrutar cada día y ser la persona que soy.

A mi hermosa Camila, quien con su amor incondicional y su tierna sonrisa ilumina cada día de mi vida, le doy gracias a Dios por tenerte, tú eres lo más importante en mi vida y hoy le doy gracias a Él porque estas a mi lado cuando hago realidad este sueño que estaba un poco postergado.

A mis padres Miguel y Nelly, por darme la vida y por ser ese apoyo incondicional, gracias por estar pendientes de mí en cada momento, y enseñarme a vivir con responsabilidad, dedicación y esfuerzo, mil gracias por la hermosa familia que me han dado.

A mis hermanos Nelly, Mónica y Miguel hemos crecido juntos y aprendimos a vivir, como amigos incondicionales compartiendo todo y aprendiendo que nuestras diferencias se convierten en riqueza cuando existe amistad y respeto, gracias por ser buenos hermanos.

A mi Directora de Tesis Ing. Margarita Zambrano y a mi Co-Director Ing. Carlos Prócel quienes me ofrecieron todos sus conocimientos y apoyo para la realización de proyecto.

Soledad González

INDICE DE CONTENIDOS

CERTIFICADO	I
DECLARACIÓN DE RESPONSABILIDAD.....	II
AUTORIZACIÓN	III
DEDICATORIA.....	IV
AGRADECIMIENTOS	V
ÍNDICE DE TABLAS	X
ÍNDICE DE FIGURAS	XI
RESUMEN	XIV
ABSTRACT	XV
GLOSARIO DE NOMENCLATURAS	XVI
CAPÍTULO 1	1
GENERALIDADES.....	1
1.1. FORMULACIÓN DEL PROBLEMA	1
1.2. IMPORTANCIA Y JUSTIFICACIÓN.....	2
1.3. OBJETIVOS	3
1.3.1. Objetivo General.....	3
1.3.2. Objetivos Específicos	3
1.4. ALCANCE	4
1.5. METODOLOGÍA	5
1.6. HERRAMIENTAS.....	8
CAPÍTULO 2	11
MARCO TEÓRICO.....	11
2.1 VIDEOJUEGOS EDUCATIVOS	11
2.1.1 El Videojuego como Actividad de Entretenimiento	11
2.1.2 Los Videojuegos como Fenómeno Social	21
2.1.3 Tipos de Videojuegos y Propuestas de Clasificación	26
2.1.4 La Polémica: Partidarios y Detractores	38
2.2 INGENIERÍA DE SOFTWARE	42

2.2.1	Definición.....	42
2.2.2	Marco de Trabajo para el Proceso	43
2.2.3.	Inteligencia Artificial – IA	45
2.3	METODOLOGÍA DE DISEÑO DE HIPERMEDIA ORIENTADA A OBJETOS (OOHDM)..	49
2.3.1	Historia	49
2.3.2	Definición.....	50
2.3.3	Características Fundamentales	51
2.3.4	Etapas o Fases.....	51
2.3.5.	Criterios de Selección de OOHDM	63
2.4.	Lenguaje Unificado de Modelado (UML)	64
2.4.1.	Definición.....	64
2.4.2.	Diagramas	64
CAPÍTULO 3		73
ANÁLISIS Y DISEÑO DEL JUEGO LÚDICO		73
3.1	ESPECIFICACIÓN DE REQUERIMIENTOS.....	73
3.1.1	Introducción.....	73
3.1.2	Identificación de Roles y Tareas.....	75
3.1.3	Especificación de Escenarios	76
3.1.4	Especificación de casos de uso por Actor	77
3.1.5	Requerimientos No Funcionales	84
3.2	DIAGRAMAS DE SECUENCIA	85
3.3	DISEÑO CONCEPTUAL.....	85
3.3.1	Diseño de los Archivos Planos	85
3.3.2	Diagrama de Clases	86
3.4	DISEÑO NAVEGACIONAL.....	88
3.4.1	Esquema Navegacional.....	89
3.4.2.	Esquema de Contextos Navegacionales.....	93
3.4.3.	Arquitectura del Sistema	94
3.5	DISEÑO DE INTERFAZ ABSTRACTA.....	95
3.5.1	Vista de Datos Abstractos	95
3.5.2.	Diagramas de Configuración	97

3.5.3. Diagramas de Estado	98
3.6. DISEÑO ESTÉTICO	98
3.6.1. Características de la Plantilla	98
3.6.2. Consideraciones de Diseño Gráfico	99
3.7. DISEÑO DE COMPONENTES	100
3.8. DIAGRAMA DE DESPLIEGUE	101
CAPÍTULO 4	102
IMPLEMENTACIÓN Y PRUEBAS DEL VIDEOJUEGO CON UNITY.....	102
4.1 UNITY.....	102
4.1.1 Definición.....	102
4.1.2 Características.....	102
4.1.3 Librerías de Unity	107
4.1.4 Estructura de la interfaz visual del videojuego hecho con Unity.....	112
4.2 CONSTRUCCIÓN DEL VIDEOJUEGO	114
4.2.1 Creación del Modelo 3D en Maya	114
4.2.2 Creación del Videojuego en Unity	115
4.3 DESARROLLO E IMPLANTACIÓN DE COMPONENTES PARA LA EL LABERINTO3D ..	118
4.4 PRUEBAS DE LA APLICACIÓN.....	121
4.5.1 Prueba de Contenido	122
4.5.2 Prueba de Interfaz de Usuario.....	123
4.5.3 Prueba de Navegación	124
4.5.4 Prueba de Componentes.....	124
4.5.5 Prueba de Configuración.....	125
4.5.6 Prueba de Seguridad.....	125
4.5.7 Prueba de Desempeño.....	125
4.6 DEPLOYMENT DE LA APLICACIÓN.....	126
CAPÍTULO 5	127
CONCLUSIONES Y RECOMENDACIONES	127
5.1 CONCLUSIONES.....	127
5.1.1. Conclusiones del Proyecto	127
5.1.2. Conclusiones Académicas	128

5.2. RECOMENDACIONES	129
5.2.1. Recomendaciones del Proyecto	129
5.2.2. Recomendaciones Académicas	131
BIBLIOGRAFÍA Y WEBGRAFÍA.....	132
ANEXOS	133
ANEXO 1. DIAGRAMAS DE SECUENCIA POR ACTORES.....	133
ANEXO 2. MANUAL DE USUARIO	137

ÍNDICE DE TABLAS

Tabla 1: Herramientas de Software.....	8
Tabla 2: Alegaciones a favor y en contra de los videojuegos.....	41
Tabla 3: Etapas de la Metodología OOHDM.....	52
Tabla 4: Configurar el videojuego.....	78
Tabla 5: Resolver el laberinto de forma manual.....	80
Tabla 6: Resolver el laberinto de forma automáticamente.....	82
Tabla7:Clase Navegacional–Nodo Formulario de la Configuración del Juego.....	89
Tabla8:Navegacional–Nodo del Laberinto de Modo Manual.....	90
Tabla 9: Clase Navegacional – Nodo del Laberinto de Modo Automático.....	90
Tabla10: Clase Navegacional – Nodo de Navegación y Entorno del Juego.....	91
Tabla 11: Clase Navegacional – Nodo de Puntajes del Juego.....	92

ÍNDICE DE FIGURAS

Figura 1. Estratos de la Ingeniería de Software.	42
Figura 2. Esquema del Diseño Navegacional	59
Figura 3. Ejemplo de un Caso de Uso.	65
Figura 4. Ejemplo de un Diagrama de Secuencia.	67
Figura 5. Ejemplo de una Clase.	68
Figura 6. Ejemplo de un Diagrama de Clases.	70
Figura 7. Ejemplo de un Diagrama de Despliegue	71
Figura 8. Ejemplo de un Diagrama de Componentes.	72
Figura 9. Actores - Laberinto3d.	76
Figura 10. Caso de Uso para El Jugador del Laberinto3d.	78
Figura 11. Archivos Planos del Laberinto3d.	85
Figura 12. Diagrama de Clases del Laberinto3d.	87
Figura 13. Modelo de Clases Navegacional del Laberinto3d.	93
Figura 14. Esquema de Contexto - Menú Principal.	94
Figura 15. Arquitectura del Laberinto3d.	94
Figura 16. Vista Abstracta del Nodo del Formulario de Configuración del Juego.	95
Figura 17. Vista Abstracta del Nodo del Juego Manual.	96
Figura 18. Vista Abstracta del Nodo del Juego Automático.	96
Figura 19. Vista Abstracta del Nodo del Formulario de la Navegación y Entorno del Juego.	97
Figura 20. Vista Abstracta del Nodo del Formulario de los Puntajes del Juego.	97
Figura 21. Diagrama de Estado del Laberinto3d.	98
Figura 22. Diseño Estético – Organización de los Elementos.	99
Figura 23. Diseño de Componentes del Laberinto3d.	100
Figura 24. Diagrama de Despliegue del Laberinto3d.	101
Figura 25. El Ícono de Carga del Paquete Unity.	103
Figura 26. Interfaz de Usuario de Unity.	103
Figura 27. Gizmo de Perspectiva 3d.	104
Figura 28. Configuración de la Visualización.	105
Figura 29. Botones de Control.	106
Figura 30. Creación de Escenas.	108
Figura 31. Generador de Terrenos.	109
Figura 32. Seteo del Renderizado.	110
Figura 33. Controlador de Primera Persona.	111
Figura 34. Estructura de la Interfaz Visual del Videojuego.	112
Figura 35. Elementos del Videojuego.	114
Figura 36. Ventana de la Aplicación del Juego en Ambiente Windows.	119
Figura 37. Ventana de la Aplicación del Juego en Ambiente Móvil.	120
Figura 38. Proceso de Prueba	121
Figura 39. Diagrama de Secuencia – Configurar El Videojuego.	134
Figura 40. Diagrama de Secuencia – Resolver El Laberinto de Modo Manual.	135
Figura 41. Diagrama de Secuencia – Resolver El Laberinto de Modo Automático.	136
Figura 42. Ventana del Menú Principal.	137
Figura 43. Ventana del Menú Principal de la Opción Manual.	138
Figura 44. Ventana de la Opción Trial Manual del Juego.	139
Figura 45. Ventana de la Opción Principiante Manual del Juego.	140
Figura 46. Ventana de la Opción Intermedio Manual del Juego.	141

Figura 47. Ventana de la Opción Avanzado Manual del Juego.	142
Figura 48. Ventana del Puntaje de La Opción Manual.	143
Figura 49. Ventana del Menú Principal de la Opción Automática.	144
Figura 50. Ventana de la Opción Trial Automática del Juego.	145
Figura 51. Ventana de la Opción Principiante Automática del Juego.	146
Figura 52. Ventana de la Opción Intermedio Automática del Juego.	147
Figura 53. Ventana de la Opción Avanzada Automática del Juego.	148
Figura 54. Ventana del Puntaje de la Opción Automática.	149

ÍNDICE DE ANEXOS

Anexo 1: Diagramas De Secuencia Por Actores.....	133
Anexo 2: Manual de Usuario	137

RESUMEN

Los videojuegos lúdicos o educativos es uno de los tipos de videojuegos que más se aplican como parte del proceso de enseñanza-aprendizaje en los niños, para el desarrollo del pensamiento y para el desarrollo psicomotriz de los mismos. Las aplicaciones de entretenimiento en 3D son las más apetecidas por niños y jóvenes cuando de jugar se trata y en los momentos de ocio. No cabe duda que la industria del videojuego es una de las que más ingresos genera a nivel mundial y los consumidores de este tipo de aplicaciones son millones de usuarios en todo el mundo.

Este proyecto de tesis, ha tenido como objetivo principal colaborar con los grupos de desarrollo de videojuegos educativos a través del análisis, diseño y desarrollo de un juego didáctico de razonamiento abstracto que consiste en un Laberinto en 3D, para ayudar al desarrollo del pensamiento de niños entre 7 y 11 años, utilizando el Game Engine Unity con el lenguaje de programación C#.NET y aplicando la metodología de software OOHDM (Metodología de Diseño de Hipermedia Orientado a Objetos), desarrollada por Daniel Schwabe y Gustavo Rossi, la cual consta de cuatro etapas: Diseño Conceptual, Diseño Navegacional, Diseño Abstracto de Interfase e Implementación.

Palabras Claves: Juegos Educativos, laberintos en 3D con IA, Aplicaciones 3D de entretenimiento, OOHDM, UML, Modelado de juegos en 3D.

ABSTRACT

The playful or educational video games it is one of the types of video games that more they are applied as part of the teaching-learning process in the children, for the development of the thought and for the development of mobility and psyche of the same ones. The entertainment applications in 3D are those more felt like by children and young when of being played it tries and in the leisure moments. It doesn't fit doubt that the industry of the video game is one of those that more revenues generate at world level and the consumers of this type of applications are in the entire world millions of users.

This thesis project has had as main objective to collaborate with the groups of development of educational video games through the analysis, design and development of a didactic game of abstract reasoning that it consists on a Labyrinth in 3D, to help to the development of the thought of children between 7 and 11 years, using the Game Engine Unity with the programming language C#. NET and applying the software methodology OOHDM (Object Oriented Hypermedia Design Methodology) developed by Daniel Schwabe and Gustavo Rossi, which consists of four stages: Conceptual Design, Navigation Design, Abstract Design of Interface and Implementation.

KeyWords: Educational games, labyrinths in 3D with AI, 3D Entertainment Applications, OOHDM, UML, 3D Games Modeling.

GLOSARIO DE NOMENCLATURAS

- Actor: Es un objeto que se encuentran fuera del sistema a modelar. Representan entes que tienen necesidad de intercambiar información con el sistema; pueden ser instanciados por usuarios, dispositivos u otros sistemas.
- Archivos Planos XML(*XML Files*): Conjunto de información organizada que contiene una colección de registros donde un sistema puede buscar, reescribir, clasificar, borrar, añadir información a los archivos cuyo formato se basa etiquetas de hipertexto.
- Blog (*Web log*): Diario en formato Web. Puede ser un diario personal o un conjunto de noticias, ordenado por fecha.
- Caso de Uso: Descripción a detalle de las actividades y procesos necesarios para el desarrollo de un sistema o aplicación.
- Copyleft: Derecho de autor que consiste en permitir la libre distribución de copias y versiones modificadas de una obra o trabajo.
- Daemon (*Demonio*): Clase especial de programa que corre en segundo plano en vez de ser controlado directamente por el usuario. Funciona sin tener relación con una Terminal o consola y, consecuentemente, sin interactuar con un humano.
- Front-end y Back-end: El front-end es la parte del software que interactúa con el usuario y el back-end es la parte que procesa la entrada desde el front-end y es gestionada por el Administrador del software.

- GPL (*General Public License*): Licencia que permite el uso y modificación del código fuente para desarrollar software libre, pero conserva los derechos de autor.
- Hipermedia: Toma su nombre de la suma de hipertexto y multimedia, una red hipertextual en la que se incluye no sólo texto, sino también otros medios: imágenes, audio, vídeo, etc. (multimedia).
- HTML (*HyperText Markup Language*): Lenguaje basado en marcas que indican las características del texto, utilizado para definir documentos de hipertexto en Web.
- HTTP (*HyperText Transfer Protocol*): Protocolo cliente-servidor utilizado para el intercambio de páginas Web (HTML).
- Internet: Es un sistema mundial de redes de computadoras, integrado por las diferentes redes de cada país del mundo y por medio del cual un usuario con los permisos apropiados puede obtener información de un servidor o computadora personal y tener comunicación directa con otros usuarios.
- Metadatos (*Datos sobre los datos*): Información que describe el contenido de los datos.
- Modelo: Es la conceptualización de un evento, un proyecto, una hipótesis, el estado de una cuestión y se representa como un esquema que posee símbolos descriptivos de características y relaciones más importantes.

- Motor de Búsqueda (*Buscador*): Es un conjunto de programas coordinados que se encargan de visitar cada uno de los sitios que integran el Web, empleando los propios hipervínculos contenidos en las páginas Web para buscar y leer otros sitios.
- Open Source (*Código abierto o código libre*): Software que distribuye de forma libre su código fuente y los desarrolladores pueden hacer variaciones, mejoras o reutilizaciones en otras aplicaciones. También conocido como *free software*.
- Página Web: Es un documento electrónico que contiene información específica de un tema en particular y es almacenado en algún sistema de cómputo que se encuentre conectado al Internet para que pueda ser consultada.
- Sitio Web: Conjunto de páginas Web referentes a un tema en particular, que incluye una página inicial de bienvenida, con un nombre de dominio y dirección en Internet. Empleado por las empresas para ofertar sus bienes y servicios.
- Tags (*Etiqueta*): Es un conjunto de caracteres que se añade a un elemento de datos para identificarlo. Por ejemplo: `<p>Esto es un párrafo</p>`
- URL (*Uniform Resource Locator*): Dirección de un recurso en la Web. Tiene el formato `protocolo://máquina.dominio:port/ruta/recurso`. Por ejemplo `http://www.uoc.edu/dt/index.html` donde no se indica el puerto porque el protocolo HTTP tiene uno por defecto (80).

- Usuario: Ente humano que usa al sistema. Un mismo usuario puede actuar como instancias en varios actores diferentes, es decir, puede jugar diferentes roles.
- VRML: Virtual Reality Modeling Language (Lenguaje para el modelado de realidad virtual) es un lenguaje que se utiliza para definir ambientes en tercera dimensión, para ser desplegados a través de navegadores de Internet.
- Web: Sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (*el Web*) se refiere a un sitio Web entero, mientras que en femenino (*la Web*) se refiere a una página Web concreta dentro del sitio Web.
- WebApps (*Aplicación Web*): Es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet su estructura general es de tres capas: El navegador Web es la primera capa, un motor usando alguna tecnología Web dinámica (ejemplo: PHP) la capa de en medio, y una base de datos como última capa.
- Workflow (*Flujo de trabajo*): Estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.

- WYSIWYG (*What You See Is What You Get* o *lo que ves es lo que obtienes*): Aplicado a la edición de páginas por ejemplo HTML significa trabajar con un documento con el aspecto real que tendrá y no únicamente con sus códigos.

CAPÍTULO 1

GENERALIDADES

1.1. Formulación del Problema

En la actualidad se vive en un mundo en lo que todo o casi todo está impregnado de tecnología. Los niños viven con total normalidad esta circunstancia. De esta forma conviven y se desenvuelven en ella adaptándola sin dificultad para su uso cotidiano. Sin embargo, existe la sensación de quererles proteger de estas tecnologías, y más bien enseñarles a usar adecuadamente las tecnologías como: Internet, Aulas virtuales, Multimedia, Realidad Virtual, Video Conferencia, Juegos Didácticos en 2D y 3D, etc.

Una gran desventaja en todo ámbito es el crecimiento acelerado de la tecnología y se debe ser consciente en preparar a nuestros alumnos para el presente y el mañana, por tanto, si se reflexiona sobre las nuevas demandas sociales es necesario formar más acorde con los tiempos actuales. La tecnología 3D y la Realidad Virtual están revolucionando el proceso de enseñanza-aprendizaje de los niños en los países desarrollados como juegos en 3D, laboratorios virtuales, simuladores 2D y 3D, kits de robótica, etc., que ayudan al desarrollo psicomotriz y de la inteligencia de los niños.

Lastimosamente aún esta tecnología es muy costosa y no está al alcance de todos, especialmente de los países en vías de desarrollo donde los centros educativos no cuentan todavía con el soporte de estos tipos de tecnología, en especial para el desarrollo psicomotriz de los niños, donde las aplicaciones de

juegos lúdicos en 2D y en 3D ayudan al desarrollo del pensamiento. Muchas escuelas de este país, principalmente los de bajos recursos económicos, no cuentan con medios tecnológicos que se pueden aplicar en la educación como: Internet, Aplicaciones de entornos 3D, Aplicaciones de Multimedia, Aulas Virtuales, Video Conferencia, etc. En este sentido se va a desarrollar una aplicación de 3D y realidad virtual como una herramienta de apoyo para los profesores y estudiantes en el proceso de enseñanza-aprendizaje y estar acorde con la tecnología actual en la materia de desarrollo del pensamiento para niños de Educación Básica.

1.2. Importancia y Justificación

Aprovechando la tecnología de aplicaciones en entornos 3D, que permite integrar varios medios como: texto, imágenes, sonido, video, animación en 3D, es necesario adecuarles en el uso correcto de este tipo de tecnología y desarrollar una aplicación en entornos 3D interactiva, el mismo que servirá como un soporte en la en la Educación Básica de la materia de Desarrollo del Pensamiento.

El facilitador y el alumno son elementos importantes en proceso de enseñanza – aprendizaje, las tecnologías son sólo un elemento más como pueden serlo todas las variables relacionadas con el contexto. En este sentido, es necesario disponer de estrategias didácticas como el uso de materiales educativos tecnológicos que son una herramienta de apoyo en el proceso del aprendizaje.

La aplicación que se desarrollará se fundamentará en las bases teóricas y prácticas, propuestas por el Dr. Pere Marqués Graells de la Universidad de Barcelona en el área de Psicopedagogía, quien justamente plantea que los videojuegos educativos tanto en 2D como en 3D ayudan al desarrollo de la inteligencia abstracta y psicomotricidad de los niños, desde el punto de vista psicopedagógico.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un videojuego educativo con técnicas de inteligencia artificial para la plataforma Android aplicando la Metodología OOHDM. Caso de Estudio: Laberinto en 3D.

1.3.2. Objetivos Específicos

- Fundamentar el marco teórico acerca de aplicativos de software 3D, juegos didácticos y motores de juegos.
- Documentar las distintas fases de la metodología OOHDM.
- Realizar el análisis y el diseño de la aplicación 3D utilizando la metodología OOHDM con UML.
- Construir el modelo arquitectónico 3D de la aplicación con una herramienta de diseño 3D.
- Desarrollar la aplicación 3D utilizando la metodología OOHDM con UML.

- Utilizar las técnicas de inteligencia artificial de planificación y sistemas de reacción basados en reglas para resolver un laberinto en 3D.

1.4. Alcance

El juego didáctico a desarrollarse se orienta en el proceso de enseñanza – aprendizaje del razonamiento abstracto de niños entre 7 y 11 años, expresado mediante la utilización de herramientas de actualidad tanto para la edición, digitalización, renderizado e integración de la información en un entorno de desarrollo 3D.

La temática del juego a desarrollarse, estará acorde al contenido de la materia de Desarrollo del Pensamiento; dicha temática estará expresada mediante la utilización de elementos de multimedia como: texto, imagen, sonido, y animación 3D, que ayudarán para el mejor entendimiento de la materia por parte del profesor y del alumno.

El juego didáctico de razonamiento lógico y espacial será un juego de laberintos con tres niveles de dificultad (básico, intermedio y avanzado). El juego también contempla el manejo de registro de usuarios y de los tiempos que el usuario se demora en salir del laberinto, los mismos que serán registrados en archivos planos. El juego además podrá mediante inteligencia artificial, generar automáticamente los laberintos según un nivel de dificultad y tendrá una opción de resolver automáticamente el mismo, donde se utilizarán técnicas de inteligencia artificial de planificación y sistemas de acción y reacción

basados en el algoritmo de la mano derecha cuya aplicación permitirá resolver estos laberintos.

El juego didáctico de razonamiento lógico y espacial será diseñado para correr en plataforma Android y podrá ser manipulado en una Tablet.

Junto al aplicativo de software se entregará un manual de usuario que permitirá entender el manejo del mismo.

1.5. Metodología

Para el desarrollo de esta Aplicación Multimedia se adoptará la Metodología de **Diseño de Hipermedia Orientado a Objetos (OOHDM)**, desarrollado por Daniel Schwabe y Gustavo Rossi. Esta metodología básicamente consta de cuatro etapas: Diseño Conceptual, Diseño Navegacional, Diseño Abstracto de Interfase e Implementación. Cada etapa de la concepción define un esquema objeto específico en el que se introducen nuevos elementos o clases.

En la **primera etapa** se construye un esquema conceptual representado por los objetos de dominio o clases y las relaciones entre dichos objetos. Se puede usar un modelo de datos semántico estructural, como el modelo de entidades y relaciones. El modelo OOHDM propone como esquema conceptual basado en clases, relaciones y subsistemas.

En la **segunda etapa**, se define la estructura de navegación a través del hiperdocumento mediante la realización de modelos navegacionales que representen diferentes vistas del esquema conceptual de la fase anterior. El Diseño Navegacional se expresa, también con un enfoque orientado a objetos,

a través de dos tipos de esquemas o modelos: el denominado esquema de clases navegacionales, con las posibles vistas del hiperdocumento a través de unos tipos predefinidos de clases, llamadas navegacionales, como son los "nodos", los "enlaces", y otras clases que representan estructuras o formas alternativas de acceso a los nodos, como los "Índices" y los "recorridos guiados"; y el esquema de contexto navegacional, que permite la estructuración del hiperespacio de navegación en subespacios para los que se indica la información que será mostrada al usuario y los enlaces que estarán disponibles cuando se acceda a un objeto u nodo en un contexto determinado.

La **tercera etapa** está dedicada a la especificación de la interfaz abstracta. Así, se define la forma en la cual deben aparecer los contextos navegacionales. También se incluye aquí el modo en que dichos objetos de interfaz activarán la navegación y el resto de funcionalidades de la aplicación, esto es, se describirán los objetos de interfaz y se los asociará con objetos de navegación. La separación entre el diseño navegacional y el diseño de interfaz abstracta permitirá construir diferentes interfaces para el mismo modelo navegacional.

En la **cuarta etapa**, es en sí la implementación del hiperdocumento o sistema hipermedial diseñado, es decir, la concreción de los modelos navegacionales y de interfase en objetos particulares con sus correspondientes contenidos y sus posibilidades de navegación. Aunque, al utilizar un enfoque de orientación a objetos podría parecer conveniente que la implementación se hiciera en un entorno de construcción de hiperdocumentos también orientado a objetos, debido al carácter abstracto del diseño, sin embargo ésta puede

hacerse fácilmente en otros entornos hipermediales que permitan trabajar con el lenguaje HTML.

Para el desarrollo de este proyecto en lo que respecta al Marco Teórico se va a utilizar una metodología de trabajo basada en la investigación Bibliográfica de fuentes de Información, y consultas en Internet. Mediante esta etapa del proyecto se pretende obtener la información necesaria y válida que permita establecer el marco teórico referencial, el cual proporcione el soporte teórico – técnico necesario para la consecución del proyecto.

Se cumplirán las siguientes actividades como parte de una metodología de desarrollo estándar:

- Análisis
- Diseño
- Desarrollo
- Pruebas

El sistema a ser desarrollado se compone de un componente hipermedial para correr sobre la plataforma Android. La Metodología OOHDM se aplicará en conjunto con la Ingeniería de Software y UML (Lenguaje de Modelamiento Unificado). Además se aplicarán como Técnicas de Inteligencia Artificial, los Sistemas de Planificación y los Sistemas de Reacción Basados en Reglas, en conjunto con la Regla de la Mano Derecha para resolver laberintos.

1.6. Herramientas

Se definen de acuerdo a las dos capas que posee la aplicación: Presentación, y Reglas del Negocio. En la Tabla 1 se resumen las principales herramientas de software utilizadas.

Tabla 1. Herramientas de Software

Sistema Operativo Windows

Nombre	Descripción
Windows 7 y 8	El Sistema Operativo Windows 7 y 8 son parte de la familia de sistemas operativos desarrollados por la empresa Microsoft para gestionar los recursos de hardware y proveer servicios a los programas de aplicación.

Lenguaje de Programación

Nombre	Descripción
C# .NET	Lenguaje de programación orientado a objetos desarrollado por Microsoft y por el Proyecto GNU MonoTouch. Este lenguaje toma lo mejor del C/C++, del Java y del Visual Basic, y está orientado al desarrollo de todo tipo de aplicaciones de software para plataformas desktop, web, dispositivos móviles y consolas de videojuegos.

Motor de Juegos 3D (3D Game Engine)

Nombre	Descripción
Unity	Es un motor de juegos 3D para el desarrollo de videojuegos. Está disponible para la Plataforma Windows, LINUX y Mac OSX, y permite crear juegos para Windows, Mac, Xbox 360, PlayStation 3, Wii, iPad y iPhone, y para la plataforma Android.

Software de Modelado y Animación 3D

Nombre	Descripción
Autodesk Maya	Es una potente solución integrada de modelado 3D, animación, efectos y renderización. Dado que Maya se basa en una arquitectura abierta, es posible programar o aplicar guiones sobre sus trabajos utilizando una API (Interfaz de Programación de Aplicaciones) extensa y bien documentada.

Para el Manejo de Archivos Planos

Nombre	Descripción
XML	XML proviene de las siglas en inglés de eXtensible Markup Language (Lenguaje de Marcas Extensible) y es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). El XML es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML).

Visio como Herramienta Case

Nombre	Descripción
Microsoft Visio	Es un software de dibujo vectorial. Visio comenzó a formar parte de los productos de Microsoft cuando fue adquirida la compañía Visio en el año 2000. Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y también permite iniciar al usuario en los lenguajes de programación.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Videojuegos Educativos

2.1.1 El Videojuego como Actividad de Entretenimiento

Todas las personas pueden tener una idea aproximada de lo que se quiere expresar cuando alguien dice que juega con videojuegos. Antes de hacer un análisis concienzudo de esta actividad lúdica, es necesario conocer el sentido y significado de lo que representa el concepto de juego.

Desde la Antigüedad, donde se hallan ejemplos concretos de esta preocupación por el estudio del juego¹ (Gorris, 1977), hasta nuestros días, filósofos, antropólogos, sociólogos, historiadores, psicólogos, pedagogos, etc., han prestado una atención específica al juego al tiempo que han desarrollado sus propias teorías interpretativas sobre el mismo, y especialmente sobre sus potencialidades. Si bien existen interesantes trabajos de revisión y síntesis de estas teorías², el interés de revisar las principales aportaciones sobre el juego se ha centrado fundamentalmente, en tratar de descubrir en la actividad del juego aquellos rasgos o características que la diferencian de otras actividades. Lo que se persigue es explicar a qué se refiere cuando se dice que los niños y jóvenes “juegan” con videojuegos.

¹J. M. Gorris hace referencia a las obras de Platón y Aristóteles donde se reconoce el valor práctico del juego en la educación del niño (Gorris, 1997: 9).

²Al respecto pueden consultarse, entre otras, las obras de Gorris (1997), Elkonin (1998), López Rodríguez (1998), Cañeque (1999) o López Mantalla (1993).

Como señala Borja al inicio de una de sus obras (Borja, 1980), “el juego, al igual que el lenguaje, es una constante antropológica que se encuentra en todas las civilizaciones y en todas las etapas de cada civilización.” Son muchos los ejemplos de personas jugando que se han ido recogiendo entre los restos arqueológicos, pinturas, escritos, etc., y que constatan la importancia del juego en la humanidad. Resulta casi imposible determinar en qué momento exacto nació el juego, pero dadas las evidencias de la práctica de juegos existentes, nadie duda en afirmar que el juego fue anterior al juguete, es decir, al objeto/instrumento utilizado para jugar. Es probable que los primeros juegos fueran de carácter competitivo, como pruebas de fuerza o habilidad, luchas, etc. posiblemente enfrentamientos durante la caza con la recompensa de obtener la pieza (López Oneto y Ortega, 1982).

Además de su constancia histórica, el juego también se manifiesta a lo largo de las distintas etapas vitales de la persona, hecho que confirma la idea de que “a todas las personas, tengan la edad que tengan, les gusta jugar” (Martín et al., 1995) y por consiguiente, juegan. El juego, además de haber sido ampliamente estudiado por el papel fundamental que representa en el desarrollo del niño, cumple funciones diferentes y adquiere significados diversos para cada persona según el período evolutivo en el que se halle.

