

**UNIVERSIDAD DON BOSCO  
FACULTAD DE INGENIERIA  
ESCUELA DE ELECTRONICA**



PROYECTO DE GRADUACION

*SG – SIM 2006*  
*DISEÑO DE SOFTWARE DE SIMULACION DE REDES*  
*NEURONALES ARTIFICIALES*

PRESENTADO POR:

**LEOPOLDO FEDERICO SILIEZAR LEDEZMA,**

PARA OPTAR POR EL TITULO DE INGENIERO EN ELECTRONICA.

**MARCOS ALEJANDRO GALLEGOS VIDES,**

PARA OPTAR POR EL TITULO DE INGENIERO EN AUTOMATIZACION.

ASESOR:

Eduardo Rivera

Facultad de Ingeniería de la Universidad Don Bosco

CIUDADELA DON BOSCO, OCTUBRE DE 2006

# CONTENIDO

---

<b>CONTENIDO .....</b>	<b>I</b>
<b>INDICE DE FIGURAS .....</b>	<b>III</b>
<b>INDICE DE TABLAS .....</b>	<b>V</b>
<b>INTRODUCCION.....</b>	<b>VI</b>
<b>Capítulo 1 - DEFINICION DEL TEMA .....</b>	<b>1-1</b>
1.1. PLANTEAMIENTO DE PROBLEMA.....	1-1
1.2. DESCRIPCIÓN DE LA PROPUESTA DEL PROYECTO .....	1-1
1.3. VALIDACION DE RESULTADOS .....	1-3
1.4. OBJETIVOS .....	1-3
1.4.1. OBJETIVOS GENERALES .....	1-3
1.4.2. OBJETIVOS ESPECÍFICOS .....	1-3
1.5. ALCANCES.....	1-4
1.6. LIMITACIONES .....	1-4
<b>Capítulo 2 - MARCO TEORICO .....</b>	<b>2-6</b>
2.1. RED NEURONAL BIOLOGICA – RED NEURONAL ARTIFICIAL .....	2-6
2.1.1. ¿QUÉ ES UNA RED NEURONAL BIOLOGICA?.....	2-6
2.1.1.1. DESCUBRIMIENTO.....	2-6
2.1.2. ¿QUÉ ES UNA RED NEURONAL ARTIFICIAL? [7].....	2-8
2.2. ANTECEDENTES DE LAS REDES NEURONALES E INTELIGENCIA ARTIFICIAL.....	2-10
2.2.1. BREVE HISTORIA .....	2-10
2.2.2. INTELIGENCIA ARTIFICIAL.....	2-13
2.2.2.1. TECNOLOGIAS DE APOYO Y CAMPOS DE DESARROLLO DE LA INTELIGENCIA ARTIFICIAL [2]. .....	2-13
2.3. REDES NEURONALES ARTIFICIALES E INTELIGENCIA ARTIFICIAL.....	2-15
2.4. APLICACIONES DE LAS REDES NEURONALES ARTIFICIALES [2].....	2-19
2.4.1. RECONOCIMIENTOS DE PATRONES.....	2-22
2.5. TIPOLOGIA DE REDES NEURONALES MÁS RELEVANTES. [2].....	2-25

2.6. ELEMENTOS DE UNA NEURONA ARTIFICIAL .....	2-27
2.6.1. ELEMENTOS DE UNA NEURONA ARTIFICIAL. [2].....	2-27
2.6.1.1. FUNCIONES DE ACTIVACIÓN.....	2-29
2.6.2. TOPOLOGIAS IMPLEMENTADAS. [1]-[2]-[3].....	2-32
2.6.2.1. RED PERCEPTRON.....	2-32
2.6.2.1.1. ALGORITMO DE APRENDIZAJE .....	2-33
2.6.2.2. RED MULTIPLE PERCEPTRON (M - PERCEPTRON).....	2-36
2.6.2.2.1. ALGORITMO DE APREDNDIZAJE.....	2-37
2.6.2.3. RED ADALINE .....	2-40
2.6.2.3.1. ALGORITMO DE APRENDIZAJE .....	2-41
2.6.2.4. RED BACKPROPAGATION. [1] .....	2-43
2.6.2.4.1. ALGORITMO DE APRENDIZAJE .....	2-45
<b>Capítulo 3 - AMBIENTE GRAFICO .....</b>	<b>3-50</b>
3.1. AMBIENTE GRAFICO BÁSICO.....	3-50
3.1.1. AMBIENTE GRAFICO DE LA RED PERCEPTRON .....	3-52
3.1.2. AMBIENTE GRAFICO DE LA RED M - PERCEPTRON .....	3-54
3.1.3. AMBIENTE GRAFICO DE LA RED ADALINE .....	3-56
3.1.4. AMBIENTE GRAFICO DE LA RED BACKPROPAGATION.....	3-57
 <b>BIBLIOGRAFIA</b>	
 <b>APENDICE A .....</b>	<b>A-1</b>
 <b>APENDICE B .....</b>	<b>B-3</b>
 <b>APENDICE C .....</b>	<b>C-12</b>
 <b>APENDICE D .....</b>	<b>D-13</b>
 <b>APENDICE E .....</b>	<b>E-15</b>

# INDICE DE FIGURAS

---

FIGURA 2-1 NEURONA BIOLÓGICA .....	2-7
FIGURA 2-2 MODELO DE UNA NEURONA ARTIFICIAL.....	2-9
FIGURA 2-3 SITUACIÓN DE LAS REDES NEURONALES EN EL CAMPO DE LA INTELIGENCIA ARTIFICIAL.....	2-17
FIGURA 2-4 FORMA GENERAL DE UNA NEURONA ARTIFICIAL.....	2-29
FIGURA 2-5 FUNCIÓN DE TRANSFERENCIA ESCALÓN POLAR.....	2-30
FIGURA 2-6 FUNCIÓN DE TRANSFERENCIA ESCALÓN BIPOLAR.....	2-30
FIGURA 2-7 FUNCIÓN DE TRANSFERENCIA LINEAL.....	2-31
FIGURA 2-8 FUNCIÓN DE TRANSFERENCIA SIGMOIDAL.....	2-31
FIGURA 2-9 REPRESENTACIÓN DEL PERCEPTRÓN DE ROSENBLATT.....	2-32
FIGURA 2-10 DISTINTAS FORMAS DE LAS REGIONES GENERADAS POR UN PERCEPTRÓN MULTINIVEL.....	2-36
FIGURA 2-11 PERCEPTRÓN MULTINIVEL.....	2-37
FIGURA 2-12 REPRESENTACIÓN DE LA RED ADALINE.....	2-41
FIGURA 2-13 REPRESENTACIÓN DE LA RED BACK-PROPAGATION STANDARD.....	2-44
FIGURA 3-1 VENTANA INICIAL.....	3-50
FIGURA 3-2 CONFIGURACIÓN INICIAL, TIPO DE RED.....	3-51
FIGURA 3-3 VENTANA INICIAL ACTIVA.....	3-51
FIGURA 3-4 VENTANA ACTIVA CON RED PERCEPTRÓN. AÑADIR ENTRADAS.....	3-52
FIGURA 3-5 VENTANA ACTIVA CON RED PERCEPTRÓN. AÑADIR NEURONA.....	3-52
FIGURA 3-6 VENTANA ACTIVA CON RED PERCEPTRÓN. AÑADIR SALIDA.....	3-53
FIGURA 3-7 VENTANA ACTIVA CON RED PERCEPTRÓN. TRAZAR LÍNEAS.....	3-53
FIGURA 3-8 VENTANA ACTIVA CON RED M-PERCEPTRÓN. AÑADIR ENTRADAS.....	3-54
FIGURA 3-9 VENTANA ACTIVA CON RED M-PERCEPTRÓN. AÑADIR NEURONAS POR CAPAS.....	3-54
FIGURA 3-10 IDENTIFICADOR DE NEURONAS POR CAPAS.....	3-55
FIGURA 3-11 VENTANA ACTIVA CON RED M-PERCEPTRÓN. AÑADIR SALIDAS.....	3-55
FIGURA 3-12 VENTANA ACTIVA CON RED M-PERCEPTRÓN. TRAZO DE LÍNEAS.....	3-56
FIGURA B-1 PREGUNTA 1.....	B-7
FIGURA B-2 PREGUNTA 2.....	B-7
FIGURA B-3 PREGUNTA 3.....	B-8
FIGURA B-4 PREGUNTA 4.....	B-8
FIGURA B-5 PREGUNTA 5.....	B-9
FIGURA B-6 PREGUNTA 6.....	B-10

FIGURA B-7 PREGUNTA 7.....	B-11
FIGURA C-1 HOJA PARA SIMULACIÓN EN MATLAB, PARA LOS ENCUESTADOS.....	C-12
FIGURA E-1 PREGUNTA 1 (ULTIMA ENCUESTA).....	E-18
FIGURA E-2 PREGUNTA 2 (ULTIMA ENCUESTA).....	E-19
FIGURA E-3 PREGUNTA 3 (ULTIMA ENCUESTA).....	E-19
FIGURA E-4 PREGUNTA 4 (ULTIMA ENCUESTA).....	E-20
FIGURA E-5 PREGUNTA 5 (ULTIMA ENCUESTA).....	E-20
FIGURA E-6 PREGUNTA 6 (ULTIMA ENCUESTA).....	E-21
FIGURA E-7 PREGUNTA 7 (ULTIMA ENCUESTA).....	E-21
FIGURA E-8 PREGUNTA 8 (ULTIMA ENCUESTA).....	E-22
FIGURA E-9 PREGUNTA 9 (ULTIMA ENCUESTA).....	E-22
FIGURA E-10 COMENTARIOS Y/O SUGERENCIAS.....	E-23
FIGURA E-11 TIEMPO ESTIMADO DE ADAPTACIÓN.....	E-23

# INDICE DE TABLAS

---

TABLA 2-1 TABLA DE ATRIBUTOS DE UN AGENTE INTELIGENTE .....	2-14
TABLA 2-2 TECNOLOGÍAS DE APOYO Y CAMPOS DE DESARROLLO DE LA INTELIGENCIA ARTIFICIAL .....	2-15
TABLA 2-3 FORMAS BÁSICAS DE COMPUTACIÓN. ....	2-18
TABLA 2-4 APLICACIONES COMERCIALES DE LAS REDES NEURONALES .....	2-20
TABLA 2-5 CONTINUACIÓN DE LAS APLICACIONES COMERCIALES DE LAS REDES NEURONALES.....	2-21
TABLA 2-6 RESUMEN DE LA CREACIÓN DE REDES NEURONALES ARTIFICIALES.....	2-26
TABLA 2-7 CONTINUACIÓN DEL RESUMEN DE LA CREACIÓN DE REDES NEURONALES ARTIFICIALES .....	2-27
TABLA B-1 DATOS ESTADÍSTICOS .....	B-5
TABLA E-2 DATOS ESTADÍSTICOS DE SEGUNDA ENCUESTA .....	E-16

# INTRODUCCION

---

La historia de la informática tal y como se conoce hoy en día es reciente. Hacia finales de los años 30 y durante la década de los 40, los trabajos de gente como Alan Turing o Von Neumann asientan las bases de la informática moderna.

En un principio se orienta hacia la computación algorítmica, es decir, la resolución de un determinado problema, obteniendo un algoritmo que manipula los datos relativos al problema. La combinación hardware - software se muestra como una potente herramienta para la resolución de problemas que el ser humano no podría resolver o que tardaría mucho tiempo en hacerlo.

La evolución del hardware ha hecho que la potencia de cálculo haya crecido de tal forma que los ordenadores hoy en día son indispensables en muchas áreas de actividad del ser humano. Sin embargo, la computación algorítmica no es suficiente cuando nos enfrentamos a ciertas tareas. Algo tan sencillo para el ser humano como reconocer una cara de otra persona, es el tipo de problema que no es tan fácil para ser resuelto por la vía algorítmica.

Debido a este tipo de problemas desde finales de los 50's se ha venido investigando en un conjunto de técnicas que utilizan un enfoque diferente para resolver los problemas. Este conjunto de técnicas y herramientas se bautizó con el nombre de Inteligencia Artificial (IA), con el cual se pretendía que los ordenadores presentaran un comportamiento inteligente, entendiendo por esto que supieran hacer frente a ciertos problemas de una manera similar a como lo hacen los seres humanos.

Dentro de la IA, se trabajó en dos enfoques distintos. Por un lado, se desarrolló lo que se conoce como el enfoque simbólico, el cual asienta sus bases en la manipulación de símbolos, en vez del mero cálculo numérico tradicional, el cual es propio de la computación algorítmica.

Herramientas como la lógica de predicados<sup>1</sup>, nos permiten manipular los símbolos y las reglas para obtener nuevas reglas. Este enfoque (el enfoque simbólico) se presta a ser muy útil en ciertos tipos de problemas, aunque en general tiene la desventaja de que a la hora de buscar la solución de un determinado problema, los métodos de deducción presentan una explosión combinatoria que hace que requiera bastante tiempo de cálculo.

El otro enfoque, es el enfoque tradicional conocido también como enfoque conexionista y es donde se encuadran las Redes Neuronales Artificiales (RNA). La idea aquí es desarrollar un sistema formado por pequeñas unidades de cálculo en cierta medida muy simples y hacer mediante conexiones entre ellas, que todo el conjunto sea capaz de resolver cierta clase de problemas. Abstract [13].

Precisamente, el presente proyecto trata de dar una herramienta para entender las RNA de una manera gráfica y más sencilla, a través de un software, dando de ésta forma un aporte a la enseñanza de la materia que tiene relación con las RNA de la Universidad Don Bosco. Proporcionando al alumno una herramienta más de apoyo, para aprender el contenido de la cátedra.

---

<sup>1</sup> Lógica de predicados: es un cálculo que permite describir lo que el usuario desea que un programa realice. Ésta es una aplicación que ha empezado a ser empleada para el desarrollo de al menos las partes críticas de un sistema. De esta manera, piezas de código especificadas formalmente, pueden ser verificadas, en principio, matemáticamente, incrementando la confiabilidad del sistema completo.[11]

Este documento esta dividido en tres capítulos principales, el primer capítulo trata sobre la definición del tema, el ¿por qué del proyecto?, y los objetivos que se pretenden alcanzar con él.

El segundo capítulo, narra un poco de los antecedentes de las Redes Neuronales Artificiales, así como su desarrollo en las diferentes ramas de la ciencia, en las cuales su aplicación es muy importante.

El tercer capítulo describe la utilización del software desarrollado, los pasos que se deben seguir para poder conformar una Red Neuronal Artificial, y sus diferentes topologías que se encuentran en él.

# **Capítulo 1 - DEFINICION DEL TEMA**

---

## **1.1. PLANTEAMIENTO DE PROBLEMA**

Hasta el momento, una de las cátedras que se imparte en la Universidad Don Bosco que pretende enseñar Redes Neuronales Artificiales se ha vuelto engorrosa en la parte práctica de la materia, debido a las complicaciones que presenta el software utilizado actualmente, el cual es conocido como Matlab.

Matlab presenta un ambiente a base de líneas de comandos, es decir, que el usuario va definiendo todo a base de palabras y símbolos, que luego el software interpreta para realizar las operaciones puestas en él. Esto implica que el desarrollo de las prácticas de RNA, se vuelven mecanografiadas, provocando en el alumno una distracción de su atención de la práctica.

Por el hecho que Matlab se utiliza con comandos, éste requiere de un tiempo relativamente extenso de adaptación para el usuario, ya que se deben conocer varias instrucciones para crear una red; Bien es cierto que Matlab posee un ambiente grafico, sin embargo, no es amistoso y en ocasiones no se logran distinguir ciertos elementos de la red creada.

## **1.2. DESCRIPCIÓN DE LA PROPUESTA DEL PROYECTO**

El desarrollo de la cátedra de Redes Neuronales Artificiales es muy importante debido a que brinda conceptos básicos de Inteligencia Artificial, éste es uno de los

objetivos fundamentales de la materia. Sin embargo, la herramienta grafica con la cual se imparten los laboratorios, no cuenta con **un ambiente amigable** en el que el usuario sea capaz de adaptarse fácilmente a la comprensión de conceptos dados en la asignatura.

Además, el tipo de software utilizado en la actualidad no fue diseñado específicamente para el propósito de la materia, porque utiliza una extensa cantidad de recursos informáticos no relacionados con la misma; también es de considerar que requiere una licencia de usuario, lo que haría incurrir en gastos adicionales a un estudiante.

Debido a lo anterior, es necesario que el software a diseñar sea una herramienta de apoyo a la materia, para que el alumno disponga de más recursos que le faciliten la visualización, comprensión y la simulación de Sistemas de Redes Neuronales Artificiales, en un ambiente más amistoso, permitiéndole una adaptación rápida al programa, así como un mejor aprendizaje de la materia.

El proyecto a realizar eliminará por completo las líneas de comando, y en sustitución de los comandos se utilizarán iconos (imágenes), los cuales demostrarán al usuario conceptos vistos en la teoría de la materia.

El software será capaz de demostrar los conceptos teóricos que se puedan expresar gráficamente, así como también, tendrá la opción de observar los datos de la red, de forma numérica, si al usuario le interesara.

### **1.3. VALIDACION DE RESULTADOS**

El software esta propuesto para el sistema operativo de Windows XP, desarrollado bajo la plataforma de Visual Basic .NET (framework 1.1).

Además, el software será puesto a prueba con diferentes tipos de usuarios, es decir, usuarios con conocimiento o no, del tema de Redes Neuronales Artificiales, para observar el buen funcionamiento del software, así como la adaptación del usuario.

### **1.4. OBJETIVOS**

#### **1.4.1. OBJETIVOS GENERALES**

Desarrollar un software capaz de fortalecer el aprendizaje de la materia de **Redes Neuronales Artificiales**.

#### **1.4.2. OBJETIVOS ESPECÍFICOS**

- Diseñar una interfaz de usuario más amigable, que permita reconocer el comportamiento y aplicación de una Red Neuronal Artificial con mayor facilidad.
- Verificar el funcionamiento de la red simulada, introduciendo patrones que le permitan al usuario comprobar el buen trabajo de ésta.
- Desarrollar un software que permita al estudiante comprobar conceptos teóricos específicos relacionados con el curso de la materia del mismo nombre.

## **1.5. ALCANCES**

- El simulador contará con cuatro diferentes tipos de redes, las cuales son: Perceptrón, Perceptrón Múltiple, ADALINE y Retro—Propagación.
- Contará con una capacidad variable de unidades de procesamiento y capas internas definidas por el usuario.
- El software será capaz de aceptar archivos de imágenes prediseñadas, o diseñadas por el usuario para el entrenamiento de la red, lo que le permitirá una mejor percepción del funcionamiento de éstas.
- El simulador se creará con Visual Basic.NET debido a que presenta facilidad en ciertos comandos, y además por ser uno de los programas de última generación en el mercado informático. Sin embargo, no se deja de lado la posibilidad de utilizar algún software adicional que facilite el desarrollo del simulador.

## **1.6. LIMITACIONES**

- El tiempo de respuesta estará sujeto a las condiciones del hardware que se involucre en la ejecución del programa.
- Los archivos de imagen que aceptará el simulador serán de extensión BMP y JPG, debido a la facilidad de creación de los mismos, y por la robustez del programa a diseñar se considera suficiente que solo admita dichos tipos de imágenes en escala de grises para su procesamiento. Otro aspecto a considerar, es que las imágenes que se

admitirán tendrán un tamaño máximo de 800 x 800 píxeles para no prolongar aun más el tiempo de respuesta.

- El simulador estará sujeto únicamente a la adquisición de datos y a su respectivo procesamiento; es decir, introducirlos a la red y finalmente mostrar los datos resultantes.
- Las redes con las cuales el simulador contará tendrán una interfaz definida para cada red; lo que conlleva a un funcionamiento individual de cada una de las topologías.

## Capítulo 2 - MARCO TEORICO

---

### 2.1. RED NEURONAL BIOLÓGICA – RED NEURONAL ARTIFICIAL

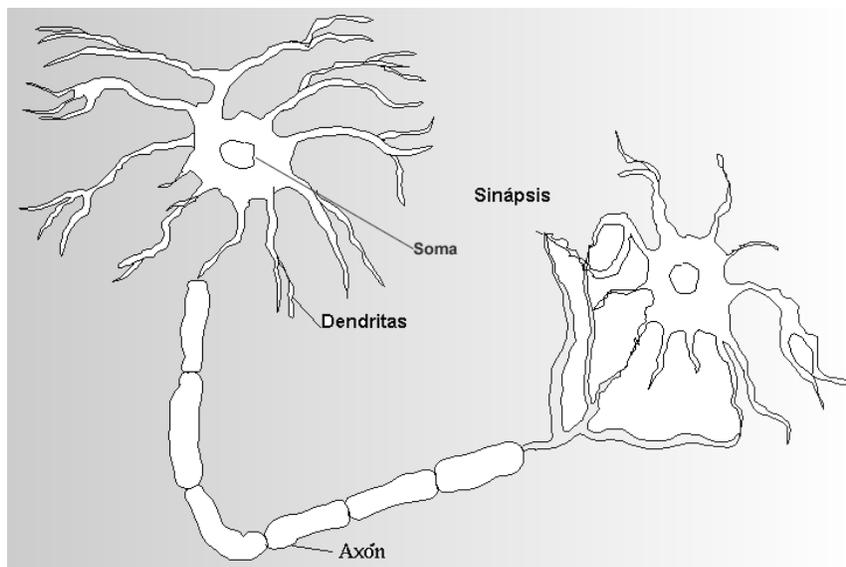
Para iniciar a desarrollar el tema de Redes Neuronales Artificiales (RNA), es necesario conocer lo que son las Redes Neuronales Biológicas. Debido a que las RNA tienen gran relación con las Redes Neuronales Biológicas.

