

# Universidad de las Ciencias Informáticas

## Facultad 5



**Título:** Aplicación de las Redes Neuronales Artificiales y Mapas de Influencia en la toma de decisiones para videojuegos de estrategia.

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

**Autor(es):** Denisse Díaz González

Yadira Malcolm de Zayas

**Tutor:** Ing. Adiel Durán Rodríguez

Ciudad de La Habana, Julio de 2010  
“Año del 52 Aniversario de la Revolución”

## DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2010.

Yadira Malcolm de Zayas

Denisse Díaz González

Adiel Duran Rodríguez

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor

*"La mejor forma de predecir el futuro es implementarlo"*

*David Heinemeier Hansson*

### AGRADECIMIENTOS

*Agradezco a todas las personas que estuvieron en mi vida para apoyarme y darme su  
confianza.*

*A la Revolución Cubana por permitirme estar en una universidad de excelencia.*

*A Yuri por su dedicación.*

*A mi tutor Adiel y a todos los profesores que hicieron posible este trabajo.*

*A mi familia por darme aliento para seguir adelante*

*y a mi compañera de tesis Denisse.*

*Yadira*

*Agradecer a todas las personas que de una forma u otra me han ayudado y apoyado a lo  
largo de mi carrera y mi vida.*

*Agradecer a todos mis profesores, amigos y compañeros.*

*Un agradecimiento especial a mi profesor de siempre Millet, por ayudarme tanto en mis  
primeros años de universidad.*

*A Yuri por su ayuda en el desarrollo de la tesis.*

*A mi familia por estar siempre pendiente de mí.*

*Y a Yadira por su esfuerzo.*

*Denisse*

## DEDICATORIA

*A mis padres por ser las personas que me guiaron en la vida.*

*A mi abuelita Alicia por darme su amor y confianza.*

*A mis hermanas, tíos, primas y primos por darme apoyo cuando lo necesité.*

*A toda mi familia.*

*A mis amistades de la infancia por soportarme.*

*A mis amigos de los diferentes grupos donde transcurrí mis estudios, por brindarme su amistad.*

*A mis amistades de la universidad.*

*Y a Denisse por su paciencia.*

Yadira

*A mi madre por ayudarme siempre, por confiar en mí, por su aliento, su dedicación y por brindarme su infinito amor.*

*A mi tío Papito por cuidar de mí siempre.*

*A mi tía Mercedes por su apoyo incondicional.*

*A mi familia en general por ayudarme en todo.*

Denisse

### RESUMEN

En la investigación realizada se emplean dos técnicas de importancia para un videojuego de estrategia, las Redes Neuronales Artificiales y los Mapas de Influencia. La combinación de estas técnicas le incorpora al videojuego la capacidad de imitar expertos jugadores humanos concediéndole así una vida útil y extensa, además de convertirlo en algo novedoso e interesante.

La presente investigación es de gran utilidad en el perfeccionamiento y desarrollo de videojuegos para dicha línea de productos. Aplicar la misma, propicia un mayor realismo en los videojuegos de estrategias, brinda una forma alternativa para la toma de decisiones y evita la posible pérdida de información al utilizar otras técnicas como la suma ponderada.

**Palabras claves:** Inteligencia Artificial, Redes Neuronales Artificiales, Mapas de Influencia, Videojuegos de Estrategia, Toma de Decisiones.

**Keywords:** Artificial Intelligence, Artificial Neural Networks, Influence Maps, Strategy Games, Decision Making Process.

## Tabla de Contenido

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	5
1.1 Introducción .....	5
1.2 Evolución de los videojuegos de estrategia .....	5
1.3 Subgéneros .....	5
1.3.1 Estrategia en tiempo real.....	6
1.3.2 Estrategia por turnos .....	8
1.3.3 Videojuegos de guerra.....	10
1.4 Videojuegos de Rol .....	11
1.4.1 Juegos de rol multi-jugador masivos en línea.....	13
1.5 Modelos cognitivos .....	14
1.5.1 Ejemplos de modelos cognitivos .....	15
1.6 Mapas de Influencia .....	19
1.7 Toma de decisiones en videojuegos .....	20
1.7.1 La inteligencia artificial en videojuegos de estrategia.....	21
1.8 Redes Neuronales Artificiales .....	22
1.8.1 La neurona natural y el modelo biológico .....	22
1.8.2 Características de las Redes Neuronales Artificiales .....	23
1.8.3 Elementos de una Red Neuronal Artificial .....	24
1.8.4 Redes con conexiones hacia adelante (Feedforward) .....	30
1.8.5 Redes con conexiones hacia adelante/hacia atrás (Feedforward/Feedback) .....	32

1.8.6 Aplicaciones de las Redes Neuronales Artificiales en videojuegos .....	35
1.8.7 Ventajas y desventajas de las Redes Neuronales Artificiales .....	37
1.9 Bibliotecas para el trabajo con Redes Neuronales Artificiales .....	38
1.10 Conclusiones parciales .....	39
<b>CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA.</b>	<b>40</b>
2.1 Introducción .....	40
2.2 Funcionamiento de los Mapas de Influencia .....	40
2.3 Decisiones de diseño de una Red Neuronal Artificial .....	41
2.4 Modelo de red seleccionado .....	42
2.4.1 Proceso de entrenamiento de la red .....	45
2.5 Combinación de las Redes Neuronales Artificiales y los Mapas de Influencia .....	49
2.6 Selección de las variables.....	51
2.7 Conclusiones parciales .....	54
<b>CAPITULO 3: RESULTADOS GENERALES</b> .....	<b>55</b>
3.1 Introducción.....	55
3.2 Descripción de la solución propuesta .....	55
3.2.1 Esquema general de la solución .....	55
3.3 Algoritmo propuesto .....	57
3.4 Aplicación Demostrativa.....	63
3.4.1 Funcionalidades de la aplicación .....	64
3.5 Reglas del juego.....	66
3.6 Conclusiones Parciales.....	68
<b>CONCLUSIONES</b> .....	<b>69</b>



REFERENCIAS BIBLIOGRÁFICAS .....	71
BIBLIOGRAFÍA .....	73
ANEXOS.....	74
GLOSARIO.....	75

## Índice de Figuras

FIG. 1 NAVE BATTLE CRUISE DE STARCRAFT.....	7
FIG. 2 VIDEOJUEGO BLACK & WHITE.....	7
FIG. 3 VIDEOJUEGO DE HISTORIA AGE OF EMPIRES .....	9
FIG. 4 VIDEOJUEGO DE FANTASÍA MASTER OFMAGIC.....	9
FIG. 5 VIDEOJUEGO FINAL FANTASY.....	10
FIG. 6 VIDEOJUEGO CLOSE COMBAT.....	10
FIG. 7 VIDEOJUEGO BALDUR GATE.....	13
FIG. 8 VIDEOJUEGO KINGDOM HEARTS. ....	13
FIG. 9 VIDEOJUEGO EVERQUEST .....	14
FIG. 10 VISTA DE UN MODELO COGNITIVO.....	15
FIG. 11 REPRESENTACIÓN DE UN MODELO COGNITIVO BASADO EN REJILLAS.....	15
FIG. 12 REPRESENTACIÓN DE UN TERRENO.....	16
FIG. 13 POLÍGONOS.....	17
FIG. 14 MALLA EN 3D CON ALAMBRES DE SOPORTE.....	17
FIG. 15 PUNTOS DE NAVEGACIÓN.....	18
FIG. 16 PUNTOS DE NAVEGACIÓN EN 3D.....	18
FIG. 17 MODELO DE UNA RED NEURONAL NATURAL.....	23
FIG. 18 FUNCIÓN SIGMOID.....	25
FIG. 19 FUNCIÓN ESCALÓN.....	25
FIG. 20 FUNCIÓN LINEAL.....	25
FIG. 21 ARQUITECTURA DE UNA RED PERCEPTRÓN MONOCAPA.....	31
FIG. 22 ARQUITECTURA DE UNA RED PERCEPTRÓN MULTICAPA.....	32
FIG. 23 PARALELEPÍPEDO.....	34
FIG. 24 MODELO D ERED NEURONAL UTILIZADO EN EL VIDEOJUEGO RAPIDO Y CURIOSO.....	36
FIG. 25 ATAQUE O HUIDA.....	37
FIG. 26 EJEMPLO DE UN MAPA DE INFLUENCIA .....	41
FIG. 27 ESTRUCTURA DE LA UNIDAD BASICA DEL MLP.....	44
FIG. 28 DISEÑO DE LA RED UTILIZADA .....	50
FIG. 29 ANÁLISIS DE LA PREGUNTA 1.....	52
FIG. 30 ANÁLISIS DE LA PREGUNTA 2.....	53
FIG. 31 RELACIÓN ENTRE CLASES.....	56
FIG. 32 INFLUENCIA Y PROPAGACIÓN DE LOS ENEMIGOS.....	58
FIG. 33 INFLUENCIA Y PROPAGACIÓN DE LAS FORTIFICACIONES JUGADOR.....	58
FIG. 34 INFLUENCIA Y PROPAGACIÓN DE LOS OBSTÁCULOS.....	59
FIG. 35 INFLUENCIA Y PROPAGACIÓN DE LOS RECURSOS.....	59

FIG. 36 INFLUENCIA Y PROPAGACIÓN DE LAS FORTIFICACIONES ENEMIGAS.....	60
FIG. 37 POSICIÓN INICIAL DEL JUGADOR.....	61
FIG. 38 VALORES DE LAS VARIABLES DE DECISIÓN.....	61
FIG. 39 MAPA GENERADO PARA ENTRENAR LA RED.....	66
FIG. 40 RESULTADO GENERADO POR LA RED.....	67

## INTRODUCCIÓN

Desde que el hombre existe en la tierra siempre le ha surgido la necesidad de jugar, ya sea con los componentes de la naturaleza o con sistemas creados por él. Con el desarrollo de la ciencia, los juegos se han perfeccionado, tanto que ya no es necesario que dos personas estén presentes una frente a la otra para poder interactuar. Este adelanto informático se conoce con el nombre de videojuegos, y dentro de esta rama se encuentran los llamados *videojuegos de estrategia*. Ellos requieren de la habilidad de planeamiento y pensamiento del jugador para conseguir la victoria. En la mayoría de estos videojuegos, al jugador se le concede una vista absoluta del mundo, controlando indirectamente las unidades bajo su poder.

Nuestro país a pesar de las limitaciones impuestas por el bloqueo estadounidense ha luchado por insertarse en el mercado mundial con la creación de videojuegos. Uno de los centros donde se desarrolla este tipo de producto es la Universidad de las Ciencias Informáticas (UCI), específicamente en el Área Temática de la facultad #5. Es de interés para esta área el estudio de los videojuegos de estrategia ya que forman parte de uno de los productos más comercializados en el mundo.

La presente investigación es de gran utilidad en el perfeccionamiento y desarrollo de los videojuegos para dicha línea de productos. Al combinar las Redes Neuronales Artificiales con los Mapas de Influencia, el videojuego toma un mayor realismo, al mismo tiempo que resulta como alternativa a utilizar para la toma de decisiones y evitar la posible pérdida de información al utilizar otras técnicas como la suma ponderada. Esta consiste en darle un nivel de importancia a cada uno de los elementos que intervienen en el videojuego y luego multiplicar cada una de las capas que conforman el mapa por un valor determinado, el cual permite conocer el nivel de importancia que posee el elemento en el mapa.

Por tal motivo se desea realizar una combinación entre los Mapas de Influencia y las Redes Neuronales Artificiales para propiciarle a cada elemento la importancia que requieren en determinada situación del videojuego, sin pérdida de información. Además usando esta técnica de Inteligencia Artificial el usuario y la PC pueden interactuar de forma más real.

El **Problema a resolver** queda formulado a modo de interrogante de la siguiente forma:

¿Cómo generar decisiones estratégicas en videojuegos utilizando técnicas de Inteligencia Artificial?

Para esta investigación se define como **objeto de estudio** las técnicas de inteligencia artificial para la toma de decisiones en videojuegos de estrategia.

El **campo de acción** que abarca esta investigación se define como: las Redes Neuronales Artificiales en la toma de decisiones en Mapas de Influencia.

El **objetivo general** es demostrar la aplicación de las Redes Neuronales y Mapas de Influencia en la toma de decisiones estratégicas en videojuegos.

## **Resultados esperados:**

La vinculación de las Redes Neuronales Artificiales con los Mapas de Influencia posibilitará la imitación de expertos jugadores humanos a la hora de tomar decisiones en los videojuegos de estrategia, debido a que las máquinas podrán ser capaces de dar respuestas similares a las del cerebro humano, para lograr este objetivo se pretende que las PC puedan ir aprendiendo en cada sesión del videojuego e interactuar con el jugador de forma inteligente con un mínimo de conocimiento inicial. Los videojuegos tomarán un carácter más realista con comportamientos inteligentes, lo que posibilitará la dinámica, entretenimiento, margen de dificultad y aumento de la vida útil del mismo. Además el uso de las Redes Neuronales Artificiales ayudará a prevenir la pérdida de información que existe al utilizar el método de la suma ponderada en los Mapas de Influencia a la hora de tomar una decisión estratégica.

## **Tareas Investigativas**

Para dar cumplimiento al objetivo de la investigación se trazaron las siguientes tareas principales:

- Caracterización de los videojuegos de estrategia para determinar el funcionamiento lógico de los elementos que intervienen.
- Caracterización de los modelos de Redes Neuronales Artificiales para la selección de modelo(s) a aplicar en la toma de decisiones en videojuegos.
- Descripción del funcionamiento de los Mapas de Influencia para su vinculación con una técnica de Inteligencia Artificial.

-Descripción de las variables y elementos seleccionados a tener en cuenta para la toma de decisiones estratégicas

-Descripción del modelo de red seleccionado para la toma de decisiones en videojuegos de estrategias.

Descripción del algoritmo propuesto que vincule los Mapas de Influencia y las Redes Neuronales Artificiales en videojuegos de estrategia para la toma de decisiones.

-Implementación de un Demo que demuestre el uso de las Redes Neuronales Artificiales y los Mapas de Influencia en videojuegos de estrategia para la toma de decisiones.

**Métodos Teóricos** usados en la investigación:

**Analítico-Sintético:** Se utiliza este método para hacer un análisis detallado de varios documentos que hacen referencia a los modelos de Redes Neuronales Artificiales, los Mapas de influencias y videojuegos de estrategia y así llevar a cabo tomas de decisiones estratégicamente adecuadas.

**Inductivo-Deductivo:** Partiendo de toda la información recogida sobre los modelos de Redes Neuronales Artificiales investigados, este método permite aplicar el modelo más adecuado en un Mapa de Influencia.

**Método Empírico** usado en la investigación:

**Entrevista:** Se realiza entrevistas a especialistas en el tema para dominar el funcionamiento de los modelos de Redes Neuronales que se utilizan en los videojuegos.

**Encuesta:** Se realiza una encuesta a personas expertas o no en el tema de los videojuegos de estrategias para obtener las decisiones a ser utilizadas en la investigación.

**Estructura del documento:**

El presente documento se estructura en resumen, introducción, tres capítulos de contenido, conclusiones, recomendaciones, bibliografía referenciada, bibliografía consultada, anexos y glosario de términos. El Capítulo 1 nombrado Fundamentación Teórica constituye una disertación sobre el tema de los videojuegos de estrategias y los conceptos fundamentales de dos técnicas muy usadas en estos, los Mapas de Influencias y las Redes Neuronales Artificiales para la toma de decisiones.

Se aborda además el estado del arte y la actualidad nacional, específicamente en la Universidad de las Ciencias Informáticas, referente al tema en cuestión. El Capítulo 2 titulado Descripción y Aplicación de las Redes Neuronales Artificiales y los Mapas de Influencia engloba lo concerniente a la descripción del modelo de red a utilizarse y la combinación de esta con los Mapas de Influencia, además se realiza una propuesta de solución para obtener las variables del mapa. El Capítulo 3 denominado Resultados Generales, aborda todo el proceso de la aplicación de los Mapas de Influencia con las Redes Neuronales Artificiales, comenzando por la propuesta del algoritmo que describe por pasos todo el proceso para lograr el vínculo entre estas técnicas, y culminando con la ejecución de un Demo que arroja como resultado la validación de la investigación.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

El presente capítulo propone al lector un acercamiento al estudio de los videojuegos de estrategia, su evolución, características fundamentales así como la puesta en práctica de la inteligencia artificial en esta rama de los videojuegos. Se explica detalladamente el concepto de Mapas de Influencia que se encuentran dentro de estos videojuegos y una de las técnicas de inteligencia artificial que se utiliza en las decisiones estratégicas, los modelos de Redes Neuronales Artificiales.

### 1.2 Evolución de los videojuegos de estrategia

Los videojuegos hicieron su aparición a principios de la década de 1970 en California. Desde ese entonces han alcanzado una gran relevancia cultural y económica. El origen de los videojuegos de estrategia está fuertemente vinculado con los juegos de mesa de estrategia. Algunos juegos de mesa antecesores de los videojuegos de estrategia fueron el ajedrez y backgammon. Este tipo de videojuego posee generalmente una de las cuatro formas de jugadas que se usan en este género, puede por turnos, en tiempo real, de estrategia militar o por tácticas. En un principio los videojuegos de este género se limitaban a reproducir los mecanismos propios de los juegos bélicos de tablero. Sin embargo, con el paso del tiempo han evolucionado hasta convertirse en uno de los géneros más complejos que existen.

### 1.3 Subgéneros

No todos los videojuegos de estrategia son iguales. Existen varios subgéneros y distintas formas de llevar a cabo la acción. Aunque la mayoría de los títulos de este género mezclan estrategia y táctica, cada uno lo hace en diferentes grados. La táctica es la disposición y posterior movimiento de las unidades dentro del campo de batalla. Por otra parte la estrategia se define como el conjunto de decisiones e ideas que se ponen en marcha durante una campaña militar, teniendo en consideración la logística, el aprovisionamiento, etc. Los subgéneros de videojuegos de estrategia se clasifican de dos formas:

***Estrategia en tiempo real y Estrategia por turnos.***



## 1.3.1 Estrategia en tiempo real

Los videojuegos de estrategia en tiempo real se conocen también por sus siglas en inglés RTS (Real-Time Strategy). Este tipo de videojuego se encuentra dentro de los favoritos de los usuarios, ya que son más dinámicos, contienen más acción y suelen incorporarse gráficos más trabajados. Generalmente aplicado solo a ciertos videojuegos de estrategia, el nombre "juego de estrategia en tiempo real" indica que la acción en el juego es continua y los jugadores tendrán que tomar sus decisiones en un estado de juego constantemente cambiante, es decir, la parte estratégica se realiza en paralelo junto con el desarrollo de la acción.[\[14\]](#) Las reglas del juego se caracterizan por la obtención de recursos, construcción de bases, investigación de tecnologías y producción de unidades, todo este ejercicio se debe realizar de forma rápida ya que en cualquier momento se puede producir un cambio que altere el curso de la historia. Además solo existe un jugador que debe controlar docenas de edificios, y eventos que pasan a su alrededor. Los jugadores expertos pueden cambiar rápidamente entre muchas tareas distintas, sin embargo los jugadores casuales tienen más problemas con esto. Muy pocos juegos de estrategia que no sean videojuegos son en tiempo real.