No es de extrañar, entonces, que buscando en diccionarios y enciclopedias se halle una enorme variedad de acepciones y definiciones para la palabra “juego”. Sin embargo tal diversidad no impide que todo el mundo sea capaz de reconocer un juego cuando lo ve o lo practica. Parece evidente que jugar es algo distinto a todo lo demás. La cuestión estriba, pues, en conocer

cuáles son los rasgos que nos permiten identificar y diferenciar esta actividad. Entre la multitud de definiciones existentes sobre el término destacan las propuestas de Huizinga y Caillois.

Johan Huizinga³, autor de la conocida obra "Homo Ludens", ha sido un referente constante para los estudiosos del tema. En esta obra Huizinga define el juego como "una acción u ocupación libre, que se desarrolla dentro de unos límites temporales y espaciales determinados, según reglas absolutamente obligatorias, aunque libremente aceptadas, acción que tiene su fin en sí misma y va acompañada de un sentimiento de tensión y alegría y de la conciencia de "ser de otro modo" que en la vida corriente. Definido de esta suerte, el concepto parece adecuado para comprender todo lo que se denomina juego en los animales, en los niños y en los adultos: juegos de fuerza, juegos de cálculo y de azar, exhibiciones y representaciones." (Huizinga, 1996). Además en esta obra hace un estudio del juego como un fenómeno cultural y no simplemente en sus aspectos biológicos, psicológicos o etnográficos, concibiéndolo como una función humana tan esencial como la reflexión y el trabajo, mostrando así la insuficiencia de las imágenes convencionales del *homo sapiens* y el *homo faber*. Considerando el juego desde los supuestos del pensamiento científico-cultural, lo ubica como génesis y desarrollo de la cultura. En sus propias palabras: "Porque no se trata, para mí, del lugar que al juego corresponda entre las demás manifestaciones de la cultura, sino en qué grado la cultura misma ofrece un carácter de juego." (*Introducción a modo de prólogo - Homo Ludens*).

³La notoriedad de Huizinga se debe a dos de sus obras: *El otoño de la Edad Media (Herfsttij der Middeleeuwen- 1919)* y *Homo Ludens (1938)*.

Roger Caillois⁴, cuya obra “Los juegos y los hombres: La máscara y el vértigo” data de 1958, parte de los planteamientos sobre el juego defendidos por Huizinga y desarrolla su propia concepción del juego. Para Callois el juego es “esencialmente una ocupación separada, cuidadosamente aislada del resto de la existencia y realizada por lo general dentro de los límites precisos de tiempo y de lugar.” (Caillois, 1986). Este autor hace hincapié en la identificación y descripción de las características de la actividad del juego.

En las aportaciones de ambos autores se encuentra una serie de cualidades del juego, que lo caracterizan y diferencian del resto de actividades. Se trata básicamente de lo que se ha denominado una concepción del juego como “cualidad intrínsecamente motivada” (Haywood et al., 1993). Sin embargo, existen otras características del juego que no se puede obviar y que se hallan, no tanto en las cualidades de la actividad sino en las consecuencias de su práctica, en sus potencialidades; en definitiva, en su funcionalidad extrínsecamente motivada.

Esta última distinción es fundamental puesto que permitirá definir las características del juego desde dos perspectivas diferenciadas y complementarias: sus cualidades intrínsecas y las funciones que cumple para el individuo y la sociedad en general.

Empezando por las primeras, por sus cualidades intrínsecas, el juego viene caracterizado por una serie de aspectos básicos:

⁴Caillois es un escritor de libros de filosofía, sociología, psicología y pedagogía; autor de más de veinte obras literarias, entre las que destacan: *Los juegos y los hombres: la máscara y el vértigo* (1958), *El Mito y los Hombres* (1938) y *El hombre y lo sagrado* (1939), entre otras.

A) El juego es una actividad libre. El juego es acción, como también lo son el trabajo y otras muchas actividades. Pero, además, es una acción auto elegida y realizada libremente (Franch y Martinell, 1994). Implica, necesariamente, que la persona acceda a él de forma voluntaria, del mismo modo que también voluntariamente pueda dejar de jugar.

El hecho de que quien juega lo hace con libertad constituye en esencia la primera cualidad que permite diferenciar a esta actividad de muchas otras obligaciones a las que se está sometido en la vida diaria.

B) El juego es autotélico. El carácter voluntario y libre del juego que conduce directamente a otra de sus características fundamentales: la finalidad del juego está en sí mismo (Huizinga, 1996). Es el proceso del propio juego y no el producto final lo que verdaderamente interesa. “...Cuando una persona juega (de forma libre, voluntaria y espontánea), no persigue ninguna finalidad particular. Sólo desea jugar...” (López Matallana, 1993).

Caillois, por su parte, define el juego como una actividad improductiva, esencialmente estéril, que no crea bienes ni riqueza. Su desarrollo es incierto, no puede determinarse, al igual que su resultado tampoco puede fijarse previamente puesto que es el propio jugador quien libremente inventa. Es esta incertidumbre la que al mismo tiempo mantiene viva la necesidad de seguir jugando y provoca la improductividad de la actividad lúdica.

En definitiva, el juego es una actividad carente de un fin en sí misma (Hetzer, 1978) aunque presente potencialidades ampliamente explotadas para la consecución de ciertas finalidades, en su gran mayoría educativas. Pero tales fines no se hallan en la actividad lúdica en sí, sino fuera de ella, por lo que el jugador no tiene porqué compartírselos necesariamente. El juego existirá si quien lo practica lo hace libremente y sin finalidad extrínseca, siendo otra cuestión si alguien externo al juego aprovecha el potencial de esta actividad lúdica para la consecución de determinados objetivos. En palabras de De Garay (1994): “El placer de un juego se termina en sí mismo. Se puede volver a jugar. Se puede jugar a otro juego. Pero un juego se agota en sí mismo. Un juego no es medio para otra cosa, sino que es un fin acabado.”

C) El juego es una actividad placentera. El juego, tal y como dice Borja, está unido a la risa, el humor, la diversión... y el placer es esencial en él. Se afirma que es una de las fuentes de placer con que cuenta el ser humano puesto que su práctica siempre se acompaña de una sensación de satisfacción.

Es necesario puntualizar que, al hablar de esta sensación de satisfacción, se hace referencia a la sensación de placer como resultado de la actividad lúdica en sí misma. Una de las características definitorias del juego era su carácter autotélico, su falta de finalidad extrínseca, por lo que la sensación de satisfacción no se corresponde con la sensación de fracaso o éxito en el sentido de alcanzar algún objetivo determinado. Aunque dentro del juego no todo sea felicidad, éste debe ser auto

elegido o aceptado con suficiente grado de libertad para resultar placentero (Ortega Ruiz, 1992).

Esta sensación de placer que caracteriza al juego, en términos generales, se traduce en risa, humor, diversión, disfrute y entretenimiento. Aun siendo conscientes de la multitud y variedad de motivaciones personales que pueden llevarnos a jugar, es muy probable que una de las principales razones común a todos nosotros sea el hecho de divertirnos. Por ello, cuando alguien siente la insatisfacción en el juego (por ejemplo, al haber perdido) se puede decir que esté jugando en el pleno sentido de la palabra; para esta persona la actividad lúdica tiene un objetivo concreto, distinto de la actividad lúdica, que no está siendo satisfecho y le impide disfrutar plenamente del juego en sí (López Matallana, 1993).

D) El juego es una actividad ficticia. Al decir que el juego es una actividad ficticia se está apuntando a su poder de evasión temporal. En el juego los participantes tienen una “conciencia específica de realidad secundaria o de franca irrealidad en comparación con la vida corriente” (Caillois, 1986).

Tal como indica Huizinga, el juego es algo distinto a la vida corriente; de hecho supone siempre un escape de ella hacia una realidad - la realidad del juego - que es particular de cada uno y totalmente ajena a la esfera de lo real.

En la situación de juego se puede encontrar realidades materiales y sociales, tales como los elementos del juego o los participantes. Pero el juego no es posible sin el hecho simbólico, sin el hecho imaginario, sin la transformación de la realidad inmediata con la que - o sobre la que - se juega (Franch y Martinell, 1994).

- E) El juego es una actividad limitada en el tiempo y el espacio.** El juego, en tanto acción que se consume en sí misma, presenta una estructura: comienza, se desarrolla y, en un momento determinado, se acaba. Se trata de una acción circunscrita en unos límites de espacio y tiempo precisos. Esta delimitación temporal y espacial, también destacada por Huizinga, lleva a Caillois a hablar del juego como una actividad separada.
- F) El juego está regulado por unas reglas o normas.** El hecho de que el juego tienda a elaborar sus propias normas (Haywood et al., 1993) es otra de sus características esenciales. Esta idea de regulación del juego es apuntada también por Huizinga y Caillois, quienes entienden que cada juego, en el seno de su propia realidad, cuenta con sus leyes y normas específicas que deben ser aceptadas o consensuadas libremente por todos y cada uno de los jugadores, puesto que si no se cumplen el juego se acaba.
- G) El juego es una actividad global.** Finalmente se piensa que otra cualidad específica del juego se halla en la experiencia global que éste representa para la persona. Entre los distintos autores que se ocupan del tema existe consenso a la hora de entender que “la riqueza del juego se

convierte en riqueza real en la medida en que se lo preserva como totalidad y que no se quieren aislar los elementos puntuales” (Franch y Martinell, 1994).

Todas y cada una de las características enunciadas anteriormente aparecen asociadas entre sí. Es difícil desligar, por ejemplo, la libertad del juego de la sensación de placer, puesto que el juego debe entenderse como una experiencia global donde todos los elementos materiales, temporales, espaciales y personales están relacionados entre sí. Pero lo más importante del carácter globalizador del juego es la actitud de los participantes en su práctica. Es esta actitud lo que verdaderamente da sentido a la actividad transformándola en juego; de esta forma las actividades adquirirán o no el sentido pleno del juego en función de la actitud del jugador (libre, voluntaria, placentera, sin objetivos extrínsecos, etc.) en el momento de su realización, en definitiva, en función de la forma en que el jugador viva la experiencia del juego.

Por otro lado, cualquier aproximación al concepto de juego, aunque sea breve y genérica, no puede ni debe obviar hacer mención de los potenciales que la propia actividad lúdica encierra. No se debe olvidar que muchas de las teorías existentes han optado por explicar el juego basándose en su funcionalidad, es decir, en lo que subyace - tanto a nivel individual como social - tras la experiencia inmediata del juego.

Haywood et al. (1993) y Franch y Martinell (1994) ofrecen una buena síntesis de las potencialidades que subyacen en el juego. Éstas afectan a cuatro dimensiones fundamentales del individuo: la dimensión motora, la intelectual, la afectiva y la social.

- 1.) El juego en el desarrollo motor.** El juego como acción o actividad contribuye directamente al desarrollo del individuo en todos sus sentidos. En el desarrollo motor, el juego implica a menudo movimiento y ejercicio físico, por lo que estimula, entre otras cosas, la precisión gestual, la coordinación de movimientos o el aumento de la fuerza y de la velocidad.
- 2.) El juego en el desarrollo intelectual.** El juego, además de movimiento, implica también comprender el funcionamiento de las cosas, solucionar situaciones, elaborar estrategias de actuación, etc. El juego, por tanto, también cumple una función esencial en el campo intelectual. Función que ha sido ampliamente aprovechada por el sector educativo que, percatándose de sus importantes beneficios, ha hecho uso del juego como medio didáctico.
- 3.) El juego en el desarrollo afectivo.** se dice que el juego es una actividad libre y placentera, con un importante poder de evasión temporal. Esta ficción característica, esta posibilidad de actuar “como si”, otorga al juego un papel fundamental en el desarrollo afectivo de la persona. El juego en su propia realidad imaginaria y a través de la repetición de situaciones, estimula la comprensión y maduración de experiencias de la vida, implica confianza e incide en otros ámbitos del desarrollo afectivo.
- 4.) El juego en el desarrollo social.** El juego es entendido también como una forma de relación. Se puede matizar las palabras de Caillois y otros autores cuando afirman que el juego es una actividad improductiva - puesto que no produce ningún tipo de bienes materiales - y decir que el juego contiene una elevada productividad relacional. Esta estrecha

asociación entre el juego y la relación social queda patente, por ejemplo, desde el momento en que se describe el juego como una actividad reglamentada, consensuada por los propios participantes que son quienes establecen lo que es o no posible, lo que cada uno debe hacer, etc.

El juego, además, desde su dimensión socializadora, implica también un aprendizaje de la vida social. Su capacidad de generación simbólica de roles lo convierte en un eficaz agente de transmisión de los valores y actitudes dominantes en nuestra sociedad; lo convierte, en definitiva, en una actividad de carácter marcadamente cultural. En otras palabras “los juegos son situaciones inventadas que permiten la participación de mucha gente en algún patrón significativo de su propia vida corporativa” (McLuhan, 1996).

No obstante lo anterior, por la propia naturaleza del presente trabajo de tesis, se utilizará el concepto de juego prestando más atención a sus cualidades intrínsecas que a las funciones extrínsecas que se acaban de exponer.

2.1.2 Los Videojuegos como Fenómeno Social

Los videojuegos constituyen un fenómeno popular que se inserta en el proceso de desarrollo tecnológico que experimenta nuestra sociedad. Como tales los videojuegos se introdujeron por primera vez en los Estados Unidos a principios de los años setenta con un éxito sin precedentes en los salones recreativos hasta entonces ocupados por máquinas tragamonedas y *pinballs* (máquinas de juego conocidas en Ecuador como *cosmos* y en España como *flipers*).

La primera producción de videojuegos surgió a principios de la década de los setenta de las manos de Nolan Bushnell, creador de la empresa Atari. Conocido como *Pong-Pong*, se trataba de un juego sencillo de tenis de mesa, compuesto por dos barras que simulaban las raquetas y un cursor que, moviéndose, atravesaba la pantalla. Pudo jugarse inicialmente en una máquina que funcionaba con monedas de la que en menos de un año, se vendieron aproximadamente 6.000 unidades en EEUU. De forma paralela a la aparición de *Pong-Pong*, la compañía Magnavox comercializó un videojuego conocido como *Odyssey* que en lugar de jugarse mediante máquinas de funcionamiento con monedas, podía utilizarse a través de las televisiones domésticas (por medio de una unidad de control acoplada al aparato de televisión, que permitía jugar insertando una tarjeta de programación).

En poco tiempo, tras la aparición de los primeros videojuegos señalados, se introdujeron mejoras considerables en los mismos y así en 1975, Atari entró en el mercado del vídeo doméstico con una versión de *Pong* que ofrecía múltiples novedades: efectos sonoros para cada error, logro o rebote, un marcador digital automático en la pantalla, etc. A partir de 1976, unas veinte compañías diferentes empezaron a dedicarse a la producción de videojuegos domésticos (Jones, 1984). A los primeros juegos (*Pong* y *Odyssey*) les siguieron otros como *Space Invaders* creado en 1979, probablemente el videojuego que ha conocido más versiones y adaptaciones, desde los modelos para ordenadores personales hasta los pensados para todo tipo de consolas (Estallo, 1995). Más tarde llegaron *Missile Comand*, *Asteroids* y *PacMan* (Egli, 1984;

Meyers, 1984; Crespi, 1983 y Goldstein, 1993), popularizados en Ecuador como “rompe-mentes” y en España como “come-cocos”.

La popularidad del término “videojuego” y el consecuente uso del mismo contrastan con la falta de especificidad cuando se realiza una revisión de la literatura existente. Si bien se hallan numerosas investigaciones sobre distintos aspectos, pocos son los autores que realizan una clarificación previa sobre el término, probablemente porque, en la mayoría de los casos, su significado se da por sobre entendido. Sin embargo, llegar a un consenso sobre este significado no es una tarea fácil. En primer lugar, porque a menudo el concepto de videojuego es utilizado de forma indistinta para hacer referencia tanto a su componente tecnológico como al tipo de juego. En otras palabras, el videojuego designa tanto al *hardware* como al *software*⁵. Junto a ello y en segundo lugar, porque la proliferación de innovaciones tecnológicas ha diversificado mucho el mercado y se tiene una multitud de posibilidades como consolas domésticas, máquinas recreativas, computadoras personales, etc., que presentan sus propias ofertas diversificadas en el mercado de los videojuegos. En definitiva, se trata de uno de los productos más notorios de la tecnología avanzada de la micro-computadora (Lin y Lepper, 1987). Así, no es difícil imaginar que la mayoría de las investigaciones que se sitúan en torno a la década de los ochenta, cuando utilizan el concepto de videojuego, le atribuyen un significado,

⁵Este doble significado del término no aparece tan acentuado cuando el *hardware* utilizado es la computadora personal, pero alcanza un elevado grado de confusión en el caso de las consolas domésticas y portátiles. Éstas a menudo son denominadas también videojuegos. Probablemente esto se deba a que, en el caso de la computadora personal, existen otras múltiples funciones además de la de servir de soporte técnico para jugar con videojuegos, cosa que no ocurre con las consolas, específicamente diseñadas para jugar con videojuegos.

ya sea referido al *hardware* o al *software*, muy distinto al que puedan atribuirle aquellas investigaciones que se han desarrollado a lo largo de los últimos años.

Los videojuegos constituyen una de las actividades de entretenimiento más populares de nuestros días. Además, su campo de actuación, desde la segunda mitad de la década de los ochenta, se ha ampliado y ha sobrepasado la frontera del entretenimiento abriendo posibilidades de uso en el ámbito educativo.

Hoy en día, limitar el concepto de videojuego a una actividad exclusivamente lúdica supone obviar las potencialidades instructivas o educativas del videojuego, estudiadas a partir de numerosas investigaciones. Por otra parte implica dejar a un lado todo un conjunto de videojuegos de gran componente didáctico que, partiendo de la combinación de la función lúdica y la pedagógica, cuentan con una gran difusión en el mercado actual de videojuegos (Estallo, 1995).

Sin embargo, remitiéndose a lo comentado sobre las características propias del juego, no se debe olvidar que estos videojuegos didácticos pueden llegar a perder el sentido propio del juego desde el momento en que quien los utilice lo haga con el objetivo de aprender y no por el simple hecho de jugar. La presente investigación hará referencia a los videojuegos diseñados para ser utilizados como medio didáctico, dejando a un lado a aquellos otros específicamente entendidos como forma de juego y actividad de entretenimiento.

Los videojuegos, como indica Provenzo (1991), son algo más que un producto informático. También son un negocio, para quienes los manufacturan y los venden, y una empresa comercial sujeta como todas, a las fluctuaciones del mercado.

De igual forma pueden definirse como un instrumento de información, que cumple importantes funciones hegemónicas en la perpetuación de muchos estereotipos o como un campo de investigación en el que el investigador puede plantear cuestiones relacionadas con el sentido de nuestra propia cultura. Esta definición ampliada de los videojuegos, a partir de la suma de los distintos elementos que rodean a este concepto, confirma el hecho de que el videojuego puede ser entendido como un fenómeno social.

A efectos operativos, en función de los objetivos del presente trabajo y teniendo en cuenta las dificultades que conlleva hallar una definición comprensiva e inequívoca del término, se entiende por videojuego todo juego electrónico con objetivos esencialmente lúdicos que, sirviéndose de la tecnología informática, funcionan en distintos soportes (fundamentalmente consolas y computadoras personales). Se hará uso del término para designar todo el *software* específicamente diseñado para jugar, independientemente de que el soporte (o *hardware*) que se utilice sean máquinas de los salones recreativos, consolas portátiles, consolas domésticas, computadoras personales, laptops o dispositivos móviles.

Con la intención de hacer aún más operativa la manera de entender este concepto, a lo largo del trabajo se recurrirá al vocablo videojuego para denominar tanto a los juegos exclusivamente lúdicos, como a aquéllos que

pueden presentar amplias potencialidades didácticas o educativas, siempre y cuando sigan manteniendo su componente lúdico como aspecto esencial (por ejemplo, la mayoría de juegos de simulación).

De esta forma, se incluye aquellos programas educativos que, cuentan con una presentación dinámica, motivadora y atractiva para sus usuarios (sonidos digitalizados, efectos visuales y sonoros especiales, elevados grados de interacción, etc.) priorizan sus objetivos didácticos, considerados lúdicos, por su rol de importancia.

2.1.3 Tipos de Videojuegos y Propuestas de Clasificación

Una consecuencia más de la diversidad que caracteriza el fenómeno de los videojuegos es la dificultad para llevar a cabo cualquier intento de clasificación. De hecho los criterios que se siguen son diversos: según el *hardware* que utilizan, según los contenidos del juego, según los destinatarios a los que van dirigidos, etc.

Inicialmente pueden establecerse dos grandes tipologías de clasificación: la que parte del tipo de *hardware* que el videojuego necesita, y la que parte de las características del propio juego (temática, tipo de juego, destrezas que implica, etc.).

La primera tipología no presenta ningún problema a la hora de crear una clasificación. Únicamente la gran rapidez con la que varían los productos hace que cualquier intento de clasificación pueda quedar desfasado en un margen de

tiempo relativamente pequeño y que la necesidad de actualización sea constante.

Nawrocki y Winner (1983) ya señalaron que los videojuegos, en función del *hardware* a través del cual son distribuidos o presentados, pueden clasificarse en cuatro tipos: máquinas de salones recreativos, computadoras personales, consolas domésticas y consolas portátiles.

Esta clasificación puede mantenerse en el momento actual, si bien se da una diversificación de tipologías importantes, sobre todo en el caso de las consolas domésticas y de las computadoras personales. Sin olvidar los comienzos de la introducción de nuevas tecnologías como la realidad virtual y las potencialidades de las denominadas “autopistas de la información”, las cuatro categorías citadas sirven para clarificar los videojuegos en función del *hardware* de que hacen uso.

Dentro de esta misma tipología, en un segundo nivel de clasificación, a la hora de diferenciar clases de consolas, el criterio utilizado es la capacidad de procesamiento de la información que tiene la máquina, es decir, el número de *bits* de la consola. Así, después del surgimiento de un primer tipo de consolas conocidas popularmente como consolas portátiles, los diseñadores de consolas han ido ampliando el potencial y las posibilidades de estas máquinas, desde los 8 *bits* hasta las plataformas multimedia que es como se conocen las consolas de 32 *bits*, culminando con la reciente creación de consolas de 64 *bits*. En este desarrollo progresivo, a partir de la aparición de las máquinas de 32 *bits* se empezó a hablar de consolas de “nueva generación”. La realidad es que el

mercado ha evolucionado de forma muy rápida y las consolas actuales poco tienen que ver con las primeras máquinas aparecidas en los ochenta.

A la hora de clasificar los videojuegos a partir de los tipos de juegos que pueden hallarse, aparecen infinidad de categorizaciones. A pesar de que el criterio central es el tipo de juego, las categorías que se establecen parten de aspectos tan diversos como la temática, el grado de interacción, las habilidades y destrezas necesarias para su uso, etc. A continuación se revisará algunas de las propuestas más interesantes.

Martín y colaboradores, en su trabajo sobre actividades lúdicas para jóvenes, dedican un apartado específico a los videojuegos, centrándose fundamentalmente en los juegos de ordenador. Estos autores, a partir de las características generales del desarrollo del juego, clasifican los videojuegos en siete tipos distintos (Martín et al., 1995):

- **Arcade.** Son los juegos de ordenador más tradicionales. En ellos el jugador a través de un personaje debe superar una serie de obstáculos de creciente dificultad, matar a los enemigos que le atacarán y coger una serie de objetos que le serán útiles en el transcurso del juego. Dentro de esta categoría, los autores citan otras variantes que no cuentan con un componente violento tan manifiesto. Es el caso de los simuladores deportivos y, especialmente, los juegos de lucha o los juegos de construcción, en los que se deben ir encajando distintas piezas para ir formando figuras determinadas a gran velocidad y con una dificultad que aumenta progresivamente.

- **Aventura.** Parten de la idea de conseguir un objetivo determinado en un ambiente de aventura y peligro en el que el jugador deberá superar dificultades, resolver problemas o enigmas, o derrotar a sus enemigos.
- **Estrategia.** En este tipo de videojuegos se suele reproducir una situación compleja en la que el jugador debe controlar una serie de variables para lograr una meta concreta.
- **Juegos de rol.** Son una simulación de los juegos de mesa que llevan el mismo nombre, donde el ordenador juega el papel de director del juego y contiene las reglas del mismo.
- **Simuladores.** Reproducciones muy sofisticadas de aparatos o actividades complejas como, por ejemplo, los simuladores de vuelo, de conducción de vehículos o de realización de deportes concretos.
- **Educativos.** Juegos en los que prima una finalidad más educativa que de puro entretenimiento.
- **Juegos de mesa.** Reproducciones de gran parte de los juegos de mesa tradicionales.

Una de las clasificaciones más completa es la desarrollada por Estallo (1995) que combina dos criterios distintos: por un lado las habilidades y recursos psicológicos necesarios para el juego y, por otro, el desarrollo y temática del juego en sí. De esta forma, se establecen cuatro grandes divisiones: juegos de arcade, juegos de simulación, juegos estratégicos y reproducciones de juegos de mesa.

A.) Juegos de arcade: aquéllos que requieren un ritmo rápido de juego, exigiendo tiempos de reacción mínimos y una atención focalizada; apenas cuentan con un componente estratégico.

- **Juegos de plataforma:** en los que el protagonista se halla en un escenario bidimensional desplazándose de izquierda a derecha y de arriba a abajo.
- **Laberintos:** cuyo eje central es el escenario que reproduce un laberinto de considerable extensión.
- **Competiciones deportivas:** que reproducen deportes distintos.
- **Juegos de acción:** de trepidante desarrollo lineal, donde los escenarios varían al eliminar el número de enemigos suficiente.

B.) Juegos de simulación: simulan actividades o experiencias raramente accesibles en la vida real. Permiten al jugador asumir el mando de situaciones o tecnologías específicas. Entre sus características principales destacan: la baja influencia del tiempo de reacción en comparación con los juegos de arcade; la utilización de estrategias complejas y la necesidad de enfrentarse a situaciones nuevas que exigen conocimientos específicos sobre la simulación (por lo que a menudo los juegos se acompañan de manuales de uso).

- **Simuladores instrumentales:** que fueron los primeros en comercializarse y tienen su origen en los simuladores de vuelo utilizados en el entrenamiento de pilotos aéreos.
- **Simuladores situacionales:** en los que, a diferencia de los instrumentales, el jugador asume un papel específico determinado por el tipo de simulación:
 - Los llamados “simuladores de dios”, de tres tipos fundamentales: bio-ecológicos, socioeconómicos y mitológicos. Los primeros suelen basarse en la simulación del desarrollo de la vida; los socioeconómicos se centran en la simulación de situaciones en las que el tema argumental es el económico y los terceros invitan al jugador a asumir el papel de una divinidad que ejerce su poder a expensas de otros dioses.
 - Simuladores deportivos, como los de golf o ajedrez, en los que el jugador asume un papel de entrenador, seleccionando jugadores o planificando una estrategia, más que desempeñando el papel de simple practicante del deporte⁶.

⁶Estallo mantiene que la diferencia entre los simuladores deportivos y los juegos deportivos clasificados como juegos de arcade se halla en la complejidad del juego y en el papel que juega la acción en el mismo. Según este autor, en el caso de los simuladores deportivos se trata de juegos de gran realismo y complejidad mientras que en el caso de los videojuegos deportivos de arcade los juegos son menos complejos y su grado de realismo es inferior.

C.) Juegos estratégicos: el jugador adopta un papel específico y sólo conoce el objetivo final. Con frecuencia los personajes son de ficción y provienen del mundo de la literatura y del cine:

- Aventuras gráficas, donde la acción se desarrolla a través de las órdenes del jugador, utilizando además una serie de objetos que van apareciendo en los distintos escenarios.
- Juegos de rol, donde el jugador puede controlar a más de un protagonista de características definidas por el propio usuario.
- Juegos de estrategia militar o *war-games*, similares a los juegos de mesa de estrategia militar.

D.) Reproducciones juegos de mesa: uno o varios jugadores se colocan alrededor de una mesa virtual para jugar con el mismo desarrollo que sus originales. Aunque el azar puede ser una parte muy importante en este tipo de juegos, también los hay en los que son necesarios estrategia y razonamiento para poder jugar y en los que el azar no aparece.

Finalmente otra clasificación bastante interesante es la desarrollada por el Dr. Pere Marquès Graells (2001) que analiza los diferentes tipos de videojuegos existentes, desde un punto de vista psicopedagógico. En esta clasificación se establecen siete categorías de videojuegos en función del tipo de juego.

A. Juegos de arcade

1. Definición: Son aquellos juegos relativamente fáciles de jugar o que no responden completamente a fuerzas físicas reales como la gravedad y la aceleración. También se refiere a aquellos juegos cuyos gráficos son poco complicados de representar y la dificultad aumenta progresivamente con cada escena superada, pero donde no hay profundidad en cuanto a historia, mitología, personalidad de los personajes, etc.

2. Consideraciones psicopedagógicas:

- Pueden contribuir al desarrollo psicomotor y de la orientación espacial de los estudiantes, aspecto especialmente útil en el caso de los más pequeños.
- Riesgos a considerar: nerviosismo, estrés y hasta angustia que pueden manifestar algunos alumnos ante las dificultades que encuentran para controlar a los personajes del juego.
- Conviene limitar el tiempo que se dedique a esta actividad y observar los comportamientos de los pequeños para ayudarles y detectar posibles síntomas de estar sometidos a una tensión excesiva.

3. Ejemplos: Pacman, Mario Bros, Sonic, Doom, Quake, Street Fighter, Arkanoid.

B. Juegos de deportes

1. Definición: Son aquellos juegos basados en cualquier clase de deporte como fútbol, baloncesto, béisbol, automovilismo, boxeo, tenis, etc. En general, los enfrentamientos se llevan a cabo siguiendo las reglas estándares del deporte aunque se suele controlar un tiempo corto de disputa definido por lo regular entre 10 y 15 minutos. Pasado este tiempo si el jugador lleva la ventaja podrá seguir jugando, en caso contrario, si el jugador está empatado con la máquina o perdiendo, se considera un juego terminado (Game Over). Por lo general no ofrecen una historia o premisa de juego.

2. Consideraciones psicopedagógicas:

- Permiten la ejercitación de diversas habilidades de coordinación psicomotora y profundizar en el conocimiento de las reglas y estrategias de los deportes.
- En algunos casos también se pueden alcanzar niveles altos de estrés.

3. Ejemplos: FIFA, PC Futbol, NBA, Formula I GrandPrix, Need For Speed.

C. Juegos de aventura y de rol

1. Definición: Un juego de rol (traducción típica en castellano del inglés role-playing game, literalmente “juego de interpretación de papeles”)

es un juego en el que, tal como indica su nombre, uno o más jugadores desempeñan un determinado rol o papel.

2. Consideraciones psicopedagógicas:

- Pueden proporcionar información y constituir una fuente de motivación hacia determinadas temáticas que luego se estudiarán de manera más sistemática en clase.
- Una de las preocupaciones de los educadores deberá ser promover la reflexión sobre los valores y contravalores que se consideran en el juego.

3. Ejemplos: King Quest, Indiana Jones, Monkey Island, Final Fantasy, Tomb Raider, Pokémon, Ultima Online, World of Warcraft.

D. Simuladores y constructores

1. Definición: Este tipo de videojuegos intentan recrear situaciones de la vida real. Los videojuegos de simulación reproducen sensaciones que en realidad no están sucediendo. Pretenden reproducir tanto las sensaciones físicas como velocidad, aceleración, percepción del entorno y una de sus funciones es dar una experiencia real de algo que no está sucediendo para de esta forma no poner en riesgo la vida de alguien.

2. Consideraciones psicopedagógicas:

- Permiten experimentar e investigar el funcionamiento de máquinas, fenómenos y situaciones.
- Además de controlar posibles estados de tensión excesiva en algunos alumnos, conviene advertir a los estudiantes que están ante un

modelo (representación simplificada de la realidad - a veces presentan una realidad imaginaria-) y que por lo tanto en el mejor de los casos sólo constituyen una aproximación a los fenómenos que se dan en el mundo físico.