#### 2.1.1. ¿QUÉ ES UNA RED NEURONAL BIOLÓGICA?

##### 2.1.1.1. DESCUBRIMIENTO.

A finales del siglo XIX se logró una mayor claridad sobre el trabajo del cerebro debido a las investigaciones de Ramón y Cajal en España y Sherrington en Inglaterra. El primero trabajó en la anatomía de las neuronas y el segundo en los puntos de conexión de las mismas o *sinapsis*. Se estima que en cada milímetro del cerebro hay cerca de 50.000 neuronas, conteniendo en total más de cien mil millones de neuronas y sinapsis en el sistema nervioso humano, cuya estructura se asemeja a la de la Figura 2-1.

El Cuerpo de la neurona o Soma contiene el núcleo y se encarga de todas las actividades metabólicas de la neurona y recibe la información de otras neuronas vecinas a través de las conexiones sinápticas (algunas neuronas se comunican solo con las cercanas, mientras que otras se conectan con miles).



**FIGURA 2-1 Neurona Biológica**

El tamaño y la forma de las neuronas es variable, pero con las mismas subdivisiones que muestra la Figura 2-1. Estas son:

1. El cuerpo de la neurona
2. Las ramas de extensión llamadas dendritas para recibir las entradas, y
3. El axón que lleva la salida de la neurona a las dendritas de otras neuronas.

Las dendritas, parten del soma y tienen ramificaciones. Se encargan de la recepción de señales de las otras células a través de conexiones llamadas sinápticas. Podría decirse que las dendritas son las conexiones de entrada de la neurona. Por su parte el axón es la "salida" de la neurona y se utiliza para enviar impulsos o señales a otras células nerviosas.

Cuando el axón está cerca de sus células destino se divide en muchas ramificaciones que forman sinapsis con el soma o axones de otras células. Esta unión puede ser

"inhibidora" o "excitadora", ya que una neurona biológica únicamente puede proporcionar dos estados uno en "alto" (verdadero) y el otro estado en "bajo" (falso). Además, cada neurona recibe de 10.000 a 100.000 sinapsis y el axón realiza una cantidad de conexiones similares. *Abstract [6]*.

### **2.1.2. ¿QUÉ ES UNA RED NEURONAL ARTIFICIAL? [7]**

Una Red Neuronal Artificial es un modelo de procesamiento de información que es inspirado por el modo de un sistema nervioso biológico, tal como el cerebro procesa información. El elemento clave de este paradigma, es la estructura original del sistema de procesamiento de información. Este sistema se compone de un gran número de elementos interconectados (neuronas), procesando y trabajando en armonía para resolver problemas específicos.

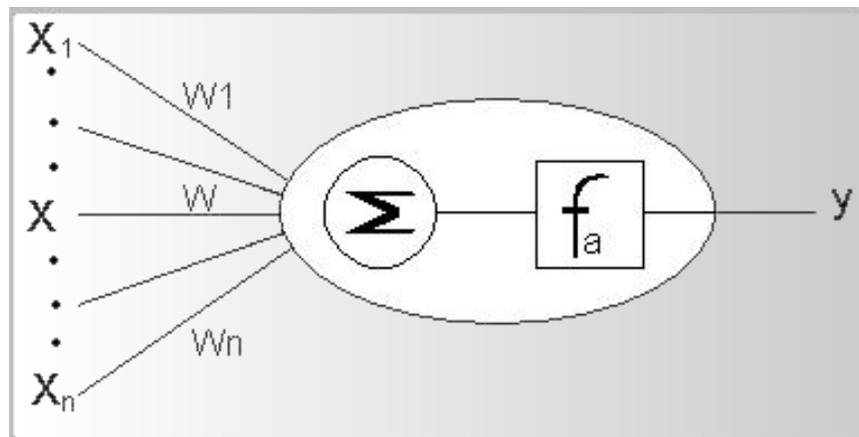
Las Redes Neuronales Artificiales, aprenden con ejemplos, así como las personas. Una RNA es configurada para una aplicación específica, tal como el reconocimiento de patrones o clasificación de datos, a través de un proceso de aprendizaje.

Aprender en sistemas biológicos implica ajustes para las conexiones sinápticas que existen entre las neuronas. Esto lo hace una Red Neuronal Artificial también, ajusta las conexiones sinápticas, que para su caso son cantidades llamados pesos sinápticos, los cuales se ajustan según el nivel de aprendizaje de la red.

Las RNAs se han aplicado a un gran número de problemas reales de complejidad considerable. Su ventaja más importante está en resolver problemas que son demasiado complejos para tecnologías convencionales, problemas que no tienen un algoritmo de solución o que su algoritmo de solución es muy difícil de encontrar.

En general, a causa de la abstracción del cerebro biológico, las Redes Neuronales Artificiales son aptas para resolver problemas que la gente puede resolver, pero las computadoras no pueden. Estos problemas incluyen reconocimiento de patrones y pronósticos (los cuales requieren el reconocimiento de tendencias de datos).

Entre los pioneros en el modelado de Neuronas Artificiales se encuentran Warren McCulloch y Walter Pitts. Estos dos investigadores propusieron un modelo matemático de neurona. En este modelo cada neurona estaba dotada de un conjunto de entradas ( $X$ ) y una salida ( $Y$ ), ver Figura 2-2.



**FIGURA 2-2 Modelo de una Neurona Artificial.**

## **2.2. ANTECEDENTES DE LAS REDES NEURONALES E INTELIGENCIA ARTIFICIAL**

### **2.2.1. BREVE HISTORIA**

La historia de la Inteligencia Artificial ha pasado por diversas situaciones, pudiéndose decir, que inicia en:

1903 Lee De Forest inventor del triodo (también llamado bulbo o válvula de vacío), podría decirse que la primera y gran máquina inteligente diseñada por el hombre fue el computador ENIAC, compuesto por 18,000 válvulas de vacío, teniendo en cuenta que el concepto de “inteligencia” es un término subjetivo, que depende de la inteligencia y la tecnología que tengamos en esa época.

En 1937, el matemático inglés Alan Mathison Turing (1912 - 1953), publicó un artículo de bastante repercusión sobre los “números calculables”, que puede considerarse el origen oficial de la Informática Teórica. En este artículo introdujo la Máquina de Turing, una entidad matemática abstracta que formalizó el concepto de algoritmo y resultó ser la precursora de las computadoras digitales. Con ayuda de su máquina, Turing pudo demostrar que existen problemas irresolubles, de los que ningún ordenador será capaz de obtener.

Es por ello y por otros descubrimientos, que a Turing se le considera el padre de la Inteligencia Artificial, por su famosa Prueba de Turing que permitiría comprobar si un programa de ordenador puede ser tan inteligente como el ser humano.

En 1951, William Shockley inventa el transistor de unión. El invento del transistor hizo posible una nueva generación de computadoras mucho más rápidas y pequeñas.

El término de Inteligencia Artificial<sup>2</sup>, fue inventado en 1956, en la conferencia de “Darmouth”, (un congreso en el que se hicieron previsiones triunfalistas a diez años que jamás se cumplieron) lo que provocó el abandono casi total de las investigaciones durante quince años.

En 1980 la historia se repitió con el desafío japonés de la Quinta Generación<sup>3</sup> que dio lugar en el auge de los sistemas expertos, pero que no alcanzó muchos de sus objetivos por lo que este campo sufrió una nueva interrupción desde inicio de los años noventa, hasta finales de dicha década y principios de la década siguiente.

En la actualidad estamos tan lejos de cumplir la famosa Prueba de Turing<sup>4</sup> como cuando se formuló: “Existirá Inteligencia Artificial cuando no seamos capaces de distinguir entre un ser humano y un programa de computadora en una conversación a ciegas...”.

---

<sup>2</sup> Inteligencia Artificial, conocida también por IA.

<sup>3</sup> La Quinta Generación de computadoras fue un proyecto ambicioso lanzado por Japón a finales de los setenta. Su objetivo era el desarrollo de un clase de computadoras que utilizarían técnicas de inteligencia artificial al nivel del lenguaje de máquina y serían capaces de resolver problemas complejos, como la traducción automática de una lengua natural a otra (del japonés al inglés, por ejemplo).

<sup>4</sup> Se llama Prueba o Test de Turing al procedimiento desarrollado por Alan Turing para identificar la existencia de inteligencia en una máquina.

Muchos de los investigadores de IA [12] sostienen que “la Inteligencia Artificial es un programa capaz de ser ejecutado independientemente de la máquina que lo ejecute, computador o cerebro...”.

Los juegos matemáticos antiguos, como el de las torres de Hanoi [13] (aproximadamente 3000 A.C.), demuestran el interés por la búsqueda de un bucle resolutor, una IA capaz de ganar en los mínimos movimientos posibles.

En 1987, Martín Fischles y Oscar Frischein describieron los atributos de un agente inteligente. Al intentar describir con un mayor ámbito (no solo comunicación) los atributos de un agente inteligente, la IA se ha expandido a muchas áreas que han creado enormes ramas de investigación y diferenciadas. Dichos atributos del agente inteligente se observan en la Tabla 2-1.

Podemos decir entonces que la IA incluye características humanas tales como el aprendizaje, la adaptación, el razonamiento, la auto corrección, el mejoramiento implícito y la percepción modular del mundo. Así podemos hablar ya no solo de un objetivo, sino de muchos dependiendo del punto de vista o utilidad que pueda encontrarse en la IA.

Muchos de los investigadores sobre IA sostienen que “la inteligencia es un programa capaz de ser ejecutado independientemente de la máquina que lo ejecute, computador o cerebro...”.

### **2.2.2. INTELIGENCIA ARTIFICIAL**

Se define [2] la Inteligencia Artificial (IA), como aquella inteligencia exhibida por artefactos creados por humanos (es decir, artificiales). El nombre también se usa para referirse al campo de la investigación científica que intenta acercarse a la creación de tales sistemas. Aunque la IA está rodeada de bastante ciencia ficción, se trata de una rama de la Informática, que trata sobre comportamientos inteligentes, aprendizaje y adaptación de máquinas.

La IA es hoy en día una de las áreas con más retos en las Ciencias de la Computación. Posee amplias relaciones con disciplinas matemáticas como el Álgebra y la Estadística tomando de éstas algunas herramientas para desempeñar su labor.

#### **2.2.2.1. TECNOLOGIAS DE APOYO Y CAMPOS DE DESARROLLO DE LA INTELIGENCIA ARTIFICIAL [2].**

Muchos de los campos de la Inteligencia Artificial son aplicados en la vida diaria, en actividades o tecnologías que posiblemente no se tenga ni idea que estén presentes. Algunas de las aplicaciones se presentan en la siguiente Tabla 2-2.

<b>Atributo</b>	<b>Descripción</b>
Actitudes mentales	Como creencias e intenciones
Capacidad de obtener conocimiento	Aprender
Resuelve problemas	Incluso la partición de problemas complejos con otros más simples.
Entiende	Posee la capacidad de crearle sentido, si es posible, ideas ambiguas o contradictorias
Planifica	Predice consecuencias, evalúa alternativas
Conoce límites	Límites de sus propias habilidades y conocimientos
Distingue	Puede distinguir a pesar de las similitudes de las situaciones
Original	Creando incluso nuevos conceptos o ideas
Generaliza	En situaciones que lo necesite.
Percibe	Puede percibir y modelar el mundo exterior
Utiliza lenguaje	Puede extender y utilizar el lenguaje y sus símbolos

**Tabla 2-1 Tabla de Atributos de un Agente Inteligente.**

Campos de la inteligencia Artificial	Tecnologías de apoyo	Aplicaciones de la IA	
Aprendizaje Automático. (Machine Learning)	Interfase de Usuario	Lingüística Computacional.	
Ingeniería del Conocimiento.(Knowlege Engineering)			
Lógica Difusa. (Fuzzy Logic)		Minería de Datos.	
Redes Neuronales Artificiales. (Artificial Neural Networks)			
Sistemas Reactivos. (Reactive Systems)		Mundos Virtuales.	
Sistemas Multi—Agente. (Multi Agent Systems)			
Sistemas Basados en Reglas. (Rule Based Systems)		Procesamiento de Lenguaje Natural.	
Razonamientos Basados en Casos. (Case Based Systems)			
Sistemas Expertos. (Expert Systems)		Visión Artificial	Robótica.
Redes Bayesianas. (Bayesians Systems)			
Vida Artificial. (Artificial Life)	Sistemas de Apoyo a la Decisión.		
Videojuegos. (Video Game)			
Computación Evolutiva. (Evolutionary Computation)	Videojuegos.		
Estrategias Evolutivas.			
Algoritmos Genéticos. (Genetic Algorithms)			
Programación Genética.			
Metaheurísticas.			
Técnicas de Representación de conocimiento.			
Redes Semánticas. (Semantic Networks)			
Frames.			

**Tabla 2-2 Tecnologías de apoyo y campos de desarrollo de la Inteligencia Artificial.**

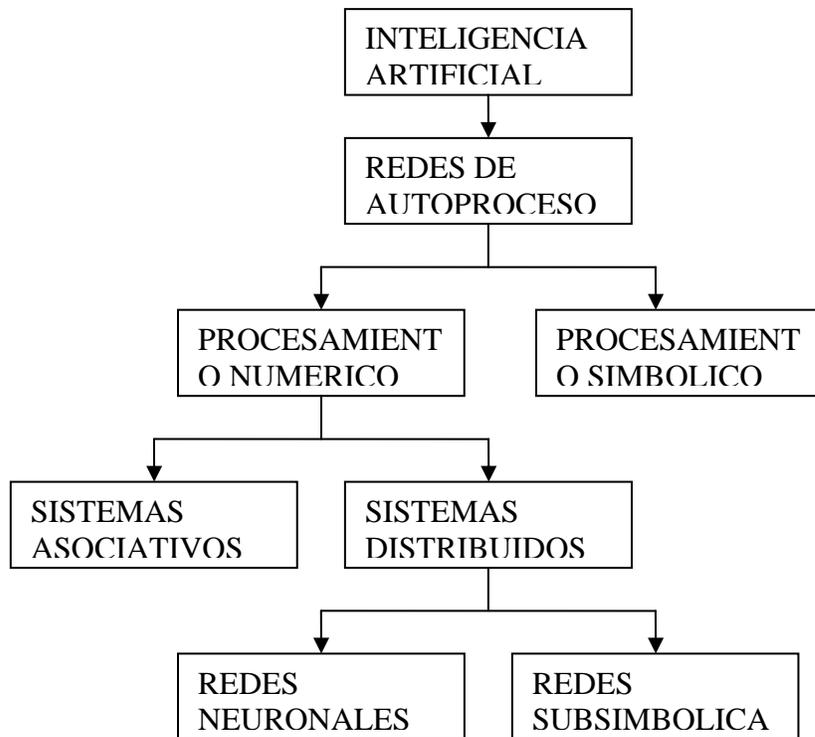
### **2.3. REDES NEURONALES ARTIFICIALES E INTELIGENCIA ARTIFICIAL**

Dentro de lo que se entiende por Inteligencia Artificial, una de las ramas más prometedoras, es la que corresponde a las denominadas *redes de auto proceso*, entendiendo como tales aquellas formadas por nodos en los que hay elementos procesadores de

información de cuyas interacciones locales depende el comportamiento del conjunto del sistema.

Como indica la Figura 2-3, dentro del campo de la Inteligencia Artificial existen dos grandes ramas de las redes de auto proceso:

- Procesamiento Numérico: Reciben directamente la señal de entrada desde el exterior y operan sobre ella. Esta rama se dedica a los sistemas constituidos por nodos, hardware interconectado entre sí, formando una red. También se les suele llamar sistemas *conectivistas* o *conexionistas*. Si la conexión entre los nodos se realiza de forma global bajo unas reglas de composición, estamos ante los llamados *sistemas distribuidos*, mientras que si la conexión se realiza agrupando en subredes, se denominan *sistemas asociativos* o redes de redes. Dentro de los sistemas distribuidos, si realizamos agrupaciones locales de los nodos para representar conceptos, estaríamos ante las denominaciones *redes subsimbólicas*; mientras que si lo que se realiza es una distribución en la que cada nodo funciona corporativamente, se denomina *red neuronal* o *neurocomputador* (cada nodo está constituido por una neurona).
- Procesamiento simbólico: Estas redes están constituidas por conceptos (nodos de la red) y por reglas sintácticas (lazos de interconexión); ambas forman las denominadas bases de conocimiento. La simulación de estas redes es casi exclusivamente software.



**FIGURA 2-3 Situación de las Redes Neuronales en el campo de la Inteligencia Artificial.**

Durante toda la década de los años 70, hasta el surgimiento con los trabajos de Hopfield, decayó el estudio de las redes neuronales y las investigaciones se orientaron al campo de la Inteligencia Artificial, gracias a lo cual se ha desarrollado una extensa teoría y unas aplicaciones que tienen gran importancia actualmente.

Por otro lado, esto no quiere decir que las teorías de redes neuronales y de Inteligencia Artificial estén separadas; de hecho, ya se ha visto cómo se puede situar e interconectar la teoría de las redes neuronales dentro del campo de la Inteligencia Artificial. Sin embargo, existen unas diferencias de las redes neuronales con respecto a lo que se entiende como Inteligencia Artificial clásica muy relevantes y que se destacan a continuación:

Para situar la teoría de redes neuronales dentro del mundo de la computación en general, y frente a la IA en particular, en la Tabla 2-3 se resume y se sintetiza parte de las ideas vistas hasta ahora acerca de las redes neuronales y las diferencias existentes respecto a la computación convencional (máquinas tipo Von Neumann) y la computación simbólica (IA). Se muestran las diferencias en función de las teorías en las que se basan, en los campos para las que son y no son apropiadas, el tipo de memoria que utilizan, cómo se construyen y qué tipo de soporte utilizan.

	<b>Computación Convencional</b>	<b>Computación Simbólica</b>	<b>Computación Neuronal</b>
<b>Basado en:</b>	Arquitectura Von Neumann	Lógica Cognitiva	Neurobiología
<b>Apropiada para:</b>	Algoritmos conocidos	Heurística	Adaptación
<b>Pero no para:</b>	Condiciones difusas	Casualidad desconocida	Cálculos precisos
<b>Memoria:</b>	Precisa, estática	Bases de conocimiento	Distribuida
<b>Construida mediante:</b>	Diseño, programación y prueba	Representación del conocimiento mas motor de inferencia	Configuración y aprendizaje
<b>Soporte:</b>	Ordenadores secuenciales	Máquinas LISP	Procesadores paralelos

**Tabla 2-3 Formas Básicas de Computación.**

## **2.4. APLICACIONES DE LAS REDES NEURONALES ARTIFICIALES [2]**

Las redes neuronales son una tecnología computacional emergente que puede utilizarse en un gran número y variedad de aplicaciones, tanto comerciales como militares. Se pueden desarrollar redes neuronales en un período de tiempo razonable y pueden realizar tareas concretas mejor que otras tecnologías convencionales, incluyendo los sistemas expertos. Cuando se implementan mediante hardware (redes neuronales en chips “Very Large Scale Integration”), presentan una alta tolerancia a fallos del sistema y proporcionan un grado de paralelismo en el proceso de datos muy grande. Eso hará posible insertar redes neuronales de bajo costo en sistemas existentes y recientemente desarrollados.

Hay muchos tipos diferentes de redes neuronales, cada uno de los cuales tiene una aplicación particular más apropiada. Se empezará relacionando ejemplos de algunas aplicaciones comerciales, separándolas en grupos según las distintas disciplinas (Ver Tabla 2-4 y Tabla 2-5):

<b>Disciplina</b>	<b>Aplicación</b>
<u>Finanzas</u>	<ul style="list-style-type: none"> <li>○ Previsión de la evolución de los precios.</li> <li>○ Valoración del riesgo de los créditos.</li> <li>○ Identificación de falsificaciones.</li> <li>○ Interpretación de firmas.</li> </ul>
<u>Manufacturación</u>	<ul style="list-style-type: none"> <li>○ Robots automatizados y sistemas de control (visión artificial y sensores de presión, temperatura, gas, etc.)</li> <li>○ Control de producción en líneas de proceso.</li> <li>○ Inspección de la calidad.</li> </ul>
<u>Medicina</u>	<ul style="list-style-type: none"> <li>○ Analizadores del habla para la ayuda de audición de sordos.</li> <li>○ Diagnóstico y tratamiento a partir de síntomas y/o de datos analíticos (electrocardiograma, encefalograma, análisis sanguíneo, etc.)</li> <li>○ Monitorización en cirugía.</li> <li>○ Predicción de reacciones adversas a los medicamentos.</li> <li>○ Lectores de rayos X.</li> <li>○ Entendimiento de la causa de los ataques epilépticos.</li> </ul>
<u>Militares</u>	<ul style="list-style-type: none"> <li>○ Clasificación de las señales de radar.</li> <li>○ Creación de armas inteligentes.</li> <li>○ Optimización del uso de recursos escasos.</li> <li>○ Reconocimiento y seguimiento en el tiro al blanco.</li> </ul>

**Tabla 2-4 Aplicaciones Comerciales de las Redes Neuronales**

<b>Disciplina</b>	<b>Aplicación</b>
Biología	<ul style="list-style-type: none"> <li>○ Aprender más acerca del cerebro y otros sistemas.</li> <li>○ Obtención de modelos de la retina.</li> </ul>
Empresa	<ul style="list-style-type: none"> <li>○ Evaluación de probabilidad de formaciones geológicas y petrolíferas.</li> <li>○ Identificación de candidatos para posiciones específicas.</li> <li>○ Explotación de bases de datos.</li> <li>○ Optimización de plazas y horarios en líneas de vuelo.</li> <li>○ Reconocimiento de caracteres escritos.</li> </ul>
Medio Ambiente	<ul style="list-style-type: none"> <li>○ Analizar tendencias y patrones.</li> <li>○ Previsión del tiempo.</li> </ul>

**Tabla 2-5 Continuación de las Aplicaciones Comerciales de las Redes Neuronales**

Se pueden buscar hechos comunes en esta lista de aplicaciones; la mayoría de ellas consisten en realizar un reconocimiento de patrones: buscar un patrón en una serie de ejemplos, clasificar patrones, completar una señal a partir de valores parciales o reconstruir el patrón correcto partiendo de uno distorsionado.

