

ESCUELA POLITÉCNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**ANÁLISIS, DISEÑO Y DESARROLLO DE UN JUEGO
DIDÁCTICO DE RAZONAMIENTO ABSTRACTO EN 3D,
PARA AYUDAR AL DESARROLLO DEL PENSAMIENTO DE
NIÑOS ENTRE 4 Y 8 AÑOS, UTILIZANDO UN GAME ENGINE
CON C# Y APLICANDO LA METODOLOGÍA OOHDM**

Previa a la obtención del Título de:

INGENIERO DE SISTEMAS E INFORMÁTICA

**POR: KARLA MARIA ALBUJA FIALLO
VERÓNICA FERNANDA MOLINA ANCHALA**

SANGOLQUÍ, Junio del 2012

DECLARACIÓN

Nosotros, Karla Maria Albuja Fiallo y Verónica Fernanda Molina Anchala, declaramos que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica del Ejército (ESPE), puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Karla Albuja

Verónica Molina

CERTIFICACIÓN

Certificamos que el presente trabajo, fue realizado en su totalidad por las Sras. Karla María Albuja Fiallo y Verónica Fernanda Molina Anchala como requerimiento parcial a la obtención del Título de **INGENIEROS DE SISTEMAS E INFORMÁTICA**, bajo nuestra supervisión.

Sangolquí, Junio de 2012

Ing. César Villacís

Ing. Germán Nacato

DEDICATORIA

Dedico este trabajo a mi hija Alisson Micaella, quien fue la razón y quien me empujo para continuar y no desfallecer, que desde el cielo me dio la fuerza para culminar con mis estudios Universitarios y lograr mis objetivos planteados.

A mis hijos adorados Julián David y Diego Nicolás quienes con su amor, admiración a sus padres me animan día a día para ser el mejor ejemplo que ellos pueden ver en mi.

A mi padre Carlos Efraín por todos sus valores, ejemplos y enseñanzas; que ha forjado en mi un espíritu de perseverancia, siendo la mujer que soy; y de esta manera lograr mis objetivos y mis metas desde los inicios de mi infancia.

A mi madre Elsa Margoth, que con sus consejos, su apoyo incondicional y por siempre estar junto a mi, como Madre y amiga.

A mi esposo por su apoyo, su amor y comprensión de todos los días que ha estado junto a mi; compartiendo mis alegrías, tristezas y tropiezos pero siempre ahí apoyándome y levantándome para continuar.

“Quien es constante y perseverante, siempre alcanzará sus objetivos con la ayuda de Dios”

Karla Albuja

DEDICATORIA

La familia es una de las joyas más preciadas que uno puede tener, sin la familia uno no puede conseguir la fuerza necesaria para lograr las metas.

Este proyecto es un esfuerzo grande que involucra a muchas personas cercanas a mí, es por eso que dedico esta tesis a mi abuelita quien siendo como una madre me acompañado y dado su amor incondicional, a mis padres por ser ese ejemplo de superación y pilares fundamentales en mi vida, mis hermanas por ser mis cómplices y mejores amigas, Paúl por compartir la vida juntos y ser un apoyo constante y principalmente a mi hijo Leandro, que es la bendición más grande que Dios me pudo haber dado sin duda mi referencia para el presente y el futuro.

“Mira que te mando que te esfuerces y seas valiente, no temas ni desmayes porque Jehová tu Dios estará contigo donde quiera que vayas” (Josue 1:9)

Verónica Molina

AGRADECIMIENTO

Le doy gracias a mi Dios por permitirme vivir cada día y ser la persona que soy y por ayudarme en todos los momentos de mi vida.

A mis padres por darme la oportunidad de recibir una Educación Superior, por su inmenso amor, ejemplo, entereza, cuidado, dedicación, amistad y consejo diario.

A mi esposo Diego y a mis hijos Julián y Nicolás; que se convirtieron en el pilar fundamental para la culminación de mi trabajo, ya que con su apoyo moral y su cariño fortalecieron mis deseos de cumplir mis objetivos trazados.

A mis Hermanos, Familiares y Amigos que a lo largo de este tiempo han sido un apoyo emocional, encontrado siempre una mano en la cual apoyarme en los momentos difíciles.

A mi compañera de tesis Verónica, que siempre ha estado ahí en las alegrías y tristezas, para lograr juntas el gran anhelo de convertirnos en los nuevos Ingenieros de la Patria.

A mi Director y Codirector de tesis quienes con su experiencia y profesionalismo supieron encaminar de la mejor manera esta obra, y a todas las personas que de una u otra manera colaboraron en su culminación.

Karla Albuja

AGRADECIMIENTO

Agradezco a Dios quién con su infinito amor me ha guiado y acompañado en todo momento de mi vida.

A mis padres por apoyar de manera incondicional este sueño, respetando mis decisiones, inculcándome valores éticos y morales; los cuales se han plasmado en mi personalidad, y en la práctica de mis hechos.

A mi esposo Paul por todo el apoyo incondicional, comprensión, amor que me ha dado siempre en las buenas y malas con sus palabras de aliento y motivación para continuar y no decaer.

A mi angelito Leandro quien con una sonrisa me impulsa cada día a seguirme superando y cumplir mis sueños y anhelos.

A mi Abuelita querida Aurora quien con su consejo y apoyo han sembrado en mí los deseos de superación, logrando ser cada día un mejor ser humano.

A familiares y amigos por haber estado ahí motivándome para continuar y culminar con este proyecto.

Mis más sinceros Agradecimientos a nuestro Director y Co-Director quienes nos brindaron todo el apoyo para la realización de nuestro tema de tesis.

Y por último, y sin ser menos importante a mi compañera de tesis Karla María con quien hemos compartido lágrimas y alegrías en este duro caminar y con quien hoy estamos cultivando el fruto de nuestra dedicación.

Verónica Molina

ÍNDICE DE CONTENIDOS

DECLARACIÓN	ii
CERTIFICACIÓN	iii
Dedicatoria	iv
Agradecimiento	v
RESUMEN	1
CAPÍTULO I	3
GENERALIDADES.....	3
CAPÍTULO II	13
MARCO TEÓRICO.....	13
2.1 Videojuegos Lúdicos	13
2.1.1 El Videojuego como Actividad de Entretenimiento.....	13
2.1.2 Los Videojuegos como Fenómeno Social	24
2.1.3 Tipos de Videojuegos y Propuestas de Clasificación.....	29
2.1.4 La Polémica: Partidarios y Detractores	39
2.1.5 La Investigación Sobre Videojuegos	43
2.2 Ingeniería de Software	48
2.2.1 Definición.....	48
2.2.2 Marco de Trabajo para el Proceso	49
2.3 Metodología de Diseño de Hipermedia Orientada a Objetos (OOHDM)	51
2.3.1 Historia	51
2.3.2 Definición.....	53
2.3.3 Características Fundamentales.....	53
2.3.4 Etapas ó Fases.....	54
2.3.5 Ventajas y Desventajas	67
2.3.6 Criterios de Selección de OOHDM	69
2.4 Lenguaje Unificado De Modelado (UML)	70
2.4.1 Definición.....	70
2.4.2 Diagramas	71
CAPÍTULO III	81
ANÁLISIS Y DISEÑO DEL JUEGO LÚDICO	81
3.1 Especificación de Requerimientos	81
3.1.1 Introducción.....	81

3.1.2	Identificación de Roles y Tareas.....	84
3.1.3	Especificación de Escenarios.....	85
3.1.4	Especificación de casos de uso por Actor.....	86
3.1.5	Requerimientos No Funcionales.....	93
3.2	Diagramas de Secuencia.....	93
3.3	Diseño Conceptual.....	93
3.4	Diseño Navegacional.....	96
3.4.1	Esquema Navegacional.....	97
3.4.2.	Esquema de Contextos Navegacionales.....	100
3.4.3.	Arquitectura del Sistema.....	101
3.5	Diseño de Interfaz Abstracta.....	102
3.5.1	Vista de Datos Abstractos.....	102
3.5.2.	Diagramas de Configuración.....	104
3.5.3.	Diagramas de Estado.....	105
3.6.	Diseño Estético.....	105
3.6.1.	Características de la Plantilla.....	105
3.6.2	Consideraciones de Diseño Gráfico.....	106
3.9	Diseño de Componentes.....	107
3.10	Diagramas de Despliegue.....	109
CAPÍTULO IV.....		110
IMPLEMENTACIÓN Y PRUEBAS DEL VIDEOJUEGO CON UNITY.....		110
4.1	Unity.....	110
4.1.1	Definición.....	110
4.1.2	Características.....	111
4.1.3	Librerías de Unity.....	116
4.1.4	Estructura de la interfaz visual del videojuego hecho con Unity.....	122
4.1.4.1	Componentes.....	122
4.2	Construcción del Videojuego.....	123
4.2.1	Creación del Modelo 3D en Maya.....	123
4.2.2	Creación del Videojuego en Unity.....	124
4.3	Desarrollo e Implantación de Componentes para la el JDRA3D.....	127
4.4	Pruebas de la Aplicación.....	129
4.5.1	Prueba de Contenido.....	130
4.5.2	Prueba de Interfaz de Usuario.....	131
4.5.3	Prueba de Navegación.....	132

4.5.4 Prueba de Componentes.....	132
4.5.5 Prueba de Configuración.....	133
4.5.6 Prueba de Seguridad.....	133
4.5.7 Prueba de Desempeño.....	134
4.5 Deployment de la Aplicación.....	134
CAPÍTULO V.....	136
CONCLUSIONES Y RECOMENDACIONES	136
5.1 Conclusiones	136
5.1.1 Conclusiones del Proyecto.....	136
5.1.2 Conclusiones Académicas	137
5.2 Recomendaciones	138
5.2.1 Recomendaciones del Proyecto	138
5.2.2 Recomendaciones Académicas.....	140
BIBLIOGRAFÍA	142
GLOSARIO DE TÉRMINOS.....	144
BIOGRAFIA.....	144
BIOGRAFIA.....	144

INDICE DE TABLAS

Tabla 2.1. Alegaciones a favor y en contra de los video juegos.....	42
Tabla 2.2. Líneas de investigación sobre videojuegos.....	43
Tabla 2.3. Etapas de la Metodología OHDM.....	56
Tabla 3.1. Configurar el videojuego.....	87
Tabla 3.2: Resolver los retos del videojuego.....	89
Tabla 3.3: Visualizar los puntajes del videojuego.....	91
Tabla 3.4. Clase Navegacional – Nodo Formulario del Menú Principal.....	97
Tabla 3.5. Clase Navegacional – Nodo de Configuración del Juego.....	97
Tabla 3.6. Clase Navegacional – Nodo de Navegación y Entorno del Juego.....	98
Tabla 3.7. Clase Navegacional – Nodo de Puntajes del Juego.....	99
Tabla 3.8. Clase Navegacional – Nodo del Manual del Juego.....	99

INDICE DE FIGURAS

Figura 2.1: Estratos de la Ingeniería de Software.....	48
Figura 2.2: Esquema del Diseño Navegacional.....	63
Figura 2.3: Ejemplo de un Caso de Uso.....	72
Figura 2.4: Ejemplo de un Diagrama de Secuencia.....	74
Figura 2.5: Ejemplo de una Clase.....	75
Figura 2.6: Ejemplo de un Diagrama de clases.....	78
Figura 2.7: Ejemplo de un Diagrama de Despliegue.....	79
Figura 2.8: Ejemplo de un Diagrama de Componentes.....	78
Figura 3.1. Actores - JDRA3D.....	84
Figura 3.2. Caso de Uso para Jugador del JDRA3D.....	86
Figura 3.3. Archivos Planos del JDRA3D.....	94
Figura 3.4. Diagrama de Clases del JDRA3D.....	95
Figura 3.5. Modelo de Clases Navegacional del JDRA3D.....	100
Figura 3.6. Esquema de Contexto - Menú Principal.....	101
Figura 3.7. Arquitectura del JDRA3D.....	101
Figura 3.8. Vista Abstracta del Nodo Formulario del Menú Principal.....	102
Figura 3.9. Vista Abstracta del Nodo del Formulario Cómo Jugar.....	103
Figura 3.10. Vista Abstracta del Nodo del Formulario para Empezar el Juego....	103
Figura 3.11. Vista Abstracta del Nodo de Navegación y Entorno del Juego.....	103
Figura 3.12. Vista Abstracta del Nodo de Puntajes del Juego.....	104
Figura 3.13. Diagrama de Estado del JDRA3D.....	105
Figura 3.14. Diseño Estético – Organización de los Elementos.....	106
Figura 3.15. Diseño de Componentes del JDRA3D.....	108
Figura 3.16. Diagrama de Despliegue del JDRA3D.....	109
Figura 4.1. El ícono de carga del paquete Unity.....	111
Figura 4.2. Interfaz de Usuario de Unity.....	112
Figura 4.3. Gizmo de Perspectiva 3D.....	113
Figura 4.4. Configuración de la Visualización.....	114
Figura 4.5. Botones de Control.....	116
Figura 4.6. Creación de Escenas.....	117
Figura 4.7. Generador de Terrenos.....	119
Figura 4.8. Seteo del Renderizado.....	120
Figura 4.9. Controlador de Primera Persona.....	121
Figura 4.10. Estructura de la interfaz visual del videojuego hecho con Unity.....	122
Figura 4.11. Elementos del Videojuego.....	124
Figura 4.12. Ventana de la Aplicación del Juego en ambiente Windows.....	127
Figura 4.13. Ventana de la Aplicación del Juego en ambiente Web.....	128
Figura 4.14: Proceso de Prueba.....	129
Figura A.1: Diagrama de Secuencia – Gestionar de Componentes.....	151
Figura A.2: Diagrama de Secuencia – Gestionar de Componentes.....	152
Figura A.3: Diagrama de Secuencia – Gestionar de Servicio de Foros.....	153
Figura B.1. Ventana del Menú Principal.....	154
Figura B.2. Ventana de Configuración del Juego para un niño.....	155

Figura B.3. Ventana de Configuración del Juego para una niña.....	155
Figura B.4. Ventana de indicación de cómo jugar.....	156
Figura B.5. Funcionamiento del Juego – Pieza a seleccionar.....	157
Figura B.6. Funcionamiento del juego – Accionamiento de la palanca.....	157
Figura B.7. Ventana del Entorno del Juego para una PC.....	158
Figura B.8. Accionamiento de la palanca.....	159
Figura B.9. Armado del juego.....	159
Figura B.10. Ventana de Finalización del Reto Superado.....	160
Figura B.11. Ventana del Puntaje.....	161

INDICE DE ANEXOS

ANEXO A.....	149
ANEXO B.....	153

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 en 3D, para ayudar al desarrollo del pensamiento de niños entre 4 y 8 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 de cuatro etapas: Diseño Conceptual, Diseño Navegacional, Diseño Abstracto de Interfaz e Implementación.