- La realidad siempre es mucho más compleja que las representaciones de los mejores simuladores.
- 3. Ejemplos:** Simulador de vuelo Microsoft, Sim City, Tamagotchi, The Incredible Machine, Theme Park.

E. Juegos de estrategia

- 1. Definición:** Este tipo de videojuegos requieren que el jugador ponga en práctica sus habilidades de planeamiento y pensamiento para conseguir la victoria. En la mayoría de los videojuegos de estrategia, al jugador se le concede una vista del mundo absoluta, controlando indirectamente las unidades bajo su poder.
- 2. Consideraciones psicopedagógicas:**
 - Exigen administrar unos recursos escasos (tiempo, dinero, vidas, armas...) prever los comportamientos de los rivales y trazar estrategias de actuación para lograr unos objetivos.
 - Quizá los mayores peligros de estos juegos sean de carácter moral, por los contravalores que muchas veces asumen y promueven. Resulta conveniente organizar actividades participativas que permitan analizar y comentar estos aspectos con los jugadores.

- 3. Ejemplos:** Estratego, Warcraft, Age of Empires, Civiltation, Lemmings, Black & White, Centurion.

F. Puzzles y juegos de lógica

- 1. Definición:** Este tipo de videojuegos también conocidos como videojuegos de inteligencia, son un género de videojuegos que se caracterizan por exigir agilidad mental al jugador. Pueden involucrar problemas de lógica, estrategia, reconocimiento de patrones, completar palabras o hasta simple azar. El género puede ser difícil de describir ya que cada uno tiene su estilo único.

- 2. Consideraciones psicopedagógicas:**

- Desarrollan la percepción espacial, la lógica, la imaginación y la creatividad.
- No se contempla riesgos específicos para este tipo de juegos, aunque como pasa con todos los videojuegos conviene evitar una excesiva adicción que podría conducir a un cierto aislamiento y falta de ejercicio físico.

- 3. Ejemplos:** 7th.Guest, Tetris.

G. Juegos de preguntas

- 1. Definición:** Este tipo de videojuegos permiten que los jugadores se sometan a pruebas de conocimiento en diferentes áreas del saber. Este tipo de videojuegos pueden ser desarrollados con técnicas de inteligencia artificial e incluso caer en el área de los sistemas expertos.

2. Consideraciones psicopedagógicas:

- Los juegos de preguntas pueden servir para repasar determinados conocimientos de todo tipo.

3. Ejemplos: Trivial, Carmen Sandiego.

2.1.4 La Polémica: Partidarios y Detractores

Cualquier aproximación al ámbito de los videojuegos desemboca, casi inevitablemente, en la controversia entre quienes defienden sus beneficios potenciales y los que critican sus presuntos efectos negativos.

Dicha controversia se desenvuelve en un campo de argumentaciones en el que se van combinando las opiniones sociales (de presencia predominante en el debate histórico) con los resultados de la investigación empírica que van surgiendo. Esta combinación es inevitable puesto que se habla de un fenómeno fundamentalmente social, que se presta a la emisión de opiniones o juicios basados en la experiencia personal - y, por tanto, limitados - difícilmente generalizables. En cualquier caso, todavía, aquéllos que defienden los videojuegos no lo hacen desde los beneficios probados sino desde los beneficios hipotéticos. Del mismo modo, aquéllos que los critican, en muchos casos, tampoco se basan en datos objetivos. No es extraño hallar noticias en prensa que alarman sobre los efectos casi devastadores de los videojuegos: causan adicción, fomentan la agresividad, interfieren en la realización de otras actividades, favorecen el sexismo, etc. Sin embargo, a menudo los resultados de la investigación sobre videojuegos no acaban de confirmar esas acusaciones. Desafortunadamente, ya lo comentaron en su momento Dorva y Pépin (1986), la polémica que rodea a los efectos positivos o negativos de los

videojuegos sobre los usuarios se ha basado más en opiniones y especulaciones que en hallazgos empíricos.

Más allá de este contraste entre la opinión pública y los resultados derivados de las investigaciones, centrándonos en la propia investigación, los hallazgos que parecen haberse encontrado han alimentado la controversia entre los “partidarios” y los “detractores”; esta discrepancia es un hecho constatado por numerosos estudiosos del tema. El estudio de Lin y Lepper, uno de los más citados en la literatura existente sobre videojuegos, en el que se analizan las correlaciones del uso de videojuegos y ordenadores por parte de los niños, recoge ampliamente el desarrollo de esta controversia.

Estos autores (Lin y Lepper, 1987) comentan que los partidarios de este tipo de juegos los defienden como fuente de aprendizaje, además de verlos como una diversión, ya que promueven la coordinación óculo-manual y enseñan habilidades específicas de visualización espacial o incluso de matemáticas. Sus usuarios adquieren también estrategias generalizadas para “aprender a aprender” en entornos nuevos; los videojuegos pueden llegar a reforzar la autoestima entre jugadores con dificultades de adaptación y, además, pueden servir como iniciación en el mundo de la informática. Los críticos, según Lin y Lepper, señalan distintas consecuencias negativas: los videojuegos pueden desplazar el tiempo dedicado al estudio o a la realización de otras actividades de ocio y pueden apoyar una tendencia a la marginación y promover la alienación entre adolescentes socialmente marginados; además, su contenido violento y frenético puede promover conductas impulsivas y agresivas entre sus usuarios, de tal forma que, en los casos más extremos, los

videojuegos pueden llegar a ser considerados como agentes del fracaso escolar y fomentadores de la delincuencia. Más anecdóticamente se les ha acusado de propiciar algunas enfermedades como la tendinitis.

Schmitt en un artículo de 1992, en el que expone algunas recomendaciones para los padres a la hora de controlar el uso de videojuegos por parte de sus hijos, hace una referencia directa a los beneficios potenciales y a los inconvenientes de los videojuegos. Esos beneficios potenciales podrían resumirse argumentando que promueven la atención hacia los detalles, la memorización de hechos, la secuenciación de la información y el desarrollo de estrategias creativas, que promueven la coordinación óculo-manual y que mejoran la percepción visual. Por otra parte, según el mismo autor, los videojuegos pueden dominar el ocio y el tiempo de estudio del individuo, impidiendo otras actividades como practicar deportes o leer y llegando a afectar al rendimiento académico; también pueden reducir las interacciones sociales con la familia y los amigos, y pueden fomentar (en el caso de determinados juegos) la aceptación de conductas violentas en la vida real.

Funk (1993) vuelve a señalar la existencia de partidarios y detractores del fenómeno, y señala que los argumentos de unos y otros se repiten. Entre los argumentos de los detractores añade la observación de que la posibilidad de promoción de una conducta antisocial aumente en los salones recreativos, y que la creatividad pueda deteriorarse porque los jugadores siempre siguen unas normas determinadas para ganar; reitera que esos detractores mantienen que los contenidos violentos influirán decisivamente sobre los valores morales y éticos y fomentarán una conducta agresiva. Por lo que hace referencia a los

partidarios de los videojuegos, Funk señala que éstos ven esta actividad de ocio como un escape benigno que puede promover el sentido del dominio y la aceptación social, así como desarrollar destrezas específicas como la coordinación óculo-manual.

No es difícil observar, por tanto, que los argumentos de partidarios y detractores, descritos por distintos autores, se repiten de forma constante. A continuación en la Tabla 2 se presenta un resumen de las alegaciones a favor y en contra de los videojuegos.

Tabla 2. Alegaciones a favor y en contra de los videojuegos.

A FAVOR	EN CONTRA
Entretienen	Provocan adicción
Ejercitan la coordinación óculo-manual	Promueven conductas violentas
Estimulan la capacidad de lógica y reflexión	Aíslan socialmente
Ayudan a concentrar la atención	Limitan la imaginación
Son una introducción a la informática	Restan tiempo a otras actividades
Son un potencial muy adecuado para distintas aplicaciones sociales	Afectan de manera negativa al rendimiento académico

Al margen de la cantidad y variedad de argumentos que se puedan encontrar, lo que parece evidente es que los videojuegos han creado - y siguen creando – una fuerte polémica social que, como se ve a continuación, se ha trasladado también al terreno de la investigación.

2.2 Ingeniería de Software

2.2.1 Definición

De acuerdo a Roger Pressman (2006), la ingeniería de software es una disciplina que integra al proceso, los métodos y las herramientas para la producción de software la cual debe estar sustentada por la gestión de la calidad. (Ver Figura 1).

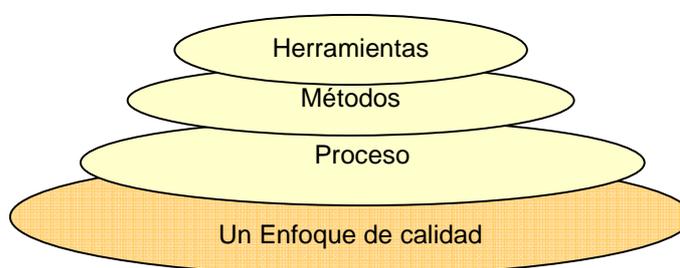


Figura 1. Estratos de la Ingeniería de Software.

El **enfoque de calidad** ayuda a fomentar una cultura de mejora continua del proceso, produciendo enfoques muy efectivos en la Ingeniería de Software.

El **proceso** es el elemento que mantiene juntos a los estratos de la tecnología y permite el desarrollo racional y a tiempo del software, es la base para el control de la gestión de los proyectos de software, establece el marco de trabajo para aplicar los métodos técnicos, se establecen los fundamentos y se asegura la calidad y el cambio se maneja de forma apropiada.

Los **métodos** proporcionan la forma técnica para construir software, los métodos son un conjunto de tareas que incluyen la comunicación, análisis de

requerimientos, modelado de diseño, construcción del programa, pruebas y soporte.

Las **herramientas** proporcionan el soporte automatizado para el proceso y los métodos.

2.2.2 Marco de Trabajo para el Proceso

Un marco de trabajo establece la base para un proceso de software completo al identificar un grupo de actividades aplicables a todos los proyectos de software, sin importar su tamaño o complejidad.

El siguiente marco de trabajo genérico del proceso se puede aplicar en la inmensa mayoría de los proyectos de software como lo menciona Pressman R. en su libro de Ingeniería de Software (Sexta Edición, 2006):

- **Comunicación.** Esta actividad del marco de trabajo implica una intensa colaboración y comunicación con los clientes; además, abarca la investigación de requisitos y otras actividades relacionadas.
- **Planeación.** Esta actividad establece un plan para el trabajo de la ingeniería del software. Describe las tareas técnicas que deben realizarse, los riesgos probables, los recursos que serán requeridos, los productos del trabajo que han de producirse y un programa de trabajo.
- **Modelado.** Esta actividad abarca la creación de modelos que permiten al desarrollador y al cliente entender mejor los requisitos del software y el diseño que logrará satisfacerlos. Esta actividad la componen dos acciones de la ingeniería del software: análisis y diseño. El análisis

abarca un conjunto de tareas de trabajo, como por ejemplo: la investigación, elaboración, negociación, especificación y validación de requisitos que conducen a la creación del modelo de análisis o la especificación de requisitos. El diseño abarca tareas de trabajo como el diseño de datos, diseño arquitectónico, diseño de la interfaz y el diseño a nivel de componentes que crean un modelo de diseño o una especificación de diseño.

- **Construcción.** Esta actividad combina la generación del código y la realización de pruebas necesarias para cubrir errores en el código.
- **Despliegue.** El software entendido como una entidad completa o un incremento completado de manera parcial, se entrega al cliente, quien evalúa el producto recibido y proporciona información basada en su evaluación.

Estas cinco actividades genéricas del marco de trabajo son útiles durante el desarrollo de programas pequeños y medianos, la creación de grandes aplicaciones en la red, y en la ingeniería de sistemas basados en computadoras grandes y complejas. Los detalles del proceso del software serán muy diferentes en cada caso, pero las actividades dentro del marco permanecerán iguales. Distintos proyectos demandan diferentes conjuntos de tareas. El equipo de software elige el conjunto de tareas con base en el problema y en las características del proyecto.

2.2.3. Inteligencia Artificial – IA

La Inteligencia Artificial (IA) estudia cómo lograr que las máquinas realicen tareas que, por el momento, son realizadas mejor por los seres humanos (Rich y Knight, 1994). La Inteligencia Artificial (IA) también se define como una disciplina que se ocupa de la creación de programas de computadora que simulan la actuación y el pensamiento humano, así como el comportamiento racional (Russel, 1995). De acuerdo a estos conceptos se puede ver que la inteligencia artificial (IA) o artificial intelligence (AI), son un conjunto de técnicas y algoritmos, que permiten simular el pensamiento humano a través de una máquina o computadora. Dentro de las técnicas más importantes que maneja la inteligencia artificial están las siguientes:

- A. **Reglas de inferencia:** Se puede representar el conocimiento mediante reglas de inferencia. Estos esquemas establecen relaciones sintácticas entre un conjunto de fórmulas llamados premisas y una asección llamada conclusión.

- B. **Técnicas de razonamiento hacia adelante y hacia atrás:** Este tipo de técnicas permiten partir de los estados iniciales de un problema y llegar a los estados objetivos y luego partir de los estados objetivos para llegar a los estados iniciales y resolver un determinado problema.

- C. **Técnicas de ensayo y error:** La técnica de ensayo y error, también conocida como prueba y error, es un método heurístico para la obtención

de conocimiento, tanto proposicional como procedural. Consiste en probar una alternativa y verificar si funciona. Si es así, se tiene una solución. En caso contrario que es el resultado erróneo, se intenta una alternativa.

D. Técnicas heurísticas: Este tipo de técnica se basa en algoritmos que pueden producir una buena solución (posiblemente la óptima), pero también pueden que produzcan ninguna solución o dar una solución no muy buena. Normalmente se basa, en un conocimiento intuitivo del programador sobre un determinado problema.

E. Lógica Difusa: La lógica difusa (fuzzy logic) es una extensión de la lógica tradicional (Booleana) que utiliza conceptos de pertenencia de conjuntos (sets) más parecidos a la manera de pensar humana. El concepto de un subconjunto difuso (subset fuzzy) fue introducido por L.A. Zadeh en 1965 como una generalización de un subconjunto exacto (crisp subset) tradicional. Los subconjunto exactos usan lógica Booleana con valores exactos como por ejemplo la lógica binaria que usa valores de 1 o 0 para sus operaciones. La lógica difusa no usa valores exactos como 1 o 0 pero usa valores entre 1 y 0 (inclusive) que pueden indican valores intermedios (Ej. 0, 0.1, 0.2, ... ,0.9,1.0, 1.1, ... , etc.). La lógica difusa también incluye los valores 0 y 1 entonces se puede considerar como un super conjunto (superset) o extensión de la lógica exacta.

- F. **Redes neuronales:** Las redes de neuronas artificiales, denominadas habitualmente como RNA (red neuronal artificial) o en inglés como: ANN (artificial neural network) son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas que colaboran entre sí para producir un estímulo de salida. En inteligencia artificial es frecuente referirse a ellas como redes de neuronas o redes neuronales.
- G. **Redes bayesianas:** Una red Bayesiana, red de Bayes, red de creencia, modelo Bayesiano (de Bayes) o modelo probabilístico en un gráfico acíclico dirigido es un modelo gráfico probabilístico (un tipo de modelo estático) que representa un conjunto de variables aleatorias y sus dependencias condicionales a través de un gráfico acíclico dirigido (DAG por sus siglas en inglés). Por ejemplo, una red Bayesiana puede representar las relaciones probabilísticas entre enfermedades y síntomas. Dados los síntomas, la red puede ser usada para computar las probabilidades de la presencia de varias enfermedades. Formalmente, las redes Bayesianas son grafos dirigidos acíclicos cuyos nodos representan variables aleatorias en el sentido de Bayes, las mismas pueden ser cantidades observables, variables latentes, parámetros desconocidos o hipótesis. Las aristas representan dependencias condicionales; los nodos que no se encuentran conectados representan variables las cuales son condicionalmente independientes de las otras.

- H. **Algoritmos genéticos:** Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico. En los años 1970, de la mano de John Henry Holland, surgió una de las líneas más prometedoras de la inteligencia artificial, la de los algoritmos genéticos. Son llamados así porque se inspiran en la evolución biológica y su base genético-molecular. Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados.
- I. **Sistemas de Reacción Basados en Reglas:** Un sistema reactivo debe tener algún tipo de base de conocimiento que describa las acciones que deben realizarse bajo ciertas circunstancias. En estos sistemas se elige una acción cada vez, no anticipa y selecciona una secuencia completa de acciones antes de realizar una primera acción.

2.3 Metodología de Diseño de Hipermedia Orientada a Objetos (OOHDM)

2.3.1 Historia

Con el crecimiento de Internet la creación de portales Web para ser publicados y obtener una verdadera acogida ha tomado nuevos enfoques tanto en el tamaño como en su complejidad; requiriendo de adecuados servicios y características de navegación como punto fundamental para el éxito de la aplicación final.

Por ello los investigadores, han dedicado su esfuerzo al estudio del análisis y diseño antes del desarrollo con el fin de crear metodologías prácticas y eficientes que ayuden a diseñar la hipermedia y admitan evolución y reusabilidad.

Una de las metodologías para aplicaciones en entorno Web que más se ha consolidado es OOHDM (*Object Oriented Hypermedia Design Methodology*) que fue propuesta y diseñada por D. Schwabe, G. Rossi, y S. D. J. Barbosa en el año 1996 (Universidad Nacional de la Plata) ya que se preocupa en todos sus diagramas de caracterizar el diseño de las aplicaciones hipermedia; enfocándose en un proceso de Ingeniería del Software.

Esta metodología es una extensión de **HDM**⁷ la cual fue creada por Franca Garzotto, Paolo Paolini y Daniel Schwabe en 1991, dentro del marco del proyecto HYTEA de la Comunidad Europea, cuyo objetivo era crear un modelo que fuera de utilidad para realizar el diseño de una aplicación de hipertexto.

⁷**HDM** (*Método de Diseño Hipermedia ó Hypertext Design Model*): Es una buena metodología pero no permite hacer reutilización cuando no existe manera de generalizar estructuras y se debe crear todas las entidades distintas.

El enfoque principal de HDM era crear un modelo consistente que posea casi en su totalidad las características conceptuales, visuales, navegacionales y abstractas de un sitio Web, es decir modelar la estructura del hipertexto-hipermedia⁸ antes de su desarrollo; lo cual fue adoptado también por OOHDM con inclusión de la orientación a objetos.

2.3.2 Definición

OOHDM es una metodología orientada a objetos y ampliamente aceptada para el desarrollo de aplicaciones hipermedia, y en particular de aplicaciones Web. Se basa en dos premisas fundamentales:

- La utilización de casos de uso para la fácil captura y definición de requisitos por parte de los usuarios y clientes no expertos; de tal manera que los analistas posteriormente puedan generar los user interaction diagrams (UIDs) o modelos gráficos que representan la interacción entre el usuario y el sistema, sin considerar aspectos específicos de la interfaz.
- Comenzar el diseño del sistema con el fin de que las necesidades de interacción del sitio Web sean las deseadas por los usuarios.

⁸**Hipermedia:** Toma su nombre de la suma de hipertexto y multimedia, una red hipertextual en la que se incluye no sólo texto, sino también otros medios: imágenes, audio, vídeo, etc. (multimedia).

2.3.3 Características Fundamentales

- *Apropiada para un comportamiento complejo.*

OOHDM provee las mejores herramientas para controlar el desarrollo de una aplicación que tiene un complejo comportamiento.

- *Separación del diseño con respecto al desarrollo.*

Permite que la complejidad del desarrollo de software sea menor ya que ésta ocurre a diferentes niveles: “dominios de aplicación sofisticados(financieros, médicos, geográficos, etc.); la necesidad de proveer acceso de navegación simple a grandes cantidades de datos, y por último la aparición de nuevos dispositivos para los cuales se deben construir interfaces *Web* fáciles de usar”.⁹

2.3.4 Etapas o Fases

OOHDM propone el desarrollo de aplicaciones Multimedia y Web a través de cinco etapas donde se combinan notaciones gráficas UML (Lenguaje de Modelado Unificado) con otras propias de la metodología y son:

- Definición u Obtención de Requerimientos
- Diseño o Modelo Conceptual
- Diseño Navegacional
- Diseño de Interfaces Abstractas
- Implementación.

⁹Tomado de: Darío Andrés Silva, Construyendo Aplicaciones Web con una Metodología de Diseño Orientado a Objetos, 2002, pag. 2.

En cada etapa se crean un conjunto de modelos orientados a objetos que describen un diseño particular. El éxito de esta metodología es la clara identificación de los tres diferentes niveles de diseño en forma independiente de la implementación, como se puede ver un resumen en la Tabla 3.

Tabla 3. Etapas de la Metodología OOHDM.

Etapas	Productos	Formalismos	Mecanismos	Descripción
<i>Obtención de Requerimientos</i>	Casos de Uso (actores, escenarios)	Plantillas del formato del documento, Diagramas de Interacción de Usuario (UIDs)	Técnicas de Observación, Entrevistas	Se crea un documento que describe actividades y requerimientos de los usuarios
<i>Diseño Conceptual</i>	Clases, subsistemas, relaciones, atributos	Modelos Orientados a Objetos	Clasificación, agregación, generalización y especialización	Se modela la semántica del dominio de la aplicación
<i>Diseño Navegacional</i>	Nodos, enlaces, estructuras de acceso, contextos navegacionales, transformaciones	Vistas Orientadas a Objetos, Cartas de navegación	Clasificación, agregación, generalización y especialización	Se tiene en cuenta el perfil del usuario y las tareas. Se enfatiza en los

	de navegación	orientadas a objetos, Clases de Contexto		aspectos cognitivos. Se crea la estructura de navegación de la aplicación
<i>Diseño de Interfaz Abstracta</i>	Objetos de la interfaz abstracta, respuestas a eventos externos, transformaciones de la interfaz	Vistas Abstractas de Datos (ADV), Diagramas de Configuración, Cartas de navegación de los ADVs	Mapeado entre la navegación y los objetos visibles	Se modelan los objetos visibles. Se describe la interfaz para los objetos de navegación. Se define el aspecto de los objetos de la interfaz
<i>Implementación</i>	Aplicación en funcionamiento	Los soportados por el entorno	Los que provea el entorno	Se realiza la puesta en producción del sistema

2.3.4.1. Definición de Requerimientos

Se basa en los diagramas de casos de uso¹⁰, los cuales son diseñados por escenarios con la finalidad de obtener de manera clara los requerimientos y acciones del sistema y de los usuarios.¹¹

Describe el alcance del sistema definiendo los requerimientos funcionales y los no funcionales. Primero que todo es necesaria la recopilación de requerimientos mediante entrevistas, observaciones o cualquier otra técnica que facilite el trabajo a los desarrolladores y que los usuarios o clientes se sientan cómodos. En este punto, se hace necesario identificar los actores¹² y las tareas que ellos deben realizar.

Luego, se determinan los escenarios para cada tarea y tipo de actor. Los casos de uso que surgen a partir de aquí, serán luego representados mediante los Diagramas de Interacción de Usuario (UIDs), los cuales proveen de una representación gráfica concisa de la interacción entre el usuario y el sistema durante la ejecución de alguna tarea. Con este tipo de diagramas se capturan los requisitos de la aplicación de manera independiente de la implementación.

Ésta es una de las fases más importantes, debido a que es aquí donde se realiza la recopilación de datos, para ello se debe proporcionar las respuestas a las siguientes preguntas:

¹⁰**Caso de Uso:** Descripción a detalle de las actividades y procesos necesarios para el desarrollo de un sistema o aplicación.

¹¹**Usuarios:** Es un ente humano que usa al sistema. Un mismo usuario puede actuar como instancias en varios actores diferentes, es decir, puede jugar diferentes roles.

¹²**Actores:** Son objetos que se encuentran fuera del sistema a modelar. Representan entes que tienen necesidad de intercambiar información con el sistema; pueden ser instanciados por usuarios, dispositivos u otros sistemas.

- ¿Cuáles son los tópicos principales que serán atendidos?
- ¿Cómo los tópicos están relacionados entre sí?
- ¿Qué categoría de usuarios serán atendidos?
- ¿Cuáles son las tareas principales que serán abordadas?
- ¿Qué tareas corresponden a qué categoría de usuarios?
- ¿Los recursos disponibles son competitivos con la información levantada?¹³

Con estas preguntas se puede recaudar de cierta manera las bases necesarias para la construcción de una aplicación hipermedial exitosa, sin embargo mientras mayor sea el nivel de profundidad de la recolección de datos, existirá mayor probabilidad de cubrir todos los requerimientos solicitados por los usuarios.

OOHDM propone dividir esta etapa en 5 sub-etapas:

- Identificación de Roles y Tareas.
- Especificación de escenarios.
- Especificación de casos de uso.
- Especificación de UIs.
- Validación de Casos de Uso y UIs.

¹³ Preguntas claves tomadas de: OOHDM Object Oriented Hypermedia Design Methodology, Agosto 2006.

2.3.4.1.1. Identificación de Roles y Tareas

Roles

Se determina los posibles roles que podrían tomar los potenciales usuarios de la aplicación, ya que los usuarios pueden tomar diferentes roles importantes en cada intercambio de información con el sistema.

Tareas

Para cada rol se debe identificar las tareas que deberá soportar la aplicación.

2.3.4.1.2. Especificación de Escenarios

Los escenarios son descripciones narrativas de cómo la aplicación será utilizada. En esta sub-etapa, cada usuario deberá especificar textual o verbalmente los escenarios que describen su tarea.

2.3.4.1.3. Especificación de Casos de Uso

Un caso de uso es la descripción a detalle de las actividades y procesos necesarios para utilizar la aplicación. Representan la interacción entre el usuario y el sistema, agrupando las tareas especificadas en los escenarios existentes.

Por ello, es muy importante identificar cuál es la información relevante en cada uno de los escenarios, para luego generar un caso de uso coherente.

2.3.4.1.4. Especificación de UIDs

Los UIDs permiten representar en forma gráfica, rápida y sencilla los casos de uso generados en la etapa anterior.

Para obtener un UIDs desde un caso de uso, la secuencia de información intercambiada entre el usuario y el sistema debe ser identificada y organizada en las interacciones. Identificar la información de intercambio es crucial ya que es la base para la definición de los UIDs.

2.3.4.1.5. Validación de Casos de Uso y UIDs

El desarrollador deberá interactuar con cada usuario para validar los casos de uso y UIDs obtenidos, mostrando y explicando cada uno de ellos para ver si el o los usuarios están de acuerdo. El usuario deberá interceder sólo en aquellos casos de uso y UIDs en que participa.

2.3.4.2. Diseño Conceptual

Está representado por los objetos del dominio, las relaciones y colaboraciones existentes establecidas entre ellos; es decir, está formado por clases, relaciones y subsistemas. Este diseño se define de acuerdo a los UIDs establecidos.

Las **clases** son descritas y usadas como en los modelos orientados a objetos tradicionales; conectadas por relaciones y los objetos son instancias de las clases. Sin embargo, los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real.

Se puede usar un modelo de datos semántico estructural (como el modelo de entidades y relaciones). Utiliza una notación similar a UML.

Importante:

- Las clases son usadas en el diseño navegacional para derivar nodos, y las relaciones para construir enlaces o vínculos.
- Si la aplicación crece el diseño conceptual debe estar preparado para ser extendido, tal como cualquier diseño orientado a objetos.

2.3.4.3. Diseño Navegacional

Un modelo navegacional es construido como una vista sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegacional provee de una vista subjetiva del modelo conceptual.

Su objetivo es permitir a la aplicación ejecutar todas las tareas requeridas por el usuario, es decir, unificar una serie de tareas para obtener el diseño navegacional de la aplicación.

El diseño navegacional es expresado en dos esquemas: el esquema de clases navegacionales y el esquema de contextos navegacionales. (Ver Figura 2.).

2.3.4.3.1. Esquema de Clases Navegacionales

Las posibles vistas del hiper-documento se generan a través de las llamadas clases navegacionales que son tipos predefinidos de clases

conformados por: nodos, enlaces, anclas¹⁴ y estructuras de acceso que son los índices o recorridos guiados, que representan los posibles caminos de acceso a los nodos.

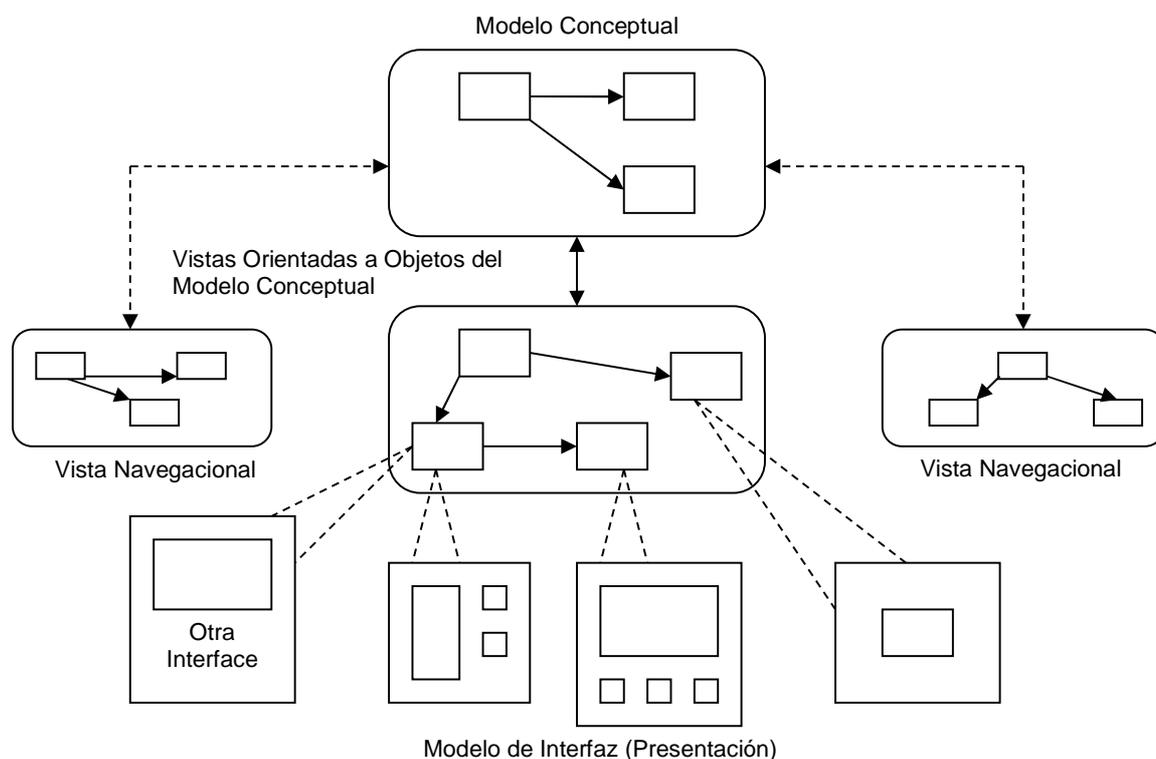


Figura 2. Esquema del Diseño Navegacional¹⁵

Nodos

- Contenedores de información básicos
- Se definen como vistas orientadas a objetos de las clases conceptuales definidas en la fase de Diseño Conceptual
- Contienen atributos simples y enlaces

¹⁴**Ancla:** Los atributos de los nodos que activan navegaciones son llamados anclas.

¹⁵ **Figura tomada del documento:** OOHDM Object Oriented Hypermedia Design Methodology, Agosto 2006.

Enlaces

- Identifican relaciones
- Implementan las relaciones definidas en el esquema conceptual
- Las clases de los enlaces especifican sus atributos, comportamiento y los objetos fuente y destino
- Representan las posibles formas de comenzar la navegación

Estructuras de Acceso

- Actúan como índices o diccionarios
- Útiles para ayudar al usuario final a encontrar la información deseada
- Ejemplos: menús, índices y tours guiados

2.3.4.3.2. Esquema de Contextos Navegacionales

Un esquema de contexto navegacional es el que organiza el espacio navegacional en conjuntos convenientes o sub-espacios que pueden ser recorridos en un orden particular y que deberían ser definidos como caminos para ayudar al usuario a lograr la tarea deseada que es obtener la información requerida, junto con los enlaces que estarán disponibles cuando se acceda a un objeto en un contexto determinado.