Muchos de los ejemplos tienen que trabajar con datos sensoriales y de percepción (señales visuales, auditivas y otras) y otros realizan filtrado o mapeo de señales de entrada. En general, los ejemplos mostrados indican que son característicos más de las personas que de los ordenadores convencionales.

A continuación se describen con más detalle aquella área de aplicación más importante para el cual los computadores neuronales, en general, se están utilizando.

#### **2.4.1. RECONOCIMIENTOS DE PATRONES**

El término reconocimiento de patrones originalmente se refería a la detección de formas simples, tales como caracteres escritos a mano, mapas del tiempo y espectros del lenguaje. Sin embargo, un objetivo más ambicioso ha sido durante todo el tiempo implementar la percepción artificial; es decir, imitar las funciones de los sistemas sensoriales biológicos en su forma más completa.

Los primeros experimentos, alrededor de 1960, estaban basados en redes neuronales elementales, como **Perceptrón**, **ADALINE** y **Matrices de Aprendizaje**. Los primeros pasos, como siempre, fueron fáciles, pero pronto se vio que la realización de los sistemas biológicos era muy difícil de alcanzar. Incluso la alta capacidad de computación alcanzable por circuitos de computación paralela no resolvía los problemas.

Especialmente en el análisis de imágenes existen requerimientos que son muy difíciles de alcanzar: a) Invariabilidad de la detección con respecto a la traslación, rotación, escala, perspectiva, oclusión parcial y casamiento simple de los objetos, especialmente en condiciones de iluminación variables, y b) relación de observaciones de varios contextos a niveles diferentes de abstracción, en orden a distinguir los eventos más selectivamente. Hay que destacar que los animales son capaces de prestar atención a objetos individuales en una escena, por cada uno de los cuales la invariabilidad de percepción debe ser válida

separadamente. Esto por lo que habría que tenerlo en cuenta para obtener la solución deseada.

Lo que también era ignorado, e que incluso el más desarrollado sistema sensorial biológico no opera de forma autónoma: la percepción sensorial está siempre muy unida con el *proceso cognitivo global*. Para la replicación de las funciones sensoriales, no es suficiente con imitar el sistema sensorial, sino que se tiene que replicar todo el cerebro con todas sus capacidades de pensamiento, y precisar la capacidad de reconocimiento por el alto grado de aprendizaje.

No sería razonable esperar la solución de todos estos problemas antes de proceder a la obtención de las posibles aplicaciones. En ingeniería, los problemas son generalmente simplificados. Tomemos, por ejemplo, la locomoción: es difícil implementar el movimiento coordinado de los miembros, pero un método más directo es aplicar ruedas y modificar el terreno construyendo carreteras. Igualmente, existen multitud de aplicaciones para las cuales soluciones artificiales, no naturales, pueden ser incluso más eficaces.

Las áreas de aplicación más importantes del reconocimiento de patrones neuronal serían las mismas que aquellas para las cuales los métodos convencionales y heurísticos han sido desarrollados durante los pasados treinta años.

- Sensación remota.

---

<sup>5</sup> Transformada de Fourier: Es un cálculo que equivale a la transformada de Laplace con argumentos complejos.

<sup>6</sup> Transformada de Mellin: Es un cálculo que equivale a la transformada de Laplace Bilateral.

- Análisis de Imágenes médicas.
- Visión en computadores industriales (especialmente para robots)
- Elementos de proceso de las entradas para computadores.

Tareas más concretas para las cuales se han desarrollado ya equipos de computadores especiales son:

- Segmentación y clasificación de regiones de imágenes.
- Reconocimiento de caracteres escritos (manuales y texto impreso).
- Reconocimiento del habla.
- Procesamiento y restauración de imágenes con ruido.
- En un nivel más ambicioso, se puede intentar lograr capacidades como:
- Análisis de imágenes (en lo referente a diferentes niveles temáticos de abstracción, tales como monitorización de uso terrestre en las bases de imágenes por satélite).
- Reconocimiento de imágenes (interpretación de escenas).
- Reconocimiento de habla (análisis e interpretación de frases habladas)

Para implementar estas tareas, existen todavía ciertos problemas básicos pendientes de solución; por ejemplo, aquellos referentes a las propiedades intrínsecas de la información de entrada, tales como:

- Los patrones primitivos más naturales (líneas, sus curvaturas y puntos final, bordes, estadísticas de grupos de puntos).
- Información visual que describen las curvaturas de las superficies.
- Textura.
- Invariantes fonológicas en el habla.

Por otro lado, la integración de estas funciones en un sistema de conocimiento de alto nivel es un objetivo cuya complejidad es de un orden de magnitud mayor de lo que generalmente se cree. Por tanto, parecen existir multitud de aplicaciones potenciales, pero todavía existen problemas no resueltos. No parece razonable continuar el desarrollo de métodos heurísticos sofisticados: se estima que sobre 30,000 artículos y unos 200 libros de texto se han escrito ya sobre técnicas de reconocimiento de patrones y visión artificial en los últimos cuarenta años, y si las soluciones hubieran estado al alcance por este camino, ya hubieran sido visibles.

## **2.5. TIPOLOGIA DE REDES NEURONALES MÁS RELEVANTES. [2]**

A continuación se muestran unas tablas resumen (Tabla 2.6 y Tabla 2.7) de los tipos de redes más conocidos, comentando esquemáticamente las aplicaciones más importantes

de cada una, sus ventajas e inconvenientes, así como ¿quién o quiénes? fueron sus creadores y ¿en qué año?.

NOMBRE DE LA RED	AÑO	APLICACIONES MAS IMPORTANTES	COMENTARIOS	LIMITACIONES	INVENTADA / DESARROLLADA POR:
PERCEPTRON	1957	RECONOCIMIENTO DE CARACTERES IMPRESOS	LA RED MAS ANTIGUA, CONSTRUIDA EN HW	NO PUEDE RECONOCER CARACTERES COMPLEJOS	FRANK ROSENBLATT
ADALINE / MADALINE	1960	FILTRADO DE SEÑALES. ECUALIZADOR ADAPTATIVO, MODEMS.	RAPIDA, FACIL DE IMPLEMENTAR CON CIRCUITOS ANALOGOS O VLSI	SOLO ES POSIBLE CLASIFICAR ESPACIOS LINEALMENTE SEPARADOS	BERNARD WIDROW.
AVALANCHA	1967	RECONOCIMIENTO DE HABLA CONTINUA. CONTROL DE BRAZOS ROBOT.	NINGUNA RED SENCILLA PUEDE HACER TODO ESTO	NO ES FACIL ALTERAR LA VELOCIDAD O INTERPOLAR EL MOVIMIENTO	STEPHEN GROSSBERG
CERE - BELLATRON	1969	CONTROL DEL MOVIMIENTO DE LOS BRAZOS DE UN ROBOT	SEMEJANZA A AVALANCHA	REQUIERE COMPLICADAS ENTRADAS DE CONTROL	DAVID MARR, JAMES ALBUS, ANDRES PELLIONEZ.
BACK - PROPAGATION	1974 - 1985	SINTESIS DE VOZ DESDE DE TEXTO. CONTROL DE ROBOTS. PREDICION. RECONOCIMIENTO DE PATRONES	RED MAS POPULAR, NUMEROSAS APLICACIONES CON ÉXITO. FACILIDAD DE APRENDIZAJE, POTENTE.	NECESITA MUCHO TIEMPO PARA APRENDIZAJE Y MUCHOS EJEMPLOS.	PAUL WERBOS, DAVID PARKER, DAVID RUMELHART.
BRAIN - ESTATE - IN - A - BOX	1977	EXTRADICION DE CONOCIMIENTOS DE BASES DE DATOS	POSIBLEMENTE MEJOR REALIZACION QUE LAS REDES DE HOPFIELD	REALIZACION Y POTENCIALES APLICACIONES NO ESTUDIADAS TOTALMENTE.	JAMES ANDERSON
NEO - COGNITRON	1978 - 1984	RECONOCIMIENTO DE CARACTERES MANUSCRITOS	INSENSIBLE A LA TRANSLACION, ROTACION Y ESCALA	REQUIERE MUCHOS ELEMENTOS DE PROCESO, NIVELES Y CONEXIONES	K.FUKUSHIMA

**Tabla 2-6 Resumen de la Creación de Redes Neuronales Artificiales**

NOMBRE DE LA RED	AÑO	APLICACIONES MAS IMPORTANTES	COMENTARIOS	LIMITACIONES	INVENTADA / DESARROLLADA POR:
SELF-ORGANIZING-MAP (SOM). TOPOLOGY-PRESERVING-MAP (TPM).	1980 - 1984	RECONOCIMIENTO DE PATRONES, CODIFICACION DE DATOS, OPTIMIZACION.	REALIZA MAPAS DE CARACTERISTICAS COMUNES DE LOS DATOS APRENDIDOS	REQUIERE MUCHO ENTRENAMIENTO	TEUVO KOHONEN
HOPFIELD	1982	RECONSTRUCCION DE PATRONES Y OPTIMIZACION	PUEDE IMPLEMENTARSE EN VLSI. FACIL DE CONCEPTUALIZAR	CAPACIDADES Y ESTABILIDAD	JOHN HOPFIELD
MEMORIA ASOCIATIVA BIDIRECCIONAL	1985	MEMORIA HETEROASOCIATIVA DE ACCESO POR CONTENIDO	APRENDIZAJE Y ARQUITECTURA SIMPLE.	BAJA CAPACIDAD DE ALMACENAMIENTO. LOS DATOS DEBEN SER CODIFICADOS.	BART KOSKO.
COUNTER-PROPAGATION	1986	COMPRESION DE IMÁGENES	COMBINACION DE PERCEPTRON Y TPM	NUMEROSAS NEURONAS Y CONEXIONES	ROBERT HECHTNIELSEN
MAQUINAS DE BOLTZMANN Y CAUCHY	1985 - 1986	RECONOCIMIENTO DE PATRONES (IMÁGENES, SONAR Y RADAR). OPTIMIZACION.	REDES SIMPLES, CAPACIDADES DE REPRESENTACION OPTIMA DE PATRONES.	LA MAQUINA DE BOLTZMANN NECESITA UN TIEMPO MUY LARGO DE APRENDIZAJE	JEFFREY HINTON, TERRY SEJNOWSKI, HAROLD SZU.
TEORIA RESONANCIA ADAPTATIVA (ART)	1986	RECONOCIMIENTO DE PATRONES (RADAR, SONAR, ETC)	SOFISTICADA POCO UTILIZADA	SENSIBLE A LA TRANSLACION DISTORSION Y ESCALA	GAIL CARPENTER, STEPHEN GROSSBERG

**Tabla 2-7 Continuación del Resumen de la Creación de Redes Neuronales Artificiales**

## **2.6. ELEMENTOS DE UNA NEURONA ARTIFICIAL**

### **2.6.1. ELEMENTOS DE UNA NEURONA ARTIFICIAL. [2].**

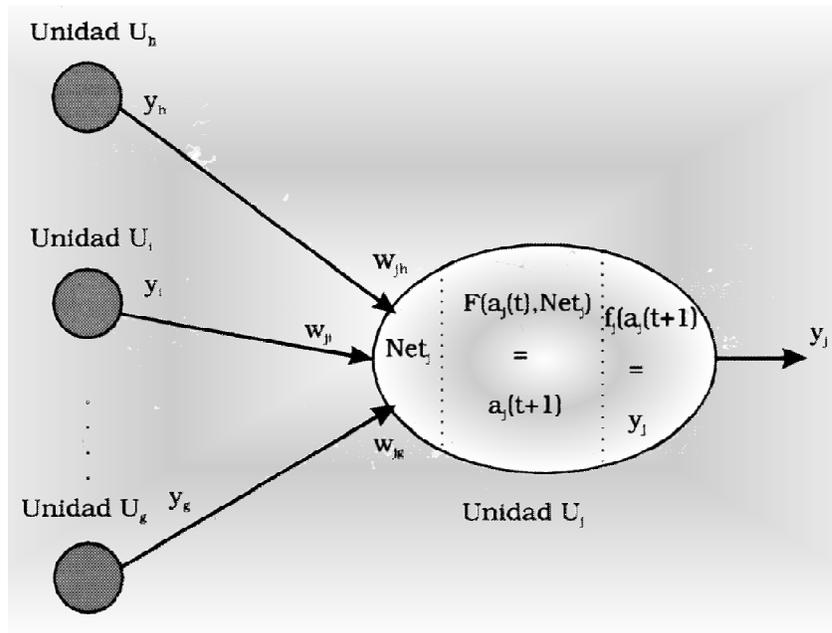
Las redes neuronales son modelos que intentan reproducir el comportamiento del cerebro. Como tal modelo, realiza una simplificación, averiguando cuáles son los elementos relevantes del sistema, bien por la cantidad de información de que se dispone es excesiva o bien porque es redundante.

El siguiente apartado realiza una idealización del funcionamiento neurobiológico, de un conjunto de neuronas vivas, que sirve de base para las redes neuronales artificiales (RNA). Generalmente, se pueden encontrar tres tipos de neuronas.

Las que toman la información de las entradas, es decir, que recibe estímulos externos, relacionados con el aparato sensorial. Se encuentran también las unidades ocultas, las cuales transmiten la información obtenida de las entradas a elementos internos que se ocupan de procesarlos.

Una vez llegada la información al final del proceso, ésta llega a las unidades de salida, cuya función es la de dar respuesta al sistema, mediante unas funciones matemáticas, la cual responde dependiendo del proceso realizado anteriormente.

Estas características, son las que pretende imitar una neurona artificial. En cada momento una neurona esta caracterizada por un valor numérico denominado estado de activación, asociado a cada unidad, existe una función de salida, que transforma el estado actual de activación de la señal, de cada una de las unidades.



**Figura 2-4 Forma General de una Neurona Artificial.**

### 2.6.1.1.FUNCIONES DE ACTIVACIÓN

Entre las unidades o neuronas que forman una red neuronal artificial existe un conjunto de conexiones que unen unas con otras. Cada unidad transmite señales a aquellas que están conectadas con su salida. Asociada con cada unidad  $U_i$  hay una función de salida  $f_i(a_i(t))$ , que transforma el estado actual de activación  $a_i(t)$  en una señal de salida  $y_i(t)$ ; es decir:

$$y_i(t) = f_i(a_i(t)) \quad \text{Ecuación 2-1}$$

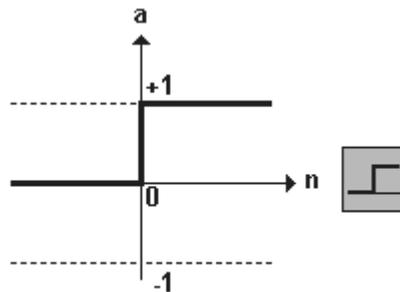
El vector que contiene todas las salidas en un instante t es:

$$Y(t) = (f_1(a_1(t)), f_2(a_2(t)), \dots, f_i(a_i(t)), \dots, f_n(a_n(t))) \quad \text{Ecuación 2-2}$$

En algunos modelos, ésta salida es igual al nivel de activación de la unidad, en cuyo caso la función de activación  $f_i$  es la función identidad  $f_i(a_i(t)) = a_i(t)$ . A menudo,  $f_i$  es de tipo sigmoïdal, y suele ser la misma para todas las unidades.

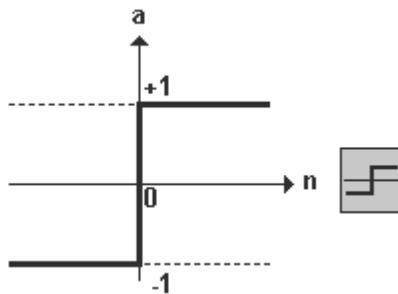
Existen cuatro funciones de transferencia típicas que determinan distintos tipos de neuronas:

- Función escalón Polar



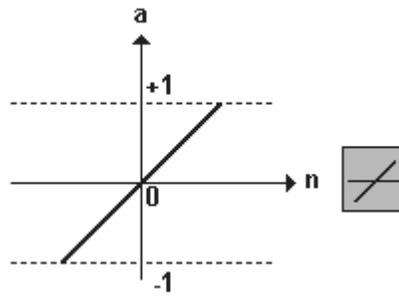
**Figura 2-5 Función de Transferencia Escalón Polar.**

- Función escalón Bipolar



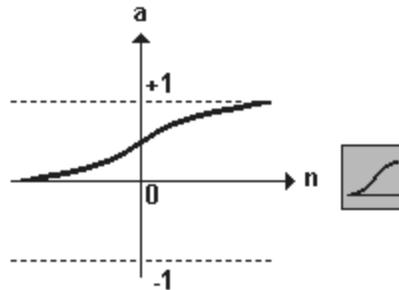
**Figura 2-6 Función de Transferencia Escalón Bipolar**

- Función lineal



**Figura 2-7 Función de Transferencia Lineal.**

- Función Sigmoidal



**Figura 2-8 Función de Transferencia Sigmoidal.**

Las funciones escalón, únicamente se utilizan cuando las salidas de la red son binarias (dos posibles valores). La salida de una neurona se activa solo cuando el estado de activación es mayor o igual que cierto valor de umbral (las funciones pueden estar desplazadas de su eje), la función lineal equivale a no aplicar función de salida, se usa muy

poco. La función sigmoideal es la más apropiada cuando se quiere como salida información analógica.

## 2.6.2. TOPOLOGIAS IMPLEMENTADAS. [1]-[2]-[3].

### 2.6.2.1. RED PERCEPTRON

Fue el primer modelo de red neuronal artificial desarrollado en 1958 por Rosenblatt, el cual tiene la capacidad de aprender a reconocer patrones sencillos.

Está formado por varias neuronas para recibir las entradas a la red ( $X$ ) y una neurona, la cual hace el procesamiento de los datos, es decir, la multiplicación de las entradas por su peso ( $W$ ) respectivo. Además, tiene una salida ( $Y$ ) que es capaz de decidir cuando una entrada a la red pertenece a una de las dos clases que es capaz de reconocer.

Ver figura 2-9.

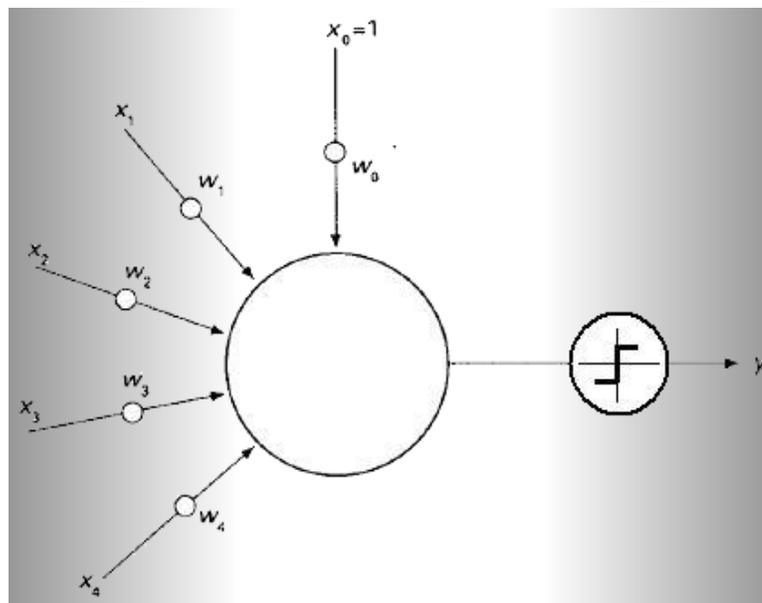


Figura 2-9 Representación del Perceptrón de Rosenblatt

### 2.6.2.1.1. ALGORITMO DE APRENDIZAJE

Típicamente el perceptrón utiliza una función de activación binaria de +1,0 o -1, para las unidades de salida.

$$f(Y\_in) = \begin{cases} 1 & \text{If } Y\_in > q \\ 0 & \text{if } -q \leq Y\_in \leq q \\ -1 & \text{if } Y\_in < -q \end{cases} \quad \text{Ecuación 2-3}$$

Donde  $Y\_in$  es la variable que contiene la adición de las multiplicaciones de las entradas con los pesos respectivos. La función del parámetro  $\theta$  es la de desplazar la gráfica del centro cartesiano.

El algoritmo que se presenta a continuación es conveniente para vectores de entradas binarias o bipolares, con salidas de la misma forma, bipolares. Además, se encuentra un parámetro adicional,  $\theta$ , el cual se le conoce como umbral, y afecta después del procesamiento de los datos, desplazando hacia la derecha o hacia la izquierda del plano cartesiano.

El algoritmo, no es esencialmente sensible a valores iniciales de pesos, o de valores de la razón de aprendizaje. A continuación se presenta el algoritmo a seguir:

*Paso 0:*

Inicializar pesos de las entradas y del bias. (Por simplicidad, se pueden tomar a 0).

Inicializar el valor de la razón de aprendizaje,  $0 < \alpha < 1$ . (Por simplicidad se puede tomar como 1).

*Paso 1:*

Mientras la condición sea falsa, hacer los pasos del 2 al 6.

*Paso 2:*

Para cada par de entrenamiento (S:T), hacer los pasos 3 al 5.

*Paso 3:*

Activar cada una de las unidades de entrada de la neurona con el valor de las entradas, desde  $i = 0$  hasta el número de entradas:

$$X_i = S_i \quad \text{Ecuación 2-4}$$

*Paso 4:*

Obtener el acumulado de las multiplicaciones de las entradas con sus respectivos pesos:

$$Y_{in} = b * W_0 + \sum_i X_i * W_i \quad \text{Ecuación 2-5}$$

Obtener la salida de la función

$$Y = \begin{cases} 1 & \text{If } Y\_in > q \\ 0 & \text{if } -q \leq Y\_in \leq q \\ -1 & \text{if } Y\_in < -q \end{cases} \quad \text{Ecuación 2-6}$$

*Paso 5:*

Actualizar pesos, si la salida no es la deseada, para el patrón actual:

Si  $Y \neq T$  entonces:

$$\begin{aligned} W_i(\text{new}) &= W_i(\text{old}) + a * T * X_i \\ W_o(\text{new}) &= W_o(\text{old}) + a * T \end{aligned} \quad \text{Ecuación 2-7}$$

Si no:

$$\begin{aligned} W_i(\text{new}) &= W_i(\text{old}) \\ W_o(\text{new}) &= W_o(\text{old}) \end{aligned} \quad \text{Ecuación 2-8}$$

*Paso 6:*

Verificar la condición de paro:

Si los pesos no han cambiado en el paso 2, pare; de lo contrario continúe.