Ejemplos de videojuegos de este género son:

**Starcraft:** Es uno de los videojuegos más vendidos de ciencia ficción para PC. Fue creado por Blizzard Entertainment. Salió a la venta el 31 de marzo de 1998. [\[14\]](#) El videojuego está basado en el siglo 26, en el que tres razas (Terran, Zerg y Protoss) de distintos planetas, luchan por la supremacía y la supervivencia. El nombre en inglés está compuesto por *Star* que significa "estrella" y *Craft* significando "embarcación o nave". La nave espacial *BattleCruiser* es una unidad pesada de aire de la raza Terran del videojuego *Starcraft* [\[12\]](#), utiliza redes neuronales artificiales para controlar los NPCs así como para guiar negociaciones, comercio y combate [\[11\]](#). Como se puede observar en la foto dispone de un gran cañón en el centro que una vez cargada toda su energía es capaz de eliminar unidades enemigas de un sólo disparo [\[12\]](#) (Fig.1)

**Pokémon Black & White.** entregas de la saga para Nintendo DS. El videojuego cuenta con un sistema de combate aleatorio, que permite luchar online contra otros jugadores desde la terminal global en el Pokémon Center. Permite utilizar un dispositivo llamado C-Gear para acceder a un buen surtido de opciones inalámbricas. Una vez hayas capturado un pokémon, podrás ver sus datos mediante la pokédex (roja la del chico, rosa la de la chica). [\[13\]](#) En Black & White, el jugador tiene una criatura que aprende del

jugador y de otras criaturas. La mente de la criatura incluye una combinación de representaciones simbólicas y sub-alterno-simbólicas, con deseos representados como redes neuronales artificiales. Las criaturas de los juegos usan técnicas de vida artificial, incluyendo redes heterogéneas en las que las neuronas son divididas en lóbulos que tienen juegos individuales de parámetros. En la combinación con los algoritmos genéticos, las criaturas usan las redes para aprender conductas y preferencias, con el tiempo.[11] (Fig.2)



**Fig.1** Nave *BattleCruiser* de *StarCraft*.



**Fig.2** Videojuego *Black & White*.

Dentro de los videojuegos de estrategia en tiempo real se derivan otros subgéneros entre los que se encuentran Build and Battle, Tácticos no wargames, Tácticos hombre a hombre y Acción táctica.

### **Build and Battle (B&B).**

Son videojuegos en los que lo principal es el ciclo de recolectar recurso, construir edificios, unidades y llevar a cabo la batalla. Su nombre en español es “Construir y batallar”. Es el subgénero que ha tenido mayor número de ventas. En este tipo de videojuegos se encuentran *Dune*, *Rise of Legends* y *Dawn of War*.

### **Tácticos no wargames.**

Son videojuegos tácticos menos orientados a la representación realista de batallas. Su traducción al español es juegos tácticos que no son de guerra. Dentro de los videojuegos de esta categoría se encuentran *Nexus* o *Ground Control*.

## **Tácticos hombre a hombre.**

Es una variante de la clasificación anterior, pero en esta se manejan unidades individuales. Algunos videojuegos de este género son *X-COM*, *Interceptor* y *Command*.

## **Acción táctica.**

Es una mezcla entre videojuego de acción y táctico hombre a hombre. En este se manejan pequeñas unidades de operaciones especiales 3D en primera persona. Ejemplos de estos videojuegos son *SWAT* y *Rainbow Six*.

### **1.3.2 Estrategia por turnos**

También se les conoce por sus siglas en inglés TBS (Turn-Based Strategy). En este tipo de videojuegos el flujo queda particionado en partes visibles llamadas turnos. Un jugador de un videojuego por turnos posee un período de análisis antes de realizar una acción, vienen dados de dos formas dependiendo de la acción, en un turno, los jugadores juegan simultáneamente o por otro lado juegan sus turnos en secuencia, es decir, van realizando sus diferentes movimientos, uno detrás de otro [\[15\]](#). Los primeros son llamados videojuegos de estrategia por turnos simultáneos y los otros, juegos de estrategia por turnos alternados.

La diferencia entre un videojuego de estrategia en tiempo real y uno por turno recae en que este último posee menos acción y detalle gráfico pero a su vez permite concentrarse más en la estrategia y depender menos de la agilidad con que se juega. Casi todos los juegos de estrategia que no sean videojuegos (juegos de mesa) son por turnos.

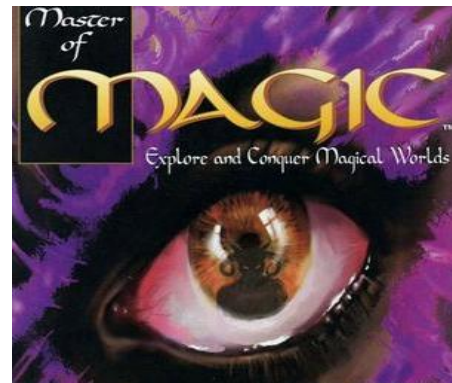
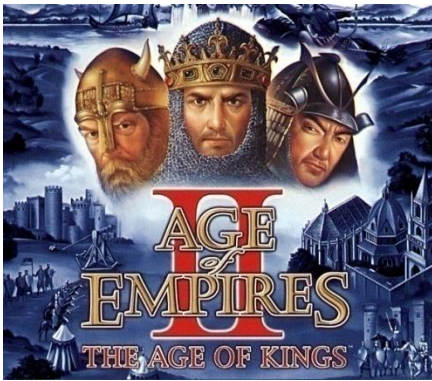
Algunos videojuegos recientes han mezclado componentes en tiempo real con componentes por turnos, estos le conceden a los jugadores 100 movimientos por día. Pueden ser tomados en cualquier momento de ese día independientemente de si otros jugadores hayan realizado sus movimientos o no.

Ejemplos de este género son:

**Age of Empires**: Es uno de los grandes juegos históricos de este género. Fue creado por Ensemble Studios y lanzado el 26 de octubre de 1997. Consiste en una línea de tiempo de 3000 años, desde la Edad de Piedra hasta la Edad de Hierro. El jugador puede elegir entre doce civilizaciones para su ataque.

Entre las civilizaciones se encuentran persa, griega, egipcia y shang. Su traducción al español significa “Era de imperio o Siglo de imperio”.\_(Fig.3)

***Master of Magic:*** Es un juego de construcción de imperios lleno de fantasías. Fue creado en 1993 por Simtex y especialmente para MSDOS. Su traducción al español “Maestro de magia” y ese es exactamente el objetivo del juego, para llegar a este objetivo se debe usar la hechicería o la táctica militar. Se desarrolla en dos mundos, el real llamado Arcanum y el alterno llamado Mirror.(Fig.4)



**Fig.2** Videojuego de historia Age of Empires    **Fig.3** Videojuego de fantasía Master of Magic

Dentro de este subgénero se encuentran tipos de videojuegos de Construcción de Imperios, Artillería y Semi por turnos.

### **Construcción de imperios**

Estos videojuegos son conocidos también como **4X**, son siglas en inglés que significan: **eXplore** (exploración del mapa), **eXpand** (expansión territorial), **eXploit** (explotación de recursos), **eXterminate** (exterminación del enemigo). Son videojuegos en los que se debe explorar, expandir, explotar y exterminar. Pueden ser por turnos o en tiempo real. Tienen como objetivo la creación y mantenimiento de un imperio. Pueden tener un ambiente histórico, fantástico o ficticio. Ejemplos de videojuegos de este tipo son **Colonization, Anacreon y Dominions**.

## Videjuegos de artillería

Son videjuegos generalmente por turnos, en los que los tanques se atacan unos a otros. También se les denominan videjuegos de disparos. Entre los ejemplos de videjuegos de artillería se encuentran ***Worms, Gunbound, Scorched Earth y Tanarus y Gorilla.***

## Semi por turnos

Este tipo de videjuego combina fases en tiempo real con otras de turnos. Ejemplos de este género son la saga ***Total War o Centurion.***

### 1.3.3 Videjuegos de guerra.

Los componentes principales de este tipo de videjuego son guerras tácticas o estratégicas en un mapa. Los videjuegos de guerra pueden ser por turnos o en tiempo real y de estrategia o táctica[15]. Estas clasificaciones dividen los juegos de guerra en 4 categorías: estrategia en tiempo real, estrategia por turnos, táctica en tiempo real y táctica por turnos.

Ejemplos de videjuegos de guerra:



**Fig.4** Videjuego *Final Fantasy*



**Fig.5** Videjuego *Close Combat*

## Táctica por turnos

Conocidos también por sus siglas TBT (Turn-Based Tactics). La jugabilidad táctica por turnos se caracteriza por la expectativa de los jugadores, por completar sus tareas usando solo las fuerzas de combate que se les proveen y usualmente por la disposición de una representación realista (o por lo

menos creíble) de operaciones y tácticas militares. Ejemplos del género son *Jagged Alliance*, la saga *X-COM*, así como *Final Fantasy Tactics*. **(Fig.5)**

## Táctica en tiempo real

Se les conoce también por sus siglas en inglés RTT (Real-Time Tactics). Compartiendo aspectos de los videojuegos de simulación y de guerra, los videojuegos de táctica en tiempo real se enfocan en aspectos operacionales y control de guerra. A diferencia de los videojuegos de estrategia en tiempo real, el manejo económico y de recursos y la construcción de edificios no forman parte de las batallas. Algunos ejemplos son *Dark Omen*, *World in Conflict* y la saga *Close Combat* **(Fig.6)**.

## 1.4 Videojuegos de Rol

Son inspirados en los juegos de rol clásicos, donde el protagonista interpreta un papel y ha de mejorar sus habilidades mientras interactúa con el entorno y otros personajes. Se debe actuar y tomar decisiones en función de las propuestas del juego y de las acciones del resto de los personajes. Su diseño propone una jugabilidad lenta. [\[15\]](#) **Juego de rol** es la traducción usual en castellano del inglés *Role-Playing Game*, literalmente: juego de interpretación de papeles. Estos videojuegos mezclan elementos narrativos y estratégicos. Son una evolución de los juegos de guerra con figuritas de plomo. A mediados de los setenta surgieron los primeros videojuegos llamados *Dungeons & Dragons*. Hoy día, los juegos de rol son auténticas “novelas” interactivas, donde el Director de Juego ejerce de narrador, ideando el guión y caracterizando a los personajes secundarios y los jugadores interpretan papel aventurero. La duración de un sólo juego de éstos (en la actualidad demandan mucho tiempo), puede variar mucho para su terminación o fin por parte del usuario promedio. Si se juega a un ritmo sano, detallista y a fondo, no es difícil observar un periodo de varios meses o más de un año.

A principios de los ochenta existía ya el propósito de adaptar los juegos de rol a la computadora. Desde entonces, surgieron estilos de juego muy diferentes entre sí, como *Aventura Original* de CPC *Amstrad*, donde el jugador tenía que teclear manualmente las instrucciones “ir hacia”, “coger objeto”, “mirar tal cosa”, otro fue *Morrowind* que posee un auténtico mundo virtual. Este género de videojuego llegó a Europa en la segunda mitad de los ochenta y en España empezó a despegar a inicios de los noventa. Actualmente, la mayoría de los jóvenes y adolescente han jugado una partida de rol, ya sea en su versión de mesa o en el ordenador. Sin embargo, no es un género al que se le produzcan versiones muy a menudo. En cambio, puede afirmarse que la calidad de estos títulos es bastante notoria, lo que aleja a los

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

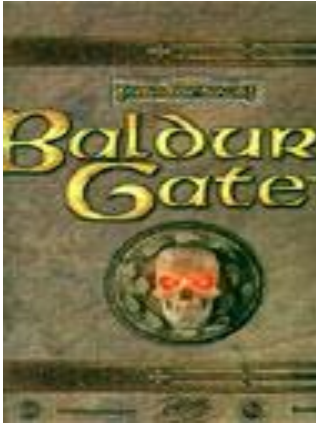
juegos de rol de otros terrenos, como el de la estrategia, donde los títulos se cuentan por decenas, pero la calidad media de todos ellos es bastante discutible

Son diversos los subgéneros que engloba esta categoría entre ellos están: Roguelike (juegos de rol multi-jugador masivos en línea), MUD (Multi User Dungeon) o los Strategic Role-Playing Games (tácticos a medio camino entre el género de estrategia y el rol) También otro subgénero no tan popular son los RPG BPT (Battle per Turn), que se caracterizan por ser un videojuego de rol normal, pero al enfrentarse con los enemigos no es en tiempo real, sino por turno.

Ejemplos de videojuegos de Rol:

**Baldur's Gate:** Su traducción al español "Puerta de Baldur", es el nombre de una saga de videojuegos de rol para ordenador. Los juegos toman el nombre de una ciudad ficticia en la que se desarrolla parte de la acción de ambos. Dicha ciudad se encontraría en los Reinos Olvidados, ubicada en el norte del río Chiontar, al sur de Aguas Profundas y al norte de Amn. Estos videojuegos se caracterizan por la amplitud de su historia y la cantidad de aventuras secundarias presentes. El jugador crea un personaje al que se le unen otros que el controla igualmente, hasta completar un equipo de seis aventureros. **(Fig.7)**

**Kingdom Hearts:** La traducción al español de este videojuego es "Corazones Unidos". Es el nombre de una saga de videojuegos, perteneciente al género de juegos de rol de acción, surgida de la unión de Square y Disney. El videojuego combina personajes y lugares de algunas películas de dibujos animados de Disney con algunos personajes de la saga Final Fantasy. La acción gira en torno a Sora, un niño de 14 años que vive en Destiny Islands junto a sus amigos, Kairi y Riku. Ellos desean salir de esa isla en la que se encuentran, pero la noche antes del viaje, una tormenta oscurece toda la isla y aparecen unas criaturas llamadas **sin corazón**. **(Fig.8)**



**Fig. 6** Videojuego Baldur Gate



**Fig.7** Videojuego Kingdom Hearts

Los subgéneros de videojuegos de rol que existen son los clásicos, acción, japoneses, tácticos y multi-jugador masivos en línea, que es uno de los más importantes.

### 1.4.1 Juegos de rol multi-jugador masivos en línea

Los **MMORPG** (del inglés Massively Multiplayer Online Role-Playing Games), son los videojuegos de rol que permiten a miles de jugadores introducirse en un mundo virtual de forma simultánea a través de Internet e interactuar entre ellos teniendo las características clásicas de un RPG. Se trata de crear un personaje, del cual puedes elegir su raza, profesión, armas, etc., al que se le puede ir aumentando sus niveles y experiencia en peleas contra otros personajes, criaturas del videojuego o realizando diversas aventuras (o misiones) llamadas quests. Este género de RPGs es diferente de un videojuego de rol en línea multi-jugador no masivo, principalmente porque los videojuegos de rol en línea tienen un número limitado de jugadores, es decir, los MMORPGs están preparados y elaborados de tal manera que admiten cualquier número de jugadores simultáneos (aunque viene limitado por la conexión del servidor).

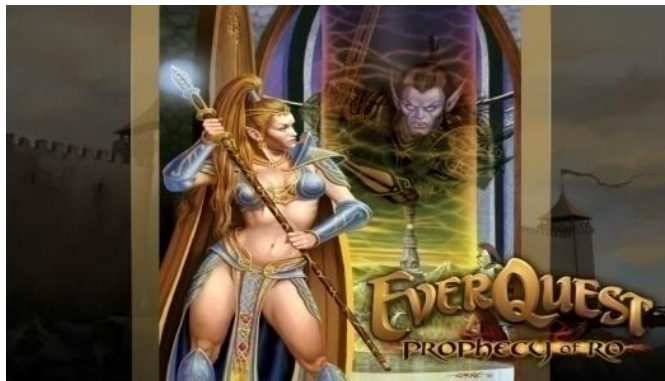
Los juegos MMORPGs se caracterizan por ser ricos en gráficos 2D y 3D. La cifra de jugadores que pueden alcanzar no está limitada, es decir, pueden tener una comunidad de 115 mil jugadores. Poseen mayores opciones de personalización y mezclan el tiempo real con los turnos. Las batallas se tornan dinámicas gracias al sistema combos.



Se adaptó a este género una de las licencias más asombrosas de la consola de SEGA, se trata de Panzer Dragoon Saga, el videojuego más caro de la generación 32/64 bits, hoy día una pieza de coleccionista.

Ejemplos de de este género son **Everquest**, **Ultima Online** y **Granado Espada**.

**Everquest (EQ):** Es un famoso **MMORPG** (Massively Multiplayer Online Role-Playing Game) que fue lanzado en Marzo de 1999. En el mundo del EverQuest, el jugador explora un mundo de fantasía, con espadas y magia, luchando con monstruos para conseguir tesoros y puntos de experiencia mientras interaccionan con otros jugadores. Cuanto más avanzan los jugadores, suben de nivel, ganando poder, prestigio y habilidades. También pueden conseguir objetos para sus personajes de muchas maneras, matando monstruos (y recogiendo el botín que éstos llevan), llevando a cabo misiones (que les son procuradas por personajes no jugadores), o consiguiendo materiales básicos (madera, hierro, oro, etc) y forjándolos ellos mismos artesanalmente. EverQuest hereda gran parte de su estructura y reglas de los famosos videojuegos de rol tales como Dragones y Mazmorras (Dungeons & Dragons), así como de los MUDs en especial del DikuMUD. **(Fig.9)**

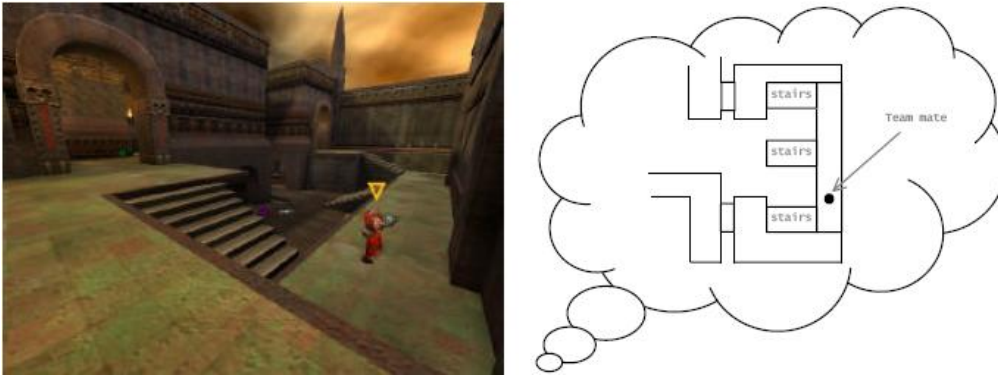


**Fig. 8** Videojuego Everquest

## 1.5 Modelos cognitivos

Son modelos internos que juegan un papel importante para el agente. A través de ellos el agente percibe y comprende el mundo virtual en el que vive, pero solo puede observar los aspectos del mundo del modelo cognitivo que posee, más allá le será imposible. Estos modelos son una representación simplificada del mundo en el que se encuentra. A pesar de los avances en el diseño 3D de hoy día y de los modelos sofisticados y precisos que existen para representar el mundo real, no son perfectos, ya que hay

demasiadas variables en juego que hacen inviable estos tipos de modelos. Las variables representan el estado del mundo. Este estado puede incluir la posición del jugador, de los enemigos, las armas reunidas por el jugador, obstáculos geométricos, ventanas, etc.[\[2\]](#)

