



UNIVERSIDAD NACIONAL DE COLOMBIA

**MODELO DE UN ENTORNO VIRTUAL
INTELIGENTE BASADO EN LA PERCEPCIÓN
Y EL RAZONAMIENTO DE SUS ELEMENTOS
CON UN PERSONAJE PARA LA
GENERACIÓN DE REALISMO**

Sandra Patricia Mateus Santiago

Universidad Nacional de Colombia
Facultad de Minas, Departamento de Ciencias de la Computación y la Decisión
Medellín, Colombia
2015

**MODELO DE UN ENTORNO VIRTUAL
INTELIGENTE BASADO EN LA
PERCEPCIÓN Y EL RAZONAMIENTO DE
SUS ELEMENTOS CON UN PERSONAJE
PARA LA GENERACIÓN DE REALISMO**

Sandra Patricia Mateus Santiago

Tesis Doctoral presentada como requisito parcial para optar al título de:

Doctor en Ingeniería – Sistemas e Informática

Director

John Willian Branch Bedoya, Ph.D

Universidad Nacional de Colombia
Facultad de Minas, Departamento de Ciencias de la Computación y la Decisión
Medellín, Colombia
2015

**MODEL OF AN INTELLIGENT VIRTUAL
ENVIRONMENT BASED ON PERCEPTION
AND REASONING OF ITS ELEMENTS WITH
A CHARACTER FOR THE GENERATION OF
REALISM**

Sandra Patricia Mateus Santiago

Doctoral thesis submitted as partial requirement for the degree of:
Doctor in Engineering – Systems and Informatics

Supervisor
John Willian Branch Bedoya, Ph.D

Universidad Nacional de Colombia
Facultad de Minas, Departamento de Ciencias de la Computación y la Decisión
Medellín, Colombia
2015

A Dios, por estar siempre presente en mis proyectos.

A mi Mami y a mi Abuelita, las mujeres que son los pilares de mi vida y me motivan a seguir adelante día a día!

Agradecimientos

Mis más sinceros agradecimientos al profesor John Willian Branch, por ser mi gran orientador en toda mi vida académica e investigativa.

Al Politécnico Colombiano Jaime Isaza Cadavid, por todo su apoyo para realizar mis estudios de Doctorado.

A los profesores Luciana Nedel de la UFRGS y Demetrio Ovalle de la UNAL, por brindarme importantes aportes para el desarrollo de mi Tesis.

A mi familia y amigos, los cuales siempre me brindaron su apoyo y su buena energía.

Resumen

Los Entornos Virtuales Inteligentes (EVI) están compuestos por la unión de elementos tomados de dos importantes áreas de la computación como son la Realidad Virtual y la Inteligencia Artificial, las cuales tienen múltiples aplicaciones en diferentes campos de investigación. A su vez, un EVI debe estar compuesto por los siguientes niveles: geométrico, cinemático, cognitivo y comportamental. Las limitaciones en un EVI se observan en el realismo que se obtiene a través de las sensaciones que emula el computador y que son percibidas por el usuario a través de sus sentidos. Dicho realismo, recae también, en el comportamiento e interacción con los elementos de un Entorno Virtual (percepción y razonamiento), necesario en aplicaciones de interés científico, de tipo militar, entrenamientos en medicina, en la preservación de patrimonio cultural, en la educación, en los videojuegos, entre otras.

En esta tesis se propone un modelo de Entorno Virtual Inteligente, que basado en la percepción y el razonamiento de sus elementos con un personaje, permita generar realismo. Para alcanzar este objetivo, se propone inicialmente un modelo geométrico y cinemático como base necesaria para el modelo de Entorno Virtual Inteligente y con ello, incrementar el desempeño del EVI en los niveles comportamental y cognitivo. En estos niveles, se implementan posteriormente, diferentes técnicas de Inteligencia Artificial aplicadas a la percepción y el razonamiento y después de comparaciones a través de diferentes métricas, se escoge la Red Neuronal Artificial que es en la que se apoya principalmente el modelo. A lo largo del desarrollo de la tesis se realizó un trabajo experimental, que valida que el modelo propuesto funciona flexible y adecuadamente, independiente del área de aplicación.

Palabras clave: Entorno Virtual Inteligente, Personaje, Inteligencia Artificial, Realidad Virtual.

Abstract

Intelligent Virtual Environments (IVE) are composed by the union of elements taken from two important areas of computing, as such the Virtual Reality and the Artificial Intelligence, which have multiple applications in various research fields. At the time, an IVE should be composed of the following levels: geometric, kinematic, cognitive and behavioral. The IVE limitations are observed in the realism obtained through of the senses that a computer emulate and the user perceives these sensations through your senses. That realism, also falls, in the behavior and interaction with the elements of a virtual environment (perception and reasoning), realism necessary in applications of scientific interest such as; military style, training in medicine, cultural heritage, education, videogames, among others.

In this thesis, a model of Intelligent Virtual Environment is proposed, which based on the perception and reasoning of its elements with a character, it can achieve realism. To achieve this goal, initially a geometric and kinematic model is proposed as a basis for the model of Intelligent Virtual Environment, and then increase performance of IVE in behavioral and cognitive levels. At these levels, subsequently are implemented, different AI techniques applied to perception and reasoning, and after comparisons across different metrics, is chosen an Artificial Neural Network to supports the model. Throughout the development of the thesis an experimental work is made, which validates that the proposed model works flexible and properly, independent of the application area.

Keywords: Intelligent Virtual Environment, Character, Artificial Intelligence, Virtual Reality.

Contenido

	Pág.
1. Introducción	1
1.1 Motivación	2
1.2 Marco Conceptual	2
1.2.1 Entornos Virtuales.....	2
1.2.2 Entornos Virtuales Inteligentes (EVI).....	3
1.3 Revisión de la Literatura	4
1.3.1 Semántica en Entornos Virtuales	5
1.3.2 Redes Neuronales Artificiales y Computación Evolutiva en Entornos Virtuales	6
1.3.3 Agentes Inteligentes en Entornos Virtuales	6
1.3.4 Entornos Virtuales Inteligentes centrados en un Personaje.....	8
1.4 Planteamiento del Problema.....	11
1.5 Hipótesis de Investigación	13
1.6 Objetivos	13
1.6.1 Objetivo General	13
1.6.2 Objetivos Específicos.....	13
1.7 Contribución de la Tesis.....	13
1.8 Divulgación de Resultados.....	14
1.9 Organización de la Tesis	15
2. Propuesta de un Modelo Geométrico y Cinemático de un Entorno Virtual, a Partir de la Caracterización de sus Elementos Relevantes	17
2.1 Modelo Geométrico	18
2.1.1 Definición de la Cantidad de Polígonos.....	19
2.1.2 Normales de los Polígonos.....	20
2.1.3 Triangulación de Polígonos de más de Tres Lados.....	21
2.1.4 Corrección de Geometría sin Pliegue Principal	22
2.1.5 Texturas Cuadradas.....	23
2.2 Modelo Cinemático.....	23
2.2.1 Características Cinemáticas de los Objetos.....	23
2.2.2 Pivote y Orientación.....	24
2.2.3 Articulaciones	24
2.2.4 Cadenas de Cinemática Relativa e Inversa	27
2.2.5 Partículas	28
2.2.6 Campos.....	29
2.3 Conclusiones	30
3. Incorporación de Técnicas de Inteligencia Artificial al Entorno Virtual	31
3.1 Percepción y Razonamiento	32

3.2	Entorno Virtual Inteligente con Redes Neuronales Artificiales	34
3.2.1	Incorporación de una RNA en un EVI para la detección de problemas visuales.....	34
3.2.2	Incorporación de una RNA en un EVI en un Ambiente Laboral	40
3.3	Entorno Virtual Inteligente con Computación Evolutiva.....	42
3.3.1	Incorporación de un Algoritmo Genético al Entorno Virtual.....	43
3.3.2	Incorporación de una Estrategia Evolutiva al Entorno Virtual.....	47
3.4	Entorno Virtual Inteligente con Agentes Inteligentes	49
3.4.1	Especificación del Sistema.....	50
3.4.2	Diseño de la Arquitectura.....	52
3.4.3	Diseño Detallado	53
3.5	Conclusiones	57
4.	Selección de la Técnica de Inteligencia Artificial para su Implementación en un Entorno Virtual Inteligente.....	58
4.1	Evaluación Cualitativa	58
4.1.1	Interactividad	58
4.1.2	Conectividad	59
4.1.3	Ubicuidad	59
4.1.4	Adaptabilidad.....	59
4.1.5	Sensibilidad al Contexto	60
4.1.6	Especificaciones del Equipo.....	60
4.1.7	Latencia o Tiempo de Respuesta del Entorno	61
4.1.8	Compleitud	63
4.1.9	Escalabilidad.....	64
4.2	Conclusiones	65
5.	Validación del Modelo de Entorno Virtual Inteligente Mediante un Prototipo de Representación Visual.....	66
5.1	Entorno Virtual Inteligente como Recurso Orientado al Aprendizaje	66
5.2	Conclusiones	70
6.	Conclusiones y Trabajos Futuros.....	73
6.1	Trabajo Futuro	74

Lista de Figuras

	Pág.
Figura 1.1: Semántica aplicada en Juegos y Simulaciones (Tutenel & Bidarra, 2008)	6
Figura 1.2: Plataforma Paladin (Jia & Zhenjiang, 2007)	6
Figura 1.3: Modelo por capas en un EVI (Latoschik & Fröhlich, 2007)	7
Figura 1.4: Agentes Sociales Inteligentes (Mascarenhas, Enz, & Paiva, 2009)	7
Figura 1.5. Modelo de Entorno Virtual para evacuación al fuego (Xi & Smith, 2014).....	8
Figura 1.6: Discurso emocional (Decker, Sichman, Sierra, & Castelfranchi, 2009).....	9
Figura 2.1: Estructura de modelado de la computación gráfica (Funge, Tu, & Terzopoulos, 1999)	17
Figura 2.2. Modelo Geométrico y Cinemático	18
Figura 2.3: Elementos de un polígono.....	19
Figura 2.4: Ejemplo de número de polígonos para una superficie de 180°	20
Figura 2.5: Renderizado por Hardware para una superficie de 5 polígonos	20
Figura 2.6: Normales adecuadas en polígonos	21
Figura 2.7: Normales inadecuadas en polígonos.....	21
Figura 2.8: Triangulación de Polígonos: a. Polígono sin triangular. b. Polígono Triangulado	22
Figura 2.9: Corrección de Geometría. a. Polígono con pliegue principal. b. Polígono sin pliegue principal.	22
Figura 2.10: Articuciones en el brazo de una retroexcavadora. a. Uso de pivotes. b. Uso de articuciones.	25
Figura 2.11: Rotación del brazo. a. El brazo rota 45 grados sobre su pivote. b. La articulación principal rota 45 grados. c. Desplazamiento de segmentos. d. No hay operación adicional con el uso de articulaciones	26
Figura 2.12: Movimientos para alcanzar el suelo. a. Rotación del brazo sobre su eje con pivotes. b. Rotación y traslación del brazo con pivotes. c. Rotación con articulaciones.....	26
Figura 2.13: Cadena de Cinemática Relativa. a. Estado inicial. b. Movimiento y rotación de la articulación.	27
Figura 2.14: Cadena de Cinemática Inversa.....	28
Figura 2.15: Ejemplo de un campo de gravedad.....	29
Figura 3.1: Modelo de Entorno Virtual Inteligente	33
Figura 3.2: Algoritmo RNA MLP	36
Figura 3.3: Evolución del error de aprendizaje de la RNA MLP.....	37
Figura 3.4: Algoritmo de la RNA Función de Base Radial (RBF)	37

Figura 3.5: Ojo en 3D a. Ojo Normal. b. Ojo Miope. c. Ojo Hipermetrope. d. Ojo con Astigmatismo.....	39
Figura 3.6: a. Formulario para la anamnesis. b. Diagnóstico. c. EVI de un problema visual.....	40
Figura 3.7: Entorno Virtual Inteligente para señales de advertencia en un ambiente laboral	41
Figura 3.8: Entorno Virtual desarrollado con UDK	41
Figura 3.9: a. Personaje identificando el riesgo del fuego. b. Personaje identificando el piso húmedo.	42
Figura 3.10: EVI con la RNA.....	42
Figura 3.11: Algoritmo Genético implementado en el Entorno Virtual	44
Figura 3.12: Cruce y Mutación en el Algoritmo Genético	45
Figura 3.13: Algoritmo Genético en el Entorno Virtual	46
Figura 3.14: EVI con el AG	46
Figura 3.15: Búsqueda en la Estrategia Evolutiva	48
Figura 3.16: Función Objetivo de la Estrategia Evolutiva	48
Figura 3.17: EVI con la EE.....	49
Figura 3.18: Fases de la Metodología Prometheus (Padgham & Winikoff, 2005).....	50
Figura 3.19: Diagrama del Entorno Virtual basado en Prometheus.....	51
Figura 3.20: Agente Personaje.....	53
Figura 3.21: Algoritmo de Colonia de Hormigas (Brownlee, 2012).....	54
Figura 3.22: Función de Búsqueda en el algoritmo de Colonia de Hormigas	55
Figura 3.23: EVI con Agentes	56
Figura 3.24: Selección de la Técnica de IA desde el editor de UDK.....	57
Figura 5.1. Metodología SAM1 (Allen, 2012)	67
Figura 5.2. Diseño del EVI basado en la metodología SAM1	68
Figura 5.3. Arrastre de las soluciones al tablero.....	69
Figura 5.4. Fin del Juego	70

Lista de Tablas

	Pág.
Tabla 1-1: Comparación de la Revisión de Literatura	11
Tabla 3-1: Clasificación de RNA por tipo de aprendizaje	35
Tabla 3-2: Comparación de los resultados obtenidos entre MLP y RBF	38
Tabla 3-3: Percepciones y acciones a ejecutar del personaje en el Entorno Virtual	41
Tabla 4-1: Evaluación cualitativa del Entorno Virtual Inteligente.....	60
Tabla 4-2: Latencia basada en repeticiones para la RNA	62
Tabla 4-3: Latencia basada en repeticiones para el AG	63
Tabla 4-4: Latencia basada en repeticiones para los Agentes	63
Tabla 4-5: Escalabilidad en la RNA	64
Tabla 4-6: Escalabilidad en el AG.....	65
Tabla 4-7: Escalabilidad en Agentes.....	65
Tabla 5-1. Comparación del Modelo Propuesto	70
Tabla 0-1: Evaluación de la Latencia en la RNA.....	82
Tabla 0-2: Evaluación de la Latencia en el AG	83
Tabla 0-3: Evaluación de la Latencia en los Agentes Inteligentes	84

1. Introducción

Los Entornos Virtuales Inteligentes (EVI) están compuestos por la unión de dos importantes áreas de la computación: la Realidad Virtual y la Inteligencia Artificial (IA), las cuales tienen múltiples aplicaciones que les permite ser ampliamente utilizadas en campos como la industria, la educación y la investigación, entre otras; además, su acción conjunta les permite alcanzar un completo nivel de aplicabilidad en diversas áreas.

La Realidad Virtual se conoce como la tecnología computacional que simula un ambiente tridimensional que en su mayoría son experiencias visuales en una pantalla o en un dispositivo estereoscópico que en algunas ocasiones incluyen sensaciones auditivas; cuando a un entorno virtual se le incorpora la Inteligencia Artificial alcanza una nueva característica dando como resultado un sistema de distribución de eventos de forma asincrónica la cual es la que lo diferencia de un Entorno Virtual convencional y de ahí su virtud de alta aplicabilidad (Lozano & Calderón, 2004).

En la actualidad, la mayoría de imágenes que se pueden observar en los principales medios de comunicación contienen elementos que son producto de la realidad virtual, como por ejemplo en los comerciales en los cuales no todos los personajes son actores, o en los documentales donde muestran todos los ángulos de estructuras o seres, o más específicamente, las películas animadas; otra muestra más concreta y clara son los videojuegos que dejan en promedio billones de dólares de ganancias al año, y si a todo eso se le agrega IA, se podría alcanzar un nivel de aplicabilidad más amplio, obteniendo un producto capaz de adaptarse a múltiples objetivos.

Las limitaciones en un EVI se observan en el realismo que se obtiene, y esto recae a su vez, en el comportamiento e interacción con los elementos de un Entorno Virtual, realismo necesario en aplicaciones de interés científico de tipo militar, entrenamientos en la medicina, en la herencia cultural, entre otras. Por todo lo anterior, se propone un modelo de EVI, que aplicado al comportamiento de sus elementos, permita encontrar un adecuado realismo.

1.1 Motivación

En esta tesis, se considera un campo denominado “Entornos Virtuales Inteligentes”, el cual está formado por el creciente traslape de las tecnologías involucradas en el desarrollo de gráficos interactivos 3-D en tiempo real y las técnicas de la Inteligencia Artificial (Aylett & Cavazza, 2001).

El interés por este campo, se puede observar desde diferentes enfoques, la combinación de ideas y tecnologías de Entornos Virtuales y la IA, son un campo muy activo, con muchos grupos de investigación diferentes preocupados en diferentes aspectos (Cavazza, y otros, 2005). Uno de esos aspectos, es el incremento del poder de procesamiento disponible para renderizado, lo que permite que se logre en un Entorno Virtual, un adecuado realismo. Sin embargo, aunque un entorno sea visualmente atractivo, si es estático y no tiene muchos cambios en los comportamientos, la experiencia de inmersión es limitada, por lo que se hace necesario crear entornos de interés dinámico, en las que un usuario en un entorno dado, tenga una sensación real de interacción con éste. La disponibilidad del poder de procesamiento y del deseo de producir entornos más dinámicos e interesantes, llama mucho la atención en la comunidad científica para aplicaciones en diferentes campos, como la medicina, la psicología, los museos interactivos, la industria, entre otros.

Un requisito similar para el soporte a usuarios, es el crecimiento del e-commerce en la web. Mientras que el contenido 3-D es poco común, es también claro que los usuarios requieren asistencia inteligente en muchos casos; las interfaces de personajes en 2-D son el comienzo para dar forma a las “Talking Heads” 3-D, las cuales son extendidas a nuevos medios tal como la televisión digital interactiva. De forma similar, entornos de chat distribuidos a larga escala, están migrando a gráficos 3-D y encontrando la necesidad de inteligencia como apoyo a las poblaciones de sus usuarios avatar.

1.2 Marco Conceptual

1.2.1 Entornos Virtuales

Independientemente del entorno interesado en su modelización virtual, desde un punto de vista exclusivamente computacional, los entornos virtuales están compuestos por tres modelos que conforman la aplicación informática en tiempo real: un modelo geométrico, un modelo comportamental (en tiempo de ejecución) y un modelo de interacción con el usuario (Lozano & Calderón, 2004).

- **Modelo geométrico:** Es el encargado de atender al bajo nivel gráfico, donde por un lado, los distintos tipos de formatos gráficos empleados, junto con el modelo interno utilizado para el lanzamiento de las ordenes de dibujado o render (grafo de escena, ordenamiento

de primitivas gráficas o display lists, etc), resumen las propiedades elementales para la visualización de cualquier Entorno Virtual.

Tradicionalmente, estos entornos son descritos con base en la colección de primitivas poligonales, líneas, texto, superficies, etc., que constituyen la información espacial visualizada en 3-D. Gracias a la implantación de hardware dedicado al procesamiento de dicha información visual, los sistemas gráficos de tiempo real, hoy en día, son capaces de dibujar millones de polígonos por segundo, lo que resulta ser una tasa de rendimiento bastante razonable, atendiendo a las numerosas aplicaciones gráficas aparecidas en los últimos años.

- **Modelo Comportamental:** Este modelo trata el comportamiento del entorno virtual como aplicación informática en tiempo de ejecución, es decir, atiende al comportamiento dinámico (cambios) de todos los objetos 3-D que el Entorno Virtual contiene. Un comportamiento considerado, típicamente se refiere al mantenimiento en tiempo de ejecución de propiedades elementales, como posición, orientación, color, etc. de todo elemento (objeto u actor) situado en el entorno virtual; el otro, sería el comportamiento autónomo típicamente asociado a la toma de decisiones de distintos Entornos Virtuales Inteligentes, u objetos autónomos que pueden formar parte de la simulación.
- **Interacción con el usuario:** Una de las principales expectativas generadas en la mayoría de entornos virtuales, es la libre interacción con los elementos 3-D del entorno. Los entornos virtuales clásicos incluyen un modelo de interacción básico dentro de la jerarquía de nodos de su grafo de escena. Este modelo, centrado de nuevo en el usuario, es representado por la información del punto de vista en el espacio 3-D, por la información del movimiento de un objeto, o por la información de las características de un objeto relacionadas con la física en el espacio 3-D, manteniendo información proveniente de algún dispositivo que registre posición, orientación del usuario y rutas de navegación del mismo. Esto es bastante útil a la hora del dibujado de la escena, ya que se tiene la información actualizada del usuario y la imagen será calculada desde su punto de vista. Por otro lado, resulta poco efectivo si el usuario quiere interactuar con una entidad situada fuera de su campo de visión.

1.2.2 Entornos Virtuales Inteligentes (EVI)

El énfasis de los EVI (Aylett & Luck, 2000) reside en incrementar las capacidades comportamentales e interactivas de los entornos virtuales clásicos. Esto se consigue mediante la incorporación de técnicas de IA situadas e interactuando con el sistema gráfico. Aunque existen diferentes técnicas de animación (ej: interpoladores VRML o nodos de secuencia de OpenGL-Performer) y sencillos modelos comportamentales que pueden ayudar a dotar de cierto dinamismo a los entornos clásicos, en la mayoría de ocasiones, su

naturaleza de guión o plan precompilado (off-line) sólo logrará mantener el interés del usuario durante un tiempo bastante limitado.

Según (Lozano & Calderón, 2004) hay tres formas de incrementar las capacidades interactivas de un entorno virtual se darían mediante la integración de sistemas de IA:

- Incremento de la actividad del Entorno Virtual: Esto se puede lograr mediante la integración de motores de razonamiento simbólico que interpreten, por ejemplo, las acciones del usuario. La integración entre un motor gráfico y un motor de razonamiento se puede alcanzar a través de un modelo de eventos que funciona en ambas direcciones y que se beneficia de los mecanismos incorporados al motor gráfico para tratar con eventos asíncronos.
- Necesidad de una representación del conocimiento: La integración de personajes inteligentes 3-D en estos entornos, ejemplifica la necesidad de una representación del conocimiento y los requerimientos que esto conlleva. Los EVI deben manejar información a distintos niveles: el nivel bajo típicamente recogerá la información geométrica (listas de vértices, esferas o cajas envolventes, etc.), accesible en todo momento para cualquier agente de simulación y el nivel alto (semántico) permitirá a los elementos del entorno, resolver distintos problemas en el mismo. El mantenimiento del modelo semántico asociado al entorno virtual 3-D es un paso necesario para la integración de distintos elementos situados en un mundo compartido (Lozano M. , 2003).
- Alternativa a la simulación física: Un ejemplo en este sentido es la idea de la Realidad Aumentada (RA) como una nueva forma de experimentar entornos virtuales.

La evolución natural de los EVI, pasa por proporcionar entornos de simulación adecuados a mundos 3-D de naturaleza cambiante y en este sentido, los entornos dirigidos por eventos proporcionan actualmente un marco adecuado para la investigación y desarrollo en EVI. Los modelos comportamentales y de comunicación clásicos, no satisfacen suficientemente estos requerimientos, debido principalmente a su alta dependencia con el modelo geométrico (grafos de escena), el cual está claramente orientado a una rápida visualización centrada en el usuario, dejando de lado importantes cuestiones comportamentales y de interacción.

1.3 Revisión de la Literatura

Un aspecto importante de la perspectiva centrada en el usuario en un Entorno Virtual Inteligente, es una integración completa de una técnica de IA en éste, dada la dificultad de

desarrollar en tiempo real dicha técnica, lo cual debe mantener un equilibrio para efectos de visualización en la interacción del usuario con los Entornos Virtuales.

Uno de los primeros EVI, fue el sistema de programación “Oz” dentro del software DIVE VR (Axling, Haridi, & Fahlen, 1996). Oz soportaba la programación basada en reglas, la cual era usada como un lenguaje de programación de alto nivel para la mejora del desempeño en un Entorno Virtual. Posteriormente Codognet (1999), desarrolló un paquete genérico basado en reglas llamado VRCC, el cual se integraba en VRML y se usaba para definir el comportamiento de agentes a través de reglas; ambos sistemas dependían de la Programación Lógica basada en Reglas CLP (Constraint Logic Programming). De ahí que el CLP fuese una técnica adecuada para soportar EVI, fundamentalmente porque calculaba soluciones rápidas y suficientes para los ciclos de interacción y proveía soluciones incrementales.

Fue así como fueron surgiendo otras aproximaciones de IA, tales como el ajuste heurístico y la búsqueda local, las cuales son alternativas para mecanismos básicos usados en programación basada en reglas, como lo realizó Codgnet (2001), el cual afirmaba que el ajuste heurístico era un solución apropiada y rápida, cuando una solución pre-definida era perturbada por la acción de un simple objeto. En ese mismo año, Calderón y Cavazza (2001), desarrollaron un EVI usando GNU Prolog y el motor de juego Unreal Tournament; Prolog producía una solución en términos de la posición de los objetos del entorno virtual desarrollado.