Fin del algoritmo.

Es de notar que los pesos son actualizados solo cuando una entrada no es nula ( $X \neq 0$ ). Así como también, solo se actualizan los pesos que no producen un valor correcto de salida,  $Y$ .

El umbral,  $\theta$ , se utiliza para ubicar valores de entrada, que no se sabe en que rango ponerlo, es decir, la red no logra clasificar dichas entradas. Esto sucede cuando no se han utilizado todos los vectores de entrada posibles para un buen entrenamiento.

### 2.6.2.2. RED MULTIPLE PERCEPTRON (M - PERCEPTRON)

Un Perceptrón múltiple es una red de tipo *feedforward* compuesta de varias capas de neuronas entre la entrada y la salida de la misma. Esta red permite establecer regiones de decisión mucho más complejas que las de dos semiplanos, como lo hace el Perceptrón simple.

La capacidad del Perceptrón con dos, tres y cuatro niveles o capas y con una única neurona de salida, se muestra en la figura 2-11. Se puede observar, que las regiones que se forman para el problema de la XOR, van siendo cada vez mas definidas. De igual manera cuando se tiene regiones mezcladas, dichas regiones se van definiendo de mejor manera entre mas capas tenga la estructura.

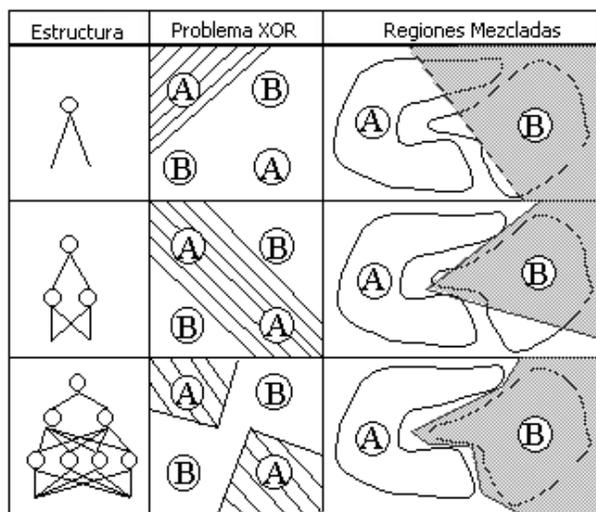


Figura 2-10 Distintas formas de las regiones generadas por un Perceptrón Multinivel

La estructura de un Perceptrón multinivel (o múltiple) es la siguiente:

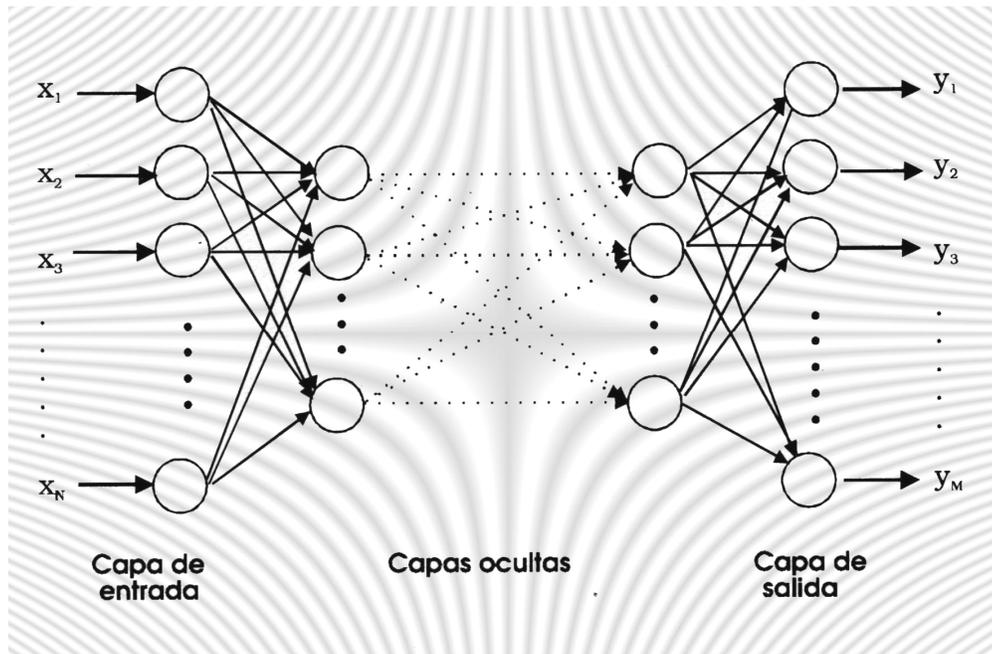


Figura 2-11 Perceptrón Multinivel

El aprendizaje que se suele usar en este tipo de redes recibe el nombre de retro propagación del error. Como función de costo global, se usa el error cuadrático medio.

#### 2.6.2.2.1. ALGORITMO DE APREDNDIZAJE

*Paso 0:*

Inicializar los pesos con valores aleatorios pequeños.

*Paso 1:*

Mientras la condición sea falsa, hacer los pasos del 2 al 9.

*Paso 2:*

Para cada par de entrenamiento (S:T), hacer los pasos 3 al 8.

*Paso 3:*

Activar cada una de las unidades de entrada de la neurona con el valor de las entradas, desde  $i = 0$  hasta el número de entradas.

$$X_i = S_i \quad \text{Ecuación 2-9}$$

*Paso 4:*

Para cada unidad oculta realizar la suma de las multiplicaciones de los pesos con las entradas:

$$Z_{in_j} = b * V_{0j} + \sum_{i=1}^n X_i * V_{ij} \quad \text{Ecuación 2-10}$$

Aplicando la función de activación, se obtiene:

$$Z_j = F(Z_{in_j}) \quad \text{Ecuación 2-11}$$

Y envía dicho valor a toda la capa siguiente.

*Paso 5:*

Para cada unidad de salida, se obtiene la suma de las multiplicaciones de los pesos con las entradas ( $Z_{in}$ ):

$$Y_{in_k} = b * W_{0k} + \sum_{j=1}^P Z_j * W_{jk} \quad \text{Ecuación 2-12}$$

Aplicando la función de activación, se obtiene:

$$Y_k = F(Y_{in_k}) \quad \text{Ecuación 2-13}$$

## Retro propagación del Error

### Paso 6:

Para cada unidad de salida ( $Y_K$ ), encuentre el error correspondiente a:

$$d_k = (T_K - Y_K) * F'(Y_{in_k}) \quad \text{Ecuación 2-14}$$

Calcular la corrección para los pesos, se usará para corregir  $W_{jk}$ :

$$\Delta W_{JK} = a d_k Z_j \quad \text{Ecuación 2-15}$$

Calcular la corrección para los pesos de los bias, se usará para corregir  $W_{0K}$ :

$$\Delta W_{0K} = a d_k b \quad \text{Ecuación 2-16}$$

### Paso 7:

Para cada unidad oculta ( $Z_j$ ), encuentre el error correspondiente a:

$$d_{in_j} = \sum_{K=1}^m d_k w_{jk} \quad \text{Ecuación 2-17}$$

Multiplicar dicho valor por la derivada de la función y encontrar el error correspondiente:

$$d_j = d_{in_j} * F'(Z_{in_j}) \quad \text{Ecuación 2-18}$$

Calcular la corrección para los pesos, se usará para corregir  $V_{ij}$ :

$$\Delta V_{ij} = a d_j X_i \quad \text{Ecuación 2-19}$$

Calcular la corrección para los pesos de los bias, se usará para corregir  $V_{0j}$ :

$$\Delta V_{0j} = a d_j b \quad \text{Ecuación 2-20}$$

## Actualizar Pesos y Bias

### Paso 8:

Para cada unidad de salida ( $Y_k$ ), actualizar los pesos y los bias.

$$\Delta W_{jk} = W_{jk} + ad_k Z_j$$

**Ecuación 2-21**

Para cada unidad oculta ( $Z_j$ ), actualizar los pesos y los bias

$$\Delta V_{ij} = V_{ij} + ad_j X_i$$

**Ecuación 2-22**

*Paso 9:*

Verificar la condición de paro, si el error es menor al error cuadrático medio, entonces parar, de lo contrario seguir entrenando.

### **2.6.2.3. RED ADALINE**

ADALINE (**AD**Aptative **L**inear **NE**uron), creada por Widrow y Hoff en 1960, en la Universidad de Stanford. Es una de las Redes Neuronales Artificiales más simples, junto con el Perceptrón. Y fue creada como un circuito adaptativo para uso en telefonía, mucho antes de que se concibiera el término “Red Neuronal Artificial”.

La arquitectura de este tipo de redes es similar a la del perceptrón (ver Figura 2-9), con la diferencia de que este permite manejar salidas análogas, ya que la salida se toma antes de pasar por la función de salida. Otra diferencia radica en el proceso de aprendizaje justo en la toma de decisión de cuando modificar los pesos.

El mecanismo de aprendizaje de la red ADALINE es denominado regla Delta de Hidrow-Hoff o regla del mínimo error cuadrático medio (LMS), el cual se basa en la búsqueda del mínimo error entre la salida deseada y la salida lineal obtenida antes de aplicarle la función de activación escalón.

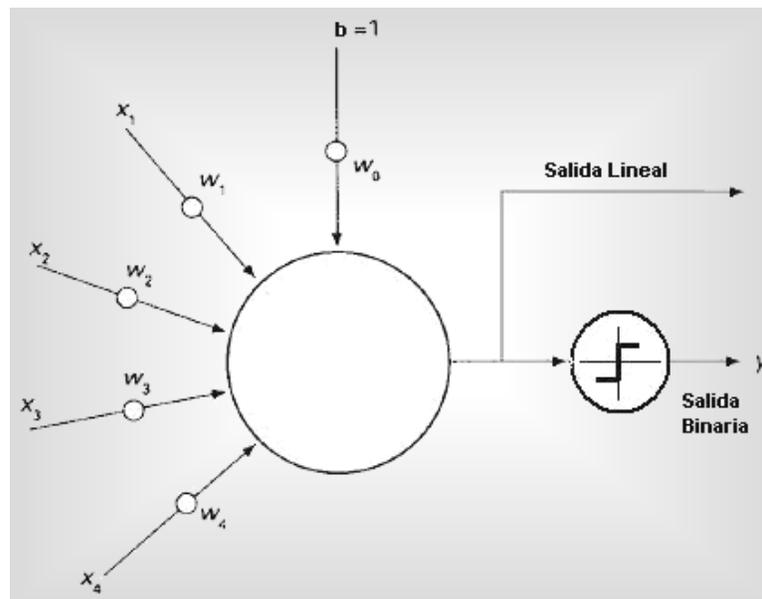


Figura 2-12 Representación de la red ADALINE

### 2.6.2.3.1. ALGORITMO DE APRENDIZAJE

Paso 0:

Inicializar pesos de las entradas y del bias, (se pueden tomar aleatoriamente). A su vez, se inicializa un valor de tolerancia para detener el entrenamiento. Dicha tolerancia servirá para definir si los valores de los pesos han cambiado en una época.

Inicializar el valor de la razón de aprendizaje,  $0 \leq \alpha \leq 1$ . (Por simplicidad se puede tomar como 1).

Paso 1:

Mientras la condición sea falsa, hacer los pasos del 2 al 6.

Paso 2:

Para cada par de entrenamiento (S:T), hacer los pasos 3 al 5.

Paso 3:

Activar cada una de las unidades de entrada de la neurona con el valor de las entradas, desde  $i = 0$  hasta el número de entradas.

$$X_i = S_i$$

**Ecuación 2-23**

Paso 4:

Obtener la respuesta de la unidad de salida:

$$Y_{in} = b * W_0 + \sum_i X_i * W_i$$

**Ecuación 2-24**

Paso 5:

Actualizar pesos, si la salida no es la deseada, para el patrón actual, desde  $i=0$  hasta el número de entradas:

$$\begin{aligned} W_i(\text{new}) &= W_i(\text{old}) + a * (T - Y_{in}) * X_i \\ W_o(\text{new}) &= W_o(\text{old}) + a * (T - Y_{in}) \end{aligned} \quad \text{Ecuación 2-25}$$

Paso 6:

Verificar la condición de paro:

$$d^2 = \frac{1}{2L} \sum_{k=1}^L d_k^2 \quad \text{Ecuación 2-26}$$

Donde  $\delta$  es el error esperado en una época, y  $\delta_k$  es el error que se obtiene con cada vector de entrada en cada época.

Si el cambio más grande del peso que ocurre en el paso 2, es más pequeño que la tolerancia específica, entonces pare; de lo contrario continúe.

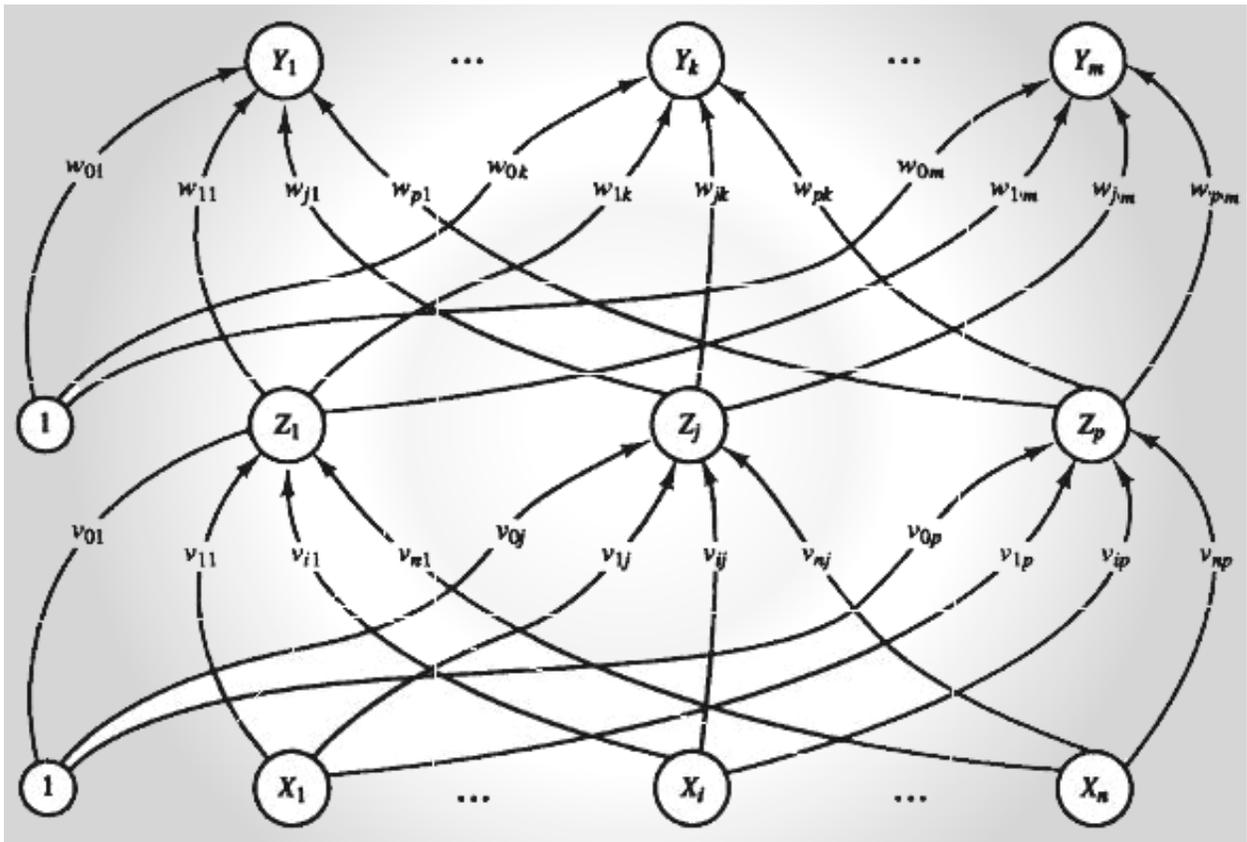
#### **2.6.2.4. RED BACKPROPAGATION. [1]**

Creada en 1986 por Rumelhart, Hinton y Williams, esta basada en la propagación del error hacia atrás.

Básicamente se tiene dos comportamientos en su etapa de aprendizaje:

- Propagación de los patrones de entrada hacia las salidas, pasando por las neuronas ocultas;
- Propagación del error hacia atrás, con la consiguiente modificación de los pesos.

Es de hacer notar que la red de retro propagación se basa en la aproximación al valor mas cercano de los pesos utilizando el gradiente y la única restricción que tiene es que las funciones de salida que se utilicen en las neuronas deben ser derivables.



**Figura 2-13 Representación de la Red Back-Propagation Standard.**

La cantidad de neuronas ocultas o de capas ocultas depende de la aplicación en la que se este trabajando, por lo general la mayoría de autores coinciden en que basta con una capa de neuronas ocultas para realizar el aprendizaje de esta red y recomiendan que se pruebe con varias cantidades de neuronas en la capa oculta. Ver figura 2-14.

#### **2.6.2.4.1. ALGORITMO DE APRENDIZAJE**

Cada función de activación definida con anterioridad puede ser usada en el algoritmo de entrenamiento. La forma de los datos (especialmente la salida deseada), es un factor muy importante para definir la función de activación.

*Paso 0:*

Inicializar los pesos con valores aleatorios pequeños (sugerencia, entre 0.05 y 0.25)

*Paso 1:*

Mientras la condición sea falsa (error cuadrático medio obtenido), hacer los pasos del 2 al 9.

*Paso 2:*

Para cada par de entrenamiento (S:T), hacer los pasos 3 al 8.

*Paso 3:*

Activar cada una de las unidades de entrada de la neurona con el valor de las entradas, desde  $i=0$  hasta el número de entradas.

$$X_i = S_i \quad \text{Ecuación 2-27}$$

*Paso 4:*

Para cada unidad oculta realizar la suma de las multiplicaciones de los pesos con las entradas:

$$Z_{in_j} = b * V_{0j} + \sum_{i=1}^n X_i * V_{ij} \quad \text{Ecuación 2-28}$$

Aplicando la función de activación, se obtiene:

$$Z_j = F(Z_{in_j}) \quad \text{Ecuación 2-29}$$

Y envía dicho valor a toda la capa siguiente.

*Paso 5:*

Para cada unidad de salida, se obtiene la suma de las multiplicaciones de los pesos con las entradas ( $Z_{in}$ ):

$$Y_{in_k} = b * W_{0k} + \sum_{j=1}^P Z_j * W_{jk} \quad \text{Ecuación 2-30}$$

Aplicando la función de activación, se obtiene:

$$Y_K = F(Y_{in_K})$$

**Ecuación 2-31**

*Retro propagación del Error*

*Paso 6:*

Para cada unidad de salida ( $Y_K$ ), encuentre el error correspondiente a:

$$d_k = (T_K - Y_K) * F'(Y_{in_K})$$

**Ecuación 2-32**

Calcular la corrección para los pesos, se usará para corregir  $W_{jk}$ :

$$\Delta W_{JK} = ad_k Z_j$$

**Ecuación 2-33**

Calcular la corrección para los pesos de los bias, se usará para corregir  $W_{0K}$ :

$$\Delta W_{0K} = ad_k b$$

**Ecuación 2-34**

*Paso 7:*

Para cada unidad oculta ( $Z_J$ ), encuentre el error correspondiente a:

$$d_{in_J} = \sum_{K=1}^m d_k w_{jk}$$

**Ecuación 2-35**

Multiplicar dicho valor por la derivada de la función y encontrar el error correspondiente:

$$d_j = d_{in_j} * F'(Z_{in_j}) \quad \text{Ecuación 2-36}$$

Calcular la corrección para los pesos, se usará para corregir  $V_{ij}$ :

$$\Delta V_{ij} = ad_j X_i \quad \text{Ecuación 2-37}$$

Calcular la corrección para los pesos de los bias, se usará para corregir  $V_{0j}$ :

$$\Delta V_{0j} = ad_j b \quad \text{Ecuación 2-38}$$

### *Actualizar Pesos y Bias*

*Paso 8:*

Para cada unidad de salida ( $Y_k$ ), actualizar los pesos y los bias.

$$\Delta W_{jk} = W_{jk} + ad_k Z_j \quad \text{Ecuación 2-39}$$

Para cada unidad oculta ( $Z_j$ ), actualizar los pesos y los bias

$$\Delta V_{ij} = V_{ij} + ad_j X_i \quad \text{Ecuación 2-40}$$

*Paso 9:*

Verificar la condición de paro, si el error es menor al error cuadrático medio, entonces parar, de lo contrario seguir entrenando.

$$d^2 = \frac{1}{2L} \sum_{k=1}^L d_{k,i}^2$$

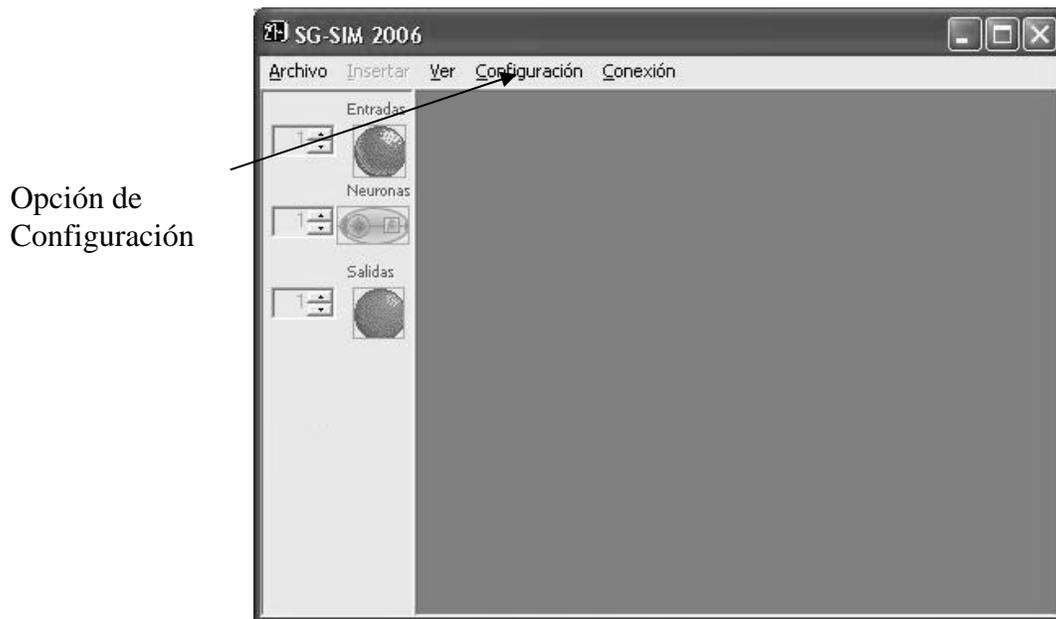
**Ecuación 2-41**

## Capítulo 3 - AMBIENTE GRAFICO

---

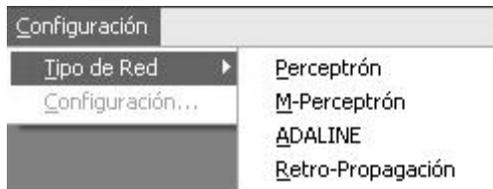
### 3.1. AMBIENTE GRAFICO BÁSICO

A continuación se presenta el primer prototipo del ambiente gráfico del software a desarrollar. La Figura 3-1 muestra la primera ventana que el usuario vería después de haber iniciado el programa.