Para cumplir con este objetivo, la presente tesis se ha estructurado de la siguiente manera:

- a) El primer capítulo titulado como Generalidades, consta de la formulación del problema, la importancia y justificación, los objetivos de la tesis, el alcance del proyecto, la metodología de trabajo y las herramientas de software a ser utilizadas.
- b) El segundo capítulo titulado como Marco Teórico, consta de una breve reseña acerca de los videojuegos lúdicos, los tipos de videojuegos, una introducción a la ingeniería de software y al marco de trabajo para este proceso, una breve descripción de la metodología OOHDM con cada una de sus cuatro fases y el lenguaje unificado de modelado (UML).
- c) El tercer capítulo titulado como Análisis y Diseño del Juego Lúdico, consta de la especificación de requerimientos, los casos de uso del sistema, los requerimientos funcionales y no funcionales del mismo, los diagramas de secuencia, el diseño conceptual, el diagrama de clases, el diseño navegacional, la arquitectura del sistema, el diseño de la interfaz abstracta y los diagramas de estado, junto con el diseño estético, el diseño de componentes y el diagrama de despliegue del sistema.
- d) El cuarto capítulo titulado como Implementación y Pruebas del Videojuego con Unity, consta una introducción al Game Engine Unity, la estructura de la interfaz visual del videojuego, una descripción de los componentes utilizados y desarrollados para la aplicación, la creación del modelo 3D con Maya, la creación del videojuego con Unity y las pruebas de la aplicación.
- e) El quinto y último capítulo consta de las conclusiones y recomendaciones del proyecto de tesis.

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, aún nos invade la sensación de querer proteger de estas tecnologías olvidándonos de que quizás nuestra tarea no sea tanto la de aislarles de mucha tecnología, sino más bien la de adecuarles en el uso adecuado de 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 concientizar en preparar a nuestros alumnos para el presente y el mañana, por tanto, si se hace una reflexión 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 aun 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 nuestro 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, 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 como una herramienta de apoyo en el proceso del aprendizaje.

La aplicación que se desarrollará permitirá tener las bases teóricas, prácticas, habilidades y destrezas en la formación integral de los niños provocando en ellos un interés y un aprendizaje fácil mediante el uso juegos educativos interactivos en 3D.

1.3. Objetivos

1.3.1. Objetivo General

Analizar, diseñar y desarrollar un juego didáctico de razonamiento abstracto en 3D, para ayudar al desarrollo del pensamiento de niños entre 4 y 8 años, utilizando un Game Engine con C# y aplicando la Metodología OOHDM.

1.3.2. Objetivos Específicos

- Revisar y documentar los conceptos teóricos acerca de aplicativos de software 3D, juegos didácticos y motores de juegos.

- Revisar y documentar las distintas fases de la metodología OOHDM.
- Realizar el análisis y el diseño de la aplicación 3D utilizando la metodología OOHDM con UML.
- Realizar el diseño y el desarrollo arquitectónico del modelo 3D de la aplicación con una herramienta de diseño 3D.
- Realizar el desarrollo y la implementación de la aplicación 3D utilizando la metodología OOHDM con UML.
- Utilizar herramientas tecnológicas de diseño 3D y programación para construir aplicaciones 3D.

1.4. Alcance

El juego didáctico a desarrollarse se orienta en el proceso de enseñanza – aprendizaje del razonamiento abstracto de niños entre 4 y 8 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, servirá como material de consulta y apoyo en 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 abstracto será un juego de rompecabezas con figuras geométricas en 3D y con tres niveles de dificultad (básico, intermedio y avanzado). El juego además contempla el manejo de registro de usuarios y de los tiempos que el usuario se demora en armar el rompecabezas, los mismos que serán registrados en archivos planos.

El juego didáctico de razonamiento abstracto será diseñado para correr en plataforma Windows y Web y podrá ser manipulado a través del teclado y ratón (mouse).

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 Interfaz 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 si la implementación del hiperdocumento o sistema hipermedial diseñado, es decir, la concreción de los modelos navegacionales y de interface 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 la Web y otro de tipo desktop que correrá sobre la plataforma Windows. La Metodología OOHDMM se aplicará en conjunto con la Ingeniería de Software, utilizando el patrón de diseño MVC (Modelo Vista Controlador) y UML (Lenguaje de Modelamiento Unificado).

1.6. Herramientas

Se definen de acuerdo a las dos capas que posee la aplicación: Presentación, y Reglas del Negocios.

Sistema Operativo Windows

Nombre	Descripción
XP, Windows 7	El Sistema Operativo Windows XP y Windows 7 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 y Mac OS X, y permite crear juegos para Windows, Mac, Xbox 360, PlayStation 3, Wii, iPad y iPhone, y también 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 Lúdicos

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

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

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.

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, añadiríamos, juegan. El juego, además de haber sido ampliamente estudiado por el papel fundamental que representa en el desarrollo del niño, cumple

(1998), Cañeque (1999) o López Mantalla (1993).

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 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,

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).

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*).

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 Caillois 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 podemos obviar y que se hallan, no tanto en las

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.

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:

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 estamos sometidos en la vida diaria.

B) El juego es autotélico. El carácter voluntario y libre del juego nos 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 refiere 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 es el hecho de divertirse. Por ello, cuando alguien siente la insatisfacción en el juego (por ejemplo, al haber perdido) no 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 apunta 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 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 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. Podríamos 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 describimos 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 la 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 la sociedad. Como tales 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 sobreentendido. 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 así se encuentran ante 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, 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

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.

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 o computadoras personales.

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 importante, 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.

Brusa, en su tesis doctoral acerca de los efectos del uso de videojuegos sobre la conducta social de los niños, hace referencia a la clasificación de los video juegos realizada por Bloom⁶ como: a) los juegos de laberinto (en los que el jugador guía al personaje del juego a través de laberintos, en los que la dificultad de juego aumenta progresivamente); b) juegos de escalada (en los que el personaje del juego escala hacia la parte superior de la pantalla, intentando evitar o destruir todo aquello que se interpone en su camino); c) juegos de invasores del espacio (en los que un enemigo desciende de la parte superior de la pantalla mientras el protagonista del juego se mueve de lado a lado, en la parte inferior de la misma, e intenta destruir a los invasores antes de que lo alcancen).

Goldstein valida la clasificación de Crawford (1984), diseñadora de juegos de computadora, quien divide los videojuegos en tres grandes categorías a partir del tipo de destrezas implicadas: juegos de acción y destreza, juegos de estrategia y juegos cognitivos. Según esta diseñadora, los juegos de acción y destreza son aquéllos que fundamentalmente enfatizan destrezas perceptivas y motoras y suelen hallarse en los salones recreativos; los juegos de estrategia se centran en la solución de problemas e incluyen los juegos de aventura y los juegos de rol, pudiéndose hallaren sistemas domésticos y portátiles.

Meggs (1992) establece cinco categorías de videojuegos en función del tipo de juego: juegos de acción/aventura (en ellos los personajes recorren distintos escenarios entornos y vencen a enemigos o superan circunstancias difíciles); juegos

⁶Bloom, S. (1982). *Video invaders*. New York: Arco Publishing. Citado por Brusa, 1987.

de acción/arca de (ponen en práctica distintas habilidades para destruir o defenderse de invasores espaciales u otro tipo de atacadores); juegos de simulación(réplicas de actividades tales como la conducción de un coche de carreras); juegos de deportes; juegos de estrategia (incluyen desde el juego del ajedrez a juegos orientados a la solución de crímenes misteriosos, o incluso el juego de los negocios de bolsa).

La peditra estadounidense Funk, partiendo de la temática de los juegos y no del tipo de videojuego como hiciera Meggs, establece otras cinco categorías (Funk,1993a): juegos de deportes (cuya acción central se refiere a cualquier tipo de deporte, incluyendo las carreras y la lucha); juegos de diversión general (cuyo objetivo principal es vencer a la máquina, sin que aparezca una historia concreta y sin que la lucha o la destrucción sean aspectos prioritarios); juegos educativos(tienen como finalidad aprender alguna información nueva o descubrir otras formas de utilizar esa información); juegos de violencia fantástica (presentan una acción principal en la que un ser fantástico debe luchar, destruir o evitar ser matado mientras intenta alcanzar un objetivo); juegos de violencia humana (presentan la misma estructura que los anteriores pero el protagonista de la historia es un ser humano).

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 (Martínet 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.

Otras clasificaciones no parten de la temática del juego ni de su tipología, sino que aplican unos criterios de clasificación que se ajustan a las necesidades concretas de investigaciones o estudios específicos.

Un ejemplo de este tipo de clasificaciones, menos popular, pero no por ello menos clarificador, es la de Garner (1992)⁷ quien clasifica los videojuegos en función de cuatro criterios distintos.

Los criterios propuestos por Garner son los siguientes:

- **La perspectiva del jugador:** se trata de un criterio de clasificación que hace referencia a la forma en que el jugador se incorpora a la actividad de juego (manipulación de objetos - en los que el jugador controla los movimientos y

⁷Garner realizó una investigación centrada en las máquinas de videojuegos que funcionan con monedas (*coin-operated video games*) por lo que en su clasificación no incluye los videojuegos domésticos.

actividades de un objeto, vehículo, arma, nave, etc. -; manipulación del personaje - en los que el jugador controla los movimientos y actividades de un personaje - ; en primera persona - en los que se crea al jugador la ilusión de que está enfrentándose directamente con la actividad que se desarrolla en la pantalla.

- **El campo de percepción:** cómo se utiliza la pantalla para presentar un universo particular a partir de un programa de juego determinado (de límites fijos - en los que la pantalla presenta al jugador los límites donde tendrá lugar el juego -; de ventana fija - en los que la pantalla no define los límites conceptuales del juego sino que los personajes u objetos pueden moverse más allá de los ejes de la pantalla, dando la sensación de que el juego se lleva a cabo en un universo y que cada pantalla o ventana representa un área del mismo -; de ventana móvil -en los que se crea la apariencia de un foco de visión móvil en lugar de fijo -).

- **Las capacidades interactivas:** competición individual, competición interactiva indirecta (los jugadores van alternando sus turnos); competición interactiva directa (los jugadores utilizan los objetos o personajes del juego para competir entre ellos) y cooperación/competición interactiva directa (los jugadores comparten el esfuerzo para vencer a los enemigos generados por el juego, sin que se excluya la competencia).

- **El escenario de juego:** naturaleza del universo del juego en el que se desenvuelve el jugador (realidad auto-generada - específicamente diseñada para el juego -, realidad transferida - simulación de nuestra vida diaria -, realidad negociada - la que combina las dos realidades anteriores en un contexto determinado -).

Sin duda alguna, 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.

- **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.

- **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: bioecológicos, 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 deporte⁸.
- **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.
- **Reproducciones juegos de mesa:** con el mismo desarrollo que sus originales.

⁸Estallo 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.

Cabe destacar el notable esfuerzo de sistematización realizado por este autor quien, a pesar de ello, reconoce la existencia de videojuegos que difícilmente pueden excluirse en alguna de las categorías descritas.

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 hablamos 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, se centra 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 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 (1993b) 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 2.1 se presenta un resumen de las alegaciones a favor y en contra de los videojuegos.

Tabla 2.1. 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 Afectan de manera negativa **distintas aplicaciones sociales** 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.

2.1.5 La Investigación Sobre Videojuegos

El interés por demostrar los efectos de los videojuegos, ya sean positivos o negativos, es la motivación de la mayoría de las investigaciones desarrolladas. Al revisar los estudios llevados a cabo en este campo se descubrió que los videojuegos han sido estudiados desde diversas perspectivas. La Tabla 2.2 muestra una síntesis de las principales líneas de investigación sobre los videojuegos.