Contexto navegacional

- Es un conjunto de nodos, enlaces, clases de contextos y otros contextos navegacionales (contextos anidados).
- Inducido de clases navegacionales
- Se definen por extensión o enumerando sus miembros
- Un índice o un tour guiado definen contextos de navegación

2.3.4.4. Diseño de Interfaces Abstractas

En esta etapa se define la forma en la cual los objetos navegacionales pueden aparecer, es decir, especifica la estructura y el comportamiento de la interfase del sistema hipermedia con el usuario. Este modelo es abstracto y, por tanto, independiente de la implementación final del sistema.

Aquí los objetos de interfaz activarán la navegación y el resto de funcionalidades de la aplicación, esto es, se describirán los objetos de interfaz y se los asociará con objetos de navegación. La separación entre el diseño navegacional y el diseño de interfaz abstracta permitirá construir diferentes interfaces para el mismo modelo navegacional.

Este diseño se basa en tres diagramas que se complementan entre sí, y son:

- Diagrama de Vista de Datos Abstractos (ADV)
- Diagrama de Configuración
- Diagrama de Estado

2.3.4.4.1. Diagrama De Vista De Datos Abstractos (ADV)

Este diagrama incluye una vista (ADV) por cada clase navegacional (nodo, enlace o estructura de acceso) que fue establecida durante la fase de Diseño Navegacional.

Se compone de una serie de cajas¹⁶ que representan las diferentes clases de objetos que aparecerán ante el usuario.

¹⁶**Caja:** Es un ADV

2.3.4.4.2. Diagrama de Configuración

En este diagrama se representan principalmente:

- Los eventos externos provocados por el usuario, como *Clic* o *DobleClic del Ratón* que maneja un ADV.
- Los servicios que ofrece el ADV (como "visualización").
- Las relaciones estáticas entre las ADVs.

2.3.4.4.3. Diagrama de Estado

Como en el caso de los modelos OO¹⁷ de Ingeniería del Software, se representa el comportamiento dinámico del sistema hipermedial mediante el establecimiento de un diagrama de transición de estados para cada ADVs, en el que se reflejan los posibles estados por los que puede pasar cada objeto de la interface (*oculto, desactivado, ampliado, reducido, normal, etc.*) y los eventos que originan los cambios de estado.¹⁸

2.3.4.5. Implementación

En esta fase se debe implementar el diseño, ya que hasta el momento todos los modelos fueron construidos en forma independiente de la plataforma de implementación y por tanto, el desarrollador ya ha identificado la información que será mostrada, cuenta con una idea básica de cómo se verán las interfaces y tiene un completo conocimiento del dominio del problema.

¹⁷OO: Orientados a Objetos.

¹⁸Tomado de la página: *Crea tu Página Web*, wmaestro.com/webmaestro/docs/portada.html

Para comenzar con la implementación el desarrollador deberá elegir dónde almacenará los objetos y con qué lenguaje o herramienta desarrollará las interfaces; para después continuar con la creación de las mismas, de acuerdo a la secuencia de requerimientos solicitados.

2.3.5. Criterios de Selección de OOHDM

OOHDM es una metodología de diseño hipermedial, que utiliza el enfoque orientado a objetos, extendiéndolo e integrándolo con técnicas de representación gráfica de relaciones entre objetos y de contextos navegacionales que son ricos en representación estructural y semántica.

Por ello, la metodología OOHDM ha sido escogida ya que reúne las características necesarias para no mezclar aspectos conceptuales (modelo del dominio) con presentación (construcción de la interfaz de usuario) gracias a que se encuentra basada en objetos para la creación de aplicaciones Web y analiza tanto el diseño como la implementación que inevitablemente influyen en todo el proceso de desarrollo.

Además establece que es tan importante el análisis de las tecnologías que pueden limitar la funcionalidad de la aplicación, como las decisiones de diseño equivocadas que pueden reducir la capacidad de extensión y reusabilidad.

2.4. Lenguaje Unificado De Modelado (UML)

2.4.1. Definición

UML (Unified Modeling Language) o Lenguaje Unificado de Modelado es un lenguaje gráfico para visualizar, especificar, construir y documentar los elementos de un sistema, establece un conjunto de notaciones, diagramas estándar para modelar sistemas orientados a objetos describiendo la semántica esencial de estos diagramas y los símbolos en ellos utilizados.

UML representa una recopilación de mejores técnicas de ingeniería que se han probado ser exitosas en el desarrollo de sistemas grandes y complejos, es por ello que UML puede usarse con una gran variedad de metodologías o procesos de desarrollo de software, siendo un lenguaje para especificar y no un método o un proceso.

2.4.2. Diagramas

UML versión 2.0 cuenta con trece tipos de diagramas que permiten modelar el problema y su solución, en el presente proyecto solo se utilizarán cinco de los trece, que se consideran los más importantes y representativos para modelar aplicaciones de software, los mismos que son:

- Diagramas de Casos de Uso
- Diagrama de Secuencia
- Diagrama de Clase
- Diagrama de Despliegue
- Diagrama de Componente

2.4.2.1. Diagramas de Caso de Uso

Un caso de uso es una descripción de las iteraciones que se producen entre un usuario y un sistema de cómputo, especifican el comportamiento deseado del sistema por parte del usuario para llevar a cabo una tarea concreta, pero no especifica cómo lo hace. Un diagrama de caso del uso es una colección de actores¹⁹, de casos del uso y de sus relaciones.

Están basados en el lenguaje natural, es decir, es accesible por los usuarios. Los diagramas del caso del uso están conectados de cerca con escenarios. Un escenario es un ejemplo de qué sucede cuando alguien obra recíprocamente con el sistema. (Ver Figura 3).

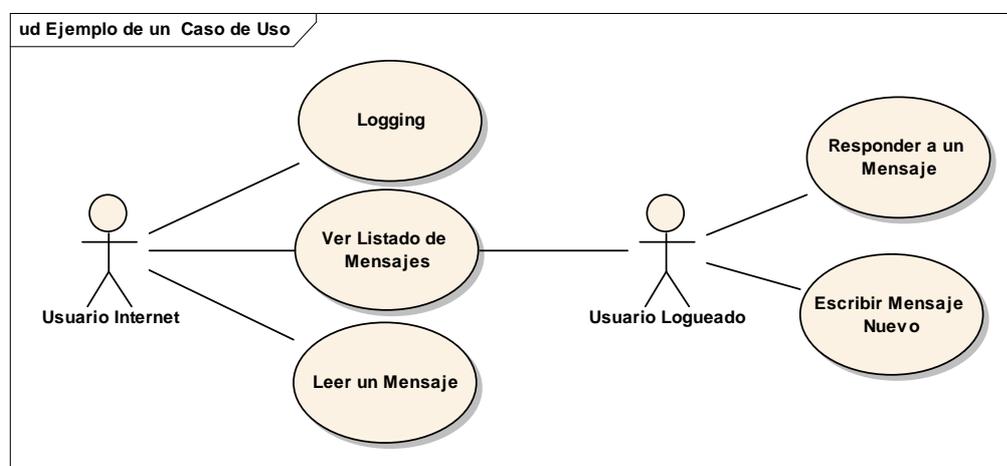


Figura 3. Ejemplo de un Caso de Uso.

¹⁹**Actor:** Representa un conjunto coherente de roles que juegan los usuarios de los casos de uso al interactuar con el sistema.

2.4.2.2. Diagramas de Secuencia

Muestran una interacción ordenada según la secuencia temporal de eventos de un caso de uso. Es un diagrama de la interacción que detalla cómo se realizan las operaciones: se envían qué mensajes y cuando. Se organizan según el tiempo, el cual mientras progresa se recorre el diagrama hacia abajo de la página. Los diagramas de secuencia se centran en la temporalidad de los mensajes que se producen entre los objetos e incluyen: Objetos, línea de tiempo, mensajes con argumentos, ciclo de vida de los objetos, información devuelta por un método y especificación de procesos concurrentes. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba hacia abajo. (Ver Figura 4).

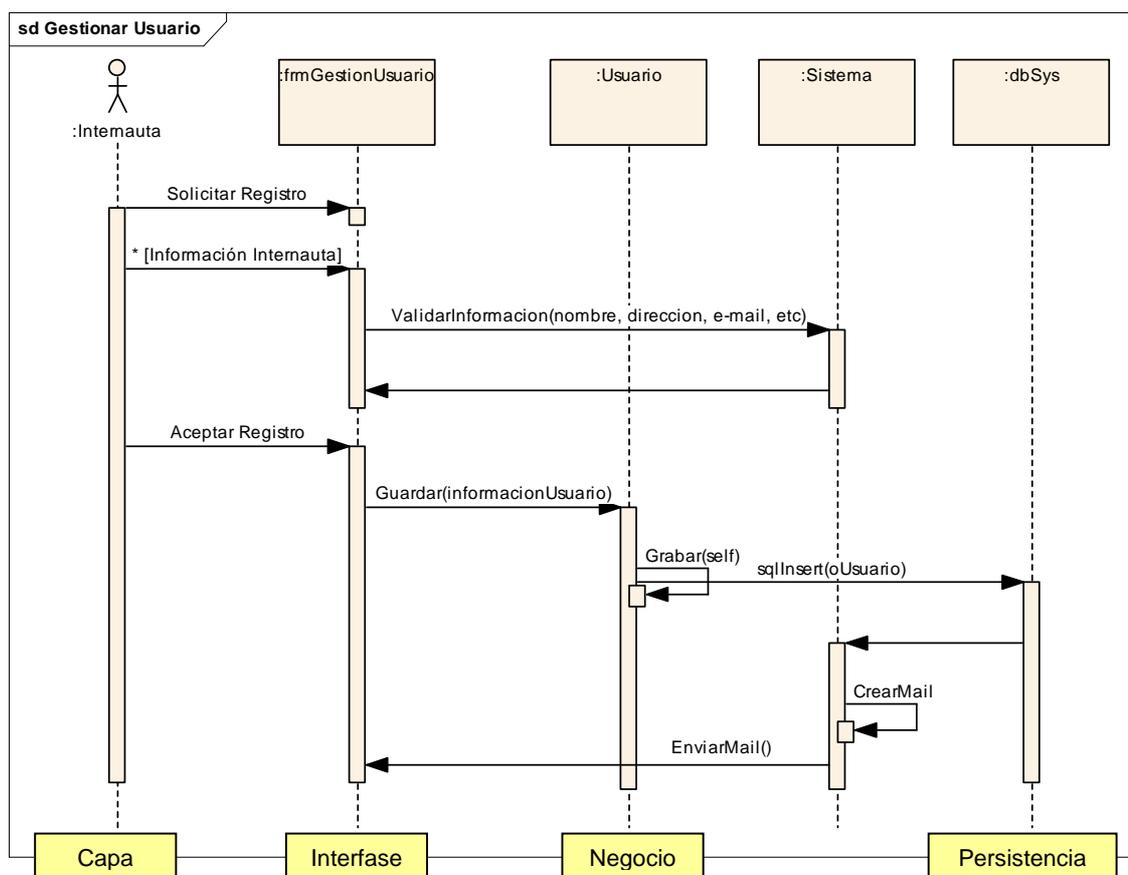


Figura 4. Ejemplo de un Diagrama de Secuencia.

2.4.2.3. Diagramas de Clase

El Diagrama de Clases²⁰ es el diagrama principal para el análisis y diseño. Modela la vista estática estructural del sistema. La definición de clase incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones. Cada diagrama de clases tiene clases, asociaciones, y multiplicidades. Adicionalmente la navegabilidad y los roles son artículos opcionales puestos en un diagrama para proporcionar claridad.

²⁰**El Diagrama de Clases:** Es el más utilizado en los modelos de sistemas OO; son los "planos" principales [Booch].

Clases en UML

Los atributos de una clase no deberían ser manipulables directamente por el resto de objetos. Por esta razón se crearon niveles de visibilidad (ver Figura 5):

- (-) Privado: es el más fuerte. Esta parte es totalmente invisible
- (#) Los atributos/operaciones protegidos están visibles para las clases amigas y para las clases derivadas de la original.
- (+) Los atributos/operaciones públicos son visibles a otras.

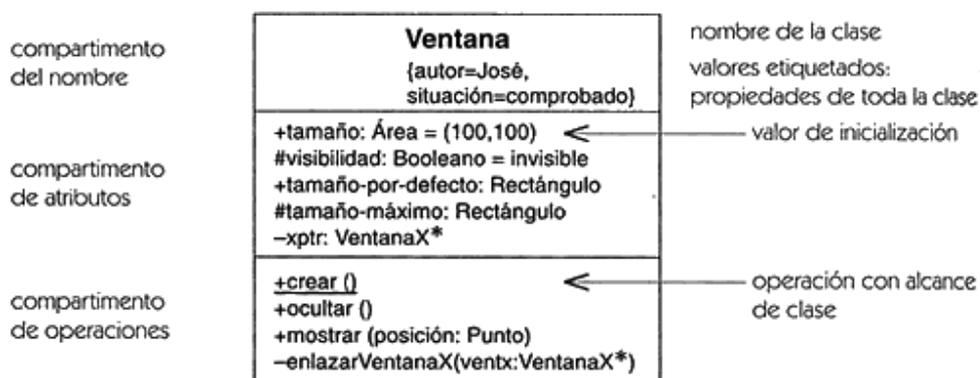


Figura 5. Ejemplo de una Clase.²¹

²¹ Figura tomada de la página web: Diagramas de Clase, <http://www.creangel.com/uml/clases.php#>.

Relaciones entre clases

Los enlaces entre objetos pueden representarse entre las respectivas clases y sus formas de relación son:

- Asociación y Agregación (vista como un caso particular de asociación)
- Generalización/Especialización.

Las relaciones de Agregación y Generalización forman jerarquías de clases.

Asociación

La asociación expresa una conexión bidireccional entre objetos. Las asociaciones representan relaciones entre instancias de clases. Puede determinarse por la especificación de multiplicidad (mínima...máxima)

- Uno y sólo uno
- 0..1 Cero o uno
- M..N Desde M hasta N (enteros naturales)
- 0..* Cero o muchos
- 1..* Uno o muchos (al menos uno)

Una asociación tiene dos extremos. Un extremo puede tener un nombre del rol para clarificar la naturaleza de la asociación.

Las asociaciones pueden detonar la navegabilidad mediante el sentido de las flechas. Las asociaciones sin flechas de navegabilidad son bidireccionales.

Agregación

Es una relación de componente, una asociación en la cual una clase pertenece a una colección. UML ofrece un tipo de agregación más poderosa

que se denomina composición. El objeto parte puede pertenecer a un todo único.

Generalización

Consiste en factorizar las propiedades comunes de un conjunto de clases en una clase más general. Los nombres usados: clase padre - clase hija. Otros nombres: superclase - subclase, clase base - clase derivada. Las subclases heredan propiedades de sus clases padre, es decir, atributos y operaciones (y asociaciones) de la clase padre están disponibles en sus clases hijas.

La Generalización y Especialización son equivalentes en cuanto al resultado: la jerarquía y herencia establecidas. La especialización es una técnica muy eficaz para la extensión y reutilización. (Ver Figura 6).

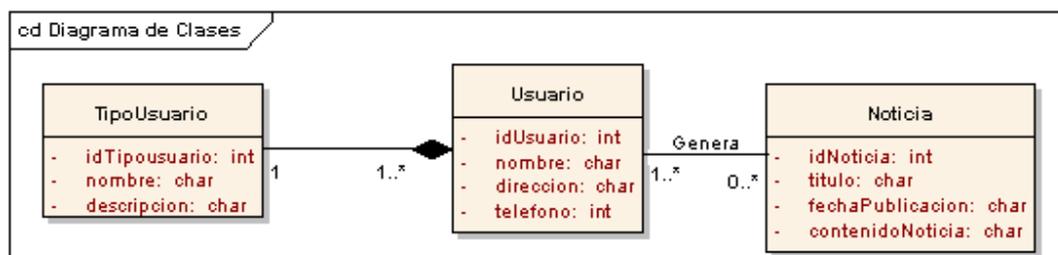


Figura 6. Ejemplo de un Diagrama de Clases.

2.4.2.4. Diagramas de Despliegue

Se utilizan para modelar la topología hardware del sistema, forman parte de la vista física del sistema. Los Diagramas de Despliegue muestran la disposición física de los distintos nodos²² que componen un sistema y el reparto de los componentes sobre dichos nodos. Con UML, estos diagramas se utilizan

²² **Nodo:** Es un elemento físico que representa un recurso con capacidad computacional.

para visualizar los aspectos estáticos de estos nodos físicos y sus relaciones y para especificar sus detalles para la construcción. (Ver Figura 7).

Usos comunes de los diagramas de despliegue:

- Para modelar sistemas: empotrados, cliente/servidor, completamente distribuidos.

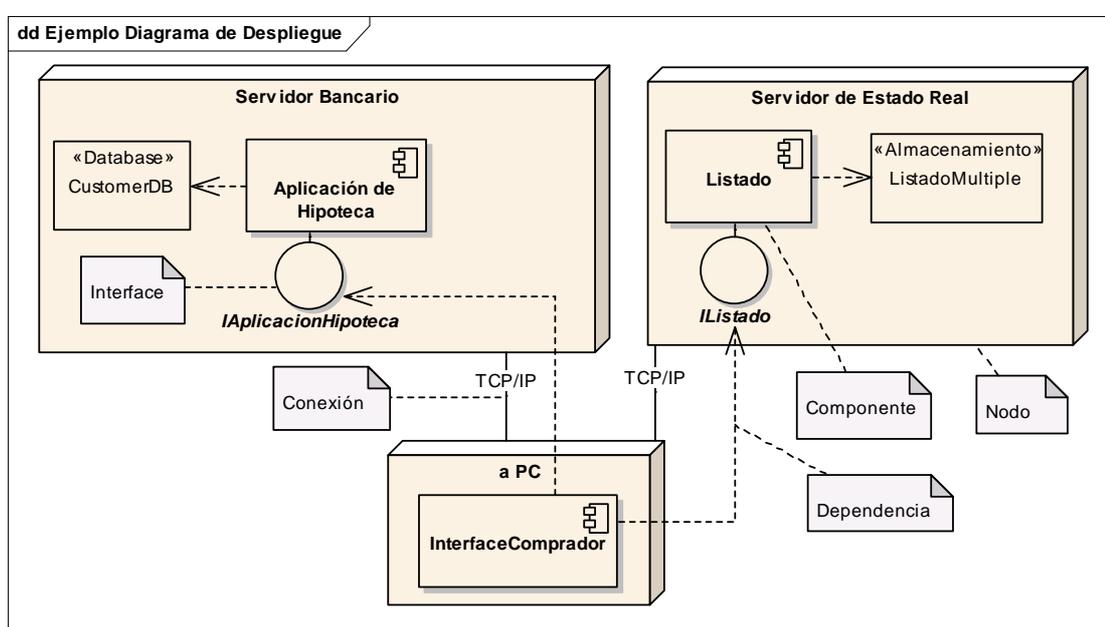


Figura 7. Ejemplo de un Diagrama de Despliegue

2.4.2.5. Diagramas de Componente

Sirve para representar la vista física de un sistema, la cual modela la estructura de implementación de la aplicación, su organización en componentes y su despliegue en nodos de ejecución. Esta vista establece las correspondencias entre las clases, los componentes de implementación y los nodos. Estos diagramas describen los elementos físicos del sistema y sus relaciones:

Los componentes representan una parte física en el sistema los cuales conforman un conjunto de interfaces y proporciona una implementación de clases.

Las relaciones de dependencia se utilizan para indicar que un componente utiliza los servicios ofrecidos por otro componente.

El diagrama también puede usarse para mostrar interfaces y las dependencias de llamada entre componentes, usando flechas con líneas discontinuas desde los componentes a las interfaces de otros componentes. (Ver Figura 8).

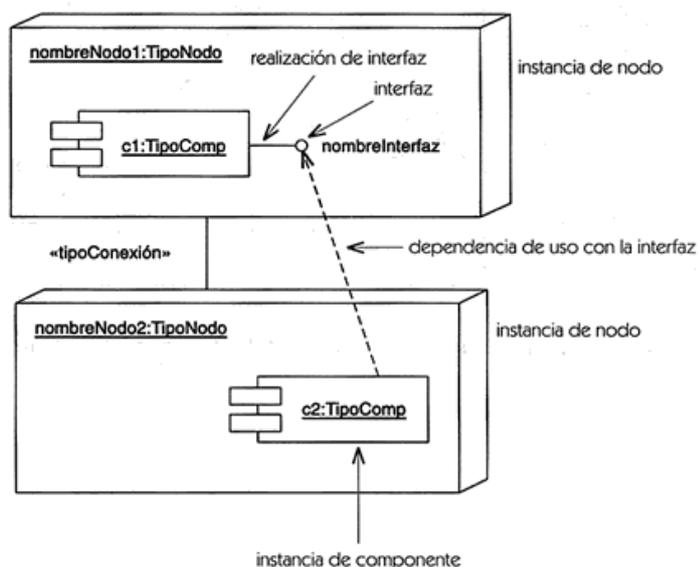


Figura 8. Ejemplo de un Diagrama de Componentes.²³

²³ **Figura tomada de la página web:** Diagrama de Componentes, <http://www.creangel.com/uml/componente.php#>.

CAPÍTULO 3

ANÁLISIS Y DISEÑO DEL JUEGO LÚDICO

3.1 Especificación de Requerimientos

La presente especificación de requerimientos pertenece al desarrollo del Proyecto "Videojuego educativo con técnicas de inteligencia artificial para la plataforma Android aplicando la Metodología OOHDM. Caso de Estudio: Laberinto en 3D" como Tesis para la obtención del título de Ingeniería en Sistemas e Informática y está desarrollada siguiendo las directrices de la metodología OOHDM y del Lenguaje Unificado de Modelado (UML) junto con la Ingeniería de Software.

3.1.1 Introducción

Propósito

El propósito del presente apartado es definir los requerimientos que debe tener el Videojuego educativo con IA para la Plataforma Android, el cual se llama Laberinto3D a partir de ahora. Con la especificación de requerimientos se formalizará las funcionalidades de la aplicación junto al cliente.

Definiciones, Acrónimos y abreviaturas

2D.- Corresponde a la abreviatura utilizada para definir un ambiente de trabajo gráfico en dos dimensiones (two dimensions).

3D.- Corresponde a la abreviatura utilizada para definir un ambiente de trabajo gráfico en tres dimensiones (three dimensions).

LABERINTO3D.- Videojuego educativo con técnicas de inteligencia artificial para la plataforma Android aplicando la Metodología OOADM. Caso de Estudio: Laberinto en 3D.

Game Engine.- Corresponde al término definido en español como Motor de Juegos que se define como un conjunto de herramientas de programación para manejar gráficos en 2D y 3D.

Gamer.- Corresponde al término definido en español como Jugador o Videojugador, que se define como el usuario de un videojuego interactivo sea en 2D o en 3D.

PC.- Corresponde a la abreviatura en inglés de Personal Computer o Computadora Personal que se define como un dispositivo electrónico utilizado para almacenar datos y procesar información de una manera ágil y organizada en archivos y carpetas.

Puntaje.- Corresponde al término definido en inglés como Score, que se define como un valor numérico alcanzado por el jugador, que puede ser representado por el tiempo utilizado para resolver una determinada tarea, créditos logrados en un determinado nivel o niveles del juego, número de obstáculos superados, número de enemigos derrotados, número de vidas ganadas o perdidas, entre otras.

Reto del Juego.- Se define como una determinada tarea o actividad a superar o resolver en el juego que se ejecuta en la PC.

IA.- Corresponde a la abreviatura en español de Inteligencia Artificial (Artificial Intelligence - AI), que se define como el desarrollo de métodos y algoritmos que permitan comportarse a las computadoras de modo inteligente.

GUI.- Corresponde a la abreviatura en inglés de Graphics User Interface o Interface Gráfica de Usuario que se define como un tipo de interface de usuario que permite al usuario interactuar con dispositivos electrónicos con imágenes en vez de líneas de comando.

FPS.- Corresponde a la abreviatura en inglés de First Person Shooter o Disparador en Primera Persona que permite ubicar una cámara que simula la presencia física de un jugador en una aplicación 3D.

3.1.2 Identificación de Roles y Tareas

A) Roles

➤ Jugador- Gamer

Es el usuario que tiene acceso al juego para poder interactuar con la aplicación en 3D, para lo cual debe registrarse como un gamer del juego y puede programar el tiempo que estará vigente dicho usuario en el juego y poder completar los retos del juego.

En la Figura 9 se muestra gráficamente al actor del LABERINTO3D, que como se mencionó anteriormente es el jugador.

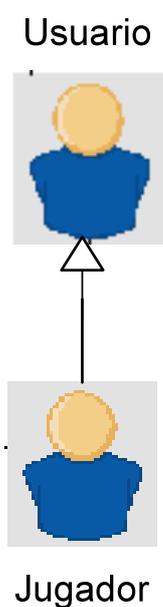


Figura 9. Actores - LABERINTO3D.

B) Tareas

➤ Jugador - Gamer

1. Configurar el videojuego
2. Resolver el laberinto de forma manual
3. Resolver el laberinto de forma automática

3.1.3 Especificación de Escenarios

Rol Jugador

- **Configurar el videojuego:** El jugador podrá configurar las diferentes opciones que tiene el videojuego como son controlar los niveles de acceso al mismo a través de un nivel manual y de un nivel automático.

- **Resolver el laberinto de forma manual:** El jugador podrá resolver los retos del videojuego que consiste en resolver el laberinto en 3D manualmente, utilizando las flechas del teclado y luego almacenar en un archivo plano el nombre del usuario y el tiempo que se demora en resolver el laberinto, en base a los tres niveles de dificultad (principiante, intermedio y avanzado) que tiene el juego.
- **Resolver el laberinto de forma automática:** El jugador podrá visualizar la resolución del laberinto en 3D, para lo cual puede seleccionar cualquiera de los niveles de dificultad que tiene el juego y la computadora resolverá los retos del mismo, utilizando técnicas de IA como lo es la regla de la mano izquierda. En esta opción se almacenará en un archivo plano el nombre de la PC y el tiempo que se demora en resolver el laberinto, en base a los tres niveles de dificultad (principiante, intermedio y avanzado) que tiene el juego.

3.1.4 Especificación de casos de uso por Actor

3.1.4.1 ACTOR: JUGADOR

1. USR-JUG-LABERINTO3D-01: Configurar el videojuego.
2. USR-JUG-LABERINTO3D-02: Resolver el laberinto de forma manual.
3. USR-JUG-LABERINTO3D-03: Resolver el laberinto de forma automática.

En la Figura 10 se muestra el caso de uso para el usuario jugador del LABERINTO3D.

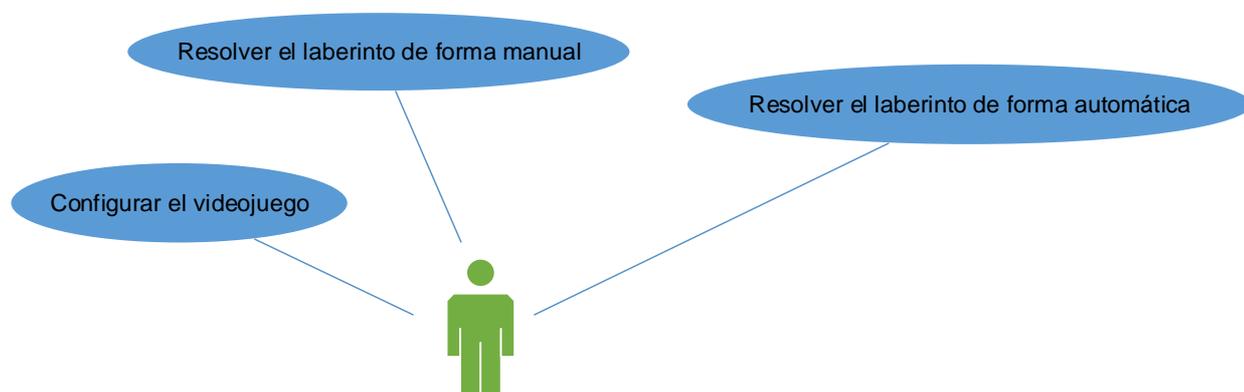


Figura 10. Caso de Uso para el Jugador del LABERINTO3D.

3.1.4.2 CASOS DE USO DEL JUGADOR DEL LABERINTO3D

En la tabla 4 hasta la tabla 6 se detallan las especificaciones de caso de uso del actor Jugador del LABERINTO3D.

Tabla 4. Configurar el videojuego.

USR-JUG-LABERINTO3D-01: Configurar el videojuego	
Resumen:	Proceso en el cual el usuario jugador configura la aplicación que se trata de un juego didáctico de razonamiento lógico, matemático y espacial en 3D.
Prioridad:	Esencial
Actores Directos:	Jugador
Escenarios	

Tipo de Escenario	Descripción	
Principal	<p>1. La aplicación en 3D ofrece al usuario la posibilidad de configurar el tamaño de la pantalla tanto para una PC normal como para un tableta.</p> <p>2. El jugador puede seleccionar la opción de:</p> <ul style="list-style-type: none"> - Configurar el tamaño de la pantalla para PC. - Configurar el tamaño de la pantalla para una tableta. 	
Secundario	Configurar el tamaño de la pantalla para PC	<p>El sistema muestra los una lengüeta con dos opciones:</p> <ul style="list-style-type: none"> - Resolución de pantalla: 1024x768 (por defecto). - Calidad de los gráficos: buena (por defecto).
Secundario	Configurar el tamaño de la pantalla para una tableta	<p>El sistema muestra los una lengüeta con dos opciones:</p> <ul style="list-style-type: none"> - Resolución de pantalla: 512x384 (por defecto). - Calidad de los gráficos: buena (por defecto).
Pre-condiciones		
Ninguna		

Tabla 5. Resolver el laberinto de forma manual.

USR-JUG-LABERINTO3D-02: Resolver el laberinto de forma manual		
Resumen:	Proceso en el cual el usuario jugador interactúa con la aplicación que se trata de un juego didáctico de razonamiento lógico, matemático y espacial en 3D.	
Prioridad:	Esencial	
Actores Directos:	Jugador	
Escenarios		
Tipo de Escenario	Descripción	
Principal	<p>1. La aplicación en 3D ofrece al usuario la posibilidad de configurar el videojuego de tal manera que pueda escoger entre tres niveles de dificultad: a) principiante; b) intermedio; c) avanzado.</p> <p>2. El jugador puede seleccionar la opción de:</p> <ul style="list-style-type: none"> - Registro del nombre. - Seleccionar opción de prueba y entrenamiento. - Seleccionar el nivel del juego. - Guardar puntaje. 	
Secundario	Registro del nombre	<p>1. El sistema muestra los controles necesarios para el ingreso del nuevo registro.</p> <p>2. El usuario ingresa los datos del nuevo registro.</p> <p>3. El sistema valida los datos ingresados.</p> <p>4. Se almacena en memoria temporalmente el nombre del</p>

		usuario del juego que luego se almacenará en disco con el valor del tiempo que se demoró en resolver el laberinto.
Secundario	Seleccionar opción de prueba y entrenamiento	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para seleccionar el nivel de prueba o entrenamiento de la aplicación. 2. El sistema genera un laberinto en 3D con técnicas de IA. 3. El sistema muestra una opción de jugar de forma manual, donde el usuario opera el juego con unas flechas para mover a un ratoncito dentro del juego hasta encontrar la salida donde está un queso. 4. El sistema valida el control de colisiones entre el ratoncito con las paredes del laberinto en 3D.
Secundario	Seleccionar el nivel del juego	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para seleccionar el nivel de dificultad de la aplicación. 2. El usuario selecciona o cambia el nivel de dificultad del juego. 3. Se almacena en memoria temporalmente el nivel del juego que luego se almacenará en disco con el resto de los datos en la sección de puntajes.
Secundario	Guardar puntaje	<ol style="list-style-type: none"> 1. Se almacenan en disco el puntaje obtenido por el usuario. 2. El sistema guarda automáticamente los valores de los puntajes obtenidos por un usuario en la aplicación.