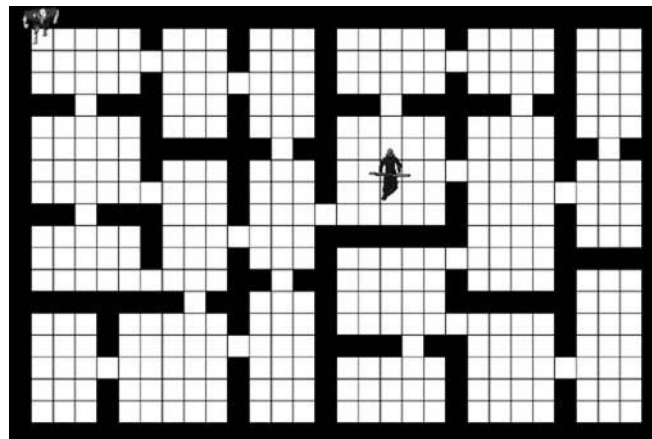


**Fig. 9** *Vista de un Modelo Cognitivo.*

## 1.5.1 Ejemplos de modelos cognitivos

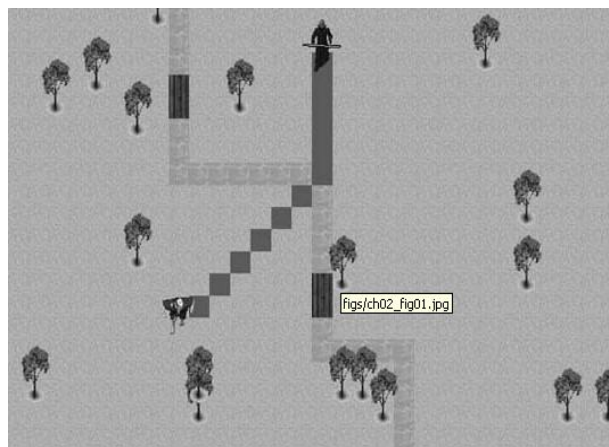
### Rejillas Regulares

Los videojuegos de guerra tienen grandes y complejos entornos basados en cuadrados o hexágonos. Se puede construir una matriz para proporcionar información sobre cada celda. Cada celda puede representar una pared en el mundo virtual. [\[2\]](#)



**Fig. 10** *Representación de un modelo cognitivo basado en rejillas.*

Esta matriz puede representar también el tipo de terreno en cada celda. La siguiente figura muestra un terreno, es decir, un modelo cognitivo de un mundo virtual de un videojuego de estrategia. A esta matriz se les han dado valores a las celdas para representar árboles, terreno, el enemigo, el jugador, etc. Otros valores pueden ser asignados para representar agua, terreno de rocas, colinas, entre otros. A partir de esos valores el agente puede planificar por donde moverse o atacar ya que toda la información se encuentra almacenada en dicha matriz. [2]



**Fig. 11** Representación de un terreno.

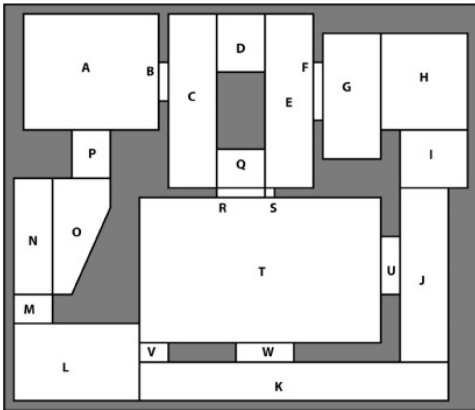
Se puede aplicar un algoritmo de búsqueda para encontrar los caminos y evitar las celdas que representan las paredes, pero también están las celdas que representan el agua y el barro, por lo que no sería un algoritmo muy eficiente. [2]

Si es necesario saber de qué color es el suelo, con tan solo buscar la celda que posee esta variable bastaría. También se puede encontrar la posición del enemigo y un lugar donde atacar. Incluso cuando este tipo de modelo es fácil de utilizar y aplicar, la desventaja de usar celdas es que los espacios de búsquedas pueden ser muy grandes. Incluso un modesto mapa de celdas de 100x100 tendrá un gráfico compuesto por 10.000 nodos y 78.000 bordes. [2]

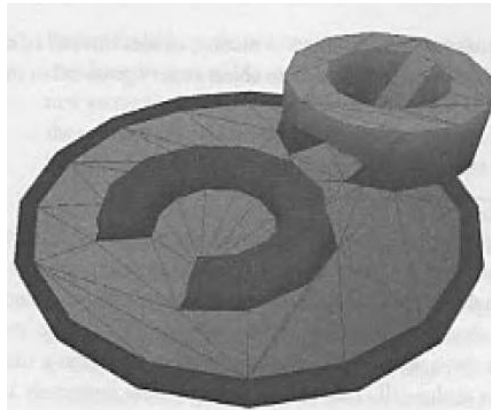
## **Navegación de malla**

Uno de los modelos más utilizados para representar mundos 3D es la malla de navegación. Es un conjunto de polígonos convexos que describen el tránsito de la superficie de un entorno 3D. Es un

programa sencillo, muy intuitivo que los personajes usan para la navegación y ser precursor en el mundo. Las mallas son muy adecuadas para representar casi cualquier tipo de entorno 3D mientras el mundo del videojuego no cambie drásticamente. Este modelo no se ocupa de obstáculos como enemigos, personajes, cajas, etc. [2]



**Fig.12** Polígonos.

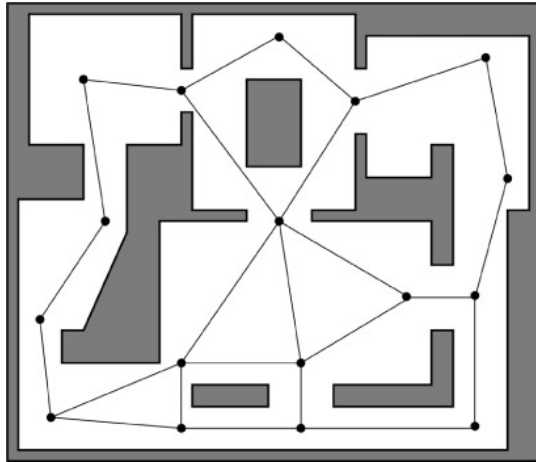


**Fig.13** Malla en 3D con alambres de soporte.

Una ventaja de la malla de navegación es que el tamaño de la malla no depende del tamaño del mundo, excepto de las formas de las paredes y la geometría del medio ambiente. En el interior de cada uno de los polígonos convexos el jugador puede circular libremente. Es fácil calcular el movimiento que el agente necesita cuando necesita hacer movimientos bruscos en las mayorías de las direcciones para evitar la colisión o el fuego enemigo. Sin embargo, el inconveniente es que puede ser difícil y tedioso encontrar todos los polígonos transitables y las formas de llegar a ellos desde los polígonos vecinos. Algunas veces este proceso no resulta con el modelo físico del mundo y pueden ser marcados algunos polígonos cuando en realidad no hay manera de avanzar en ellos. [2]

## Puntos de Visibilidad

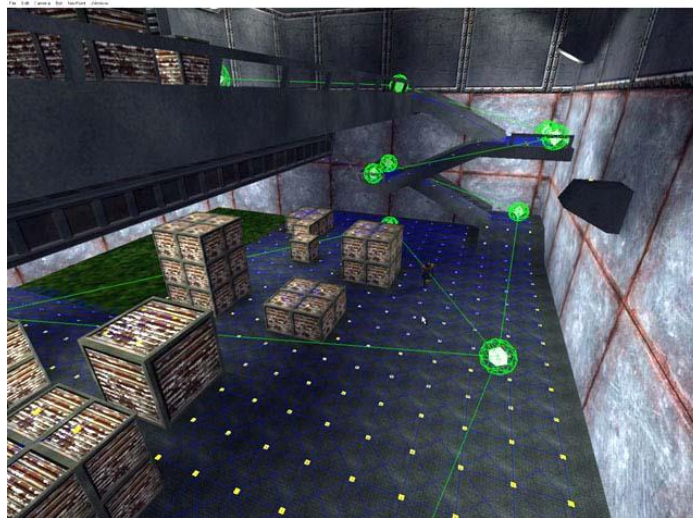
Un Punto de Visibilidad (POV) gráfico de navegación es creado mediante la colocación de puntos, por lo general a mano, en los lugares importantes del medio ambiente, de manera que cada nodo tiene una línea gráfica de la vista del otro nodo. Colocados cuidadosamente, los nodos hacen una conexión gráfica de todas las áreas importantes en la geometría del mundo. [2]



**Fig. 14** *Puntos de Navegación.*

Cada uno de los puntos en el mapa está conectado a otros que se pueden ver desde el nodo actual. Estos nodos no tienen que colocarse únicamente en 2D como se muestra en la **Fig.15**.[\[2\]](#)

La posición de los nodos se puede encontrar en el mundo 3D para representar posibilidades de circulación como se muestra en la siguiente figura. [\[2\]](#)



**Fig. 15** *Puntos de Navegación en 3D.*

Una de las características de los **POV** es que pueden ser fácilmente ampliados para incluir a los nodos que poseen información más allá de la conectividad de datos. Por ejemplo, los nodos se pueden añadir fácilmente a un gráfico para representar francotiradores, posiciones o emboscadas. El inconveniente es que si un mapa de juego es grande y complejo, el diseñador del mapa se puede gastar un montón de tiempo en posicionamiento y ajuste del gráfico. Un gráfico POV también puede ser un problema si va a incluir cualquier tipo de función de generación de mapa, puesto que entonces debe desarrollar un método automatizado para generar la estructura gráfica, así como para los nuevos mapas.[\[2\]](#)

## 1.6 Mapas de Influencia

Para ser capaz de navegar a través del mundo virtual y encontrar ciertos lugares, el jugador también tiene una representación del nivel o el mapa en el que se encuentra. Dichos mapas son llamados Mapas de Influencia. Ellos proporcionan una perspectiva que permite la evaluación estratégica y la toma de decisiones basadas en el estado actual del videojuego, lo que permite que el jugador posea una representación espacial de los conocimientos sobre el mundo. Está formado por varias capas, cada una representando las diferentes variables, más una representación geográfica del mapa de juego. Cuando una decisión debe ser hecha por el jugador, algunas o todas estas capas se combinan a través de una suma ponderada proporcionando una idea global de la idoneidad de cada área en el mapa de la decisión actual. Sin embargo, la utilización de una suma ponderada tiene ciertas limitaciones. Un modelo de Red Neuronal Artificial puede ser utilizado en lugar de una suma ponderada para analizar los datos del mapa y tomar una decisión estratégica.[\[8\]](#)

Cada capa de dato es un repositorio de información sobre diferentes mundos y cada celda es una base de datos relevantes para todas las unidades y los recursos que ocupan la celda. Por ejemplo, las capas pueden almacenar los datos para el combate, la fuerza, los activos vulnerables, la visibilidad de la zona, el recuento de cuerpo, los recursos y transitabilidad de cada celda en el mapa.[\[8\]](#)

A partir de los elementos que se encuentran en las capas el jugador tiene la necesidad, en un estado del videojuego, de tomar una decisión para satisfacer una estrategia determinada. Las decisiones que se llevan a cabo durante un videojuego pueden ser estratégicas o tácticas.[\[8\]](#)

## 1.7 Toma de decisiones en videojuegos

Se define como la acción de realizar conclusiones lógicas y movimientos razonables. Esta acción está limitada a la cantidad de información que se procese. Es decir, mientras más acciones el videojuego permita realizar a los agentes, más respuestas o estados se generarán. Existen dos tipos de soluciones a los problemas que pueda enfrentar un agente: **tácticas** y **estratégicas**.

Una **solución estratégica** es vista a largo plazo para poder satisfacer las metas más difíciles, lo que puede involucrar un conjunto de acciones para su conclusión. Ejemplo de este tipo de solución es un juego militar donde debemos mover un grupo de soldados desde el campo de batalla hacia la parte vulnerable de una ciudadela para evitar un ataque sorpresa.

Una **solución táctica** se relaciona con soluciones menos complejas que pueden ser alcanzadas a corto plazo y usualmente involucran la ejecución de una acción o habilidad específica. Esta solución puede presentarse en la acción de trasladar a los soldados hacia el punto destino, ya que deberán encontrar un camino cercano al óptimo y actuar de forma coherente en caso de encontrar enemigos en el trayecto.

Cualquier agente controlado por el motor de Inteligencia Artificial obtiene información del mundo por medio de la percepción y toma decisiones razonables e inteligentes en relación a la acción que se da como respuesta.

La **percepción** es un proceso en el que se recibe información del mundo virtual en el que se encuentra el agente. Las capacidades de percepción de un agente pueden variar en diferentes aspectos como: frecuencia con la que reciben los datos, capacidad para retenerlos, su sensibilidad espacial, etc. Existen dos tipos de percepciones, la externa y la interna.

La **percepción externa** se refiere a los sucesos del mundo virtual fuera del propio agente. Generalmente se pone de manifiesto en agentes con características humanas. A estos le son concebidos los sentidos del oído y la vista.

La **percepción interna** proporciona información al agente sobre él mismo. Es decir, le informa sobre sus motivaciones, estado interno y deseos.

## 1.7.1 La inteligencia artificial en videojuegos de estrategia

La inteligencia artificial es la rama de la ciencia informática que se encarga de desarrollar agentes en mundos virtuales con la capacidad de recibir entradas, es decir, de percibir su entorno y actuar en este, también debe procesar salidas que sean racionales con la característica de que sean semejantes a la del mundo real. Existen varios tipos de procesos para obtener un agente racional, este documento trabajará con uno en especial, las Redes Neuronales Artificiales (semejantes al funcionamiento del cerebro humano). Las RNA están dentro de una de las técnicas de inteligencia artificial para un agente, se denomina **Técnica No Determinista**.[\[2\]](#)

La Técnica no determinista posee un grado de incertidumbre y es algo impredecible. No hay que codificar explícitamente todas las posibles situaciones en el juego, ya que los elementos inteligentes pueden incluso extrapolar sus comportamientos y desarrollar otros nuevos o emergentes. Un ejemplo de comportamiento no determinista es un personaje non-player que tiene que aprender a adaptarse a las tácticas de combate de un jugador. Este aprendizaje puede usar una red neuronal, una técnica bayesiana, o un algoritmo genético.[\[2\]](#)

También existen las técnicas **Deterministas** aplicadas a los videojuegos, pero no serán tratadas a profundidad en la investigación. Para dar un breve resumen de este tipo de técnica podemos decir que en ella el comportamiento del agente es especificado por completo por los programadores. Las más usadas son las máquinas de estado, sistemas de reglas de producción y los árboles de decisión.[\[2\]](#)

La inteligencia artificial desarrollada para videojuegos se puede utilizar para desarrollar estrategias más o menos exitosas mediante la selección natural. Este pedazo de código en un videojuego hace que los componentes manejados por la computadora parezcan tomar decisiones inteligentes. Puede ayudar a modificar preferencias en un videojuego de estrategia en tiempo real de tal manera que un oponente controlado por la computadora podría evolucionar, escoger mejores acciones, aprender y más, pero esta posee limitada disponibilidad de recursos para realizar sus cálculos en un videojuego de este tipo, por lo que muchas veces opta por obtener una solución aceptable pero no óptima. Esta desventaja no suele convertirse en un problema ya que los juegos deben ser interesantes y divertidos, no imposibles de ganar, hay que mantener un balance adecuado.[\[2\]](#)



## 1.8 Redes Neuronales Artificiales

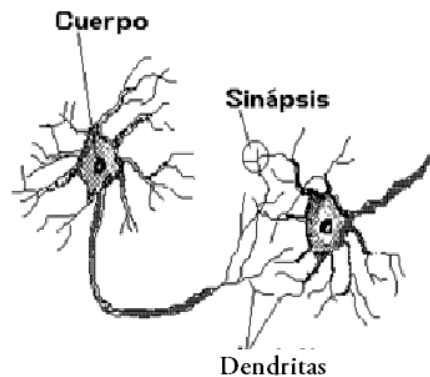
Las Redes Neuronales Artificiales son una simulación de las redes de neuronas que forman al cerebro. Son un elemento importante de las denominadas tecnologías de inteligencia artificial y tienen como objetivo fundamental dar respuestas similares a la que es capaz de dar el cerebro. Estas tienen una capacidad de aprendizaje a partir de un entrenamiento que se les debe hacer para dar cumplimiento a una determinada tarea. Su velocidad de respuesta es muy similar a la del cerebro. El ser humano no necesita pensar mucho para poder reconocer un objeto, cumplir una acción, entre otras órdenes hechas al cerebro, de igual manera es el funcionamiento de una Red Neuronal Artificial. Deben ser construidas de manera que el conocimiento adquirido se encuentre distribuido a través de toda la red para que al ocurrir un error en una parte de la misma se puedan seguir generando respuestas correctas.[\[1\]](#)

### 1.8.1 La neurona natural y el modelo biológico

El modelado de las Redes Neuronales Artificiales está basado en la estructura y funcionamiento de los sistemas nerviosos, en el cual una neurona es el elemento fundamental. Cada neurona recibe información de una neurona precedente, a través de estructuras llamadas dendritas, cuando se combinan producen como resultado que la neurona adquiera un nivel de activación, estas regulan la transmisión del impulso eléctrico mediante unos elementos bioquímicos, los neurotransmisores y generan señales de tipo químico; además emiten impulsos eléctricos a lo largo de una fibra delgada denominada axón hacia las dendritas de otras neuronas, esta señal es de tipo eléctrico. Entre las dendritas y el axón se produce una conexión denominada sinapsis.[\[1\]](#) (Fig.17)

Se puede considerar que la información que se encuentra en el cerebro esta codificada como impulsos eléctricos que se transmiten a través de las neuronas y los impulsos se verán modificados básicamente en las sinapsis, cuando las señales son químicas la codificación del aprendizaje estará en las sinapsis y en la forma que dejen pasar o prohibir las señales.[\[1\]](#)

En resumen, se podría decir que la información en el cerebro se transmite mediante impulsos eléctricos, que viajan por las neuronas biológicas y por señales químicas que comunican las neuronas con otras neuronas en sus bordes usando unas estructuras llamadas sinapsis.[\[1\]](#)



**Fig. 16** Modelo de una Red Neuronal Natural.

## 1.8.2 Características de las Redes Neuronales Artificiales.

Una red neuronal posee cuatro aspectos fundamentales que forman su caracterización: topología, mecanismo de aprendizaje, tipo de asociación que se realiza entre la información de entrada y salida, y la forma en la que se representan estas informaciones.

Otros aspectos fundamentales son: el peso, las fases de operación, además de otros aspectos importantes.

**Pesos:** Las redes neuronales pueden tener factores de pesos adaptables o fijos. Las que tienen pesos adaptables emplean leyes de aprendizaje para ajustar el valor de la fuerza de interconexión con otras neuronas. Si se utilizan pesos fijos, su tarea debe estar previamente definida. Los pesos son determinados a partir de una descripción completa del problema a tratar. Los pesos adaptables son muy importantes, ya que no se conoce de antemano su valor correcto. [\[10\]](#)

**Fases de operación:** Una red neuronal tiene dos fases de operación, la de entrenamiento y la de recuperación de lo aprendido. En la fase de recuperación se le debe dar a la red una serie de estímulos de entrada y salida (dependiendo del caso), para que pueda ajustar los pesos de interconexión y minimizar al máximo el error en la salida que se va generar.

La fase de entrenamiento se describe en el **epígr. 1.8.4.4**

**Necesidad de un patrón:** Las redes neuronales necesitan un patrón para reconocer y poder ejecutar algún tipo información. Estas resuelven problemas de asociación, evaluación y reconocimiento de patrones.[\[10\]](#)

**Tolerancia a Fallos:** La red puede tener fallos, por lo que debe implementarse con procesamientos independientes, esto trae como ventaja fundamental que al ocurrir algún daño o destrucción parcial de la información en la red no afecte a la misma, pues cada procesamiento se encarga de diferentes partes del funcionamiento de ella.[\[10\]](#)

## 1.8.3 Elementos de una Red Neuronal Artificial

### 1.8.3.1 La neurona artificial

La neurona artificial es un dispositivo simple de cálculo que genera una salida a partir de un vector con datos de entrada. Esta neurona está compuesta por varios elementos:

**-Conjunto de entradas:** Conjunto de variables o datos de entrada.

**-Pesos sinápticos:** Intensidad de interacción entre neuronas.

**-Regla de propagación:** Proporciona el valor de potencial de la neurona en función de las entradas y sus pesos.

**-Función de Activación:** estado de activación actual de la neurona en comparación con el estado anterior y su potencial.