Tabla 2.2. Líneas de investigación sobre videojuegos.

El perfil de los jugadores

- Diferencias de género
- Edad, lugar de juego, preferencias

Los efectos positivos y negativos de los videojuegos

- Videojuegos y agresividad
- Videojuegos y adicción
- Videojuegos y habilidades sociales
- Videojuegos y rendimiento escolar
- Potencial instructivo de los videojuegos

Otras aplicaciones sociales para los videojuegos

- Como medio didáctico
- Como test psicológico
- Como herramienta para el tratamiento

Se encuentran investigaciones preocupadas por analizar las características del perfil de los jugadores, las diferencias de género en el juego o la relación establecida entre la práctica de esta actividad y el uso de la computadora.

Junto a las anteriores, se encuentran las investigaciones orientadas a descubrir hipotéticos efectos nocivos o supuestos beneficios. Como se mencionó anteriormente, los esfuerzos por demostrar las bondades o maldades de estos artilugios han marcado infinidad de trabajos. Algunos, centrados en el análisis de la relación entre el uso de videojuegos y las conductas agresivas; otros preocupados por su posible carácter adictivo, por los efectos fisiológicos de su práctica o por su relación con el rendimiento académico; incluso se han llevado a cabo trabajos enfocados al análisis del posible potencial instructivo de esta actividad lúdica.

Más allá del estudio de las posibles consecuencias del uso de este tipo de juegos, se han puesto en marcha otra serie de investigaciones con la pretensión de estudiar algunas de sus posibles aplicaciones sociales, tales como su utilización como medio didáctico, su aplicación en el campo de las necesidades educativas especiales, o su uso como instrumento para el tratamiento de pacientes oncológicos o en proceso de rehabilitación por quemaduras, etc.

Existen interesantes trabajos de revisión de las investigaciones sobre el tema, tanto en el contexto español como fuera de él. En el primer caso, se encuentra, por ejemplo, el análisis realizado por Etxeberria (1996) quien agrupa los distintos temas objeto de investigación en el ámbito de los videojuegos en doce categorías distintas: adicción, autoestima, aprendizaje, cambios fisiológicos, entrenamiento, efectos negativos, espacial, resolución de problemas, sexo, sociabilidad, terapia y violencia. Fuera de nuestro entorno, Griffiths (1993), autor de varios estudios sobre los efectos de videojuegos, elaboró también una interesante síntesis sobre el estado de la investigación acerca de los videojuegos a principios de los noventa que puede resultar de gran utilidad como primera aproximación a este tema. Hace poco que este mismo autor ha presentado los resultados de un análisis de la literatura existente sobre los videojuegos violentos y la conducta agresiva (Griffiths,1999), muy similares a los publicados un año antes por Dill y Dill (1998) en su revisión acerca de la violencia de los videojuegos.

Conscientes de que una revisión exhaustiva de los estudios realizados sobre pasa con creces los límites de este capítulo introductorio, nos limitaremos a presentar de

manera esquemática aquellos aspectos que mejor ilustran el estado de la investigación en este ámbito.

No es difícil descubrir que la investigación en el campo de los videojuegos cuenta con una clara tradición anglosajona. En Ecuador, y en clara contradicción con la polémica social que el fenómeno de los videojuegos ha provocado, los datos disponibles aún son pocos. No obstante, cabe remarcar que, en los últimos años, el número de trabajos sobre el tema se ha ido incrementado poco a poco.

Al centrarse en el análisis de algunas características bastante comunes en las investigaciones - más concretamente en las investigaciones acerca de los efectos de los videojuegos sobre la conducta de sus usuarios –se puede observar que, en términos generales, aparecen algunos aspectos a tener en cuenta.

En primer lugar, este tipo de investigaciones se caracteriza por tratarse de estudios que miden efectos a corto plazo. Sigue constatándose la necesidad de llevar a cabo estudios a largo plazo, que confirmen los efectos comprobados hasta el momento (Anderson y Ford, 1986; Griffiths, 1991; Provenzo, 1991; Pacala et al.,1995; etc.). Incluso que permitan descubrir otros efectos que pueden no aparecer de forma inmediata y surgir gradualmente después de un período de tiempo.

Se debe tomar en cuenta, también, que una gran parte de la investigación realizada - sobre todo en la década de los ochenta y principios de los noventa - se ha llevado a cabo en los salones recreativos. En la actualidad, con la introducción y expansión de

los sistemas domésticos, estos salones han perdido popularidad y cada vez son más los aficionados que juegan en casa. Lógicamente, las características y condiciones de juego de unos y otros difieren en aspectos importantes (entorno, motivaciones para el juego, inversión económica, posibilidades de control). Sin duda, todas estas diferencias deben ser tomadas en consideración por la investigación actual.

Por otro lado, con frecuencia los estudios se desarrollan en situaciones artificiales de laboratorio o se fundamentan en auto informes de los sujetos. Este tipo de situaciones artificiales pueden contar con múltiples variables externas no controladas, que pueden distorsionar los resultados de una investigación determinada (Funk,1993b). Tal es el caso de la selección de los videojuegos a la hora de determinar si los videojuegos con contenido violento repercuten en la conducta agresiva de los usuarios (Provenzo, 1991).

No todos los videojuegos son iguales y, lógicamente, no todos presentan el contenido violento de la misma forma - por ejemplo, no es igual un videojuego de violencia fantástica que uno de violencia humana -. Diferencias que en la actualidad se acentúan aún más desde el momento en que los videojuegos aumentan en realismo a través de la sofisticación del *software* y *hardware* y de la aplicación de la tecnología de la realidad virtual, al tiempo que su accesibilidad se amplía mediante las llamadas autopistas de la información.

De esta forma, y como se señalará más adelante, aunque puedan establecerse relaciones significativas entre el uso de videojuegos y el desarrollo de habilidades

espaciales y psicomotrices, es muy difícil discernir si los videojuegos son su causa real, considerando el caso del potencial instructivo de los videojuegos lúdicos, pero la falta de datos concluyentes no permiten confirmar que esta relación sea causal, ya que se necesitaría combinar la psicología, la pedagogía y la informática para realizar estudios más profundos que prueben efectivamente el potencial de los videojuegos lúdicos o educativos.

2.2 Ingeniería de Software

2.2.1 Definición

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.

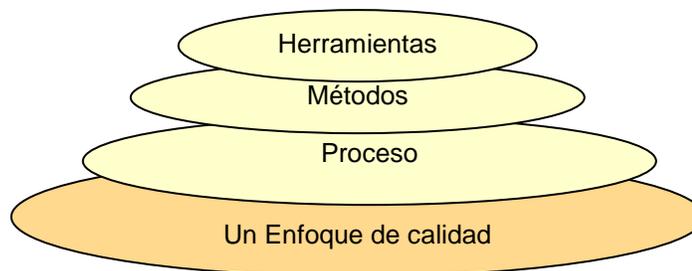


Figura 2.1: Estratos de la Ingeniería de Software

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.

2.3 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**⁹ 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.

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-hipermedia¹⁰ antes de su desarrollo; lo cual fue adoptado también por OOHDM con inclusión de la orientación a objetos.

⁹**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

¹⁰**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).

2.3.2 Definición

OOHDM es una metodología orientada a objetos y ampliamente aceptada para el desarrollo de aplicaciones hipertexto, 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.

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”.¹¹

2.3.4 Etapas ó Fases

OOHDM propone el desarrollo de aplicaciones 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.

¹¹Tomado de: Darío Andrés Silva, Construyendo Aplicaciones Web con una Metodología de Diseño Orientado a Objetos, 2002, pp 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.

Tabla 2.3: 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 requerimiento s 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

<p><i>Diseño Navegacional</i></p>	<p>Nodos, enlaces, estructuras de acceso, contextos navegacionales, transformaciones de navegación</p>	<p>Vistas Orientadas a Objetos, Cartas de navegación orientadas a objetos, Clases de Contexto</p>	<p>Clasificación, agregación, generalización y especialización</p>	<p>Se tiene en cuenta el perfil del usuario y las tareas. Se enfatiza en los aspectos cognitivos. Se crea la estructura de navegación de la aplicación</p>
<p><i>Diseño de Interfaz Abstracta</i></p>	<p>Objetos de la interfaz abstracta, respuestas a eventos externos, transformaciones de la interfaz</p>	<p>Vistas Abstractas de Datos (ADV), Diagramas de Configuración, Cartas de navegación de los ADVs</p>	<p>Mapeado entre la navegación y los objetos visibles</p>	<p>Se modelizan los objetos visibles. Se describe la interfaz para los objetos de navegación. Se define el aspecto de</p>

				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 uso¹², los cuales son diseñados por escenarios con la finalidad de obtener de manera clara los requerimientos y acciones del sistema y de los usuarios.¹³

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 actores¹⁴ y las tareas que ellos deben realizar.

¹²**Caso de Uso:** Descripción a detalle de las actividades y procesos necesarios para el desarrollo de un sistema o aplicación.

¹³**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.

¹⁴**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.

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:

- ¿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?¹⁵

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.

¹⁵ Preguntas claves tomadas de: OOHDM Object Oriented Hypermedia Design Methodology, Agosto 2006

OOHDM propone dividir esta etapa en 5 subetapas:

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

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 UID 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 2.3.)

2.3.4.3.1 Esquema de Clases Navegacionales

Las posibles vistas del hiperdocumento se generan a través de las llamadas clases navegacionales que son tipos predefinidos de clases conformados por: nodos,

enlaces, anclas¹⁶ y estructuras de acceso que son los índices o recorridos guiados, que representan los posibles caminos de acceso a los nodos.

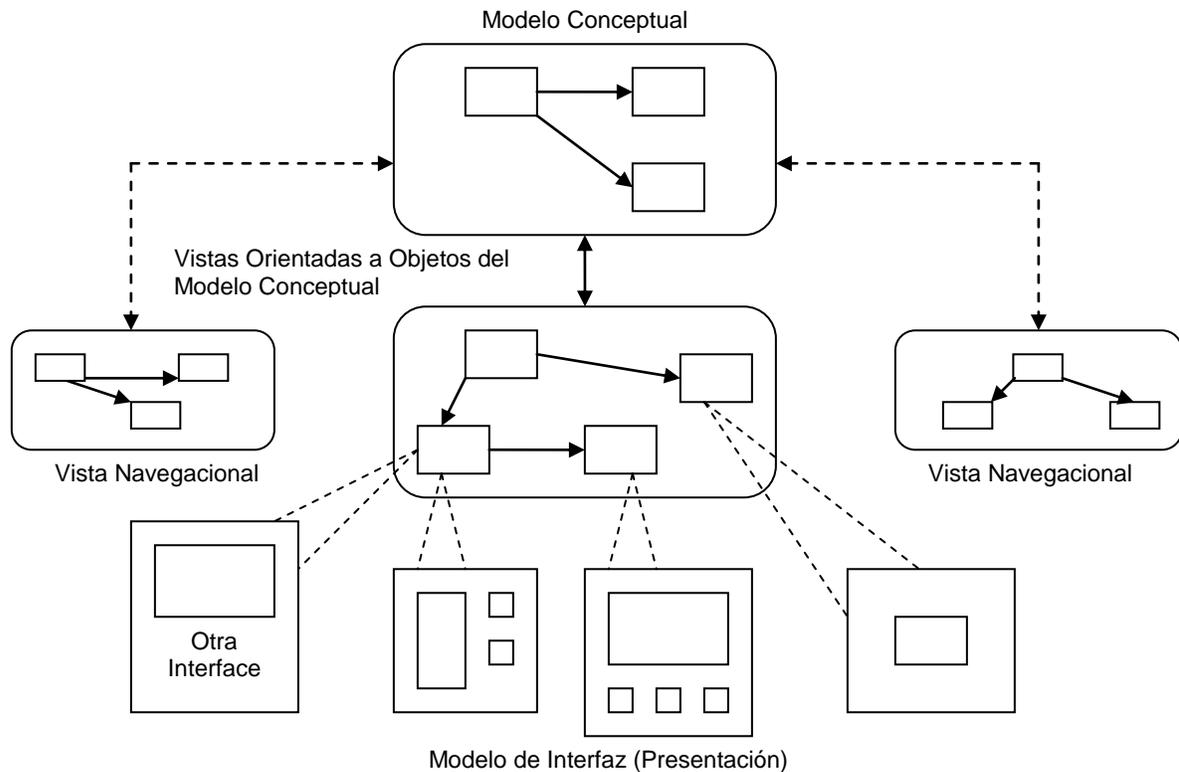


Figura 2.2: Esquema del Diseño Navegacional¹⁷

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

Enlaces

¹⁶ **Ancla:** Los atributos de los nodos que activan navegaciones son llamados anclas.

¹⁷ Figura tomada del documento: OOHDM Object Oriented Hypermedia Design Methodology, Agosto 2006

- 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 subespacios 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 interface 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 cajas¹⁸ que representan las diferentes clases de objetos que aparecerán ante el usuario.