La IA adquiere gran relevancia en la personificación y también en la robótica, sin dejar de lado la perspectiva gráfica, la cual es de gran importancia en todo lo que se refiere a Realidad Virtual. Es así como trabajos actuales, como el de Kaigom y Roßmann (2014) se enfocan en la simulación de robots inteligentes que pueden interactuar con su entorno, a través de la simulación de la capacidad de acomodar fuerzas externas a través de entornos 3-D y luego llevados a pruebas físicas.

1.3.1 Semántica en Entornos Virtuales

Desde otro enfoque, Tuteneel & Bidarra (2008) aplican la Semántica en Juegos y simulaciones (Figura 1.1), lo que les permite crear una ontología robusta para la generación automática de un mundo virtual, pese a que no incorpora Planificación. En esta misma línea, Bonis (2009) desarrolló una plataforma para museos virtuales, con una descripción semántica del contenido para interactuar con el museo desarrollado.



Figura 1.1: Semántica aplicada en Juegos y Simulaciones (Tutenel & Bidarra, 2008)

1.3.2 Redes Neuronales Artificiales y Computación Evolutiva en Entornos Virtuales

Escasos trabajos, incorporan Redes Neuronales Artificiales y/o Computación Evolutiva como técnicas de IA en Entornos Virtuales: Jia y Zhenjiang (2007) desarrollaron la Plataforma “Paladin” (Figura 1.2), la cual utiliza agentes colaborativos y una técnica poco utilizada en los EVI como es la Neuroevolución, la cual en este trabajo, combina las Redes Backpropagation con los Algoritmos Evolutivos, lo que crea una metodología muy adecuada, sin embargo, el entorno de este trabajo es en 2D y la colaboración de los agentes es limitada.

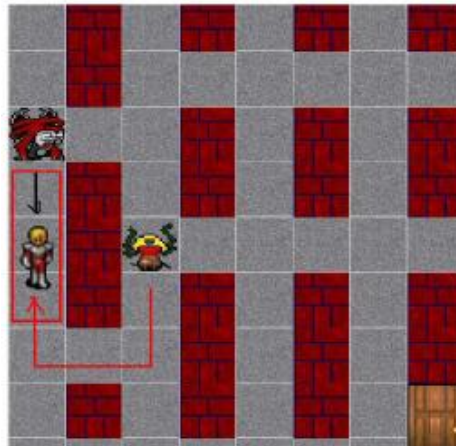


Figura 1.2: Plataforma Paladin (Jia & Zhenjiang, 2007)

1.3.3 Agentes Inteligentes en Entornos Virtuales

De todas las técnicas de IA que existen, los Agentes y Sistemas Multiagentes, han sido los más utilizados en Entornos Virtuales: En esta línea, DosSantos y Osorio (2004) trabajan con Agentes Virtuales Inteligentes orientados hacia la enseñanza – aprendizaje, basados en reglas simples y con navegación asistida y poca recuperación de información. Latoschik &

Fröhlich (2007) utilizan reflexión semántica para un EVI (Figura 1.3), utilizando un modelo por capas, incorporándola así en agentes inteligentes, aunque no muestran una metodología clara en la aplicación de los EVI.

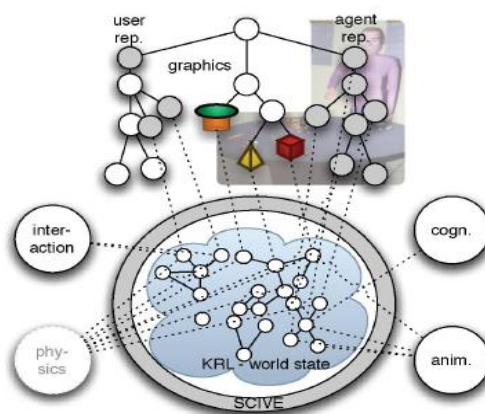


Figura 1.3: Modelo por capas en un EVI (Latoschik & Fröhlich, 2007)

Mang-Xian & Hai-ming (2008) en su trabajo, utilizan agentes colaborativos como un modelo de grupos de agentes virtuales, utilizando el protocolo KQML; los problemas que muestran los autores utilizando esta metodología, son el tiempo computacional, el control de los objetos y la autonomía de los agentes. Mascarenhas *et al.* (2009) trabajan lo que denominan Agentes Sociales Inteligentes (Figura 1.4), para identificar diferencias en dos culturas; ellos de forma general, utilizan una tupla de cinco elementos, que resume las características esenciales en sus agentes, sin embargo, este trabajo se ve limitado a la integración de una cultura adicional y se ve afectado por la influencia de cada componente.



Figura 1.4: Agentes Sociales Inteligentes (Mascarenhas, Enz, & Paiva, 2009)

Siguiendo con los agentes, el artículo de Luzardo y Hernández (2010), es una apropiada introducción a la temática de los EVI y a lo que se denomina los Humanos Virtuales Autónomos, sin embargo en este trabajo no se especifica ningún modelo.

En el tema de sistemas tutoriales inteligentes orientados hacia entornos virtuales: Buche y Querrec (2011) desarrollaron un sistema tutorial inteligente con algunos entornos virtuales en 3-D, en donde integran Pegase (un sistema tutorial genérico y adaptable), con sistemas multi-agentes para tomar decisiones de tipo pedagógico; como un sistema tutorial, es un trabajo muy completo, pero solo está enfocado al sector educativo. Clemente *et al.* (2014) presentan un sistema tutorial inteligente como parte de un EVI para entrenamiento, en el que el modelo de estudiante propuesto, emplea una ontología como un módulo diagnóstico pedagógico y un módulo de solución de conflictos, con el objetivo de inferir los objetivos de aprendizaje que ha adquirido el estudiante, probándolo finalmente en un laboratorio virtual de biotecnología en 3-D.

Por otra parte, el uso de agentes se utiliza en vehículos inteligentes para desarrollar diferentes entornos de estos, comenzando desde la simulación y finalizando con pruebas físicas a full escala, haciendo uso de diferentes niveles de virtualización para sensores, agentes, escenarios y entornos (Kurt, Vernier, Biddlestone, Redmill, & Özgüner, 2014). Y en simulaciones de evacuación, lo cual es muy usado para simular emergencias humanas, usando un entorno virtual con tecnología de juegos y agentes inteligentes (Figura 1.5), (Xi & Smith, 2014) s.

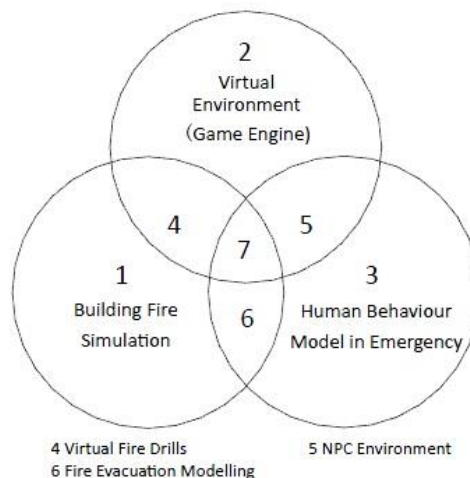


Figura 1.5. Modelo de Entorno Virtual para evacuación al fuego (Xi & Smith, 2014)

1.3.4 Entornos Virtuales Inteligentes centrados en un Personaje

A medida que crecía la capacidad computacional, fueron apareciendo personajes sintéticos y humanos virtuales en los EVI, en los cuales el papel de la IA ha sido fundamental. Algunos ejemplos de personajes (Aylett & Cavazza, 2001), se pueden observar en: personajes no humanos (Sims, 1995), mecánicos (Prophet, 1996), ficticios como Creatures (Grand & Cliff, 1998), Woggles (Loyall & Bates, 1993) o Teletubbies (Aylett R., 1999); también animales como: pescados (Bandi & Thalmann, 1998), delfines (Martinho,

Paiva, & Gomes, 2000) o perros (Blumberg & Galyean, 1997); y humanos virtuales como Marilyn (Thalmann & Thalmann, 1998) y agentes pedagógicos (Rickel & Johnson, 2000).

Decker *et al.* (2009) proponen una técnica de interacción basada en reconocimiento de discurso emocional (Figura 1.6), en la cual permiten que el usuario tome parte en un diálogo con actores virtuales sin restricciones en estilo o expresividad. Las limitaciones de esta técnica, se encuentran en la definición de las categorías emocionales y el hecho de que su impacto depende de las consideraciones del género que definen, como el número de elementos que manejan.



Figura 1.6: Discurso emocional (Decker, Sichman, Sierra, & Castelfranchi, 2009)

Bee (2010) integra la mirada de usuarios en un sistema de narrativa para permitir interacciones entre los personajes. En este trabajo, se identifican dos grupos de usuarios: uno que está permanentemente con mirada fija al personaje y la otra muestra un comportamiento de mirada natural entre los ojos de los personajes, las caras y la escena; en el primero no hay una interactividad como tal, pero se mantiene la mirada en el personaje, mientras que en el segundo, se muestran diferentes comportamientos de miradas dependiendo de cómo el personaje esté interactuando. Los autores concluyen con que en una reunión social, no se dan diferencias significativas entre un grupo y el otro.

Whiting *et al.* (2010) proponen un framework para los niveles cognitivo y comportamental de un EVI, centrado en la acción del personaje a través de RNA y Agentes en un entorno 3-D. En este mismo sentido, Martínez *et al.* (2006) implementan un EVI basado en Agentes, en los cuales integran el aprendizaje a través de RNA.

Liu *et al.* (2011) desarrollan un framework para el modelado de humanos virtuales con un alto nivel de autonomía, a nivel comportamental y de movimiento en un entorno virtual; el framework incluye un módulo de percepción, un módulo de decisión y un módulo de control de movimiento autónomo. Gilbert y Forney (2015) perfeccionaron un tour guiado por un avatar en un almacén de ropa virtual en un mundo 3-D de Second Life, a través de

agentes virtuales y utilizando una variante robusta del Artificial Intelligence Markup Language (AIML) y teniendo como reto el Test de Turing.

Y finalmente, respecto a EVI distribuidos, se encuentra la propuesta de (Vic & Alghazzawi, 2014) los cuales crean una red global de Entornos Inteligentes distribuidos, basados en algunas tecnologías y frameworks MMO (Massively Multiuser Online) y con ello un sistema híbrido inteligente para incrementar la funcionalidad de Entornos Inteligentes y así fomentar la cooperación entre usuarios.

A continuación, se presenta la Tabla 1-1, en la cual se hace una comparación de algunos trabajos relacionados anteriormente y separados por temática, con el fin de brindar claridad en los aportes a realizar en esta Tesis.

Autores	Tema	Modelo Geométrico y Cinemático	Técnica de IA			Motor de Videojuego	Dim.
			RNA	CE	Agentes		
Jia & Zhenjiang, 2007	Neuroevolución	x	✓	✓	x	x	2-D
Mascarenhas <i>et al.</i> , 2009	Agentes Sociales Inteligentes	x	x	x	✓	x	3-D
Buche y Querrec, 2011	STI	x	x	x	✓	x	3-D
Clemente <i>et al.</i> , 2014	STI + EVI	x	x	x	✓	x	3-D
Kurt <i>et al.</i> , 2014	Vehículos Inteligentes	x	x	x	✓	x	3-D
Bee, 2010	Sistema de Narrativa	x	x	x	✓	x	3-D
Liu <i>et al.</i> , 2010	Framework de humanos virtuales – A*	x	x	x	x	x	3-D
Gilbert y Forney, 2015	Second Life - AIML	x	x	x	✓	x	3-D
Xi y Smith, 2014	EVI - Emergencias Humanas	x	x	x	✓	✓	3-D

Whiting et al., 2010	EVI – Personaje - Acción	x	✓	x	✓	x	3-D
Mang-Xian & Hai-ming, 2008	EVI - Aprendizaje	x	x	x	✓	x	2-D
Martínez et al., 2006	EVI - Aprendizaje	x	✓	x	✓	x	2-D
Vic & Alghazzawi, 2014	EVI Distribuidos	x	x	x	x	x	3-D

Tabla 1-1: Comparación de la Revisión de Literatura

La integración de EVI, es un problema raramente tratado desde el punto de vista del modelo computacional que soporta el entorno virtual. Dado que estos entornos están llamados ser los entornos de simulación de distintos tipos de EVI, deberán ser capaces de gestionar los nuevos requerimientos dinámicos impuestos. En este sentido, los modelos basados en el envío de recepción asíncrona de eventos proporcionan un modelo de simulación adecuado para la actual intersección entre la IA y los gráficos 3-D en tiempo real.

La mayoría de trabajos de EVI, desarrollados y reportados en la literatura hasta el momento, se han visto limitados por un modelo, que aplicado al comportamiento de los elementos de un EVI, permita encontrar un adecuado realismo. Este realismo, está basado en los elementos que permiten un desempeño inteligente tales como: la percepción, el aprendizaje, la comunicación mediante un lenguaje natural y el razonamiento. Todo lo anterior, así sea visto en forma individual, debe tener un proceso de planificación, para coordinar los niveles de control y animación en personajes, que se encuentra limitado en la literatura.

Con base en lo anterior, el desempeño inteligente se centra en determinar la mejor secuencia de acciones a ser ejecutadas en un EVI, tomando el éxito de los personajes y de los recursos del entorno. Por esta razón, es conveniente explorar un modelo que permita encontrar un adecuado realismo basado en la percepción y el razonamiento de un EVI con un personaje u objetos del mismo.

1.4 Planteamiento del Problema

A través del tiempo se ha podido evidenciar cómo han evolucionado la tecnología y las formas de comunicación, incluyendo las novedosas técnicas informáticas como la Realidad Virtual y la IA, que cada vez cogen más auge y abarcan campos diferentes a los relacionados estrictamente con el ámbito tecnológico. Es por eso que deja de ser estrictamente un apoyo para el desarrollo de actividades personales y se convierte en una herramienta de gran

utilidad en el desarrollo del campo profesional, incluyendo la medicina, el entretenimiento y la educación, entre otros.

En la última década, los entornos virtuales han alcanzado una gran calidad gráfica proporcionada por los adelantos de la tecnología a la par de la capacidad de detalle de los objetos y personajes que lo componen. Es por ello, que diferentes áreas de investigación han centrado su atención en el desarrollo de EVI que alcancen grandes capacidades de comportamientos complejos e interactivos para alcanzar un alto nivel de realismo (Cavazza, y otros, 2005). Según Whiting *et al.* (2010), existe un dilema interesante para los animadores y diseñadores que están constantemente creando nuevos entornos con el deseo de incorporar personajes que sean autónomos, dado que cada entorno requiere un proceso complejo y de tiempo intensivo que debe realizar un programador experto, porque los modelos son creados para resolver problemas específicos y porque por su naturaleza específica, no se pueden reutilizar fácilmente.

Alrededor del mundo, en universidades, centros de investigación y empresas se encuentran ejemplos de EVI desarrollados sin una metodología clara que vele por la administración de recursos, la optimización de geometría y la compatibilidad de lenguaje, de máquina y de plataforma, o que han sido adaptados de una manera muy básica y que no cuentan con una metodología que permita crear un EVI con cada uno de sus elementos, basados en un estándar que garantice un proceso fluido desde que se idea hasta que se ejecuta. Los EVI traen consigo problemas en tiempo real al solucionar sus algoritmos; ellos dependen de la integración estrecha entre la interactividad natural de un entorno virtual en términos de la visualización centrada en el usuario y la manipulación de objetos, y los aspectos interactivos de la resolución de problemas de algoritmos de IA.

En el ámbito laboral, se necesita el conocimiento, la cooperación y la coordinación de las personas, para que por medio de un trabajo en equipo, desarrollen lo necesario para recrear los entornos en los cuales se van a realizar entrenamientos o inducciones acerca de un trabajo o actividad. Todo eso, al igual que en el campo de la enseñanza, requiere recursos intelectuales, físicos y por ende económicos, los cuales tienden a ser considerados muy altos, precisamente por la falta de conciencia sobre la utilidad y los grandes beneficios de los EVI, debido al desconocimiento del tema y a la ausencia de una guía que facilite su implementación en el medio.

Por lo expuesto anteriormente, se hace necesario crear un modelo de referencia de un Entorno Virtual Inteligente, apoyado en la percepción y el razonamiento de sus elementos, con el fin de alcanzar un alto grado de realismo. Esto, con el fin de permitir la expansión, conocimiento e implementación de este modelo en diversos campos, debido a que actualmente, las técnicas de simulación por computador están adquiriendo cada vez mayor relevancia, debido a su capacidad para optimizar procesos e imitar situaciones o actividades que, por su naturaleza, pueden resultar peligrosas, costosas o entretenidas.

1.5 Hipótesis de Investigación

Es posible construir un modelo de Entorno Virtual Inteligente que genere realismo a la hora de interactuar con los elementos de un Entorno Virtual con un personaje, independiente al campo de aplicación. El componente de inteligencia se proporcionará en la percepción y el razonamiento del mismo, de acuerdo a la técnica de Inteligencia Artificial que proporcione mejores resultados entre agentes inteligentes, redes neuronales y/o computación evolutiva; se plantea que una vez implementada la técnica en el modelo del Entorno Virtual Inteligente, éste permita tener un dinamismo entre el personaje y sus elementos.

1.6 Objetivos

1.6.1 Objetivo General

Proponer un modelo de Entorno Virtual Inteligente basado en la percepción y el razonamiento de sus elementos con un personaje para la generación de realismo.

1.6.2 Objetivos Específicos

1. Proponer un modelo geométrico y cinemático que sirva de base al Entorno Virtual, a partir de la caracterización de sus elementos relevantes.
2. Incorporar técnicas de Inteligencia Artificial al Entorno Virtual para la evaluación de la percepción y el razonamiento de un personaje con los elementos del mismo.
3. Determinar la técnica de Inteligencia Artificial adecuada por medio de un análisis comparativo para su implementación en un Entorno Virtual Inteligente.
4. Validar el modelo de Entorno Virtual Inteligente mediante un prototipo de representación visual para la verificación de los resultados obtenidos.

1.7 Contribución de la Tesis

Las siguientes son las contribuciones de la tesis:

- Un modelo de los niveles geométrico y cinemático para desarrollar un Entorno Virtual 3-D, flexible al entorno deseado e independiente del área de aplicación, para crear entornos virtuales en menos tiempo, e incrementar el desempeño del entorno virtual en los niveles comportamental y cognitivo.

- Un modelo de Entorno Virtual Inteligente con un desempeño dinámico definido en la percepción y el razonamiento del mismo, como características de un comportamiento dinámico e interactivo, con el fin de generar realismo, necesario en la aplicaciones de Realidad Virtual.
- En términos de desarrollo de software, esta tesis propone y evidencia el uso de motores de videojuegos, con el fin de obtener un Entorno 3-D y equilibrado con una técnica de Inteligencia Artificial que lo hace más funcional y más realista.
- La validación de las técnicas del modelo, se experimentaron en tres campos diferentes: un EVI orientado a la medicina, otro EVI orientado a la seguridad laboral y finalmente un EVI orientado a la educación, lo que permite consolidar que el modelo propuesto, funciona independiente del área de aplicación.

1.8 Divulgación de Resultados

Durante la investigación, fueron sometidos resultados parciales ante comités académicos de eventos y revistas de divulgación nacional e internacional, entre los cuales se tienen:

- Aplicación de Redes Neuronales Artificiales en Entornos Virtuales Inteligentes. Revista Información Tecnológica. La Serena, Chile. Categoría A1 Publindex, Colciencias. Año: 2014. Volumen: 25. Núm.: 5. Páginas 103-112. ISSN: 0718-0764.
- Caracterización del modelo geométrico y cinemático de un entorno virtual. Ingeniería de Software e Ingeniería de Conocimiento. Sello Editorial Universidad de Medellín. 1ª Edición 2014. ISBN: 978-958-8815-31-2.
- Detección de Problemas Visuales mediante un Entorno Virtual Inteligente. Revista Vínculos – Universidad Distrital Francisco José de Caldas. Edición 18. Enero – Junio 2013. Volumen 10, Número Uno. ISSN 1794-211X.
- Aplicación de Redes Neuronales Artificiales en la Detección de Problemas Visuales. 11º Congreso Interamericano de Computación aplicada a la Industria de Procesos – CAIP’2013. ISBN: 978-9972-2885-6-2.
- Modelo de un Personaje en un Entorno Virtual Inteligente. Revista Información Tecnológica. La Serena, Chile. Categoría A1 Publindex, Colciencias. Año: 2012. Volumen: 23. Núm.: 3. Páginas 103-112. ISSN: 0718-0764.

- Modelo de Referencia de un Entorno Virtual Inteligente para la obtención de un adecuado Realismo Visual. Mateus y Branch. 10º Congreso Interamericano de Computación Aplicada a la Industria de Procesos, CAIP 2011. Mayo 30 a Junio 3. Girona, España.
- Integration of Artificial Intelligence Techniques in a Virtual Environment. 17th International Conference on Human-Computer Interaction - HCI International 2015. Proceedings by Springer. Artículo aceptado para ser presentado como poster en el Evento en Agosto del 2015.

1.9 Organización de la Tesis

Esta tesis está organizada de la siguiente manera: En el Capítulo 2, se desarrolla la propuesta del modelo geométrico y cinemático para un entorno 3-D. En el Capítulo 3, se detalla la metodología utilizada para la incorporación de técnicas de Inteligencia Artificial en el Entorno 3-D y obtener de esta manera el Modelo de EVI. En el Capítulo 4, se muestra la evaluación de las técnicas de Inteligencia Artificial y la selección de la Red Neuronal Artificial para la validación final del EVI. En el Capítulo 5, se muestra el prototipo final de validación realizados con motores de videojuegos para el EVI. Por último, en el Capítulo 6 se exponen las conclusiones y los trabajos futuros.

2. Propuesta de un Modelo Geométrico y Cinemático de un Entorno Virtual, a Partir de la Caracterización de sus Elementos Relevantes

Un Entorno Virtual hace uso de la Realidad Virtual para permitir la experimentación de un ambiente generado en 3-D, usando computación gráfica y con ello, lograr una interacción entre el ser humano y un mundo creado en el computador.

La estructura de modelado de la computación gráfica, tiene los siguientes niveles (Ver Figura 2.1): geométrico, cinemático, físico, comportamental y cognitivo (Funge, Tu, & Terzopoulos, 1999).



Figura 2.1: Estructura de modelado de la computación gráfica (Funge, Tu, & Terzopoulos, 1999)

En la estructura expuesta, los tres primeros niveles (geométrico, cinemático y físico) involucran la animación de bajo nivel y aplican la física del mundo real al movimiento producido en un entorno. Los otros dos niveles (comportamental y cognitivo), se dirigen a crear un modelo ejecutable del proceso de la idea de un personaje en un entorno; el nivel cognitivo, busca lograr metas de largo plazo, mientras que en el nivel comportamental, se reacciona y se buscan metas inmediatas.

En este sentido, los niveles geométrico y cinemático deberían ser mínimamente invasivos y fáciles de implementar, con el fin de no limitar los modelos cognitivos y de comportamiento, pero esto no sucede generalmente, dado que estos niveles más bajos se

consideran como cajas negras (Whiting, Dinerstein, Egbert, & Ventura, 2010). Es por esto, que los niveles geométrico y cinemático, se deben diseñar de forma adecuada para que posteriormente, se pueda incorporar exitosamente un personaje o cualquier elemento, en dicho entorno.

Con base en lo anterior, en este capítulo se propone un modelo de referencia de la geometría y cinemática de un entorno virtual, flexible al entorno deseado e independiente del área de aplicación, para crear entornos virtuales en menos tiempo, e incrementar el desempeño del entorno virtual en los niveles comportamental y cognitivo, el cual se puede resumir en la Figura 2.2.

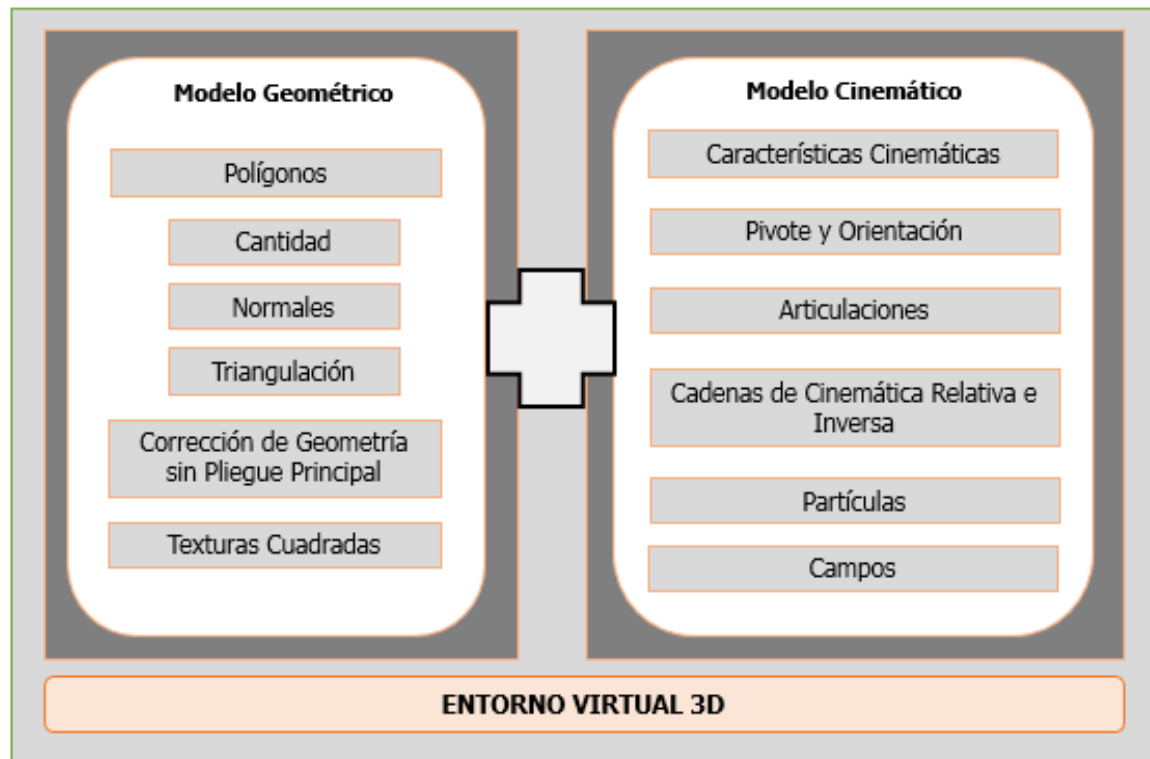


Figura 2.2. Modelo Geométrico y Cinemático

Este capítulo está organizado de la siguiente manera: en la sección 2.1 se hace la caracterización del modelo geométrico, en la sección 2.2 se hace la caracterización del modelo cinemático y por último se exponen las conclusiones. Cabe señalar que estas secciones, son construcciones propias basadas en la experiencia y en pruebas realizadas en diferentes herramientas de modelado.