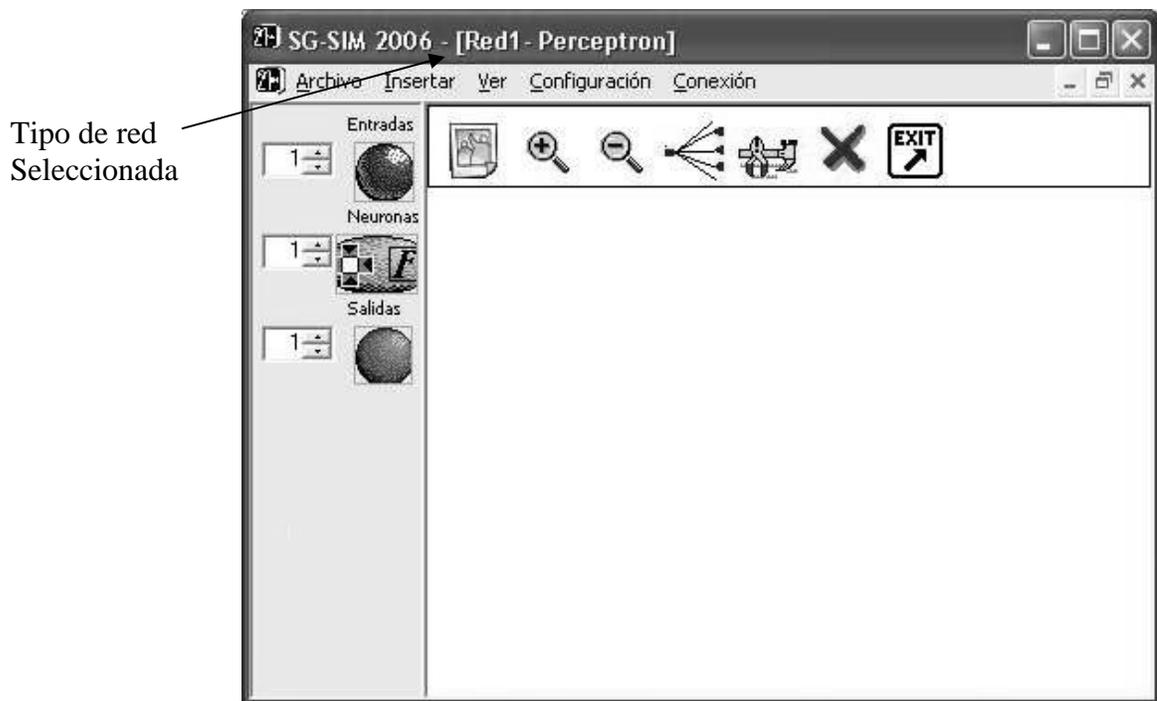
**Figura 3-1 Ventana Inicial.**

El primer paso que se debe de hacer, es seleccionar la topología en la que se va a trabajar. Para ello es necesario seleccionar de la barra de menú, la opción de configurar / tipo de red; de la siguiente manera, ver Figura 3-2:



**Figura 3-2 Configuración inicial, tipo de red.**

Al escoger cualquier tipo de red, se activa la ventana para poder empezar a dibujar la red que el usuario necesite. Ver Figura 3-3:



**Figura 3-3 Ventana Inicial Activa.**

Una vez seleccionado el tipo de red, en la parte superior de la misma ventana, aparecerá el nombre de la topología escogida, tal como se muestra en la Figura 3-3.

### 3.1.1. AMBIENTE GRAFICO DE LA RED PERCEPTRON

Después de haber seleccionado el tipo de red, Perceptrón, se procede a crear dicha estructura. Presionando el botón de entrada, se crea una entrada correspondiente a la cantidad puesta en la caja numérica, así como se muestra en la Figura 3-4.

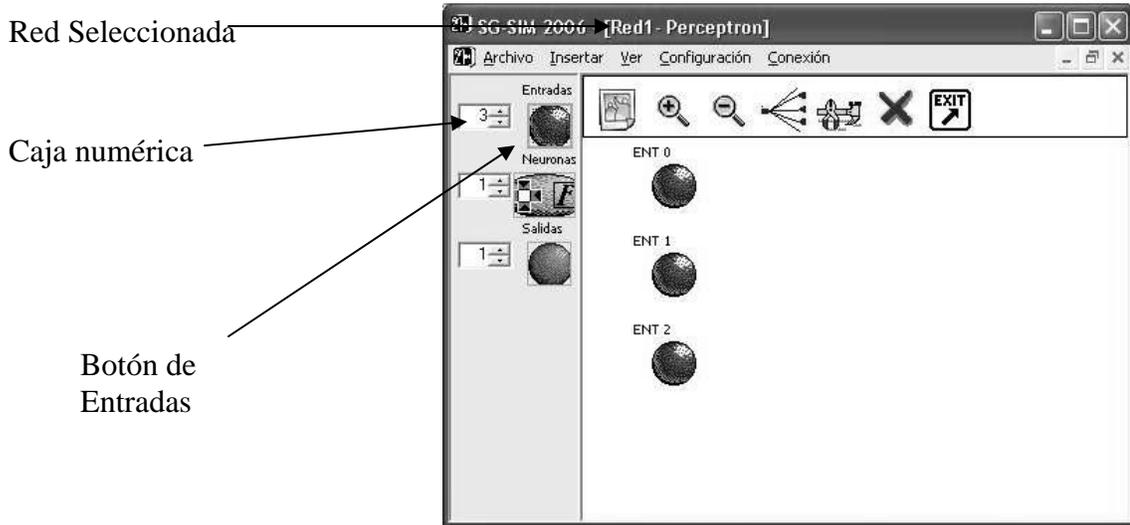


Figura 3-4 Ventana activa con Red Perceptrón. Añadir Entradas

Luego de haber creado las entradas, se procede a crear la neurona que hará el procesamiento de los datos. Por ser un Perceptrón simple, solo se permite la entrada de una sola neurona. Ver Figura 3-5.

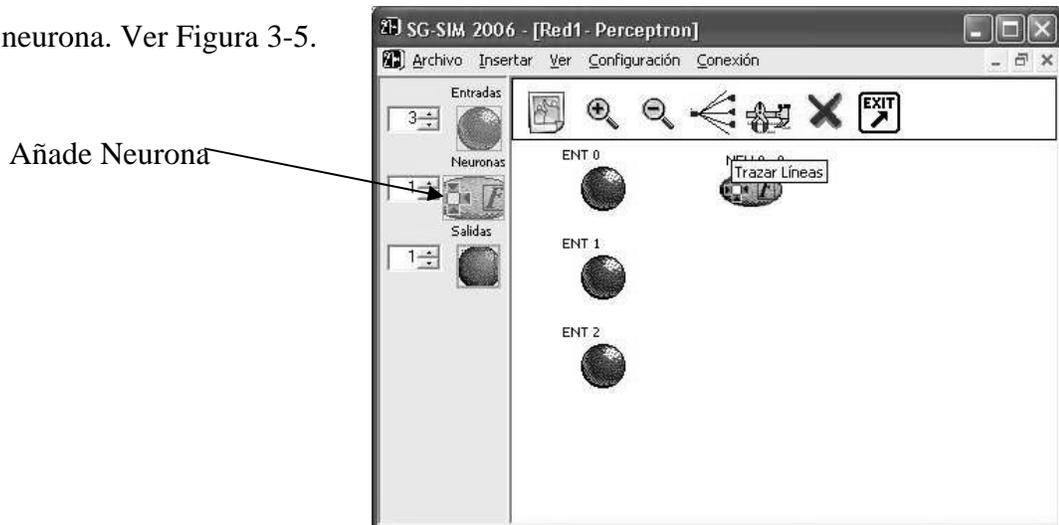
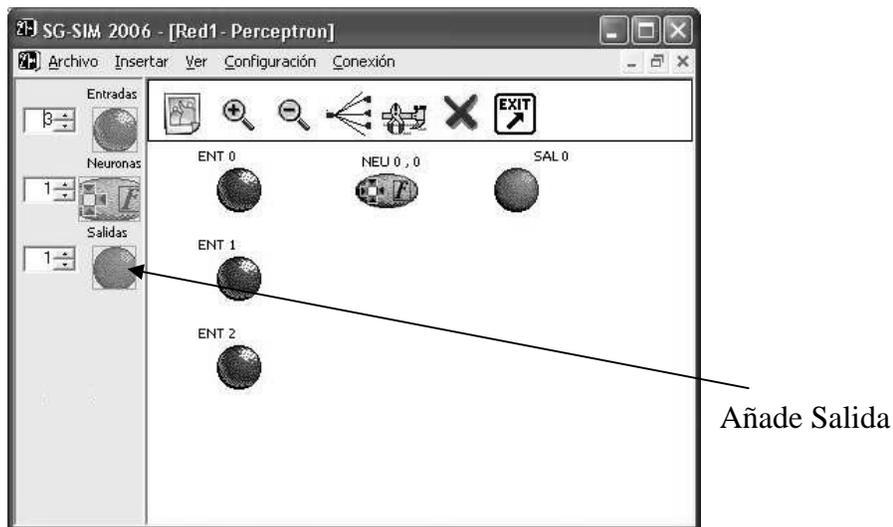


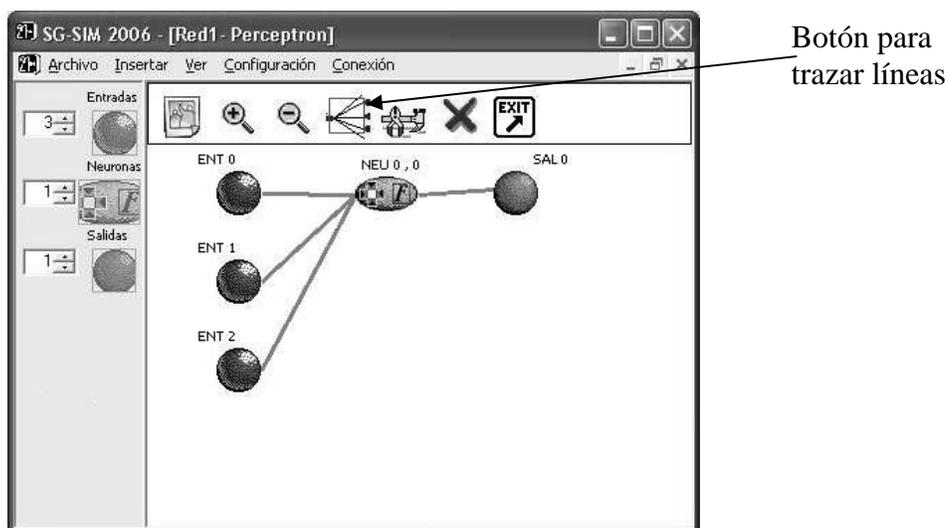
Figura 3-5 Ventana activa con Red Perceptrón. Añadir Neurona.

Luego para finalizar la creación de la red, se le asigna la salida a toda la red. Ver Figura 3-6.



**Figura 3-6** Ventana activa con Red Perceptrón. Añadir Salida.

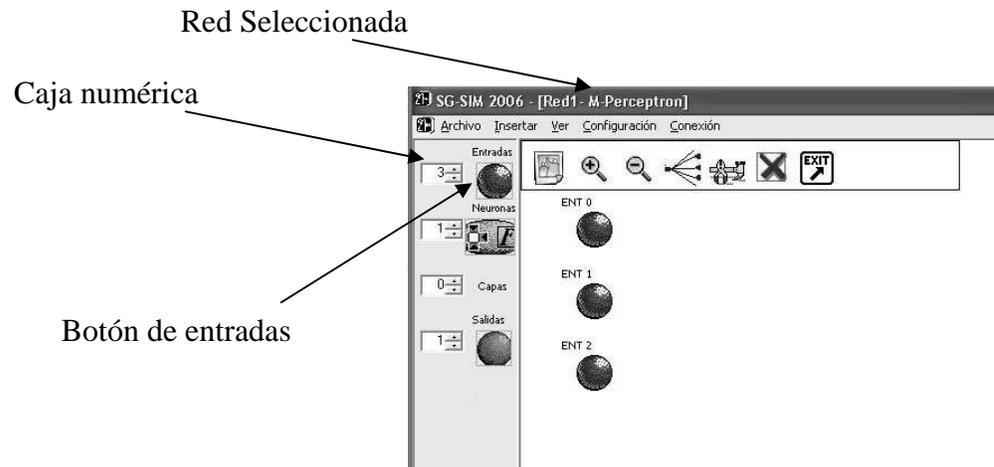
Luego, para completar el desarrollo de la red se trazan las líneas que unen a cada una de las neuronas. Ver Figura 3-7.



**Figura 3-7** Ventana activa con Red Perceptrón. Trazar Líneas.

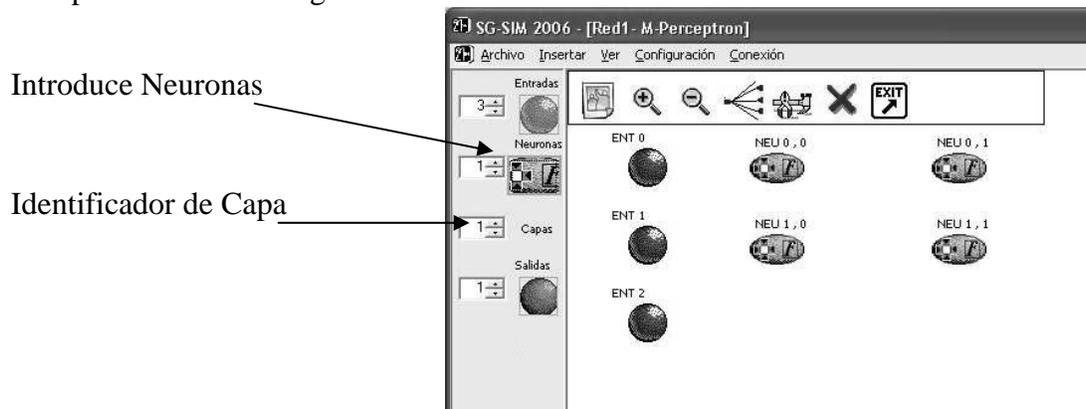
### 3.1.2. AMBIENTE GRAFICO DE LA RED M - PERCEPTRON

Después de haber seleccionada el tipo de red, M-Perceptrón, se procede a crear dicha estructura. De forma parecida a la red anterior, presionando el botón de entrada, se crea una entrada correspondiente a la cantidad puesta en la caja numérica, así como se muestra en la Figura 3-8.



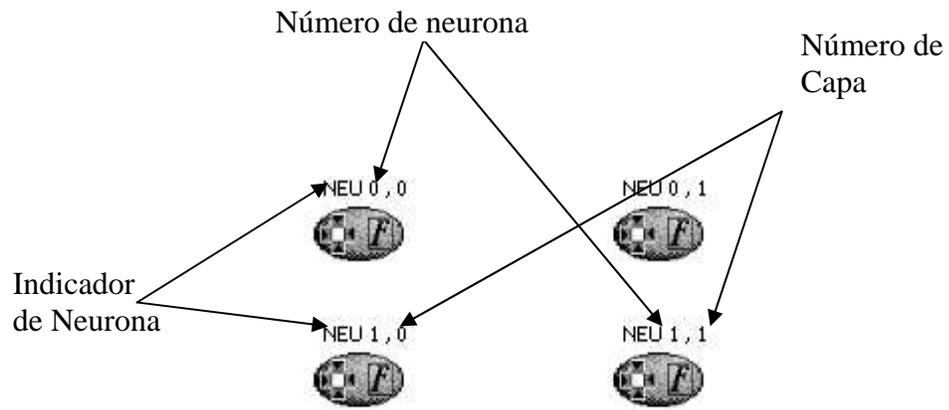
**Figura 3-8 Ventana Activa con Red M-Perceptrón. Añadir Entradas.**

Luego de haber creado las entradas, se procede a crear la neurona que hará el procesamiento de los datos. Por ser un M-Perceptrón, se permite la entrada de neuronas en formato de capas ocultas. Ver Figura 3-9.



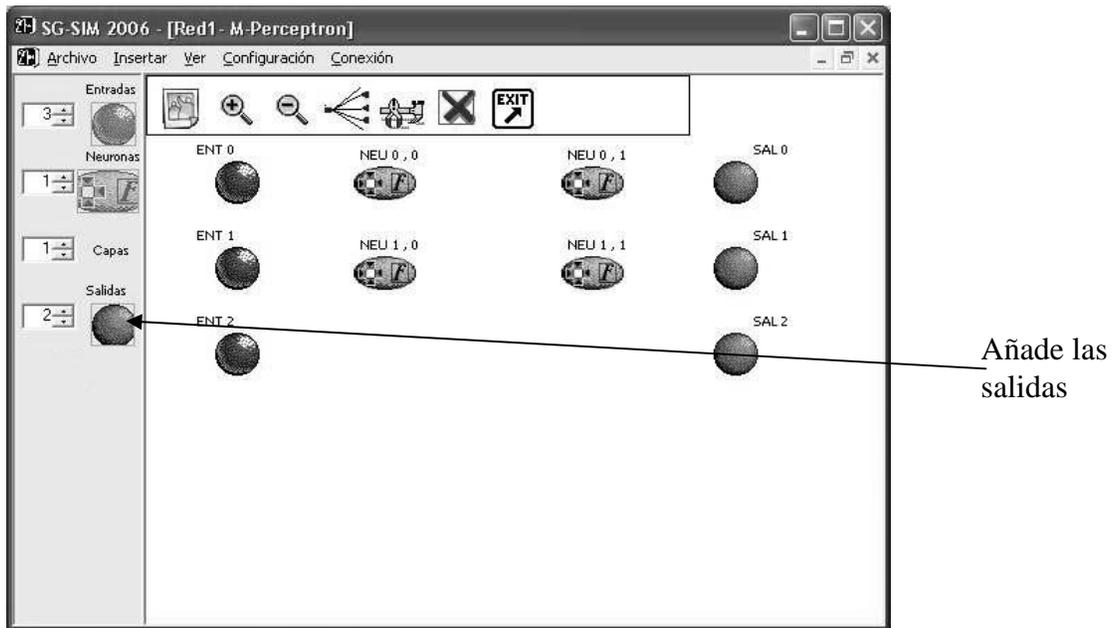
**Figura 3-9 Ventana Activa con Red M-Perceptrón. Añadir Neuronas por capas.**

Para identificar cada una de las neuronas correspondientes a cada capa, se utiliza una notación, la cual es la siguiente: *NEU X,Y*. En donde *X*, indica el conteo del número de neuronas que existe en la capa *Y*. Y *NEU* indica que es una neurona Ver Figura 3-10.



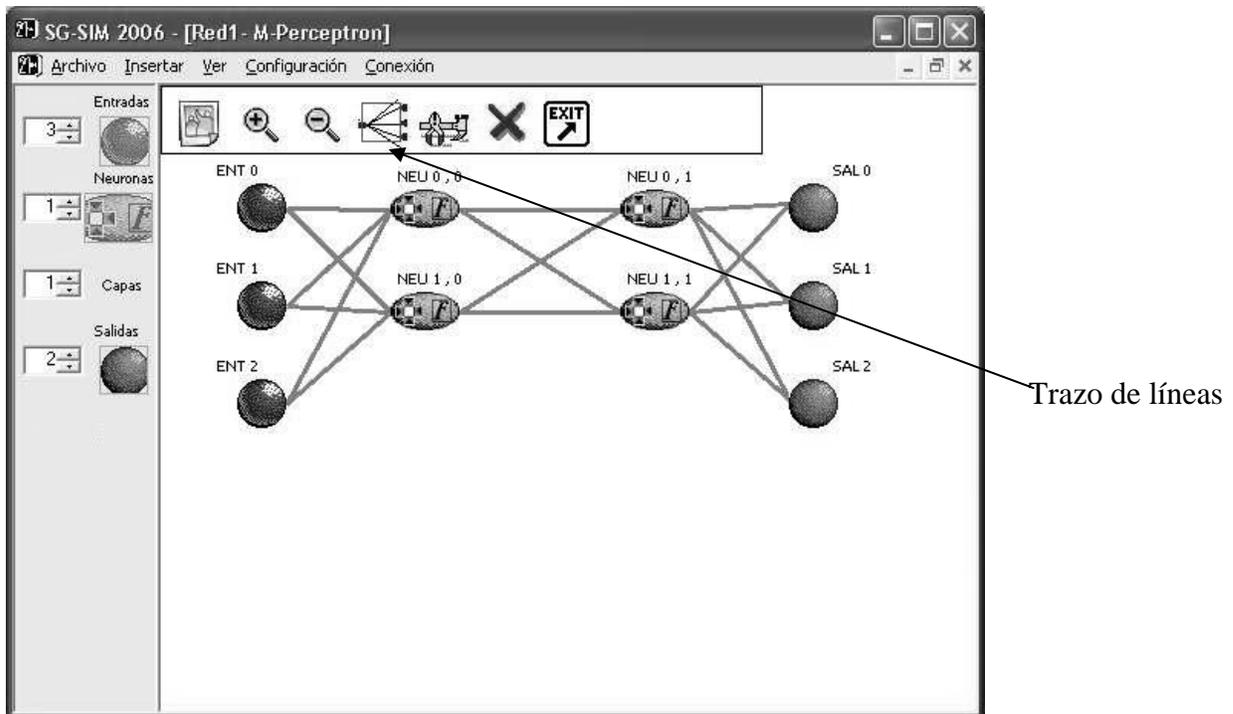
**Figura 3-10 Identificador de neuronas por capas.**

Luego, de haber establecido cuantas neuronas por capa se va a utilizar, se añade la o las salidas. Como se observa en la Figura 3-11.