2.3.4.4.2 Diagrama de Configuración

En este diagrama se representan principalmente:

- Los eventos externos provocados por el usuario, como *Clic o Doble Clic 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 OO¹⁹ 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

¹⁸**Caja:** Es un ADV

¹⁹**OO:** Orientados a Objetos

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.²⁰

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.

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 Ventajas y Desventajas

Ventajas

²⁰Tomado de la página: *Crea tu Página Web*, wmaestro.com/webmaestro/docs/portada.html

- Clara identificación de los tres diferentes niveles de diseño en forma independiente de la implementación.
- Su forma de representación gráfica es bastante completa y permite representar en forma precisa elementos propios de las aplicaciones hipermedia, tales como nodos, anclas, vínculos, imágenes, estructuras de acceso y contextos.
- En la etapa de diseño navegacional se pueden crear enlaces entre nodos cualesquiera que permiten una verdadera interoperabilidad entre los mismos.
- El desarrollador puede entender y lograr en cada etapa lo que el usuario realmente necesita, gracias a que en el análisis y diseño, el usuario es parte fundamental en la validación del producto obtenido.
- Al generar una cantidad considerable de documentación a través de sus distintas etapas, permite llevar un control del desarrollo de las mismas y tener la posibilidad de realizar una rápida detección, corrección de errores y mantención.
- La utilización de UIDs permite representar en forma clara, rápida y precisa los casos de uso obtenidos.

Desventajas

- Requiere de cierto conocimiento e investigación para aprender la metodología, debido a los modelos que utiliza.
- El diseño pierde un poco de continuidad del modelo navegacional al diseño de interfaz, dado que se pasa a utilizar otro tipo de modelo.
- En ciertos casos OOHDM podría exagerar la cantidad de reglas y pasos (a veces complicados de seguir) para realizar distintos mapeos entre un diagrama y otro por lo cual el desarrollador podría perderse y olvidar detalles fundamentales a ser especificados.
- El diseño navegacional posee una gran cantidad de diagramas que muchas veces entregan información similar a la entregada por los UIDs y las ADVs.

2.3.6 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 actores²¹, 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.

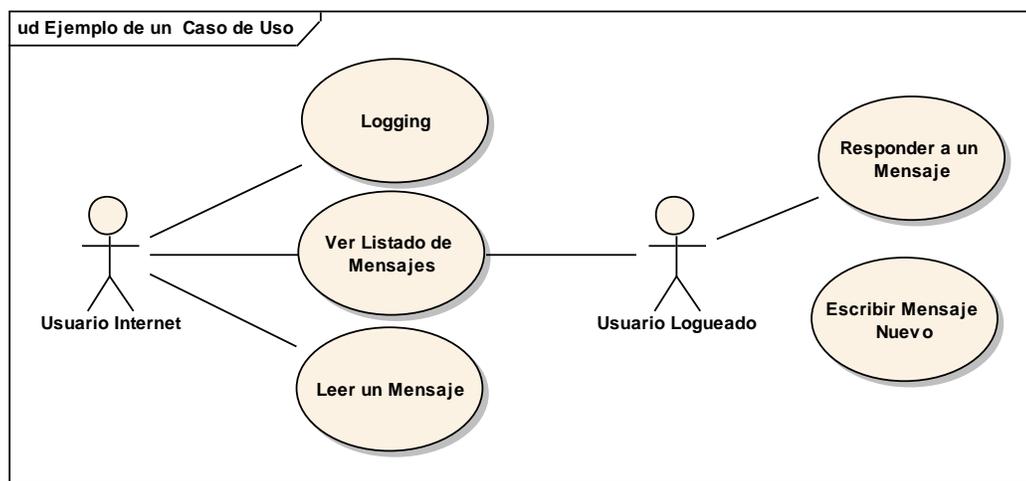


Figura 2.3 Ejemplo de un Caso de Uso

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

²¹ **Actor:** Representa un conjunto coherente de roles que juegan los usuarios de los casos de uso al interactuar con el sistema

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.

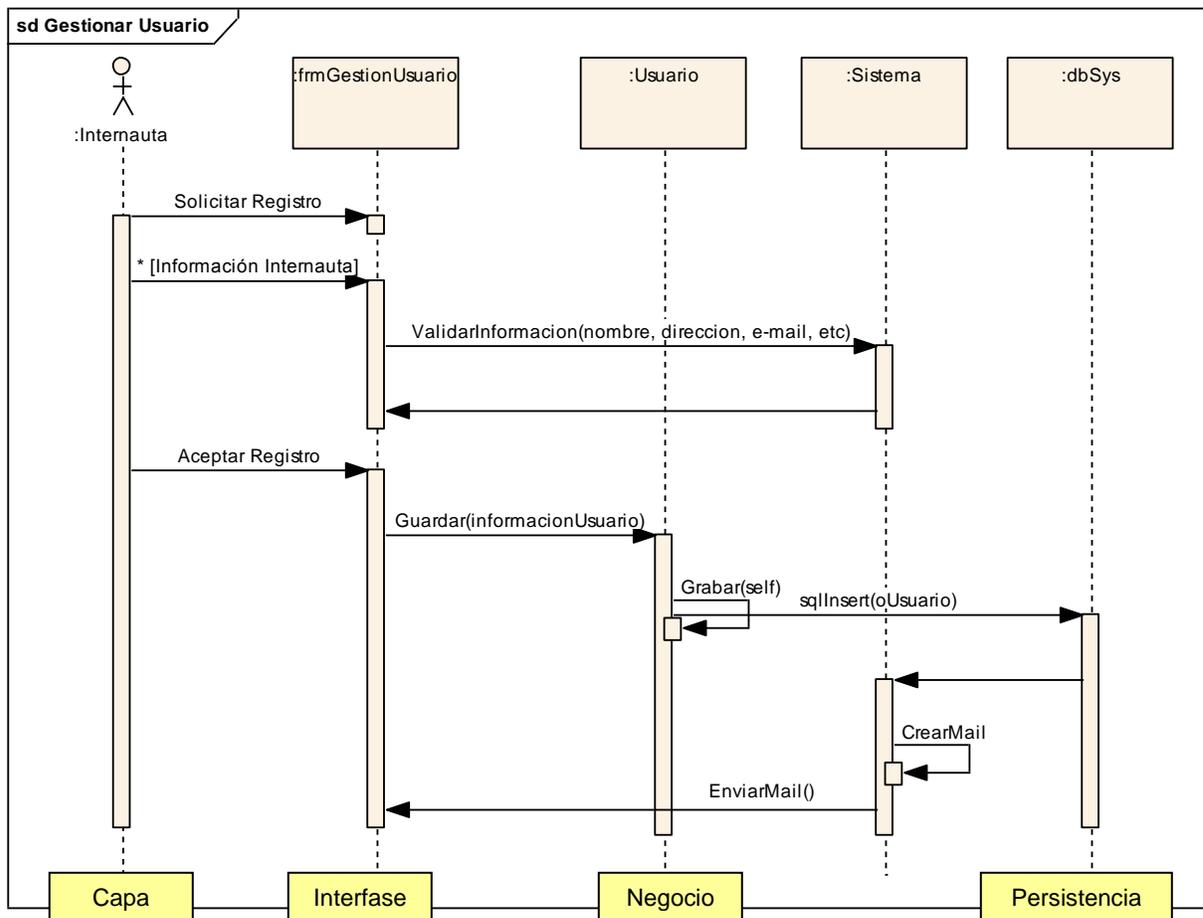


Figura 2.4: Ejemplo de un Diagrama de Secuencia

2.4.2.3 Diagramas de Clase

El Diagrama de Clases²² es el diagrama principal para el análisis y diseño. Modela la vista estática estructural del sistema.

²²**El Diagrama de Clases:** Es el más utilizado en los modelos de sistemas OO; son los “planos” principales [Booch].

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.

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:

- (-) 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.

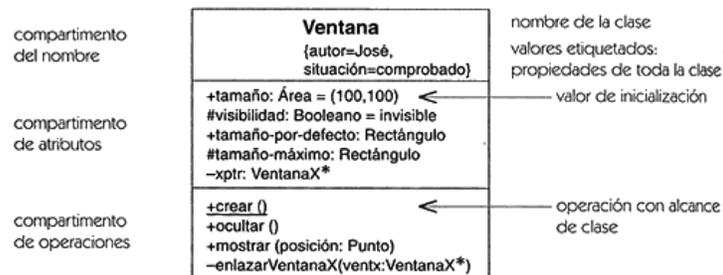


Figura 2.5: Ejemplo de una Clase²³

²³ Figura tomada de la página: 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.

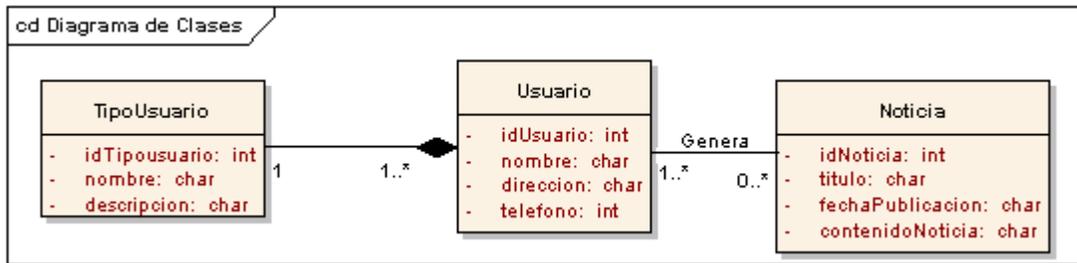


Figura 2.6: Ejemplo de un Diagrama de Clases

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 nodos²⁴ que componen un sistema y el reparto de los componentes sobre dichos nodos.

Con UML, estos diagramas se utilizan para visualizar los aspectos estáticos de estos nodos físicos y sus relaciones y para especificar sus detalles para la construcción.

Usos comunes de los diagramas de despliegue:

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

²⁴**Nodo:** Es un elemento físico que representa un recurso con capacidad computacional.

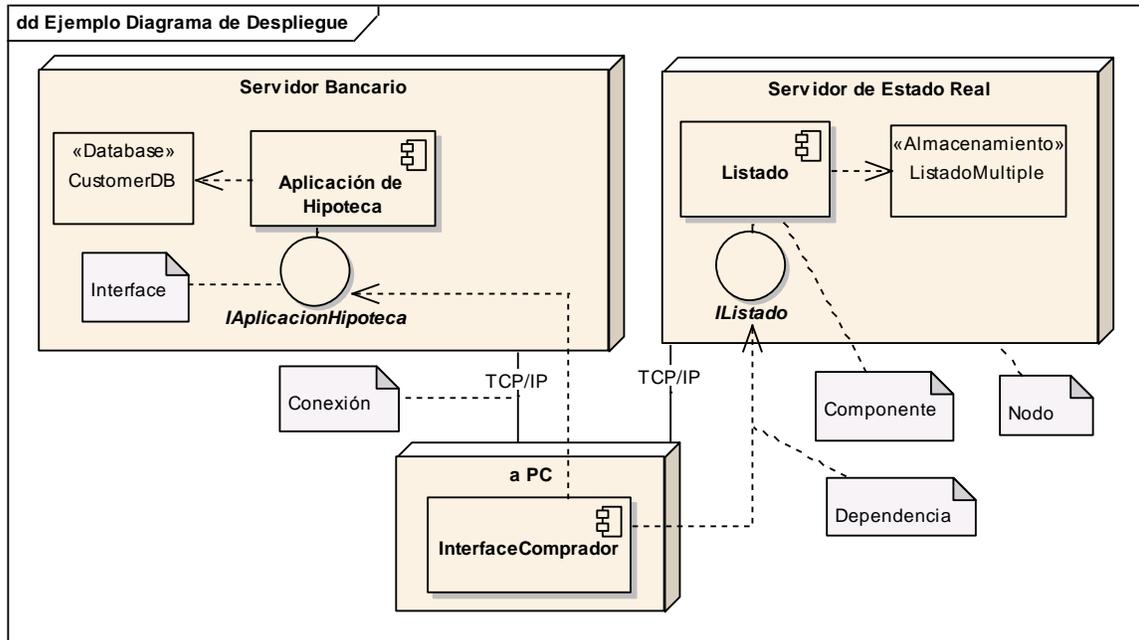


Figura 2.7: Ejemplo de un Diagrama de Despliegue

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.

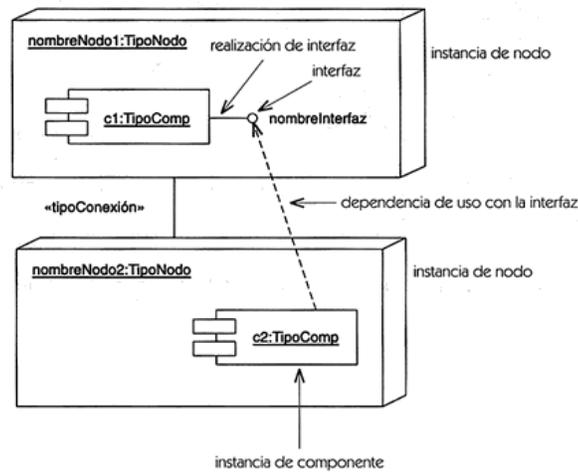


Figura 2.8: Ejemplo de un Diagrama de Componentes²⁵

²⁵ Figura tomada de la página: Diagrama de Componentes, <http://www.creangel.com/uml/componente.php#>

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL JUEGO LÚDICO

3.1 Especificación de Requerimientos

La presente especificación de requerimientos pertenece al desarrollo del Caso de Estudio "Juego Didáctico de Razonamiento Abstracto en 3D, para Ayudar al Desarrollo del Pensamiento de Niños entre 4 y 8 años" 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

3.1.1.1 Propósito

El propósito del presente apartado es definir los requerimientos que debe tener el Juego Didáctico de Razonamiento Abstracto en 3D(JDRA3D) a partir de ahora. Con

la especificación de requerimientos se formalizará las funcionalidades de la aplicación junto al cliente.