**-Función de salida:** Proporciona la salida actual de la neurona de acuerdo a su estado de activación[\[10\]](#)

### 1.8.3.2 Funciones de Transferencia o de Activación

Existen diversas funciones para la activación de una neurona artificial, estas funciones son conocidas como de transferencia o activación. Cada capa oculta y la capa de salida tienen un complemento para combinar la señal de entrada y una función de activación para determinar su nivel de activación, en la

**Ecuación 1** se muestra la función de activación. [\[8\]](#)

$$a = \sum_{i=1}^{i=n} x_i w_i$$

**Ecuación 1**

Cada valor de entrada se calculará multiplicando la unidad de entrada por su peso asociado. Esto da la activación de la unidad como se muestra en la **Ecuación 1**. Donde **a** es la activación, **n** es el número de las unidades, **X<sub>i</sub>** es la unidad, y **W<sub>i</sub>** es el peso.[\[8\]](#)

Dentro de estas variables se encuentra el valor umbral que tiene como objetivo brindarle a la red una flexibilidad mayor en el proceso de aprendizaje. Para determinar la señal de salida o grado de excitación de la neurona artificial se utiliza la función de activación a lo que se le adiciona el valor umbral. En la **Ecuación 2** se muestra como se calcula la función de propagación. [\[9\]](#)

$$h(x; b, w) = b + \sum_{i=1}^n w_i x_i$$

**Ecuación 2**

Existen diferentes funciones de activación, algunas de las funciones más utilizadas se conocen con el nombre de: Escalón, Sigmoid y Lineal.[\[8\]](#)



**Fig. 18** *Función Sigmoid*



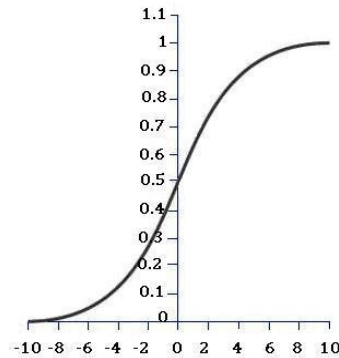
**Fig. 19** *Función Escalón*



**Fig. 20** *Función Lineal*

## Función Sigmoid

Esta es una función no-lineal. Devuelve valores continuos en el intervalo (0, 1) con asíntotas horizontales  $f(x) = 0$  y  $f(x) = 1$ . [9]

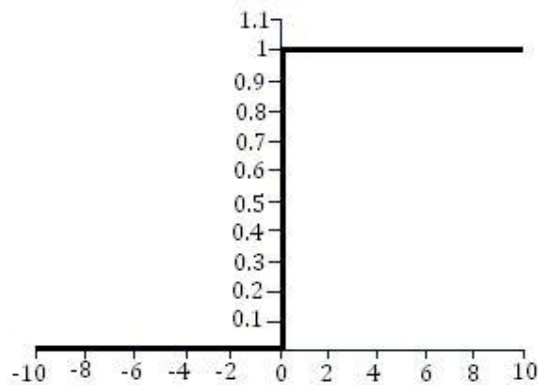


## Función Escalón

Esta función ha sido utilizada desde los principios de las redes neuronales, pero su derivada era muy difícil de evaluar. [9]

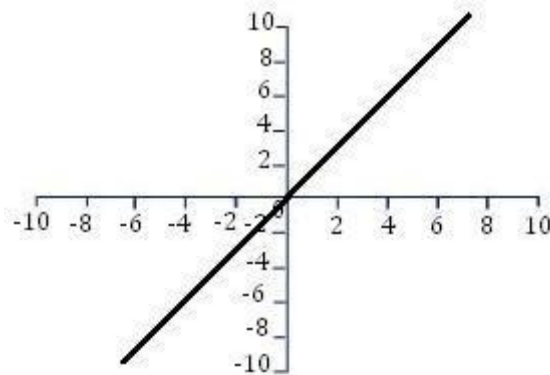
$$f(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases}$$

Ecuación 3



## Función Lineal

Posee una salida similar a la entrada neta de la neurona. Es común que se use en la capa de salida de un MLP.[\[9\]](#)



### 1.8.3.3 Aprendizaje de las Redes Neuronales Artificiales

Las redes neuronales artificiales se clasifican de acuerdo a su arquitectura y aprendizaje. El aprendizaje de las redes neuronales se basa en el entrenamiento de la red con patrones nombrados patrones de muestra o de entrenamiento. El proceso consiste en hacer que la red ejecute los patrones iterativamente, cambiando los pesos de las sinapsis hasta que estos converjan a un conjunto de pesos óptimos que hagan una buena representación de los patrones, cumplido esto mostrara una respuesta satisfactoria para esos patrones.[\[6\]](#) En general el algoritmo de entrenamiento se detiene cuando se ha alcanzado un error menor que una cantidad especificada; pero en ocasiones este criterio no es suficiente.

Existen diferentes tipos de aprendizaje entre los que se encuentran: el supervisado, el no supervisado, el hibrido y reforzado.

#### Aprendizaje supervisado

El aprendizaje supervisado consiste en presentarle a la red un conjunto de patrones y la salida deseada para estos patrones, para ello se usa una fórmula matemática de minimización del error que permita ajustar los pesos para dar una salida cercana a la salida deseada.[\[1\]](#)

Se puede mostrar un esquema general de este tipo de aprendizaje de la siguiente forma:

1. Inicializar los pesos de las sinapsis aleatoriamente.
2. Para cada patrón P perteneciente al conjunto de los patrones que tenemos
  - 2.1 Mostrar el patrón de entrada P y hacer la dinámica de la red para poder calcular la salida de la red S, no es la salida deseada, sino un patrón aleatorio ya que los pesos inicialmente eran aleatorios.
  - 2.2 Hallar el cálculo del error entre la salida de la red S, y la salida deseada del patrón P.
  - 2.3 Ajustar los pesos usando la regla de aprendizaje para disminuir el error medio.
3. Si el error es mayor de cierto criterio volver a hacer el paso 2; si todos los ejemplos se han clasificado correctamente, terminar.

Las redes más significativas que usan este aprendizaje supervisado son el Perceptrón, el Perceptrón multicapa y la red de Hopfield.[\[1\]](#)

## **Aprendizaje no supervisado**

En este tipo de aprendizaje no se necesitan mostrar los patrones para la salida, porque el algoritmo y la regla de modificación de las conexiones producen patrones de salida consistentes. Cuando la red procesa patrones con un alto grado de similitud produce la misma salida para ambos, quiere decir que clasifica los patrones en categorías de patrones parecidos, se debe de decidir previamente en qué grado deberán parecerse. Normalmente se usa el error cuadrático medio para determinar la similitud. En general los tipos de métodos de aprendizaje no supervisados usan presentaciones modélicas de los objetos a reconocer y clasificar.[\[9\]](#)

En el no supervisado (o auto organizado) no existe ningún maestro externo que indique si la red neuronal está operando correcta o incorrectamente, pues no se dispone de ninguna salida objetivo hacia la cual la red neuronal deba tender. Así, durante el proceso de aprendizaje la red auto organizado debe descubrir por sí misma rasgos comunes, regularidades, correlaciones o categorías en los datos de entrada, e incorporarlos a su estructura interna de conexiones. Se dice, por tanto, que las neuronas deben auto organizarse en función de los estímulos (datos) procedentes del exterior. Existen diferentes tipos de aprendizaje no supervisados entre los que podemos distinguir el aprendizaje por componentes principales y el aprendizaje competitivo.

**Aprendizaje por componentes principales:** Consiste en hallar características principales a componentes que son comunes a muchos patrones de entrada para ello un pequeño número de neuronas coopera en la representación del patrón de entrada.[\[9\]](#)

**Aprendizaje competitivo y cooperativo:** En este tipo de aprendizaje las neuronas pugnan entre sí para representar a una clase o patrón de entrada. La neurona que ha sido seleccionada es aquella cuyos pesos incidentes se asemejan más al patrón de entrada. La base del aprendizaje es reforzar las conexiones de la unidad ganadora y debilitar las otras, y con esto lograr que los pesos de la unidad ganadora se parezcan cada vez más al patrón de entrada. En el aprendizaje competitivo las neuronas compiten unas con otras con el fin de llevar a cabo una tarea dada. Con este tipo de aprendizaje, se pretende que al presentársele a la red un patrón de entrada, sólo una de las neuronas de salida (o un grupo de vecinas) se active. Por tanto, las neuronas compiten por activarse, quedando finalmente una como neurona vencedora y anulada el resto, que son forzadas a sus valores de respuesta mínimos. El objetivo de este aprendizaje es categorizar (clusterizar) los datos que se introducen en la red. De esta forma, las informaciones similares son clasificadas formando parte de la misma categoría y, por tanto, deben activar la misma neurona de salida. Las clases o categorías deben ser creadas por la propia red, puesto que se trata de un aprendizaje no supervisado, a través de las correlaciones entre los datos de entrada.[\[9\]](#)

## **Aprendizaje de refuerzo**

Este aprendizaje es una variante del aprendizaje supervisado al cual se le informa a la red una crítica de corrección de salida y no la respuesta correcta en sí. Se basa en la idea de no disponer de un ejemplo completo del comportamiento deseado, es decir, de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada.

Se emplea información sobre el error cometido, pero existe una única señal de error, que representa un índice global del rendimiento de la red. No se suministra explícitamente la salida deseada. En ocasiones se denomina aprendizaje por premio-castigo.

En el proceso de entrenamiento es importante distinguir entre el nivel de error alcanzado al final de la fase de aprendizaje y el error que la red entrenada comete ante patrones no utilizados.[\[9\]](#)



## Híbrido

Es un aprendizaje que se caracteriza por hacer una combinación de los aprendizajes supervisados y los no supervisados. Este aprendizaje puede ser usado por las redes neuronales para hacer una disminución del tiempo computacional que puede ser alcanzado con uno solo de los existentes.[\[9\]](#)

### 1.8.3.4 Entrenamiento de la red

A la hora de hacer el entrenamiento, se le debe asignar a la red pesos aleatorios que serán ajustados para poder alcanzar un estado que permita generar su comportamiento ante patrones desconocidos.

Para ello se hace una comparación de los resultados calculados por la red con los valores de salida proporcionados en la fase de entrenamiento para los diferentes conjuntos de entrada; se trata de minimizar el error cometido ajustando los pesos de las conexiones. Esto se va a ir realizando iterativamente hasta que cumpla con la condición de parada especificada por el programador de la red.

A medida que se va entrenando la red esta aprende a reconocer rasgos generales que permitan dar una buena respuesta a los patrones no introducidos, esto es de gran importancia ya que permite que la red tenga un buen desempeño, cuando la red puede dar este tipo de respuestas se puede decir que ha aprendido y tiene la capacidad de ejecutar acciones dependiendo del entorno en donde esta se encuentre.[\[9\]](#)

### 1.8.3.5 Cálculo del error

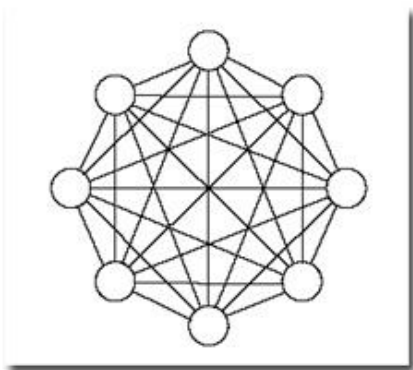
Se necesita calcular el error cometido durante el proceso de entrenamiento, existen varias ecuaciones para esta función entre las que se encuentran: el error cuadrado medio, la suma de los errores al cuadrado y la raíz del error cuadrado medio.[\[9\]](#)

## 1.8.4 Redes con conexiones hacia adelante (Feedforward)

El tipo más común de Red Neuronal Artificial es la red Feedforward, en la que cada capa de neuronas está conectada con la capa siguiente. Cada unidad en una sola capa puede estar completamente conectada a cada unidad en la siguiente capa. Hay varios tipos de redes Feedforward, el Perceptrón Monocapa (Layer Perceptron), el Perceptrón Multicapa (Multi-Layer Perceptron), entre otros.

## Perceptrón Monocapa

En este tipo de red las conexiones que se establecen son laterales y ocurren entre neuronas que pertenecen a la única capa de neuronas de la red. Puede haber conexiones auto-recurrentes. Las salidas de la red tendrán correspondencia con el tipo de función de transferencia utilizada. Además tienen una utilización básicamente en tareas relacionadas con la auto-asociación, es decir, para regenerar informaciones de entrada que se van a presentar distorsionadas o incompletas. [\[10\]](#)



**Fig.21** *Arquitectura de una red Perceptrón Monocapa*

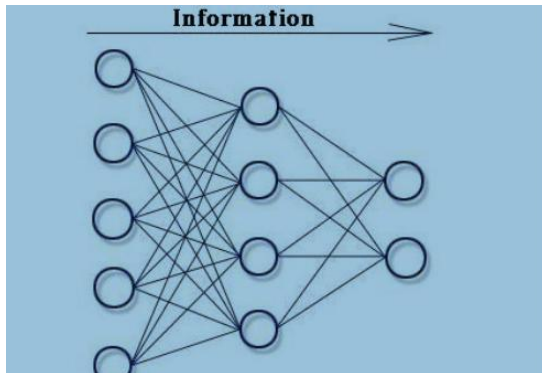
## El Perceptrón Multicapa

Este tipo de red es uno de los más comunes. Tiene un basamento en una red perceptrón simple con la diferencia de que la cantidad de capas intermedias u ocultas puede ser mayor o igual que una. Esta red es de tipo unidireccional y contiene una o más capas entre las capas de entrada y salida. [\[10\]](#)

Un Perceptrón Multicapa requiere de no más de tres capas para poder generar regiones de decisiones de cualquier complejidad. Por estar formada por varias capas permite resolver problemas que no sean linealmente separables. Como método de aprendizaje usan el supervisado.

En este tipo de red las unidades de procesamiento en capas se interconectan, las neuronas de cada capa no presentan conexiones entre sí. Cada neurona que se encuentre en una capa le dará una entrada a cada una de la siguiente capa, es decir, que cada neurona transmitirá una señal de salida a cada una de las neuronas de la capa siguiente. [\[1\]](#)

Tienen gran importancia dada a su capacidad de auto-adaptar los pesos de las neuronas de las capas intermedias para poder así aprender de la relación que existe entre un conjunto de patrones de entrada y salida. Presentan una gran capacidad de dar salidas satisfactorias a entradas que el sistema no ha visto en su fase de entrenamiento.



**Fig. 22** *Arquitectura de una red Perceptrón Multicapa.*

En general, hay una capa de entrada, una capa de salida y un número (o ninguna) de capas ocultas. El número de neuronas en las capas de entrada y de salida se determinará a partir de la aplicación específica del dominio del problema. Para la o las capas ocultas no existen cifras definitivas a calcular de un problema. A menudo, el ensayo y error se utiliza para encontrar un buen número. Todas las neuronas de la capa de entrada están conectadas a todas las neuronas de la capa oculta a través de los bordes. Del mismo modo, todas las neuronas de la capa oculta se conectan con todas las neuronas de la capa de salida. Se asocia un peso con cada borde. Normalmente, estos pesos se inicializan aleatoriamente dentro de un rango específico, dependiendo de la aplicación en particular.[\[4\]](#)

### **1.8.5 Redes con conexiones hacia adelante/hacia atrás (Feedforward/Feedback)**

La información que va a circular en este tipo de red será tanto hacia adelante como hacia atrás durante el funcionamiento de la red, es decir, que las conexiones que tendrá la red es de tipo feedforward tanto como feedback.

En general estas suelen ser bicapas, por lo tanto existen dos conjuntos de pesos: los que le corresponden a las redes con conexiones hacia adelante, de la capa de entrada que va hasta la capa de salida, así como las conexiones hacia atrás que serán desde la capa de salida hacia la capa de entrada. El valor de estos pesos de este tipo de conexiones no debe de coincidir, ya que serán diferentes en la mayoría de los casos. Entre los modelos de este tipo de red se encuentran la **Red de Hopfield y los Mapas de Kohonen**.

## **Red de Hopfield**

En una red de Hopfield, cada unidad está conectada a todas las demás unidades y no hay diferenciación entre las unidades de entrada y de salida. Las redes de Hopfield son utilizados para el almacenamiento y el reconocimiento de patrones, tales como imágenes y mapas. Son redes de adaptación probabilística, recurrentes y completamente conectadas. Funcionan como una memoria asociativa no lineal.[\[8\]](#)

## **Mapas de Kohonen**

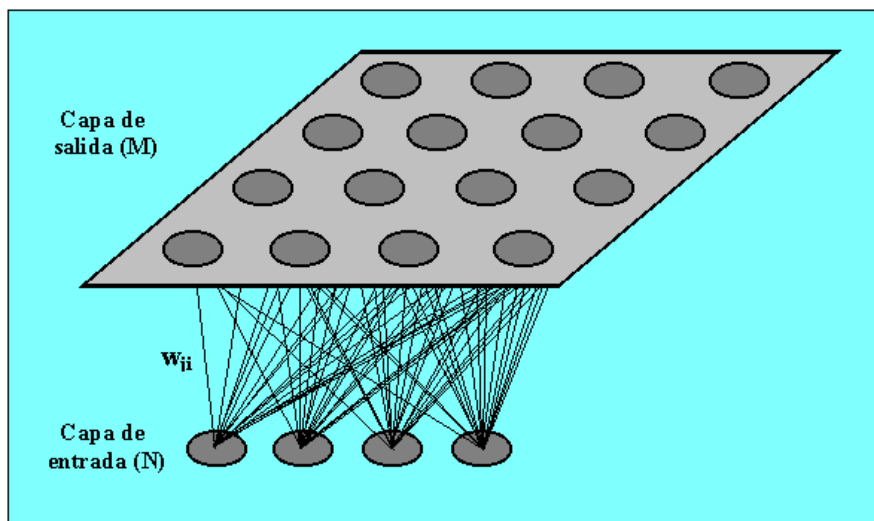
Estos mapas se denominan mapas auto-organizados o SOM (Self-Organizing Maps). Fueron desarrollados por Teuvo Kohonen a partir de 1989, se basan en las redes neuronales para realizar un análisis y categorización automática del contenido semántico de documentos textuales y basados en ciertas evidencias descubiertas a nivel cerebral y con un gran potencial de aplicabilidad práctica. El resultado gráfico de este análisis es un mapa 2D de categorías en las que cada categoría ocupa un espacio proporcional a las frecuencias de sus componentes. Los patrones más frecuentes ocupan un espacio mayor a expensas de los menos habituales. El algoritmo toma un conjunto N dimensional de objetos como entrada y entrena una red neuronal que converge finalmente a la forma de un mapa 2D. Los mapas de Kohonen se encuentran entre los modelos más realistas del funcionamiento cerebral. Entre sus objetivos esta hacer más digerible la representación de grandes conjuntos de información textual. Una función mapa de Kohonen es un tipo de auto organización de mapa en el que sólo hay una capa con las unidades de entradas, que se organizan de acuerdo a los valores de entrada. Los mapas de Kohonen pueden ser utilizados para detectar patrones en los complejos conjuntos de datos.[\[3\]](#)

Este tipo de red neuronal, mediante un aprendizaje no supervisado, puede ser de gran utilidad en el campo del análisis exploratorio de datos, debido a que son sistemas capaces de realizar análisis de

clúster, representar densidades de probabilidad y proyectar un espacio de alta dimensión sobre otro de dimensión mucho menor.[\[3\]](#)