**Figura 3-11 Ventana Activa con Red M-Perceptrón. Añadir Salidas.**

Luego, para completar el desarrollo de la red se trazan las líneas que unen a cada una de las neuronas. Ver Figura 3-12.



**Figura 3-12** Ventana Activa con Red M-Perceptrón. Trazo de Líneas.

### 3.1.3. AMBIENTE GRAFICO DE LA RED ADALINE

La arquitectura ADALINE es muy parecida a lo que es la arquitectura del Perceptrón simple, por lo tanto, se puede realizar una red de éste tipo siguiendo los mismos pasos que los del Perceptrón.

### **3.1.4. AMBIENTE GRAFICO DE LA RED BACKPROPAGATION**

La arquitectura de Retro propagación es muy parecida a lo que es la arquitectura del Perceptrón múltiple, por lo tanto, se puede realizar una red de éste tipo (Retro propagación) siguiendo los mismos pasos que los del M-Perceptrón.

# CONCLUSIONES

---

- ✓ El SG-SIM 2006 es un simulador de Redes Neuronales Artificiales capaz de entrenar diferentes topologías, además acepta imágenes para el entrenamiento de las redes con las cuales éste trabaja. Proporcionando así una aplicación adicional al simular y a su vez brindándole al alumno la oportunidad de incursionar un poco en lo que es el reconocimiento de imágenes. De esta manera, SG-SIM 2006 no se limita únicamente al puro procesamiento de datos introducidos por el usuario, sino que también da la opción de introducir imágenes que motivan al estudiante a indagar en otras ramas en las que esta inmerso las Redes Neuronales Artificiales.
  
- ✓ Según el muestreo que se realizó, se encontró que un poco más del 55 % de los encuestados mostró una buena adaptación después de pocos minutos de utilizar el SG-SIM 2006, demostrando de esta manera que el simulador tiene una mayor amigabilidad, con respecto al software utilizado actualmente para impartir los laboratorios de la cátedra de Redes Neuronales. Dando a conocer la facilidad con la que el alumno puede adaptarse al simulador SG-SIM 2006, sin la necesidad de tener que escribir líneas de comandos que distraigan la atención del estudiante. De esta manera se aporta una opción más con la que puede contar la comunidad estudiantil y sin desear desmerecer la técnica utilizada por el programa que actualmente se ha estado utilizando para el desarrollo de las prácticas de laboratorio.

- ✓ El SG-SIM 2006 es una buena alternativa para el desarrollo de simulaciones de RNAs, debido a que proporciona beneficios para la comprensión y desarrollo de la materia de Redes Neuronales de la Universidad Don Bosco; es de resaltar que el simulador (SG-SIM 2006) posee una interfaz interactiva en donde el usuario puede construir visualmente la Red que desea, facilitando la comprensión de conceptos haciendo uso de medios gráficos, mientras que Matlab carece de ellos. Por lo que el simulador SG-SIM 2006 se convierte en una muy buena opción, como herramienta, para el aprendizaje y autoaprendizaje del alumno.
  
- ✓ Para el desarrollo del SG SIM 2006 se requirió de profundizar más sobre las topologías de Redes Neuronales Artificiales, así como sus respectivas técnicas para el entrenamiento de ellas, fue así como se descubrió que el procedimiento para entrenar una Red Perceptron de Múltiples Capas, está basado en el cálculo del Error Cuadrático Medio, como lo es en la Red de Retro Propagación, con diferencias que estriban en: su Funciones de Activación y sus correspondientes Vectores de Entradas, es decir que para el entrenamiento de dicha Red se requiere de considerar entradas discretas y por lo consiguiente las funciones de activación que obtienen su respectiva salida deben de ser también discretas, para cumplir con el principio elemental de una Red Perceptron.

- ▼ Dentro de las pruebas realizadas al SG SIM 2006, se tomó en cuenta hacer una comparación de resultados obtenidos teniendo como referencia las épocas que duró el entrenamiento generado en Matlab, para ello se realizaron 10 pruebas por red tanto en Matlab como en el simulador. Y el resultado de éstas generó una desviación promedio de 27.33 para la Red Perceptron, un 64.67 para la Red M-Perceptron, un 24.78 para la Red ADALINE y finalmente un 53.45 para la Red Retro Propagación. Todas éstas a favor de Matlab, haciendo notar que los avanzados estudios y técnicas desarrolladas e implementadas en Matlab, demuestran mejores resultados.
  
- ▼ Para la implementación del algoritmo de aprendizaje de cada una de las redes desarrolladas en el SG SIM 2006, se tomo en cuenta un valor máximo de épocas a cumplir, renovándose cada vez que éste alcance su valor si el usuario así lo desea, para lograr así una buena estabilidad al simulador. Adicionalmente se consideró una opción de paro incondicionado, que le permite al usuario detener el proceso de entrenamiento de alguna de las redes implementadas basándose en un tiempo de respuesta estimado por el usuario.
  
- ▼ Durante el desarrollo del simulador, se fueron realizando pruebas de depuración de procedimientos, en su mayor parte para la generación de los objetos gráficos a presentar como lo son los arreglos de: Entradas, Neuronas por Capa, Salidas, Vectores de Entrada

de forma Gráfica o por celdas convencionales, generación y obtención de Pesos Iniciales y Finales respectivamente, así también los formularios y demás contenedores de objetos gráficos, entre otros. Para lo cual se tuvo que descartar el procedimiento de generación casi en su totalidad de los contenedores de objetos gráficos tales como: cuadros de diálogo, paneles de controles, etc. Y esto provocó tomar la decisión de mejorar la versión de la interfaz gráfica de usuario que se estaba utilizando por una que nos permitiera optimizar el tiempo de respuesta para la generación gráfica en tiempo de desarrollo de los objetos.

# RECOMENDACIONES

---

- ▼ Al programador. Debido a que el SG-SIM 2006 es la primera parte de un simulador de Redes Neuronales Artificiales, es necesario completar dicho software, añadiendo las diferentes topologías restantes que son impartidas en la materia de Redes Neuronales, de la Universidad Don Bosco. Además, se recomienda añadir alguna gráfica que indique el comportamiento del error alcanzado con respecto al número de épocas, esto ayudaría al usuario a detener el entrenamiento de la red, si los pesos que se van calculando no son los adecuados para el buen entrenamiento de la misma (red).
  
- ▼ Al usuario. Antes de iniciar a utilizar al SG-SIM 2006, es recomendable conocer del tema de Redes Neuronales Artificiales, pues esto hace que el tiempo de adaptación del usuario con el software se reduzca en gran medida. También, es de tener en cuenta que el entrenamiento con imágenes es limitado, y a pesar que puede soportar imágenes de un máximo de 800 x 800 píxeles, no se recomienda introducir imágenes arriba de los 25 x 25 píxeles pues el SG-SIM 2006 se retrasa en gran medida, tanto en cargar el ambiente gráfico apropiado, así como en el cálculo de los pesos en el entrenamiento, dicho cálculo puede extenderse hasta horas por época. Cabe resaltar de nuevo, que el simulador de redes es capaz de soportar imágenes de hasta 800 x 800 píxeles, es decir, un máximo de 640,000 píxeles.

- ▼ Al usuario. Al momento de iniciar el simulador SG-SIM 2006 y escoger la topología necesitada, es importante que tome en cuenta la estructura de esa red, debido a que el simulador trabaja con topologías que no requieren de capas ocultas y de otras que sí, por lo que al estar construyendo cierta topología donde no corresponde, podría causarle pérdida de tiempo. Además, puede existir alguna red que el tiempo de entrenamiento puede llevarle muchas horas, para lo cual se recomienda detener el entrenamiento y guardar dicha red para continuar posteriormente. El archivo generado por el simulador es de extensión \*.SG, y también se recomienda únicamente abrirlo con el simulador SG-SIM 2006, esto reduciría al mínimo la posibilidad de una pérdida en alguno de los valores guardados de la red.

# BIBLIOGRAFIA

---

[1] Falsett Laurene, "Fundamentals of Neural Networks; Architectures, Algorithms and Applications". Prentice Hall, Jersey 1994.

[2] Hilera José y Martínez Víctor, "Redes Neuronales Artificiales; Fundamentos, modelos y aplicaciones". Addison – Wesley Iberoamericana, S.A. De la edición RA-MA 1995.

[3] Freeman, James y Skapura David, "Neural Networks; Algorithms, Applications, and Programming Techniques". Addison – Wesley Publishing Company, 1991.

[4] Veelenturf, L.P.J., "Analysis And Applications Of Artificial Neural Networks". Prentice Hall International, 1995.

[5] Blank, Leland y Taquín, Anthony, "Ingeniería Económica", traducción por Arango Medina, Gladis; Editorial Mc Graw Hill, edición 1999.

[6] Universidad de Antioquia, Facultad de Ingeniería, unidad de investigación. Mecatrónica. <http://ingenieria.udea.edu.co/investigacion/mecatronica/mectronics/redes.htm>  
[Consulta: Abril de 2006].

[7] Universidad de los Andes, Facultad de Ingeniería. Publicaciones del Ing. Lourdes Arroyo.

<http://tecnologiaysociedad.uniandes.edu.co/html/memoria/2003/matrix/icaro/noticia-4.html>

[Consulta: Mayo de 2006]

[8] Universidad Tecnológica de Pereira, Facultad de Ingeniería Electrónica. *Tutorial de Redes Neuronales*. <http://ohm.utp.edu.co/neuronales/> [Consulta: Agosto de 2006]

[9] Herramientas en GNU/Linux para estudiantes universitarios, [http://es.tldp.org/Presentaciones/200304curso-galisa/redes\\_neuronales/curso-galisa-redes\\_neuronales-html/x17.html](http://es.tldp.org/Presentaciones/200304curso-galisa/redes_neuronales/curso-galisa-redes_neuronales-html/x17.html) [Consulta: Agosto de 2006]

[10] Artículo publicado por PCWORLD. *Inteligencia Artificial y Redes Neuronales Permiten imitar el cerebro humano*.

<http://www.idg.es/pcworld/articulo.asp?id=34270>. [Consulta: Febrero de 2006]

[11] Tecnológico de Monterrey, Material de Apoyo de “Lógica computacional” [http://w3.mor.itesm.mx/~logica/log9808/log\\_pred.html](http://w3.mor.itesm.mx/~logica/log9808/log_pred.html)

[12] REYES, Gary "Proyecto de inteligencia artificial" en Universidad de Guayaquil. <http://cuzrojaguayas.org/inteligencia/>

[13] PSCOACTIVA, “Las Torres de Hanoi”, Juego interactivo de inteligencia. [http://www.psicoactiva.com/juegos/hanoi/jg\\_hanoi.htm](http://www.psicoactiva.com/juegos/hanoi/jg_hanoi.htm)

## APENDICE A

---

### Encuesta

Indicaciones: Marque con una "x" la casilla o complete con letra legible su respectiva respuesta.

1. ¿Qué es lo primero que se le viene a la mente cuando le hablan de Redes Neuronales Artificiales?

\_\_\_\_\_

2. ¿Puede mencionar al menos una aplicación?

\_\_\_\_\_

3. ¿Le gustaría conocer y aprender (más) de ellas por medio de un SIMULADOR?

SI

NO

4. Si tuviera la opción de elegir entre un SIMULADOR basado en Líneas de Comando (p.ej. MATLAB<sup>®</sup>) o basado en AMBIENTE GRAFICO, ¿Cuál eligiera?

SIMULADOR GRAFICO

LINEAS DE COMANDO

5. ¿Qué le gustaría que contenga un SIMULADOR de Redes Neuronales Artificiales?: (puede marcar las opciones que desee)

Diferentes tipos de Redes

Entrenamiento y Comprobación

Arquitectura de las Redes

Obtención de Pesos Finales

Demostraciones y Aplicaciones

Ambiente Gráfico y Amigoso

Ayuda

Otros

Otros:

\_\_\_\_\_

6. ¿Qué beneficios espera obtener después de correr un SIMULADOR de Redes Neuronales Artificiales?

\_\_\_\_\_

7. ¿Cuál es su especialidad?:

Ing. en Computación

Ing. Electrónica

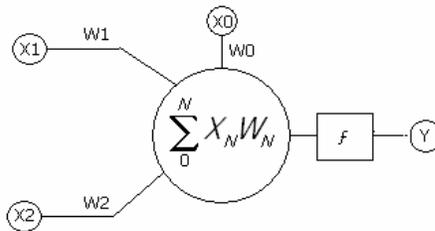
Ing. Eléctrica

Ing. en Telecomunicaciones

Ing. en Automatización

MUCHAS GRACIAS POR SU TIEMPO

8. A continuación se presenta una topología de Red Neuronal Artificial, trate de hacerla con los programas siguientes:
- Matlab 6.5
  - SG-Sim v1.1 beta



Comentarios:

Con respecto al software Matlab 6.5: \_\_\_\_\_

---



---



---

Con respecto al software SG-Sim V1.1 beta: \_\_\_\_\_

---



---



---

9. Podría hacer alguna sugerencia con respecto al software SG-Sim v1.1 beta: \_\_\_\_\_

---



---



---

MUCHAS GRACIAS POR SU TIEMPO

## APENDICE B

---

### **B.1 TABULACION DE ENCUESTAS**

Para comprobar la aceptación del simulador, así como su funcionalidad, fue necesario entrevistar a cierto grupo de personas para que hicieran pruebas en el software. De ésta manera es como el software ha ido evolucionando hasta llegar a una primera interfaz gráfica (Ver Figura 3-1) en donde se encuentra a la vista lo necesario para realizar la topología que el usuario desee.

Después de haber llegado a la primera interfaz de usuario, se comenzó a tabular una encuesta, cuyo objetivo era recolectar la opinión de diferentes tipos de usuarios, con conocimiento o no, del tema de Redes Neuronales Artificiales para observar la adaptación del software en proceso. Además de hacer una comparación con el software Matlab 6.0, debido a que éste es el utilizado actualmente, para el desarrollo de la cátedra de la materia de Redes Neuronales Artificiales de la Universidad Don Bosco.

### **B.2 ENCUESTAS**

En el Apéndice A se encuentra la encuesta pasada a los usuarios que utilizaron software en desarrollo.

## B.2.1 TAMAÑO DE LA MUESTRA

Para obtener un número estadístico del cual se tuviera confiabilidad, se optó por utilizar la siguiente fórmula [5]:

$$n = \frac{s^2 * N * P * Q}{E^2 * (N - 1) + s^2 * P * Q}$$

**Ecuación B-1**

Donde:

n: Es el número de personas a encuestar.

$s^2$ : Es el nivel de confiabilidad.

N: Es el tamaño de la población.

P: Es la probabilidad a favor.

Q: Es la probabilidad en contra.

$E^2$ : Es el error de la encuesta.

Los valores utilizados, se muestran en la siguiente tabla:

$S^2$	N	E	$E^2$	P	Q	n
0,8	873	0,06	0,0036	0,5	0,5	52,2879732

**Tabla B-1 Datos Estadísticos**

El tamaño de la población, se consideró de esa manera, debido a que son el número total de estudiantes inscritos en el Ciclo II 2006, en la Universidad Don Bosco, de las carreras que llevan en su pénsum, la materia de Redes Neuronales Artificiales.

Obteniendo de esta manera un total de 53 personas a encuestar. De este total, se obtuvo una muestra de 50 personas encuestadas, llegando así muy cerca de los valores puestos en la Tabla B-1.

### **B.3 EXPOSICION DE RESULTADOS**

Los resultados arrojados por las encuestas fueron los siguientes:

La gran parte de las muestras tomadas, tienen una idea muy arraigada que Redes Neuronales Artificiales tiene una gran relación con lo que es Inteligencia Artificial. Dicho tema, tiene gran atención por muchas personas, debido a que es un campo en el cual, éste país (El Salvador), esta incursionando.

A pesar, que muchos tenían una noción que relacionaba Redes Neuronales e Inteligencia Artificial, muy pocos sabían el uso que se le podría dar, sin embargo todos mostraron un entusiasmo por saber un poco más del tema de Redes Neuronales Artificial.

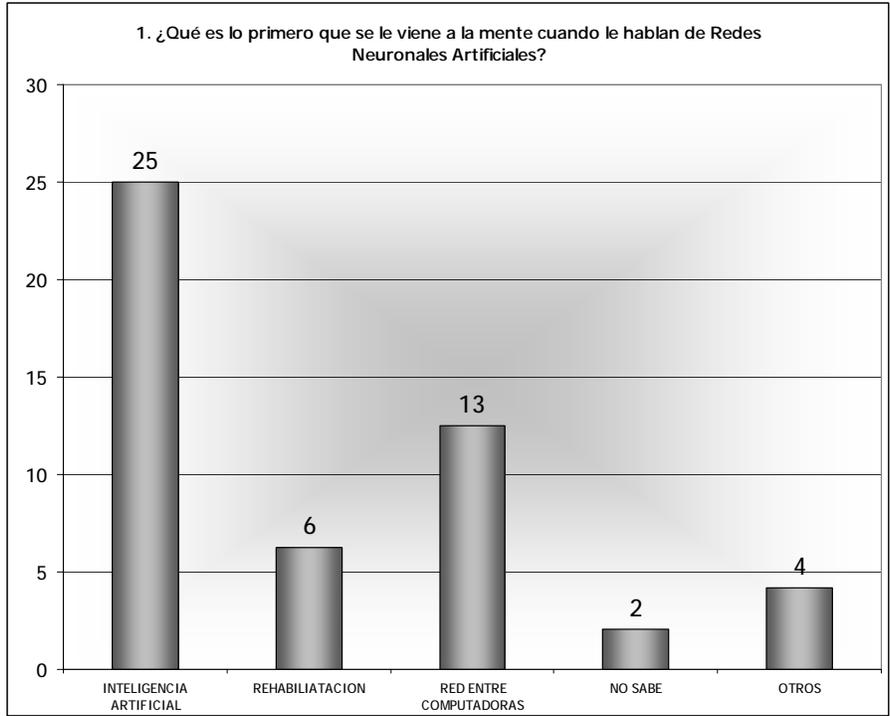
Es de añadir, que los encuestados, tienen una gran inclinación sobre los simuladores gráficos, con iconos interactivos los cuales permiten una mejor adaptación y comprensión de lo mostrado en la pantalla.

Además, los encuestados consideran que el programa debe de ser intuitivo, es decir, que no sea complicado de utilizar, debido a que muy pocos utilizan la ayuda que se encuentra instalada en un software.

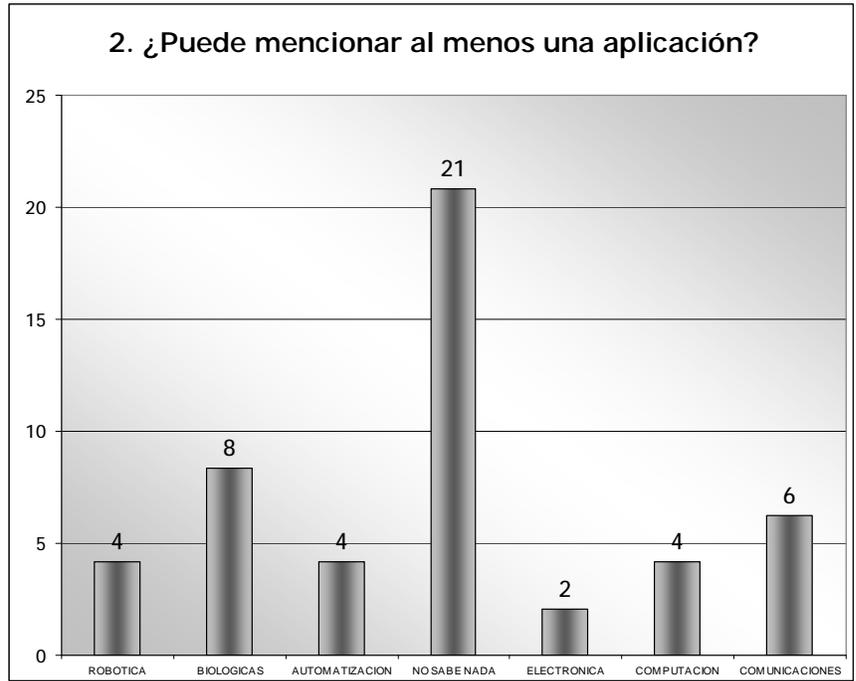
Después que los encuestados, plantearon muchas sugerencias para el software, se les presentó software que se ha desarrollado para la simulación de Redes Neuronales Artificiales, así como el software que se utiliza hasta el momento para impartir la cátedra. Obteniéndose de esta manera criticas muy satisfactorias.