3.1.1.2 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).

JDRA3D.- Corresponde a la abreviatura del sistema definido como Juego Didáctico de Razonamiento Abstracto 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 Video-jugador, 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.

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 3.1 se muestra gráficamente al actor del JDRA3D, que como se mencionó anteriormente es el jugador.

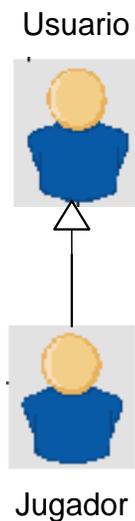


Figura 3.1. Actores - JDRA3D.

B) Tareas

➤ Jugador - Gamer

1. Configurar el videojuego
2. Resolver los retos del videojuego
3. Visualizar los puntajes del videojuego

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 el tiempo de duración del juego y los niveles de dificultad del mismo.
- **Resolver los retos del videojuego:** El jugador podrá resolver los retos del videojuego que se presentan tomando en cuenta un tiempo límite que él se planteará para resolver estos retos.
- **Visualizar los puntajes del videojuego:** El jugador podrá visualizar los diez mejores puntajes almacenados luego de completar los retos del videojuego.

3.1.4 Especificación de casos de uso por Actor

3.1.4.1 ACTOR: JUGADOR

1. USR-JUG- JDRA3D-01: Configurar el videojuego.
2. USR-JUG- JDRA3D-02: Resolver los retos del videojuego.
3. USR-JUG- JDRA3D-03: Visualizar los puntajes del videojuego.

En la Figura 3.2 se muestra el caso de uso para el usuario jugador del JDRA3D.

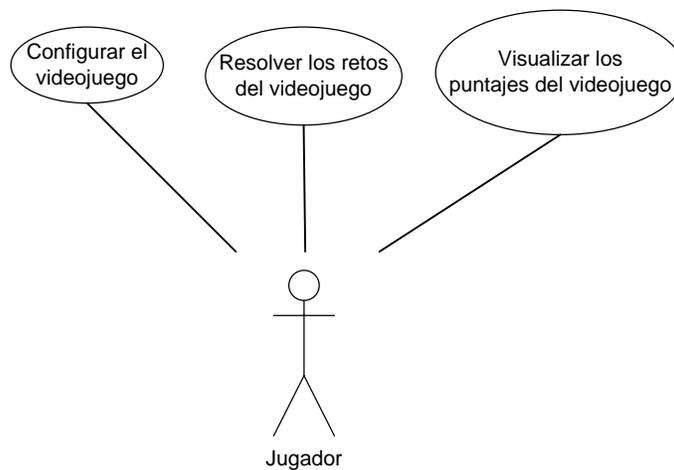


Figura 3.2.Caso de Uso para Jugador del JDRA3D.

3.1.4.2 CASOS DE USO DEL JUGADOR DEL JDRA3D

En la tabla 3.1 hasta la tabla 3.3 se detallan las especificaciones de caso de uso del actor Jugador del JDRA3D.

Tabla 3.1: Configurar el videojuego.

USR-JUG- JDRA3D-01: Configurar el videojuego		
Resumen:	Proceso en el cual el usuario jugador configura la aplicación que se trata de un juego didáctico de razonamiento abstracto 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. Además el usuario puede escoger un niño o una niña como jugadores.</p> <p>2. El jugador puede seleccionar la opción de:</p> <ul style="list-style-type: none"> - Registro del nombre. - Programar tiempo. - Seleccionar el nivel del juego. - Seleccionar tipo de jugador (niño o niña). 	
Secundario	Registro	1. El sistema muestra los controles necesarios para el

	del nombre	<p>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 usuario del juego que luego se almacenará en disco con el resto de los datos en los mejores puntajes.</p>
Secundario	Programar tiempo	<p>1. El sistema muestra los controles necesarios para la programación del tiempo.</p> <p>2. El usuario ingresa o cambia los valores del tiempo.</p> <p>3. Se almacena en memoria temporalmente el tiempo del juego que luego se almacenará en disco con el resto de los datos en los mejores puntajes.</p>
Secundario	Seleccionar el nivel del juego	<p>1. El sistema muestra los controles necesarios para seleccionar el nivel de dificultad de la aplicación.</p> <p>2. El usuario ingresa o cambia el nivel de dificultad del juego.</p> <p>3. Se almacena en memoria temporalmente el nivel del juego que luego se almacenará en disco con el resto de los datos en los mejores puntajes.</p>
Secundario	Seleccionar	<p>1. El sistema muestra los controles necesarios para</p>

	tipo de jugador	<p>seleccionar el tipo de jugador de la aplicación.</p> <p>2. El usuario ingresa o cambia el tipo de jugador del juego.</p> <p>3. Se almacena en memoria temporalmente el tipo de jugador del juego que luego se almacenará en disco con el resto de los datos en los mejores puntajes.</p>
Pre-condiciones		
Ninguna		

Tabla 3.2: Resolver los retos del videojuego.

USR-JUG- JDRA3D-02: Resolver los retos del videojuego	
Resumen:	Proceso en el cual el usuario jugador interactúa con la aplicación que se trata de un juego didáctico de razonamiento abstracto 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 navegar por el videojuego de tal manera que pueda jugar, donde previamente programó un tiempo estimado y un nivel de dificultad que se crea conveniente para superar los retos y se almacena en disco el puntaje logrado.</p> <p>2. El jugador puede seleccionar la opción de:</p> <ul style="list-style-type: none"> - Generar regleta de figuras geométricas. - Mover palanca. - Colocar ficha en la torre de madera. - Guardar puntaje. 	
Secundario	Generar regleta de figuras geométricas	<p>1. El sistema muestra un control en forma de regla que se presenta para cargar las fichas aleatorias de figuras geométricas que corresponden al reto que se desea superar.</p> <p>2. El sistema valida las fichas colocadas dentro de la regleta del juego.</p>
Secundario	Mover palanca	<p>1. El sistema muestra un control en forma de palanca que se necesita para generar fichas aleatorias de figuras geométricas.</p> <p>2. El sistema valida las fichas colocadas dentro de la mesa del juego.</p>

Secundario	Colocar ficha en la torre de madera	<ol style="list-style-type: none"> 1. El sistema muestra un control en forma de torre de madera que se necesita para colocar las fichas de las figuras geométricas de acuerdo a la regleta de figuras generadas aleatoriamente. 2. El sistema valida las fichas colocadas dentro de la torre de madera del juego.
Secundario	Guardar puntaje	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para guardar la aplicación. 2. Se almacenan en disco los mejores puntajes.
Pre-condiciones		
1. El usuario debe ingresar al sistema autenticado como Jugador.		

Tabla 3.3: Visualizar los puntajes del videojuego.

USR-JUG- JDRA3D-01: Visualizar los puntajes del videojuego	
Resumen:	Proceso en el cual el usuario jugador visualiza los puntajes logrados por los diferentes usuarios registrados en la aplicación que se trata de un juego didáctico de razonamiento abstracto en 3D.
Prioridad:	Esencial

Actores	Jugador	
Directos:		
Escenarios		
Tipo de Escenario	Descripción	
Principal	<p>1. La aplicación en 3D ofrece al usuario la posibilidad de almacenar en disco el puntaje logrado en el juego.</p> <p>2. El jugador puede seleccionar la opción de:</p> <ul style="list-style-type: none"> - Regresar al menú principal. - Salir de la aplicación. 	
Secundario	Regresar al menú principal	El sistema muestra los controles necesarios para que el usuario pueda jugar de nuevo, regresando al menú principal.
Secundario	Salir de la aplicación	El sistema muestra los controles necesarios para que el usuario pueda salir del juego.
Pre-condiciones		
1. El usuario debe 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 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 retos resueltos.

3.2 Diagramas de Secuencia

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

3.3 Diseño Conceptual

3.3.1 Diseño de los Archivos Planos

El diagrama de la Figura 3.3 muestra el diagrama de de los archivos planos que se utilizan para leer y escribir la información de los jugadores y sus puntajes del juego.



Figura 3.3. Archivos Planos del JDRA3D.

3.3.2 Diagrama de Clases

La Figura 3.4 muestra el diagrama de clases del sistema, que como se puede ver tiene más de 20 clases.

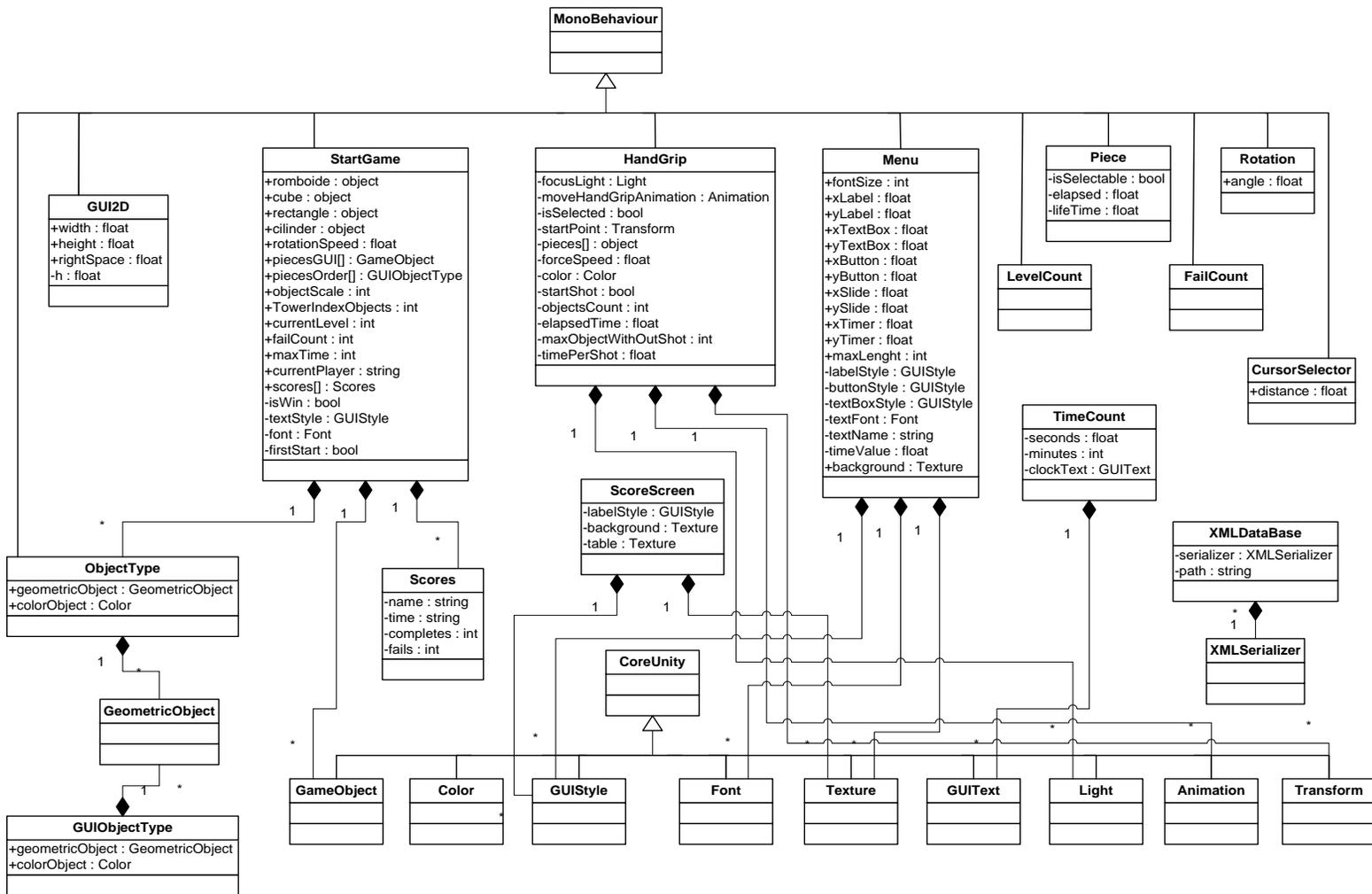


Figura 3.4. Diagrama de Clases del JDRA3D.

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 del Menú Principal
- Página del Formulario del Manual del Juego
- Página del Formulario de Configuración del Juego
- 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 retos
- Guardar Puntajes
- Visualizar Puntajes

3.4.1 Esquema Navegacional

A) Clases Navegacionales

Las tablas 3.4, 3.5, 3.6, 3.7 y 3.8 resumen las principales características de las clases navegacionales del sistema.