## Arquitectura

Un modelo SOM está compuesto por dos capas de neuronas. La capa de entrada (formada por  $N$  neuronas, una por cada variable de entrada) se encarga de recibir y transmitir a la capa de salida la información procedente del exterior. La capa de salida (formada por  $M$  neuronas) es la encargada de procesar la información y formar el mapa de rasgos. Normalmente, las neuronas de la capa de salida se organizan en forma de mapa bidimensional como se muestra en la **Fig.3**, aunque a veces también se utilizan capas de una sola dimensión (cadena lineal de neuronas) o de tres dimensiones.[\[7\]](#)



**Fig. 23** Paralelepípedo.

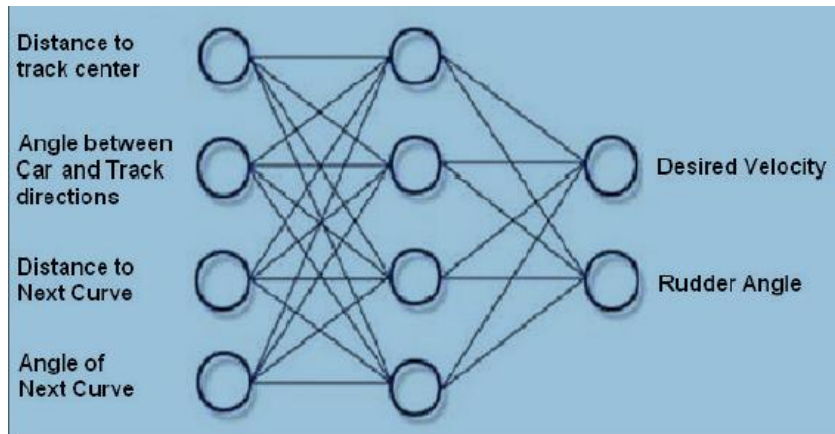
## 1.8.6 Aplicaciones de las Redes Neuronales Artificiales en videojuegos

Las redes neuronales artificiales pueden ser de diferentes topologías y modelos de aprendizaje, esto depende del uso que se les dé. Uno de sus usos más comunes incluye el reconocimiento de patrones, el aprendizaje, la memoria y la predicción.[\[1\]](#)

Las redes neuronales artificiales se utilizan para una serie de aplicaciones comerciales que incluyen una diversidad de industrias, incluyendo el comercio, la alimentación, las finanzas, la medicina, la ciencia y la ingeniería. Estas pueden ser aplicadas a diferentes campos como son el control de eficiencia de máquinas, el reconocimiento de firmas, predicciones en el tiempo, monitoreo, mercadotecnia, diagnóstico, optimización y en la valoración de efectos sísmicos.[\[1\]](#)

La industria de juegos de ordenador no es diferente de las industrias antes mencionadas en términos de la variedad de aplicaciones de las Redes Neuronales Artificiales. Una red neuronal se puede utilizar para tomar decisiones o interpretar los datos basados en entradas y salidas que se dan en el transcurso del videojuego. La salida de la red puede ser una decisión, una clasificación o una predicción. Algunos ejemplos de videojuegos que incluyen Redes Neuronales Artificiales para diversas tareas son: 3000AD (BC3K) donde se utilizan redes neuronales para controlar los NPCs así como para guiar negociaciones, comercio, y combate. BC3K las usa para toma de decisiones orientadas a objetivos básicos y pathfinding, con una combinación de aprendizaje supervisado y no supervisado. El juego Dirt Track Racing usa una red neuronal para manejar alrededor de la pista. Por último Heavy Gear las usa para la parte del control de mecanismos Mech, cada Mech usa redes especializadas para aspectos particulares. [\[11\]](#)

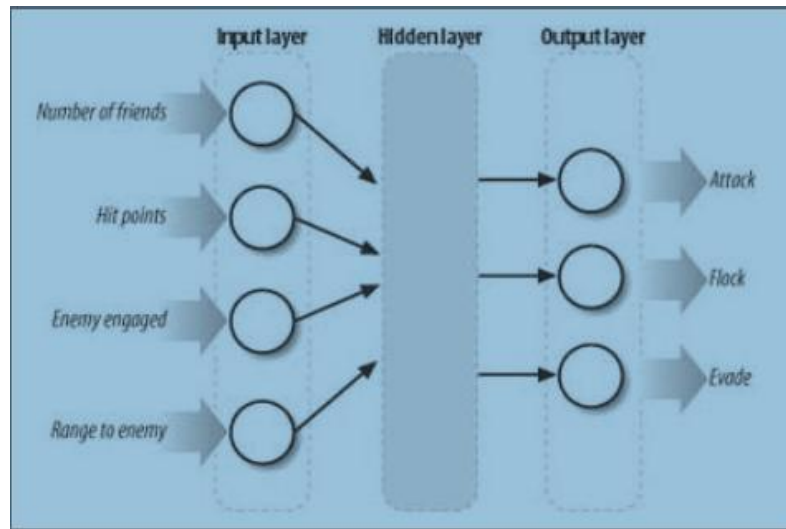
Otro ejemplo de la puesta en práctica de las Redes Neuronales Artificiales es en el videojuego "Rápido y Curioso" realizado en La Universidad de Ciencias Informáticas en la facultad # 5, el cual consistía en controlar un coche de carreras.[\[1\]](#)



**Fig. 24** Modelo de Red Neuronal utilizado en el videojuego "Rápido y Curioso".

De acuerdo con la interfaz del videojuego, los programadores de Inteligencia Artificial tenían acceso a la información que se guardó en 4 variables de entrada: la "distancia al centro de la carretera", el "ángulo entre el coche y direcciones tema", la "distancia a la curva siguiente" y el "ángulo de la curva siguiente" la información almacenada en estas variables servían para saber cuando el coche estaba en peligro de salir de la pista, cuando el coche conducía en el camino equivocado, cuando había una curva peligrosa cerca y así sucesivamente. Usando esos valores, se calculaba el ángulo del "timón" y la "velocidad deseada del coche".[1]

Un ejemplo de cómo utilizar una Red Neuronal Artificial es en un videojuego de Rol, para manejar un agente en el proceso de toma de decisiones (es decir, si va a atacar, evadir o pasear, dependiendo si un enemigo está en las proximidades o no). [1]



**Fig. 25** Ataque o Huida

Otra forma de utilizarse es en la Evaluación de la Amenaza. Esta neurona artificial se utiliza en una simulación de la estrategia para dar al ejército controlado algunos de los medios para predecir el tipo de amenaza presentada por el jugador en cualquier momento durante el juego, por lo que podría estar preparado para defenderse. [1]

### 1.8.7 Ventajas y desventajas de las Redes Neuronales Artificiales

Las redes neuronales son una máquina de aprender inspiradas en el cerebro humano. Esta técnica se utiliza ampliamente para aplicaciones tanto industriales como científicas, pero en ocasiones no son flexibles y suficientes para determinados problemas. A continuación se describen algunas ventajas y desventajas que poseen.

#### **Ventajas [1]**

- Paralelización:** Fácil de paralelizar, ya que cada neurona puede trabajar de forma independiente.
- Capacidad de aprendizaje:** Las redes neuronales pueden aprender mediante el ajuste de sus pesos.
- Robustez:** Las redes neuronales artificiales pueden tratar con cierta cantidad de ruido en la entrada. Incluso si parte de una red neuronal está dañada (similar al daño cerebral parcial), puede aún realizar tareas a determinados dominios.



**-Generalización:** Una red neuronal artificial puede hacer frente a los nuevos patrones que son similares a los adquiridos.

**-No linealidad:** Problemas no lineales son difíciles de resolver matemáticamente. Las redes neuronales artificiales pueden hacer frente a cualquier problema que se puede representar con patrones.

## **Desventajas**

-Las Redes Neuronales Artificiales no han sido capaces de imitar realmente el cerebro humano o la inteligencia.

-No se pueden extraer las normas subyacentes que puedan deducirse de la red neuronal después de haber sido entrenadas.

-La complejidad computacional toma a menudo mucho tiempo, y a veces ni siquiera convergen. Pero una vez que se tiene éxito, puede ser copiado a otros sistemas con facilidad y el beneficio puede ser significativo.

-La escala de una red neuronal artificial no es una cuestión simple. Cuando queremos ampliar o reducir el número de neuronas, tenemos otra red, así que tenemos que formar la nueva.

## **1.9 Bibliotecas para el trabajo con Redes Neuronales Artificiales**

Para el trabajo con las redes neuronales artificiales se han creado bibliotecas que facilitan la aplicación de las redes en diferentes campos de la ciencia. Entre las bibliotecas más usadas se encuentran ***Libperceptronnetwork***, ***LTI-Lib***, ***DLIB*** y ***FLOOD***.

***Libperceptronnetwork***: Es una biblioteca que modela una red de múltiples capas del perceptrón del feedforward, un tipo bien conocido y entendido de red neuronal. Puede ser utilizada para una gran cantidad de tareas de reconocimiento y de clasificación. Además de proporcionar los algoritmos estándares de propagación y del backpropagation, la biblioteca también proporciona variaciones al algoritmo del backpropagation, tales como métodos opcionales del peso-decaimiento, del término-ímpetu y de la óptimo-tolerancia. Este tipo de red tiene como desventaja que se limita matemáticamente en sus capacidades.

**LTI-Lib:** Es una biblioteca orientada objeto con algoritmos y estructuras de datos usados con frecuencia en el procesamiento de imágenes y visión de computadora. La meta principal del LTI-Lib es proporcionar una biblioteca orientada objeto en C++, que simplifica el mantenimiento de código compartido, proporcionando los algoritmos rápidos que se pueden utilizar en aplicaciones reales. Es fácil de utilizar debido a la especificación de una programación constante de interfaz para todas las clases.

**DLIB:** Es una biblioteca de C++. Proporciona una abstracción de la plataforma para tareas comunes tales como interconexión con servicios de red o crear interfaces utilizando gráficos. Pone muchos algoritmos útiles en ejecución tales como listas encadenadas, árboles de búsqueda binaria, álgebra lineal y utilidades de la matriz.

**FLOOD:** Es una biblioteca abierta de redes neuronales artificiales en C++ para modelar datos. Soluciona problemas tales como la regresión de la función y el reconocimiento de patrones.

## 1.10 Conclusiones parciales

En el presente capítulo se abordó el tema de los videojuegos, un mundo muy amplio de tecnología y fantasía que se va desarrollando con técnicas avanzadas de Inteligencia Artificial, como son las Redes Neuronales Artificiales. Los videojuegos que se han llevado el protagonismo en el empleo de esta técnica han sido los de estrategias y combates. Algunos de los videojuegos que reflejaron el uso de las Redes Neuronales Artificiales fueron Battle-Cruiser, 3000AD, Black & White, Creatures, Dirt Track Racing, y Heavy Gear. Con los conocimientos adquiridos a lo largo del capítulo se podrá seleccionar el modelo de red neuronal artificial y definir las capas que tendrá el Mapa de Influencia para realizar la toma de decisiones en un videojuego de estrategia.

# CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

---

## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS REDES NEURONALES ARTIFICIALES Y MAPAS DE INFLUENCIA

### 2.1 Introducción

El presente capítulo propone un modelo de red neuronal artificial como solución en la toma de decisiones y los pasos a tener en cuenta a la hora de diseñar una red. Además se explica detalladamente el funcionamiento de los Mapas de Influencias y la forma de combinar estos con una Red Neuronal Artificial.

### 2.2 Funcionamiento de los Mapas de Influencia

Cuando se genera un mapa de influencia, primero son calculados los valores iniciales para cada celda en cada capa del mapa, basado en el estado actual del juego. es decir, si una capa de datos es la fuerza de combate, entonces la fuerza de combate de todas las unidades tendría que ser calculado para cada celda. Una vez que estos valores iniciales se calculan, los valores en cada celda necesitan ser reproducido a las celdas cercanas, lo que crea una influencia en cada celda. Esta propagación de influencia da una imagen mucho más precisa de la situación estratégica actual, ya que no sólo se muestran las unidades militares en un videojuego, sino también lo que están haciendo, lo que pueden hacer y las áreas que potencialmente influyen. [8]

Después que los valores iniciales se calculan para cada capa y las influencias son propagadas, los datos pueden ser preparados para su utilización en la evaluación estratégica.

En la **Fig.23** se muestra un mapa de influencia, el ejemplo ilustra la propagación de proceso. La cuadrícula en la parte superior izquierda muestra una situación del juego posible en el que los tanques de los dos jugadores se colocan en celdas diferentes. En este ejemplo, cada depósito de luz del jugador será una influencia positiva a esa celda mientras que cada tanque para el jugador oscuro proporcionará una influencia de uno negativo. La cuadrícula en la parte superior derecha muestra cómo cada uno de estos tanques se extiende o propaga su influencia a las celdas cercanas. Sólo dos tanques se muestran para mayor claridad.[8]

En este ejemplo, cada valor de la influencia se propaga un solo espacio en cada dirección y la influencia se redujo a la mitad cuando se propagó. El resultado de esta multiplicación para cada celda se muestra en

## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

la cuadrícula en la parte inferior izquierda. Las celdas con un valor positivo se ven influidas por el reproductor de la luz y las celdas con los valores negativos son influenciadas por el jugador oscuro. La parrilla del final en la parte inferior derecha muestra la misma influencia de valores como la red de la izquierda, pero la influencia se muestra en escala de grises.[8]

Los mapas de influencia pueden ser utilizados para la evaluación estratégica y la toma de decisiones, ya que su estructura permite hacer inferencias acerca de las características inteligentes de diferentes lugares en el medio ambiente. Por ejemplo, áreas que presentan altas estrategias de control pueden ser identificados, así como los puntos débiles en las defensas de un adversario, el enemigo frente, costados y parte trasera, lugares de camping, áreas estratégicamente vulnerable, choque, puntos en el terreno, entre otras características significativas que los jugadores humanos deben elegir a través de la experiencia o la intuición.[8]

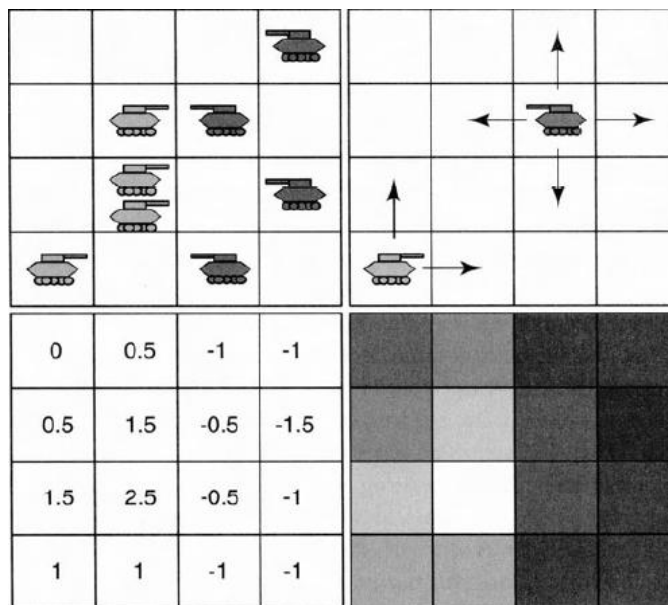


Fig. 26 Ejemplo de un Mapa de Influencia.

### 2.3 Decisiones de diseño de una Red Neuronal Artificial

Existen muchas opciones para diseñar una Red Neuronal Artificial. En primer lugar, hay que decidir sobre la arquitectura de la red a utilizar, el tipo y su función de activación. Luego se debe calcular el número de

## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

---

unidades de entrada que será igual al número de variables del juego que están alimentando a la red, el cual debe ser lo más pequeño posible, para reducir la complejidad del problema. El número de unidades de salida será igual al número de los diferentes resultados.[\[8\]](#)

El número de nodos ocultos es arbitrario y el enfoque habitual es comenzar con unos pocos y añadir más dependiendo del rendimiento de la red. Después se necesita decidir qué tipo de funciones de activación se utilizarán, como lineal, escalón o sigmoid.[\[8\]](#)

En tercer lugar, tiene que elegir la forma de representar las entradas y salidas de la red; si van a ser números reales o de una cadena de bits. En una situación de juego, donde las variables son números reales, por lo general, sería mejor para representar las entradas a la red.[\[8\]](#)

Los resultados podrían asignar directamente una decisión o acción que se necesite o puede representar una clasificación intermedia. En estos casos, la salida podría ser representada como enteros que corresponden a las diferentes opciones o cada salida podría corresponder a una variable, de modo que cada una devuelve un valor entre 0 y 1. [\[8\]](#)

En cuarto lugar, utilizar una regla de aprendizaje adecuado a las necesidades de la decisión a tomar. Luego decidir sobre los datos de entrenamiento que utilizará para formar la red. [\[8\]](#)

Y por último, la red necesitará que se le realicen pruebas para encontrar un punto adecuado para detener la formación, donde la precisión y la capacidad de generalizar los nuevos datos se maximizan. Esto se basa generalmente en una medida de error, la más usada es la del error cuadrado medio (MSE).[\[8\]](#)

### 2.4 Modelo de red seleccionado

A partir del estudio realizado en el Capítulo 1 sobre los modelos de redes neuronales artificiales existentes, para la propuesta solución se decidió utilizar el Perceptrón Multicapa (MLP por sus siglas en inglés). El MLP es una de las redes neuronales artificiales más usadas, debido a su gran capacidad para resolver problemas de mayor complejidad, fundamentalmente de clasificación, en los que las variables no son linealmente separables. Esto hace que facilite su aplicación en distintos escenarios dentro de los videojuegos.

## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

---

Contrario a las redes lineales, no se le hace necesario a este tipo de red conocer con anticipación el valor de la entrada. El MLP es capaz de trabajar como un aproximador universal de funciones. Esta red es de tipo unidireccional y contiene una o más capas entre la capa de entrada y la capa de salida. Por estar formada por varias capas permite resolver problemas que no sean linealmente separables.

Una de las principales ventajas que presentan este tipo de redes es su capacidad para generalizar, esto les permite hacer un reconocimiento de patrones en el conjunto de entrenamiento. Cuando la red ya ha sido entrenada y se le presenta una nueva situación, esta podrá reaccionar satisfactoriamente, a pesar de que la misma no haya sido entrenada para ello. Otra de sus grandes ventajas es su rapidez en el tiempo de respuesta, esta condición la ratifica como candidata principal para ser usada en los videojuegos, donde la realidad y eficiencia es el factor fundamental. Puede generar regiones de decisiones de cualquier complejidad.

Tienen como principal desventaja a la hora de usarla en escenarios de realidad virtual su costo computacional, pero es posible obtener topologías de perceptrones que sean capaces de ejecutarse en tiempo real. A este tipo de red la componen varias capas de perceptrones interconectados entre sí: una capa de entrada con tantas neuronas como elementos tenga el vector de información que va a ser introducido, una o más capas ocultas que realizarán la mayor parte del procesamiento de los datos y por último una capa de salida que muestre los resultados finales. Los pesos sinápticos que irán adjuntos a las conexiones que se establecerán entre los perceptrones de capas adyacentes van a ser el conocimiento que tendrá la red.

El Perceptrón es el elemento más pequeño que contiene el MLP.[\[7\]](#) Este Procesa un conjunto de información para convertirla en una señal de salida. Está compuesto por un conjunto de variables que dan paso a una solución o decisión a tomar. Dentro de estas variables se encuentra el valor umbral que tiene como objetivo brindarle a la red una mayor flexibilidad en el proceso de aprendizaje.

## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

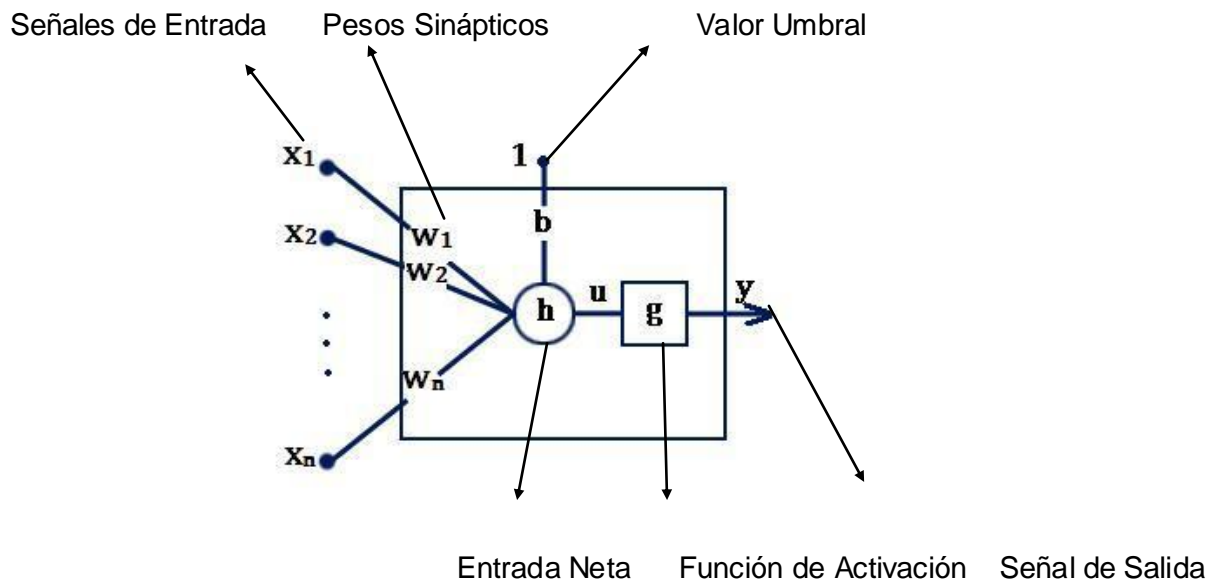


Fig. 27 Estructura de la unidad básica del MLP.

En la **Ecuación 1** se muestra como se calcula la función de propagación.

$$h(x; b, w) = b + \sum_{i=1}^n w_i x_i$$

### Ecuación 1

Para determinar la señal de salida o grado de excitación de la neurona artificial se propone utilizar las siguientes funciones de activación Sigmoid y Lineal.

### Función Sigmoid

Esta es una función no-lineal utilizada en la capa oculta. Posee una derivada continua, fácil de evaluar y de calcular. Cuando da una salida cercana al valor mínimo significa que la neurona no se encuentra activada y si la salida es cercana al valor máximo entonces la neurona está activada. [7]

# CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

---

$$f(x) = \frac{1}{(1 + e^{-x/c})}$$

**Ecuación 4**

## **Función Lineal**

La función Lineal se utilizó en la capa de salida, donde es usada con frecuencia. Esta función posee una salida similar a la entrada neta de la neurona.

$$f(x) = x$$

**Ecuación 5**

### **2.4.1 Proceso de entrenamiento de la red**

Uno de los métodos utilizados para entrenar redes multicapas es el método del gradiente descendente, pues es muy usado ya que posee gran eficiencia computacional. El algoritmo de aprendizaje por propagación del error hacia atrás (back-propagation en inglés) brinda una variante de solución para dicho método, siendo este uno de los más utilizados para entrenar redes neuronales artificiales [\[7\]](#)

En este algoritmo los pesos de las neuronas de cada una de las capas serán ajustados a partir del error cometido en la siguiente capa y será multiplicado por la derivada de la función de activación usada; por tal motivo es de gran importancia que la función de activación posea facilidades a la hora de evaluarse y derivarse.

El back-propagation consta de dos pasos principales: Una corrida hacia adelante (feed-forward en inglés) donde se le aplica un patrón de entrada a la red con sus pesos actuales. Las salidas de las unidades que se encuentren en cada nivel se van a calcular empezando por la capa de entrada y trabajando hacia adelante en dirección a la capa de salida. Se hace una comparación de la salida real de la red con la salida deseada y se hace un cálculo del error. Luego se ejecuta una pasada hacia atrás en la que se hace una propagación de la derivada del error a través de la red y así todos los pesos son ajustados en proporción a su responsabilidad en el error de la salida.



## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

---

### Pasos del algoritmo back-propagation:

- I. Inicializar todos los pesos ( $W_{ij}$ ) con valores aleatorios pequeños (es usual seleccionar números aleatorios en el rango de -0.1 a 0.1).
- II. Presentar una entrada como patrón y especificar la salida deseada.
- III. Calcular las entradas netas de todas las neuronas.
- IV. Calcular las salidas reales de todas las unidades con los valores de las entradas netas usando la función de activación correspondiente.
- V. Encontrar el error para todas las neuronas.  $D_j$  es el valor de la salida deseado y  $Y_j$  es el valor de salida real para la  $j$ -ésima neurona, los errores se calculan según la capa:

- 1 Para las neuronas de salida:

$$E_j = (Y_j - D_j) * (1 - Y_j)$$

#### Ecuación 6

- 2 Para las neuronas en las capas ocultas:

$$E_j = Y_j * (1 - Y_j) * \sum Z_k * W_{kj}$$

#### Ecuación 7

Donde  $k$  recorre todas las neuronas de la capa siguiente conectadas a la neurona  $j$  y  $Z_k$  representa los errores calculados para las salidas de estas neuronas en dependencia de la capa en que se encuentren.

- VI. Ajustar los pesos usando la ecuación

$$W_{ij}(n+1) = W_{ij}(n) + t * E_j * Y_i + h * (W_{ij}(n) - W_{ij}(n-1))$$

#### Ecuación 8

Donde  $W_{ij}(n+1)$ ,  $W_{ij}(n)$  y  $W_{ij}(n-1)$  indican el peso nuevo, presente y anterior respectivamente,  $t$  es un número real que denota la velocidad de aprendizaje y  $h$  determina el efecto de los cambios de pesos

## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

---

previos sobre la dirección actual de movimiento en el espacio de los pesos (su valor es constante y está entre 0 y 1).

VII. Presentar otra entrada e ir nuevamente al *paso II*.

Este procedimiento se repite hasta que el error tenga un valor cercano a 0. Si el número de ejemplos excede la cantidad de conexiones de la red, puede no ser posible hacer 0 el error. [7]

### 2.4.1.1 Cálculo del error

#### Error cuadrado medio

La función del error cuadrado medio hace un cálculo del error cometido por la red para un conjunto de datos de entrenamiento. En esta ecuación hay varios parámetros que deben ser fijados, una tasa de aprendizaje, el impulso y los valores de peso inicial. La tasa de aprendizaje generalmente se configura en torno a 0.2, el impulso se establece en 0.9 y los pesos iniciales son elegidos al azar dentro de un rango, entre 1 y -1. Estas variables a menudo necesitan ser sincronizadas para lograr un rendimiento óptimo.

$$\varepsilon = \frac{\sum(nc - nd)^2}{m}$$

#### Ecuación 9

Donde ***nc*** representa la salida real, ***nd*** la salida que se desea para cada conjunto de datos de entrenamiento y ***m*** es la cantidad de neuronas de la capa de salida. A la hora de ajustar los pesos sinápticos de las conexiones, es necesario conocer cuánto influye cada neurona en el error total cometido por la red; por lo que se hace un cálculo del error asociado a cada una de estas neuronas en las capas de salida y en la capa oculta.

#### Cálculo del error de las neuronas de la capa de salida:

$$\Omega_i^s = \Delta n_i^s * f(n_i^s)$$

#### Ecuación 10

## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

---

Donde  $\Omega_i^s$  representa el error cometido por la neurona i-ésima,  $\Delta n_i^s$  es la diferencia que existe entre la salida real y la salida deseada, y  $f(nc_i^s)$  es la derivada de la función de activación evaluada en el valor de la neurona i-ésima.

**Cálculo del error de las neuronas de la capa oculta:**

$$\Omega_i^o = \left( \sum \Omega_i^s * w_{ij} \right) * f(nc_i^o)$$

**Ecuación 11**

Donde  $\Omega_i^o$  es el error cometido por cada neurona en la capa oculta. Esta ecuación es una función asociada con el error ( $\Omega_i^s$ ) cometido en la capa de salida a la cual se van a conectar estas neuronas multiplicadas por los pesos de cada conexión, por la derivada de la función de activación utilizada, evaluada en el valor de la neurona en la capa oculta. Lo primero que se hará será calcular el error en la capa de salida y se propaga hacia la capa oculta ajustando los pesos de forma iterativa. Para hacer este cálculo del error es necesario tener la derivada de la función de activación, por lo que generalmente se utiliza la Sigmoid. Los pesos que se encuentran en la capa de entrada no se van a actualizar ya que sus valores son dados como parte del proceso de entrenamiento.

### 2.4.1.2 Ajuste de los pesos

Partiendo del error se puede hacer un cálculo de la variación que es necesaria hacer en los pesos actuales:

$$\Delta w = \rho * \Omega_i * nci$$

**Ecuación 12**

Donde  $\rho$  es la tasa de aprendizaje, un multiplicador que afecta la cantidad de ajuste de cada peso,  $\Omega_i$  el error cometido y  $nc_i$  el valor de la neurona i-ésima.

## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

---

La tasa de aprendizaje es un valor pequeño aleatorio que estará entre 0.1 y 0.9. Si se tiene un valor muy alto esto puede provocar que se rebase el peso óptimo, por lo que uno bajo, puede demorar el proceso de entrenamiento.

Para obtener un nuevo peso ( $W_{ac}$ ) se le adiciona la variación ( $\Delta W$ ) al peso de la iteración anterior ( $W_{vj}$ ).

$$W_{ac} = W_{vj} + \Delta W$$

### Ecuación 13

#### 2.5 Combinación de las Redes Neuronales Artificiales y los Mapas de Influencia

Cuando una decisión debe ser hecha por un jugador de un videojuego, algunas o todas las capas de un mapa de influencia se combinan a través de una suma ponderada proporcionando una idea global de la idoneidad de cada área en el mapa de la decisión actual. Sin embargo, la utilización de una suma ponderada tiene ciertas limitaciones. La presente investigación explica cómo una red neuronal artificial puede ser utilizada en lugar de una suma ponderada, para analizar los datos del mapa de influencia y tomar una decisión estratégica. [8]

A continuación se muestra el diseño de red utilizado en la solución propuesta. La red requiere una unidad de entrada correspondiente a cada celda en cada capa del mapa de influencia, lo que resulta en un número total de unidades de entrada igual al número de capas multiplicado por el número de las celdas, se utilizarán matrices representando las variables iniciales. Además, no tendrá que ser una unidad de salida para cada celda del mapa de influencia, de manera que cada celda recibirá un valor final de la conveniencia de la red, las neuronas de salida representan la decisión a tomar en un determinado estado del juego y la posición a la cual el jugador debe moverse al tomar la decisión. Para el número de unidades ocultas inicialmente se creará un número arbitrario, aproximadamente en el rango de 10 a 20, que tendrá que ajustarse para alcanzar un rendimiento óptimo.

## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

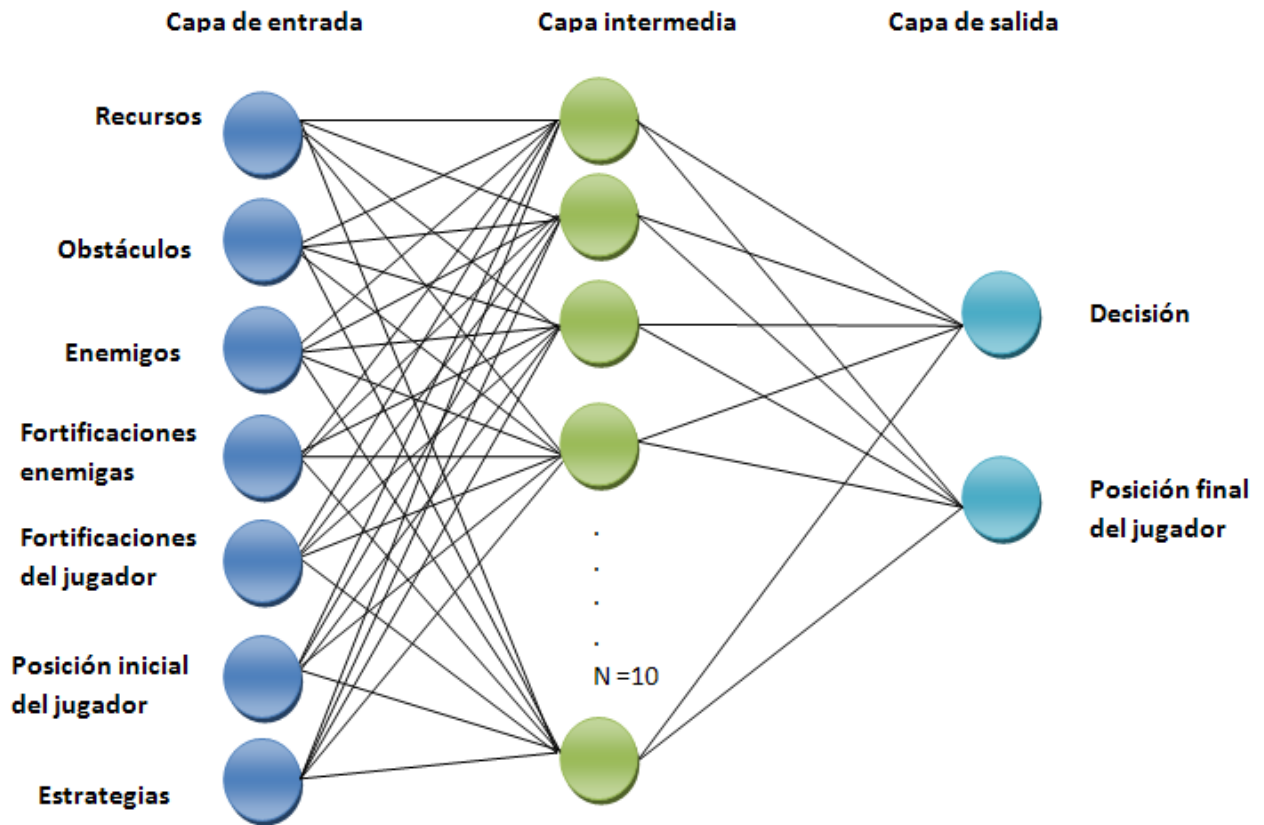


Fig. 17 Diseño de red propuesto

No tendrá que ser una red neuronal artificial para cada decisión que se debe hacer, ya que el conjunto de pesos estará dado en un rango entre -1 y 1, diferente para cada decisión. El diseño será el mismo, la única diferencia de las otras redes será la forma en que se entrenará. Como resultado tendrá un conjunto de diferentes pesos. Por lo tanto, una vez que el diseño de la red se ha completado, el mismo se puede utilizar para cada instancia de la red, o de cada decisión que se está haciendo. La mejor manera de implementar esto sería utilizar una red neuronal artificial única y un conjunto de neuronas de salida, una para la posición final del jugador, es decir, a la celda donde tiene que ubicarse y la segunda neurona guardará el rango de las decisiones a tomar por el jugador, de modo que la red neuronal artificial puede simplemente utilizar el conjunto de pesos que corresponde a la decisión actual que se está haciendo. Además, es posible tener más de un juego de pesos para cada decisión, cada conjunto de pesos es el

# CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

---

resultado de un conjunto de entrenamiento diferente y por lo tanto toma un enfoque diferente de la decisión. Este método podría ser utilizado para simular diferentes personalidades que el agente pueda adquirir, esto da la facilidad de responder de manera diferente en cada situación, en función de ciertos atributos, tales como la forma agresiva o defensiva del agente.

## 2.6 Selección de las variables

### Variables de entrada

Los elementos de entrada van a representar variables de importancia que puede tener un videojuego de estrategia. A continuación se hace una propuesta de algunas variables que pueden ser utilizadas, tener en cuenta que pueden establecerse cuantas variables se necesiten en el videojuego a crear.

- **Ubicación del enemigo:**

Analiza las posibles celdas donde puedan encontrarse las fuerzas enemigas y de esta forma evitar un encuentro no deseado.

- **Ubicación de obstáculos:**

Define las áreas donde resultará más problemático el avance de un específico personaje o equipo.

- **Ubicación de recursos:**

Define las áreas donde se encuentran recursos importantes para la batalla (comida, armas, municiones, etc.).

- **Fortificaciones enemigas:**

Define las áreas donde se encuentran ubicadas fuerzas enemigas.

- **Fortificaciones del jugador:**

Define las áreas donde se encuentran ubicadas fuerzas del jugador.

Las siguientes variables representan la posición que va a tener el agente en el momento de iniciado el juego y la estrategia a realizar será otra variable a tener en cuenta, estas forman parte del videojuego pero no se encuentran dentro del Mapa de Influencia.

## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

---

- **Posición inicial del jugador:**

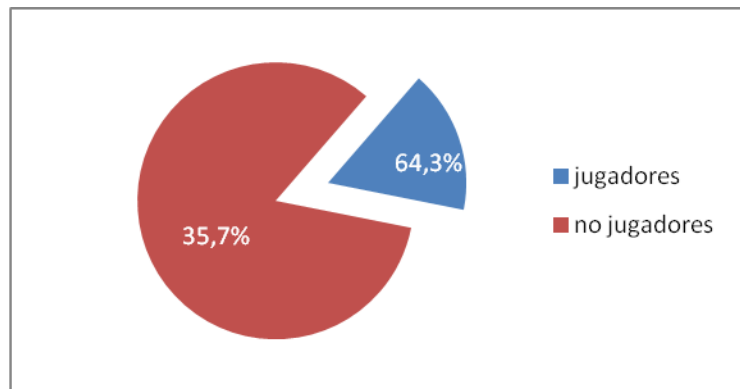
Esta será la posición inicial que ocupará el jugador al comenzar el videojuego.

- **Estrategia a realizar**

Una estrategia es una meta que el jugador desea satisfacer y son cumplidas a largo plazo mediante acciones o pasos a realizar.

### Variables de salida

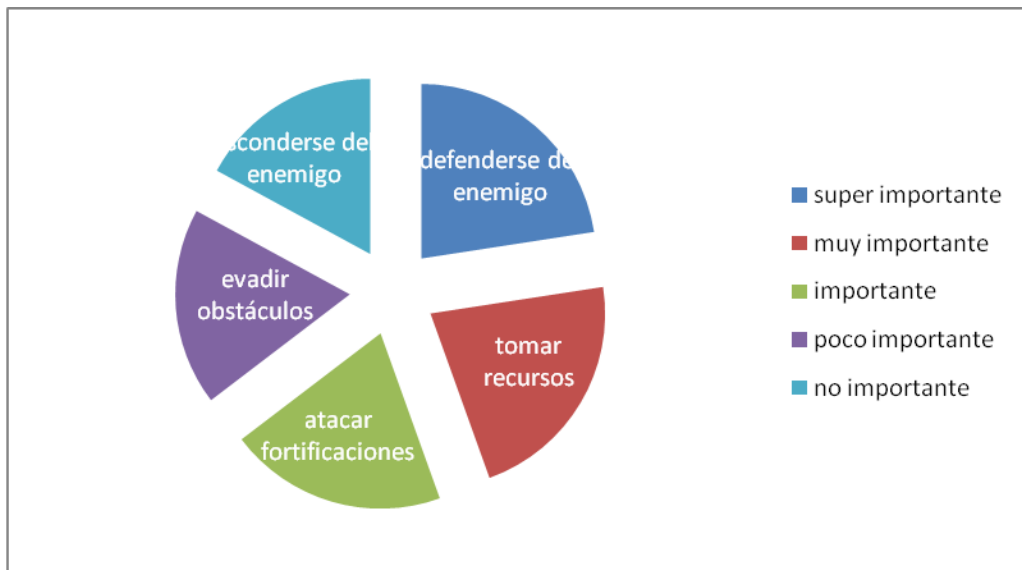
Se realizó una encuesta a personas expertas o no en el tema de los videojuegos de estrategia. La encuesta consta de dos preguntas y tiene como objetivo encontrar las estrategias de mayor importancia para el jugador en un estado del videojuego (*Ver Anexo1*). Los resultados arrojados se evidencian en las siguientes gráficas.



