



**UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL**

**FACULTAD DE CIENCIAS DE LA INGENIERÍA  
CARRERA DE INGENIERÍA INFORMÁTICA Y CIENCIAS  
DE LA COMPUTACIÓN**

**HERRAMIENTA DE APRENDIZAJE DE LAS 4 OPERACIONES  
FUNDAMENTALES CON RECONOCIMIENTO DE CARACTERES  
MANUSCRITOS**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO EN INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN**

**JUAN SEBASTIÁN GARCÍA PINTO**

**DIRECTOR: ING. RODRIGO PROAÑO**

**Quito, Julio 2015**

© Universidad Tecnológica Equinoccial. 2015

Reservados todos los derechos de reproducción

# DECLARACIÓN

Yo **JUAN SEBASTIÁN GARCIA PINTO**, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Tecnológica Equinoccial puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

---

JUAN SEBASTIÁN GARCÍA PINTO

1716384704

## CERTIFICACIÓN

Certifico que el presente trabajo que lleva por título “**herramienta de aprendizaje de las 4 operaciones fundamentales con reconocimiento de caracteres manuscritos**”, que, para aspirar al título de **Ingeniero en Informática y Ciencias de la Computación** fue desarrollado por **Juan Sebastián García Pinto**, bajo mi dirección y supervisión, en la Facultad de Ciencias de la Ingeniería; y cumple con las condiciones requeridas por el reglamento de Trabajos de Titulación artículos 18 y 25.

---

ING. RODRIGO PROAÑO

**DIRECTOR DEL TRABAJO**

1708549041

## **DEDICATORIA**

A Dios por ser la guía y ayudarme a superar los problemas.

A mis padres Dra. Galud Pinto e Ing. Marco García, a mi hermana Soledad García por el amor, la paciencia, el apoyo, la comprensión, los valores que me han inculcado a lo largo de toda la vida.

A mi familia por brindarme su ayuda y siempre estar cuando los he necesitado especialmente a mi abuelita Gladys por todo su amor y sus atenciones.

A todas las personas que me ayudaron, que me brindaron su apoyo, me dieron consejos y siempre estuvieron presentes quiero dedicarles este trabajo de tesis.

## **AGRADECIMIENTO**

El presente trabajo de tesis me gustaría agradecer a Dios por ser el eje fundamental.

A mis padres que sin su apoyo, guía, paciencia no podría haber culminado la carrera, quienes supieron guiarme y enseñarme que sin esfuerzo no se consiguen grandes cosas.

De igual manera agradecer al Director de Tesis Ing. Rodrigo Proaño por su valioso aporte a este trabajo, así como a todos los profesores que en las aulas comparten su conocimiento con los estudiantes

Finalmente agradecer a todas las personas que fueron de gran ayuda en toda la etapa de estudiante, quienes me brindaron su apoyo incondicional, sus ánimos, consejos quiero agradecerles de todo corazón.

# ÍNDICE DE CONTENIDOS

	PÁGINA
<b>RESUMEN</b>	<b>VIII</b>
<b>ABSTRACT</b>	<b>IX</b>
<b>1. INTRODUCCIÓN</b>	<b>2</b>
<b>2 MARCO TEÓRICO</b>	<b>6</b>
<b>2.1 TÉCNICAS DE ESTUDIO EN EDUCACIÓN PRIMARIA</b>	<b>6</b>
<b>2.2 RECONOCIMIENTO DE CARACTERES</b>	<b>7</b>
<b>2.3 REDES NEURONALES ARTIFICIALES</b>	<b>8</b>
2.3.1 ARQUITECTURA	8
2.3.2 SEGÚN SU ESTRUCTURA EN CAPAS	10
2.3.2.1 Redes mono capa	10
2.3.2.2 Redes multicapa	10
2.3.3 SEGÚN EL FLUJO DE DATOS DE LA RED	11
2.3.3.1 Redes unidireccionales o de propagación hacia adelante	11
2.3.3.2 Redes de retro propagación	11
2.3.4 CONSTRUCCIÓN DEL SET DE ENTRENAMIENTO	11
<b>2.4 INTELIGENCIA ARTIFICIAL</b>	<b>12</b>
2.4.1 MODELO NEURONAL	12
2.4.1.1 La neurona artificial	12
2.4.1.2 Función de activación de una neurona artificial.	12
2.4.1.4 Estructura Básica de una Red Neuronal Artificial	15
2.4.1.5 Capacidades de las Redes Neurales Artificiales	16
2.4.2 REDES NEURONALES DE TIPO BIOLÓGICO.	17
2.4.2.1 Redes Neuronales Dirigidas a una Aplicación.	17
2.4.2.2 Redes Neuronales Artificiales Supervisadas y No Supervisadas.	17
2.4.2.3 Reglas de entrenamiento Supervisado	17
2.4.2.4 Reglas de entrenamiento No Supervisado	18
2.4.3 MODELOS NO SUPERVISADOS	19
2.4.3.1 Redes de Memoria Asociativa de Pesos Fijos.	19