Pre-condiciones
1. El usuario debe ingresar al sistema autenticado como Jugador.

Tabla 6. Resolver el laberinto de forma automáticamente.

USR-JUG-LABERINTO3D-02: Resolver el laberinto de forma automática	
Resumen:	Proceso en el cual la computadora interactúa con el usuario jugador y con la aplicación que se trata de un juego didáctico de razonamiento lógico, matemático y espacial en 3D.
Prioridad:	Esencial
Actores Directos:	PC
Escenarios	
Tipo de Escenario	Descripción
Principal	<p>1. La aplicación en 3D ofrece al usuario la posibilidad de configurar el videojuego de tal manera que pueda escoger entre tres niveles de dificultad para que la PC trabaje con IA: a) principiante; b) intermedio; c) avanzado.</p> <p>2. El jugador puede seleccionar la opción de:</p> <ul style="list-style-type: none"> - Seleccionar opción de prueba y entrenamiento para que resuelva la PC con IA. - Seleccionar el nivel del juego para que resuelva la PC con IA. - Guardar puntaje.

Secundario	Seleccionar opción de prueba y entrenamiento	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para seleccionar el nivel de prueba o entrenamiento de la aplicación. 2. El sistema genera un laberinto en 3D con técnicas de IA dependiendo del nivel de dificultad. 3. El sistema resuelve con técnicas de IA el laberinto en 3D, para lo cual la computadora toma el control sobre el ratoncito y lo mueve desde el inicio hasta la salida donde está un queso, buscando un camino y planificando la salida. 4. El sistema valida el control de colisiones entre el ratoncito con las paredes del laberinto en 3D.
Secundario	Seleccionar el nivel del juego	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para seleccionar el nivel de dificultad de la aplicación. 2. El usuario selecciona o cambia el nivel de dificultad del juego. 3. El sistema genera un laberinto en 3D con técnicas de IA dependiendo del nivel de dificultad. 4. El sistema resuelve con técnicas de IA el laberinto en 3D, para lo cual la computadora toma el control sobre el ratoncito y lo mueve desde el inicio hasta la salida donde está un queso, buscando un camino y planificando la salida. 5. El sistema valida el control de colisiones entre el ratoncito

		<p>con las paredes del laberinto en 3D.</p> <p>6. Se almacena en memoria temporalmente el nivel del juego que luego se almacenará en disco con el resto de los datos en la sección de puntajes de la PC.</p>
Secundario	Guardar puntaje	<p>1. Se almacenan en disco el puntaje obtenido por la PC.</p> <p>2. El sistema guarda automáticamente los valores de los puntajes obtenidos por la PC en la aplicación.</p>
Pre-condiciones		
1. El usuario no necesita ingresar al sistema autenticado como Jugador.		

3.1.5 Requerimientos No Funcionales

- El sistema debe poseer un esquema de perfil de usuario con un solo rol definido que es el del jugador, que garantice el acceso al juego, de tal manera que un jugador no registrado no tendrá acceso a todos los datos u operaciones disponibles.
- El sistema debe poseer un esquema de funciones para generar automáticamente el laberinto en 3D, utilizando técnicas de IA.
- El sistema debe poseer un esquema de funciones para resolver automáticamente el laberinto en 3D, utilizando técnicas de IA.

- El sistema debe poseer un diseño de almacenamiento de datos que permita leer y escribir en archivos planos el puntaje del jugador en función del tiempo que se demora y de los niveles de dificultad superados.

3.2 Diagramas de Secuencia

Se encuentran en el **Anexo 2** del presente trabajo y representan el diseño de navegación de la aplicación móvil.

3.3 Diseño Conceptual

3.3.1 Diseño de los Archivos Planos

El diagrama de la Figura 11 muestra el diagrama de los archivos planos que se utilizan para leer y escribir la información de los jugadores y sus puntajes del juego.



Figura 11. Archivos Planos del LABERINTO3D.

3.3.2 Diagrama de Clases

La Figura 12 muestra el diagrama de clases del sistema, que como se puede ver tiene más de 15 clases.

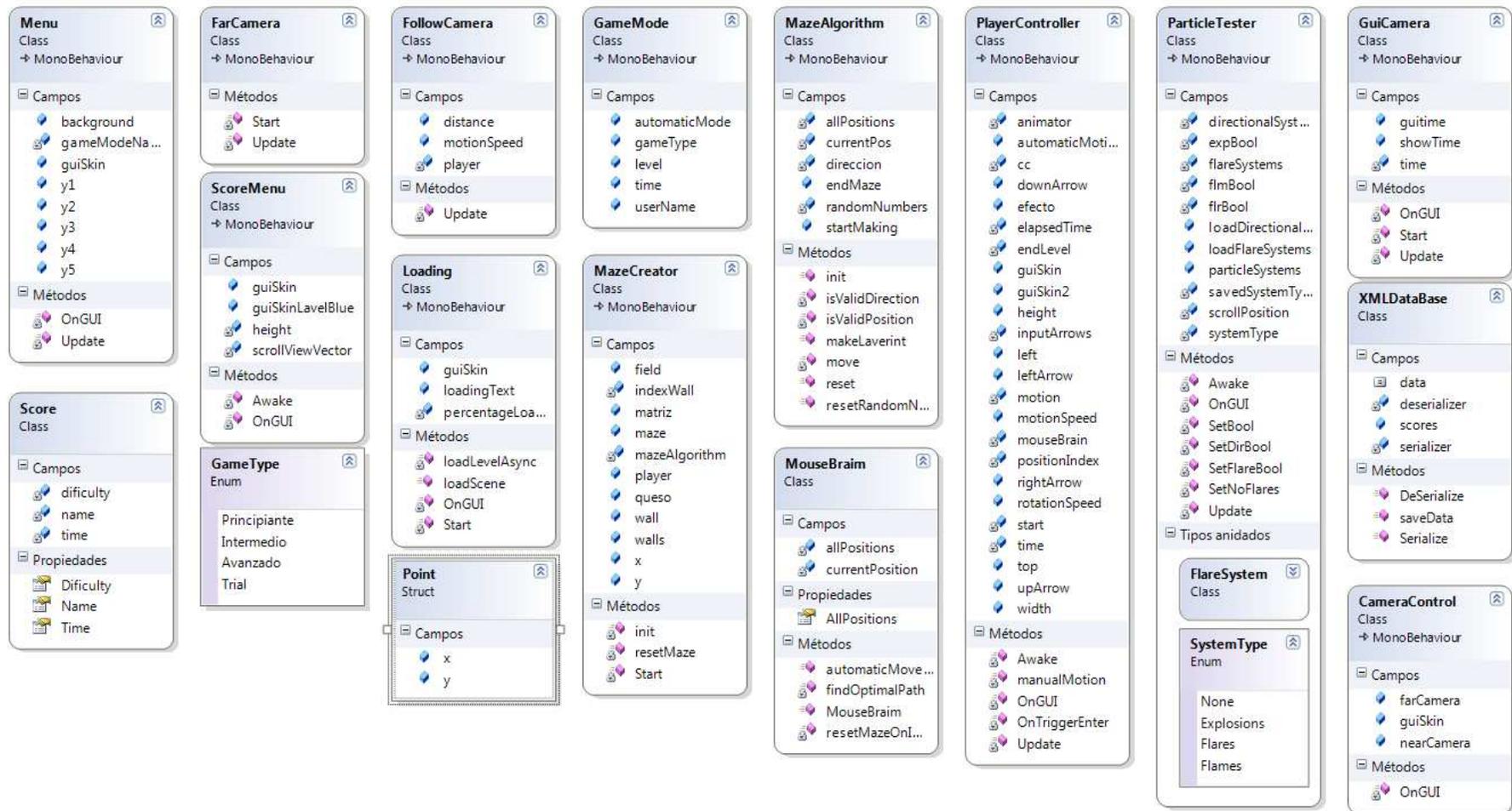


Figura 12. Diagrama de Clases del LABERINTO3D.

3.4 Diseño Navegacional

El sistema está basado en formularios, por lo tanto contiene una mezcla adecuada de estética, contenido y tecnología. Posee los siguientes objetos y contextos navegacionales:

A) *Objetos Navegaciones*

- Página del Formulario de Configuración del Juego
- Página del Laberinto de Modo Manual
- Página del Laberinto de Modo Automático
- Página del Formulario de Navegación y Entorno del Juego
- Página del Formulario de los Puntajes del Juego

B) *Contextos Navegacionales*

- Iniciar Sesión
- Configurar Juego
- Resolver el laberinto de forma manual
- Resolver el laberinto de forma automática
- Guardar Puntajes
- Visualizar Puntajes

3.4.1 Esquema Navegacional

A) Clases Navegacionales

Las tablas 7 hasta la 11 resumen las principales características de las clases navegacionales del sistema.

Tabla 7. Clase Navegacional – Nodo Formulario de la Configuración del Juego.

Nombre: Nodo Formulario de la Configuración del Juego
Clases Conceptuales (CC):
Atributos: <ul style="list-style-type: none"> ➤ Resolución de la pantalla ➤ Calidad de los gráficos ➤ Entradas (Input)
Descripción: Nodo donde se encuentran las diferentes opciones de configuración tanto para una PC como para una tableta.
Enlaces: <ul style="list-style-type: none"> ➤ Al Nodo del Laberinto de Modo Manual ➤ Al Nodo del Laberinto de Modo Automático ➤ Al Nodo de los Puntajes

Tabla 8. Clase Navegacional – Nodo del Laberinto de Modo Manual.

Nombre: Nodo del Laberinto de Modo Manual
Clases Conceptuales (CC):
Atributos: <ul style="list-style-type: none"> ➤ Nombre Usuario ➤ Nivel Trial ➤ Nivel Principiante ➤ Nivel Intermedio ➤ Nivel Avanzado
Descripción: Nodo donde se encuentran los diferentes niveles de dificultad para el usuario, para empezar a jugar con el videojuego.
Enlaces: <ul style="list-style-type: none"> ➤ Al Nodo del Laberinto de Modo Automático ➤ Al Nodo de los Puntajes

Tabla 9. Clase Navegacional – Nodo del Laberinto de Modo Automático.

Nombre: Nodo del Laberinto de Modo Automático
Clases Conceptuales (CC):
Atributos: <ul style="list-style-type: none"> ➤ Nombre Usuario ➤ Nivel Trial ➤ Nivel Principiante ➤ Nivel Intermedio

➤ Nivel Avanzado
Descripción: Nodo donde se encuentran los diferentes niveles de dificultad para la PC con IA, para empezar a jugar con el videojuego.
Enlaces:
➤ Al Nodo del Laberinto de Modo Manual
➤ Al Nodo de los Puntajes

Tabla 10. Clase Navegacional – Nodo de Navegación y Entorno del Juego.

Nombre: Nodo de Navegación y Entorno del Juego
Clases Conceptuales (CC):
Atributos:
➤ Bandera de Control
➤ Objeto de tipo Cámara
➤ Flechas de Navegación
Descripción: Nodo donde se encuentra el área de trabajo para el usuario que es el laberinto en 3D y su navegación.
Enlaces:
➤ Al Nodo del Laberinto de Modo Manual
➤ Al Nodo del Laberinto de Modo Automático

Tabla 11. Clase Navegacional – Nodo de Puntajes del Juego.

Nombre: Nodo de Puntajes del Juego
Clases Conceptuales (CC):
Atributos: <ul style="list-style-type: none"> ➤ Nombre del Jugador ➤ Tiempo del Juego ➤ Nivel de Dificultad
Descripción: Nodo donde se encuentran los diferentes puntajes de los usuarios y la PC, obtenidos en los diferentes niveles.
Enlaces: <ul style="list-style-type: none"> ➤ Al Nodo del Laberinto de Modo Manual ➤ Al Nodo del Laberinto de Modo Automático

B) Modelo de Clases Navegacionales

La Figura 13 muestra el Modelo de Clases Navegacionales del sistema, en la cual a través del Nodo Formulario Menú Principal se acceden a los otros nodos.

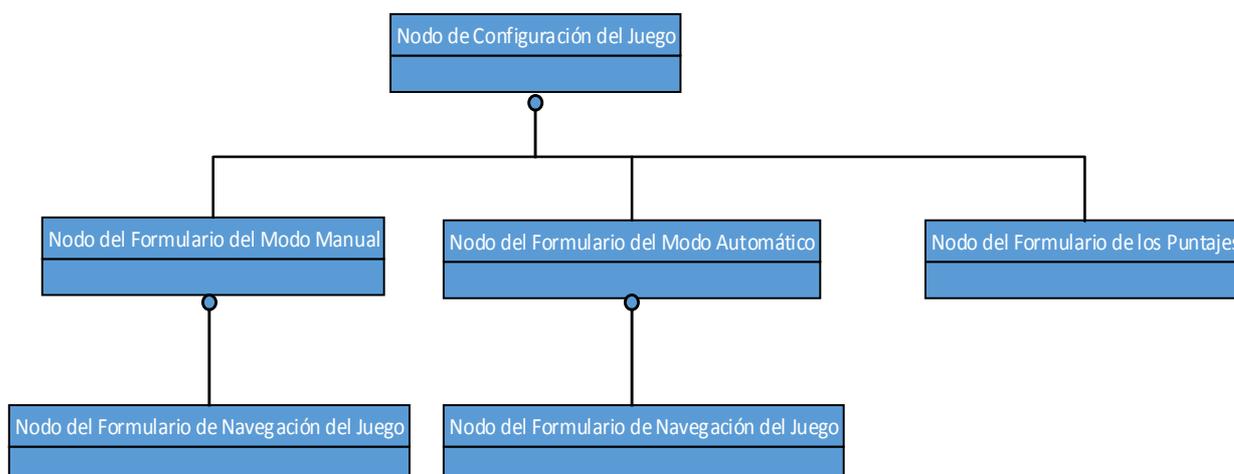


Figura 13. Modelo de Clases Navegacional del LABERINTO3D.

3.4.2. Esquema de Contextos Navegacionales

El sistema posee una estructura compuesta basada en el concepto de diseño arquitectónico de la Ingeniería de Software, que consta de cinco elementos, como se puede ver en la Figura 14:

- a) Configuración del Juego
- b) Modo Manual del Juego
- c) Modo Automático del Juego
- d) Puntajes del Juego
- e) Navegación y entorno del juego

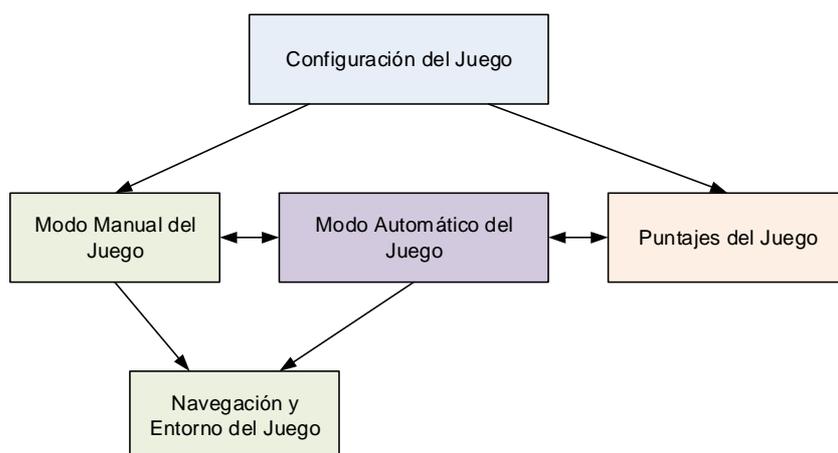


Figura 14. Esquema de Contexto - Menú Principal.

3.4.3. Arquitectura del Sistema

La aplicación de software es un sistema de Información que posee la siguiente arquitectura Modelo-Vista-Controlador, como se muestra en la Figura 15:

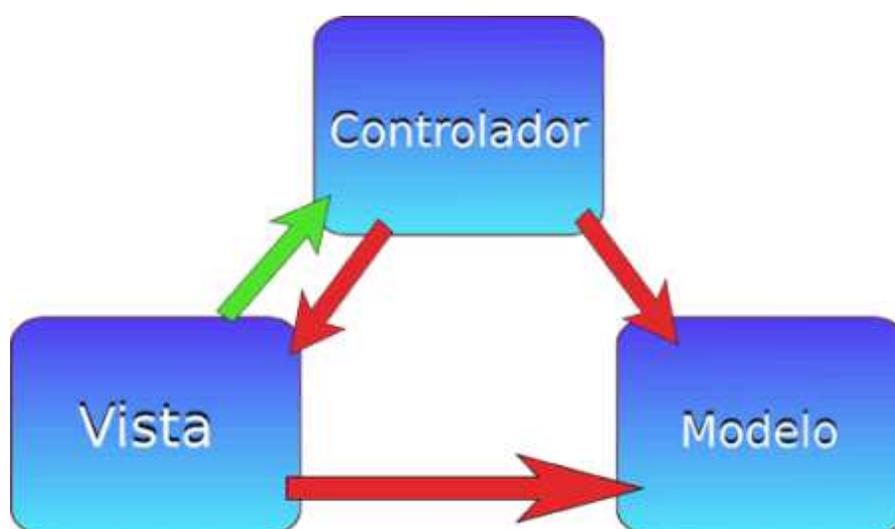


Figura 15. Arquitectura del LABERINTO3D.

Los componentes de la aplicación desktop y móvil con los datos en los archivos XML se encuentran instalados en una misma PC o Tablet respectivamente.

3.5 Diseño de Interfaz Abstracta

Las interfaces están basadas tanto en una arquitectura de contenido (forma en la que los objetos se estructuran para su presentación y navegación) como en una arquitectura de Aplicación Móvil (forma en la que la aplicación se estructura para gestionar la interacción del usuario).

Se detallan a continuación las interfaces del usuario Jugador; ya que posee una interacción directa con el sistema.

3.5.1 Vista de Datos Abstractos

- Nodo del Formulario de Configuración del Juego

El diagrama muestra un formulario de configuración del juego con el título "ADV Nodo del Formulario de Configuración del Juego". El formulario está dividido en secciones:

- Gráficos**
 - Resolución de la pantalla PC: <1024x768>
 - Resolución de la pantalla Tableta: <512x384>
 - Calidad de los gráficos: <bueno>
- Entradas**
 - <Horizontal(+) flecha derecha>
 - <Horizontal(-) flecha izquierda>
 - <Vertical(+) flecha arriba>
 - <Vertical(-) flecha abajo>

Figura 16. Vista Abstracta del Nodo del Formulario de Configuración del Juego.

➤ Nodo del Formulario del Juego Manual

El diagrama muestra una ventana de software con el título "ADV Nodo del Formulario del Juego Manual". Dentro de la ventana, hay un formulario con el título "Formulario de Información del Juego". Debajo del título del formulario, hay un botón con un icono de flecha y el texto "<Registro del usuario>". A continuación, hay cinco campos de entrada de texto, uno por línea, con los siguientes textos: "Trial", "Principiante", "Intermedio", "Avanzado" y "Puntajes".

Figura 17. Vista Abstracta del Nodo del Juego Manual.

➤ Nodo del Formulario del Juego Automático

El diagrama muestra una ventana de software con el título "ADV Nodo del Formulario del Juego Automático". Dentro de la ventana, hay un formulario con el título "Formulario de Información del Juego". Debajo del título del formulario, hay cinco campos de entrada de texto, uno por línea, con los siguientes textos: "Trial", "Principiante", "Intermedio", "Avanzado" y "Puntajes".

Figura 18. Vista Abstracta del Nodo del Juego Automático.

- Nodo del Formulario de la Navegación y Entorno del Juego

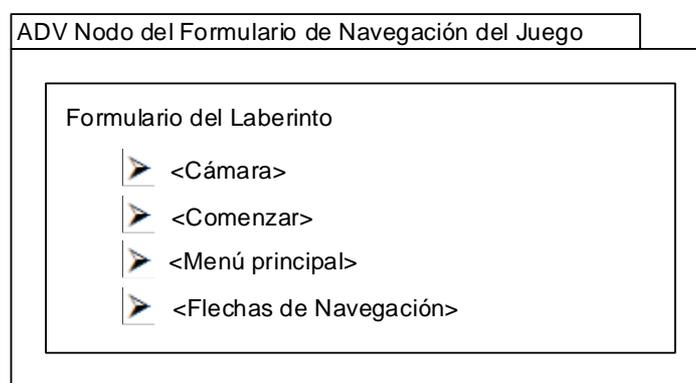


Figura 19. Vista Abstracta del Nodo del Formulario de la Navegación y Entorno del Juego.

- Nodo del Formulario de los Puntajes del Juego

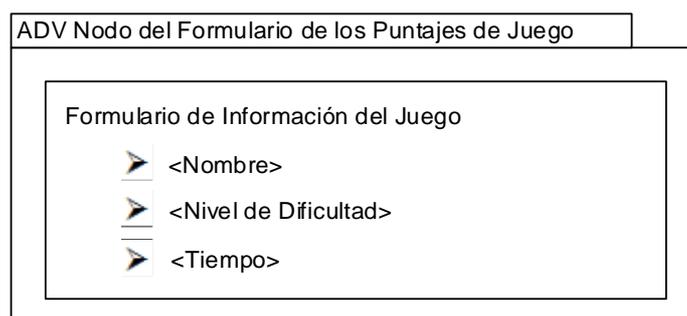


Figura 20. Vista Abstracta del Nodo del Formulario de los Puntajes del Juego.

3.5.2. Diagramas de Configuración

Son las mismas vistas abstractas descritas anteriormente con la particularidad de que todos los enlaces poseen la opción de *Mouse Clicked* que permite acceder a todas las opciones del juego.

3.5.3. Diagramas de Estado

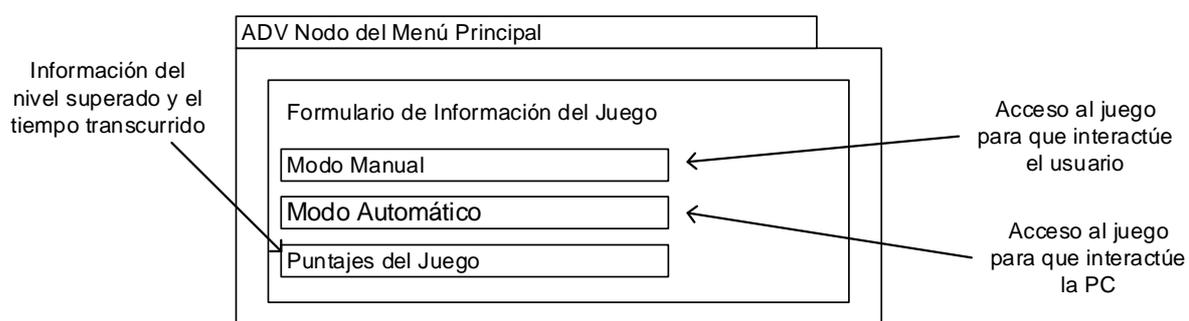


Figura 21. Diagrama de Estado del LABERINTO3D.

Todos los diagramas de Estado de los otros menús poseen la misma lógica que el Menú Principal y cada etiqueta de los links posee un nombre relacionado con el detalle del artículo o con el servicio a ser desplegado.

3.6. Diseño Estético

3.6.1. Características de la Plantilla

- Espacio en blanco: 10%
- Contenido: 90%
- Organización de los elementos:

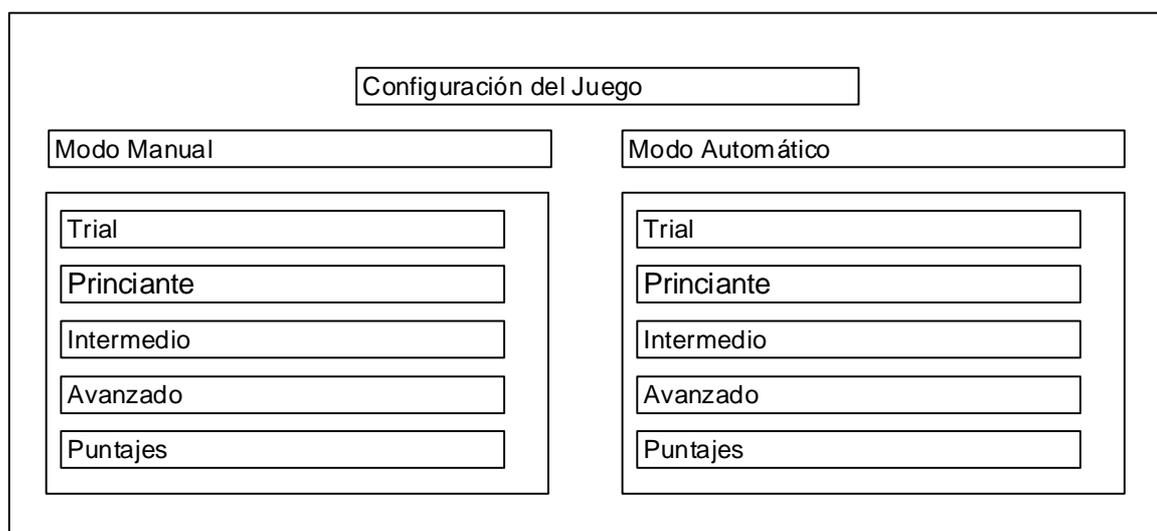


Figura 22. Diseño Estético – Organización de los Elementos.

- Agrupación de contenido dividida por: manual de usuario, configuración del juego, navegación y entorno del juego y puntajes del mismo.
- Se extiende el mapa del juego LABERINTO3D, a medida que el usuario se desplaza por el centro del laberinto, con el fin de que el usuario se sienta más cómodo navegando por la aplicación móvil.
- El tamaño del sistema: 100% del espacio total disponible, en una resolución de 1024 x 768 para PC y 512 x 384 para una tableta.

3.6.2. Consideraciones de Diseño Gráfico

Cabecera: Animaciones 3D hechas con Maya:

- Tipo de letra: Copperplate Gothic Bold
- Tamaño promedio: 33
- Mezcla de colores RGB para los fondos y colores del entorno del juego.

Colores Globales de la aplicación: café, negro y en menor cantidad: blanco y colores pasteles. Tipos de Fuente: Arial. Tamaño de Fuente: 12px. Estilo del portal en general: Archivo template.

3.7. Diseño de Componentes

En la Figura 23 se muestra el Diseño de Componentes del Juego, donde como se puede ver, el juego es una aplicación móvil que utiliza los componentes de Unity y C# para desarrollar aplicaciones de este tipo. Además se desarrolló el componente GameApp que permite integrar todos los componentes de Unity y C# en la aplicación del juego.

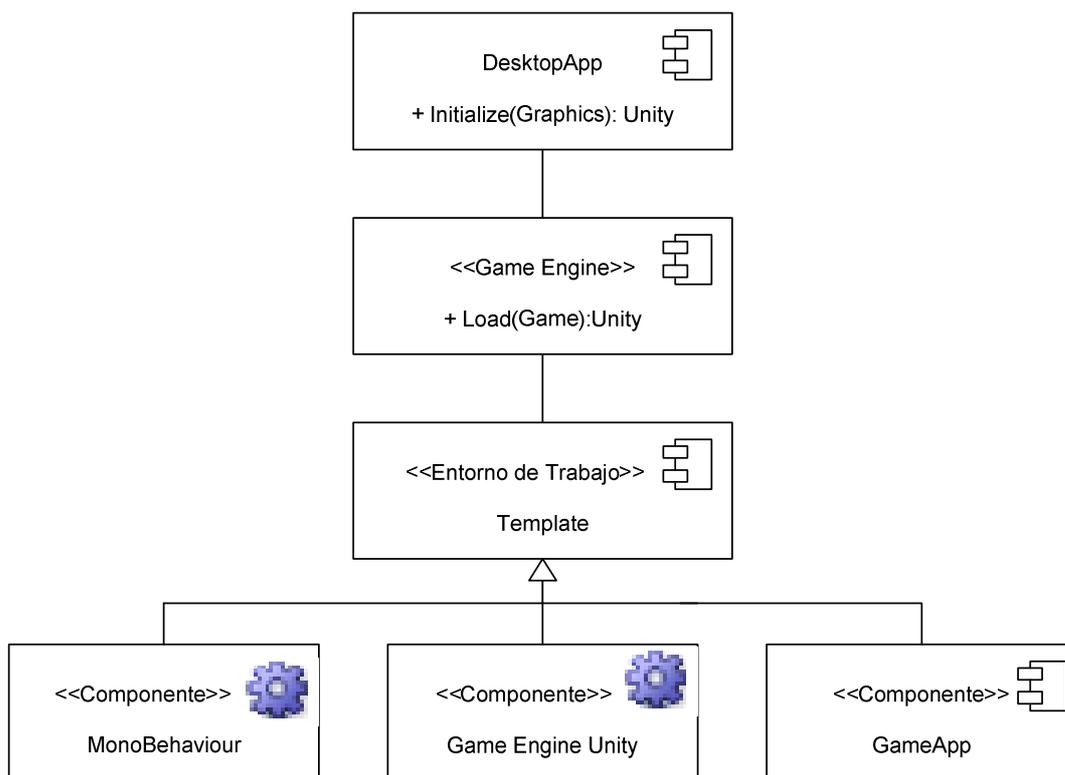


Figura 23. Diseño de Componentes del LABERINTO3D.

3.8. Diagrama de Despliegue

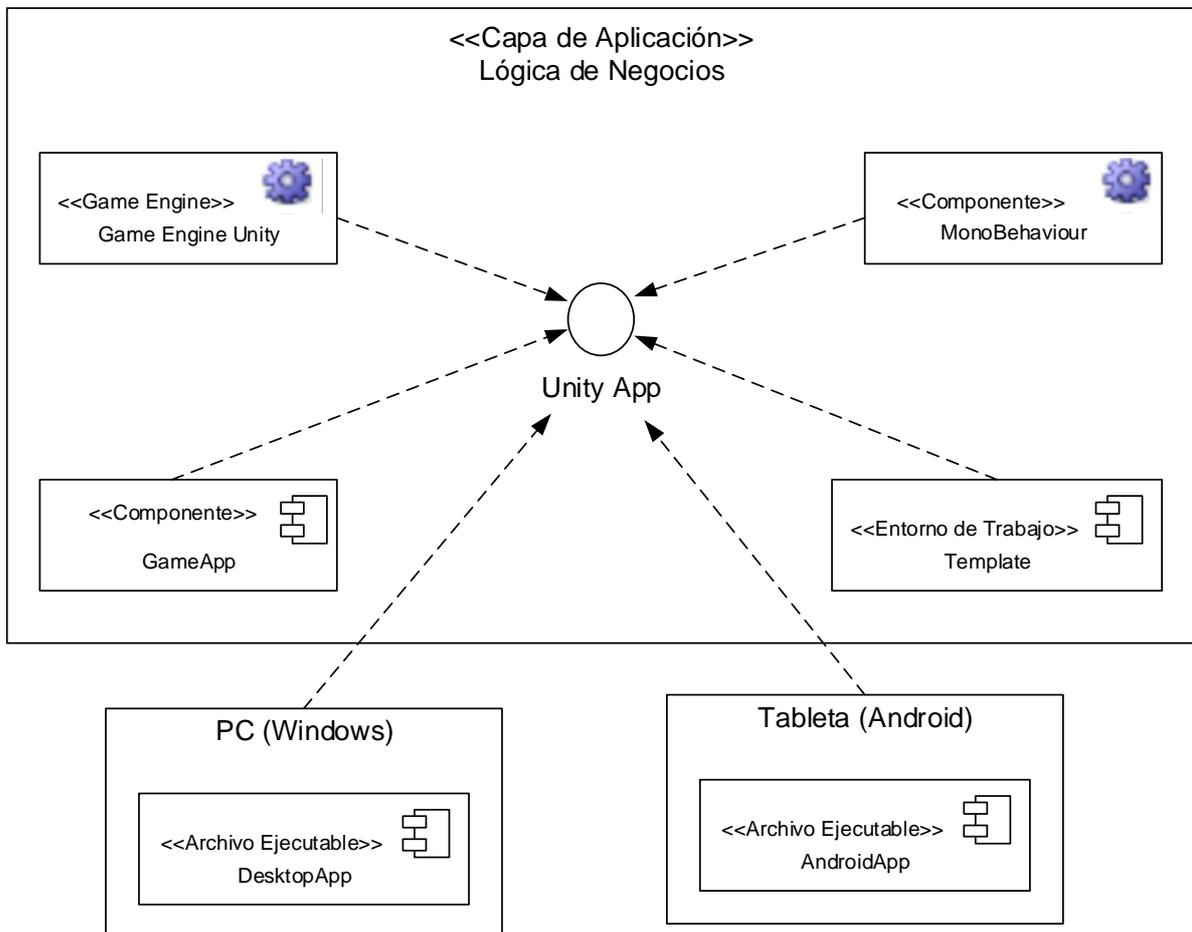


Figura 24. Diagrama de Despliegue del LABERINTO3D.