2.1 Modelo Geométrico

Al usar el término geometría, se hace referencia a los modelos compuestos por polígonos en las diferentes áreas del desarrollo de contenido 3-D. Los polígonos son la

expresión básica de los modelos tridimensionales generados por computador, los cuales se componen de tres elementos tales como los vértices, los bordes y las caras, como se puede observar en la Figura 2.3.

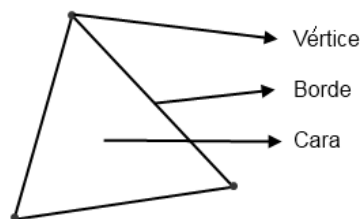


Figura 2.3: Elementos de un polígono

Un vértice es un punto cuya única propiedad son tres coordenadas en el espacio, haciendo referencia únicamente a lo visual. Un borde es la línea recta entre dos vértices el cual goza de dos propiedades (distancia y dirección, que se relacionan con los dos vectores en sus extremos). Una cara es la representación de tres o más bordes acotados y posiblemente, el más importante de los tres elementos que conforman el polígono, dado que a diferencia de los dos primeros, la cara es la que goza de forma para que el ser humano la interprete inmediatamente, debido a que todos los objetos que lo rodean se pueden representar con caras y no con puntos (vértices) o líneas (bordes).

2.1.1 Definición de la Cantidad de Polígonos

La cantidad de polígonos es de suma importancia porque tiene una relación con la calidad visual de los modelos y porque influye directamente en las interacciones cinemáticas.

Considerando que la cantidad de polígonos lleva consigo un costo en memoria, es necesario manejar una relación para los modelos que existan en un Entorno Virtual. Con base en pruebas, se pudo determinar una proporción de la cantidad de polígonos en relación con la curvatura de su superficie, que permite un nivel de calidad visual alto, razonable con la memoria y exacto frente a la cinemática de los motores gráficos.

El número de polígonos en una superficie curva se debe determinar con la siguiente fórmula para los modelos (Ecuación 2.1).

$$N = \frac{\text{Perímetro}}{18} * \frac{\min(\text{radio})}{\max(\text{radio})} \quad (\text{Ecuación 2.1})$$

donde:

N : es el número entero aproximado hacia arriba de polígonos de la superficie.

18: es la constante que representa un polígono cada 18 grados de la curva.

$\min(\text{radio})$: es la distancia mínima de la curva al origen de un plano cartesiano, si dicha curva se colocara en dicho plano.

$\max(\text{radio})$: es la distancia máxima de la curva al origen de un plano cartesiano, si dicha curva se colocara en dicho plano.

Para un ejemplo de una superficie de 180° (Figura 2.4), 10 unidades de ancho y 5 de alto, la fórmula daría como resultado lo que aparece en la (Ecuación 2.2).

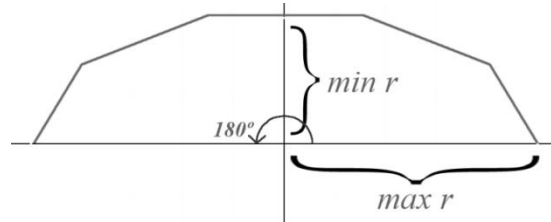


Figura 2.4: Ejemplo de número de polígonos para una superficie de 180°

$$N = \frac{180}{18} * \frac{5}{10} = 5 \quad (\text{Ecuación 2.2})$$

En la Figura 2.5, se muestra la superficie de sólo cinco polígonos, de acuerdo con el resultado de la (Ecuación 2.2).

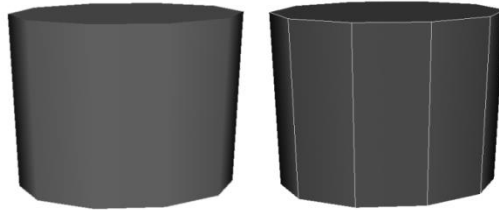


Figura 2.5: Renderizado por Hardware para una superficie de 5 polígonos

2.1.2 Normales de los Polígonos

En los motores gráficos, los polígonos solo tienen una cara, debido a que su revés no se considera para efectos visuales o cinemáticos. Como norma para determinar la cara que el motor va a considerar, se usa la normal de los polígonos. La normal se define cuando, en la creación del polígono, todos sus vértices se crean en el sentido opuesto a las manecillas del reloj. Así, la cara que se ve siguiendo esta definición, es perpendicular a la normal.

Teniendo en cuenta que sólo se usa una cara de cada polígono, todos los polígonos de cada modelo deben compartir el sentido de las normales para efectos visuales y cinemáticos y evitar perforaciones en modelos o fallas en colisiones. La Figura 2.6, tiene todas sus normales hacia afuera (representadas por las líneas alrededor de la Figura), que es adecuado para la representación de cualquier modelo.

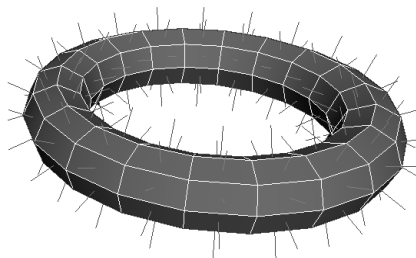


Figura 2.6: Normales adecuadas en polígonos

La Figura 2.7, tiene el mismo número de polígonos que la Figura 2.6, pero algunas caras están en sentido opuesto, causando omisión en el renderizado y colisiones imprecisas.

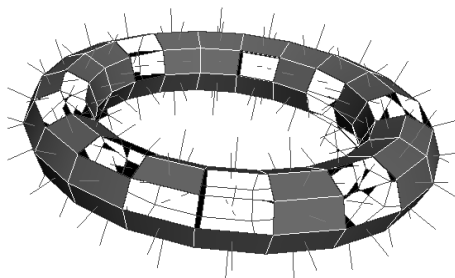


Figura 2.7: Normales inadecuadas en polígonos

2.1.3 Triangulación de Polígonos de más de Tres Lados

Como se mencionó al iniciar la Sección 2.1, se identifica al polígono como elemento básico en los Entornos Virtuales; los polígonos se definieron anteriormente como un plano conformado por tres o más vértices. Cualquier Entorno Virtual, motor gráfico o sistema 3-D sólo trabaja con polígonos de tres lados; aunque recibe polígonos de más lados, internamente los segmenta para llevarlos a su mínima expresión triangular y poder mostrar e interpretar los elementos tridimensionales.

Si la geometría existente en el Entorno Virtual tiene polígonos de más de tres lados se debe considerar el proceso de triangulación, que conlleva dos consecuencias importantes: tiempo en el proceso de triangulación y conjunto de las fallas que se pueden presentar en polígonos de más de tres lados con vértices no coplanares; por ejemplo, si se tiene un polígono de siete vértices y uno de ellos no se encuentra a la misma altura de los demás, la triangulación puede generar más segmentos de los necesarios, haciendo uso de más memoria y posibles fallas de colisión.

Con base en lo anterior, es considerablemente ideal, que los elementos existentes en un Entorno Virtual se triangulen, debido a que se le evita un proceso innecesario y riesgoso al sistema en cada carga de la geometría. Cabe destacar, que el conteo de polígonos aumentaría explícitamente al triangular los objetos 3-D, pero internamente el costo de memoria y recursos sería el mismo o menor, ya que si un elemento no se trianguló previamente, el sistema lo triangularía (Figura 2.8).



Figura 2.8: Triangulación de Polígonos: a. Polígono sin triangular. b. Polígono Triangulado

2.1.4 Corrección de Geometría sin Pliegue Principal

La geometría sin pliegue principal es una condición muy particular, que aplica a los polígonos y tiende a ser inestable en los Entornos Virtuales, porque puede generar cualquier tipo de falla; dicha condición representa una contradicción en el estado natural de un polígono. Como se indicó anteriormente, la normal de un polígono es la que da información al entorno virtual, del lado por el cual hay visualización y colisión. En la geometría sin pliegue principal, la normal no posee dirección alguna, debido a que la posición de los vértices causa que la cara sea inconsistente. Para ilustrar esta condición se puede tomar un polígono de cuatro lados (cuadrado), formado por los vértices $\{A, B, C, D\}$ como se ve en la Figura 2.9a.



Figura 2.9: Corrección de Geometría. a. Polígono con pliegue principal. b. Polígono sin pliegue principal.

Posteriormente, se cambia el orden de los dos últimos vértices, dando como resultado una figura similar a un reloj de arena (Figura 2.9b), causando que no se pueda definir una normal para el polígono (como se explicó en la sección 2.1.2). Un polígono con esta condición representa una inconsistencia desde sí mismo y puede desencadenar fallas de cálculo, visualización y colisión, entre otros.

Con base en lo anterior, en la construcción de un Entorno Virtual, se debe realizar un proceso de corrección de geometría sin pliegue principal.

2.1.5 Texturas Cuadradas

Así como es de importante el nivel gráfico de los Entornos Virtuales, también lo es la estética, como característica que el ser humano impregna a todas sus obras. Por ello, en la actualidad, los sistemas similares a motores gráficos cuentan con capacidades suficientes para adornar los Entornos Virtuales tanto como se requiera o se desee. Así, el texturizado de la geometría es, tal vez, la última característica a considerar, pero una muy importante si el Entorno Virtual tiene como objetivo interactuar con usuarios.

Con base en lo anterior, las texturas deben ser cuadradas, esto significa que la imagen usada como textura debe tener dimensiones proporcionales de 1:1.

2.2 Modelo Cinemático

La capacidad de un Entorno Virtual en la que los objetos pueden simular el movimiento físico de los objetos del mundo real se denomina Cinemática, la cual proporciona propiedades a los objetos para que sus movimientos sean controlados, previstos o sean el resultado de simulaciones que en el mundo real serían difíciles, costosas o riesgosas de tener. La Cinemática es el equivalente a tener objetos con propiedades y comportamientos del mundo físico en un mundo donde se cumplen las leyes físicas universales.

2.2.1 Características Cinemáticas de los Objetos

Una de las características más importantes de los Entornos Virtuales es la calidad de la similitud que puede tener con la realidad; como en el mundo natural, el movimiento que describe un objeto frente a la gravedad o inercia, es lo que lo hace ver coherente.

Así mismo, el Entorno Virtual es tan convincente, como sea el movimiento de sus objetos. Para lograr dicha fidelidad hacia las leyes de la física del mundo real, es importante que cada objeto, luego de pasar por un proceso de modelado correcto, posea ciertas calidades físicas, como la masa.

La interrelación de los elementos en un Entorno Virtual hace que la masa sea un componente fundamental, porque permite la influencia de campos uniformes, gravedad, inercia, turbulencia y otras fuerzas que podría ofrecer un Entorno Virtual; cada uno de estos elementos y la capacidad de colisionar entre sí, hace que los objetos tengan un comportamiento natural y que haya control sobre el Entorno Virtual. Por el contrario, si la información de masa no se considera como debe ser, entonces se puede obtener un escenario caótico o limitado en interrelación, causando resultados inesperados y afectando indirectamente a otros elementos.

En el caso particular de las colisiones, es altamente factible que el evento de objetos colisionando, sea fundamental en la mayoría de los Entornos Virtuales, independientemente

de sus objetivos. Este evento puede ser el de una entidad que se desplaza por una superficie, un objeto que sostiene otro, un elemento golpeando a otro, etc. En todos estos casos, los objetos deben recurrir a las características cinemáticas.

Un Entorno Virtual debe considerar las siguientes características cinemáticas:

- Masa: es la característica más importante dentro de este grupo y hace referencia al peso de cada objeto. Debido a la masa, los objetos reaccionan frente a colisiones y a campos de fuerza.
- Centro de Gravedad: es el punto central del objeto.
- Fricción: característica no muy común, pero a medida que los Entornos Virtuales se vuelven más precisos, gana importancia e indica la resistencia al desplazamiento sobre otro objeto. Las propiedades de la fricción se aplican a las caras del objeto.
- Rebote: un número considerado de entornos da importancia a esta característica, que obliga a los objetos a cambiar de rumbo tras una colisión.

2.2.2 Pivote y Orientación

Todo objeto 3-D tiene un punto central, llamado pivote, el cual representa el punto absoluto en el espacio del entorno virtual y sirve para desplazar, rotar y escalar cada objeto en el espacio tridimensional.

Cuando un sistema construye y reconoce un objeto 3-D, la interrelación entre el entorno y el objeto nace de este punto, pues le da información espacial del objeto al entorno, para controlar su posición, rotación y escala. Esa es la razón principal de que un objeto tenga definido su pivote y su orientación. Además, la orientación del objeto debe ser la misma que la del Entorno Virtual para que se consoliden posiciones, distancias y nortes. Por ejemplo, si se tiene un Entorno Virtual en el cual el eje Y indica la altura y se modela un edificio cuya altura se oriente en el eje Z, cuando el Entorno Virtual cargue dicho edificio, se tendrá el edificio acostado. Otro ejemplo, sería el de un auto cuyo desplazamiento se controla con el eje Z de su pivote; si el Entorno Virtual considera el eje Z como el eje vertical, el auto se desplazará hacia arriba como un cohete.

2.2.3 Articulaciones

Las articulaciones tienen un principio elemental pero altamente funcional que se puede relacionar, por ejemplo, con los huesos de un animal. Aunque su principal uso se da para controlar objetos orgánicos, no se queda ahí sino que se expande a diferentes tipos de usos en la cinemática.

Las articulaciones son objetos que no se consideran para visualización, sino únicamente para controlar otros objetos cuya visualización es obligatoria. Las articulaciones son jerárquicas y se asimilan con los árboles, en donde la posición y escala de un nodo

depende proporcionalmente de su padre, más no de su rotación. De esta característica surge la facilidad de controlar el movimiento de otros objetos de manera más precisa y útil en los Entornos Virtuales.

Un objeto se puede controlar siguiendo su eje, pero esto se vuelve tedioso cuando los elementos ganan complejidad. Para desglosar la idea, se tomará como ejemplo el brazo de una retroexcavadora, el cual tiene tres segmentos, el brazo que tiene un primer y segundo segmento y la pala que sería el tercer segmento. Para manipular el brazo correctamente se tendría que recurrir a múltiples operaciones; se parte de que el brazo está totalmente vertical (Figura 2.10) y se desea que el brazo toque el suelo para cavar. En la Figura 2.10a se hará el movimiento usando los pivotes de cada elemento (intersección de cada segmento) y en la Figura 2.10b se hará el movimiento usando las articulaciones (línea que atraviesa el brazo).



Figura 2.10: Articulaciones en el brazo de una retroexcavadora. a. Uso de pivotes. b. Uso de articulaciones.

Una vez definido lo anterior, el primer segmento del brazo se puede rotar 45° en el sentido del reloj (Figura 2.11a y Figura 2.11b), pero como no hay ninguna relación entre el primer y segundo segmento, este último se debe desplazar con funciones seno y coseno, para que conserve su posición en relación con el brazo (Figura 2.11c y Figura 2.11d).

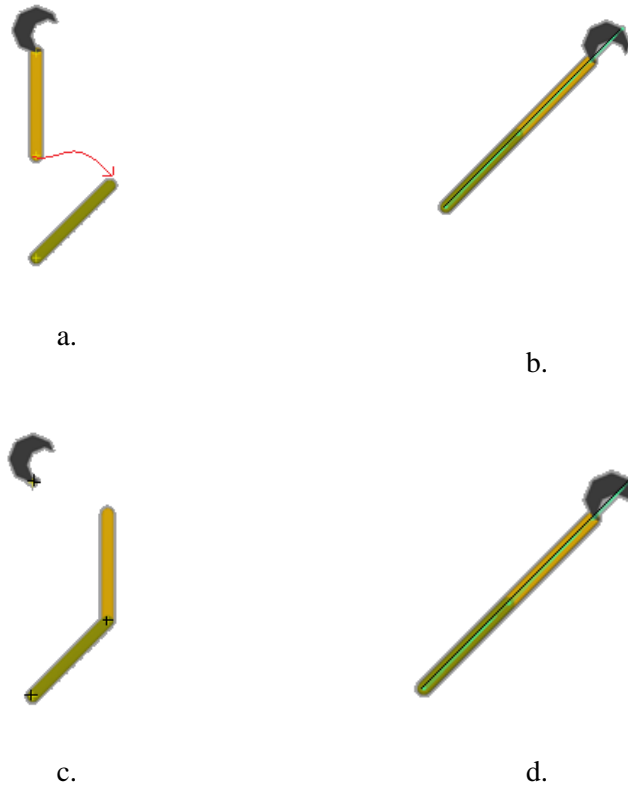


Figura 2.11: Rotación del brazo. a. El brazo rota 45 grados sobre su pivote. b. La articulación principal rota 45 grados. c. Desplazamiento de segmentos. d. No hay operación adicional con el uso de articulaciones

Luego, se debe rotar otros 45° para formar un triángulo en el que la punta del brazo toque el suelo; a su vez, este proceso se debe repetir con todas las partes dependientes como la pala, la cual se debe desplazar de la misma manera y rotar (Figura 2.12a y Figura 2.12c). Finalmente, para el caso de pivotes, la pala se debe trasladar y rotar sobre su eje para igualar el movimiento del brazo con las articulaciones de la Figura 2.12c (Figura 2.12b).

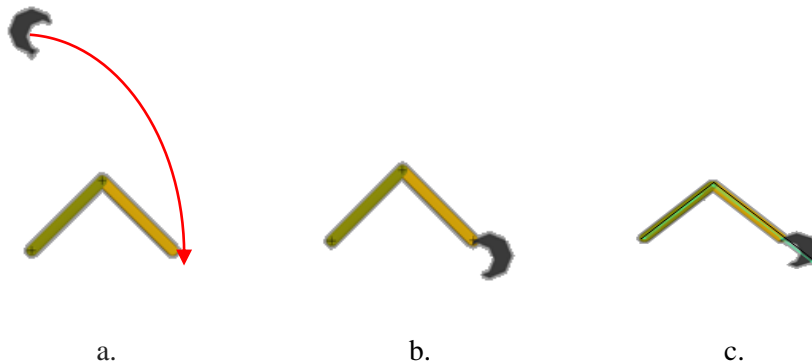


Figura 2.12: Movimientos para alcanzar el suelo. a. Rotación del brazo sobre su eje con pivotes. b. Rotación y traslación del brazo con pivotes. c. Rotación con articulaciones.

Las articulaciones simplifican esta cantidad de operaciones, pues ya poseen principios de unión y relación entre sí.

Para situaciones como las anteriores, en la que se usan movimientos tan elementales y que son muy frecuentes en entornos activos, las articulaciones son útiles y ahorran recursos de cálculo y permiten más control sobre los modelos.

2.2.4 Cadenas de Cinemática Relativa e Inversa

Con el uso de las articulaciones, aparecen funciones que ofrecen aún más simplicidad y precisión en los movimientos. Las cadenas de cinemática relativa e inversa constan de elementos mecánicos virtuales que controlan las articulaciones de manera más simple, utilizando la traslación en el espacio para afectar la rotación de cada articulación.

Una cadena de cinemática relativa es una línea recta, la cual va del origen de una articulación, a su próxima; el origen de la línea es el mismo de la articulación y la ubicación espacial de ambos orígenes se controla desde la articulación, es decir, la ubicación del origen de la cadena cinemática sigue el origen de la articulación que controla, pero el otro extremo de la cadena se controla independientemente. Dicha línea no tienen límite escalar, pero sí el otro extremo de la articulación, el cual sólo cumple con ser la intersección de la cadena sin cambiar su tamaño.

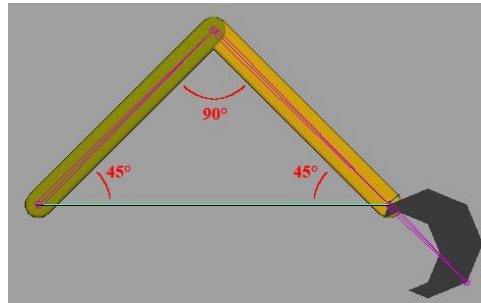
En la Figura 2.13, se ve el estado inicial de las articulaciones de un brazo, en el cual la primera articulación se controla con una cadena de cinemática relativa (Figura 2.13a), que posteriormente se mueve al noroeste, alterando la rotación de la articulación, para mantenerse bajo el control de la cadena (Figura 2.13b). Así, con el movimiento espacial de la cadena que es más simple, la rotación de la articulación que es más compleja se controla.



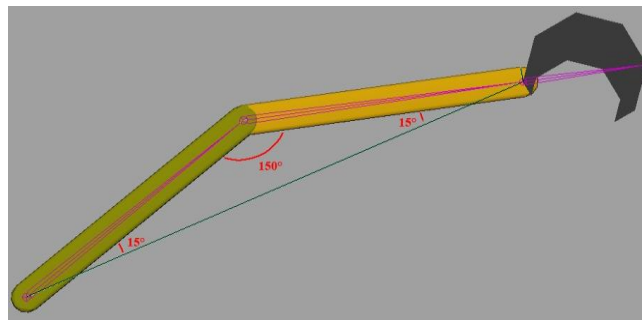
Figura 2.13: Cadena de Cinemática Relativa. a. Estado inicial. b. Movimiento y rotación de la articulación.

De una manera más eficiente funcionan las cadenas de cinemática inversa, debido a que expanden su control a más de una articulación; también es una línea recta, pero va a los extremos opuestos de las articulaciones que están bajo su control y su nombre deriva de la proporción inversa que mantienen sus ángulos entre sí, cuando hay movimiento en la cadena. Otra característica que describe su comportamiento, es la que surge del triángulo en la cual, la suma de sus ángulos siempre va a ser igual a 180° . Las cadenas de cinemática inversa

junto con dos articulaciones, forman un triángulo en su expresión más simple; tomando como ejemplo el brazo de la retroexcavadora, la cadena de cinemática inversa comparte su origen con el de la articulación principal y llega hasta el extremo posterior de la articulación del brazo (Figura 2.14a).



a.



b.

Figura 2.14: Cadena de Cinemática Inversa

La línea horizontal que atraviesa la Figura 2.14 es la cadena de cinemática inversa. El segundo extremo de la cadena se desplaza y las dos articulaciones que controla la cadena se mueven para conservar la influencia de la cadena; en esa misma situación, se puede observar el cambio en los ángulos (Figura 2.14b).

2.2.5 Partículas

Las partículas son componentes bastante útiles en los Entornos Virtuales y son pequeños elementos con características particulares que permiten implementar objetos que no se pueden obtener con figuras poligonales, como el agua o el fuego.

Un emisor (punto o superficie) genera las partículas. Posteriormente, las siguientes características de las partículas las hacen abundantes en dinamismo:

- Dirección: inmediatamente la genera el emisor, la partícula se crea apuntando a una dirección específica, lo cual no implica que se mueva después de su generación.
- Impulso: si bajo esta característica hay un valor, la partícula se moverá a la velocidad de esta característica, en la dirección de la partícula.

- **Conservación:** es la capacidad de no perder el impulso que tiene. Al carecer de esta característica, la partícula puede llegar a ser más natural, de lo contrario, su desplazamiento sería infinito y no reflejaría ningún comportamiento del mundo real, exceptuando el movimiento en el espacio exterior.
- **Figura:** la forma básica de una partícula es el punto, pero las partículas pueden tomar diferentes figuras como las de círculos, líneas, esferas, etc. A partir de lo anterior, pueden tomar color y transparencia.
- **Colisión:** es la capacidad de interactuar cinemáticamente ante otros objetos poligonales.
- **Vida útil:** como su nombre lo indica, es la vida útil que tendrá cada partícula en el tiempo, es decir, cuánto tiempo existirá cada partícula luego de generarla. Este atributo es muy importante, debido a que, si no se considera, los Entornos Virtuales se llenarán de estos objetos copando sus recursos y terminando colapsado.

2.2.6 Campos

Los campos son secciones espaciales que afectan objetos con propiedades dinámicas. Una partícula u objeto poligonal, al entrar en la sección del campo, se afecta con éste; dichos campos describen que tipo de influencia tienen sobre los objetos que estén en ellos y a que magnitud lo hacen. Por ejemplo, se puede tener un campo que simule la gravedad; en este caso, si una partícula que viaja horizontalmente llega a pasar por la sección de influencia del campo, empezará a perder su movimiento paulatinamente a razón de la magnitud del campo y caerá, tal como lo muestra la Figura 2.15.