2.4.3.2 Modelos Supervisados	19
2.4.4 APRENDIZAJE DE LA RED NEURONAL	19
2.4.4.1 Aprendizaje Hebbiano	19
2.4.4.2 Regla de Entrenamiento del Perceptrón	20
2.4.5 ENTRENAMIENTO DE LAS REDES NEURONALES ARTIFICIALES	21
2.4.5.1 Entrenamiento Supervisado	21
2.4.5.2 Entrenamiento No Supervisado	21
2.4.6 PERCEPTRÓN	22
2.4.6.1 Arquitectura	22
2.4.7 ALGORITMO RETRO PROPAGACIÓN	25
2.4.7.1 Arquitectura de la Red Retro Propagación	25
2.4.7.2 Algoritmo de Entrenamiento	26
2. 4.7.3 Propagación hacia Adelante	27
2.4.7.4 Propagación hacia Atrás	28
2.4.7.5 Ajuste de Pesos de la Capa de Salida	28
2.4.7.6 Ajuste de Pesos de las Capas Ocultas	29
2.4.8 PERCEPTRÓN MULTICAPA	30
<b>2.5 APLICACIÓN WEB</b>	<b>31</b>
2.5.1 PROCESAMIENTO PAGINA WEB ESTATICA	31
<b>3. METODOLOGÍA</b>	<b>34</b>
<b>3.1. ALCANCE</b>	<b>34</b>
<b>3.2. HERRAMIENTAS / TECNICAS</b>	<b>34</b>
<b>3.3. MÉTODOS</b>	<b>34</b>
3.3.1 UNIDAD DE ANÁLISIS	35
3.3.2 MODELO DE PROTOTIPO	35
3.3.2.1 Comunicación	36
3.3.2.2 Plan rápido	37
3.3.2.3 Modelado diseño rápido	37
3.3.2.4 Construcción del prototipo.	37
3.3.2.5 Desarrollo, entrega y retroalimentación	37
<b>4 ANÁLISIS DE RESULTADOS</b>	<b>39</b>
<b>4.1 COMUNICACIÓN</b>	<b>39</b>
4.1.1 FACTIBILIDAD TÉCNICA	39



4.1.2 FACTIBILIDAD ECONÓMICA	40
4.1.3 FACTIBILIDAD OPERATIVA	40
4.1.4 REQUERIMIENTOS NO FUNCIONALES	41
4.1.5 REQUERIMIENTOS FUNCIONALES	41
CASOS DE USO DEL SISTEMA	41
4.1.5.1 Caso de uso ingresar al sistema	42
4.1.5.2 Diagrama secuencial Ingresar al Sistema	45
4.3.5.3 Caso de uso Seleccionar Operación Matemática.	45
4.1.5.4 Diagrama secuencial Selección operación matemática	48
4.1.5.5 Caso de uso Ingresar número.	49
4.1.5.6 Diagrama secuencial ingresar número	51
4.1.5.7 Caso de uso procesar red neuronal	51
4.1.5.8 Diagrama secuencial procesar red neuronal.	53
4.1.5.9 Caso de uso respuesta	54
4.1.5.10 Diagrama secuencial entrenar red neuronal	55
4.1.5.11 Caso de uso visualizar resultado	56
4.1.5.12 Diagrama secuencial visualizar resultado	58
<b>4.2 PLAN RÁPIDO</b>	<b>58</b>
<b>4.3 MODELADO DISEÑO RÁPIDO</b>	<b>61</b>
<b>4.4 CONSTRUCCIÓN DEL PROTOTIPO</b>	<b>62</b>
<b>4.5 DESARROLLO, ENTREGA Y RETROALIMENTACIÓN</b>	<b>64</b>
4.5.2 DESARROLLO DE LAS INTERFACES DE USUARIO Y CÓDIGO FUENTE DE LA APLICACIÓN	64
4.5.2 .1 Entrenamiento	64
4.5.2 .2 Set de Entrenamiento	65
4.5.2.3 Números en el set de entrenamiento	66
<b>4.6 PRUEBAS</b>	<b>82</b>
<b>5 CONCLUSIONES Y RECOMENDACIONES</b>	<b>85</b>
<b>5.1 CONCLUSIONES</b>	<b>85</b>
<b>5.2 RECOMENDACIONES</b>	<b>86</b>
<b>BIBLIOGRAFÍA</b>	<b>89</b>