Tabla 3.4. Clase Navegacional – Nodo Formulario del Menú Principal

Nombre: Nodo Formulario del Menú Principal
Clases Conceptuales (CC):
Atributos: <ul style="list-style-type: none">➤ Bandera de Control
Descripción: Nodo donde se encuentran las diferentes opciones del menú para acceder al juego
Enlaces: <ul style="list-style-type: none">➤ Al Nodo de Configuración del Juego➤ Al Nodo de Puntajes del Juego➤ Al Nodo del Manual del Juego

Tabla 3.5. Clase Navegacional – Nodo de Configuración del Juego

Nombre: Nodo de Configuración del Juego
Clases Conceptuales (CC):
Atributos:

<ul style="list-style-type: none"> ➤ Usuario ➤ Tiempo de Juego ➤ Nivel de Dificultad ➤ Tipo de Jugador
<p>Descripción: Nodo donde se encuentran las diferentes opciones de configuración para empezar a jugar con el videojuego</p>
<p>Enlaces: Al Nodo de Navegación y Entorno del Juego</p>

Tabla 3.6. Clase Navegacional – Nodo de Navegación y Entorno del Juego

<p>Nombre: Nodo de Navegación y Entorno del Juego</p>
<p>Clases Conceptuales (CC):</p>
<p>Atributos:</p> <ul style="list-style-type: none"> ➤ Retos Completados ➤ Selecciones Fallidas ➤ Control del Tiempo ➤ Reto del Juego ➤ Control del Juego ➤ Nivel de Dificultad del Juego
<p>Descripción: Nodo donde se encuentran todos los retos del juego que el usuario puede superar en función del nivel de dificultad en el que se encuentra</p>
<p>Enlaces: Al Nodo de Puntajes del Juego</p>

Tabla 3.7. 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 ➤ Retos Completados ➤ Número de Selecciones Fallidas ➤ Nivel de Dificultad
Descripción: Nodo donde se encuentran los resultados de los mejores puntajes obtenidos por los usuarios en el juego

Tabla 3.8. Clase Navegacional – Nodo del Manual del Juego

Nombre: Nodo del Manual del Juego
Clases Conceptuales (CC):
Atributos: <ul style="list-style-type: none"> ➤ Bandera de Control
Descripción: Nodo donde se encuentra un vídeo informativo del funcionamiento y del manejo del videojuego

B) Modelo de Clases Navegacionales

La Figura 3.5 muestra el Modelo de Clases Navegacionales del sistema, en la cual a través del Nodo Formulario Menú Principal se acceden a los otros nodos.

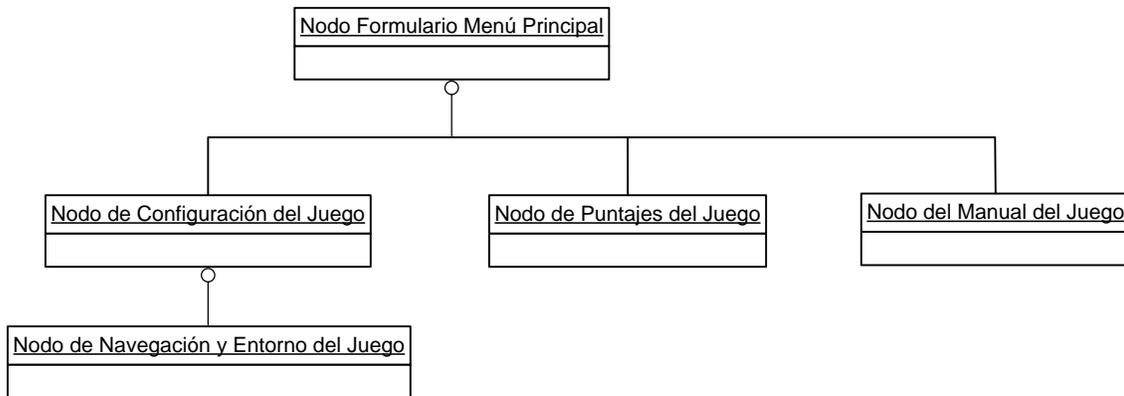


Figura 3.5. Modelo de Clases Navegacional del JDRA3D.

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, como se puede ver en la Figura 3.6:

A) Menú Principal

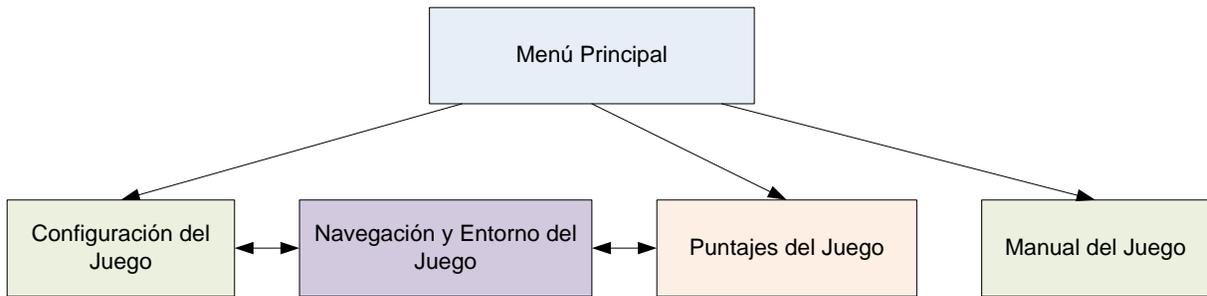


Figura 3.6. Esquema de Contexto - Menú Principal.

3.4.3. Arquitectura del Sistema

La aplicación de software es un sistema de Información que posee la siguiente arquitectura Cliente/Servidor de 2 capas, como se muestra en la Figura 3.7:

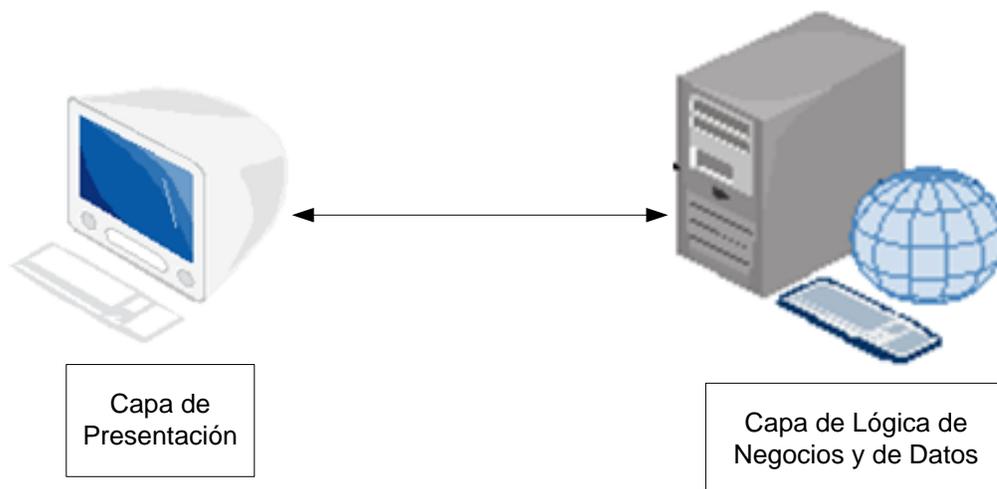


Figura 3.7. Arquitectura del JDRA3D.

Importante: Los servidores de Web y de Datos se encuentran instalados en una misma máquina.

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 Desktop (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 Formulario del Menú Principal**

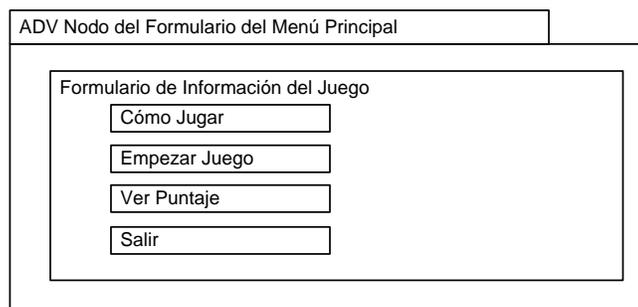


Figura 3.8. Vista Abstracta del Nodo Formulario del Menú Principal.

➤ **Nodo del Formulario Cómo Jugar**

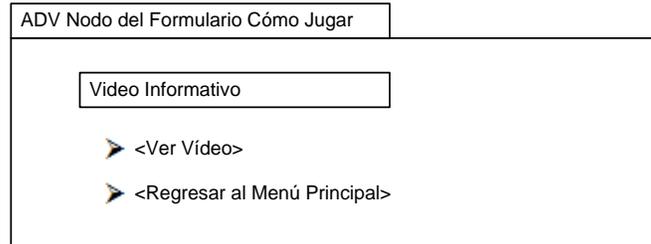


Figura 3.9. Vista Abstracta del Nodo del Formulario Cómo Jugar.

➤ **Nodo del Formulario para Empezar el Juego**

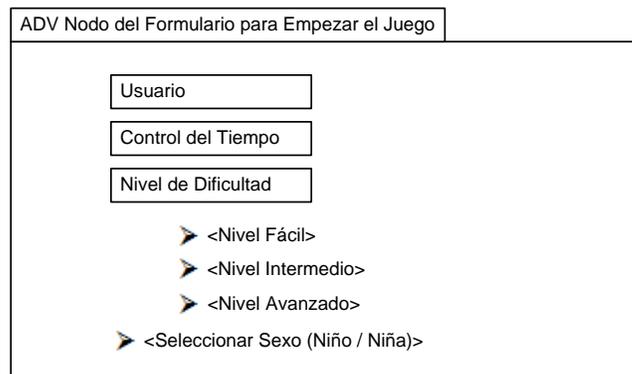


Figura 3.10. Vista Abstracta del Nodo del Formulario para Empezar el Juego.

➤ **Nodo de Navegación y Entorno del Juego**

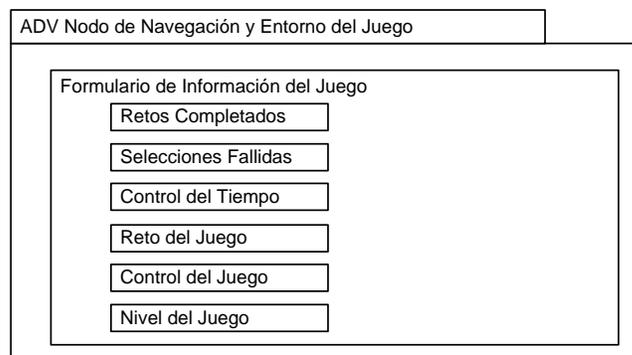


Figura 3.11. Vista Abstracta del Nodo de Navegación y Entorno del Juego.

➤ **Nodo de Puntajes del Juego**

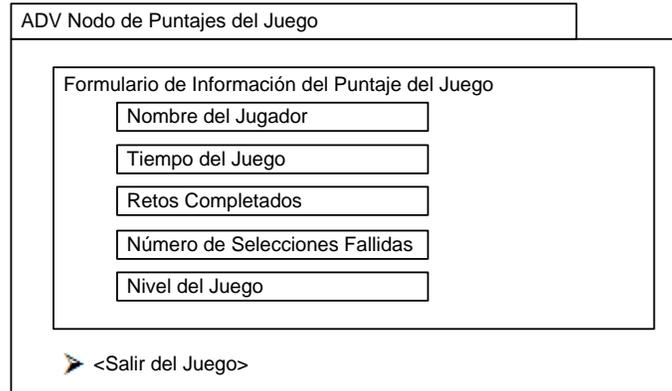


Figura 3.12. Vista Abstracta del Nodo de Puntajes del Juego.

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

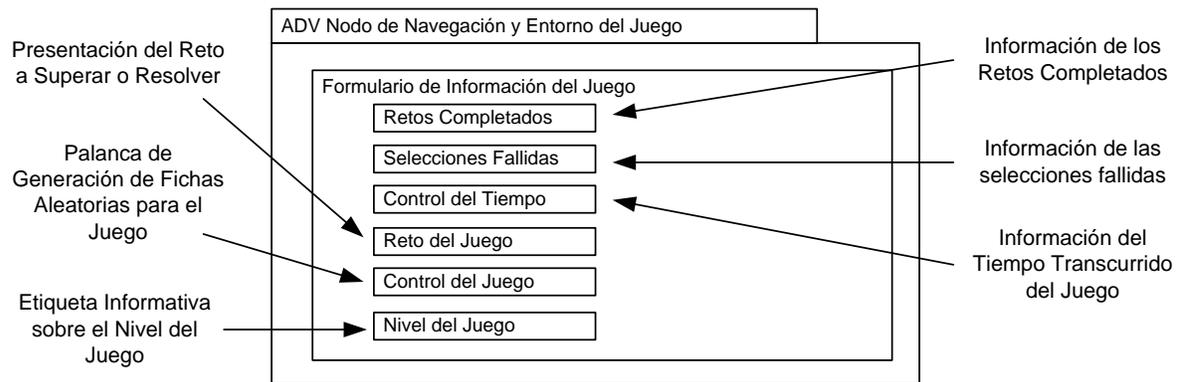


Figura 3.13. Diagrama de Estado del JDRA3D.