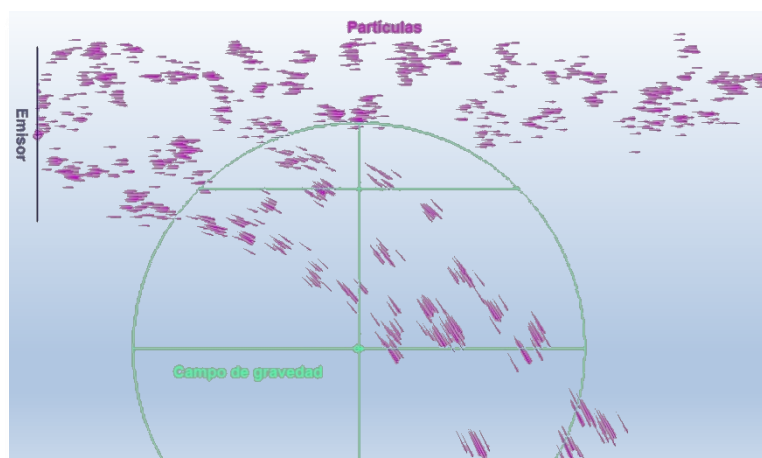


Figura 2.15: Ejemplo de un campo de gravedad

2.3 Conclusiones

En la revisión de trabajos previos alrededor de la temática de Entornos Virtuales, se puede observar que dichos entornos se limitan a una tarea específica y que, además, no cuentan con una caracterización o con un modelo general que permita crear un Entorno Virtual con cada uno de sus elementos, basados en un estándar que garantice un proceso fluido desde que se concibe, hasta que se ejecuta. Los Entornos Virtuales traen consigo problemas en tiempo real al solucionar sus algoritmos, debido a que dependen de la integración estrecha entre su interactividad natural, en términos de la visualización centrada en el usuario y la manipulación de objetos.

Se propuso un modelo geométrico y cinemático de referencia para un Entorno Virtual, los cuales incluyen un esquema con un mecanismo factible de usar, con el fin de que el entorno pueda ejecutar un razonamiento apropiado del mismo, al componer a partir de éste, los modelos cognitivos y de comportamiento.

3. Incorporación de Técnicas de Inteligencia Artificial al Entorno Virtual

En un Entorno Virtual 3-D, un computador genera impresiones sensoriales que se envían a un usuario a través de sus sentidos; el tipo y la calidad de estas impresiones, determinan el nivel de inmersión y la sensación de presencia en el Entorno Virtual. Lo ideal sería enviar al usuario información por intermedio de todos sus sentidos, usando una alta resolución, alta calidad y consistencia en todo momento. Sin embargo, la realidad es muy diferente, dado que muchas aplicaciones estimulan sólo algunos de los sentidos, utilizando una baja calidad en las representaciones al igual que poseen una falta de sincronización de la información (Lozano M. , 2004).

Con base en lo anterior, es que surgen los Entornos Virtuales Inteligentes (EVI), los cuales están compuestos por la unión de dos importantes áreas de la computación: la Realidad Virtual y la Inteligencia Artificial (IA), las cuales tienen múltiples aplicaciones que les permiten ser ampliamente utilizadas en campos como la medicina, la industria, la educación y la investigación, entre otras; además, su acción conjunta les permite alcanzar un completo nivel de aplicabilidad en diversas áreas (Mateus & Branch, 2012). La conducta inteligente según (Whiting, Dinerstein, Egbert, & Ventura, 2010), sugiere que la creación de modelos cognitivos y de comportamiento para un personaje en un entorno, puede presentar muchos cambios porque: (1) requiere de un proceso complejo y de tiempo intensivo que debe realizar un programador experto; (2) porque los modelos se crean para resolver problemas específicos en un entorno dado; y (3) porque su naturaleza específica no se puede reutilizar fácilmente; lo anterior, plantea un importante dilema para los animadores y diseñadores que están constantemente creando nuevos entornos, con el deseo de incorporar personajes que sean autónomos.

El capítulo anterior, se enfocó en proponer un modelo de los niveles geométrico y cinemático de un entorno virtual. Éste capítulo se centra, en los niveles comportamental y cognitivo, a través de la incorporación de diferentes técnicas de Inteligencia Artificial en el Entorno Virtual, tales como las Redes Neuronales, la Computación Evolutiva y los Agentes Inteligentes, con el fin de proponer un modelo de Entorno Virtual Inteligente, que pueda resolver diferentes problemas y que se pueda reutilizar.

Este capítulo está organizado de la siguiente manera: en la sección 3.1, se exponen los conceptos de percepción y el razonamiento, y se explica el modelo propuesto; en la sección 3.2, se explica la Red Neuronal Artificial implementada en el entorno virtual; en la

sección 3.3, se exponen las técnicas de computación evolutiva y en la sección 3.4, se muestra la metodología de agente inteligente implementada en el entorno virtual; por último se exponen las conclusiones del capítulo.

3.1 Percepción y Razonamiento

Los EVI deben alcanzar grandes capacidades de comportamientos complejos e interactivos para alcanzar un alto nivel de realismo (Cavazza, y otros, 2005). Este realismo, está basado en los elementos que permiten un desempeño inteligente tales como: la percepción, el aprendizaje, la comunicación mediante un lenguaje natural y el razonamiento. De acuerdo a lo anterior y al modelo planteado, este trabajo se centra solo en la percepción y el razonamiento, en los niveles cognitivo y comportamental.

Según (Martinho, Paiva, & Gomes, 2000), la percepción es considerada como todos los eventos del entorno virtual que son filtrados, acorde a los intereses y a la ubicación del personaje y está basada en dos principios: (1) Una percepción limitada, en la que un personaje no percibe todos los eventos, sino sólo percibe los de su área asociada; y (2) Una percepción imprecisa, en la que el personaje no percibe el entorno virtual tal como es, sino sólo percibe los eventos relevantes y asociados a éste. También describen al razonamiento, como un proceso desarrollado por un conjunto de reglas de producción las cuales son condiciones basadas en el modelo del mundo, en el estado del objetivo, en la característica del comportamiento y en la información del estado interior. Dados los conceptos anteriores, el personaje implementado en el Entorno Virtual Inteligente de ésta tesis, razonará basado en la repercusión de su objetivo interno y a la prioridad de la acción a realizar, en los niveles cognitivo y comportamental.

Con base en lo anterior, la Figura 3.1 muestra la propuesta del modelo de EVI. A un Entorno Virtual 3-D realizado a partir del modelo geométrico y cinemático propuesto y apoyado en un motor de videojuegos, se le incorporan técnicas de Inteligencia Artificial, con el fin de que se produzca a partir de una percepción dada con un personaje, un razonamiento adecuado respecto al Entorno Virtual. Las técnicas de IA implementadas en el Entorno Virtual fueron: Redes Neuronales Artificiales, Computación Evolutiva y Agentes Inteligentes. En dicha Figura, las intenciones de la percepción, se refiere a la guía que se le da al personaje para escoger la acción adecuada y la memoria es usada para recordar los objetos percibidos y almacenar las intenciones que pueden ser interrumpidas por otros eventos del entorno.

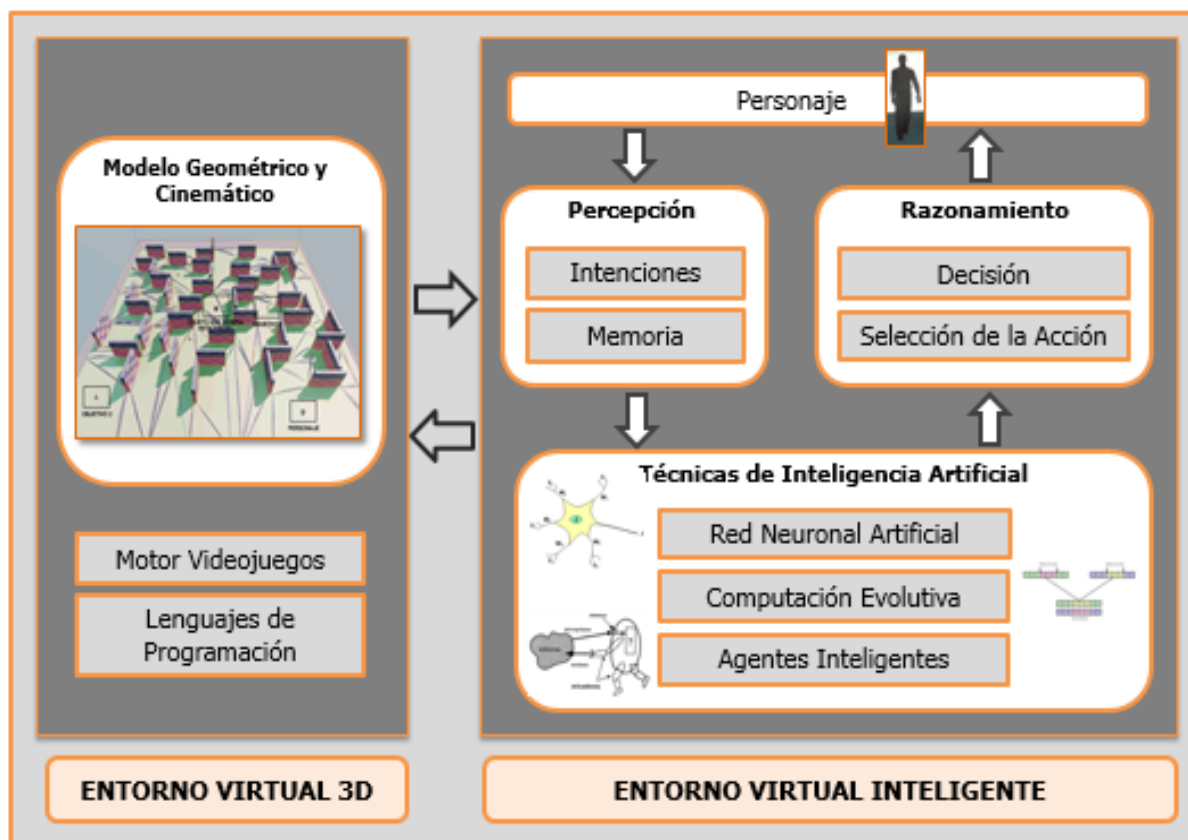


Figura 3.1: Modelo de Entorno Virtual Inteligente

En este modelo, las técnicas de Inteligencia Artificial implementadas serán las responsables de coordinar y manejar la percepción y el razonamiento. Esto puede verse como un conjunto clasificador, que toma un conjunto de percepciones $P_1 \dots P_n$ y las combina, para tomar un razonamiento adecuado $R_1 \dots R_n$. Este razonamiento para ejecutar determinada acción, está apoyado en una de las técnicas de Inteligencia Artificial nombradas en el modelo. Así el sistema, decidirá y seleccionará una acción de acuerdo al razonamiento que realice (Ecuación 3.1):

$$S(\{P_1, R_1\}, \{P_2, R_2\}, \dots, \{P_n, R_n\}) \rightarrow A \quad (\text{Ecuación 3.1})$$

donde S es el conjunto clasificador (el modelo en sí), P_i se refiere a las percepciones recibidas y R_i al razonamiento desarrollado para tomar finalmente una acción A .

Con dicho modelo, se quiere alcanzar las siguientes características de un EVI descritas por Mang-xian y Hai-ming (2008):

- Decisivo: cualquier acción que tome el personaje, se verá reflejada en un plan eficaz.

- Tiempo Real: El personaje debe responder en tiempo real a las percepciones del entorno y de la misma forma, razonar de forma adecuada a la percepción recibida.
- Ordenado: Que siga la secuencia adecuada en cuanto a su comportamiento.

Con base en lo anterior, a continuación se describe cada una de las técnicas de Inteligencia Artificial aplicadas a un Entorno Virtual.

3.2 Entorno Virtual Inteligente con Redes Neuronales Artificiales

En esta sección, se presenta la creación de dos Entornos Virtuales Inteligentes con Redes Neuronales Artificiales (RNA): uno de prueba, en el que se simulan varias RNA y se realiza el diagnóstico de problemas visuales. El otro EVI, se crea para la percepción y el razonamiento de señales de advertencia en un Entorno laboral. En este trabajo, se abordan las limitantes del tiempo en el aprendizaje de la RNA y el costo computacional, las cuales son un factor común en la mayoría de los trabajos consultados. Para ello, se simulan diferentes tipos de RNA y entre éstas, se eligió aquella que requirió menor tiempo de entrenamiento, logrando así la reducción en costos por tiempo y recursos, con el fin de adaptarla posteriormente, con el motor de videojuegos.

3.2.1 Incorporación de una RNA en un EVI para la detección de problemas visuales

Para realizar el diseño del ojo humano, primero se realizó un estudio sobre la anatomía del ojo normal y de los ojos que poseen problemas visuales como Miopía, Hipermetropía y Astigmatismo, y se eligieron algunas características claves que hacen que se ilustre el problema que puede poseer un paciente. Para el modelo de los ojos con Miopía e Hipermetropía el tamaño del globo ocular cambia, mientras que para el ojo con astigmatismo, la que cambia es la córnea y deberá ser más ovalada que la córnea del ojo normal.

Identificación de Variables

Para el reconocimiento de las variables con las que se entrenan las RNA, el optómetra le hace una anamnesis al paciente, la cual consiste en realizarle una serie de preguntas de su vida personal, de su familia, de posibles síntomas, de los motivos de consulta, entre otros, con los que se llega a un diagnóstico del problema visual que posee. También se tienen en cuenta las fórmulas que arrojan los dispositivos utilizados por el especialista. Esta información se lleva a datos numéricos para que sean procesadas correctamente por las RNA.

Una vez identificadas las variables, se construyen los posibles patrones para cada tipo de problema de refracción (astigmatismo, miopía e hipermetropía) y también para un ojo normal, información que será utilizada para el entrenamiento de las RNA.

Caracterización de diferentes tipos de RNA

En esta etapa, se hace una exploración de diferentes tipos de RNA y se identifica que el tipo de aprendizaje es un punto clave en el desarrollo de la RNA, puesto que implica que una unidad de procesamiento es capaz de cambiar su comportamiento entrada/salida como resultado de los cambios en el medio. Por lo anterior, se clasifican algunas RNA por su aprendizaje (Tabla 3-1).

RNA	Tipo de Aprendizaje
Perceptrón	Supervisado
MLP (Multi Level Perceptron)	Supervisado
SOM (Self-Organizing Maps)	No supervisado
LVQ (Learning Vector Quantization)	No supervisado
Hopfield	No supervisado
RBF (Radial Basis Function)	Híbrido

Tabla 3-1: Clasificación de RNA por tipo de aprendizaje

Luego de conocer los tipos de RNA según su aprendizaje, se simularon cada una en MATLAB, para conocer su funcionamiento con el fin de elegir la que más se adaptara a la solución del problema visual a detectar para esta prueba y revisando que se pudiera implementar sin inconveniente en el motor de videojuego; una vez experimentado con las RNA se llegó a la conclusión de elegir, las RNA MLP y RBF por ser redes que permiten clasificación de variables según la información ingresada, por ser de aprendizaje supervisado e híbrido respectivamente y porque permiten resolver problemas de clasificación, lo que se ajustaría finalmente al modelo a proponer.

RNA Perceptrón Multicapa (MLP)

El perceptrón multicapa es la primera topología de RNA empleada y el algoritmo de entrenamiento para una capa oculta, se presenta en la Figura 3.2, el cual es una adaptación del algoritmo de (García, y otros, 2002).

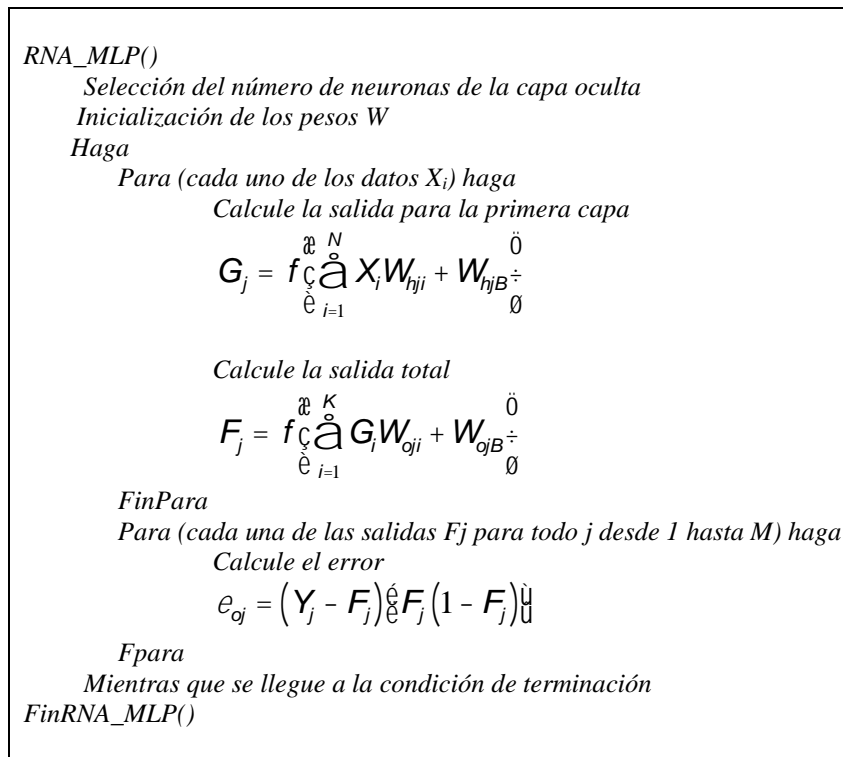


Figura 3.2: Algoritmo RNA MLP

Posteriormente, se simuló en MATLAB la RNA ingresando los datos de entrada, que son los resultados de las 12 variables definidas que componen la anamnesis de cada uno de los 120 pacientes simulados, con su respectivo objetivo, que son cada uno de los problemas de refracción. Lo anterior, con el fin de que la RNA pueda identificar cada conjunto de datos a qué problema visual corresponde; este proceso se realiza en el entrenamiento de la RNA. Básicamente la arquitectura de la RNA MLP para el caso planteado, es una Red 12-10-4 (12 neuronas de entrada, 10 ocultas y 4 de salida).

Se empleó un 80% de las entradas disponibles para entrenar y se reservó un 10% para validación y el otro 10% para prueba, con el fin de verificar que los resultados que otorga la RNA si eran satisfactorios; esto es lo que se define como el proceso de overfitting (Training, Validation, Testing), generando en la matriz de confusión, donde se interpreta que la diagonal principal son las clasificaciones correctas, por lo tanto se tuvo una clasificación del 99,2%. En la Figura 3.3, se muestra el punto en el que el error cuadrático de estas variables se encuentra al principio la red y se adapta progresivamente al conjunto de aprendizaje, acomodándose al problema.

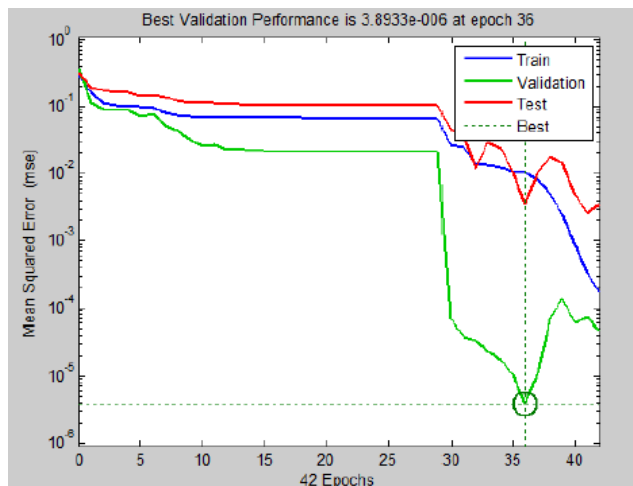


Figura 3.3: Evolución del error de aprendizaje de la RNA MLP

Este entrenamiento dio resultados satisfactorios, debido a que agrupó correctamente la cantidad de datos ingresados, según el tipo de problema visual.

RNA Función de Base Radial (RBF)

La RBF es una red cuyas funciones de activación son gaussianas. El algoritmo de entrenamiento se representa en la Figura 3.4 y es también una adaptación del algoritmo de (García, y otros, 2002).

```

RNA_RBF()
  Selección del número de neuronas de la capa oculta
  Inicialización de los  $W_i$  (Centro de las Funciones de Base Radial)
  Haga
    Para (cada uno de los datos  $X_i$ ) haga
      Asigne  $X_i$  a  $q_i$ (cluster) tal que  $\|X_i - W_i\|^2$  sea la mínima entre las
      paralelas
    FinPara
    Para (cada una de los cluster  $q_i$ ) haga
      
$$W_i = \frac{1}{\|q_i\|} \hat{a}_{j \in q_i} x_j$$

    Fpara
    Mientras que ninguno de los  $X_i$  cambie de cluster
      Encuentre la varianza de los datos mediante
      
$$S_i^2 = \frac{1}{|q_i| - 1} \hat{a}_{x \in q_i} (x - W_i)^T (x - W_i)$$

      Encuentre los pesos de la capa de salida mediante
      
$$W^T = j^T T$$

    FinRNA_RBF()

```

Figura 3.4: Algoritmo de la RNA Función de Base Radial (RBF)

Este tipo de RNA es de aprendizaje híbrido, por lo tanto se ingresan datos de la capa de entrada y capa oculta. Se utilizaron el mismo número de neuronas de entrada y en la capa oculta, se utilizaron 40 neuronas con función de activación Gaussiana.

Una vez finalizada la simulación de cada problema visual se obtuvo resultados positivos en la detección del problema visual del ojo normal y ojo con astigmatismo, pero en la detección de problema visual en los ojos con hipermetropía y astigmatismo no fue acertado el diagnóstico, concluyendo que este tipo de RNA RBF no es el más adecuado para este caso.

Selección de la RNA para el Entorno

Al implementar las dos topologías diferentes de RNA, como fueron la MLP y la RBF, los resultados obtenidos en cada una de los tipos de RNA mencionadas anteriormente son comparados en cuanto a características (Tabla 3-2) como porcentaje de clasificación correcta, tiempo requerido para el entrenamiento y tamaño de la red.

Descripción	MLP	RBF
Número de neuronas óptimo para la capa Oculta	10	40
Tiempo de entrenamiento de la Red	40 seg	120 seg
Número de iteraciones realizadas en el entrenamiento	42	420
Porcentaje promedio de Clasificación	75%	73%
Error promedio	1,0416%	1,0408%

Tabla 3-2: Comparación de los resultados obtenidos entre MLP y RBF

Una vez obtenidos y analizados los resultados de las simulaciones de las RNA MLP y RBF, se observa la RNA MLP se acerca a los resultados esperados con éxito en cada una de los problemas visuales simulados, en cambio en la RNA RBF no fue óptimo al simular Miopía e Hipermetropía por lo que se descarta esta RNA y por su complejidad en el procesamiento y construcción de la RNA, mirando a implementarla posteriormente en el motor de videojuegos.

La RNA MLP además de destacarse por su proximidad en los diagnósticos, se resalta por su clasificación de conjuntos de datos de forma rápida, lo que hace que se cumpla las limitaciones de límite de tiempo y costo computacional, características importantes para que el EVI, pueda cumplir con la característica de tiempo real.

Implementación de la RNA MLP en el Entorno Virtual Inteligente

Se realizó para esta prueba, un modelo geométrico en 3D de los ojos humanos (un ojo normal y tres que representan los problemas de refracción), tomando como base un polígono para la construcción de la esfera en el modelado. El ojo con miopía se modifica, para que la forma sea más alargada en el eje X, en la construcción del ojo con Hipermetropía se modificó para que se alargara en el eje Y para que cumpla con una de sus características,

en el ojo con Astigmatismo se planteó una córnea alargada. Posteriormente, se procedió al texturizado y luego al renderizado, como se muestra en la Figura 3.5.

En dicha Figura, en el ítem a, el ojo es normal, porque presenta su forma totalmente redonda; el ítem b representa un ojo con miopía, porque tiene un diámetro mayor que el ojo normal de manera que se alargue de forma horizontal generando que la refracción no llegue adecuadamente a la retina; en el ítem c, se observa un ojo con astigmatismo por tener la córnea más ovalada generando una visión borrosa; y por último, en el ítem d, se visualiza un ojo con hipermetropía presentando un alargamiento de forma vertical lo que no permite que la refracción llegue adecuadamente a la retina.

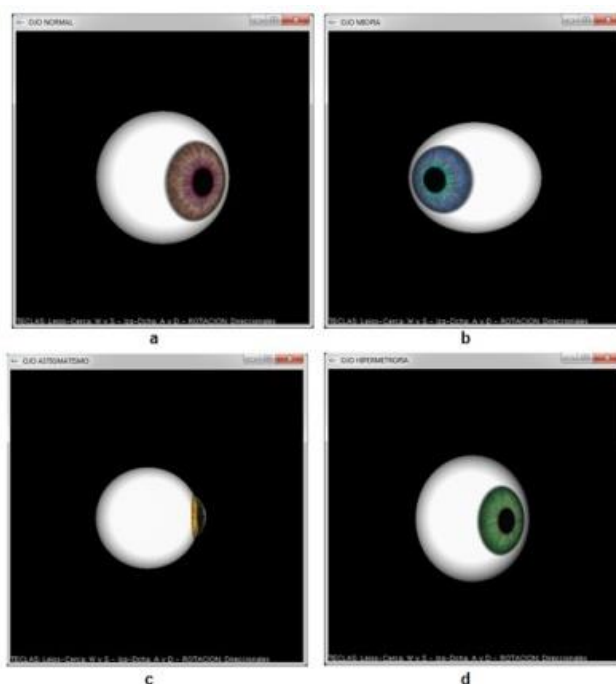


Figura 3.5: Ojo en 3D a. Ojo Normal. b. Ojo Miope. c. Ojo Hipermetrope. d. Ojo con Astigmatismo

Una vez definido el modelo geométrico de los ojos diseñados se construye el modelo comportamental, el cual es indispensable en un EVI permitiendo un comportamiento dinámico de forma que el usuario pueda interactuar con los diseños de ojos modelados en 3D. Posteriormente, para elaborar una adecuada visualización del sistema, se utiliza el ingreso de preguntas por medio de un formulario que le realiza la anamnesis al paciente (Figura 3.6); según las respuestas obtenidas, la Red Neuronal entrenada genera un diagnóstico (Figura 3.6b) y el resultado es presentado mostrando el problema visual con los ojos en 3D desarrollados en OpenGL. Finalmente al usuario se le presenta el problema visual en un Entorno Virtual que funciona de manera inteligente, de acuerdo al proceso con la RNA MLP que tuvieron sus respuestas.