# ÍNDICE DE FIGURAS

	PÁGINA
<b>Figura 1.</b> Red monocapa recurrente .....	9
<b>Figura 2.</b> Red multicapa .....	9
<b>Figura 3.</b> Función de activación lineal con su expresión matemática.....	13
<b>Figura 4.</b> Función de activación sigmoideal con su expresión matemática .....	13
<b>Figura 5.</b> Función de activación sigmoideal bipolar.....	14
<b>Figura 6.</b> Función de activación escalón.....	14
<b>Figura 7.</b> Redes neurales de una sola capa.....	15
<b>Figura 8.</b> Redes neurales multicapa.....	15
<b>Figura 9.</b> Diagrama del entrenamiento supervisado.....	18
<b>Figura 10.</b> Diagrama del entrenamiento no supervisado.....	18
<b>Figura 11.</b> Unidad procesadora básica del perceptrón.....	23
<b>Figura 12.</b> Red perceptrón de dos capas .....	24
<b>Figura 13.</b> Red perceptron entradas y salidas.....	24
<b>Figura 14.</b> Retro propagación.....	26
<b>Figura 15.</b> Retro propagación conectada completamente.....	26
<b>Figura 16.</b> Fase de propagación hacia adelante .....	28
<b>Figura 17.</b> Ajuste de pesos de salida .....	29
<b>Figura 18.</b> Calculo de los valores de las capas ocultas.....	29
<b>Figura 19.</b> Figura de aplicación web .....	32
<b>Figura 20.</b> Diagrama del modelo de prototipos .....	36
<b>Figura 21.</b> Caso de uso general del sistema .....	41
<b>Figura 22.</b> Ingresar al sistema.....	43
<b>Figura 23.</b> Diagrama secuencial ingresar al sistema.....	45
<b>Figura 24.</b> Caso de uso selección operación matemática .....	46
<b>Figura 25.</b> Diagrama secuencial selección operación matemática.....	48
<b>Figura 26.</b> Ingresar número.....	49
<b>Figura 27.</b> Diagrama ingresar número .....	51
<b>Figura 28.</b> Procesar red neuronal.....	51
<b>Figura 29.</b> Diagrama de secuencia del procesar red neuronal.....	53
<b>Figura 30.</b> Reconocimiento de número .....	54
<b>Figura 31.</b> Diagrama de secuencia entrenar red neuronal. ....	55
<b>Figura 32.</b> Visualizar resultado.....	56
<b>Figura 33.</b> Diagrama de secuencia entrenar red neuronal. ....	58
<b>Figura 34.</b> Diagrama interfaz inicio.....	59
<b>Figura 35.</b> Diagrama interfaz menú principal .....	59

<b>Figura 36.</b> Diagrama interfaz suma básica.....	60
<b>Figura 37.</b> Diagrama interfaz suma media .....	60
<b>Figura 38.</b> Ingreso al menú principal .....	62
<b>Figura 39.</b> Interfaz del menú principal .....	63
<b>Figura 40.</b> Nivel suma básico .....	64
<b>Figura 41.</b> Nivel suma medio.....	64
<b>Figura 42.</b> Hoja de texto del reconocimiento.....	66
<b>Figura 43.</b> Números en fase de entrenamiento .....	67
<b>Figura 44.</b> Página de inicio.....	68
<b>Figura 45.</b> Menú principal de la aplicación .....	69
<b>Figura 46.</b> Suma básica con la respuesta correcta .....	70
<b>Figura 47.</b> Suma básica con la respuesta incorrecta .....	71
<b>Figura 48.</b> Suma medio .....	74
<b>Figura 49.</b> Resta básica .....	76
<b>Figura 50.</b> Resta medio .....	77
<b>Figura 51.</b> Multiplicación nivel básica.....	78
<b>Figura 52.</b> Multiplicación nivel medio .....	79
<b>Figura 53.</b> División nivel básica.....	80
<b>Figura 54.</b> División nivel medio .....	81