**Fig.29** Análisis de la Pregunta1

El 64,3% de los encuestados han jugado un videojuego de estrategia, de los que el 77,8% son hombres y el 22,2% son mujeres. Por lo anterior se pudo observar que el mundo del videojuego crece día a día. Con esta muestra se analizó el nivel de importancia que le conceden a las estrategias planteadas durante el desarrollo del juego.

## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA



**Figura 10.** Análisis de la pregunta 2

Según las respuestas obtenidas, se pudo apreciar que la estrategia más importante es la de defenderse del enemigo, con un total de 37 pts, luego le siguen obtener recursos con un total de 36 pts, atacar fortificaciones enemigas con 33 pts, evadir obstáculos con 30 pts y por último esconderse del enemigo con 28 pts.

A partir de la encuesta realizada se obtuvo un número de acciones para satisfacer las estrategias planteadas. Estas acciones serán las variables de salida de la red incluyendo la posición final del jugador.

### **Avanzar:**

Seguir adelante evitando obstáculos y/o el ataque sorpresivo y no deseado del enemigo.

### **Esconderse:**

Pasar inadvertido cuando las condiciones no sean favorables.

### **Atacar:**

Momento oportuno para tomar ventaja sobre el enemigo.



## CAPÍTULO 2: DESCRIPCIÓN Y APLICACIÓN DE LAS RNA Y MAPAS DE INFLUENCIA

---

### **Defenderse:**

Contraatacar ante el ataque enemigo.

### **Posición final del jugador:**

Posición que ocupará el jugador en una posición del mapa al ejecutar la acción que dará cumplimiento a una determinada estrategia.

### **2.7 Conclusiones parciales**

En el presente capítulo se realizó un análisis profundo sobre el modelo de red neuronal artificial propuesto, el Perceptrón Multicapa. Además se explicó de manera detallada el funcionamiento de los Mapas de Influencia y la combinación con la Red Neuronal Artificial. Se plantea una propuesta de los elementos de entrada y salida de la red, importantes a la hora de generar una decisión final. Los elementos antes mencionados, constituyen un paso fundamental para desarrollar el algoritmo que da solución al objetivo planteado.

### CAPITULO 3: RESULTADOS GENERALES

#### 3.1 Introducción

En la investigación además de la solución encontrada, mediante el uso de las Redes Neuronales Artificiales como técnica de Inteligencia Artificial aplicada a los Mapas de Influencia en la toma de decisiones estratégicas, se obtuvo un Demo para mostrar el resultado de la investigación.

#### 3.2 Descripción de la solución propuesta

La propuesta de la investigación es vincular los Mapas de Influencia utilizados en algunos videojuegos de estrategia con una técnica de Inteligencia Artificial para la toma de decisiones, dicha técnica es muy conocida y aplicada en el mundo del videojuego, las Redes Neuronales Artificiales.

En los epígrafes anteriores se describió las ventajas de utilizar un Perceptrón Multicapa, por lo que es el modelo propuesto en la investigación, en este modelo se calcularon una serie de funciones útiles para el entrenamiento de la red entre las que se encuentran la función de activación Sigmoid y Lineal, además del algoritmo de aprendizaje por propagación del error hacia atrás (Backpropagation), el cual es uno de los métodos más utilizados para entrenar redes neuronales artificiales y minimizar el error, este es una variante de solución para el método del gradiente descendente. La función para calcular el error cometido durante el proceso de entrenamiento nombrado error cuadrado medio y posee como criterio de parada el algoritmo Backpropagation.

A través de una encuesta realizada se obtuvieron las variables de salida que representan pasos a seguir para lograr una determinada estrategia. Se realizó la combinación de los Mapas de Influencia y las Redes Neuronales Artificiales logrando evitar la pérdida de información donde se hizo necesario el uso de una biblioteca para el trabajo con las redes neuronales artificiales, en este caso la biblioteca Flood.

##### 3.2.1 Esquema general de la solución

A continuación se propone un número de clases para el trabajo y vinculación de las Redes Neuronales Artificiales y los Mapas de Influencia.

**Mapa:** En esta clase se crean las matrices que conforman el Mapa de Influencia. Cada matriz guarda los valores de las influencias y propagaciones de los elementos correspondientes a cada capa. Además se define un valor para designar la posición del jugador en el mapa y un valor que representa la estrategia a cumplir por el jugador.

**Entrenamiento:** La clase se encarga de generar, a partir de los datos de las capas y las decisiones correctas, un conjunto de datos de entrenamiento, los que serán almacenados en un fichero o base de datos, dependiendo de como se desee implementar en el videojuego.

**AdministraciónRed:** Esta clase crea la red donde se define la cantidad de unidades de entradas, unidades ocultas y unidades de salidas. Posteriormente define el algoritmo de entrenamiento a utilizar en dicha red y se le pasan los datos de entrenamiento guardados con anterioridad. Esta clase devuelve una salida en correspondencia con el entrenamiento recibido.

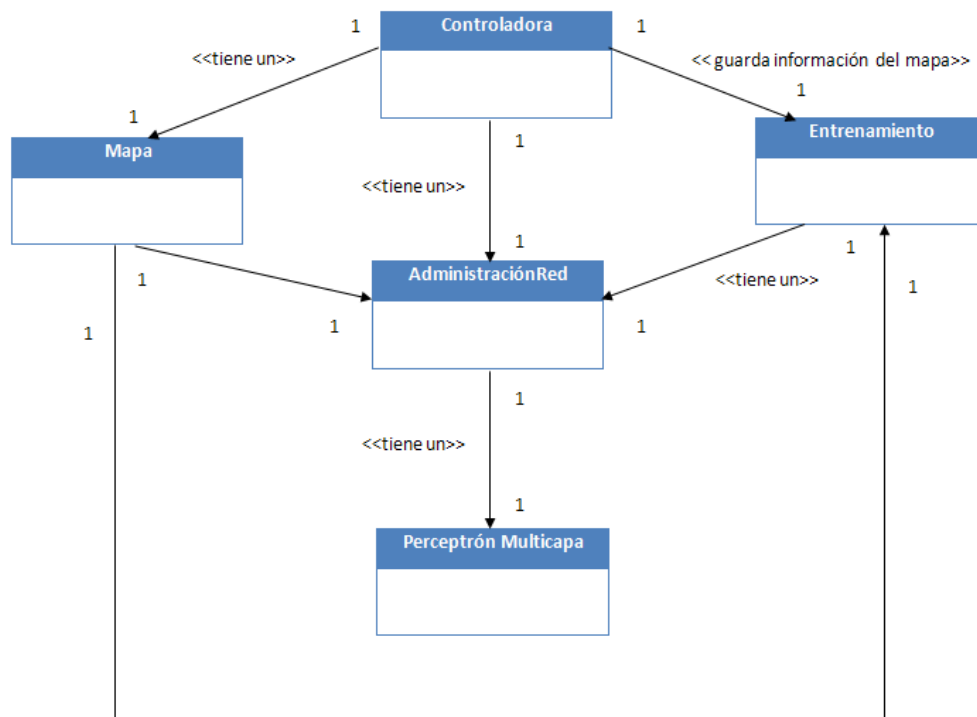


Fig. 31 Relación entre clases.

### 3.3 Algoritmo propuesto

#### 1. Determinación de las variables de entrada

##### **Influencias de las variables del Mapa:**

Un Mapa de Influencia representa el mundo virtual en donde se encuentra un jugador en un videojuego de estrategia. Está compuesto por capas, donde cada capa va a estar representada por una matriz. Las matrices van a estar conformadas por elementos con los que el agente interactúa, recursos, obstáculos, enemigos, etc. A cada uno de estos elementos se le asigna un valor, este valor será asignado de acuerdo a la influencia que posee en el jugador, es decir, si la influencia es negativa va a tomar valor negativo entre -1 y 0, y si es positiva entre 0 y 1. Estos valores se asignan también de acuerdo al nivel de importancia que se le confieran a los elementos dependiendo de las características del videojuego a implementar y se guardarán en sus respectivas matrices. Además de los elementos de las capas, es necesario trabajar con algunas variables que no se encuentran en el mapa, pero están presentes en el videojuego y su uso es esencial, por ejemplo la posición del jugador.

##### **Como establecer los valores de propagación:**

Además de los valores de influencia, se necesita conocer los valores de propagación de las influencias de cada elemento en su respectiva capa. Los valores de la propagación se calculan para cada celda de la capa sumando todos los valores de influencia que incidan sobre dicha celda. Los valores de influencia de cada elemento fueron asignados de forma tal que la suma de todas las influencias que afecten a una celda, se encuentre en el rango de -1 y 1. A continuación se muestran ejemplos con valores asignados a diferentes variables.

En este caso, están presentes dos enemigos en su capa correspondiente, con sus influencias, las cuales ocurren sólo en las celdas adyacentes al elemento. En el caso en que las influencias de los elementos coincidan en la misma celda esta aumentará su valor de propagación, dándole un valor máximo a -1 para el caso de la capa de enemigo. **(Fig.32)**

## CAPITULO 3: RESULTADOS GENERALES

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	-0.12	-0.12	-0.12	0	0	0
0	0	-0.12	-0.25	-0.12	0	0	0
0	0	-0.12	-0.12	-0.25	-0.12	-0.12	0
0	0	0	0	-0.12	-0.25	-0.12	0
0	0	0	0	-0.12	-0.12	-0.12	0
0	0	0	0	0	0	0	0

**Fig.32** *Influencia y propagación de los enemigos*

La influencia de las fortificaciones del jugador poseen valores positivos, la siguiente tabla muestra las influencias de dos fortificaciones y su propagación señaladas con un borde más ancho, el valor máximo que pueden adquirir estas celdas es la cantidad de elementos que pueden influir sobre dicha celda, la suma de estos elementos arrojará como valor máximo 1. **(Fig.33)**

0	0	0	0	0	0	0	0
0	0	0	0.1	0	0	0	0
0	0	0.2	0.2	0.2	0	0	0
0	0.1	0.2	0.41	0.2	0.3	0	0
0	0	0.2	0.2	0.4	0.2	0.2	0
0	0	0	0.2	0.2	0.41	0.2	0.1
0	0	0	0	0.2	0.2	0.2	0
0	0	0	0	0	0.1	0	0

**Fig.33** *Influencia y propagación de las fortificaciones jugador*

## CAPITULO 3: RESULTADOS GENERALES

En el caso de los obstáculos su influencia se dará en la celda en la que se encuentre el elemento, es decir, que los valores de influencia y propagación estarán dados sólo en las celdas donde estén presentes los obstáculos. Por tanto el valor máximo que puede influir sobre el elemento es -1. **(Fig.34)**

0	0	0	0	0	0	0	0
0	0	0	0	0	0	-1	0
0	0	0	0	0	0	0	0
0	0	0	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	-0.5	0	0	-0.5	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

**Fig.34** *Influencia y propagación de los obstáculos*

Este elemento posee características similares a las del obstáculo, con la diferencia que poseerá valores positivos y su valor máximo de influencia es 1. **(Fig.35)**

0	0	0	0	0	0	0	0
0	0.5	0	0	0	0	0.25	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0.2	0	0	0.75	0	0
0	0	0	0	0	0	0	0
0.2	0	0	0	0	0	0	0

**Fig.35** *Influencia y propagación de los recursos*

Las fortificaciones enemigas presentan un diseño de influencia similar a la que posee la fortificación del jugador, pero en este caso sus valores de influencia y propagación son positivos. En la siguiente

tabla se muestra un ejemplo donde la matriz guarda valores de influencia y propagación de tres fortificaciones enemigas. **(Fig.36)**

0	0	0	0	0	0	0	0
0	0	0	-0.1	0	-0.1	0	0
0	0	-0.2	-0.2	-0.4	-0.2	0.2	0
0	-0.1	-0.2	-0.51	-0.4	-0.51	-0.2	-0.1
0	0	-0.2	-0.2	-0.5	-0.2	0.2	0
0	0	0	-0.2	-0.2	0.2	0	0
0	0	-0.1	-0.2	-0.41	-0.2	-0.1	0
0	0	0	-0.2	-0.2	-0.2	0	0

**Fig.36** *Influencia y propagación de las fortificaciones enemigas*

Luego de definir los elementos del Mapa de Influencia y establecer sus valores de propagación, se deben definir además dos variables significativas, la posición inicial del jugador y la estrategia a cumplir en un estado del videojuego

**Posición inicial del jugador:** Representa la posición que inicialmente ocupa el jugador en el mapa, este valor se calcula de la siguiente manera, teniendo en cuenta las variables: **(fila, columna)**, posición del jugador en el mapa, **n** y **m** dimensiones del mapa, donde **n** es la cantidad de filas y **m** la cantidad de columnas

$$\text{pos\_jugador} = (m * \text{fila} + \text{columna} + 1) / (n * m / 2) - 1$$

La siguiente tabla muestra un ejemplo de cuales serian los valores de la posición del jugador en una matriz de 8 \* 8.





**Posición final del jugador:** Posición que ocupa el jugador en el mapa después de ejecutar una determinada acción, para dar cumplimiento a una estrategia.

Posición que ocupará el jugador en una posición del mapa al ejecutar la acción que dará cumplimiento a una determinada estrategia, este valor se calcula de la siguiente manera, teniendo en cuenta las variables: **n** y **m** dimensiones del mapa, donde **n** es la cantidad de filas y **m** la cantidad de columnas, **pos\_jugador**, posición donde se moverá el jugador para ejecutar la acción. Como esta posición está determinada en el intervalo de -1 a 1 es necesario transformarlo a valores de filas y columnas de la siguiente manera:

$$p = (\text{pos\_jugador} + 1) * (n * m / 2) - 1$$

$$\text{fila} = p / 8$$

$$\text{columna} = p \bmod 8$$

### 3. Algoritmo usado para generar los datos de entrenamiento

Este es un algoritmo usado para la obtención de los datos de salida que se necesitan para entrenar a la red. Dicho algoritmo se utiliza para no pasar los datos de forma manual debido a lo engorroso que resulta. Según la estrategia seleccionada, el algoritmo elige entre todos los puntos del mapa donde estén ubicados los elementos correspondientes a dicha estrategia, el punto donde la suma de todas las influencias de todas las capas en esa celda tenga el valor más positivo. Dicho de otra forma, el algoritmo tomará como objetivo el punto donde haya menos influencias que perjudiquen al jugador y más influencias que lo beneficien. Este sería el punto seleccionado como objetivo para cumplir la estrategia.

Luego, elijará un movimiento tratando de hacer llegar al jugador al punto seleccionado como objetivo por la vía más cercana y segura. Esto se realiza analizando la posición relativa del punto objetivo al jugador, para seleccionar, en la dirección del punto, la celda adyacente con mayor valor de influencias. Esta celda será la posición hacia donde se moverá el jugador. De esta manera se asegura que por lo menos ese primer paso para cumplir la estrategia se realice por la celda menos perjudicial al jugador y dirigiéndose al objetivo.

Este algoritmo permitirá generar automáticamente una serie de decisiones para entrenar la red, cada una de las cuales daría una aproximación de la decisión que tomaría una persona para ese estado del mapa.

Una vez terminado el entrenamiento, la red tendrá un comportamiento similar al algoritmo teniendo en cuenta que este es utilizado sólo como un medio para probar la vinculación de los mapas con la red con mayor rapidez. Si se quisiera tener un comportamiento mejor se deberá entrenar la red manualmente con decisiones tomadas por jugadores humanos, es decir, en un videojuego real.

#### **4. Realizar el entrenamiento de la red**

Para realizar el entrenamiento de la red, los vectores con los datos de entrada y salida se almacenan y serán utilizados por la red para entrenarse, mediante un conjunto de algoritmos de entrenamiento que posee. Estos datos serán traducidos de manera tal que la red pueda generar una salida satisfactoria para el conjunto de datos de entrada.

#### **5. Respuesta de la red**

Después de entrenada la red, se le pasa el vector con el estado de todas las matrices de influencia de un mapa, la posición del jugador y la estrategia a cumplir, y esta se encargará de dar una respuesta que estará definida por un vector que contiene dos valores: la posición final del jugador y la decisión a realizar por el jugador. Estas respuestas estarán dadas de acuerdo al entrenamiento realizado a la red con anterioridad.

### **3.4 Aplicación Demostrativa**

Con el fin de mostrar de manera práctica los resultados alcanzados en la presente investigación, se desarrolló un Demo sobre la base del algoritmo propuesto. Dicha aplicación muestra la vinculación de los Mapas de Influencia con las Redes Neuronales Artificiales. De manera breve se describen las clases utilizadas en la realización de la aplicación. La plataforma utilizada para la confección de la aplicación fue el Visual Studio2005 y se programó en el lenguaje orientado a objetos C++. Se utilizó la biblioteca Flood para el trabajo con la Red Neuronal Artificial donde se realiza todo el procesamiento de la información para poder generar una decisión final.

### 3.4.1 Funcionalidades de la aplicación

A continuación se muestran las funcionalidades que presenta el Demo, el cual se realizó para mostrar el funcionamiento del algoritmo propuesto.

#### -Nuevo mapa

Esta opción genera un nuevo mapa donde el usuario puede adicionar los elementos que desee. Cada uno de estos elementos se describe a continuación:

- *Fortificaciones del jugador: Se le asignó un valor de 0.41 y posee valores de influencia de 0.2 y 0.1, su valor de influencia irá disminuyendo de acuerdo a la distancia que posea respecto a la celda en donde se encuentre el elemento.*
- *Enemigo: Se le asignó un valor negativo igual a -0.2 ya que influye de manera negativa en el jugador y su influencia es de -0.5*
- *Obstáculos: Se le asignó un valor negativo igual a -0.5, ya que influye de manera negativa en el jugador.*
- *Recursos: Se le asignó un valor igual a 0.5, este elemento influye de manera positiva sobre el jugador, por tal motivo se le confirió un valor positivo.*
- *Fortificaciones enemigas: Se le asignó un valor negativo igual a -0.41 ya que influye de manera negativa en el jugador y su influencia es de -0.5 y -0.1, su influencia será más negativa en las celdas adyacentes a la celda que posee el valor real.*
- *Posición inicial del jugador: El jugador se va identificar con un valor igual a 1.*

#### -Selección de las estrategias

Mediante una lista el usuario puede seleccionar la estrategia que desea que el agente realice.

**Estrategia:** La estrategia seleccionada por el jugador tendrá un valor entre 0 y 1.

- Tomar recurso: 0.1
- Atacar enemigo: 0.3

- Atacar fortificaciones enemigas: 0.5
- Esconderse: 0.7
- Defenderse: 0.9

**Tipo:** La obtención de los datos de entrenamiento de la red se hace a través de las siguientes formas:

**Manual:** Se crea un nuevo mapa donde se agregan los elementos y luego se guardan los datos en el fichero.

**Automático:** Se generan mapas aleatorios de acuerdo a una cantidad especificada por el usuario, brinda la opción de detener el entrenamiento cuando este lo desee. Y por último se guardan los datos en el fichero.