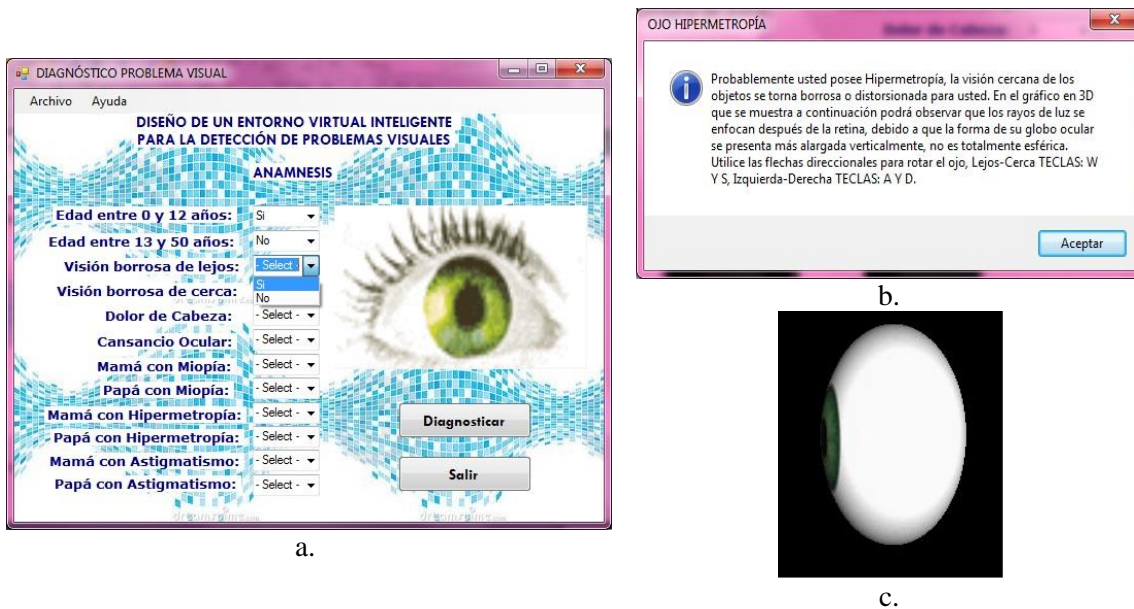


Figura 3.6: a. Formulario para la anamnesis. b. Diagnóstico. c. EVI de un problema visual

3.2.2 Incorporación de una RNA en un EVI en un Ambiente Laboral

De acuerdo a lo analizado y probado en el Entorno de la sección 3.2.1, para este caso, solo se implementa la RNA MLP en un personaje que se mueve en Entorno Virtual creado con la herramienta UDK, para identificar señales de advertencia en un ambiente laboral. Se utilizó UDK, dado la importancia que tiene este motor en el desarrollo de videojuegos famosos. El Entorno Virtual fue desarrollado por el Grupo de Computación Gráfica de la Universidade Federal do Rio Grande do Sul de Porto Alegre - Brasil. Posteriormente, fue adaptado para el caso de estudio de este trabajo, siguiendo el modelo geométrico y cinématico descrito en el capítulo anterior, e implementando en éste algunas mejoras para poder incorporar de manera adecuada, inicialmente la técnica de Path Finding; dicha técnica ayuda a detectar en un entorno determinado, rutas, caminos y pasajes por donde debe andar un ente visual o personaje (Figura 3.7). Para la aplicación de las reglas de optimización del Path Finding, se tuvo en cuenta la simplificación en la creación de escenarios, en donde hubiese una visión ordenada y en lo posible, simple de este mismo. Otro aspecto fundamental al modelar el Entorno Virtual, es hacer que, dentro de la visión de la creación de un escenario, se creen caminos bien delimitados y lo suficientemente amplios como para que un personaje (de proporción definida por el usuario) pueda pasar a través de estos caminos. La Figura 3.7 muestra un Entorno Virtual, con el fin de probar el Path Finding; en dicho entorno, adicionalmente, se colocaron unos obstáculos aleatorios, con el fin de dificultar el recorrido del personaje hacia los objetivos. En la Figura 3.8, se puede observar el Entorno Virtual renderizado.

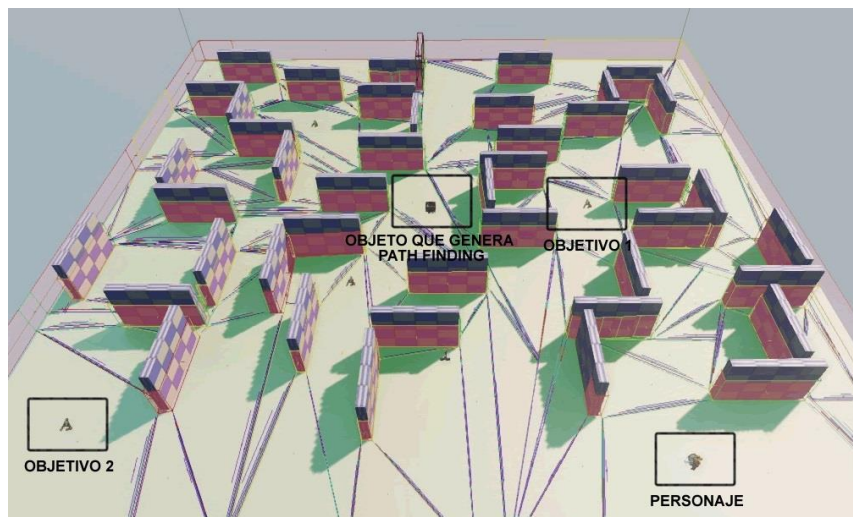


Figura 3.7: EVI para señales de advertencia en un ambiente laboral



Figura 3.8: Entorno Virtual desarrollado con UDK

Las percepciones que se simularon en este Entorno Virtual y su respectiva acción a realizar, a través del razonamiento con la técnica de IA, se representan en la Tabla 3-3.

Percepción	Acción a Ejecutar
Detecta Fuego	Acciona la alarma contra incendios
Detecta Riesgo Eléctrico	Apaga los interruptores de luz
Detecta piso húmedo	Avisa al personal de limpieza
Puesto de Trabajo - No sabe qué hacer	Usa el citófono para pedir ayuda

Tabla 3-3: Percepciones y acciones a ejecutar del personaje en el Entorno Virtual

Posteriormente, en este Entorno, se implementó la MLP, en las máquinas de estados integradas en el código fuente del Game Engine, para realizar varias operaciones dentro de un ambiente laboral controlado. El escenario describe unas oficinas en la que suceden varias señales de advertencia. Cuando el resultado de la RNA arroje como resultado 1, significa que hay presencia de alguna de las señales expuestas previamente (percepciones) y se debe realizar una acción para solucionar este acontecimiento (Figura 3.9).

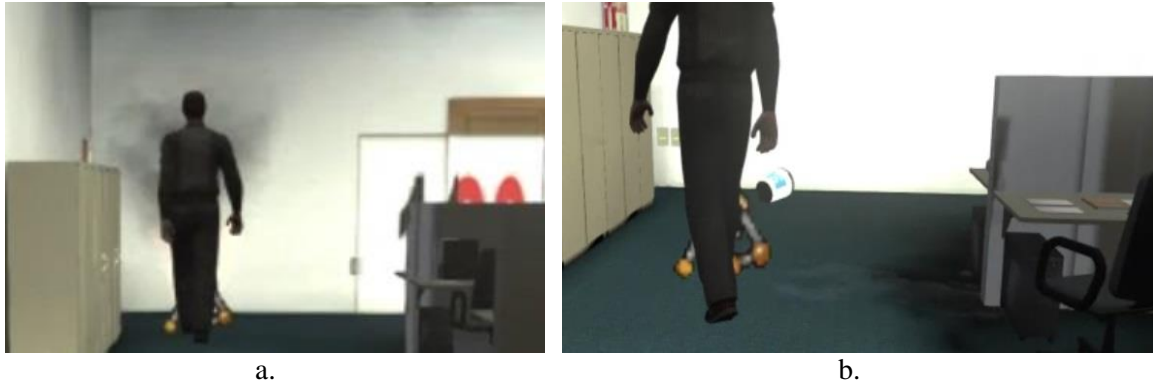


Figura 3.9: a. Personaje identificando el riesgo del fuego. b. Personaje identificando el piso húmedo.

Finalmente, en la Figura 3.10 se puede observar el EVI ejecutándose con la RNA; en este caso, está ejecutándose la acción de usar el citófono para pedir ayuda.

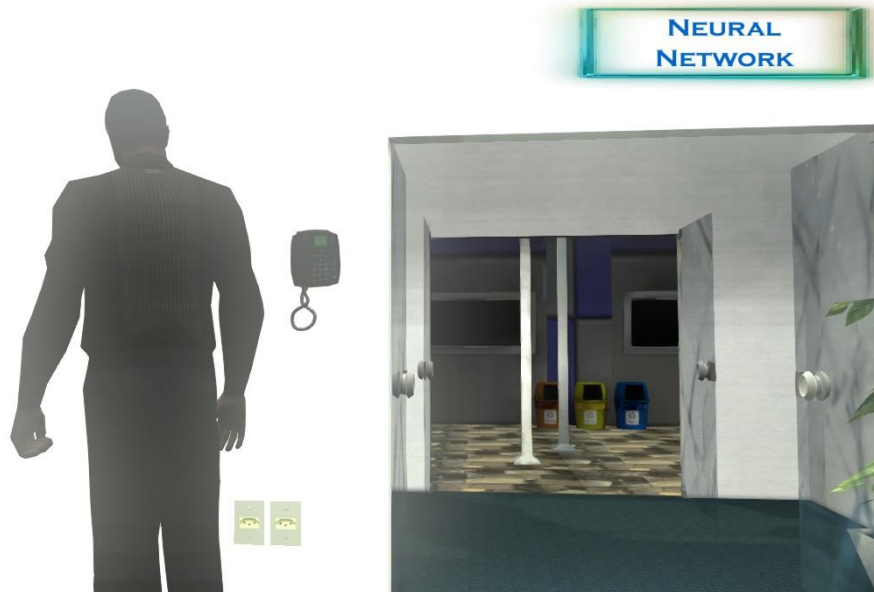


Figura 3.10: EVI con la RNA

Las RNA han generado un gran aporte en diferentes campos, debido a que por medio de patrones, realizan diagnósticos precisos que ayudan a agilizar los procedimientos, como es el caso de los problemas visuales, o comportamientos dinámicos, en el caso del ambiente laboral.

3.3 Entorno Virtual Inteligente con Computación Evolutiva

La Computación Evolutiva es un enfoque alternativo para abordar problemas complejos de búsqueda y aprendizaje a través de modelos computacionales de procesos evolutivos. Las implantaciones concretas de tales modelos se conocen como algoritmos

evolutivos. El propósito genérico de los algoritmos evolutivos consiste en guiar una búsqueda estocástica haciendo evolucionar a un conjunto de estructuras y seleccionando de modo iterativo las más adecuadas (Pérez, 1996). Dentro de los paradigmas existentes en los algoritmos evolutivos, en esta tesis se seleccionaron los Algoritmos Genéticos y las Estrategias Evolutivas, para incorporarlas al Entorno Virtual y compararlas con el fin de evaluar resultados.

3.3.1 Incorporación de un Algoritmo Genético al Entorno Virtual

El objetivo del Algoritmo Genético (AG) es maximizar el resultado de las soluciones candidatas en una población, en razón de una función objetivo o de aptitud desde el dominio de problema. La estrategia para el algoritmo genético es emplear repetidamente sustitutos de los mecanismos genéticos de recombinación y mutación en la población de soluciones candidatas, donde la función de aptitud aplicada a una representación decodificada de un candidato, rige las contribuciones probabilísticas y puede aplicarse a la posterior generación de soluciones candidatas (Brownlee, 2012).

El objetivo al incorporar todas las técnicas de IA, es encontrar cuál genera un resultado superior; en esta aplicación en específico, en un entorno de aprendizaje en donde un personaje tiene como objetivo razonar a partir de una serie de percepciones, para decidir qué acción realizar.

De manera general, en un algoritmo genético se aplican los conceptos de convertir, adaptar, seleccionar parejas, generar una selección aleatoria a través de la combinación de estas parejas y generar mutaciones entre éstas para nuevos individuos.

El proceso de adaptación de esta técnica al entorno virtual, consiste en realizar una valoración aleatoria, teniendo en cuenta el número máximo para un número binario de 5 dígitos, es decir, 1 1 1 1 1, cuyo valor es igual a 32 realizando la conversión de binario a entero. Este valor recibe una operación exponencial la cual es elevar a la potencia 2 (32^2) para obtener un valor de 961.

Almacenados en un arreglo de 4 posiciones, correspondiente a las percepciones descritas (detección de riesgo de fuego, riesgo eléctrico, piso húmedo y puesto de trabajo), el objetivo de este es que cada vez que el personaje revisa cada estado, se ejecutan todas las funciones correspondientes al algoritmo genético. Una vez finalizado un ciclo, es decir, cada vez que el personaje revisa un estado, se evalúan los resultados obtenidos por los cálculos con los generados aleatoriamente en el arreglo.

De manera general, el algoritmo genético utilizado, se describe en la Figura 3.11, el cual es el encargado de llamar a todas las funciones que hacen los diferentes procesos de cálculo en la población y se aplica dentro de la técnica de Pathfinding.

```

AlgoritmoGenetico ()
  ComenzarPoblacion ()
  ConvertirPoblacion ()
  Adaptado = CalidadIndividuo ()
  Adaptabilidad ()
  MostrarPoblacion ()
  SeleccionParejas ()
  Torneo ()
  MostrarGanador ()
  Copiar ()
  MostrarPoblacion ()
  SeleccionParejas ()
  CombinacionMutacion ()
  Si (adaptado >= adaptacionAleatoria[objetivoactual])
    exitoso = true
  Finsi
FinAlgoritmoGenetico

```

Figura 3.11: Algoritmo Genético implementado en el Entorno Virtual

De acuerdo a la Figura 3.11, primero se inicializa la población a evaluar, esto es, desarrollar la matriz con datos, comenzando con el número correspondiente de cada individuo y la creación de su correspondiente número binario y así se crean los cromosomas. Posteriormente, se busca cuál individuo está mejor adaptado, basado en su número binario convertido a entero. La Adaptabilidad del individuo se calcula de acuerdo a la (Ecuación 3.2).

$$f_i = q_i / \sum_j q_j \quad (\text{Ecuación 3.2})$$

en donde q_i representa al individuo y q_j cada uno de los otros individuos. Siendo así, se divide la calidad del individuo por la sumatoria de las calidades de los otros individuos.

Luego, se muestra la población y sus correspondientes parejas, con quienes se cruzarán para generar nuevos individuos con el fin de detectar el mejor adaptado durante los ciclos. La selección de parejas, consiste en asignar una pareja de manera circular inversa. Es decir, el primer individuo se cruzará con el último individuo. El segundo con el penúltimo y el tercero con el antepenúltimo. El torneo es donde se realiza los cálculos al individuo más apto y se saca a un ganador de entre los valores de las parejas. Posteriormente, se realiza copia de los individuos ganadores del torneo y se realiza una redistribución en la matriz para la realización de la operación de combinación y mutación. En la función de combinación y mutación es donde se realizan los cálculos más importantes: se realizan cruces entre las nuevas parejas resultantes de los pasos anterior (Torneo y Copia). Se hace un cruce de genes entre los números binarios, con el fin de resultar un nuevo individuo, producto del cruce, como se plantea en la teoría evolutiva (Figura 3.12).

Al incorporar el Algoritmo Genético en el entorno virtual, fue necesario hacer algunas mejoras, con el fin de realizar un adecuado aprendizaje aplicado desde la

optimización del mismo. Se incorporaron dos nuevas variables: un booleano para identificar la primera ronda de chequeo, es decir, cuando el personaje ya ha recibido las cuatro percepciones efectivamente; y un arreglo de strings para almacenar el valor ganador del cruce, una vez se terminan todos los cálculos. Esto es para que una vez comience un nuevo ciclo de chequeo, los valores de inicialización de la población tengan este valor más apto.

```

CombinacionMutacion ()
  Para ( i=0, i < filas / 2 , 1) haga
    individuoA = poblacion[i][0]
    parejaA = parejas[i]
    cadenaADN = ""
    individuoB = poblacion[parejaA][1]
    puntoCruce = numeroAleatorio entre 0 y 4

    Para ( j=0, j < puntoCruce, 1) haga
      cadenaADN += individuoA[j]
    FinPara

    Para ( j=puntoCruce , j < individuoA.tamaño, 1)
      cadenaADN += individuoB[j]
    FinPara

    poblacionTemporal[i][0] = i
    poblacionTemporal[i][1] = cadenaADN
  FinPara

  mutado = (parejas.tamaño / 2) + 1
  individuoA = poblacion[mutado][1]
  gen = numeroAleatorio entre 0 y 4

  Si (individuoA[gen] == "0") ent
    individuoA[gen] = "1"
  sino
    individuoA[gen] = "0"
  FinSi
FinCombinacionMutacion

```

Figura 3.12: Cruce y Mutación en el Algoritmo Genético

Una vez hecho el primer ciclo de chequeos, la variable auxiliar de chequeo se vuelve falsa con el fin de permitir un nuevo ciclo, incluyendo el valor más apto resultado del primer ciclo de cada estado. Desde la primera vuelta se comienza a guardar los valores en el arreglo de los individuos mejor adaptados de la población y guardados en la posición correspondiente a cada estado y así cuando la primera ronda termina, la función que inicializa la población se modifica con el fin de agregar en primera fila al mejor adaptado hasta entonces.

En la Figura 3.13, se puede observar una de las pruebas del Algoritmo Genético en el entorno virtual, en el cual se encuentra el personaje y los diferentes puntos de control que

simulan las percepciones, el razonamiento mediante el AG y posteriormente la acción que debe realizar.



Figura 3.13: Algoritmo Genético en el Entorno Virtual

En este punto, al comparar las técnicas desarrolladas (RNA y AG), se evidencia que la RNA tiene su concepción directa aplicable a la Inteligencia Artificial orientada al aprendizaje, mientras que los AG se basan en leyes y teorías de la evolución de las especies de la naturaleza, en la cual se muestra como los individuos se adaptan al cruzarse unos con otros, para resultar en nuevas y mejores especies. Esto se evidencia de igual manera en el desarrollo de esta tesis: la RNA tuvo un desarrollo y adaptación más factible por su misma naturaleza de aprendizaje, mientras que el AG siendo este pensado para la optimización al momento de solucionar problemas complejos, tuvo una adaptación más inverosímil al momento de incorporarla al entorno virtual.

Finalmente, en la Figura 3.14 se puede observar el EVI ejecutándose con el AG; en este caso, está ejecutándose la acción notificar al personal de limpieza.

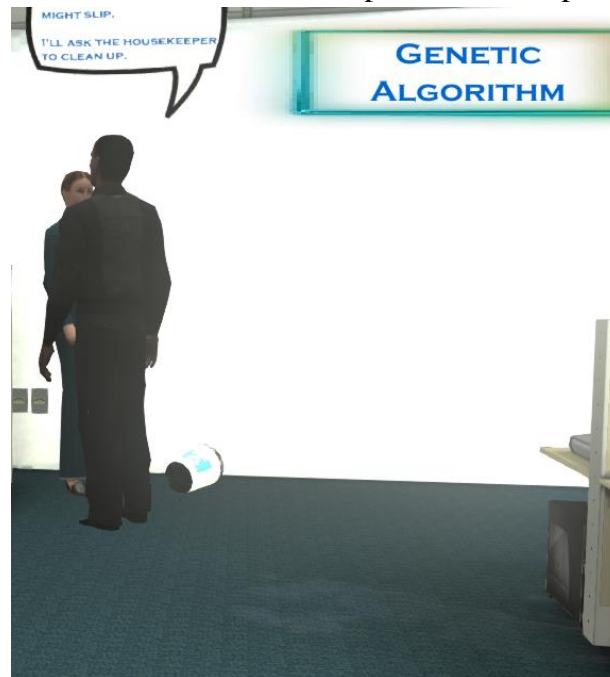


Figura 3.14: EVI con el AG

3.3.2 Incorporación de una Estrategia Evolutiva al Entorno Virtual

El objetivo de las Estrategias Evolutivas (EE) es maximizar la idoneidad de la colección de soluciones candidatas en el contexto de una función objetivo de un dominio. El objetivo se logra mediante la adopción de variación dinámica, de un sustituto para los descendientes con modificaciones, donde la cantidad de variación se adapta dinámicamente con heurísticas basadas en el rendimiento. Enfoques contemporáneos co-adaptan los parámetros que controlan la cantidad y el sesgo de variación con las soluciones candidatas (Brownlee, 2012).

La Estrategia Evolutiva cuenta con similitudes al Algoritmo Genético, en donde se cuenta con una población de individuos que tienen como objetivo unirse, combinarse o cruzarse, con el fin de generar nuevos individuos mejor adaptados para cumplir el rol de la evolución dentro de la población creada inicialmente.

Si se realiza una comparación para encontrar las diferencias entre estas dos técnicas, se puede concluir que: en las EE, los valores de la población se interpretan como vectores de número reales; al momento de la reproducción, los padres son seleccionados de manera aleatoria, los hijos se generan y se insertan en la futura generación; estos nuevos individuos se auto adaptan y la mutación se codifica en el individuo. En los algoritmos AG los individuos se interpretan como número enteros; se seleccionan padres de acuerdo a sus capacidades, es decir, primero se evalúan antes de realizar el primer paso evolutivo.

La EE tiene como objetivo optimizar a través de cálculos que simulan el comportamiento o estrategias de los genes, la generación de nuevas especies en la naturaleza, es decir, en el entorno virtual se usan estas estrategias para crear patrones de aprendizaje, no solo el aprendizaje a nivel de cálculo, sino también en el comportamiento que podría lograr un ente sintético y simular este comportamiento en un entorno virtual.

En el desarrollo de esta técnica para incorporarla al entorno virtual, se crearon variables de control de población, cantidad de genes y número de hijos que representan a las nuevas generaciones. Realizando una selección aleatoria de valores con los cuales competir al momento de ejecutar el código, se tiene como objetivo ver qué tan optimizado es la generación de las estrategias contra la aleatoriedad. Además se crearon dos estructuras de datos y un arreglo cuyos propósitos son: una estructura funciona como almacenamiento temporal de una matriz de datos; otra estructura almacena los datos de vectores, estrategias y aptitud; y en el arreglo se almacenan datos aleatorios con los que se competirá luego de los cálculos del algoritmo. Al igual que en el AG, primero se inicializa la población a evaluar. Luego, se realiza una búsqueda que tiene como objetivo encontrar al individuo mejor adaptado (Figura 3.15).

```

Busqueda ()
  PoblacionEstrategiaEvolutiva[]
  HijosEstrategiaEvolutiva

  Poblacion = IniciarPoblacion (EspacioBusqueda, TamanoPob)
  mejor= Poblacion[0] // organizado de menor a mayor

  Para ( i=0 , i < max_genes, 1) haga

    Para (j = 0, j < num_hijos, 1) haga
      hijos = mutacion (Poblacion[i], EspacioBusqueda)
    FinPara

    Para (j=0, j < num_hijos, 1) haga
      hijo[j].aptitud = function_objetivo (hijo[j].vec)
    FinPara

    union = hijos+poblacion

    Si (union[0].aptitud] < mejor.aptitud) ent
      mejor = union.primerero
    FinSi

    poblacion = union.primerero(pop_size)
  FinPara

  Retorne mejor
FinBusqueda

```

Figura 3.15: Búsqueda en la Estrategia Evolutiva

El algoritmo de la función objetivo se describe en la Figura 3.16.

```

FuncionObjetivo ()
  Entero of_i
  Real acum=0

  Para (of_i= 0, of_i < vec.Length, 1) haga
    acum = acum + Square(vec[of_i]);
  FinPara

  Retorne acum
FinFuncionObjetivo

```

Figura 3.16: Función Objetivo de la Estrategia Evolutiva

Finalmente, luego de ser creada la población, dentro de la función de búsqueda, se hace la mutación cuyo objetivo es realizar las operaciones para los vectores y las estrategias de los individuos en la población. En estas mutaciones, se aplica el método de Gauss-Newton, usando parámetros aleatorios.

Finalmente, en la Figura 3.17 se puede observar el EVI ejecutándose con la EE; en este caso, está ejecutándose la acción de ir a la alarma de incendios.



Figura 3.17: EVI con la EE

Como conclusiones preliminares, se puede establecer que para ambos casos, tanto los AG como las EE, entre mayor es la población, mayor es el tiempo de cálculo, pero esto beneficia la precisión para operaciones que requieran rapidez en la entrega de respuestas en menos iteraciones.