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 ó 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:

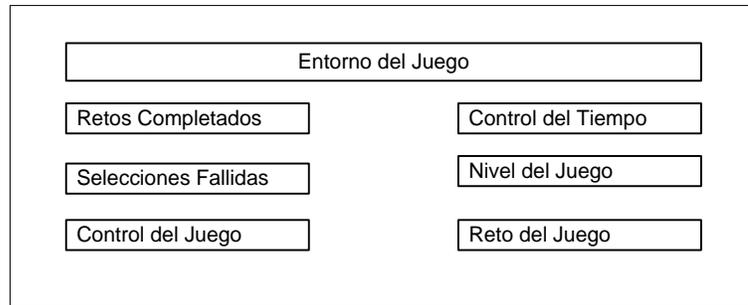


Figura 3.14. Diseño Estético – Organización de los Elementos

- Agrupación de contenido dividida por: manual de usuario, configuración del juego, navegación y entorno del juego y puntajes del mismo.
- No se extiende con la barra de desplazamiento horizontal con el fin de que el usuario se sienta más cómodo navegando por el portal.
- El tamaño del sistema: 80% del espacio total disponible, en una resolución de 800 x 600.

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 del portal: morado, verde 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 3.14 se muestra el Diseño de Componentes del Juego, donde como se puede ver, el juego es una aplicación desktop 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.

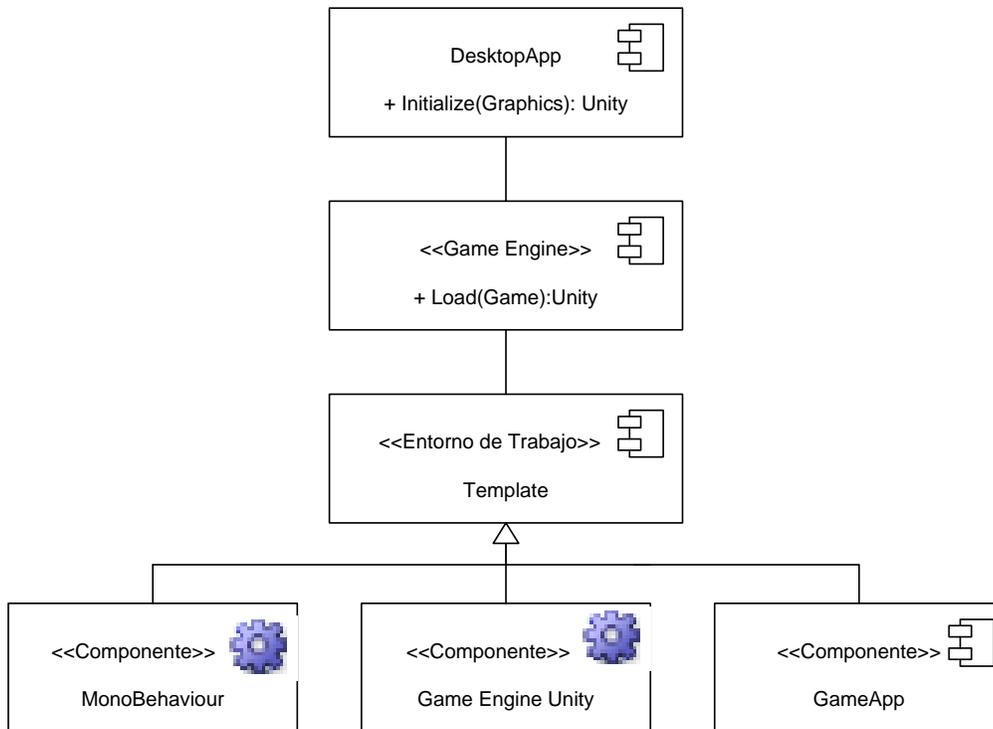


Figura 3.15. Diseño de Componentes del JDRA3D.

3.8. Diagramas de Despliegue

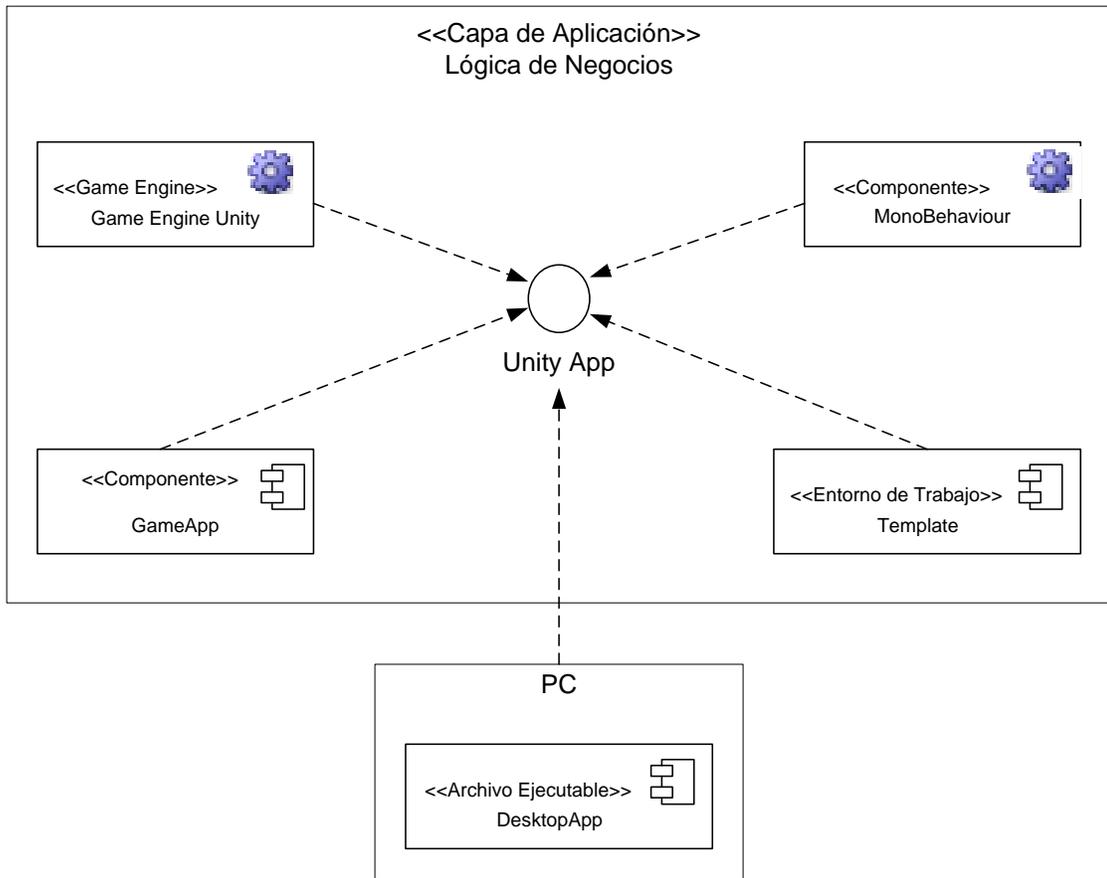


Figura 3.16. Diagrama de Despliegue del JDRA3D.

CAPÍTULO IV

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 Unityweb 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 4.1)



Figura 4.1. 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 4.2.

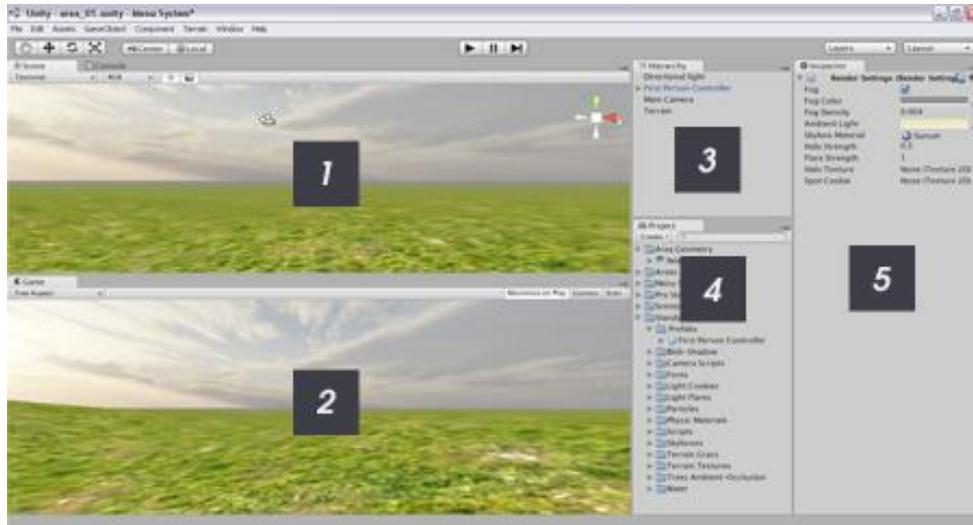


Figura 4.2. Interfaz de Usuario de Unity.

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 previsualización 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 Scripts 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 4.3)

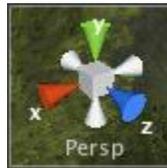


Figura 4.3. Gizmo de Perspectiva 3D.

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 4.4)



Figura 4.4. 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 mostrara 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 4.5. Se puede usar Q, W, E, R para alternar entre cada uno de los controles, que se detalla a continuación:



Figura 4.5. 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 4.6.

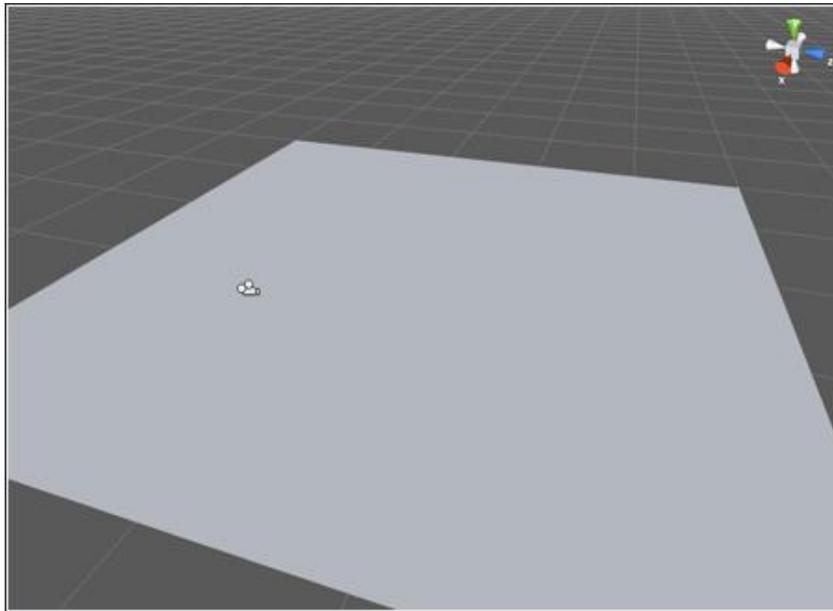


Figura 4.6. Creación de Escenas.

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 4.7. A continuación se explicará cada una de las propiedades más importantes de los terrenos (terrain):

- A) **Width:** El ancho en metros del terreno.
- B) **Lenght:** 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).

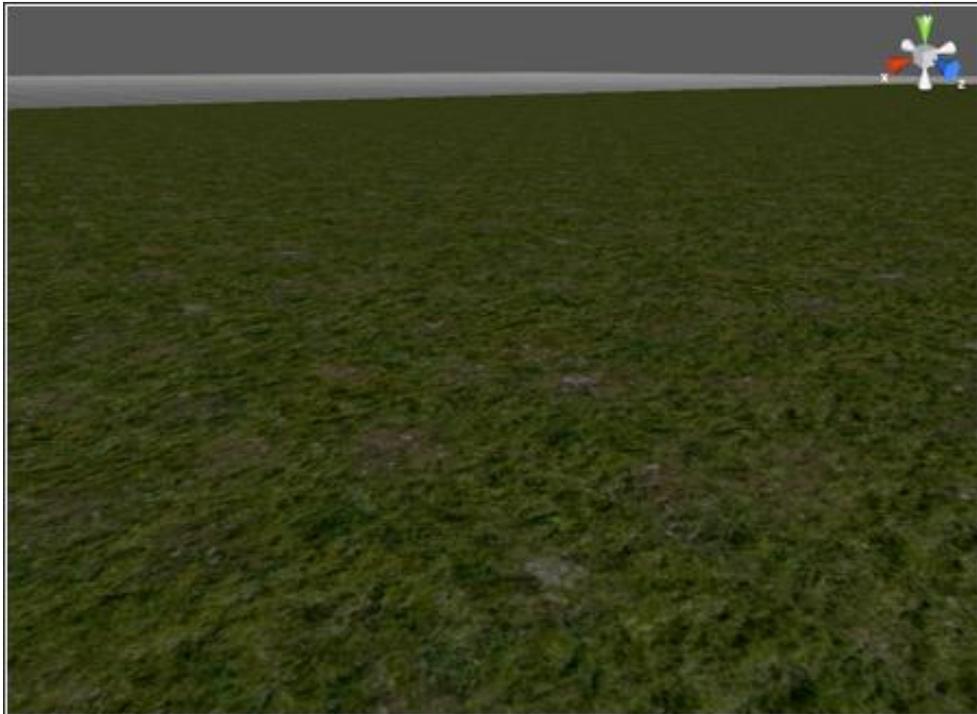


Figura 4.7. Generador de Terrenos.