## ÍNDICE DE TABLAS

	PÁGINA
<b>Tabla 1.</b> Factibilidad técnica.....	39
<b>Tabla 2.</b> Factibilidad Económica.....	40
<b>Tabla 3.</b> Entrenamiento Red Neuronal Artificial.....	44
<b>Tabla 4.</b> Selección operación matemática.....	47
<b>Tabla 5.</b> Dibujar número.....	50
<b>Tabla 6.</b> Reconocimiento de número.....	53
<b>Tabla 7.</b> Reconocimiento de la Respuesta.....	55
<b>Tabla 8.</b> Visualizar respuesta.....	57

## ÍNDICE DE ANEXOS

	PÁGINA
<b>Anexo 1.</b> Manual de usuario.....	92
<b>Anexo 2.</b> Carta pruebas iniciales.....	101

## RESUMEN

Este trabajo tiene como finalidad desarrollar una aplicación que pueda reconocer caracteres manuscritos, aplicados al aprendizaje numérico de las 4 operaciones matemáticas básicas. Esta aplicación permite generar números aleatorios en 2 niveles de dificultad que son básico y medio, sobre las 4 operaciones fundamentales, la aplicación reconoce el carácter dado por el usuario, genera respuesta matemática sobre el número que reconoce y si esta correcta o incorrecta la respuesta. El reconocimiento de caracteres se implementó utilizando técnicas de inteligencia artificial, el proceso de reconocimiento de la respuesta de las operaciones fundamentales fue implementado con algoritmos matemáticos y se integró las 2 técnicas para obtener la respuesta final. Para el trabajo se investigó algunas técnicas de reconocimiento de caracteres como el Perceptron Multicapa y retro-propagación, algoritmos y librerías como Neuroph que permitieron el reconocimiento del carácter manuscrito y su interacción con el sistema web.

## **ABSTRACT**

This work aimed to develop an application that can recognize handwritten characters, applied to the numerical learning of the 4 basic mathematic operations. This application allows you to generate random numbers in 2 different difficulty levels those are basic and intermediate, on the 4 fundamental operations, the application recognizes the character given by the user, generates mathematical answer on the number that recognizes and if this answer is right or wrong. Character recognition is implemented using artificial intelligence techniques, the process of recognition of the response of the fundamental operations was implemented with mathematical algorithms and joined the 2 techniques to get the final answer. For the thesis work some character recognition techniques such as Multilayer Perceptron and retro-propagation algorithms and libraries as Neuroph that allowed the manuscript character recognition and its interaction with the web system was investigated.

## **INTRODUCCIÓN**



# 1. INTRODUCCIÓN

En la actualidad existen herramientas virtuales que permiten la realización de las 4 operaciones matemáticas básicas, estas herramientas ingresan la información a través del teclado, tanto las operaciones como el resultado, lo que hace al sistema monótono.

Con este proyecto se pretende realizar una aplicación que guíe de manera personalizada la realización de las 4 operaciones matemáticas fundamentales estas son suma, resta, división y multiplicación de tal manera que el usuario escriba su respuesta y la herramienta con técnicas Inteligentes reconozca el patrón que se ingresó y con algoritmos matemáticos indique si la respuesta es correcta o incorrecta.

Para la realización de esta herramienta se va a aplicar técnicas de inteligencia artificial específicamente reconocimiento de caracteres.

El reconocimiento de caracteres abarca un conjunto de métodos y algoritmos que permiten realizar una fase de entrenamiento que permitirá reconocer de forma automática caracteres.

En nuestros días se vive en una sociedad completamente digitalizada y es bueno tener herramientas que nos ayuden en tareas de aprendizaje.

Los niños cuando están aprendiendo a realizar las operaciones matemáticas básicas tienden a cometer errores tanto en la escritura como en la lógica de la operación. Con esta herramienta se pretende brindar mayor facilidad al usuario para que pueda aprender de una manera más dinámica desde la escritura de los números como el del proceso lógico de las 4 operaciones matemáticas fundamentales estas son suma, resta, división y multiplicación.