El entorno virtual desarrollado, se presta para realizar la operación con poblaciones reducidas, al igual que con otros de los parámetros para la solución del problema. Aún con pocos datos, la velocidad de respuesta y efectividad en los dos casos a comparar tienen un tiempo de respuesta similar, siendo óptimos los AG que, aunque cuentan con un código más extensos, no cuentan con tantas variables alojadas en arreglos multidimensionales como en el caso de las EE, cuya operación, aunque más concisa, cuenta con muchas iteraciones que afectan el tiempo de ejecución en tiempo real.

3.4 Entorno Virtual Inteligente con Agentes Inteligentes

Un agente puede ser un sistema reactivo o bien deliberativo, que presenta cierto grado de autonomía, en el sentido de que se le puede delegar una tarea a éste, y el mismo sistema determina la mejor manera de llevar a cabo esta tarea. A tales sistemas se les llama agentes, porque se piensa en ellos como seres activos, productores útiles de acciones: se envían a su entorno para lograr metas y persiguen activamente estas metas, averiguando por sí mismos la mejor forma de lograr estos objetivos (Bordini, Hübner, & Wooldridge, 2007).

En esta tesis, se utilizó la metodología orientada a agentes denominada Prometheus, la cual consiste de tres fases a seguir (Padgham & Winikoff, 2005): especificación del

sistema, diseño de la arquitectura y diseño detallado, la cual se puede observar en la Figura 3.18. Para esta parte, también se hicieron algunas pruebas con JASON implementado en UnrealScript, pero debido a la alta recursividad de JASON al hacer tantas iteraciones, hace que Unreal se bloquee porque limita el número de iteraciones por cuadro por segundo, además de que sacrifica el tiempo real, al hacer sockets entre JASON y Visual C++.

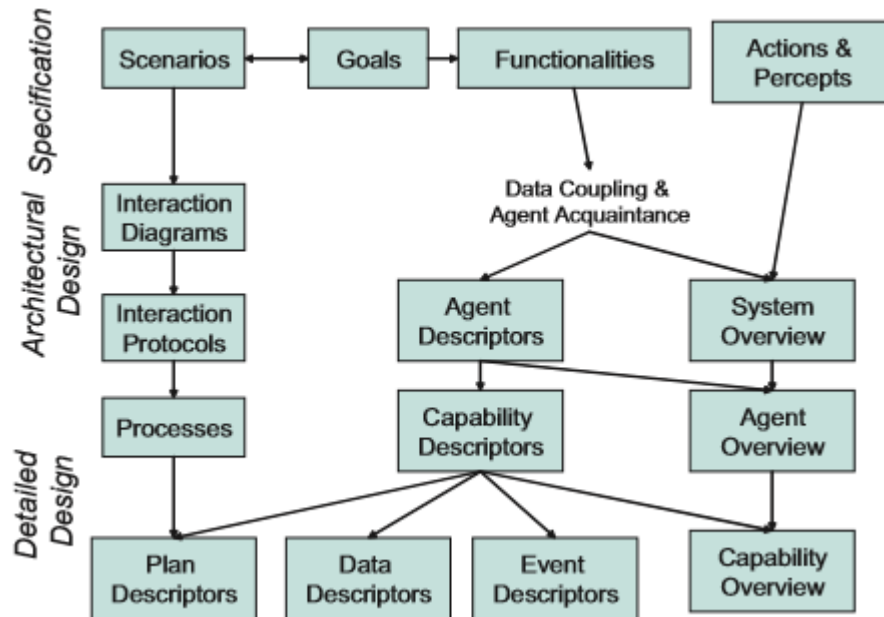


Figura 3.18: Fases de la Metodología Prometheus (Padgham & Winikoff, 2005)

3.4.1 Especificación del Sistema

Basados en el Entorno Virtual propuesto y desarrollado, la Figura 3.19 muestra la visión general de las especificaciones del sistema, la cual se realizó basada en el caso presentado por (Rahman & Kamil, 2008).

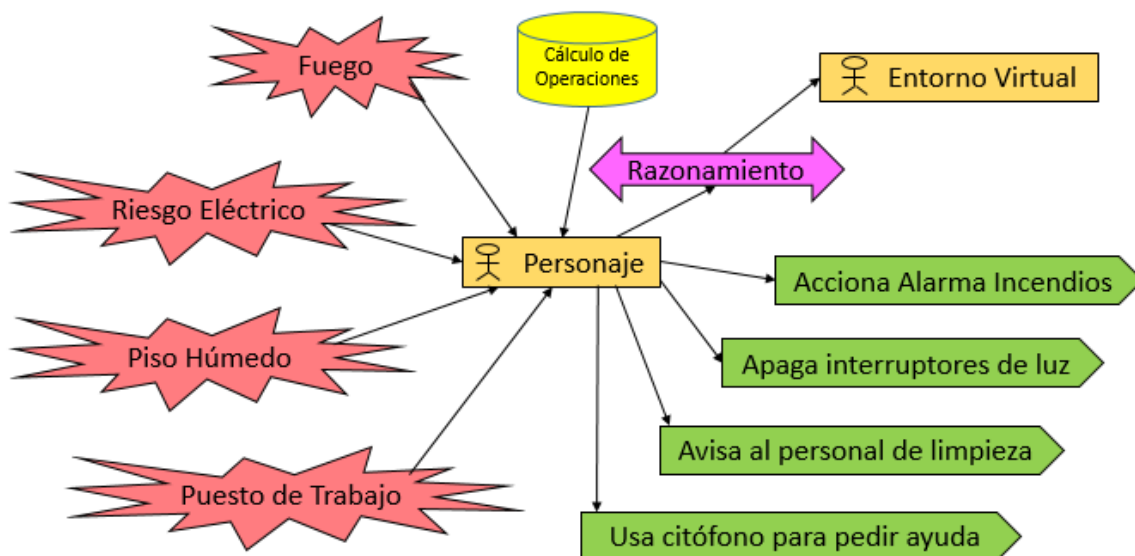


Figura 3.19: Diagrama del Entorno Virtual basado en Prometheus

Objetivos Generales

Para el caso de estudio, de brindar señales de advertencia en un ambiente laboral, los objetivos serían básicamente dos: Brindar información gráfica a las personas sobre lo que se debe hacer al percibir una señal que pueda representar algún tipo de peligro (Objetivo #1) y con lo anterior maximizar la seguridad en un entorno laboral (Objetivo #2). Como sub-objetivos se tendrían: superar algún tipo de obstáculo (Objetivo #3), evitar un posible accidente (Objetivo #4).

Funcionalidades

Se han definido dos funcionalidades para manejar y operar la interacción durante la ejecución del Entorno Virtual, con el fin de alcanzar los objetivos: Identificar posibles obstáculos (Funcionalidad #1) que para el caso de estudio, serían los cubículos de las oficinas y paredes, y determinar la ruta factible (Funcionalidad #2), esto es, tomar la acción adecuada, acorde a la percepción recibida y a la aleatoriedad aplicada al ejecutar la acción.

Escenarios

Los escenarios realizados son: “hay humo en un rincón” y con ello se activa la percepción de detectar fuego; “hay chispas eléctricas saliendo de una lámpara” y esto activa la percepción de identificar el riesgo eléctrico; “hay agua en el piso” y es cuando detecta el piso húmedo y “dirigirse al puesto de trabajo”, cuando no sabe qué hacer. También estarían como escenarios, “acercándose a la alarma de incendios”, “dejando el entorno a oscuras y simular un bajón de luz”, “conversando con la señora encargada de la limpieza” y “acercándose al citófono”, que correspondería a cada acción a ejecutar.

3.4.2 Diseño de la Arquitectura

Agentes

En el Entorno, hay dos tipos de agentes involucrados en el diseño de la arquitectura: Agente Personaje y Agente Entorno virtual. El Personaje es el actor principal y es capaz de responder, reaccionar e interactuar con el Entorno, a través del algoritmo de Colonia de Hormigas, para calcular la ruta adecuada de la acción a realizar en tiempo real. El Entorno en sí, sería el otro agente (actor externo), con el que interactúa el personaje.

Percepciones

Las percepciones del personaje son las mismas descritas en la Tabla 3-3: detectar fuego (P#1), detectar riesgo eléctrico (P#2), detectar piso húmedo (P#3) y el puesto de trabajo (percepción P#4).

Acciones

De acuerdo a la percepción, están las acciones a realizar: accionar alarma contra incendios (A#1), apagar interruptores de luz (A#2), avisar a personal de limpieza (A#3) y usar citófono para pedir ayuda (A#4).

Protocolo

Este define la interacción entre el Agente Personaje y el Agente Entorno Virtual y definirá la acción a realizar o ruta factible y está compuesto por los siguientes mensajes:

- Actualización Ubicación: Este mensaje es de tipo informativo y es enviado desde el agente personaje al agente entorno. Cuando el agente personaje cambia su ubicación, debe informarla al entorno.
- Cambio en el entorno: Este mensaje es de tipo petición-respuesta. Una vez el agente personaje realiza un cambio en el entorno (activar alarma, por ejemplo) le informa al agente entorno, que a su vez, debe responder si dicho cambio es permitido o no, así como actualizar la información del entorno.

Acoplamiento de Datos

Muestra el proceso de la transferencia de datos en el sistema, en el agente Entorno, tanto para transferir como para recibir del agente Personaje.

Los datos están almacenados en un sistema gestor de bases de datos, al cual pueden acceder los dos agentes y los datos compartidos entre los agentes son: ubicación del agente personaje y acciones realizadas en el entorno.

3.4.3 Diseño Detallado

Agente Personaje

Este agente de tipo deliberativo, determina la acción a realizar dado el razonamiento que hace a partir de la percepción recibida. El diagrama de este agente, se puede observar en la Figura 3.20.

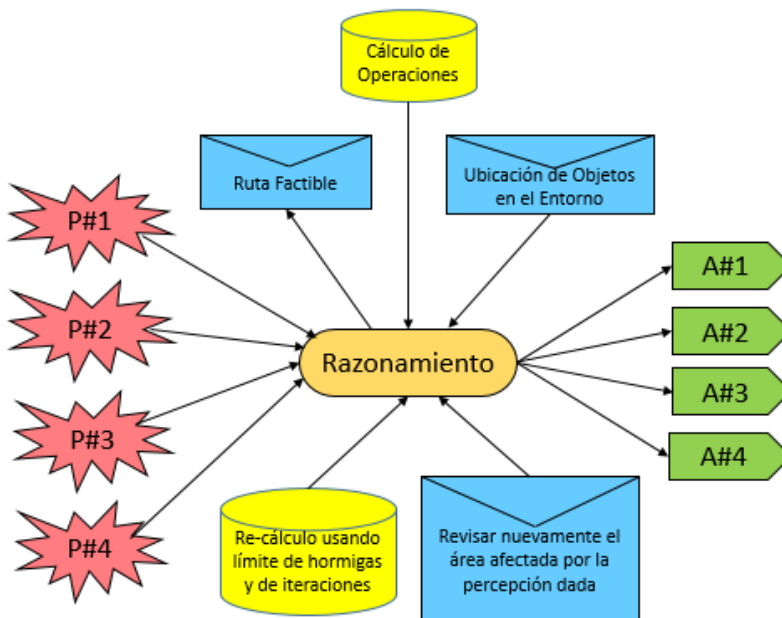


Figura 3.20: Agente Personaje

Cuando el personaje empieza a interactuar con el entorno, el agente recibe información sobre la ubicación de objetos en el entorno. El cálculo de operaciones y el re-cálculo del algoritmo tienen la información de los datos que se manejan. La acción a realizar es calculada por el algoritmo de Colonia de Hormigas.

El Sistema de Colonia de Hormigas tiene como principio el usar las técnicas de enjambre y la generación de feromonas como medio de comunicación para crear caminos con destinos, como el mismo enjambre y lugares con alimentos. Como metáfora, cuando se comienza la creación de un enjambre, las hormigas deambulan alrededor de manera aleatoria con el fin de crear caminos esparciendo sus feromonas y así múltiples viajes se realizan para lograr dar forma a un patrón organizado. Para esto, se usó como base el algoritmo propuesto por (Brownlee, 2012) (Figura 3.21).

```

Input: ProblemSize, Populationsize, m, ρ, β, σ, q0
Output: Pbest
1 Pbest ← CreateHeuristicSolution(ProblemSize);
2 Pbestcost ← Cost(Sh);
3 Pheromoneinit ←  $\frac{1.0}{\text{ProblemSize} \times P_{\text{best}_{\text{cost}}}}$ ;
4 Pheromone ← InitializePheromone(Pheromoneinit);
5 while ¬StopCondition() do
6   for i = 1 to m do
7     Si ← ConstructSolution(Pheromone, ProblemSize, β,
8       q0);
9     Sicost ← Cost(Si);
10    if Sicost ≤ Pbestcost then
11      Pbestcost ← Sicost;
12      Pbest ← Si;
13    end
14    LocalUpdateAndDecayPheromone(Pheromone, Si, Sicost,
15      σ);
16  end
17  GlobalUpdateAndDecayPheromone(Pheromone, Pbest,
18    Pbestcost, ρ);
19 end
20 return Pbest;

```

Figura 3.21: Algoritmo de Colonia de Hormigas (Brownlee, 2012)

Este algoritmo es reconocido por realizar gran cantidad de ciclos de manera recursiva para llegar a una solución óptima. Por esto requiere unos límites para llegar a dicho resultado. Estos se determinan por la cantidad de hormigas y una cantidad determinada de ciclos o repeticiones para llegar al cálculo óptimo; lo que se quiere es hacer que el cálculo del agente, supere a la aleatoriedad en el menor número de repeticiones posibles.

En este algoritmo, se usan dos estructuras para almacenar los datos: una, almacena los lugares que han de visitar las hormigas y otra, que almacena los datos del lugar más adecuado de manera temporal, que son finalmente, las acciones a realizar. Se tienen en cuenta el número máximo de iteraciones y el número máximo de hormigas.

La función principal de este algoritmo, consiste en llamar la función de búsqueda (Figura 3.22) que tiene como objetivo identificar cuál es el recorrido óptimo, es decir, en cada estado del personaje, se debe solucionar todo el algoritmo y solo puede pasar al siguiente estado, cumpliendo con tener un valor óptimo menos que el propuesto. La función de búsqueda tiene como objetivo encontrar, haciendo uso de cálculos probabilísticos de otras funciones, el mejor recorrido dadas unas distancias; para dichas distancias se utilizó el algoritmo Berlin52 (Traveler Salesman Problem) (Brownlee, 2012).

```

BúsquedaAgentes ()

    mejor.vector = permutación_aleatoria (lugares);
    mejor.cost = cost (mejor.vector, lugares);
    inicializar_feromona =  $1.0 / (\text{lugares.tamaño} * \text{mejor.cost})$ 
    feromona = init_feromona_matriz (lugares.tamaño, inicializar_feromona);

    Para (s_i=0, s_i < max_it, 1) haga
        Para (s_j=0, s_j < num_hormigas, 1) haga
            candidato.vector = pasoapaso (lugares, feromona, heuristica, codicia)
            candidato.cost = costo (candidato.vector, lugares)

            Si (candidato.cost < mejor.cost) ent
                mejor = candidato
            FinSi
        FinPara
    FinPara

    Retorne mejor
FinBúsquedaAgentes

```

Figura 3.22: Función de Búsqueda en el algoritmo de Colonia de Hormigas

En el algoritmo descrito en la Figura 3.22, la permutación aleatoria realizar un trueque aleatorio de los lugares. De esta manera siempre se va a tener un objetivo diferente y los cálculos siempre serán diferentes, aunque la finalidad será la misma. La función de costo, entrega el cálculo de distancia entre los lugares. La matriz feromona, ayuda en el cálculo del recorrido del personaje.

Posteriormente, se realizan los cálculos según un resultado aleatorio que da como resultado, o ir por un camino “codicioso”, o por un camino “de probabilidad”. La diferencia es que el camino “codicioso”, usa un cálculo dado previamente y el otro camino recalcula la probabilidad dada, para un resultado más adecuado.

También interviene una función de selección codiciosa, la cual compara unos valores generados previamente para identificar cuál es el mayor entre una lista y entregar el siguiente lugar al cual se debe viajar. Y la selección probabilística recalcula, usando los valores previos de la probabilidad, para determinar el siguiente mejor lugar a visitar.

Agente Entorno Virtual

Este Agente se encarga en sí, de proporcionarle al personaje la ubicación de los objetos del entorno y de mostrar gráficamente las acciones realizadas por el personaje.

Finalmente, en la Figura 3.23 se puede observar el EVI ejecutándose con los Agentes Inteligentes; en este caso, está haciendo el razonamiento al detectar el riesgo eléctrico, debido a que hubo un bajón de luz en el entorno.

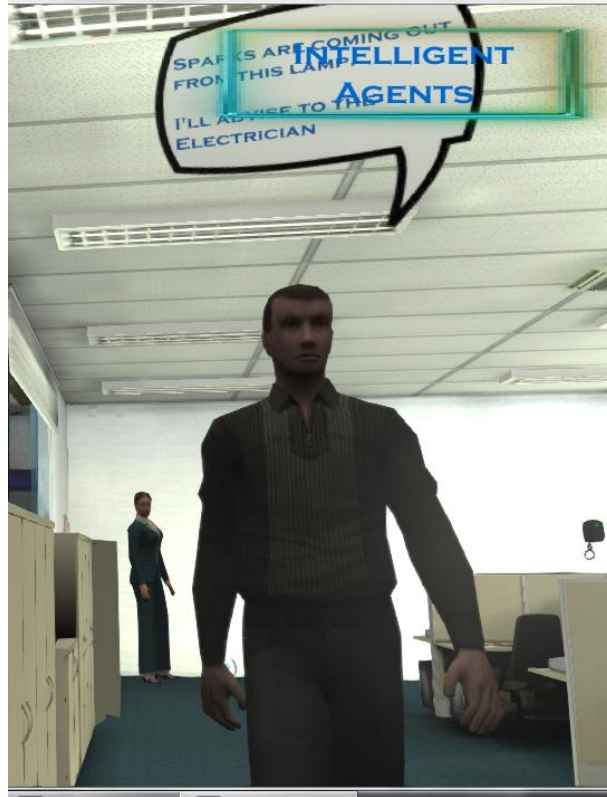


Figura 3.23: EVI con Agentes

En conclusión, el objetivo fundamental se convierte en encontrar el recorrido óptimo, sin tener que realizar desplazamientos extras. La técnica lo soluciona efectivamente, inclusive a un nivel bastante alto, en donde, sin ayuda de datos reales suministrados y usando sólo cálculos probabilísticos, llega a diferentes soluciones óptimas, dadas las repeticiones y el número de hormigas en el agente Personaje. Cuanto más tiempo tenga esta técnica para solucionar el problema, entregará un resultado aún más adecuado.

La técnica cuenta con un sistema de cálculos que si bien son simples, entran en complejidad al momento de realizar operaciones con una herramienta como los motores de videojuegos. Al tener que realizar tantas iteraciones dados los parámetros de número máximo de hormigas y de iteraciones, esto puede saturar los límites establecidos por cada herramienta. En este caso UDK cuenta con un límite de iteraciones por cada cuadro por segundo (FPS) y una vez superado esto, la herramienta simplemente deja de funcionar; lo anterior, hace la implementación y potencial de esta técnica también se vea reducido por las limitaciones del software usado.

Finalmente, en la Figura 3.24 se muestra cómo se selecciona la técnica a ejecutarse en el EVI desde el editor de UDK.

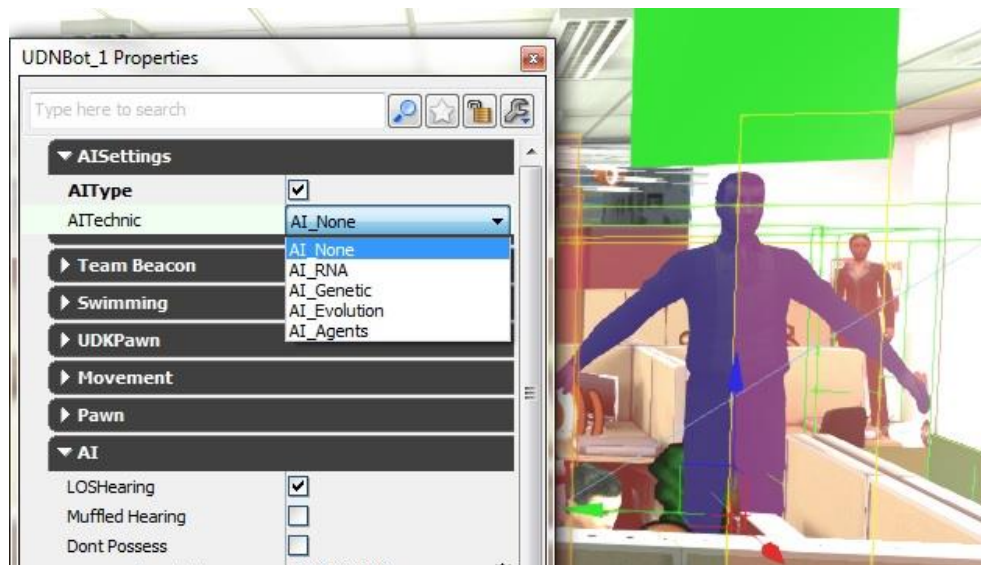


Figura 3.24: Selección de la Técnica de IA desde el editor de UDK

3.5 Conclusiones

De manera general, para la mayoría de solución de problemas se pueden requerir métodos de optimización para encontrar el mejor resultado posible, como los Algoritmos Genéticos o los Agentes Inteligentes, las cuales se caracterizan por usar métodos en donde se determina una población y entre más grande sea, mejor oportunidad hay de encontrar un resultado más adecuado. Estas técnicas son ideales para operaciones en donde no se requiere una respuesta muy rápida y que al mismo tiempo, no es relevante si se sacrifica el desempeño de máquina, a la vez que se pueden desarrollar a nivel de CPU sin requerir especificaciones técnicas muy altas.

Para efectos visuales, en donde se requiere resultados en tiempo real y en donde no se permite superar un número determinado de operaciones por ciclos (lo que se sucede en la mayoría de aplicaciones de Realidad Virtual), las Redes Neuronales cuentan con una gran desempeño, a poco costo de operación, es decir, se puede lograr un nivel muy alto de aprendizaje y optimización, sin sacrificar desempeño y sin requerir especificaciones técnicas muy altas, lo cual es ideal para su aplicabilidad en Realidad Virtual.

4. Selección de la Técnica de Inteligencia Artificial para su Implementación en un Entorno Virtual Inteligente

En este capítulo, se realizan varias comparaciones entre las técnicas de IA implementadas, con el fin de demostrar entre éstas, la que presente un desempeño más adecuado en un Entorno Virtual. Para ello, se detallarán algunas métricas a nivel cualitativo y cuantitativo (Restrepo, 2012) haciendo con ello una evaluación de desempeño del Entorno Virtual con las técnicas de IA implementadas. En esta comparación también se incluyeron algunos conceptos sugeridos por (Zhang, 2010).

En la evaluación cualitativa, se utilizarán como métricas las siguientes: interactividad, conectividad, ubicuidad, invisibilidad, adaptabilidad y sensibilidad; y en la evaluación cuantitativa: especificaciones técnicas, latencia, completitud y escalabilidad.

Este capítulo está organizado de la siguiente manera: En la sección 4.1, se explica la evaluación del EVI de forma cualitativa; en la sección 4.2, se muestra la evaluación del EVI de forma cuantitativa; y por último se detallan las conclusiones.

4.1 Evaluación Cualitativa

Según (Restrepo, 2012), un sistema inteligente se puede evaluar de forma cualitativa mediante un vector conformado por seis características intrínsecas, las cuales son: interactividad, conectividad, ubicuidad, adaptabilidad y sensibilidad al contexto. A continuación se realiza la explicación de cada una, adaptadas a tres de las técnicas de IA implementadas.

4.1.1 Interactividad

Esta característica está relacionada con la reducción de la interacción explícita debido a que el sistema se supone que usa su inteligencia para inferir las situaciones y necesidades del personaje a partir de las actividades observadas.

- En la RNA es Alta. La forma que están dispuestos los cálculos hacen que la RNA tenga un alto nivel de interactividad, puesto que una vez que percibe un estado dado dentro del entorno, este realiza las operaciones pertinentes de manera inmediata y sin variación en el apartado gráfico, generando confianza en la operación realizada.
- En el AG es Alta. Los AG al percibir los eventos dentro del entorno, realiza las acciones pertinentes.

- En los Agentes es Alta. Esta técnica se comporta de manera asimilativa, debido a que reconoce los estados del entorno para tomar las decisiones pertinentes.

4.1.2 Conectividad

Radica en la búsqueda de la libertad que se proporciona al personaje para escoger qué tipo de información desea, cuándo y dónde.

- En la RNA es Alta. Los cálculos dispuestos dentro de la técnica se realizan de manera eficiente, a una gran velocidad de percepción.
- En el AG es Media. Las operaciones realizadas según el evento dado son optimizadas, aunque altas en complejidad. Esto limita el nivel de optimización máximo que pueda suministrarse en las operaciones.
- En los Agentes es Baja. Esta técnica con los cálculos que realiza, tiene un alto consumo de recursos técnicos. Aplicada en las herramientas, hace que éstas lleguen a sus límites, lo cual no lo hace recomendada. Si bien genera unos resultados no sólo óptimos, sino también organizados, aplicarla en un ambiente de tiempo real, implica que debe hacerse de manera limitada.