**Entrenar:** Esta opción toma los datos guardados en el fichero y se encarga de introducirlos a la red para que pueda entrenarse.

### -Tomar decisión

Las Redes Neuronales Artificiales necesitan ser entrenadas para tomar decisiones estratégicas en el estado actual de un videojuego, las siguientes funcionalidades brindan la opción de generar o introducir datos de entrenamiento almacenados en un fichero, que será utilizado luego por la red para el aprendizaje.

De acuerdo a una estrategia seleccionada, esta opción muestra la acción a seguir por el jugador para dar cumplimiento a dicha estrategia.

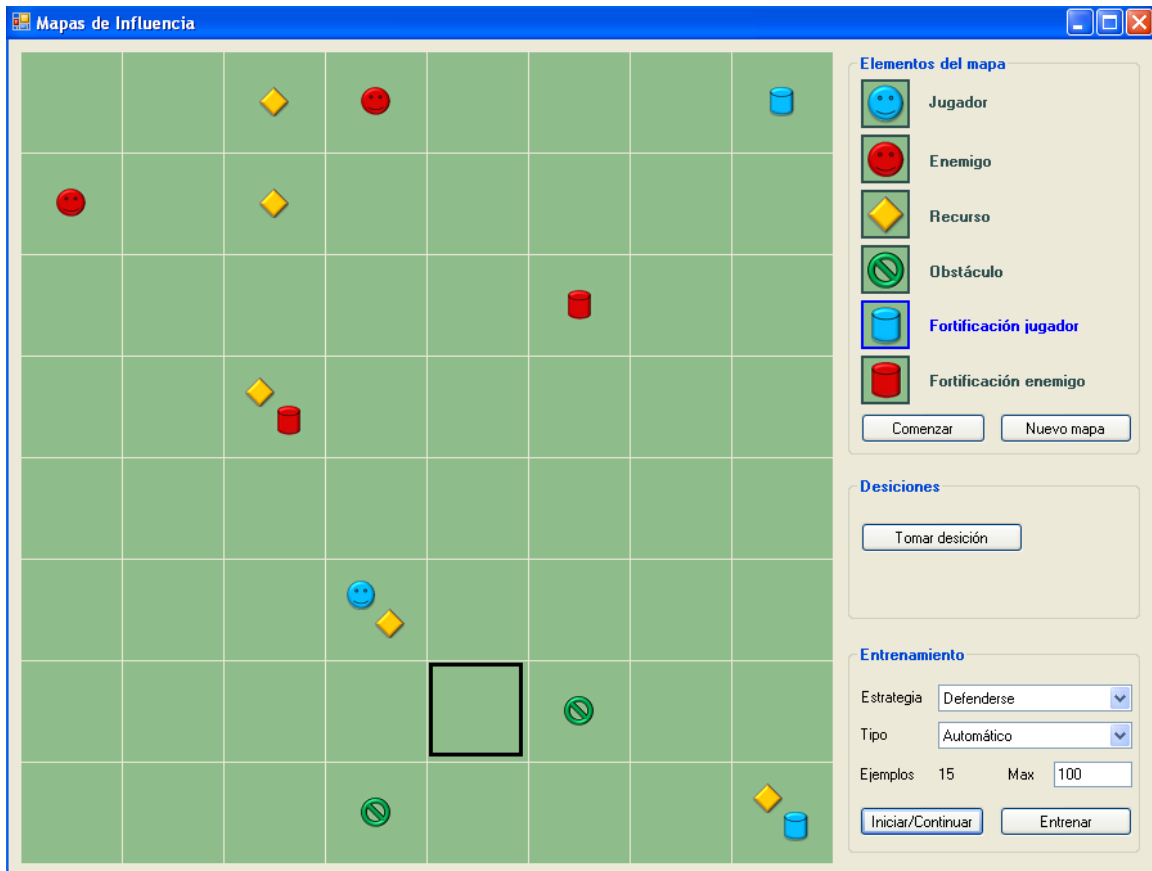
Acciones a realizar: Las decisiones mostradas en la aplicación serán los pasos que el jugador llevará a cabo para cumplir determinada estrategia y están dadas en un rango entre 0 y 1.

1. Avanzar: 0.2
2. Esconderse: 0.4
3. Atacar: 0.6
4. Defenderse: 0.8

### 3.5 Reglas de juego

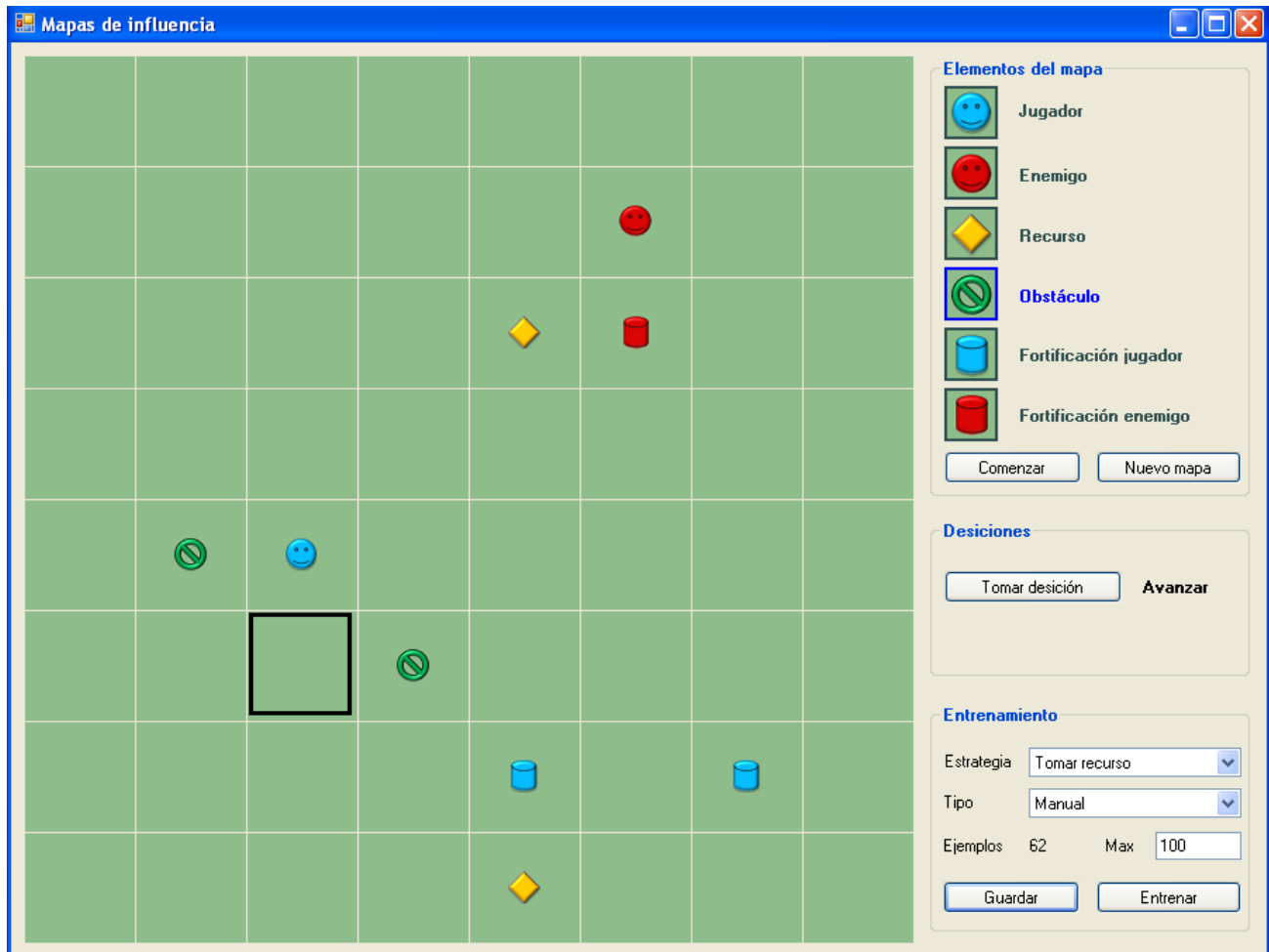
- **Tomar recursos:**El jugador tomará el recurso con mayor valor positivo.
- **Atacar enemigo:**Atacar al enemigo con menor influencia negativa.
- **Atacar fortificaciones enemigas:**Atacar la fortificación enemiga con menor influencia negativa.
- **Esconderse:** Ubicarse dentro de las fortificaciones del jugador.
- **Defenderse:** Ubicarse en un punto donde esté presente alguna influencia de una de sus fortificaciones.

A continuación se muestran ejemplos de la aplicación en el proceso de entrenamiento de la red y a la hora de arrojar un resultado de acuerdo a una estrategia dada:



**Fig. 39** Mapa generado para entrenar la red.

En la **Fig.40** se muestra un mapa generado con la decisión **Avanzar**, esta acción debe ser ejecutada por el jugador para dar cumplimiento a la estrategia **Tomar recurso**.



**Fig.40** Resultado generado por la red.

### 3.6 Conclusiones Parciales

En el capítulo se realizó una propuesta de solución al problema planteado sobre la toma de decisiones en videojuegos de estrategia, vinculando las Redes Neuronales Artificiales con los Mapas de Influencia. Se confeccionó un Demo que pone en práctica la presente investigación y sirve de apoyo en la validación del algoritmo propuesto.

### CONCLUSIONES

Utilizando una Red Neuronal Artificial para generar una decisión de un Mapa de Influencia se obtiene como resultado una apariencia inteligente en un agente que se encuentra interactuando dentro de un videojuego de estrategia. Se logra un ambiente realista en la escena virtual logrando la capacidad de imitar a expertos jugadores ya que la máquina interactúa con el jugador de una forma más real debido a su adaptación a diversas situaciones, además se evita la pérdida de información que existe al utilizar otras técnicas para la toma de decisiones, como el método convencional de la suma ponderada que proporciona un aproximación.

El Mapa de Influencia es otra de las técnicas importantes de un videojuego de estrategia para la organización de la información a la hora de tomar una decisión. Almacenan la información de todo lo acontecido en un estado del videojuego, por tanto, es un repositorio donde las variables o elementos del mundo virtual van a estar almacenados en capas distintas. Además de las capas es necesario tener en cuenta variables muy importantes que se encuentran dentro del videojuego y no forman parte del mapa, como ejemplo de una variable propuesta en la investigación se encuentra la posición inicial del jugador, un aspecto esencial a la hora de tomar una decisión. Cuando el agente que interactúa con el escenario necesita realizar una estrategia, estas capas se combinan para dar una visión más precisa de la situación actual. Con la ayuda de las Redes Neuronales Artificiales el agente va a tomar características inteligentes en cuanto a la toma de decisiones.

Sin embargo es importante señalar que estas requieren ajuste propio, y como tal, el uso de una Red Neuronal Artificial no está enfocado en reducir tiempo y desarrollo de pruebas, sino en generar respuestas óptimas y que se acerquen más a las que podría dar el cerebro humano.

Como resultado de la investigación se obtuvo un Demo que demuestra la eficacia de unir dos técnicas utilizadas en los videojuegos de estrategia, las Redes Neuronales Artificiales y los Mapas de Influencias. En la investigación realizada se vinculan dos términos usualmente encontrados en diversas bibliografías y que en su mayoría se encuentran en idiomas extranjeros, por lo que se recomienda la investigación como material de consulta.



### RECOMENDACIONES

A este trabajo se le recomiendan los siguientes aspectos:

1. Diseñar videojuegos de estrategia donde se pueda aplicar un módulo para la toma de decisiones basado en las Redes Neuronales Artificiales y los Mapas de Influencias.
2. Utilizar el algoritmo propuesto como vía de solución para los problemas de toma de decisiones en videojuegos de estrategia.
3. Utilizar la aplicación propuesta como medio de apoyo para verificar el resultado de los datos durante el aprendizaje de la red.

### REFERENCIAS BIBLIOGRÁFICAS

1-Desarrollo de Elementos Inteligentes. (2009). *Conferencia#6 "Técnicas de aprendizaje artificial. Redes Neuronales y Algoritmos Genéticos*. La Habana, Cuba: Universidad de las Ciencias Informáticas.

2-Desarrollo de Elementos Inteligentes. (2009). *Conferencia #1 "Introduction to the Game Artificial Intelligence"*. La Habana, Cuba: Universidad de las Ciencias Informáticas.

3-Dürsteler, J. C. (30 de abril del 2001 ). Mapas de Kohonen. *La revista digital de InfoVis.net* .

4-García, M. M. *Análisis Multicriterial*. Universidad de las Ciencias Informáticas(UCI).

5-Lopez, R. (Febrero 10, 2007). *(Flood A1) An Open Source Neural Networks C++ Library for Modeling of Data*. Technical University of Catalonia(UPC), Barcelona, España : International Center for Numerical Methods in Engineering (CIMNE).

6-Martín, V. (31 de mayo de 2009). *www.malavida.com*. Recuperado el 23 de enero de 2010, de *www.malavida.com*: <http://www.malavida.com/blog/b/8027/el-arte-de-la-estrategia-ii>

7-Palmer, A. M. (Enero del 2002). Tutorial sobre Redes Neuronales Artificiales:Los Mapas Autoorganizados de Kohonen. *Revista Electrónica de Psicología (Vo.l 6, No.1)* .

8-Rabin, S. (2004). *AI Game Programming Wisdom 2*. Hingham, Massachusetts: Charles River Media.

9-Rodríguez, A. T. (10 de junio de 2008). *Definición del comportamiento de carros autónomos en un videojuego de carreras empleando redes neuronales artificiales*. Ciudad de la Habana: Universidad de las Ciencias Informáticas.

10-Rodríguez., I. A. (Julio 2008). *Teoría de errores aplicada a los modelos de Redes Neuronales Artificiales utilizados en el proyecto "Desarrollo de Elementos Virtuales Inteligentes"*. La Habana: Universidad de las Ciencias Informáticas (UCI).

11-Urgellés, A. H. (junio de 2008). *APLICACIONES DE LAS REDES NEURONALES EN ENTORNOS VIRTUALES*. Ciudad de La Habana, Cuba: Universidad de las Ciencias Informáticas.

12-*www.rutarelativa.com*. (s.f.). Obtenido de *www.rutarelativa.com*. (21 junio 2010). Obtenido de (<http://www.rutarelativa.com/videojuegos/battlecruiser-starcraft-ii-lego/>)

13-*www.sosgamers.com*. (s.f.). Obtenido de <http://www.sosgamers.com/2010/06/nuevos-detalles-de-pokemon-black-white/>

## REFERENCIAS BIBLIOGRÁFICAS

---

14- *Martin, V.* (2009, mayo 31). El arte de la estrategia 2

15- **Polaino, Rafael Carrasco.** *Propuesta de tipología básica de los videojuegos de PC y consola.* Madrid, España. ICONO14, 2006. No14

## BIBLIOGRAFÍA

Lopez, R. (Febrero 10, 2007). *(Flood A1) An Open Source Neural Networks C++ Library for Modeling of Data*. Technical University of Catalonia(UPC), Barcelona, España : International Center for Numerical Methods in Engineering (CIMNE).

(Mayo del 2006)El lenguaje de los videojuegos, sus pliegues y recortes en las prácticas sociales. *Revista Novedades Educativas*.

Murillo Blanco, Javier y Laureano Collado, Javier y García-Margariño García, Iván (2005).*Implementación de juegos de estrategia con programación evolutiva*. [Trabajo de curso]

AndrésGarcíaManzano(2005).Videojuegosdesimulaciónentareaseducativas.<http://observatorio.cnice.mec.es/>.  
<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=artide&sid=441>

Arianna Herrera Bacallao y Yunerkis Prevot Urgelles(Junio 2008)*Aplicaciones de las Redes Neuronales en Entornos Virtuales*(Tesis).Universidad de las Ciencias Informaticas, La Habana, Cuba.

Mat Buckland(2005)*Programming Game AI by Example*. Wordware Publishing.

Begoña Alfageme y Pedro Sánchez(2002)*Aprendiendo habilidades con videojuegos*.Revista Científica de Comunicación y Educación.Murcia

Arianna Herrera Bacallao y Yunerkis Prevot Urgellés.(junio 2008) *Aplicaciones de las Redes Neuronales en Entornos Virtuales*.Universidad de las Ciencias Informáticas.Cuba.

Juan R. Rabunal y Julian Dorado(2006)*Artificial Neural Networks in Real –Life Applications*.Universidad de A Coruna, España.

Dpto:Realidad Virtual.(enero 2009)*Videojuegos*.Universidad de las Ciencias Informaticas.

Facultad de Ingeniería (UDELAR). (Noviembre 2008). *Seminario de Videojuegos y Tecnologías Interactivas*.

Rafael Carrasco Polaino. (ICONO 14 N°7 2006)*Propuesta de tipología básica de los videojuegosde PC y consola*. REVISTA DE COMUNICACIÓN Y NUEVAS TECNOLOGÍAS.

## ANEXOS

“Encuesta para validar la importancia de las estrategias a tener en cuenta por un jugador de videojuego de estrategia”

Sexo:

**Preguntas:**

1)-¿Has jugado un videojuego de estrategia?

SI \_\_\_\_\_ NO \_\_\_\_\_

2)- Coloque un número del 1 al 5 según el nivel de importancia que usted le concede a las estrategias llevadas a cabo por un jugador en un videojuego.

**Nivel de Importancia**

\_1\_ No importante

\_2\_ Poco importante

\_3\_ Importante

\_4\_ Muy importante

\_5\_ Súper importante

**Estrategias a tener en cuenta:**

a) \_\_\_\_\_ Tomar recursos para la batalla y avanzar.

b) \_\_\_\_\_ Escondarse del enemigo.

c) \_\_\_\_\_ Evadir obstáculos.

d) \_\_\_\_\_ Atacar las fortificaciones enemigas.

e) \_\_\_\_\_ Defenderse del enemigo.

## GLOSARIO

**Algoritmo de entrenamiento:** Proceso que le permite a una red neuronal artificial aprender de un conjunto de patrones.

**Aprendizaje:** Proceso mediante el cual un individuo adquiere conocimiento y experiencia, de forma que le permita modificar su comportamiento y adaptarlo a nuevas condiciones de su entorno.

**Inteligencia artificial:** Rama de la informática que se encarga de crear software y hardware capaces de imitar la inteligencia humana.

**Perceptrón:** Tipo de neurona artificial. Unidad básica de un perceptrón multicapa.

**Perceptrón multicapa:** Tipo de red neuronal artificial formada por múltiples capas de perceptrones.

**Propagación del error hacia atrás:** Algoritmo de aprendizaje que se puede utilizar para entrenar un perceptrón multicapa. Tiene como objetivo minimizar el error cometido por la red ante un conjunto de patrones de entrenamiento ajustando los pesos entre sus conexiones.

**Red Neuronal Artificial (RNA):** Técnica no determinista que emula el cerebro humano. Está compuesto por un conjunto de neuronas interconectadas que colaboran entre sí para obtener una salida.

**Técnica no determinista:** Técnica de la inteligencia artificial que incorpora un alto grado de incertidumbre.

**Mapa de influencia:** Representación del nivel o el mapa en el que se encuentra el agente en un videojuego de estrategia.

**Modelos cognitivos:** Modelos internos del mundo del videojuego.

**Táctica:** disposición y posterior movimiento de las unidades dentro del campo de batalla.

**Estrategia:** conjunto de decisiones e ideas que se ponen en marcha durante una campaña militar.

**Videojuegos de Estrategia:** videojuegos que requieren que el jugador ponga en práctica sus habilidades de planeamiento y pensamiento para conseguir la victoria.