El software resultante obtuvo buenas opiniones, debido a que se apegaba bastante a los que los encuestados pretendían obtener, y por lo tanto llegaban a entender en una solo sesión conceptos que no se lograba anteriormente con Matlab.

La siguiente información puede ser corroborada con los gráficos obtenidos de las encuestas llenadas por una muestra de los alumnos de la Universidad Don Bosco, las cuales se presentan a continuación.



**Figura B-1 Pregunta 1.**



**Figura B-2 Pregunta 2.**

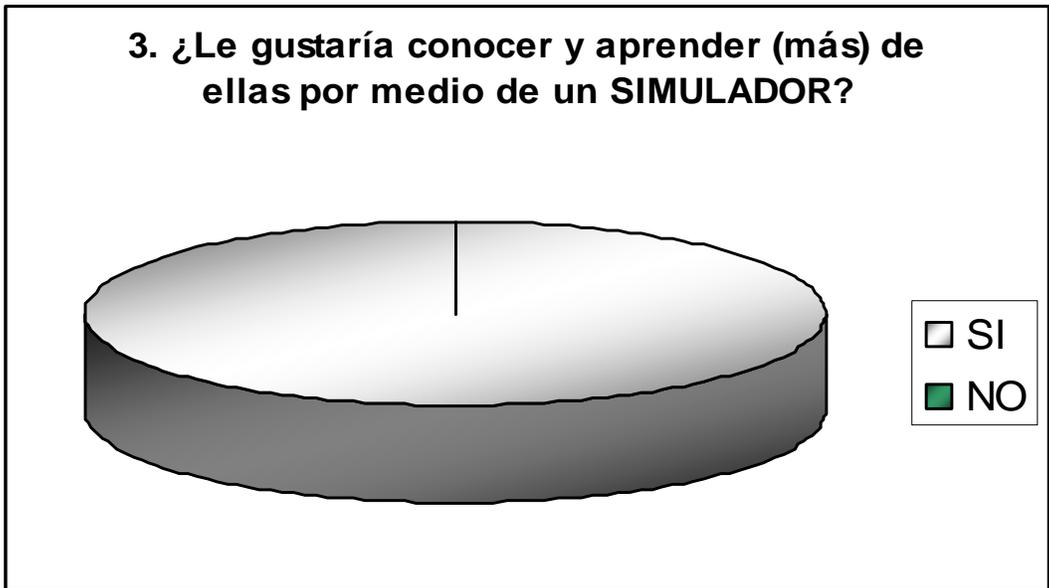


Figura B-3 Pregunta 3.

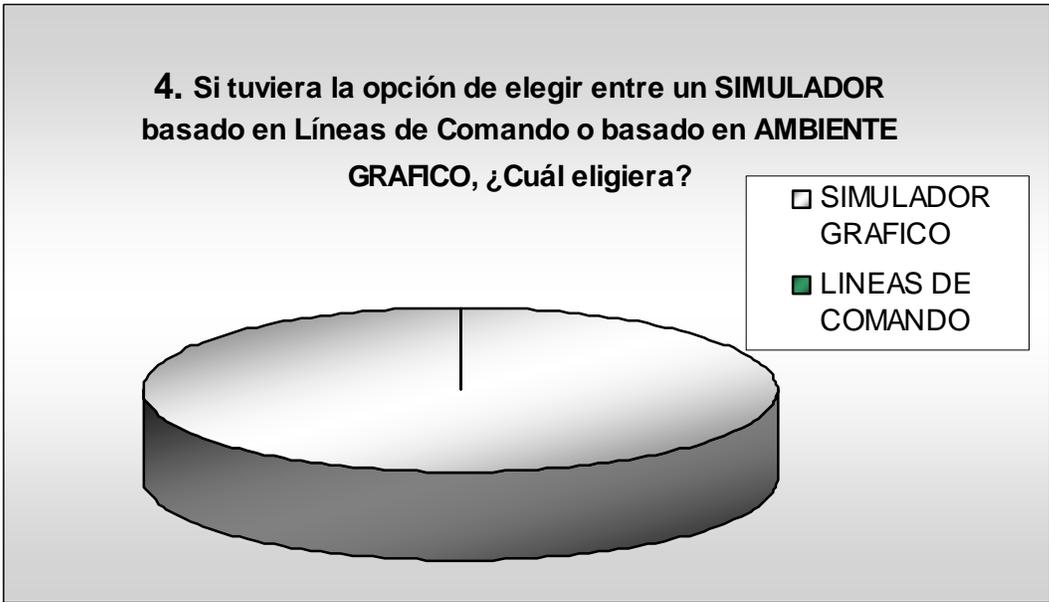
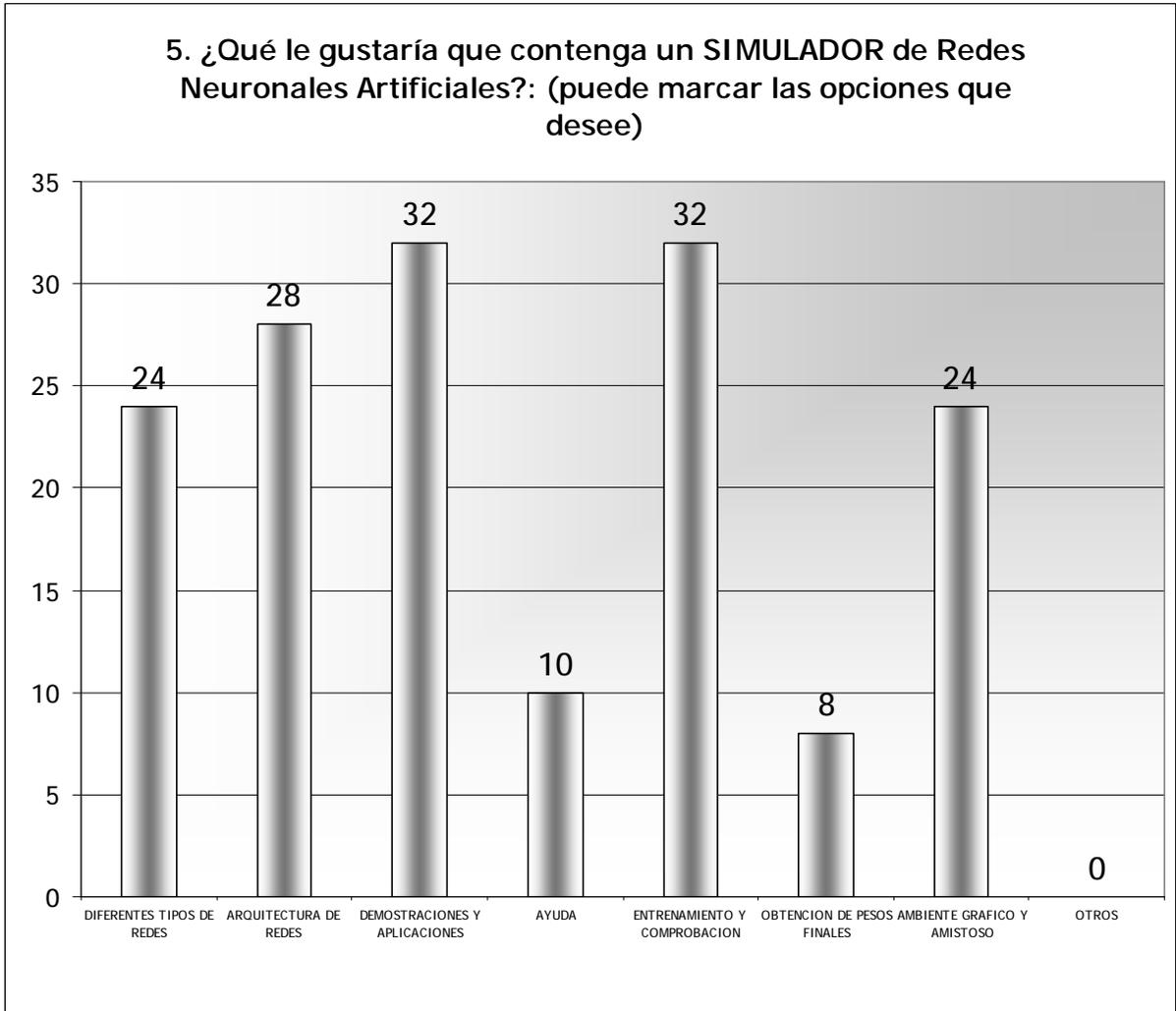
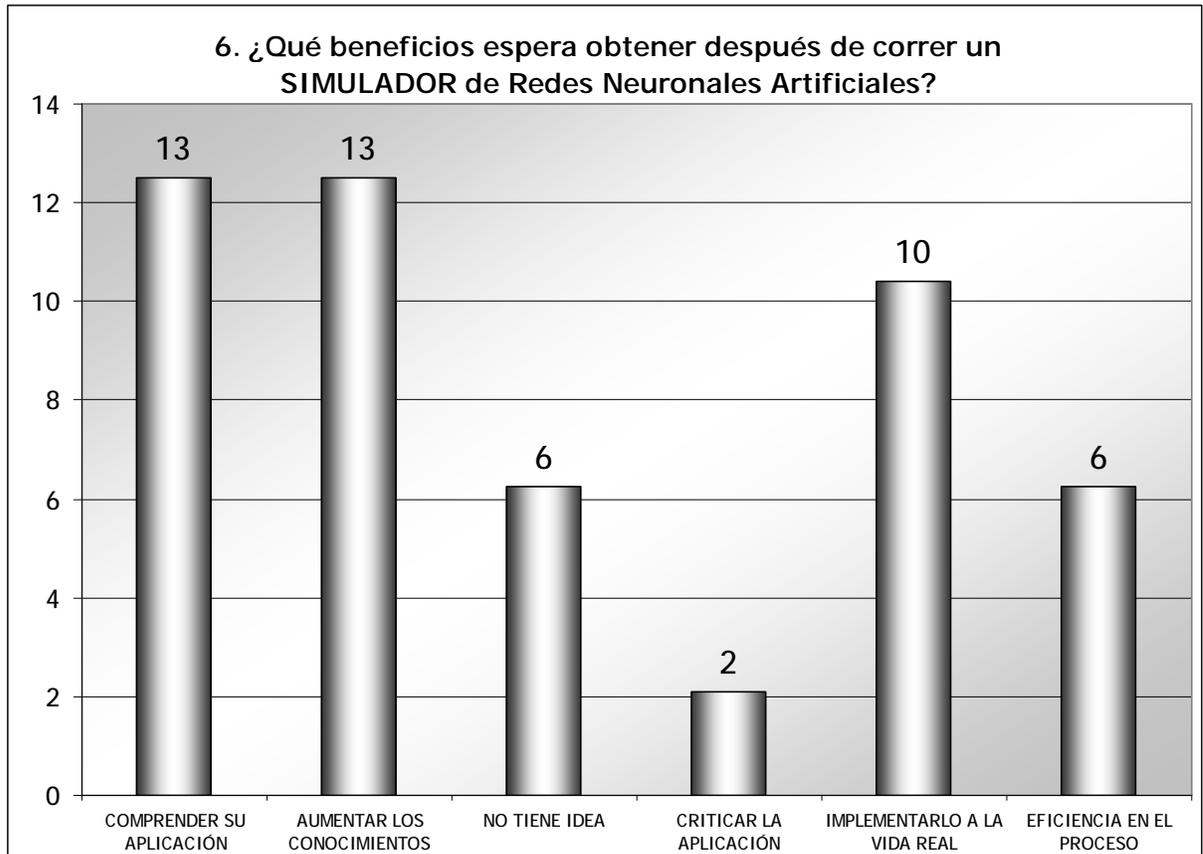


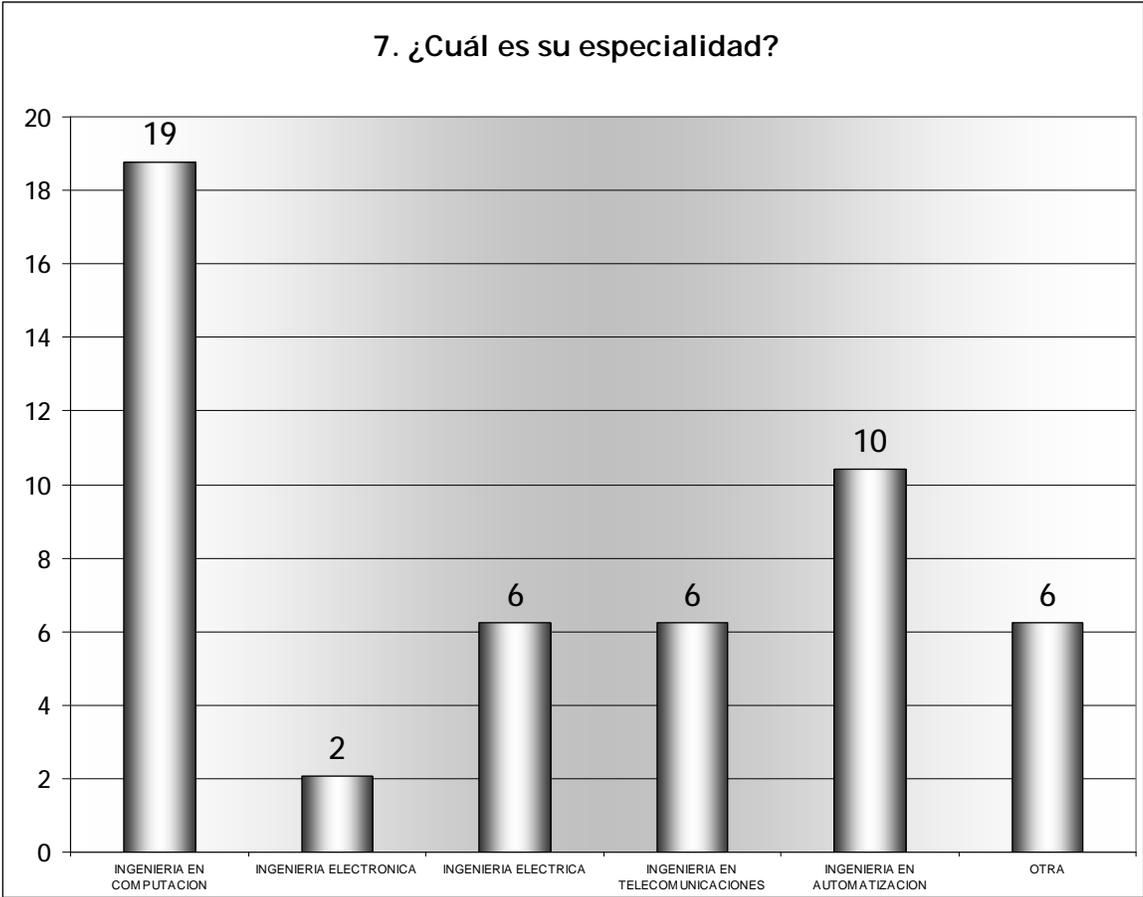
Figura B-4 Pregunta 4.



**Figura B-5 Pregunta 5.**



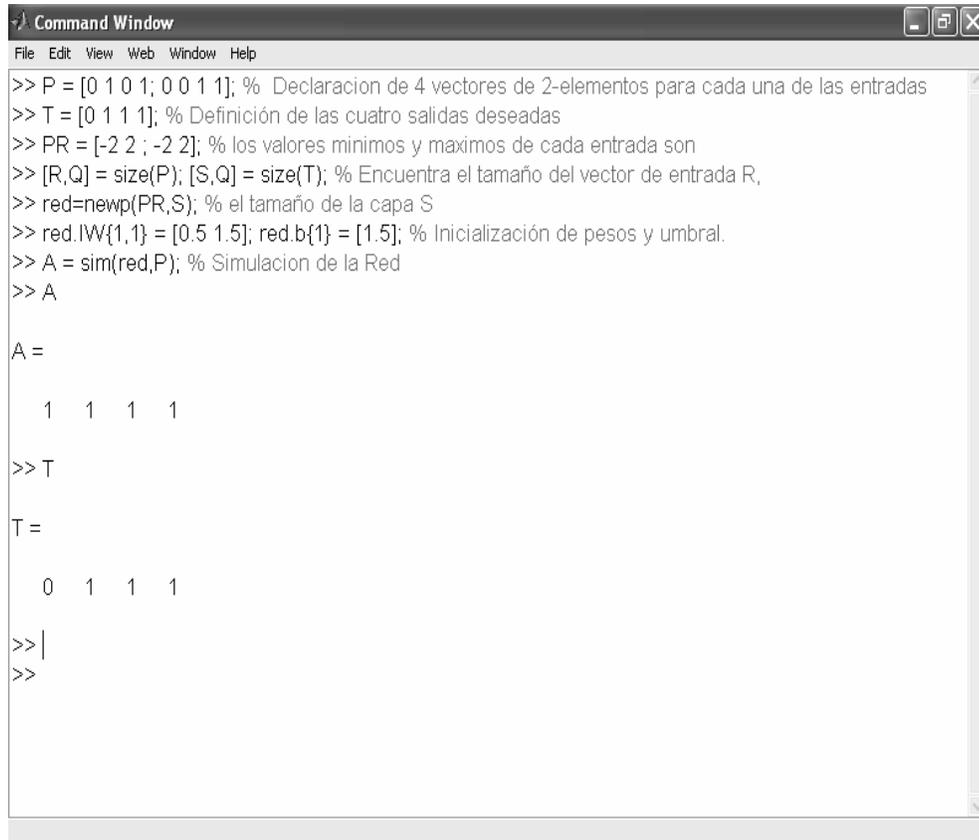
**Figura B-6 Pregunta 6.**



**Figura B-7 Pregunta 7.**

## APENDICE C

---



```
Command Window
File Edit View Web Window Help
>> P = [0 1 0 1; 0 0 1 1]; % Declaracion de 4 vectores de 2-elementos para cada una de las entradas
>> T = [0 1 1 1]; % Definición de las cuatro salidas deseadas
>> PR = [-2 2 ; -2 2]; % los valores minimos y maximos de cada entrada son
>> [R,Q] = size(P); [S,Q] = size(T); % Encuentra el tamaño del vector de entrada R,
>> red=newp(PR,S); % el tamaño de la capa S
>> red.IW{1,1} = [0.5 1.5]; red.b{1} = [1.5]; % Inicialización de pesos y umbral.
>> A = sim(red,P); % Simulacion de la Red
>> A

A =

    1    1    1    1

>> T

T =

    0    1    1    1

>> |
>>
```

**Figura C-1 Hoja para simulación en Matlab, para los encuestados.**

## APENDICE D

---



Indicaciones: Para la siguiente serie de preguntas, conteste honestamente. Dicha información será útil para el trabajo de graduación, con el cual estará interactuando:

1. ¿Cuál es el conocimiento que tiene acerca de las Redes Neuronales Artificiales?

Avanzado                      Medio                      Bajo                      Nulo

2. ¿Se comprende la topología de cada red entrenada?

Bastante                      Medio                      Poco                      Nada

3. ¿Se interpreta de una manera fácil la asignación de entradas y salidas?

Bastante                      Medio                      Poco                      Nada

4. ¿Se comprende la asignación de pesos a la Red?

Bastante                      Medio                      Poco                      Nada

5. ¿Están claros los pasos a seguir, para realizar una red de cualquier tipo?

Bastante                      Medio                      Poco                      Nada

6. Considera útil la opción paso a paso, que poseen algunas redes

Bastante                      Medio                      Poco

7. Considera necesario las opciones de Abrir y Guardar, una red.

Bastante                      Medio                      Poco

8. Como clasifica la aceptación de imágenes dentro del programa.

Muy Buena                      Buena                      Mala

9. ¿Cómo clasifica el ambiente grafico del simulador SG-SIM?

Muy Bueno

Bueno

Regular

Malo

10. Comentario y/o sugerencias adicionales: \_\_\_\_\_

\_\_\_\_\_

---

Espacio reservado al instructor:

Tiempo de adaptación: \_\_\_\_\_ min.

## APENDICE E

---

### E.1 TABULACION DE ENCUESTAS

Para comprobar la aceptación del simulador, así como su funcionalidad, fue necesario entrevistar por segunda vez a cierto grupo de personas para que hicieran pruebas en el software.

Después de haber interactuado con la interfaz de usuario, se comenzó a tabular una encuesta, cuyo objetivo era recolectar la opinión de diferentes tipos de usuarios, con conocimiento o no, del tema de Redes Neuronales Artificiales para observar la adaptación del software en proceso.

### E.2 ENCUESTAS

En el Apéndice D se encuentra la segunda encuesta pasada a los usuarios que utilizaron software en desarrollo.

#### E.2.1 TAMAÑO DE LA MUESTRA

Para obtener un número estadístico del cual se tuviera confiabilidad, se optó al igual que en la encuesta anterior por utilizar la siguiente fórmula [5]:

$$n = \frac{s^2 * N * P * Q}{E^2 * (N - 1) + s^2 * P * Q}$$

Ecuación E-2

Donde:

$n$ : Es el número de personas a encuestar.

$S^2$ : Es el nivel de confiabilidad.

$N$ : Es el tamaño de la población.

$P$ : Es la probabilidad a favor.

$Q$ : Es la probabilidad en contra.

$E^2$ : Es el error de la encuesta.

Los valores utilizados, se muestran en la siguiente tabla:

$S^2$	$N$	$E$	$E^2$	$P$	$Q$	$n$
0,8	873	0,06	0,0036	0,5	0,5	52,2879732

**Tabla E-2 Datos Estadísticos de Segunda Encuesta**

El tamaño de la población, se consideró de esa manera, debido a que son el número total de estudiantes inscritos en el Ciclo II 2006, en la Universidad Don Bosco, de las carreras que llevan en su pénsum, la materia de Redes Neuronales.

Obteniendo de esta manera un total de 53 personas a encuestar. De este total, se obtuvo una muestra de 52 personas encuestadas, llegando así muy cerca de los valores puestos en la Tabla E-1.

### **E.3 EXPOSICION DE RESULTADOS**

Los resultados arrojados por las encuestas fueron los siguientes:

Existe un buen porcentaje de estudiantes que tienen conocimientos medios de Redes Neuronales Artificiales y que éstos a su vez comprendan y entiendan las topologías y formas de entrenar de las mismas.