CAPÍTULO 4

IMPLEMENTACIÓN Y PRUEBAS DEL VIDEOJUEGO CON UNITY

4.1 Unity

4.1.1 Definición

Unity es un motor gráfico 3D para PC y Mac que viene empaquetado como una herramienta para crear juegos, aplicaciones interactivas, visualizaciones y animaciones en 3D y tiempo real. Unity puede publicar contenido para múltiples plataformas como PC, Mac, Nintendo Wii y iPhone. El motor también puede publicar juegos basados en la web usando el plugin Unity web player, que es un componente que permite interpretar código y los mapas de archivos 3D.

4.1.2 Características

A) Instalación

Puede ser instalado fácilmente bajo ambientes Windows y Mac, para lo cual se puede descargar gratuitamente la licencia de prueba en la dirección Web: <http://unity3d.com/unity/download/>. Una vez descargada e instalada la aplicación aparecerá el siguiente ícono de activación del paquete. (Ver Figura 25)



Figura 25. El ícono de carga del paquete Unity.

B) Interfaz de Usuario

La Interfaz de Usuario de Unity tiene 5 áreas principales de trabajo, numeradas en la Figura 26.

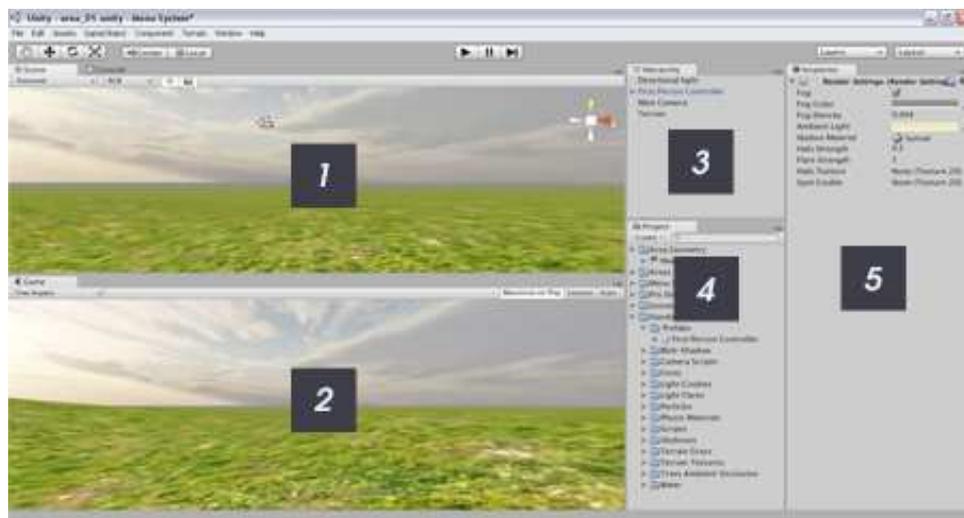


Figura 26. Interfaz de Usuario de Unity.

1. **Vista de Escena:** La escena es el área de construcción de Unity donde se construye visualmente cada escena de un juego.
2. **Vista de Juego:** En la vista de juego se obtiene una pre-visualización de un juego. En cualquier momento se puede reproducir un juego y jugarlo en esta vista.
3. **Vista de Proyecto:** Esta es la librería de assets para un juego, similar a la librería en Flash. Se puede importar objetos 3D de distintas

aplicaciones a la librería, se puede importar texturas y crear otros objetos como Scrips o Prefabs que se almacenaran aquí. Todos los assets que se importan en un juego se almacenaran aquí para que se puedan usar en un juego.

4. **Vista de Jerarquía:** La vista de jerarquía contiene todos los objetos en la escena actual.
5. **Vista de Inspector:** La vista de inspector sirve para varias cosas. Si se seleccionan objetos entonces mostrará las propiedades de ese objeto donde se pueden personalizar varias características del objeto. También contiene la configuración para ciertas herramientas como la herramienta de terrenos si se tiene un terreno seleccionado.

C) Modos de Visualización

Por defecto la vista de escena tiene una perspectiva 3D de la escena. Se puede cambiar esto por un número de vistas Ortográficas: 1) top-down; 2) side; 3) front. En la parte derecha de la vista de la escena se puede ver un “Gizmo” que parece una caja con conos que salen de ella. (Ver Figura 27).



Figura 27. Gizmo de Perspectiva 3D.

1. **Perspectiva 3D top-down (arriba-abajo):** En este modo se visualiza la escena y sus objetos vistos desde arriba y desde abajo, para lo cual se hace clic en el cono verde del “Gizmo” de perspectiva.

2. **Perspectiva 3D side (derecha):** En este modo se visualiza la escena y sus objetos vistos desde arriba y desde abajo, para lo cual se hace clic en el cono rojo del “Gizmo” de perspectiva.

3. **Perspectiva 3D front (frontal):** En este modo se visualiza la escena y sus objetos vistos desde arriba y desde abajo, para lo cual se hace clic en el cono azul del “Gizmo” de perspectiva.

D) Configuración de la Visualización

En la esquina izquierda de la vista de escena se encuentra un conjunto de botones para cambiar la configuración general de la visualización. De izquierda a derecha se tienen tres botones que son: 1) Render Mode; 2) Color Modes; 3) Interruptor de luces; 4) Interruptor de skybox, lense flare y niebla. (Ver Figura 28)



Figura 28. Configuración de la Visualización.

1. **Render Mode:** La primera opción es Render mode o modo de renderización en castellano. Por defecto aparecerá en “Textured”. Si se hace clic en este botón aparecerá una lista desplegable con un número de diferentes opciones de renderizado:

a) **Textured:** Las texturas se renderizan en la vista.

b) **Wireframe:** Las superficies no se renderizan, solo se ve la malla.

c) **Textured Wireframe:** Las texturas se renderizan, pero también se ve la malla.

2. **Color Modes:** La segunda opción es el modo de color, que aparece como “RGB” por defecto. Si se hace clic sobre este botón aparecerá una lista desplegable que mostrará los modos de color disponibles:
 - a. **RGB:** Todos los colores son renderizados.
 - b. **Alpha:** El modo es cambiado a “Alpha”.
 - c. **Overdraw:** El modo es cambiado a “Overdraw”.
 - d. **Minimaps:** El modo es cambiado a “Minimap”.
3. **Interruptor de luces:** El siguiente botón enciende o apaga la iluminación del escenario. Apagar la iluminación resultará en una escena mostrada sin luces; lo que puede ser útil para el rendimiento y también si no hay luces en la escena. Encender la luz provocará que las luces tengan efecto sobre la escena. Si no tienes luces en la escena esta será oscura, ya que no hay luz.
4. **Interruptor de skybox, lense flare y niebla:** El último botón activa y desactiva estos tres efectos. Esta opción es útil para desactivar los efectos por razones de rendimiento o visibilidad al trabajar sobre una escena.

E) Botones de Control

Debajo de las opciones de visualización se tiene una fila con 4 botones, como se muestra en la Figura 29. Se puede usar Q, W, E, R para alternar entre cada uno de los controles, que se detalla a continuación:



Figura 29. Botones de Control.

1. **Hand Tool (Q):** Este control permite moverse alrededor en la vista de escena. Mantener ALT permitirá rotar, COMMAND/CTRL permitirá hacer zoom y SHIFT incrementa la velocidad de movimiento mientras se usa la herramienta.
1. **Translate Tool (W):** Permite mover cualquier objeto seleccionado en la escena en los ejes X, Y y Z.
2. **Rotate Tool (E):** Permite rotar cualquier objeto seleccionado en la escena.
3. **Scale Tool (R):** Permite escalar cualquier objeto seleccionado en la escena.

4.1.3 Librerías de Unity

Las principales librerías que maneja el Unity son cuatro como se explican a continuación:

4.1.3.1 Creación de Escenas (Scenes Creator)

Unity tiene un DLL que maneja la creación de escenas en 3D donde se ubican todos los elementos u objetos del juego en 3D como planos, edificios, terrenos, cielo, personajes, etc, como se puede ver en la Figura 30.

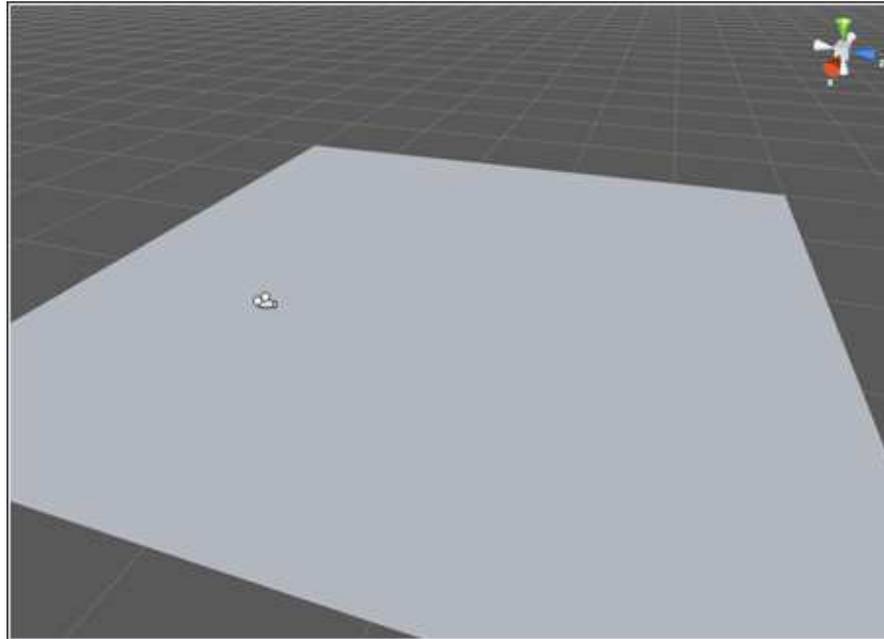


Figura 30. Creación de Escenas.

4.1.3.2 Generador de Terrenos (Terrain Generator)

Unity tiene un DLL para generar terrenos los mismos que son generados como una malla plana que se puede texturizar y esculpir sin salir del editor. Los terrenos tienen algunas propiedades importantes, como la longitud del terreno y algunas propiedades que controlan el nivel de detalle del terreno, como se puede ver en la Figura 31. A continuación se explicará cada una de las propiedades más importantes de los terrenos (terrain):

- A) *Width*:** El ancho en metros del terreno.
- B) *Length*:** La longitud en metros del terreno.
- C) *Height*:** La máxima altura en metros del terreno.

- D) *Heightmap Resolution:*** La resolución del heightmap. Debe tenerse en cuenta que debe ser potencia de 2 + 1. (Ejemplo: 129.513)
- E) *Detail Resolution:*** La resolución del mapa de detalles, cuanto más resolución, más precisión a la hora de dibujar los detalles sobre el terreno y colocar objetos.
- F) *Control Texture Resolution:*** La resolución de las texturas pintadas sobre el terreno, más resolución = más detalle, menor solución = más rendimiento.
- G) *Base Texture Resolution:*** Esta es la resolución base de la textura que se renderiza desde distancia (LOD).

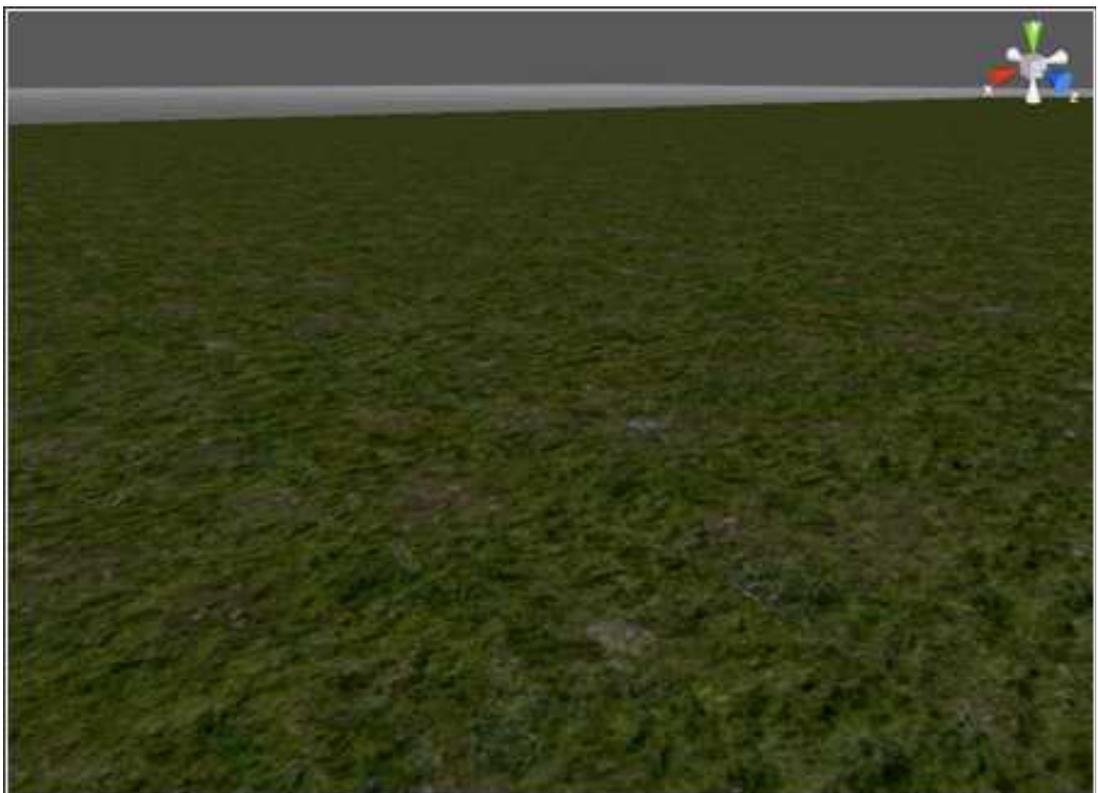


Figura 31. Generador de Terrenos.

4.1.3.3 Seteo del Renderizado (Render Settings)

Unity tiene una DLL para setear el renderizado de los objetos 3D de un juego y añadir efectos especiales a un juego como: a) Niebla (Fog); b) Color de la Niebla (Fog Color); c) Luz de Ambiente (Ambient Light); d) Material de la Caja del Cielo (Skybox Material); e) Fuerza de la Luz (Halo Strength); f) Fuerza del Fuego (Flare Strength); g) Textura de la Luz (Halo Texture); h) Mancha de Galleta (Spot Cookie), como se puede ver en la Figura 32.



Figura 32. Seteo del Renderizado.

4.1.3.4 Controlador de Primera Persona (First Person Controller)

Unity incluye un DLL para el Controlador Estándar en Primera Persona que hace que los juegos con vista en primera persona sean realmente sencillos de configurar. Para manejar este controlador en la escena, se debe ir a la vista de proyecto y seleccionar “Standard Assets -> Prefabs”. Dentro de esa carpeta hay un prefabricado (prefab) llamado “First Person Controller”. Se arrastra este objeto a la vista de escena y se lo posiciona de forma que el cilindro toque el terreno, pero no esté por debajo, como se muestra en la Figura 33.

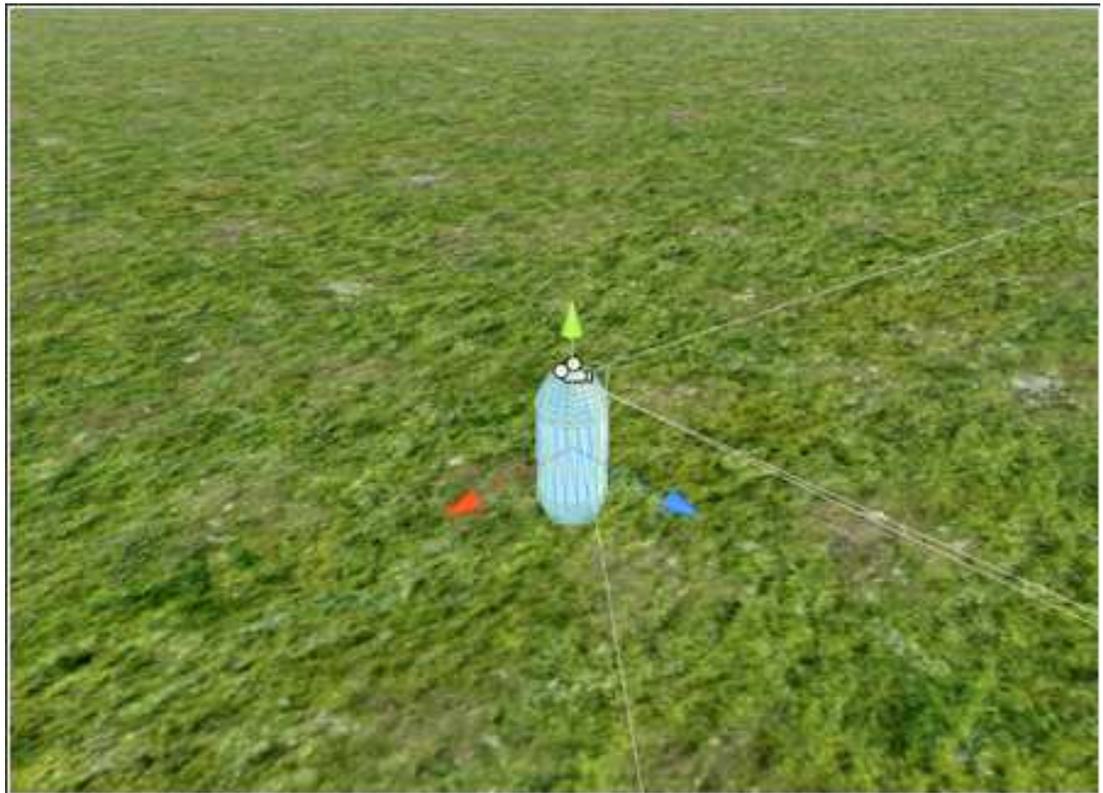


Figura 33. Controlador de Primera Persona.

El controlador FPS puede ser personalizado a través del inspector, donde se pueden modificar propiedades como:

- A) **Speed:** La velocidad de movimiento.
- B) **Jump Speed:** La velocidad de salto.
- C) **Gravity:** La fuerza de la gravedad.
- D) **Height:** La altura del controlador.
- E) **Radius:** El radio del controlador.
- F) **Slope Limit:** Limita el ángulo por el que puede caminar el jugador. Si por ejemplo se indica 30 grados, el jugador no podrá subir por superficies cuyo ángulo respecto al plano será mayor que 30 grados.

4.1.4 Estructura de la interfaz visual del videojuego hecho con Unity



Figura 34. Estructura de la interfaz visual del videojuego hecho con Unity.

La Figura 34 muestra la estructura de la interfaz visual del videojuego hecho con Unity, que como se puede ver cumple con un diseño estético y adecuado para que jueguen los niños.

4.1.4.1. Componentes

Son programas independientes entre sí que poseen sus propias clases dentro del core del Unity. La mayor parte vienen instalados con la aplicación y otros pueden adicionarse.

El grupo de componentes que vienen con la instalación básica del Unity son:

- El componente de Creación de Escenas o **ScenesCreator**.
- El componente de Generación de Terrenos o **TerrainGenerator**.
- El componente del Seteo del Renderizado o **RenderSettings**.
- El componente del Controlador de la Primera Persona o **FirstPersonController**.
- El componente de programación de scripts de Unity conocido como **MonoBehaviour**.
- El componente del juego llamado **GameApp**, creado con el **MonoBehaviour** que se compone por más de 15 clases.

4.2 Construcción del Videojuego

4.2.1 Creación del Modelo 3D en Maya

Maya de Autodesk se utilizó para la creación de los gráficos y objetos, edición de materiales en 3D y la configuración de la iluminación del juego. Los elementos que se diseñaron en Maya para Unity fueron (ver Figura 35):

- El plano del laberinto.
- Los cubos del laberinto que representan las paredes.
- El ratón que representa al usuario en modo manual y a la PC en el modo automático.

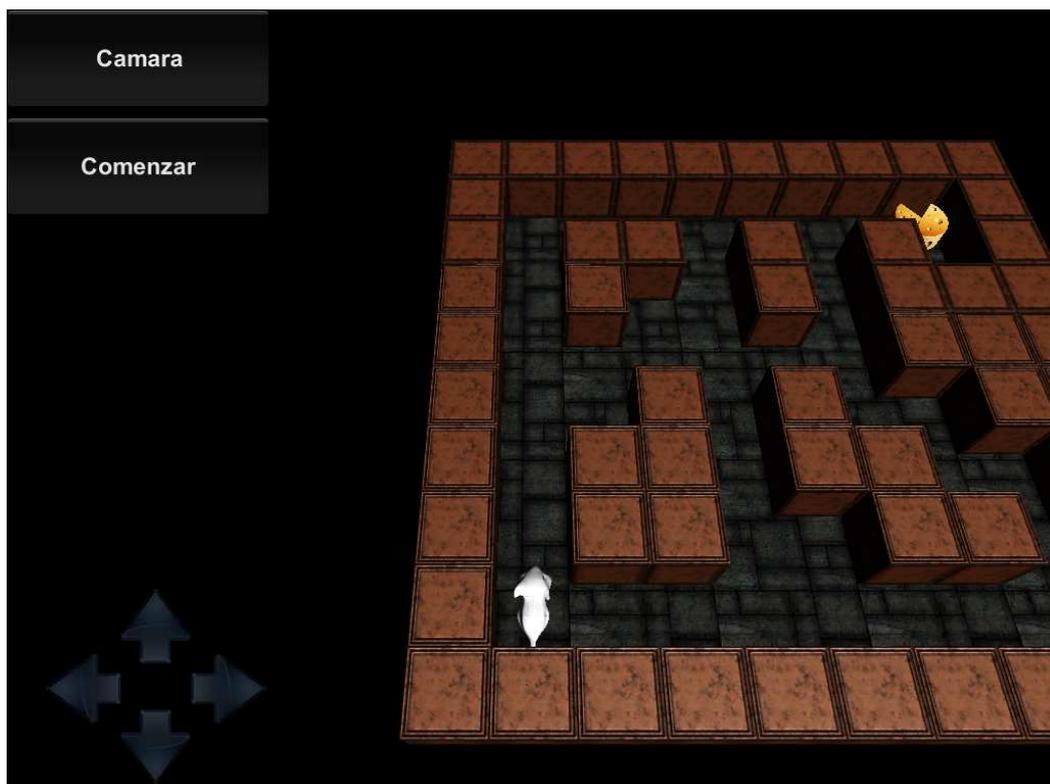


Figura 35. Elementos del Videojuego.

4.2.2 Creación del Videojuego en Unity

El videojuego fue desarrollado enteramente con el motor de juegos Unity y con la herramienta de programación Mono C# para la creación de las clases del juego del componente GameApp creado, las mismas que son:

- **Menu:** Esta clase se utiliza para manejar todos los elementos que conforman el menú principal, como son las opciones de: a) Trial; b) Principiante; c) Intermedio; d) Avanzado; e) Puntajes.
- **Scores:** Esta clase se utiliza para controlar el puntaje del juego obtenido por los usuarios y la PC.
- **GameControl:** Esta clase controla la ejecución del juego independientemente de las opciones de: a) Trial; b) Principiante; c) Intermedio; d) Avanzado.
- **FarCamera:** Esta clase permite visualizar todo el laberinto a través de una cámara.
- **FollowCamera:** Esta clase permite que una de las cámaras siga al ratón que está en el laberinto.
- **Loading:** Esta clase permite manejar la ventana que carga el juego y opera un buffer de comunicación que calcula el porcentaje de carga del mismo.
- **GameMode:** Esta clase controla el tipo de juego que puede ser manual, donde el que opera el juego es un usuario; o también puede ser automático, donde el que opera el juego es la PC con IA. Además esta clase se encarga de almacenar los estados del juego como son el nombre del usuario, el nivel de dificultad (principiante, intermedio, avanzado) y el tiempo que se demora en completar el laberinto.

- **MazeAlgorithm:** Esta clase se encarga de realizar todas las operaciones matemáticas y generar con IA el mapa del laberinto, lo cual permite obtener una matriz de unos y ceros, donde los ceros representan el camino a seguir y los unos representan las paredes del laberinto.
- **MazeCreator:** Esta clase se encarga de pasar del entorno matemático a un entorno virtual en 3D, para lo cual utiliza la clase **MazeAlgorithm** para operar el laberinto.
- **MouseBrain:** Esta clase se encarga de controlar el movimiento del ratón en modo automático, donde se utiliza IA para analizar la ruta a seguir y llegar a la meta que consiste en encontrar el queso. Esta clase utiliza parte de los algoritmos de la clase **MazeAlgorithm** para moverse por el camino representado por un grupo de ceros y almacena en una lista enlazada todas las posiciones recorridas basada en la regla de la mano izquierda.
- **PlayerController:** Esta clase controla manualmente el movimiento del ratón por el laberinto.
- **ParticleTester:** Esta clase se encarga de manejar un sistema de partículas para generar los efectos de explosión que tiene el juego y que se produce cuando el usuario representado por un pequeño ratón, encuentra el queso en el laberinto.
- **GUICamera:** Esta clase se encarga de controlar la interfaz gráfica del usuario en 2D, donde consta el menú y sus opciones.
- **CameraController:** Esta clase se encarga de administrar las dos cámaras del juego.

- ***XMLDataBase***: Esta clase se utiliza para manejar los archivos planos del juego donde se almacena los datos del juego.

Adicionalmente se utilizó tres componentes básicos del motor de juegos Unity, los mismos que son:

- ***Componente CoreUnity***: Este componente es el centro de creación de los aplicativos del Unity, ya que se constituye en el alma del Game Engine del Unity, donde se encuentran todas las clases que manejan toda la computación gráfica del juego como: vectores, matrices, imágenes, texturas, color, fuentes, objetos 2D y 3D, luces, cámaras, animaciones, física y transformación de objetos.
- ***Componente MonoBehaviour***: Este componente es un conjunto de DLLs implementados por el proyecto Mono C# para Unity que es compatible con C# .NET y es el que se encarga de compilar el programa con las librerías del Unity y manipular los objetos mediante scripts de programación.
- ***Componente GUI***: Este componente es el que permite crear las interfaces gráficas del usuario en 2D como los componentes para formularios tales como: botones, etiquetas, sliders, etc.

Finalmente se compila el programa para Windows y Dispositivos Móviles, generándose un archivo ejecutable y un applet respectivamente, que permite correr la aplicación con todos sus elementos.

4.3 Desarrollo e Implantación de Componentes para la el Laberinto3D

Todos los componentes con sus clases poseen la siguiente estructura de archivos:

- Un archivo donde constan todos los datos y funciones miembro de la clase, con al menos un constructor (constructor por defecto).
- Archivos de inicialización del modo gráfico en Unity y para la creación de objetos propios del motor de juegos.
- Un archivo .xml para mapear todo el código fuente, las imágenes y los archivos planos de datos cuando se lo requiera.



Figura 36. Ventana de la Aplicación del Juego en ambiente Windows.

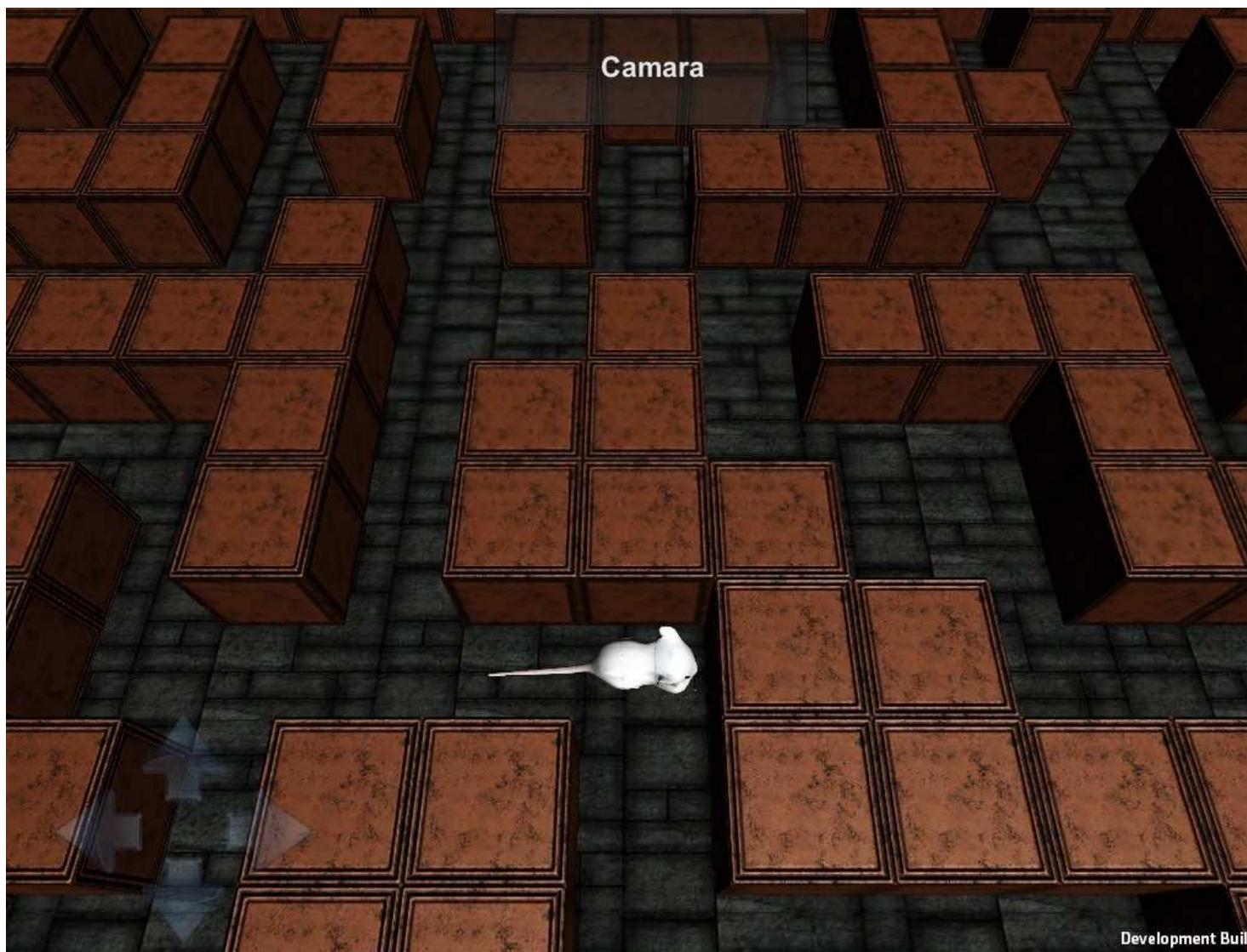


Figura 37. Ventana de la Aplicación del Juego en ambiente Móvil.