4.1.3 Ubicuidad

Cualidad de la computación de estar presente en todas partes.

- En la RNA es Media. La RNA se desarrolló en un entorno muy controlado, con pocas entradas, pero se podría readaptar el código de manera que pueda aceptar más libertad de movilidad.
- En el AG es Media. Igual que la RNA se desarrolló bajo un entorno muy controlado.
- En los Agentes es Media. A pesar de que se desarrolló en entorno muy controlado también, permitió ampliar este control relativamente más sencillo que en las otras técnicas.

4.1.4 Adaptabilidad

Habilidad de los entornos de ser altamente adaptativos, característica que se relaciona en gran medida en cómo las personas interactúan con el entorno.

- En la RNA es Alta. La forma en que esta técnica se implementó en los entornos, muestra cómo puede aprender a desempeñarse y adaptarse a éste, de manera ágil y rápida.
- En el AG es Alta. Una vez que el personaje logra solucionar una de las percepciones, la técnica se adapta fácilmente al entorno.

- En los Agentes es Media. A pesar de que arroja un resultado óptimo, su adaptabilidad se encuentra limitada a los parámetros dados.

4.1.5 Sensibilidad al Contexto

Un sistema es sensible al contexto si éste usa el contexto para proporcionar información relevante y/o servicios al usuario, donde la relevancia depende de la tarea del usuario.

- En la RNA es Alta. La manera en que se identifican efectivamente las percepciones y qué decisión debe tomarse según un caso dado, hace que esta técnica tenga un alto nivel de sensibilidad, debido a que sabe dónde se encuentra y qué debe hacer en los casos específicos.
- En el AG es Media. : Al referirse a un aprendizaje real de máquina, se puede inferir que esta técnica no está consciente de encontrar un objetivo, sino de entregar el mejor resultado posible según sus operaciones, sin un objetivo muy claro.
- En los Agentes es Alta. Los agentes en sí, no están muy enterados de su entorno sino que, basado en sus cálculos de optimización, logran llegar a una solución específica, sin que implique saber lo que están haciendo.

El resumen de lo anterior, se pueden observar en la Tabla 4-1.

Característica	Red Neuronal	Algoritmos Genéticos	Agentes Inteligentes
Interactividad	Alta	Alta	Alta
Conectividad	Alta	Media	Baja
Ubicuidad	Media	Media	Media
Adaptabilidad	Alta	Alta	Media
Sensibilidad al Contexto	Alta	Media	Alta

Tabla 4-1: Evaluación cualitativa del Entorno Virtual Inteligente

De la anterior tabla, se puede inferir que la RNA entregó resultados de manera superior sobre las otras técnicas comparadas, basado en las métricas cualitativas utilizadas.
Evaluación Cuantitativa

Para realizar la evaluación cuantitativa (Restrepo, 2012), se adoptan las siguientes métricas: especificaciones del equipo, latencia, completitud y escalabilidad.

4.1.6 Especificaciones del Equipo

Esta métrica adopta uno de los siguientes valores: 0, para características deficientes del equipo comparadas con los últimos dispositivos hardware en el mercado y el número de

tareas a realizar; 1, para características óptimas del equipo comparadas con los últimos dispositivos hardware en el mercado y el número de tareas a realizar.

En el caso del Entorno Virtual Inteligente creado, la métrica de especificaciones del equipo vale 1, dado que es necesario tener un computador con la suficiente potencia gráfica para ejecutar la aplicación y pueda tener un alto grado de realismo. Al realizar el entorno bajo la herramienta UDK, por ser un ambiente de desarrollo que entrega una aplicación con render en tiempo real de alta calidad, éste necesita, de una manera especial, tarjetas aceleradoras de gráficos. En el entorno desarrollado en este trabajo se utilizaron dos equipos; el de escritorio cuenta con las siguientes características:

- Procesador: Intel Core 2 Quad 6600 2.40 GHz
- Memoria RAM: 8 GB Corsair Vengeance 1600 MHz
- GPU: EVGA GTX 480 1.5 GB

Y para el caso del portátil, las especificaciones son:

- Tipo: HP Pavilion dv6-12771a
- Procesador: AMD Turion (tm) X2 Dual Core Mobile RM-75 2.20 GHz
- Memoria RAM: 4 GB.

El motor de videojuegos utilizado para el desarrollo del EVI, hace que se den múltiples procesos, desde el cálculo hasta representación física visual que se aplica en el ámbito de renderización. De ahí las especificaciones tanto de CPU, como de GPU. Mientras que las técnicas se desarrollan netamente en CPU, la representación visual de éstas, se hacen en el motor de render de las herramientas.

4.1.7 Latencia o Tiempo de Respuesta del Entorno

Esta métrica se define en esta tesis, como el tiempo que tarda el entorno en realizar una o varias tareas, orientadas en lograr la adaptación del personaje presente en el entorno. Las medidas en esta métrica, se tomaron teniendo en cuenta el tiempo que toma en realizar todas las operaciones asignadas, sabiendo que se están ejecutando en tiempo real, en un motor de render que tiene un número determinado de iteraciones por cada cuadro que se dibuja por la pantalla, es decir, si el personaje solicita recalcular la técnica actual, ésto se hace, en totalidad en un solo cuadro. Los motores de render tienen como objetivo alcanzar 60 cuadros en un segundo. Cada cuadro tiene una presencia en pantalla de aproximadamente 0,016 segundos ($1 \text{ segundo} / 60 \text{ cuadros} = 0,016$), es decir, que la operación debe realizarse de manera que no afecte el flujo de cuadros por segundo. Para evidenciar las medidas de cuadros por segundo, se toman las muestras mientras la aplicación está en ejecución, a través de uno de los comandos de UDK y ésta muestra los datos de cuadros por segundos y mili segundos en tiempo real según la máquina en la que está ejecutándose.

Para evaluar la latencia, también se tomaron algunas muestras y los datos están en el Anexo 1. Las muestras son tomadas de 10 iteraciones que se hicieron en cada una de las técnicas, por ejemplo, la RNA se ejecutó 10 veces, y se tomaron las muestras de cuántas

veces debe repetir una acción para pasar a la siguiente percepción; entre menos repeticiones, mejor optimizada está la operación, pero esto no refleja necesariamente un aprendizaje adecuado.

Con base en todo lo anterior, se decide también que la RNA se comporta más adecuadamente y presenta menor consumo de máquina.

Red Neuronal Artificial

Esta técnica cuenta con el menor tiempo de respuesta, lo que genera una gran similitud al momento de compararla con el aprendizaje humano. Cada vez que el personaje se encuentra con una percepción, debe comparar los resultados iniciales con las acciones que debe tomar y en caso de ser equivocadas, debe recalcular la curva generada inicialmente.

En la muestra tomada se puede apreciar que no afecta el desempeño en la cantidad de cuadros por segundo, por lo que se puede decir que cada vez que se realiza un re-cálculo, le toma al computador alrededor de 0,016 y 0,019 segundos para completarse (de acuerdo a lo explicado anteriormente, de alcanzar los 60 cuadros por segundo o 0,016).

Por otra parte, en la Tabla 4-2, se muestra un resumen del tiempo que toma en razonar en cada percepción, para tomar la acción adecuada, usando la RNA (La tabla completa se encuentra en el Anexo A).

Percepción	Número de Repeticiones	Latencia (Segundos)
Detecta Fuego	2	40,19
Detecta Riesgo Eléctrico	0	2,82
Detecta Piso Húmedo	0	0,85
Puesto de Trabajo	0	0,75

Tabla 4-2: Latencia basada en repeticiones para la RNA

Algoritmos Genéticos

Los AG, al momento de tomar las medidas de recalculan sus funciones, generan una disminución en el conteo de cuadros por segundo muy leve, que hace que la aplicación pase de 60 a alrededor de 50 o 40 cuadros por segundo. Esto, sólo se evidencia haciendo pruebas de rendimiento a nivel de debug interno de la herramienta. Al no ser un cálculo que se hace todo el tiempo, sino cada vez que el personaje no sabe qué hacer, entonces no es significativo la reducción en cuadros por segundo. El cálculo en promedio para esta técnica fue entre 0,02 y 0,025 segundos aproximadamente (de acuerdo a lo explicado anteriormente de alcanzar los 60 cuadros por segundo o 0,016).

Por otra parte, en la Tabla 4-3, se muestra un resumen del tiempo que toma en razonar en cada percepción para tomar la acción adecuada, usando el AG (La tabla completa se encuentra en el Anexo A).

Percepción	Número de Repeticiones	Latencia (Segundos)
Detecta Fuego	0	2,71
Detecta Riesgo Eléctrico	0	2,85
Detecta Piso Húmedo	0	0,85
Puesto de Trabajo	1	9,9

Tabla 4-3: Latencia basada en repeticiones para el AG

Agentes Inteligentes

Esta es la técnica con más complejidad en sus cálculos. Es tan alto que, para poder adaptarla al entorno virtual, se hizo necesario limitar el número de iteraciones y valores iniciales de cálculo, en este caso, el número de hormigas utilizadas para encontrar al punto óptimo al momento de entregar el resultado.

Esta técnica, con valores muy bajos de iteraciones y de número de hormigas, logró hacer que se disminuyeran los cuadros por segundo hasta 15. Esto la convierte en la técnica más ineficiente en consumo, tras demostrar que puede hacer que las operaciones del resto de los sistemas de la herramienta se detuvieran hasta que terminase la técnica. El tiempo que tomó en promedio esta técnica para los cálculos fue de 0,06 segundos (de acuerdo a lo explicado anteriormente de alcanzar los 60 cuadros por segundo o 0,016).

Por otra parte, en la Tabla 4-4, la cual ese muestra un resumen del tiempo que toma en razonar en cada percepción, para tomar la acción adecuada, usando Agentes (La tabla completa se encuentra en el Anexo A).

Percepción	Número de Repeticiones	Latencia (Segundos)
Detecta Fuego	1	21,45
Detecta Riesgo Eléctrico	3	38,4
Detecta Piso Húmedo	1	9,13
Puesto de Trabajo	0	0,75

Tabla 4-4: Latencia basada en repeticiones para los Agentes

4.1.8 Completitud

La completitud es la capacidad de generar un cambio o no en el entorno ante un evento. Esta métrica puede tomar dos valores y debe tenerse en cuenta un tiempo máximo para la realización del cambio: 0, cuando no hay cambio ante un evento y 1, cuando hay cambio ante un evento.

Todas las técnicas logran hacer de manera efectiva, el razonamiento adecuado para cada percepción, por lo que tendrían una Completitud de 1. Algunas de las técnicas logran desarrollar las acciones de manera inmediata sin equivocarse, pero la RNA logra un aprendizaje en tiempo real que inclusive puede repetir el circuito inmediatamente, después de que lo ha solucionado una vez, logrando así equivocarse cada vez menos hasta que logra un total aprendizaje.

4.1.9 Escalabilidad

La escalabilidad determina como varía el desempeño del EVI cuando el número de tareas a realizar aumenta. Su medición se basa en determinar en qué proporción se altera el tiempo de latencia. Para su medición, se define la (Ecuación 4.1).

$$E_s = \frac{l_{\max} - l_{\min}}{n_{total}} \quad (\text{Ecuación 4.1})$$

donde, l_{\max} y l_{\min} son los valores máximo y mínimo de latencia, respectivamente y n_{total} es el número total de tareas. La escalabilidad viene asociada con uno de los siguientes valores cualitativos:

- Escalabilidad buena: cuando se cumple la (Ecuación 4.2).

$$\Delta E_s = \left(\frac{E_s \text{ actual} - E_s \text{ anterior}}{E_s \text{ anterior}} \right) < 0.5(50\%) \quad (\text{Ecuación 4.2})$$

- Escalabilidad Deficiente: cuando se cumple la Ecuación 4.3.

$$\Delta E_s = \left(\frac{E_s \text{ actual} - E_s \text{ anterior}}{E_s \text{ anterior}} \right) > 0.5(50\%) \quad (\text{Ecuación 4.3})$$

donde E_s actual es la Escalabilidad al aumentar el número de tareas y E_s anterior es la Escalabilidad sin aumentarlas.

En este caso, las tres técnicas presentaron una buena escalabilidad y están dadas es minutos/tarea. La Tabla 4-5, Tabla 4-6 y Tabla 4-7, muestran los resultados de la RNA, los AG y los Agentes, respectivamente.

$E_s \text{ actual}$	$E_s \text{ anterior}$	E_s	Valor Cualitativo
1,23	1,19	0,03	Escalabilidad Buena

Tabla 4-5: Escalabilidad en la RNA

$E_s actual$	$E_s anterior$	E_s	Valor Cualitativo
0,40	0,55	-0,27	Escalabilidad Buena

Tabla 4-6: Escalabilidad en el AG

$E_s actual$	$E_s anterior$	E_s	Valor Cualitativo
1,50	1,07	0,4	Escalabilidad Buena

Tabla 4-7: Escalabilidad en Agentes

4.2 Conclusiones

Haciendo una comparación entre las tres técnicas, teniendo como objetivo el demostrar cual técnica soluciona mejor el problema del guión, se puede decir que tanto el algoritmo genético como las estrategias evolutivas, lo solucionan más eficientemente.

Sin embargo, si se quiere demostrar como un computador, desde un entorno virtual, aprende, la red neuronal muestra de una manera más detallada, cómo desde una concepción inicial, en donde se tiene un personaje que debe realizar unas tareas que nunca ha hecho, debe repetir una cierta cantidad de veces el mismo ejercicio para lograr un verdadero aprendizaje, que es lo análogo con la realidad humana, en donde se deben repetir a ensayo/error, tareas para llegar a dominar el caso dado.

5. Validación del Modelo de Entorno Virtual Inteligente Mediante un Prototipo de Representación Visual

En este capítulo, se detalla otro prototipo de Entorno Virtual, para la validación del modelo, sólo con la RNA, por ser la técnica de IA seleccionada por todas sus bondades demostradas en el capítulo anterior.

Dado que se habían realizado dos prototipos de pruebas, el de problemas visuales realizado con OpenGL (orientado a la medicina) y el ambiente laboral en UDK (orientado a señales de seguridad), se decidió hacer este último prototipo de validación tanto en UDK como en Unity, por ser los dos motores de videojuegos más utilizados en la actualidad y esta vez, orientado a la educación.

Actualmente el sistema educativo, además de mantener su rol histórico de enseñanza de contenidos por áreas, se ha interesado en encontrar técnicas que le permitan fragmentar la información en pequeños tópicos que faciliten al estudiante una mejor comprensión de cada uno de éstos. En el contexto de los desarrollos tecnológicos aplicados a la educación, los Entornos Virtuales entran a jugar un papel significativo dentro de los recursos informáticos como una tecnología que permite al usuario visualizar la información en tiempo real, proponiendo la facilidad de interactuar con los contenidos de una manera dinámica.

Dado lo anterior, en este capítulo se plantea el uso de un Entorno Virtual Inteligente como un recurso orientado al aprendizaje, cuyo propósito es tratar de desarrollar aptitudes y mejorar la actitud de los estudiantes, al generar competencias entre un personaje virtual y un humano, al competir en desarrollar problemas de índole matemático, con el fin de generar un entorno más ameno para el aprendizaje de las mismas.

5.1 Entorno Virtual Inteligente como Recurso Orientado al Aprendizaje

Los Objetos de Aprendizaje son contenidos educativos virtuales que tienen como fin facilitar una educación flexible y personalizada, permitiendo que los estudiantes y docentes

puedan adaptar los recursos didácticos de acuerdo con sus propias necesidades, inquietudes, estilos de aprendizaje y enseñanza (Roanigo & Bramati, 2011). En esta sección, se habla sólo de un recurso orientado al aprendizaje, que es la fase inicial para que se pueda convertir en un Objeto de Aprendizaje, debido a que no contiene aún los metadatos descriptivos que permiten su reutilización, búsqueda y selección.

Entre las diferentes metodologías que existen para el diseño de recursos orientados al aprendizaje, se revisaron ISDMeLO (Brito, 2010), MACOBA (Margain, Muñoz, & Álvarez, 2010), MIDOA (Barajas, Muñoz, Álvarez, & Alma, 2007), SCORM (Rustici), ADDIE (Branch, 2010) y SAM (Allen, 2012), optando por esta última por ser una de las más actualizadas y basados también en la opinión de un experto del área.

SAM (Successive Approximation Model) es un modelo de desarrollo ágil que garantiza el aprendizaje, la retención y el impacto en el desarrollo del diseño instruccional; se desarrolla en pequeños procesos repetitivos, con el fin de llegar con éxito al producto final. Existen 2 tipos de SAM: SAM1 es para proyectos pequeños, para personas que no tengan gran experiencia en los Objetos de Aprendizaje y para personas que trabajan solas o en conjunto. El modelo está claramente definido, es manejable, estimula la creatividad y la experimentación y revela constantemente el diseño a medida que evoluciona (Figura 5.1). SAM2 es una versión más detallada y ampliada de SAM1 para situaciones en las que el desarrollo no se puede integrar completamente con el diseño; se utiliza cuando los proyectos son más grandes y tiene mayor cantidad de contenido y cuando primero se trabaja toda la fase de diseño y después la fase de desarrollo (Allen, 2012). Con base en todo lo anterior, se selecciona a SAM1.



Figura 5.1. Metodología SAM1 (Allen, 2012)

Con base en la Figura 5.1 de la metodología SAM1, el EVI de prueba con la RNA cumple con las siguientes etapas:

- En la etapa de evaluación, se hace un análisis de los objetivos propuestos y de la herramienta que se va a desarrollar: los objetivos giran en torno a resolver unos límites

y la herramienta que se desarrolló fue basada en un jugador virtual (que actuará con la RNA) y un humano.

- En la etapa de diseño se hacen unos bosquejos tratando de visualizar como quedará la aplicación, como se muestra en la Figura 5.2 y teniendo en cuenta el modelo geométrico y cinemático propuesto en el Capítulo 2 y el modelo de EVI del Capítulo 3.

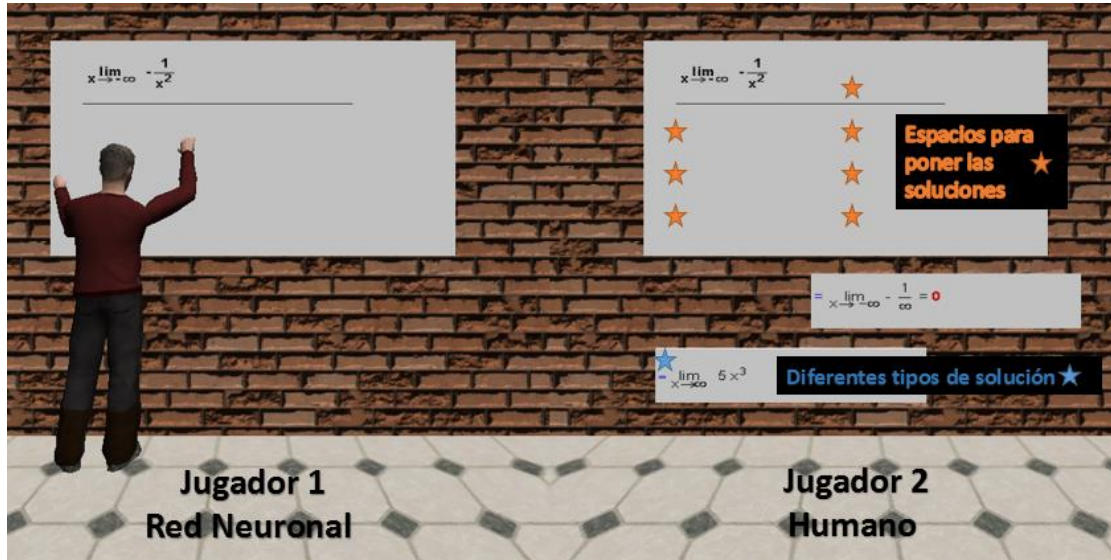


Figura 5.2. Diseño del EVI basado en la metodología SAM1

- En la etapa de desarrollo, se trabaja todo el prototipo de representación visual: El EVI fue desarrollado utilizando un PC con procesador Intel Core 2 Quad 6600 2.40 GHz, memoria RAM 8 GB Corsair Vengeance 1600 MHz y GPU EVGA GTX 480 1.5 GB, bajo el sistema operativo Windows 8.1. Las implementaciones del EVI fueron desarrolladas tanto en Unity como en UDK.

La mecánica del EVI como juego será la siguiente: Se muestran 2 tableros, como se observa en la Figura 5.2; cada tablero corresponde a un jugador: el jugador número 1, al lado izquierdo de la pantalla, es el personaje representando la RNA y el jugador número 2, al lado derecho de la pantalla, es el jugador humano. Ambos tienen como objetivo solucionar un ejercicio de cálculo (el mismo para ambos jugadores) propuesto en un tablero. El personaje representando la RNA realiza operaciones usando sus técnicas internas. Cada vez que va a realizar una operación exitosa, dibuja el resultado en el tablero. En caso de equivocarse, la dibuja igualmente, pero luego de hacerlo, la borra al percibir su error.

El humano tiene como interacción el tomar piezas expuestas alrededor del tablero para ponerlas de manera secuencial, ilustrando la solución. El jugador humano puede, en caso de no estar seguro de la solución dada, quitar las piezas para reorganizar la operación.

Al final, quien termine la solución primero, gana. Las mecánicas para la aplicación son las siguientes:

- Haciendo uso del mouse, se puede navegar a través de la interfaz del Entorno con un cursor.
- Usando la navegación, se puede realizar una interacción de tomar o sujetar piezas con una solución matemática correspondiente al flujo de desarrollo mostrado en un tablero, usando el click de manera sostenida izquierdo del mouse, como se muestra en la Figura 5.3.

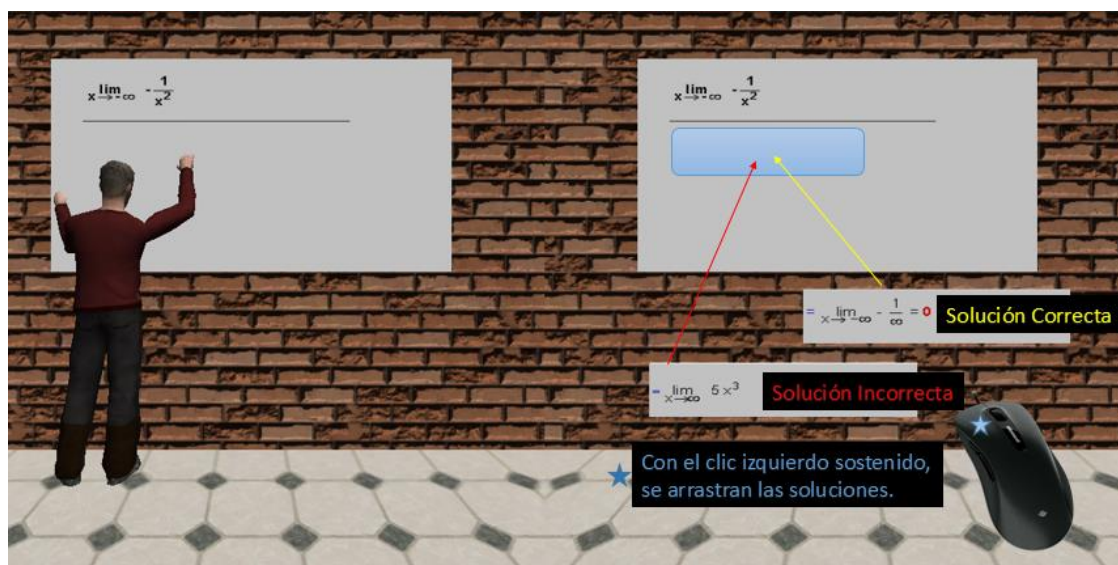


Figura 5.3. Arrastre de las soluciones al tablero

- Una vez agarrada la pieza, puede desplazarla en el tablero para ubicarla en el siguiente espacio de solución. Soltando el click izquierdo libera la pieza. En caso de no ubicarla en un espacio de solución, la pieza volverá automáticamente al lugar original. El usuario tendrá la posibilidad de volver a jugar el mismo reto, o escoger uno más avanzado, como se muestra en la Figura 5.4 , que para este caso, ganó la RNA.

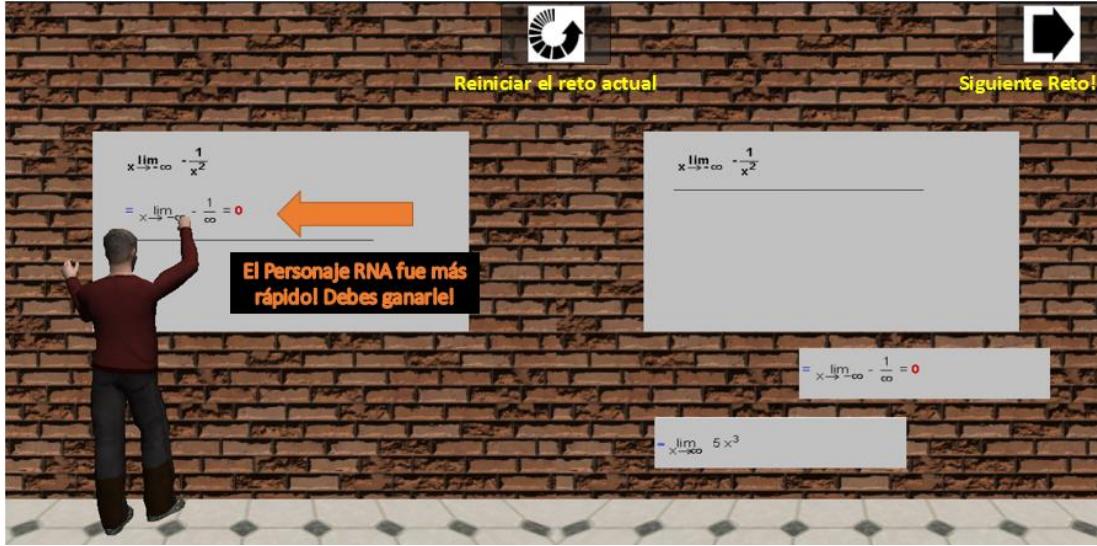


Figura 5.4. Fin del Juego

Finalmente, el modelo propuesto fue comparado con algunos modelos reportados en la literatura que tienen Entornos Virtuales Inteligentes, como se muestra en la Tabla 5-1.