La herramienta tendrá la funcionalidad de capturar la respuesta que el usuario proporciona a la operación matemática y mediante redes neuronales artificiales aplicadas al reconocimiento de caracteres así como también métodos como el modelo de Kohonen, el Perceptron multicapa y retro propagación, realizan el proceso de reconocimiento del carácter y con algoritmos matemáticos evaluar la respuesta, es aquí donde se va a personalizar si el niño está equivocado en la respuesta y la herramienta pueda asistir en la enseñanza de la operación con la respuesta correcta, así el usuario incremente su destreza realizando más ejercicios de las 4 operaciones matemáticas.

El reconocimiento de caracteres es una de las aplicaciones que en la actualidad cuenta con mayor difusión dentro de la utilización de redes neuronales artificiales, por lo que se pretende realizar esta herramienta para una asistencia personalizada al usuario en este caso al niño.

Los niños al estar aprendiendo a escribir cometen errores en la escritura del número así como también están aprendiendo a realizar la operación correctamente, con esta herramienta se pretende la asistencia al niño en la etapa de aprendizaje de las operaciones matemáticas fundamentales, como en la de la escritura de los números, el usuario deberá escribir correctamente el número para que la herramienta reconozca de que se trata.

El reconocimiento de caracteres, método de Perceptron Multicapa, retro propagación y aprendizaje no supervisado son técnicas de Inteligencia Artificial que va a ayudar a la realización de esta herramienta.

El objetivo general es desarrollar una herramienta web que mediante técnicas de inteligencia artificial, reconozca los parámetros ingresados de manera manual al sistema con el fin de que el usuario se adiestre en la elaboración de los números así como en la lógica de la operación logrando una mayor facilidad en el aprendizaje de las 4 operaciones fundamentales en niños de edad escolar.

Los objetivos específicos del proyecto son:

1. Reconocer los números y las respuestas formuladas por el usuario las cuales serán ingresadas manualmente por medio del mouse, pantalla táctil de cualquier dispositivo que tenga acceso a internet y estas serán reconocidas por el modelo de red neuronal.
2. Evaluar los resultados ingresados por el usuario a través del reconocimiento de caracteres y posteriormente con algoritmos matemáticos verificar su respuesta.
3. Asistir al niño cuando este cometa un error o cuando el niño se equivoque en la respuesta brindándole nuevos ejercicios para que siga con el proceso de aprendizaje y siga escribiendo números para llegar a dibujarlos correctamente.

## **MARCO TEÓRICO**

## 2 MARCO TEÓRICO

### 2.1 TÉCNICAS DE ESTUDIO EN EDUCACIÓN PRIMARIA

La técnica de estudio es una herramienta que ayuda y facilita el estudio para alcanzar los objetivos, estas técnicas son un conjunto de procedimientos y estrategias de carácter meta cognitivo y cognitivo que se desarrollan a través de la práctica.

Sin embargo para el desarrollo de dichas habilidades de estudio por parte de los niños es importante la colaboración de los padres y de las madres.

- **Crear un buen ambiente de estudio en casa:** Tener un lugar fijo donde pueda estudiar, que contenga el material de estudio necesario aislado de ruidos y distracciones.
- **Establecer un tiempo diario dedicado al estudio:** Que se establezca un horario fijo en la tarde indicando el tiempo de estudio y el tiempo que se da a otras actividades de ocio, revisar el material escolar y notas de los profesores así como también los libros del día siguiente.
- **Dando ejemplo de interés por la cultura:** Esto se realiza a través de la lectura, biblioteca familiar, conversaciones, etc.
- **Mantener una estrecha colaboración con los profesores:** Debe existir una colaboración tanto con padres como con los profesores.
- **Animar al estudio sin sermonear:** Se demuestra cuánto vale un pequeño esfuerzo ya que los sermones pueden llegar a ser contraproducentes y es mejor una pequeña motivación.
- **Seguir diariamente el hábito de estudio de los niños pero sin agobiar:** Hay que interesarse por los exámenes pruebas tareas, que materia tiene más dificultades pero hay que guardar un equilibrio entre el interés y la presión por que mejore. (familias apoclam, 2013)

## 2.2 RECONOCIMIENTO DE CARACTERES

El reconocimiento de caracteres es un campo que ya tiene sus años de estudio. Interfaces con las entradas manuscritas han sido investigadas desde 1959. El desarrollo de escáneres ópticos de alta resolución y Tablet digitalizadoras durante la década de 1960 y principios de 1970 mejoró el reconocimiento tanto el carácter como el reconocimiento de la escritura. La mayoría de los métodos son los métodos de procesamiento fuera de línea que utilizan cadenas de información contextual para aumentar la precisión del reconocimiento.