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 4.8.



Figura 4.8. 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 4.9.



Figura 4.9. 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 indicamos 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 4.10. Estructura de la interfaz visual del videojuego hecho con Unity.

La Figura 4.10 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 niños.

4.1.4.1 Componentes

Son programas independientes entre sí que poseen sus propias clases dentro del core (alma) 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 como se muestra en la Figura 3.4 de la sección 3.3.4.

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 4.11):

- El cuarto del juego.
- La mesa para el juego.
- La palanca para el juego.

- Las figuras geométricas de las fichas del juego.
- La torre de fichas del juego.



Figura 4.10. 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:

- **GUI2D:** Esta clase se utiliza para manejar objetos en 2D.

- **StartGame:** Esta clase se utiliza para manejar todos los elementos del videojuego como: la palanca, la torre de fichas, la regleta de fichas, etc.
- **HandGrip:** Esta clase se utiliza para manejar la palanca de activación de las fichas del juego.
- **Menu:** Esta clase se utiliza para manejar todos los elementos que conforman el menú principal, como son las opciones de: a) Cómo Jugar; b) Empezar Juego; c) Ver Puntaje; d) Salir.
- **Piece:** Esta clase se utiliza para manejar cada una de las piezas del juego que son figuras geométricas.
- **Rotation:** Esta clase se utiliza para controlar la rotación de las piezas del juego.
- **TimeCount:** Esta clase se utiliza para controlar el tiempo de duración del juego en función del nivel de dificultad escogido en el mismo.
- **Scores:** Esta clase se utiliza para controlar el puntaje del juego obtenido por el usuario.
- **ScoreScreen:** Esta clase se utiliza para publicar los puntajes en la pantalla del monitor en una ventana.
- **ObjectType:** Esta clase se utiliza para controlar el tipo de figura geométrica que aparece en el juego.
- **LevelCount:** Esta clase se utiliza para controlar el número de aciertos obtenidos por el jugador.
- **FallCount:** Esta clase se utiliza para controlar el número de fallas obtenidas por el jugador.

- ***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 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 para la Web generándose un archivo ejecutable y un applet respectivamente, que permite correr la aplicación con todos sus elementos, como se muestra en la Figura 4.11 y en la Figura 4.12.

4.3 Desarrollo e Implantación de Componentes para la el JDRA3D

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 4.11. Ventana de la Aplicación del Juego en ambiente Windows.

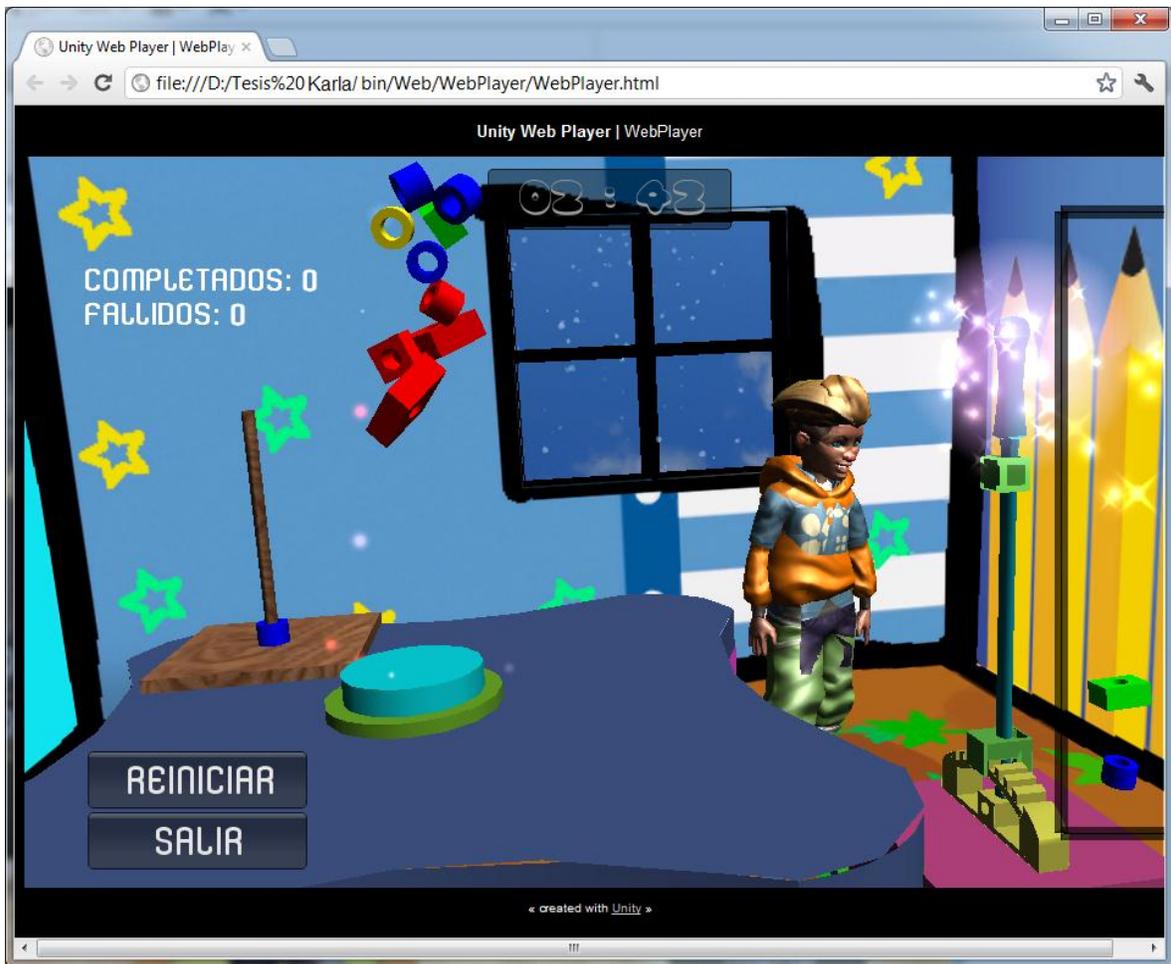
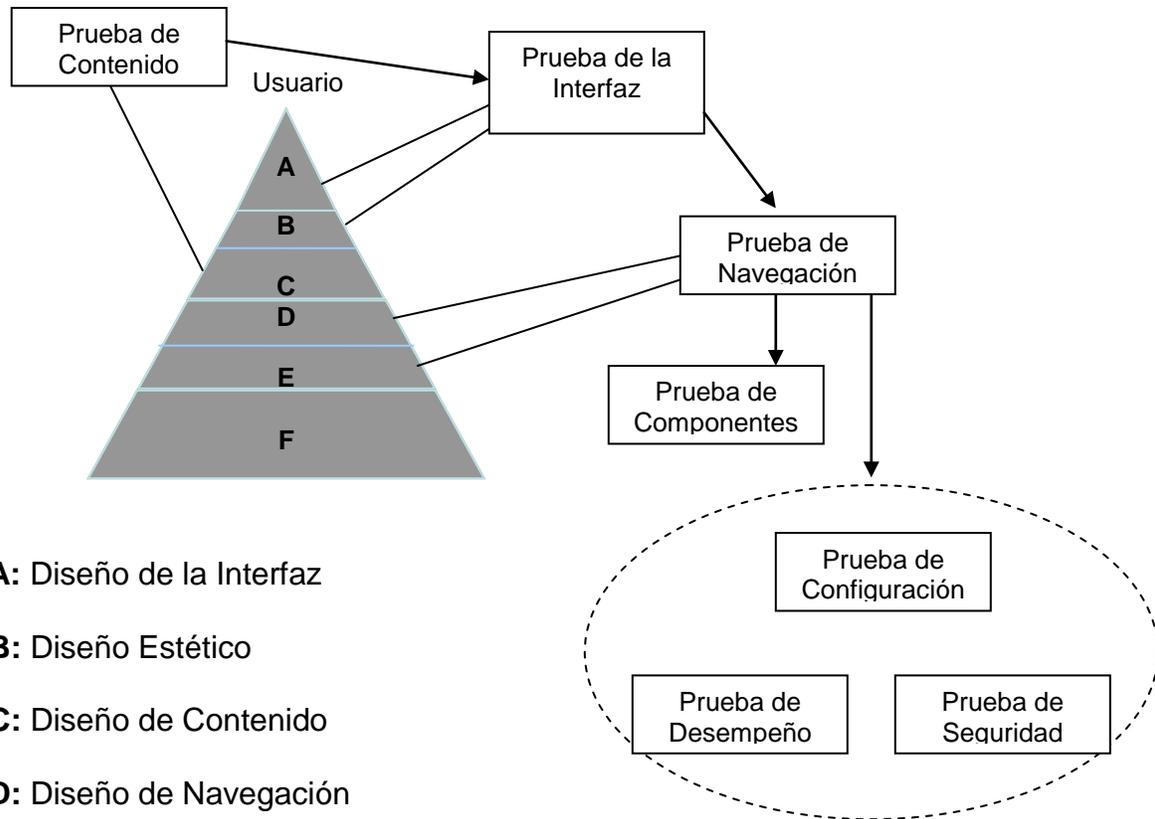


Figura 4.13. Ventana de la Aplicación del Juego en ambiente Web.

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.



- A:** Diseño de la Interfaz
- B:** Diseño Estético
- C:** Diseño de Contenido
- D:** Diseño de Navegación
- E:** Diseño Arquitectónico
- F:** Diseño de Componentes

Figura 4.15: Proceso de Prueba²⁶

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:

²⁶ 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 como para la Web; que en este caso es un 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 logrando 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 tomando 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 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) Cómo Jugar; c) Empezar Juego; d) 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 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 la Web.

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 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 MVC respondió a varias condiciones de carga y soportó adecuadamente las transacciones en un ambiente local tipo desktop, con un solo usuario a la vez. En este apartado no se aplica la fórmula de prueba de aplicaciones Web y distribuidas propuesta en el libro de Roger Pressman de Ingeniería de Software, que es:

$$P = N \times T \times D^{27}$$

Donde:

- **N**, es el número de usuarios concurrentes.
- **T**, es el número de transacciones en línea por usuario por unidad de tiempo.
- **D**, es la carga de datos procesada por el servidor por transacción

4.5 Deployment de la Aplicación

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

²⁷ Fórmula tomada del Capítulo 20 “Cómo Probar Aplicaciones Web” – Ingeniería Web – Pressman

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

CAPÍTULO V

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 75%, el 25% 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 cámara principal del juego permanece fija en los ejes x, y, z, durante todo el proceso de la navegación y resolución de los retos, considerando que los usuarios del juego son niños y que pueden perder la atención y el foco del escenario y de los objetos que están dentro, que fue lo que pasó en las primeras pruebas de desempeño que se realizaron.

- Al final del presente trabajo se puede apreciar que la correcta utilización del game engine Unity, fue posible crear un videojuego con excelentes características de funcionalidad, navegabilidad, desempeño y compatibilidad.

- En la presente aplicación de un videojuego en 3D se utilizó la metodología OOHDM con UML, debido a que es una metodología orientada al diseño y desarrollo de aplicaciones multimedia desktop y web, 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 de los autores 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 o Web 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 y de diseño

gráfico, 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 DCC 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

Libros:

- PRESSMAN Roger, Ingeniería de Software Un Enfoque Práctico, Mc. Graw Hill, Madrid – España, 2002.
- LAMARCA María Jesús, Hipermedia/Multimedia. Universidad Complutense de Madrid – 2006.
- GARCÍA RUBIO Ramón, QUIRÓS SUÁREZ Javier, GONZÁLEZ Santiago Martín, SANTOS GALLEGO Ramón, FERNÁNDEZ MORÁN Samuel;Diseño Gráfico de Contenidos para Internet, Pearson Prentice Hall
- BLACKMAN Sue, Beginning 3D Game Development with Unity, Apress, New York – USA, 2011.
- SCHNEIDER Electric Instituto, Manual de Formación Unity Pro, Instituto Schneider Electric, Barcelona – España, 2008.
- KELLER Eric, PALAMAR Todd, HONN Anthony, Mastering Autodesk Maya 2011, Sybex, USA, 2010.

Web:

- Página Oficial de Unity (2012)

Disponible: <http://unity3d.com/>

- Arocena, F. (Abril 2003), Crea tu Página Web

Disponible: <http://wmaestro.com/webmaestro/docs/portada.html>

- Consejos de estilo Gráfico para WWW (2006)

Disponible: <http://dmi.uib.es/people/acoca/estilo/index.html>

- Página Oficial de Maya (2012)

Disponible: <http://usa.autodesk.com/maya/>

GLOSARIO DE TÉRMINOS

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 en 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: Dice que cualquiera que redistribuye el software, con o sin cambios, debe dar la libertad de copiarlo y modificarlo. Garantiza que cada usuario tiene libertad.

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. Por ejemplo de un documento los metadatos serían: el título, el nombre del

autor, la fecha de creación y modificación, y un conjunto de palabras clave que identifiquen su contenido.

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 ó 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.