4.4 Pruebas de la Aplicación

Se ha establecido el siguiente proceso de pruebas que inicia con la verificación de una adecuada navegabilidad, estándar de colores, tamaño y tipo de letra; para finalizar con la verificación de infraestructura y seguridad de la aplicación. (Ver Figura 38).

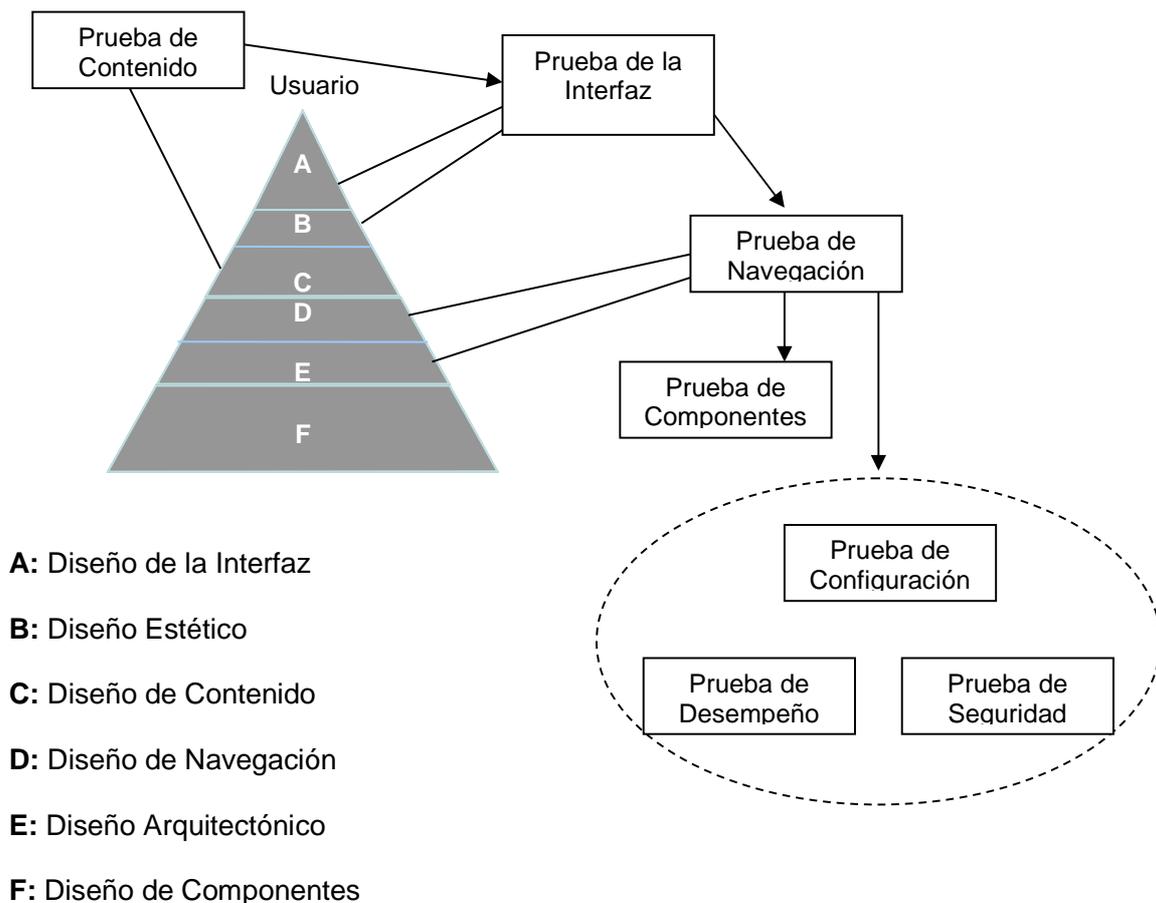


Figura 38. Proceso de Prueba²⁴

Se ha realizado las siguientes pruebas con la finalidad de resolver todos los posibles conflictos de conectividad, desempeño y navegabilidad de la aplicación; y son:

²⁴ Figura tomada del Capítulo 20 “Cómo Probar Aplicaciones Web” – Ingeniería Web – Pressman

4.5.1 Prueba de Contenido

Se ha realizado con varios niños (usuarios) una revisión minuciosa de los siguientes tipos de contenido:

- Estático: Referente a la información estática que se muestra en la aplicación.
- Dinámico: Referente a la información encontrada en los archivos planos de datos que está integrada a la aplicación tanto para Windows, para la Web y para Dispositivos Móviles; que en este caso es del 100% por el hecho de utilizar el Game Engine Unity.

A través de la técnica de observación y lectura de todos los elementos y enlaces de la aplicación y sin olvidar el correcto agrupamiento de los temas a ser mostrados; se ha logrado corregir los 4 aspectos más relevantes de los contenidos que son:

- Errores tipográficos y/o equívocos gramaticales.
- Errores semánticos (información incompleta o ambigua)
- Errores en la organización de la información para ser mostrada al usuario final.
- Errores de obtención de información incorrecta de los archivos planos de datos, al momento de ser desplegada en las diferentes ventanas de la aplicación.

4.5.2 Prueba de Interfaz de Usuario

Se ha tomado en cuenta las siguientes consideraciones:

Prueba de Mecanismos de la Interfaz

- A) Enlaces:** Cada uno de los enlaces de la aplicación, sean estos enlaces internos o externos a la aplicación, se enlacen al objeto deseado.
- B) Formato:** El computador o el dispositivo móvil recibe toda la información y no existe pérdida de datos en la ejecución de la aplicación, los campos del formato tienen el ancho y tipos de datos adecuados.
- C) Ventanas Dinámicas:** La navegación en cada una de las ventanas de la aplicación maneja memoria dinámica para construir y destruir los objetos que tiene cada una de estas con los scripts de programación hechos en Mono C# para garantizar su correcto despliegue.

Prueba de Facilidad de Uso

Para este tipo de pruebas se ha tomado en consideración aspectos como: grado de usabilidad (es fácil encontrar lo que se busca), interacción con el usuario (menús desplegables, botones, estética (colores), despliegue (resolución de la pantalla).

4.5.3 Prueba de Navegación

Para la realización de esta prueba se ha establecido una verificación de todos los enlaces de la aplicación.

Se ha analizado junto a los usuarios que los enlaces creados lleven hacia el contenido o la funcionalidad adecuada y sobretodo que estos enlaces sean comprensibles conforme se realiza la navegación.

Además se ha verificado que los nombres de los nodos sean significativos para los usuarios, como por ejemplo: a) Menú Principal; b) Modo Manual; c) Modo Automático; d) Trial; e) Nivel Principiante; f) Nivel Intermedio; g) Nivel Avanzado; h) Puntajes del Juego. Con esta prueba se ha logrado ejercitar ampliamente la navegación de la aplicación por parte de los desarrolladores y de los usuarios finales.

4.5.4 Prueba de Componentes

Las pruebas a nivel de componente se las ha realizado primeramente integrando el componente desarrollado en el juego que es el **GameApp** sin ningún inconveniente en su creación. Se ha realizado pruebas en función de la entrada de datos en formularios, definiendo los tipos de datos permitidos para cada uno de los campos de entrada, verificando que exista una correcta validación de la información, que la información se envíe y se reciba desde los archivos planos hasta la aplicación y que no exista pérdida de información durante la ejecución del programa.

Se evaluó la consistencia de la información en los archivos planos de datos, verificando un correcto almacenamiento.

4.5.5 Prueba de Configuración

Se ha analizado la arquitectura Cliente/Servidor que maneja Unity para crear aplicaciones de tipo Desktop para Windows y para Dispositivos Móviles, donde se ha especificado lo siguiente:

- Se ha instalado la versión de Unity y configurado para que funcione en modo local y no distribuido.
- Se ha generado un archivo ejecutable que corra bajo Windows y un applet para que corra en Dispositivos Móviles.

4.5.6 Prueba de Seguridad

La aplicación no implementa ninguna seguridad de acceso, por lo que puede cualquier usuario acceder a la aplicación. Además la aplicación puede ser copiada e instalada en cualquier computadora con el sistema operativo Windows y con el sistema operativo Android para dispositivos móviles y no tendrá ningún problema de funcionamiento.

4.5.7 Prueba de Desempeño

Con esta prueba se determinó cómo la aplicación y su entorno de trabajo bajo una arquitectura Cliente / Servidor respondió a varias condiciones de carga y soportó adecuadamente las transacciones en un ambiente local tipo desktop y móvil, con un solo usuario a la vez.

4.6 Deployment de la Aplicación

Para la puesta en producción de la aplicación es necesario tener lo siguiente:

- Compilar la aplicación con todos los componentes necesarios para que funcione adecuadamente tanto en ambiente Windows como en ambiente Android.
- Generar un archivo ejecutable de la aplicación para instalar y correr en cualquier computadora el aplicativo con el sistema operativo Windows y con el sistema operativo Android.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

5.1.1. Conclusiones del Proyecto

- Para la creación del juego se utilizó el game engine Unity el cual utiliza un conjunto de componentes para manejo de gráficos y de la física de objetos en tres dimensiones.
- Las pruebas de contenido, función, estructura, facilidad de uso, navegabilidad, y desempeño; ayudaron a detectar y corregir los errores antes de la puesta en producción del aplicativo en 3D.
- Para el presente trabajo en función de los requerimientos del videojuego, el game engine Unity cubrió por sí solo el 65%, el 35% fue adaptación y desarrollo propio. En esta conclusión se considera como medida del software el número de componentes que se utilizaron para desarrollar esta aplicación. Unity aportó con 3 componentes (CoreUnity, MonoBehaviour, GUI). Se adaptó y se creó un componente (GameApp).
- La técnica de inteligencia artificial de planificación permitió generar automáticamente los laberintos según el nivel de dificultad y la técnica de

inteligencia artificial de los sistemas de acción y reacción basados en el algoritmo de la mano derecha cuya aplicación permitió resolver los laberintos de forma automática por el avatar de la computadora.

- El videojuego fue implementado tanto sobre la plataforma Android como sobre la plataforma Windows. En la plataforma Android el tiempo de carga de los laberintos generados con IA es mucho mayor que en la plataforma Windows en una proporción de 10 a 1 y esto se debe a que la capacidad de procesamiento de una tableta es mucho menor que la de una PC.

- En la presente aplicación de un videojuego en 3D se utilizó la metodología OOADM con UML, debido a que es una metodología orientada al diseño y desarrollo de aplicaciones multimedia tipo desktop, web y móviles, aportando con diagramas útiles y prácticos que permiten llevar un proceso de desarrollo organizado y eficiente.

5.1.2. Conclusiones Académicas

- Con respecto a la parte investigativa se concluye que para alcanzar el objetivo propuesto, fue necesario adquirir nuevos conocimientos de los ya recibidos en la Universidad, debido a que el desarrollo de aplicaciones multimedia enfocadas a los videojuegos tienen un alto grado de complejidad y que gracias al aporte de los motores de juegos, este desarrollo se hace más simple y

práctico, lo que ha hecho que se haya creado una nueva industria a nivel mundial que es la de la Industria de los Videojuegos.

- Los motores de juegos (game engines), como las herramientas de diseño y modelado para 3D y las herramientas de programación orientadas a objetos se han ido fortaleciendo, tanto en sus arquitecturas como en sus funcionalidades en estos últimos años.
- Tanto Los motores de juegos como las lenguajes de programación orientados a objetos, constituyen las nuevas herramientas a ser investigadas por sus funcionalidades diversas, calidad y sobretodo porque abaratan costos de instalación, desarrollo e implementación de soluciones de entretenimiento.

5.2. Recomendaciones

5.2.1. Recomendaciones del Proyecto

- Realizar una adecuada configuración del motor de juegos (game engine) Unity, ya que existe una gran variedad de módulos para implementar nuevas funcionalidades a los juegos en 3D como la física y el movimiento de objetos utilizando inteligencia artificial y redes neuronales entre algunos de los módulos que se pueden encontrar en el mercado para Unity.
- Para la adecuada selección de componentes adicionales a ser instalados en el motor de juegos Unity, se recomienda escoger los que tengan mayores

comentarios positivos en los foros y con mayores votaciones, ya que serían los más calificados por su estabilidad y funcionalidad.

- En todo proyecto desktop, Web o móvil, se debería definir una fase de pruebas que podría ser integrada con el desarrollo del proyecto o como un documento separado para controlar la calidad del producto desde sus inicios de creación.
- Utilizar el motor de juegos Unity cuando cubra hasta más de un 50% de los requerimientos solicitados por los usuarios, ya que con su gran variedad de extensiones, facilidad de uso y documentación actualizada se pueden adaptar y crear las funcionalidades adicionales.
- Que se formen grupos de trabajo de al menos dos personas o de pares para que uno de ellos se centre al diseño y modelado en 3D, mientras que el otro se dedique al proceso de programación para acortar los tiempos de desarrollo e implementación de los videojuegos. Además se recomienda que tengan conocimientos básicos de programación orientada a objetos, de animación e IA, para que en el proceso de creación de videojuegos adquieran los conocimientos extras necesarios para cumplir con este objetivo.
- Utilizar el motor de juegos Unity para la creación y mantenimiento de videojuegos, ya que no solo se ahorra tiempo y dinero, sino que se logra una verdadera concepción de las tareas de un desarrollador de aplicativos de simulación 3D tal como ciertos autores lo consideran al videojuego, el cual tendrá todas las facilidades para: expandir y mejorar la estética, navegabilidad y servicios de entretenimiento que es el meollo de un videojuego.

5.2.2. Recomendaciones Académicas

- Que la Universidad forme equipos de investigación entre docentes y estudiantes para manejar, personalizar y adaptar motores de juegos propietarios y libres con lenguajes de programación orientados a objetos, e incluso se debería crear foros de discusión en la página del DECC para que los estudiantes participen con sus dudas y respuestas.

- Que dentro de la malla curricular de la carrera de Ingeniería de Sistemas e Informática, se incluyan tópicos avanzados de programación para videojuegos enfocados a los dispositivos móviles que es la tendencia actual en el uso masivo de aparatos electrónicos para la comunicación y el entretenimiento, así como el uso e investigación de la inteligencia artificial en los videojuegos donde participen estudiantes y docentes.

BIBLIOGRAFÍA Y WEBGRAFÍA

- Arocena, F. (Abril de 2003). Obtenido de Crea tu Página Web:
<http://wmaestro.com/webmaestro/docs/portada.html>
- Blackman, S. (2011). *Beginning 3D Game Development with Unity*. New York – USA:
Apress.
- Consejos de estilo gráfico para WWW*. (2006). Obtenido de
<http://dmi.uib.es/people/acoca/estilo/index.htmml>
- García Rubio, R., Quirós Suárez, J., González Santiago, M., Santos Gallegos, R., &
Fernández Morán, S. (2006). *Diseño Gráfico de Contenidos para Internet*.
Madrid: Prentice Hall.
- Keller, E., Palamar, T., & Honn, A. (2010). *Mastering Autodesk Maya 201*,. USA:
Sybex.
- Lamarca, M. J. (2006). *Hipermedia/Multimedia*. Madrid España: Universidad
Complutense de Madrid .
- Página Oficial de Maya*. (2012). Obtenido de <http://usa.autodesk.com/maya/>
- Pressman, R. (2002). *Ingeniería de Software Un Enfoque Práctico, II*. Madrid España:
Mc. Graw Hill.
- SCHNEIDER Electric Instituto. (2008). *Manual de Formación Unity Pro*. Barcelona –
España: Instituto Schneider Electric.
- Unity. (2012). Obtenido de Página Oficial de Unity (2012): <http://unity3d.com/>

ANEXOS

Anexo 1. Diagramas De Secuencia Por Actores

ACTOR: USUARIO - JUGADOR

1. USR-JUG-LABERINTO3D-01: Configurar el videojuego. (Ver Figura 39).
2. USR-JUG- LABERINTO3D -02: Resolver el Laberinto de Modo Manual.
(Ver Figura 40).
3. USR-JUG- LABERINTO3D -03: Resolver el Laberinto de Modo Automático.
(Ver Figura 41).

1. USR-JUG-LABERINTO-01: Configurar el videojuego.

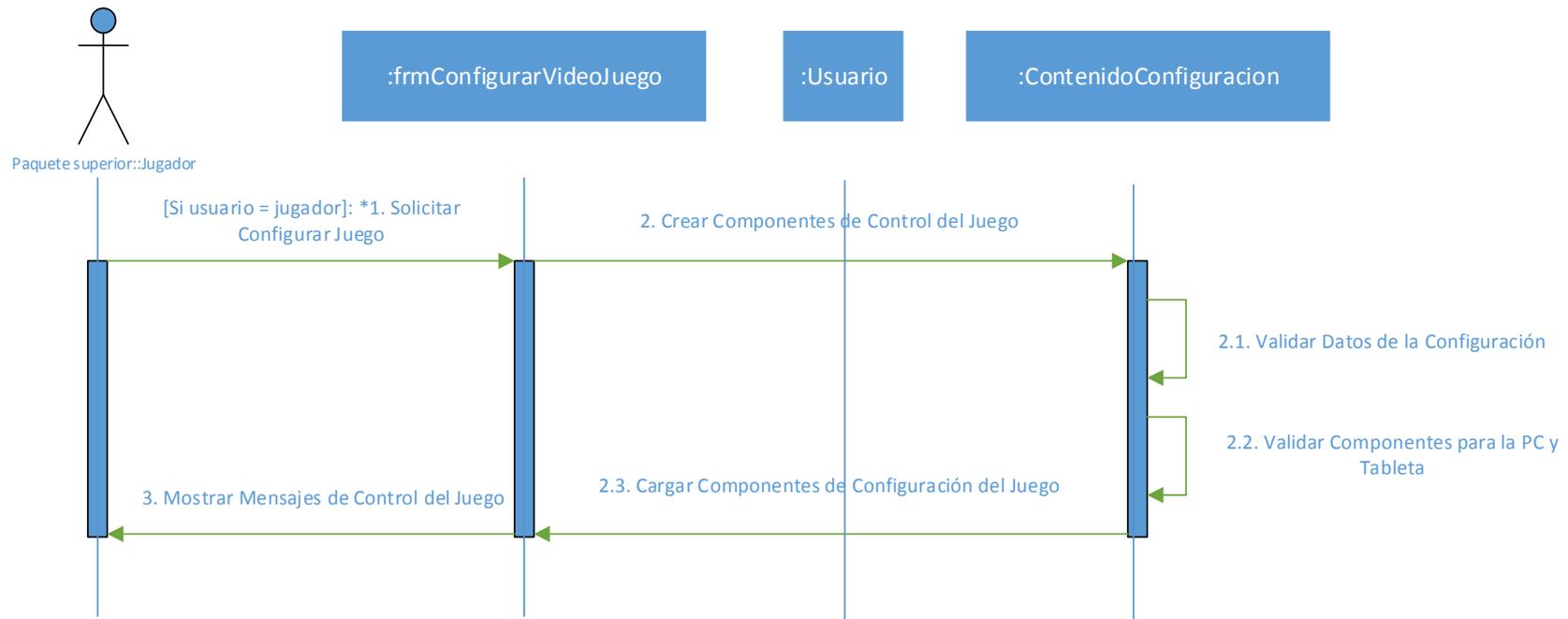


Figura 39. Diagrama de Secuencia – Configurar el Videojuego.

2. USR-JUG- LABERINTO-02: Resolver el Laberinto de Modo manual.

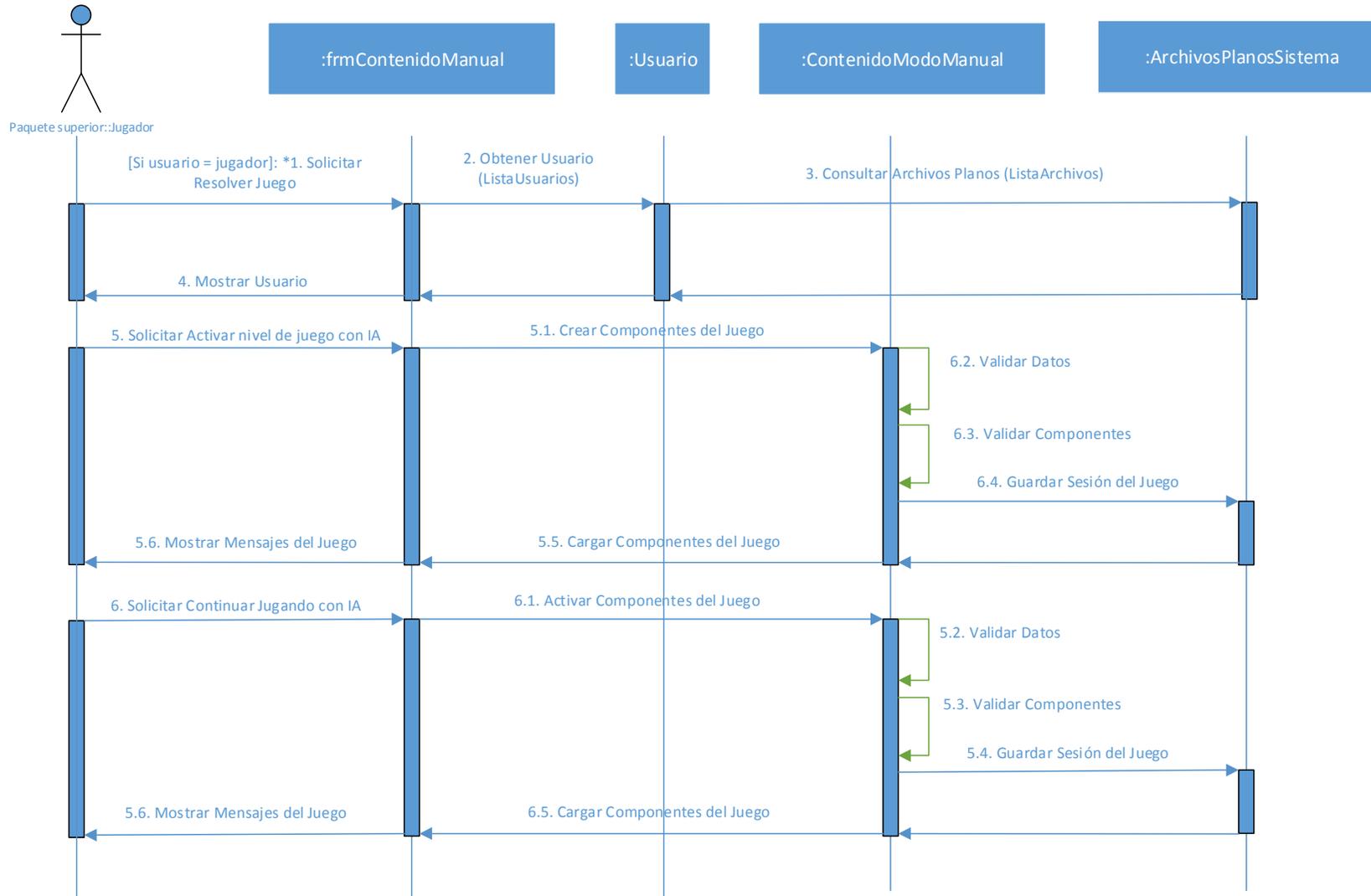


Figura 40. Diagrama de Secuencia – Resolver el Laberinto de Modo Manual.

3. USR-JUG- LABERINTO-03: Resolver el Laberinto de Modo Automático.

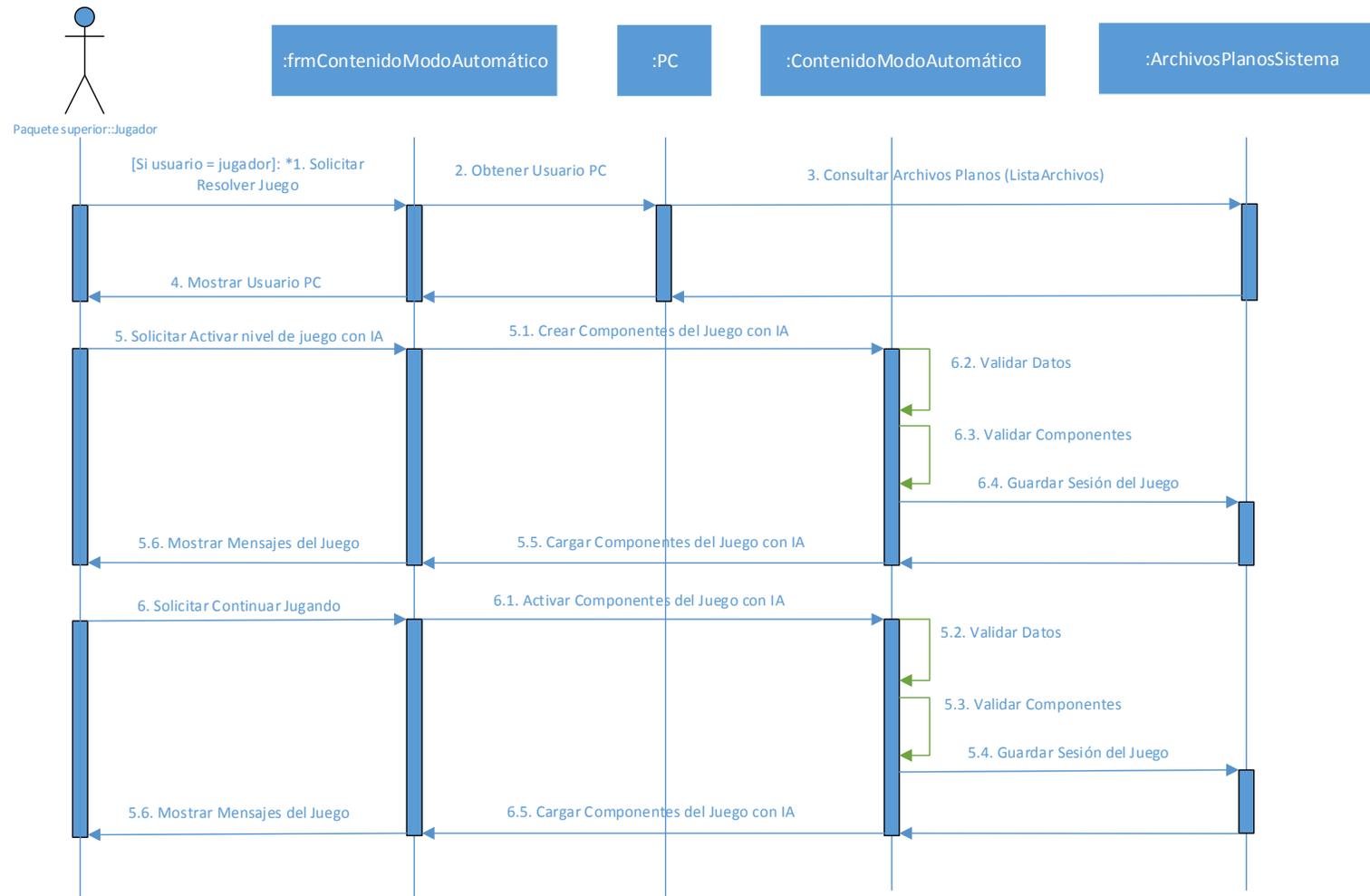


Figura 41. Diagrama de Secuencia – Resolver el Laberinto de Modo Automático.

Anexo 2. Manual de Usuario

Laberinto3D- Juego Didáctico de Razonamiento Lógico en 3D

Ventana de Configuración del Juego

En la Figura 42 se muestra la ventana de configuración del videojuego, donde el usuario selecciona la resolución de la pantalla para el juego.

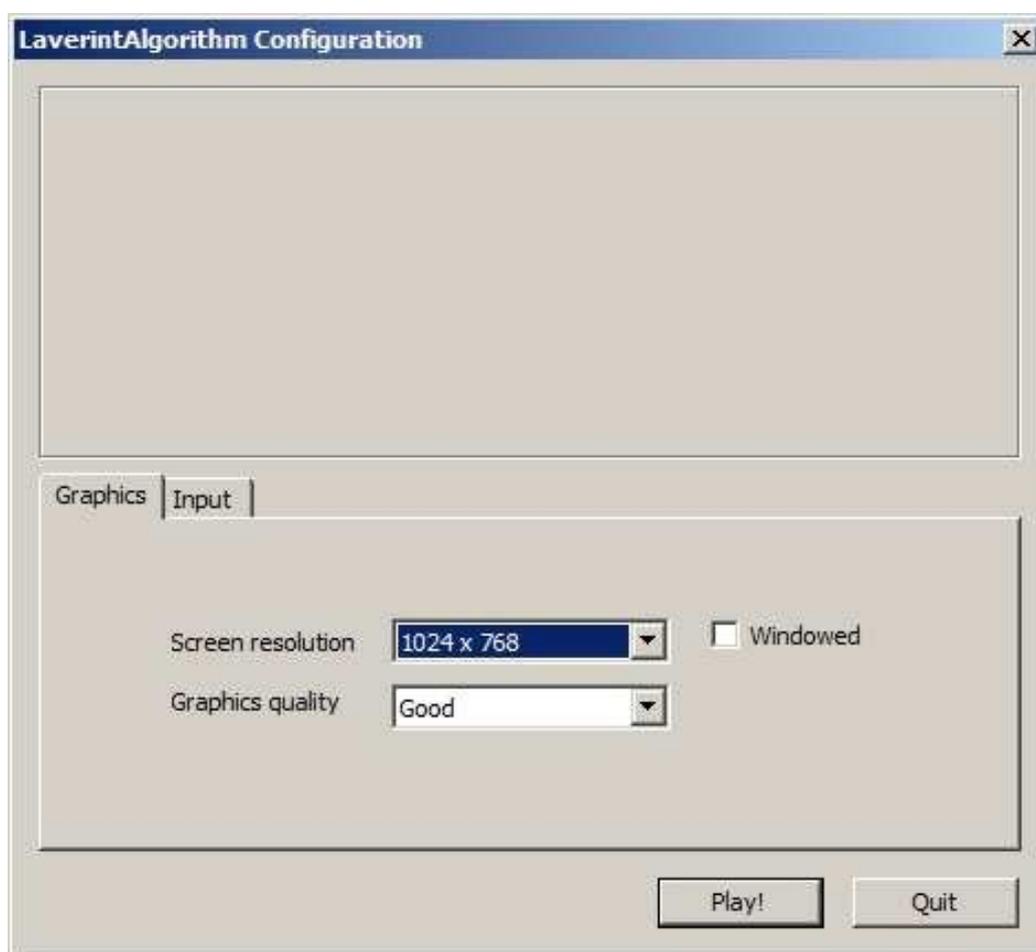


Figura 42. Ventana del Menú Principal.

Ventana del Menú Principal de la Opción Manual

En la Figura 43 se muestra la pantalla del menú principal de la opción manual que consta de cinco opciones: a) Trial; b) Principiante; c) Intermedio; d) Avanzado; e) Puntajes. Esta opción permite que el usuario se registre en el juego insertando su nombre en una caja de texto.



Figura 43. Ventana del Menú Principal de la Opción Manual.

Ventana de la Opción Trial Manual

En la Figura 44 se muestra la pantalla de la opción Trial Manual de la opción Manual del Juego, donde se puede seleccionar una opción a la vez.