En el reconocimiento de caracteres en línea la señal de entrada es la posición de la punta de la pluma, cuando se refiere a la punta, se habla de la presión que esta emite en un sistema que se aplica a dispositivos como Tablet en donde la señal de entrada es cuantificada de 1 bit sobre la superficie de escritura.

La segmentación se lleva a cabo mediante la construcción de una cadena de caracteres que se denominan caracteres candidatos de la cadena adquirida por los trazos realizados en la superficie de la Tablet.

En general, los métodos lineales restan precisión para velocidades de reconocimiento en tiempo real. En la actualidad, la mayoría de la investigación en el reconocimiento de caracteres lineales se ha centrado en torno a los sistemas de un solo carácter de entrada [2, 3, 7, 8]. Los caracteres se introducen uno a la vez.

Esto proporciona al usuario una retroalimentación inmediata para que los errores se puedan corregir a medida que ocurren. Típicamente, existe un método simple para que el usuario pueda conocer el comienzo y el final de cada carácter, el Unistrokes desarrollado por Xerox Corporation en 1993 es reconocida como un ejemplo de un carácter único.

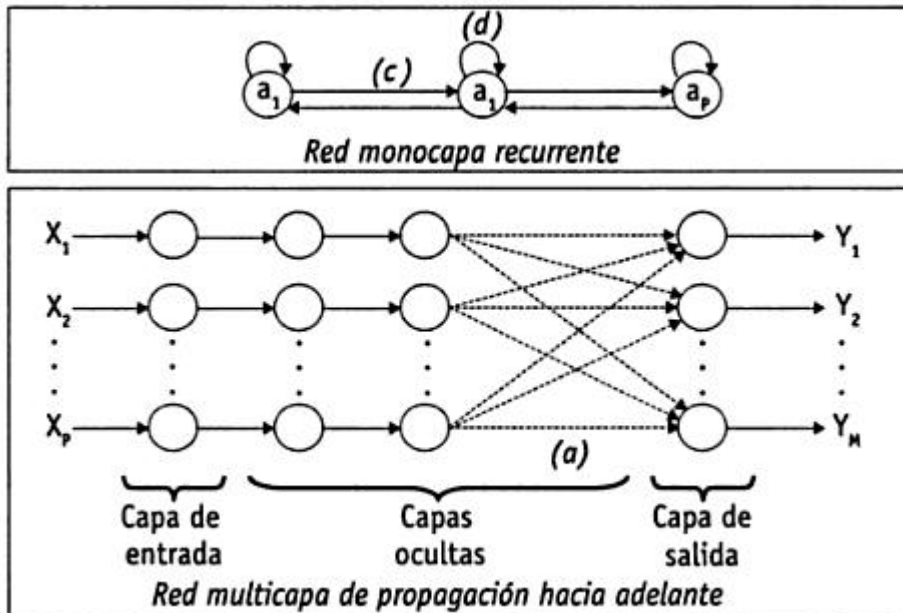
Los caracteres Unistrokes fueron desarrollados para ser escritos uno arriba del otro a fin de minimizar el estado real requerido para el reconocimiento. (Confer, Chapman, 2000) (Parizeau, Lemieux, Gagné)