Modelo	Modelo Geométrico y Cinemático	Técnica de IA implementada			Motor de Videojuego	Dim.
		RNA	Computación Evolutiva	Agentes		
Mateus, 2014	Si	Si	Si	Si	Si	3-D
(Xi & Smith, 2014)	No	No	No	Si	Si	3-D
Whiting <i>et al.</i> , (2010)	No	Si	No	Si	No	3-D
(Mang-Xian & Hai-ming, 2008)	No	No	No	Si	No	2-D
(Jia & Zhenjiang, 2007)	No	Si	Si	Si	No	2-D
Martinez <i>et al.</i> , (2006)	No	Si	No	Si	No	2-D

Tabla 5-1. Comparación del Modelo Propuesto

De la anterior tabla, cabe destacar, que en todos los trabajos sólo se trabajó un Entorno Virtual Inteligente, a diferencia de lo mostrado en esta tesis, en la cual se hicieron pruebas en tres entornos en diferentes áreas, todos en 3-D y en diferentes motores de videojuegos.

5.2 Conclusiones

Los Entornos Virtuales Inteligentes aplicados como Objetos de Aprendizaje, genera un aporte importante a la educación, debido a que se puede mostrar de una forma dinámica los

tópicos de algunas asignaturas y esto puede ser aplicable en cualquier contexto. También puede ser utilizado en el campo profesional como una herramienta de ayuda para el docente, o como una herramienta de estudio divertida y dinámica para el estudiante.

A partir del modelo de EVI, presentado en esta tesis, fue posible generar realismo a la hora de interactuar con los elementos de un Entorno Virtual, incluyendo un personaje, independiente al campo de aplicación. El componente de inteligencia fue proporcionado en la percepción y el razonamiento del mismo, de acuerdo a la técnica de Inteligencia Artificial que proporcionó los resultados más adecuados entre Agentes Inteligentes, Redes Neuronales y Computación Evolutiva, que para los entornos probados en esta tesis, fue la Red Neuronal. Una vez implementada la técnica en el modelo del EVI, ésta permitió tener un dinamismo entre el personaje y sus elementos.

6. Conclusiones y Trabajos Futuros

Se ha propuesto un modelo de Entorno Virtual Inteligente definido en la percepción y el razonamiento del mismo, como características de un comportamiento dinámico e interactivo, con el fin de generar realismo, necesario en las aplicaciones de Realidad Virtual. Dentro de éste modelo, se propuso también un modelo de los niveles geométrico y cinemático para desarrollar un Entorno Virtual 3-D, flexible al entorno deseado e independiente del área de aplicación, para crear entornos virtuales en menos tiempo, e incrementar el desempeño del entorno virtual en los niveles comportamental y cognitivo.

En términos de desarrollo de software, esta tesis propone y evidencia el uso de motores de videojuegos, con el fin de obtener un Entorno 3-D con una técnica de Inteligencia Artificial, que lo hace más funcional y más realista. En videojuegos, normalmente se desarrollan comportamiento prefabricados, pero muy pocos con un verdadero aprendizaje estratégico y algunos otros sí, pero a muy bajo nivel, debido a la gran cantidad de cálculo requerido para un óptimo aprendizaje. Con estos experimentos y los resultados obtenidos, se puede ver por ejemplo, que es posible que en un Entorno Virtual Inteligente, un personaje aprenda comportamientos.

Las técnicas de IA cuentan con operaciones complejas que requieren repeticiones exhaustivas, que a su vez afectan el aprendizaje en tiempo real. Es posible lograr aprendizaje real de máquina, pero en muchos casos, sacrificando desempeño de ésta misma. La técnica que se seleccione puede ser eficiente, pero desde la perspectiva de un objetivo dado, algunas técnicas pueden tener mejor desempeño que otras.

Analizando la concepción de dichas técnicas, en donde cada una toma como referencia a la ciencia de la naturaleza, como los Algoritmos Genéticos y las Estrategias Evolutivas, que se basan en la evolución de los seres vivos para que en un futuro resulten mejores especies, y que las Redes Neuronales Artificiales se basan en las neuronas del cerebro y de sus métodos para el aprendizaje humano, se puede concluir desde su misma concepción, la velocidad de desempeño. Es decir, si se toma la evolución de una especie animal, sus ancestros y las futuras especies, esto es algo que a la naturaleza le toma mucho tiempo, evolución que tarda en desarrollarse por generaciones. Ahora, si se observa cómo se comportan las neuronas, se puede notar que éstas están en un sistema controlado por

pulsos eléctricos que estimulan el entorno a velocidad increíblemente rápidas, pero que al mismo tiempo, es un sistema relativamente simple; estas neuronas se organizan de tal manera, que a través de pulsos eléctricos, se puede acceder a cualquier tipo de información almacenada en la memoria, hasta lograr una adecuación optimizada.

De las técnicas implementadas en esta tesis: la Red Neuronal desde su concepto se entiende como una técnica con el objetivo de realizar aprendizaje de máquina y se puede ver que dentro de los resultados, hay evidencia de repeticiones que indican un aprendizaje dado parámetros tanto fijos como aleatorios. Para el caso de el Algoritmo Genético, se evidencia una alta optimización en la resolución de un problema que, aunque sea generando datos aleatorios, puede encontrar una solución bastante rápida, reduciendo el número de iteraciones; pocos casos en donde se repite la operación, y un caso muy aislado en donde se ve una alta repetición en un mismo caso, pero que deja de repetirse en los siguientes. Las Estrategias Evolutivas comparten la misma tendencia de los Algoritmos Genéticos, alta eficiencia para resolver un problema, dados unos parámetros de entrada como el número de población, número de genes, entre otros. En el caso de los Agentes, la técnica cuenta con un sistema de cálculos que si bien son simples, entran en complejidad al momento de realizar operaciones en una herramienta como los motores de videojuegos; al tener que realizar tantas iteraciones dados los parámetros de número máximo de hormigas y de iteraciones, esto puede saturar los límites establecidos por cada herramienta.

Las limitaciones de software de cada herramienta, hacen parte del resultado final de la operación. En el caso específico de UDK, existen dos grandes limitaciones: UnrealScript, es un lenguaje de programación con el que se programa en la herramienta, cuenta con variables de tipo float, pero estas a su vez son de tipo precisión sencilla. Esto quiere decir, que los dígitos decimales de estos números están limitados a 6 dígitos. Al usar cálculos probabilísticos con más dígitos por ser valores tan bajos, significa que desde el sexto dígito en adelante van a ser obviados. Esto se presta para errores y resultados no deseados. Otro limitante, es la cantidad de operaciones permitidas por cada cuadro por segundo. No solo en una aplicación, sino en toda aquella en donde se realicen una cantidad alta de iteraciones, puede generar errores, haciendo que las operaciones no logren todo su propósito.

6.1 Trabajo Futuro

Como trabajo futuro se plantea, la combinación de diferentes técnicas de IA, probar la neuroevolución o RNA incorporadas en los agentes, por ejemplo, para tratar de obtener un Entorno Virtual mucho más inteligente, sin descuidar el tiempo real necesario para este tipo de aplicaciones de Realidad Virtual.

Dado que los Entornos se hicieron bajo computadores de uso cotidiano, como un computador de escritorio y portátiles (una única CPU y velocidad de transferencia de datos limitado), se recomienda probar estos modelos en computadores conectados en paralelo para lograr un mejor desempeño.

Referencias

- Allen, M. (2012). *Leaving ADDIE for SAM. An Agile Model for Developing the Best Learning Experiences*. ASTD Press.
- Axling, T., Haridi, S., & Fahlen, L. (1996). Virtual reality programming in Oz. *Proceeding of the 3rd Eurographics workshop on virtual environments*. Montecarlo.
- Aylett, R. (1999). Virtual teletubbies: reapplying robot architecture to virtual agents. *Proceeding, 3rd International Conference Autonomous Agents* (págs. 339-349). ACM Press.
- Aylett, R., & Cavazza, M. (2001). *Intelligent Virtual Environments - A state of the art*. Eurographics - STARs.
- Aylett, R., & Luck, M. (2000). Applyin artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence*.
- Bandi, S., & Thalmann, D. (1998). Space Discretization for Efficient Human Navigation. *Computer Graphics Forum*. 17 (3), 195-206.
- Barajas, A., Muñoz, J., Álvarez, F., & Alma, G. (2007). Developing Large Scale Learning Objects for Software Engineering Process Model through MIDOA Model., (pág. 6).
- Bee, N. (2010). Interacting with a Gaze-Aware Virtual Character. *International Workshop on Eye Gaze in Intelligent Human Machine Interaction*.

- Blumberg, B., & Galyean, T. (1997). Multi-level control for animated autonomous agents: Do the right thing...oh no, not that. *Creating personalities for synthetic actors*. Springer-Verlag, 74-82.
- Bonis, B. (2009). A platform for virtual museums with personalized content. *Springer Science +Business Media*.
- Bordini, R., Hübner, J., & Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*. Ed. Wiley.
- Branch, R. (2010). *Instructional Design: The ADDIE Approach*. Springer Science+Business Media, LLC.
- Brito, J. (2010). *Metodologías de Desarrollo de Objetos de Aprendizaje*. Obtenido de Portal OpenCourseWare de la Universidad Nacional de Córdoba, Argentina.: <http://www.ocw.unc.edu.ar/proed/objetos-de-aprendizaje-y-educacion-bfpromesas-o/actividades-y-materiales/modulo-3>
- Brownlee, J. (2012). *Clever Algorithms - Nature Inspired Programming Recipes*. Australia: Creative Commons. ISBN: 978-1-4467-8506-5.
- Buche, C., & Querrec, R. (2011). An expert system manipulating knowledge to help human learners into virtual environment. *Expert Systems with Applications*. Volume 38, Issue 7, 8446–8457.
- Calderon, C., & Cavazza, M. (2001). Intelligent Virtual Environments for Spatial Configuration Tasks. *Proceedings of Virtual Reality International Conference*. Laval, France.
- Cavazza, M., Lugin, J.-L., Hartley, S., Renard, M., Nandi, A., Jacobson, J., & Crooks, S. (2005). Intelligent virtual environments for virtual reality art. *Elsevier - Computer & Graphics*. Volume 29, Issue 6, 852-861.
- Clemente, J., Ramirez, J., & De Antonio, A. (2014). Applying a student modeling with non-monotonic diagnosis to Intelligent Virtual Environment for Training/Instruction. *Expert Systems with Applications - Elsevier*. Volume 41, Issue 2, 508–520.
- Codognet, P. (1999). Animating autonomous agents. *IEEE International Conference on Distributed Multimedia Systems*. Aizu, Japan: IEEE Press.
- Codognet, P. (2001). Behaviours for virtual creatures by constraint-based adaptive search. *Symposium on Artificial Intelligence and Interactive Entertainment* (págs. 25-30). Stanford, USA: Working Notes of the AAAI Spring.

- Decker, Sichman, Sierra, & Castelfranchi. (2009). Emotional Input for Character-based Interactive Storytelling. *Proc. of 8th International Conference on Autonomous Agents and Multi-agents Systems*.
- DosSantos, C., & Osorio, F. (2004). An intelligent and adaptive virtual environment and its application in distance learning. *AVI '04*. Gallipoli, Italy.
- Funge, J., Tu, X., & Terzopoulos, D. (1999). Cognitive modeling: knowledge, reasoning and planning for intelligent characters. *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, (págs. 29-38).
- García, J., Carmona, E., Gallardo, L., González, M., Fernández, A., & González, M. (2002). Desarrollo de un sistema automático de discriminación del campo visual glaucomatoso basado en un clasificador neuro-fuzzy. En *Revista Scielo*. 77 (12) (págs. 669-676). Archivos de la Sociedad Española de Oftalmología.
- Gilbert, R., & Forney, A. (2015). Can avatars pass the Turing test? Intelligent agent perception in a 3D virtual environment. *International Journal of Human-Computer Studies*. Volume 73, 30-36.
- Grand, S., & Cliff, D. (1998). Creatures: Entertainment software agents with artificial life. *Autonomous agents and multi-agent systems*. Vol 1 N 1, 39-57.
- Jia, L., & Zhenjiang, M. (2007). Entertainment Oriented Intelligent Virtual Environment with Agent and Neural Networks. *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*. Ottawa, Canada.
- Kaigom, E., & Roßmann, J. (2014). Developing Virtual Testbeds for Intelligent Robot Manipulators - An eRobotics Approach. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, 1589-1594.
- Kurt, A., Vernier, M., Biddlestone, S., Redmill, K., & Özgüner, Ü. (2014). Chapter 2 – Testing of Intelligent Vehicles Using Virtual Environments and Staged Scenarios. En *Advances in Intelligent Vehicles* (págs. 45-64). Academic Press.
- Latoschik, M., & Fröhlich, C. (2007). Semantic Reflection for Intelligent Virtual Environments. Charlotte, North Carolina, USA.
- Liu, W., Zhou, L., Xing, W., Liu, X., & Yuan, B. (2011). Creating Autonomous, Perceptive and Intelligent Virtual Humans in a Real-Time Virtual Environment. *Tsinghua Science & Technology*. Volume 16, Issue 3, 233–240.
- Loyall, A., & Bates, J. (1993). Real-time control of animated broad agents. *Proc. 15th Conference of the Cognitive Science Society*, (págs. 664-669).

- Lozano, M. (2003). An efficient synthetic vision system for 3D multi-character systems. *Intelligent Agents - 4th International Workshop IVA*. Lecture Notes in Computer Science.
- Lozano, M. (2004). *Animación Comportamental de Personajes Inteligentes 3D basada en MINIMIN-HSP*. Valencia: Tesis Doctoral. Universitat de València.
- Lozano, M., & Calderón, C. (2004). Entornos Virtuales 3D clásicos e inteligentes: hacia un nuevo marco de simulación para aplicaciones gráficas 3D interactivas. *Journal en Inteligencia Artificial*, Vol 8 N 23.
- Luzardo, G., & Hernández, J. (2010). *Inteligencia Artificial en Ambientes Virtuales: Humanos Virtuales Autónomos (HVA) como Agentes Virtuales Inteligentes (3DIVA)*. Obtenido de <http://blog.espol.edu.ec/gluzardo/files/2009/03/trabajo-final.pdf>
- Mang-Xian, Q., & Hai-ming, Y. (2008). Investigation and Realization of Multi-Agent interaction behavior in Intelligent Virtual Environment. *International Conference on Cyberworlds*. IEEE Computer Society.
- Margain, M., Muñoz, J., & Álvarez, F. (2010). *Metodología de Aprendizaje Colaborativo fundamentada en patrones para la producción y uso de Objetos de Aprendizaje*. Obtenido de Red de Revistas Científicas de América Latina, El Caribe, España y Portugal: <http://www.redalyc.org/pdf/674/67411476005.pdf>
- Martinez, M., Gayoso, C., & Suarez, S. (2006). Intelligent virtual environments: Operating conditioning and observational learning in agents using neural networks. *Intelligent Environments, 2006. IE 06. 2nd IET International Conference on (Volume:1)* (págs. 127-133). IEEE Explore.
- Martinho, C., Paiva, A., & Gomes, M. (2000). Emotions for a Motion: Rapid Development of believable Panthematic Agents in Intelligent Virtual Environments. *Applied Artificial Intelligence*, 14-33.
- Mascarenhas, S., Enz, S., & Paiva, A. (2009). Using rituals to express cultural differences in synthetic characters. *Autonomous Agents and Multiagent Systems*, (págs. 305-312). Budapest, Hungary.
- Mateus, S., & Branch, J. (2012). Modelo de un Personaje en un Entorno Virtual Inteligente. En *Revista Información Tecnológica*. 23 (3) (págs. 103-112). La Serena, Chile: ISSN: 0718-0764.
- Padgham, L., & Winikoff, M. (2005). Prometheus: a practical agent-oriented methodology. Ed. TermLing.
- Pérez, A. (1996). *Una Introducción a la Computación Evolutiva*.

- Prophet, J. (1996). *Technosphere*. Interpretation.
- Rahman, A., & Kamil, A. (2008). Agent-Based Simulation using Prometheus Methodology in Evacuation Planning. *Information Technology*, 8, págs. 1-8. ISBN: 978-1-4244-2328-6.
- Restrepo, S. (2012). *Modelo de Inteligencia Ambiental basado en la integración de Redes de Sensores Inalámbricas y Agentes Inteligentes*. Tesis de Maestría. Universidad Nacional de Colombia - Sede Medellín.
- Rickel, J., & Johnson, W. (2000). Taskoriented Collaboration with Embodied Agents in Virtual Worlds. *Embodied Conversational Agents*. MIT Press.
- Rozanigo, Z., & Bramati, P. (2011). *Objetos de Aprendizaje*. Obtenido de Repositorio Institucional de la UNLP: <http://sedici.unlp.edu.ar/handle/10915/19934>
- Rustici, S. (s.f.). *SCORM*. Obtenido de <http://scorm.com/scorm-explained/>
- Sims, K. (1995). Evolving 3D morphology and behaviour by competition. *Artificial Life*, (págs. 353-372).
- Thalmann, M., & Thalmann, D. (1998). The virtual humans story. *IEEE Annals of the History of Computing*. 20(2), 50-55.
- Tutenel, T., & Bidarra, R. (2008). The Role of Semantics in Games and Simulations. *ACM Computers in Entertainment*. Vol 6 N 4, Article 57.
- Vic, M., & Alghazzawi, D. (2014). Towards MMO Intelligent Environments. 2014 *International Conference on Intelligent Environments*. IEEE Computer Society.
- Whiting, J., Dinerstein, J., Egbert, P., & Ventura, D. (2010). Cognitive and behavioral model ensembles for autonomous virtual characters. *Computational Intelligence*. 26(2)., 142-159.
- Xi, M., & Smith, S. (2014). Simulating Cooperative Fire Evacuation Training in a Virtual Environment Using Gaming Technology. *IEEE Virtual Reality 2014* (págs. 139-140). Minneapolis, Minnesota, USA: ©2014 IEEE.
- Zhang, D. (2010). *Multi-agent based control of large-scale complex systems employing distributed dynamic inference engine*. PhD thesis. Georgia Institute of Technology.

A. Muestra de cada una de las técnicas para la evaluación de la latencia

Para las variables de las tablas, los números indican el número de veces que tuvo que recalcular para poder solucionar los criterios de las técnicas. Cuando hay un 0, significa que el cálculo que hizo inicialmente fue efectivo y no tuvo que repetir. El tiempo está en minutos:segundos:milisegundos y representa lo que se demoró el personaje en solucionar todo el circuito. Por ejemplo, si tomamos la muestra 9 de la Tabla 0-1 que corresponde a la RNA, el personaje realiza una iteración en caso de detectar fuego, es decir, una vez identifica que hay presencia de éste, no supo qué hacer y usa el citófono para pedir ayuda, una vez termina la consulta, ya sabe qué debe hacer y camina por el entorno; cuando detecta el riesgo eléctrico, de una vez informa al encargado de las redes eléctricas y apaga los interruptores de luz; de igual forma cuando detecta el piso húmedo, de una vez notifica al personal de limpieza y finalmente, pasa por la percepción de puesto de trabajo y no sabe qué hacer en dos ocasiones. La Tabla 0-2 corresponde a las muestras tomadas del AG y la Tabla 0-3 a las muestras tomadas de los Agentes Inteligentes

Muestra	Repeticiones & Output - (1,1,1,-1)				Tiempo
	Detecta Fuego	Detecta Riesgo Eléctrico	Detecta Piso Húmedo	Puesto Trabajo	
1	2 (-1, -1)	0 (1)	0 (1)	0 (-1)	1:19:45
2	1 (-1)	0 (1)	1 (-1)	1 (1)	1:23:89
3	1 (-1)	0 (1)	1 (-1)	0 (-1)	1:09:60
4	1 (-1)	0 (1)	0 (1)	0 (-1)	1:03:84
5	1 (-1)	1 (-1)	1 (-1)	0 (-1)	1:20:16
6	0 (1)	0 (1)	0 (1)	0 (-1)	0:40:96
7	0 (1)	0 (1)	0 (1)	1 (1)	0:54:16
8	1 (-1)	0 (1)	0 (1)	0 (-1)	1:03:94
9	1 (-1)	0 (1)	0 (1)	2 (1, 1)	1:30:28
10	0 (1)	1 (-1)	2 (-1, -1)	0	1:09:47

Tabla 0-1: Evaluación de la Latencia en la RNA

La Tabla 0-2 corresponde a las muestras tomadas del AG y la Tabla 0-3 a las muestras tomadas de los Agentes Inteligentes.

Muestra	REPETICIONES / OUTPUT - (X, X, X, X) X = (>Rand(962))				Tiempo
	Detecta Fuego	Detecta Riesgo Eléctrico	Detecta Piso Húmedo	Puesto Trabajo	
1	0 (676.0, 233.0)	0 (625.0, 67.0)	0 (900.0, 757.0)	1 (625.0, 746.0)	0:55:89
2	0 (400.0, 205.0)	0 (900.0, 180.0)	0 (900.0, 631.0)	0 (729.0, 558.0)	0:40:48
3	0 (841.0, 37.0)	0 (676.0, 2.0)	0 (900.0, 533.0)	0 (742, 53)	0:40:07

4	0 (961.0, 363.0)	0 (900.0, 473.0)	0 (784.0, 704.0)	0 (961.0, 681.0)	0:41:12
5	0 (961.0, 533.0)	0 (841.0, 280.0)	0 (841.0, 258.0)	0 (784.0, 511.0)	0:40:28
6	0 (961.0, 654.0)	0 (441.0, 256.0)	2 (121.0, 699.0), (441.0, 699.0), (961.0, 699.0)	0 (961.0, 188.0)	0:56:13
7	0 (841.0, 732.0)	1 (484.0, 652.0), (729.0, 652.0)	2 (576.0, 739.0), (729.0, 739.0), (841.0, 739.0)	0 (676.0, 489.0)	1:08:63
8	0 (900.0, 98.0)	0 (625.0, 452.0)	1 (529.0, 578.0), (676.0, 578.0)	0 (784.0, 422.0)	0:48:27
9	0 (961.0, 661.0)	0 (529.0, 33.0)	0 (676.0, 17.0)	0 (225.0, 128.0)	0:40:10
10	0 (900.0, 772.0)	0 (961.0, 469.0)	0 (784.0, 328.0)	0 (441.0, 844.0), (256.0, 844.0), (900.0, 844.0)	1:08:54

Tabla 0-2: Evaluación de la Latencia en el AG

Muestra	Repeticiones & Output - (1, 1, 1, -1)				Tiempo
	Detecta Fuego	Detecta Riesgo Eléctrico	Detecta Piso Húmedo	Puesto Trabajo	
1	0 (2340, 8615)	2 (14924, 8964), (25012, 14924), (21944, 25012)	0 (5720, 8656)	0 (5408, 8723)	1:07:99
2	1 (24492, 8766), (15288, 24492)	3 (28652, 8493), (29276, 28652), (33748, 29276), (7644, 33748)	1 (31408, 8537), (31408, 31408)	0 (3640, 8430)	1:50:08
3	0 (1820, 8472)	1 (33748, 8685), (33748, 33748)	2 (18408, 8104), (31408 rand 18408), (16848, 31408)	0 (5408, 8579)	1:14:58
4	0 (780, 8044)	1 (18148, 8008), (12688, 18148)	1 (31408, 8072), (2912, 31408)	0 (5408, 8704)	1:05:66
5	1 (20384, 8501), 9672 rand 20384)	2 (21892, 8047), (33748, 21892), (26520, 33748)	1 (16536, 8208), (31408, 16536), (16380, 31408)	0 (5408, 8102)	1:47:72
6	1 (25584, 8410), (5720, 25584)	1 (33748, 8129), (26520, 33748)	1 (31408, 8279), (31408, 31408)	0 (5408, 8199)	1:23:93
7	2 (26520, 8277), (46644, 26520), (2964, 46644)	0 (7852, 8441)	1 (31408, 8051), (5408, 31408)	0 (3640, 8402)	1:32:13

8	1 (26208, 8156), (5720, 26208)	1 (33748, 8424), (33748, 33748)	1 (31408, 8702), (4784, 31408)	0 (5408, 8789)	1:21:70
9	1 (44616, 8319), (2340, 44616)	1 (30836, 8336), (3172, 30836)	1 (31408, 8422), (17524, 31408)	0 (1820, 8278)	1:25:46
10	0 (2340, 8301)	2 (14092, 8596), (33748, 14092), (31408, 33748)	1 (31408, 8437), (22828, 31408)	0 (5408, 8636)	1:13:92

Tabla 0-3: Evaluación de la Latencia en los Agentes Inteligentes

$x(t_1, t_2, \dots) \rightarrow x = \text{número de repeticiones, } t_1, t_2, \dots = \text{valores entregados}$