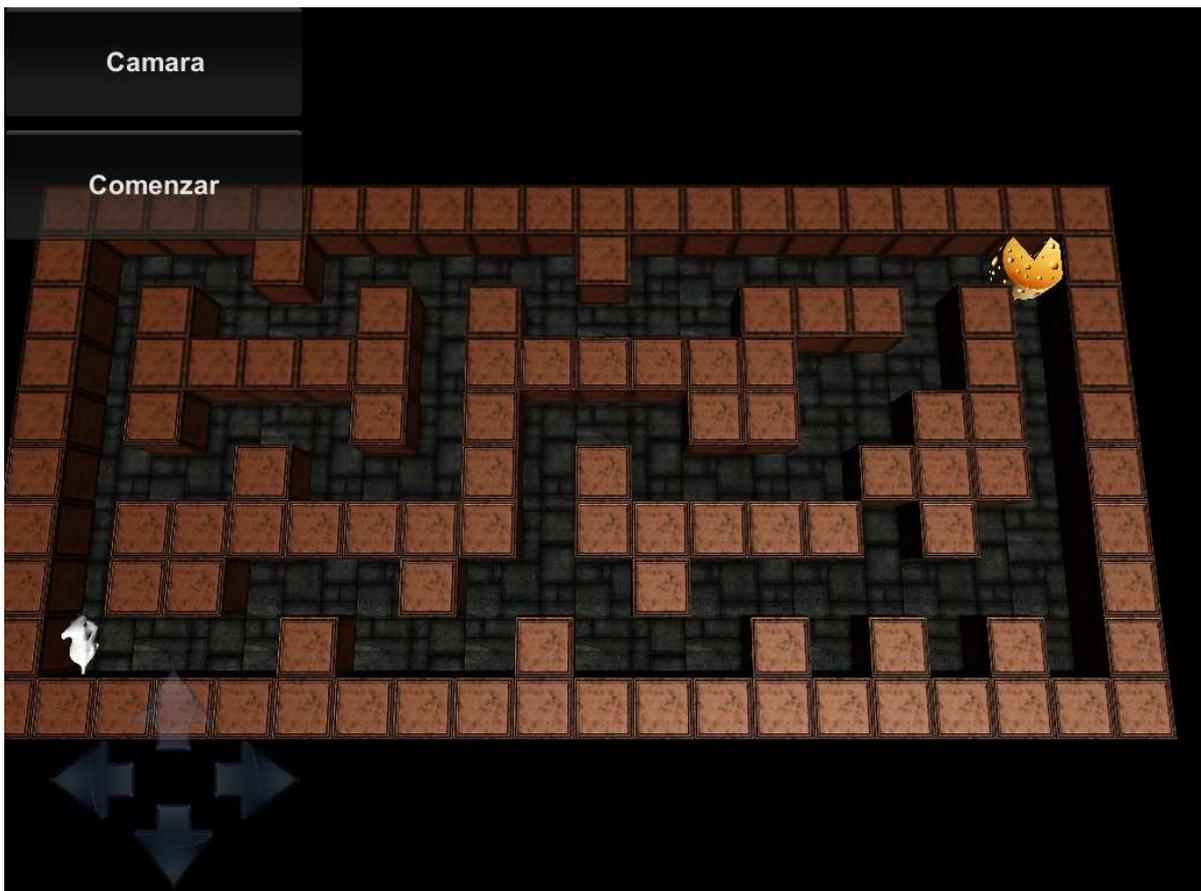


Figura 44. Ventana de la Opción Trial Manual del Juego.

Ventana de la Opción Principiante Manual

En la Figura 45 se muestra la pantalla de la opción Principiante Manual de la opción Manual del Juego, donde se puede seleccionar una opción a la vez.

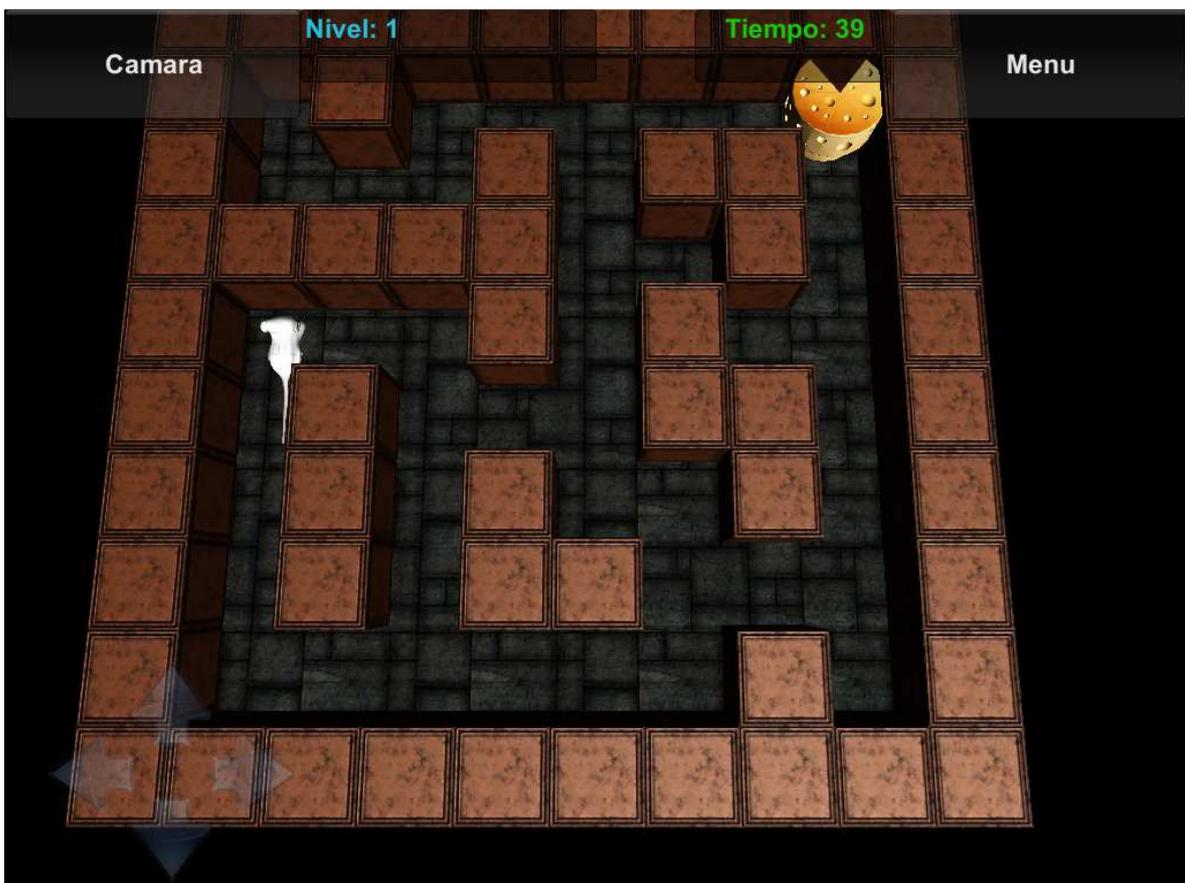


Figura 45. Ventana de la Opción Principiante Manual del Juego.

Ventana de la Opción Intermedio Manual

En la Figura 46 se muestra la pantalla de la opción Intermedio Manual de la opción Manual del Juego, donde se puede seleccionar una opción a la vez.

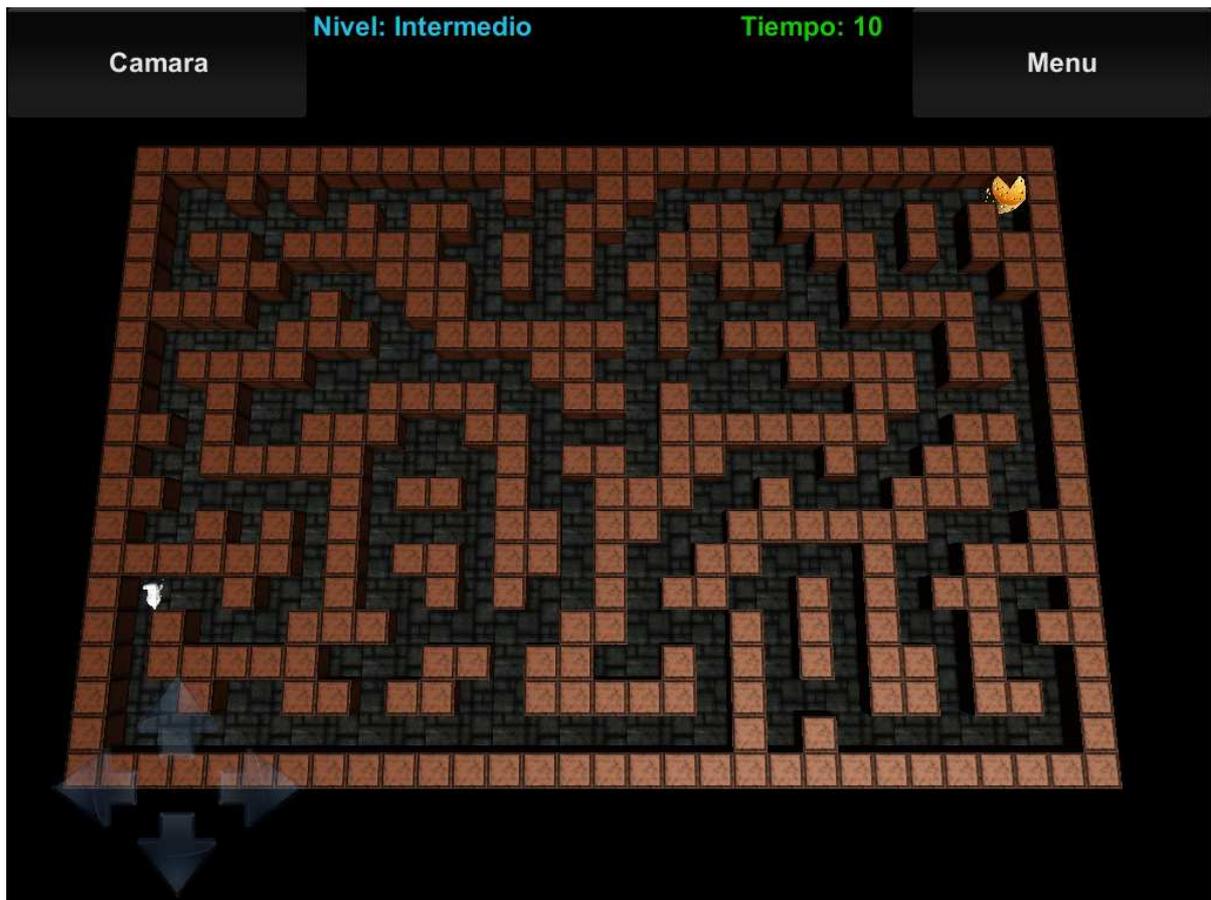


Figura 46. Ventana de la Opción Intermedio Manual del Juego.

Ventana de la Opción Avanzado Manual

En la Figura 47 se muestra la pantalla de la opción Avanzado Manual de la opción Manual del Juego, donde se puede seleccionar una opción a la vez.

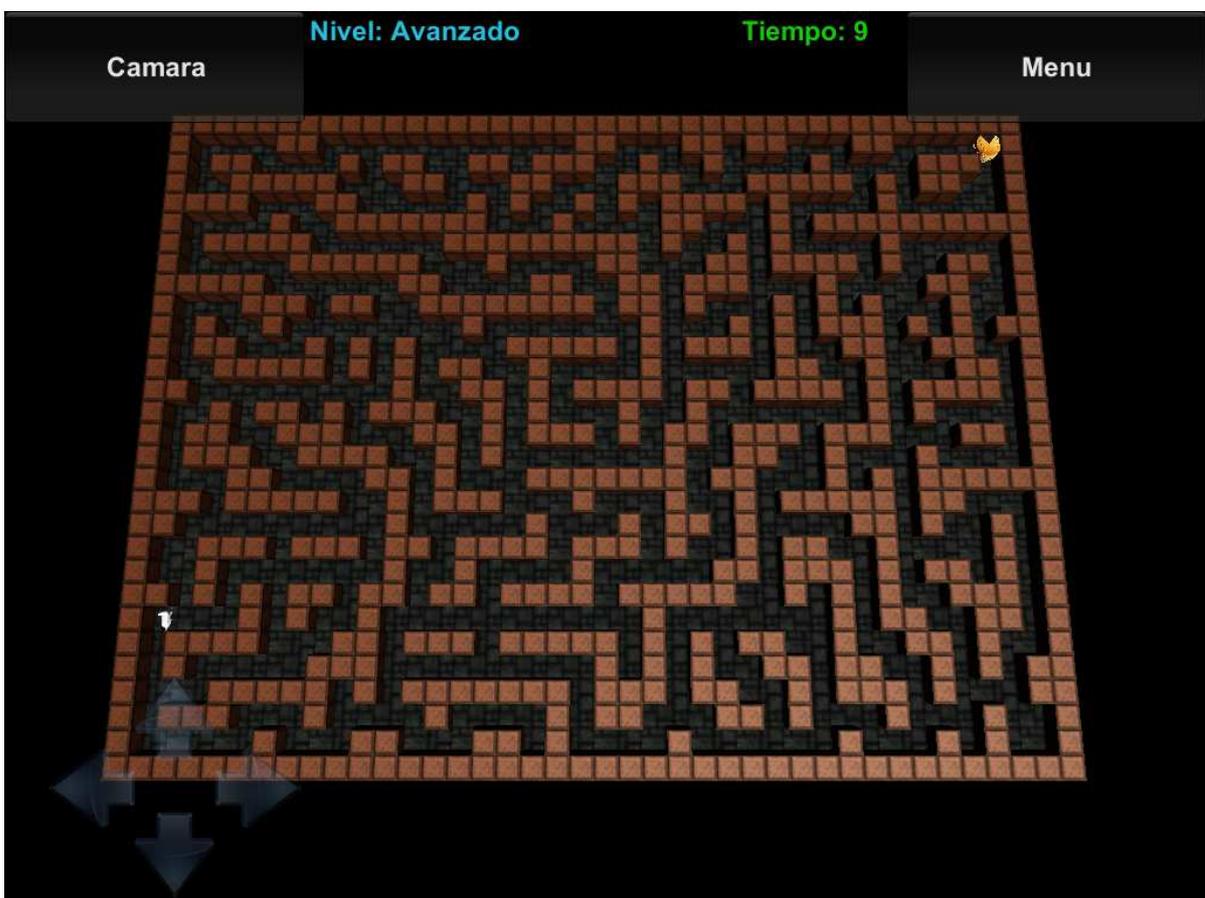


Figura 47. Ventana de la Opción Avanzado Manual del Juego.

Ventana de Puntajes Manual

En la Figura 48 se muestra la pantalla de los puntajes obtenidos por los diferentes usuarios que han jugado el juego manualmente.



	Nombre	Dificultad	Tiempo
1	Soledad	Principiante	34 seg
2	Soledad	Intermedio	40 seg
3	Soledad	Avanzado	52 seg
4	Cristina	Principiante	57 seg
5	Cristina	Intermedio	119 seg
6	Cristina	Avanzado	94 seg
7	Daniela	Principiante	63 seg
8	Daniela	Intermedio	72 seg
9	Daniela	Avanzado	129 seg

Menu

Figura 48.Ventana del Puntaje de la Opción Manual.

Ventana del Menú Principal de la Opción Automática

En la Figura 49 se muestra la pantalla del menú principal de la opción automática que consta de cinco opciones: a) Trial; b) Principiante; c) Intermedio; d) Avanzado; e) Puntajes. Esta opción permite que la computadora resuelva los laberintos con IA.



Figura 49. Ventana del Menú Principal de la Opción Automática.

Ventana de la Opción Trial Automática

En la Figura 50 se muestra la pantalla de la opción Trial Automática de la opción Automática del Juego, donde se puede seleccionar una opción a la vez. Esta opción le permite al usuario visualizar diferentes laberintos desde un pequeño hasta un grande, que los resuelve la computadora con IA.

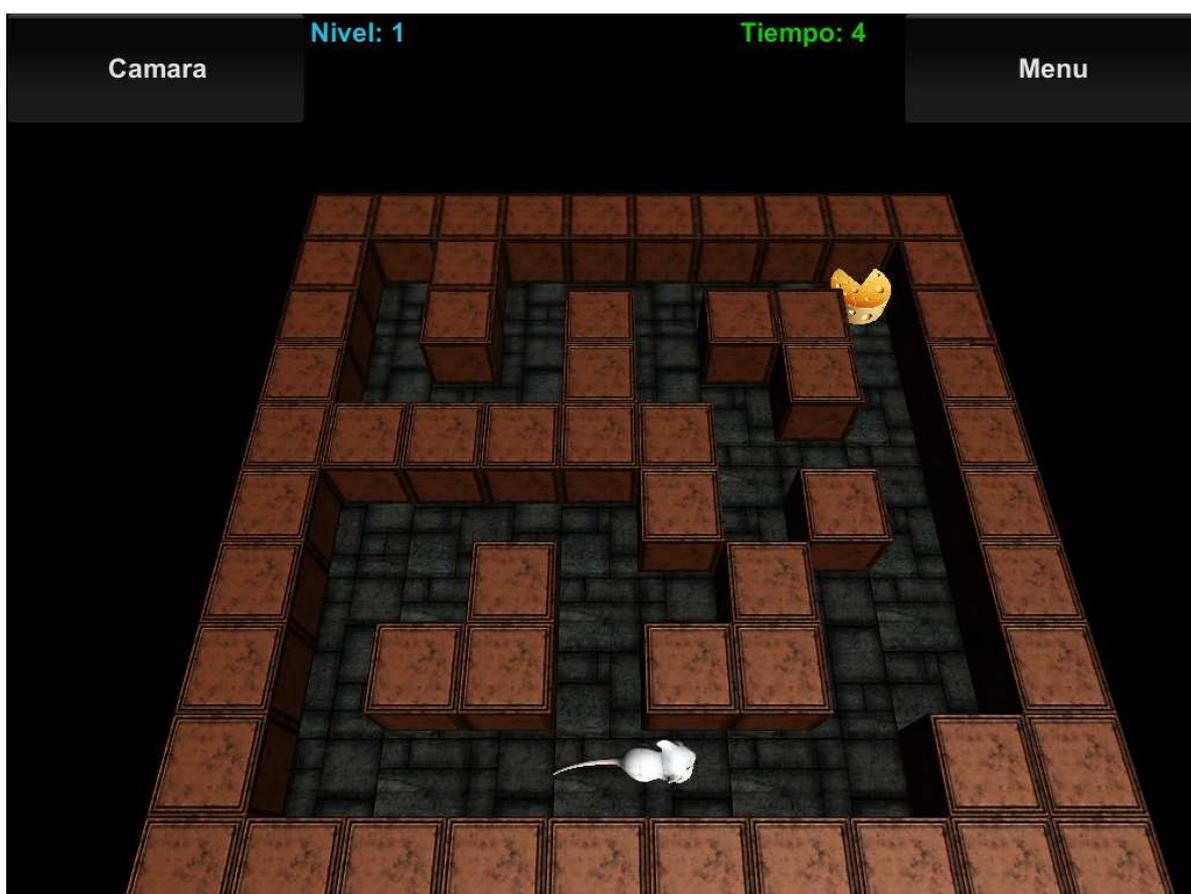


Figura 50. Ventana de la Opción Trial Automática del Juego.

Ventana de la Opción Principiante Automática

En la Figura 51 se muestra la pantalla de la opción Principiante Automática de la opción Automática del Juego, donde se puede seleccionar una opción a la vez. Esta opción le permite al usuario visualizar un laberinto pequeño que lo resuelve la computadora con IA.

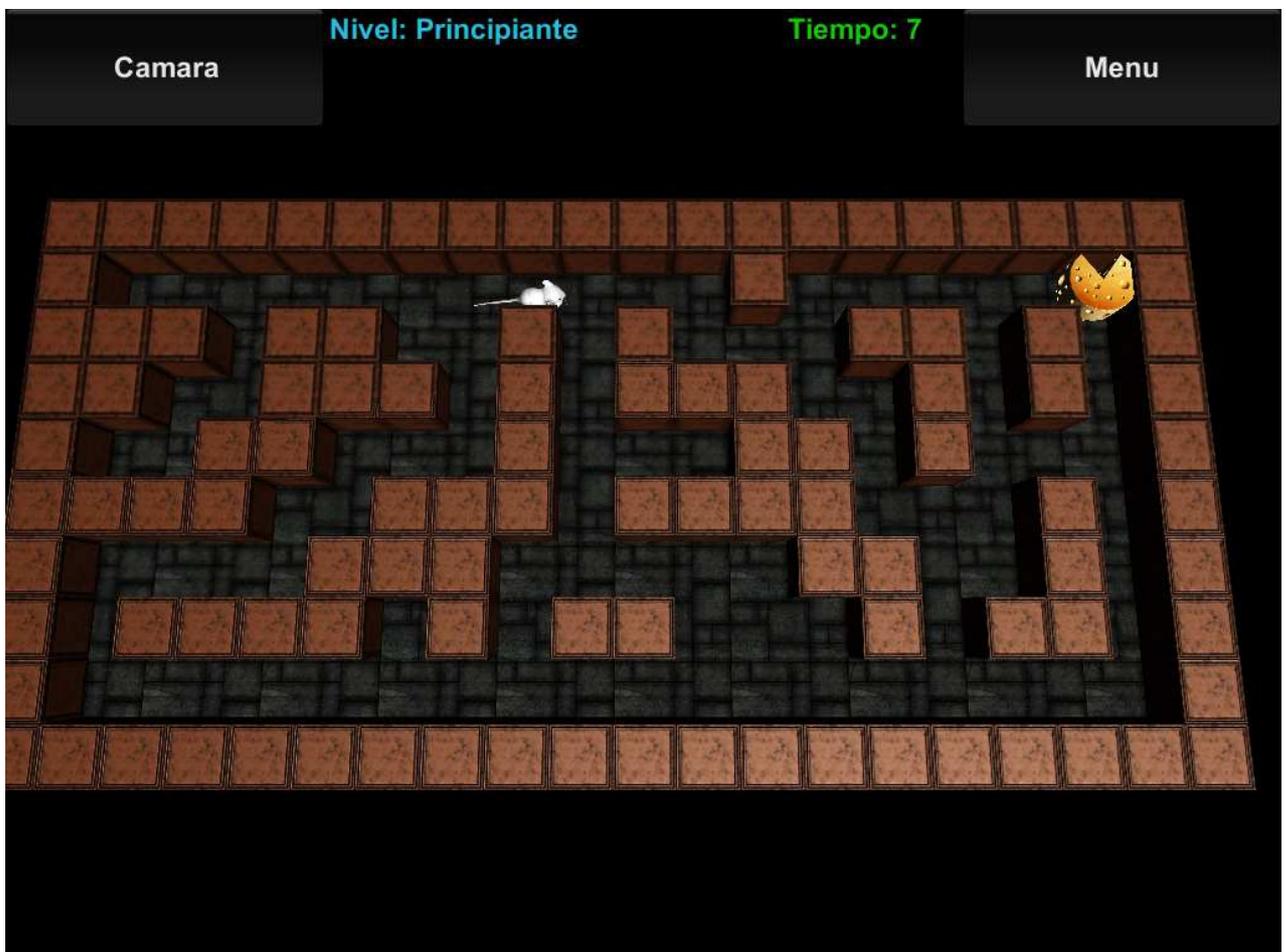


Figura 51. Ventana de la Opción Principiante Automática del Juego.

Ventana de la Opción Intermedio Automática

En la Figura 52 se muestra la pantalla de la opción Intermedio Automática de la opción Automática del Juego, donde se puede seleccionar una opción a la vez. Esta opción le permite al usuario visualizar un laberinto de tamaño mediano que lo resuelve la computadora con IA.

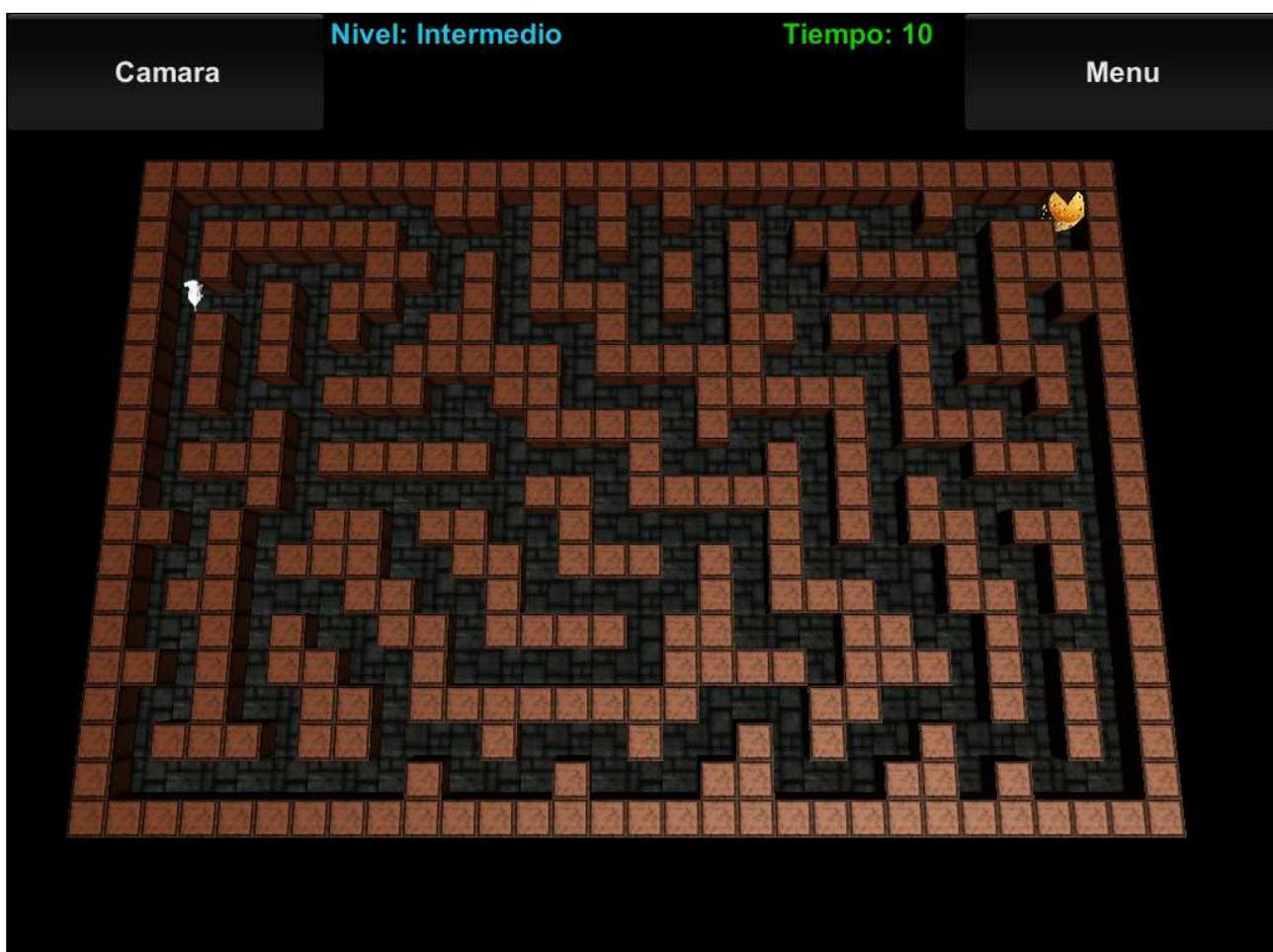


Figura 52. Ventana de la Opción Intermedio Automática del Juego.

Ventana de la Opción Avanzada Automática

En la Figura 53 se muestra la pantalla de la opción Avanzada Automática de la opción Automática del Juego, donde se puede seleccionar una opción a la vez. Esta opción le permite al usuario visualizar un laberinto de tamaño grande que lo resuelve la computadora con IA.

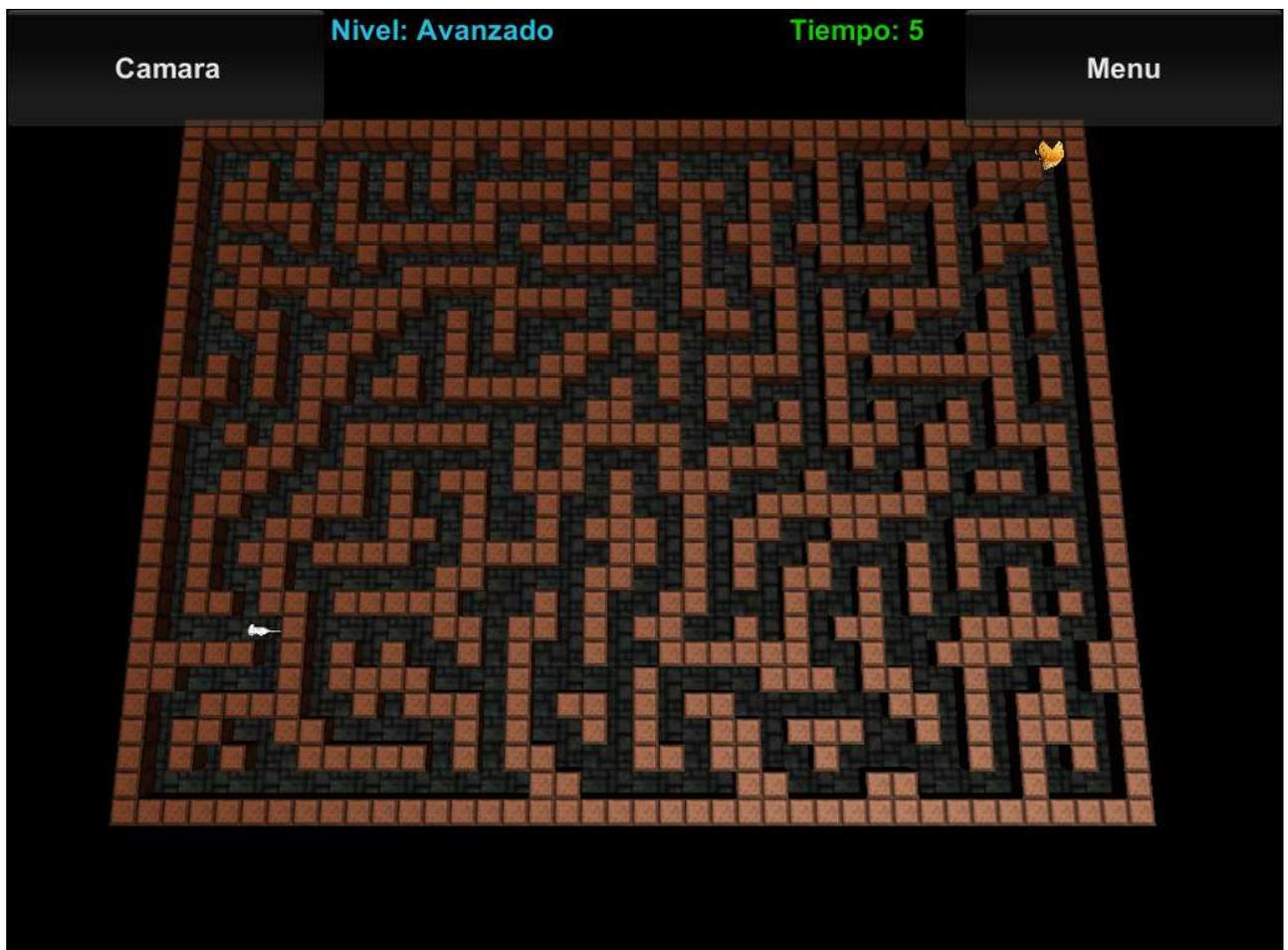


Figura 53. Ventana de la Opción Avanzada Automática del Juego.

Ventana de Puntajes Automática

En la Figura 54 se muestra la pantalla de los puntajes obtenidos por los diferentes usuarios que han jugado el juego manualmente incluido la PC que juega automáticamente.



	Nombre	Dificultad	Tiempo
1	Soledad	Principiante	34 seg
2	Soledad	Intermedio	40 seg
3	Soledad	Avanzado	52 seg
4	Cristina	Principiante	57 seg
5	Cristina	Intermedio	119 seg
6	Cristina	Avanzado	94 seg
7	Daniela	Principiante	63 seg
8	Daniela	Intermedio	72 seg
9	Daniela	Avanzado	129 seg
10	PC	Principiante	12 seg
11	PC	Intermedio	38 seg
12	PC	Avanzado	51 seg

Menu

Figura 54. Ventana del Puntaje de la Opción Automática.