## **2.3 REDES NEURONALES ARTIFICIALES**

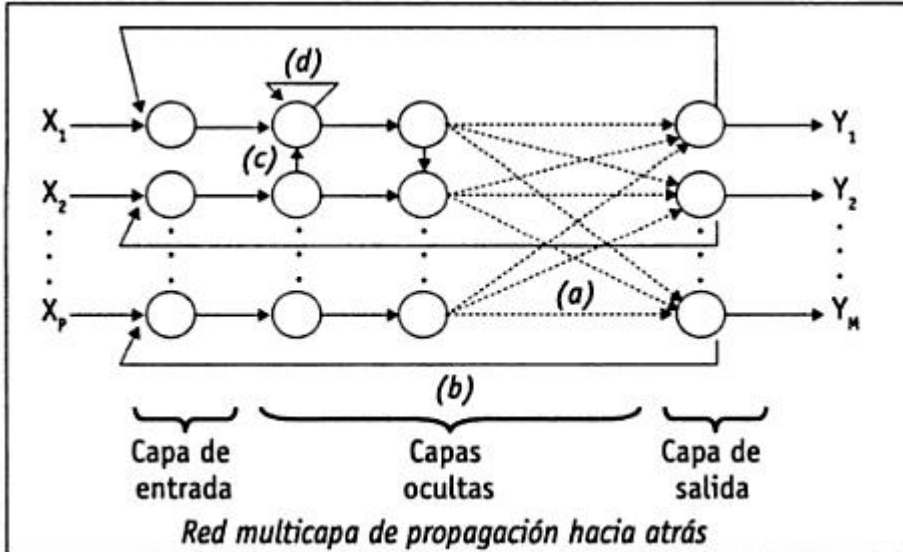
### **2.3.1 ARQUITECTURA**

El método común para el diseño de sistemas de reconocimiento de caracteres es el de arriba hacia abajo. Ambos enfoques hardware y software se han utilizado en el problema del reconocimiento de caracteres con buenos resultados. Un conjunto de características y un método de extracción de características son seleccionados y el problema de clasificación resultante se resuelve mediante una red neuronal. La forma general de los campos receptores de entrada que se emplean en tareas como visión binocular de los vertebrados se ha modelado mediante funciones paralelas de Dennis Gabor. La salida de los estos campos receptores están acopladas a pequeñas redes para detectar la posición.

El modelo no fue diseñado específicamente para reconocimiento de caracteres y se puede enseñar cualquier conjunto de imágenes que podrían estar representados por las funciones de Dennis Gabor. Las funciones de Dennis Gabor se adaptan bien a este tipo de aplicaciones de reconocimientos de caracteres, ya que permiten la reconstrucción de imagen de calidad razonable, con un pequeño número de funciones de base.



**Figura 1.** Red Monocapa Recurrente  
(Flores, 2008)



**Figura 2.** Red Multicapa  
(Flores, 2008)



Este método de extracción de características no es biológico, pero está claramente bien definido en términos de procesamiento numérico convencional. Las salidas del Campos de Recepción de Gabor son las entradas a una red neuronal entrenada mediante el aprendizaje de propagación hacia atrás o retro propagación. La activación de las neuronas de salida de la red de propagación hacia atrás se utiliza para clasificar las imágenes de entrada.

La disminución del número de funciones de base requeridos reduce el número de conexiones necesarias en la clasificación red.

La arquitectura de la red neuronal se puede clasificar de la siguiente forma:

### **2.3.2 SEGÚN SU ESTRUCTURA EN CAPAS**

#### **2.3.2.1 Redes mono capa**

Esta red está compuesta por una única capa de neuronas, la misma que conecta con sus capas laterales, este tipo de red se utiliza para auto asociación.

#### **2.3.2.2 Redes multicapa**

Esta red está compuesta por varias capas tanto de entrada, capa oculta como la capa de salida.

### **2.3.3 SEGÚN EL FLUJO DE DATOS DE LA RED**

#### **2.3.3.1 Redes unidireccionales o de propagación hacia adelante**

Esta red tiene como característica que ninguna capa de salida es capa de entrada, es decir que la información circula en un único sentido desde la capa de entrada hacia la capa de salida.

#### **2.3.3.2 Redes de retro propagación**

Esta red tiene como característica que las salidas de las neuronas pueden ser usadas como entradas, redes de retro propagación que presentan lazos cerrados se denomina sistemas recurrentes.

### **2.3.4 CONSTRUCCIÓN DEL SET DE ENTRENAMIENTO**

Las redes de retro propagación proporcionan un método muy eficaz para llevar a cabo la clasificación supervisada no lineal. Este aprendizaje puede ser mejorado presentando ejemplos de la red en función de espacio n-dimensional. La presentación de una red con ejemplos de esta naturaleza ofrece una mayor oportunidad para el aprendizaje de las relaciones funcionales que rigen estos límites. Esta estrategia para la elección de los prototipos de capacitación con el apoyo de componentes principales y el análisis de las estadísticas que se utilizan más los valores propios de una matriz de covarianza para localizar la dirección de la máxima variación en función del espacio.