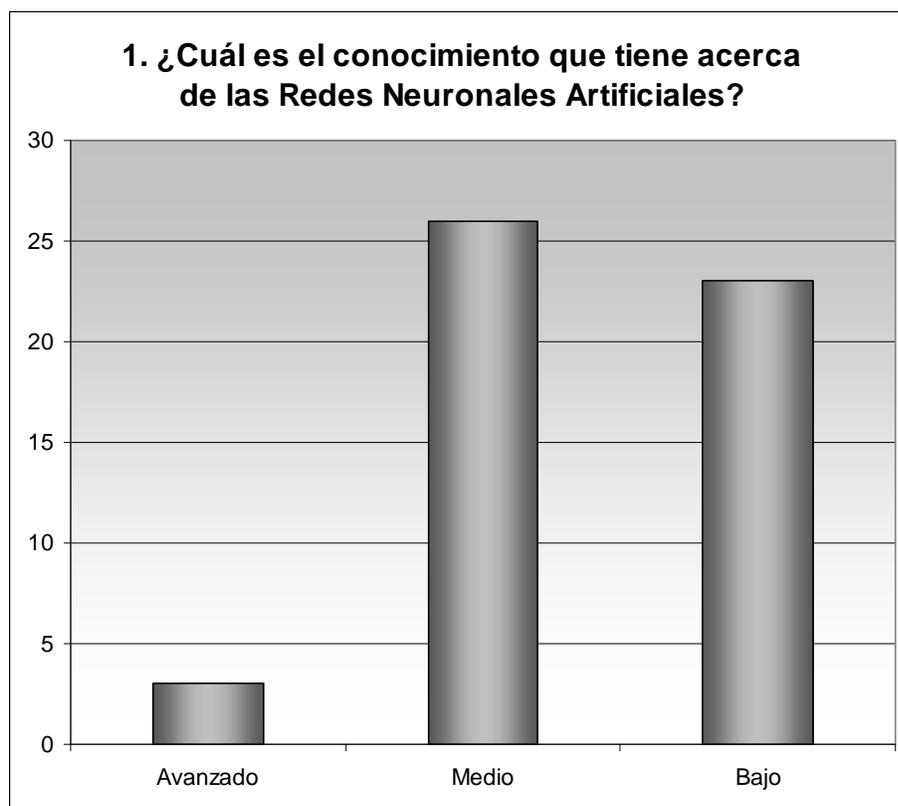
Considerando también la amigabilidad del simulador, se puede observar que los encuestados tuvieron una adaptabilidad a las funciones elementales del mismo, así como un fácil reconocimiento de los objetos gráficos con los que cuenta. Identificando muy bien las entradas, las salidas y sus elementos de procesamiento.

Sobre las opciones adicionales con las que cuenta el simulador podemos decir basándonos en la encuesta, se mantienen un poco niveladas las opiniones acerca de guardar y abrir archivos que contengan la información más relevante de una red en el SG SIM para continuar o detener un entrenamiento muy prolongado.

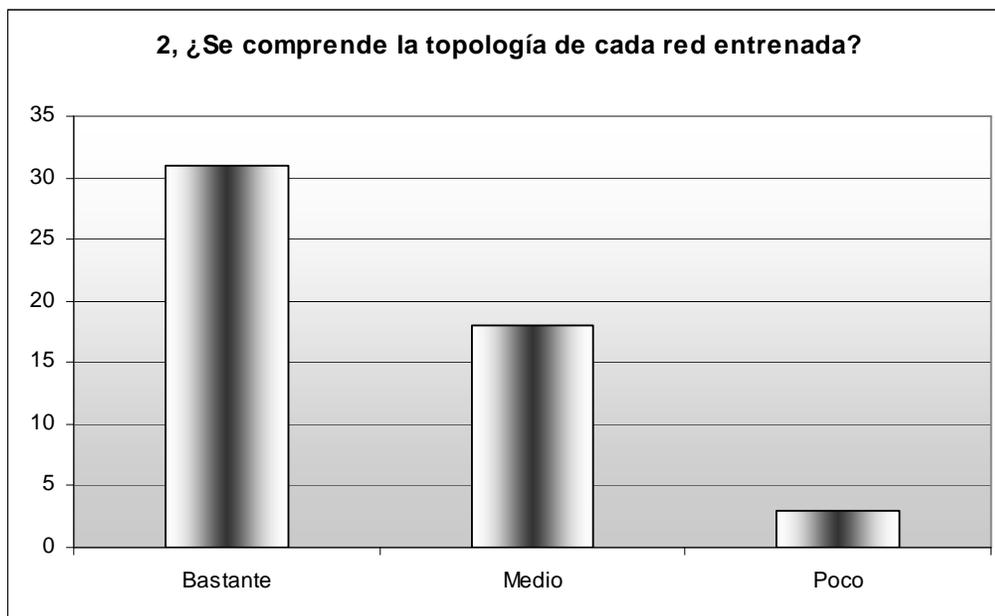
Además cuando se les preguntó a los encuestados sobre cómo clasificarían el ambiente gráfico del simulador más de la mitad coincidió en que les parecía muy bueno.

Finalmente se les tomó el tiempo de uso de la aplicación, para que sirviera como una referencia y así estimar un tiempo de adaptación que requirieron dichos encuestados, dando como resultado más de la mitad realizaron la prueba en menos de cinco minutos, lo cual nos permite decir que hubo una adaptación menor o igual a este tiempo.

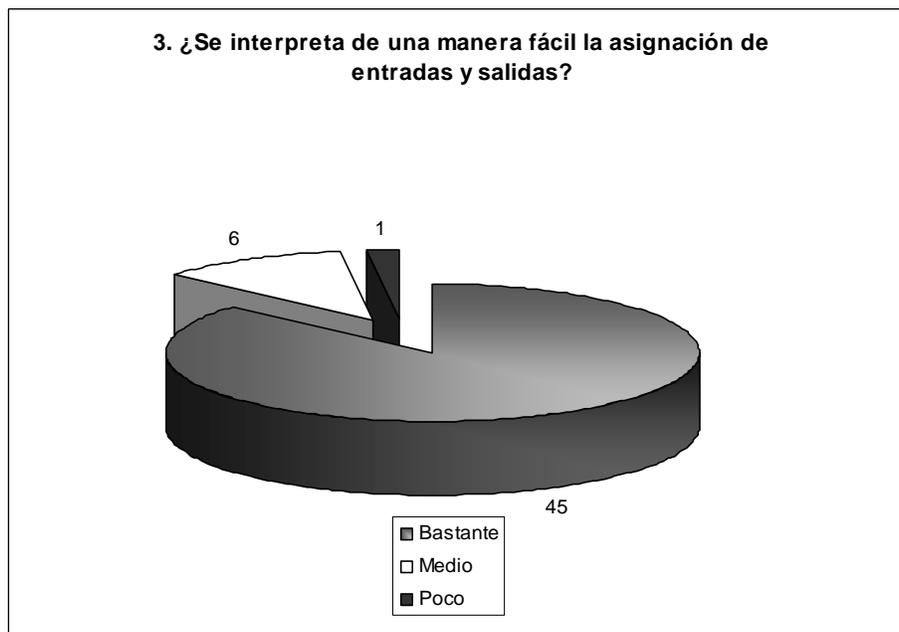
La siguiente información puede ser corroborada con los gráficos obtenidos de las encuestas llenadas por una muestra de los alumnos de la Universidad Don Bosco, las cuales se presentan a continuación.



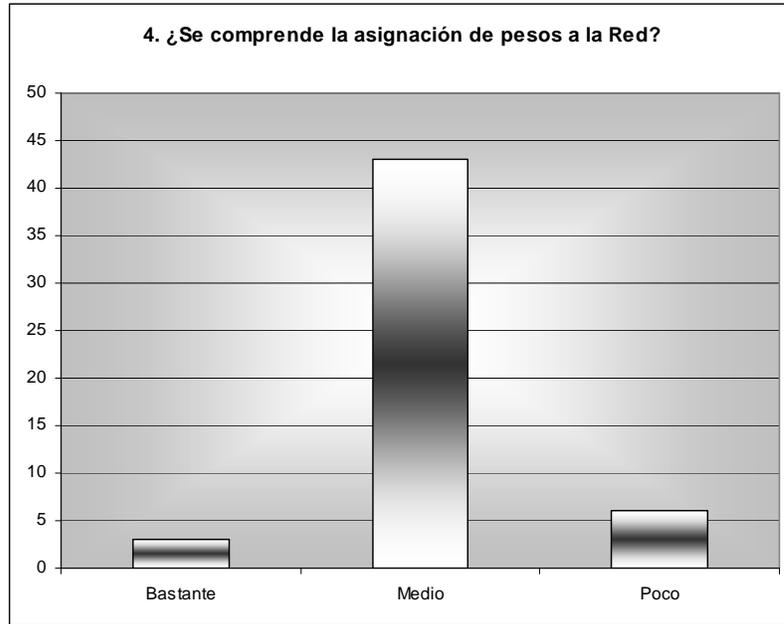
**Figura E-1 Pregunta 1 (Última Encuesta).**



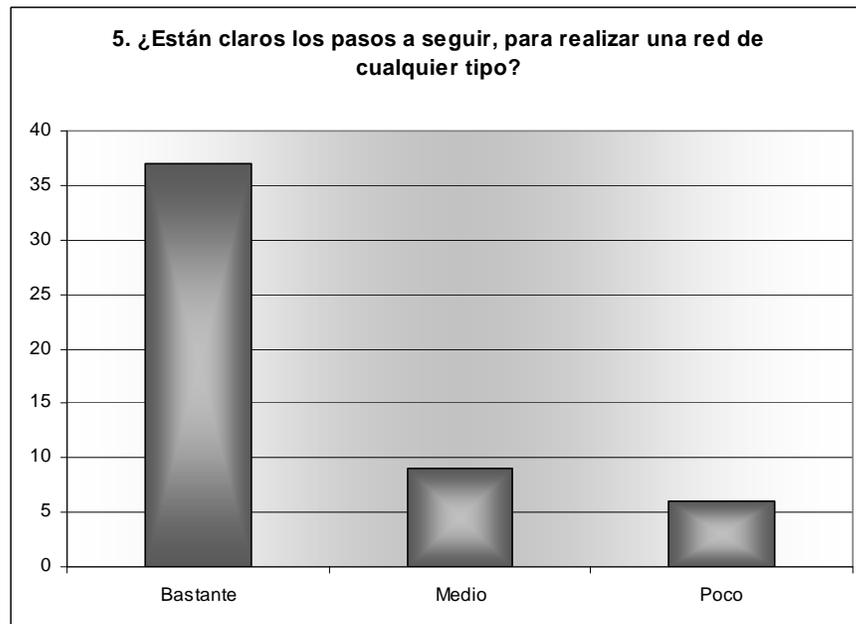
**Figura E-2 Pregunta 2 (Ultima Encuesta).**



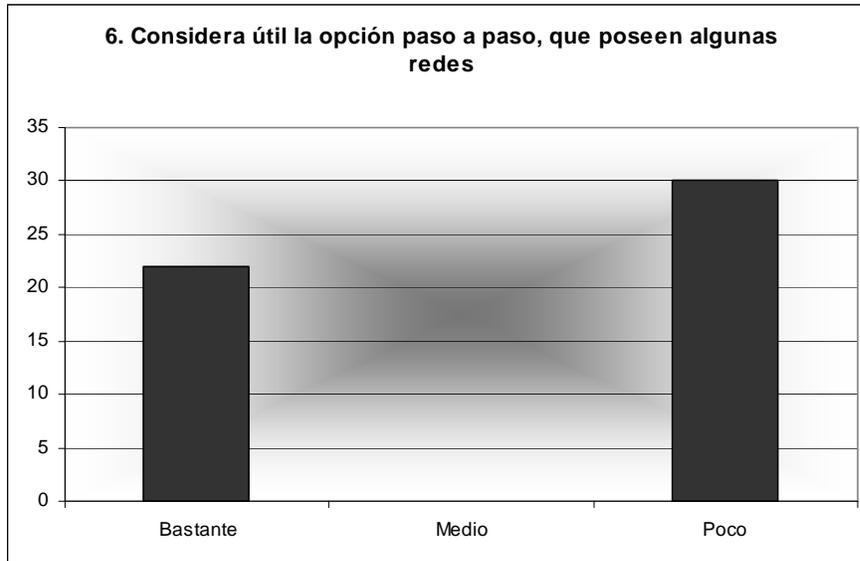
**Figura E-3 Pregunta 3 (Ultima Encuesta).**



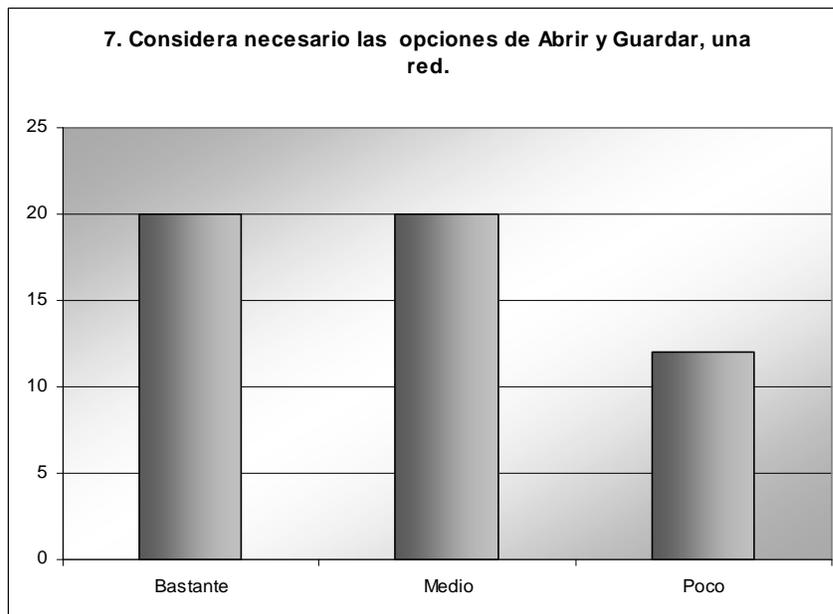
**Figura E-4 Pregunta 4 (Ultima Encuesta).**



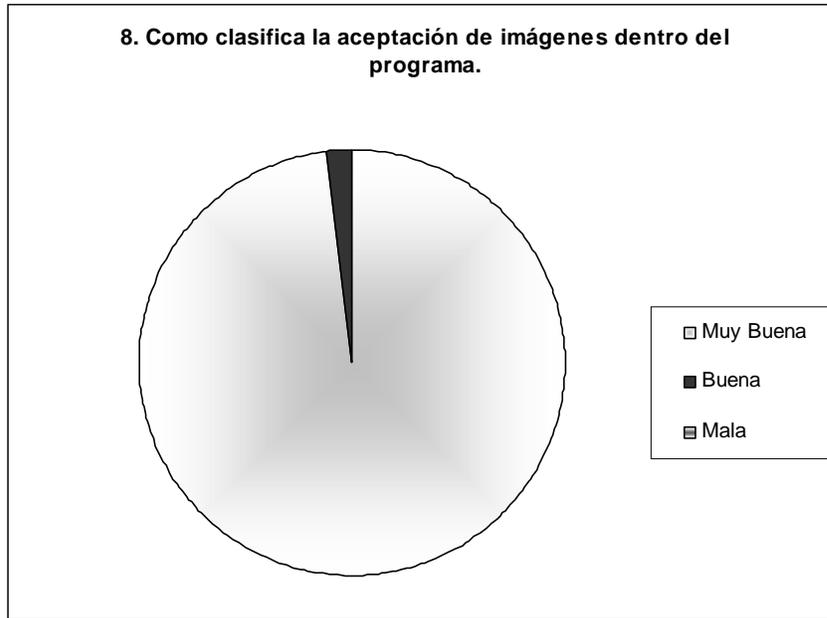
**Figura E-5 Pregunta 5 (Ultima Encuesta).**



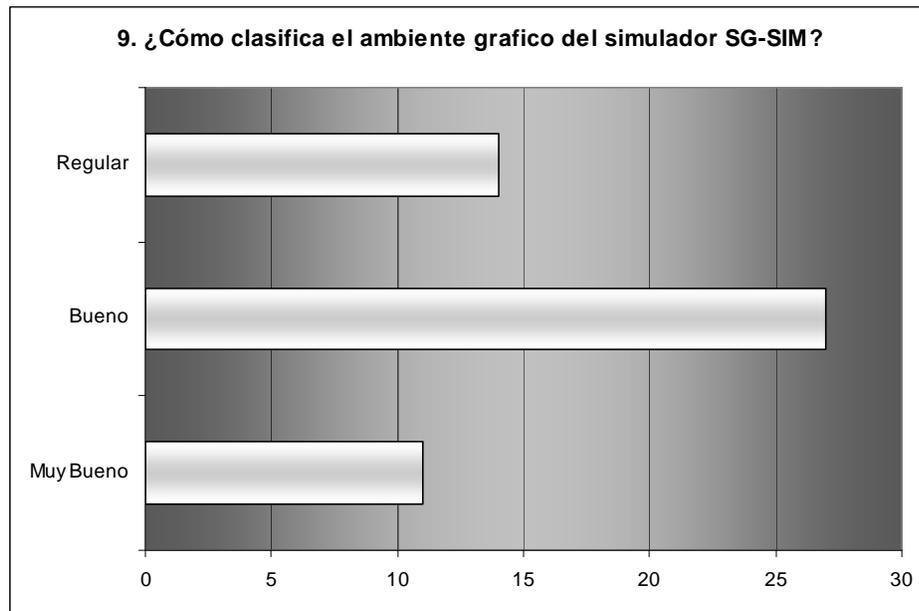
**Figura E-6 Pregunta 6 (Última Encuesta).**



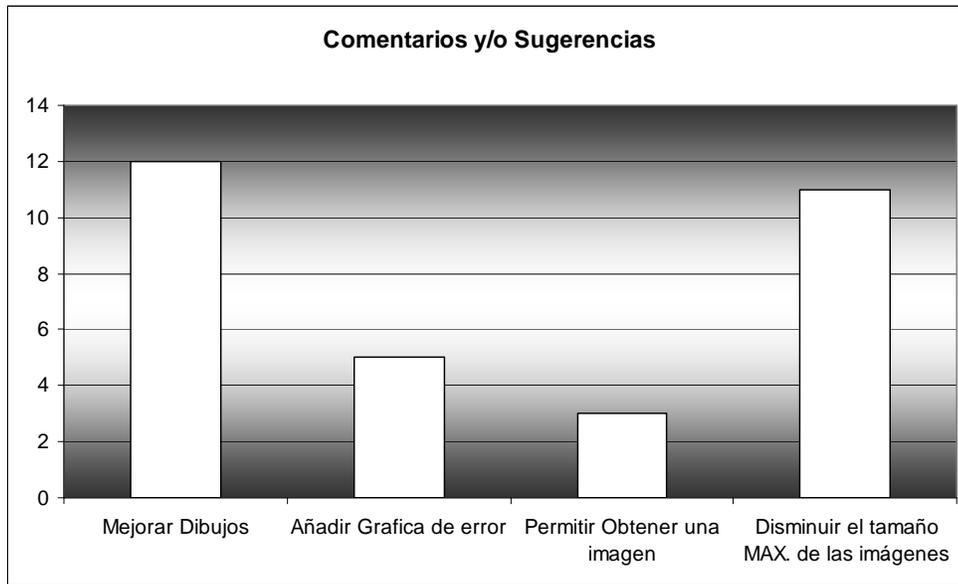
**Figura E-7 Pregunta 7 (Última Encuesta).**



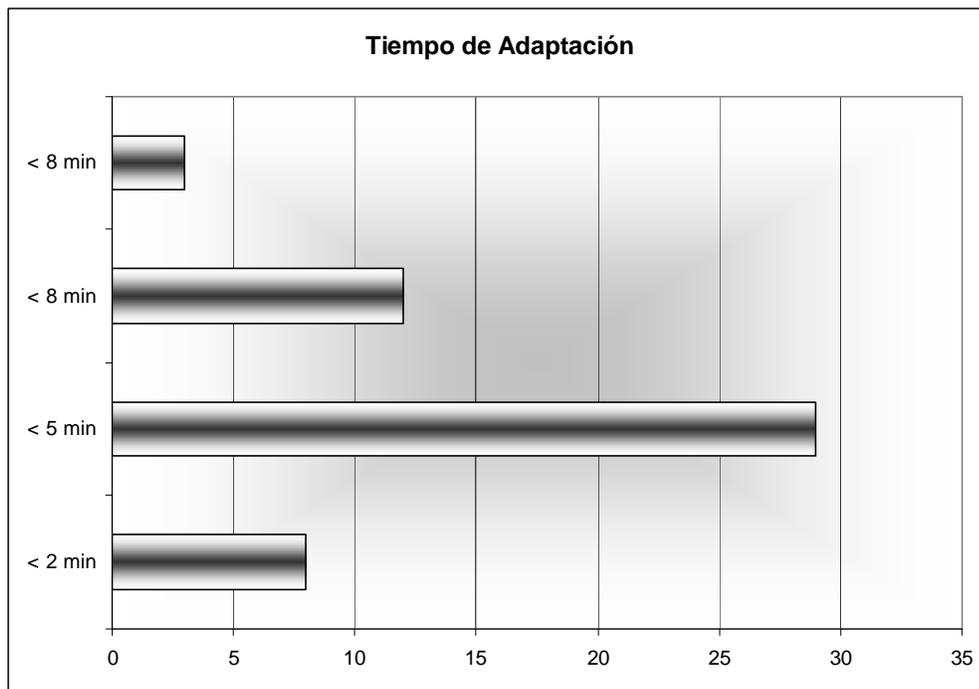
**Figura E-8 Pregunta 8 (Ultima Encuesta).**



**Figura E-9 Pregunta 9 (Ultima Encuesta).**



**Figura E-10 Comentarios y/o Sugerencias.**



**Figura E-11 Tiempo Estimado de Adaptación.**