Sin embargo, derivando componentes principales, ya sea métodos numéricos convencionales o las redes neuronales es computacionalmente costosa. El nuevo método de clasificación superior se ha desarrollado asumiendo que los componentes del vector son estadísticamente independientes. (Garris, 1991)

## 2.4 INTELIGENCIA ARTIFICIAL

### 2.4.1 MODELO NEURONAL

#### 2.4.1.1 La neurona artificial

Las Redes Neuronales Artificiales partieron de una simulación abstracta de los sistemas nerviosos biológicos formado por el conjunto de unidades llamadas “nodos” o “neuronas” que se interconectaban unas con otras, en los sistemas nerviosos biológicos esto tiene semejanza con las dendritas y axiomas.

El primer modelo de red neuronal artificial fue propuesto por McCulloch y Pitts en el año de 1943 en términos de un modelo computacional de actividad nerviosa simulando la red neuronal biología, este modelo es un modelo binario y cada neurona tiene un escalón de umbral prefijado.

#### 2.4.1.2 Función de activación de una neurona artificial.

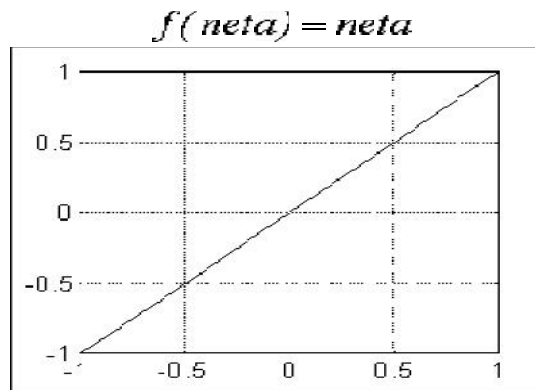
Para las neuronas que forman la red neuronal artificial existe un conjunto de conexiones que las unen, cada neurona transmite señales a aquellas neuronas que están conectadas a su salida. “Asociada con cada unidad  $U_j$ , hay una función de activación  $f_{(Fact)}$ , que transforma la entrada  $neta$  esta unidad presenta como resultado al estar conectada con otras unidades que le suministran información, en el valor de salida  $y_j$ ”.

$$y_j(t) = Fact_j(Neta_j(t))$$

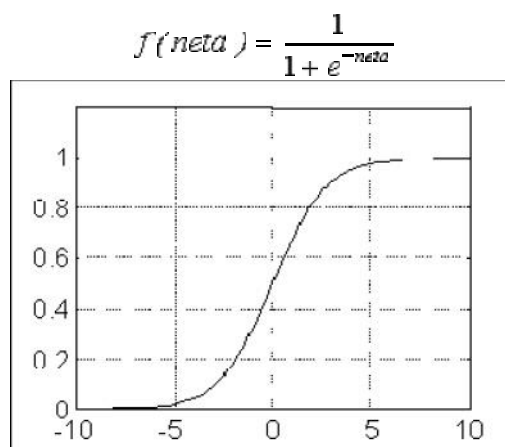
Calculo de la salida evaluando la neta en la función de Activación.

Entre las funciones de activación más utilizadas se tienen:

Función de Activación Lineal

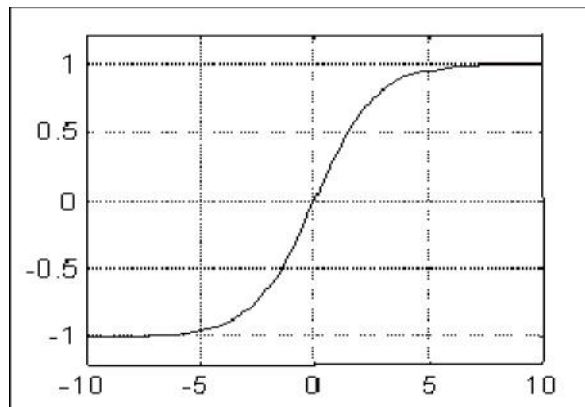


**Figura 3.** Función de Activación Lineal con su expresión matemática.  
(López, 2000)



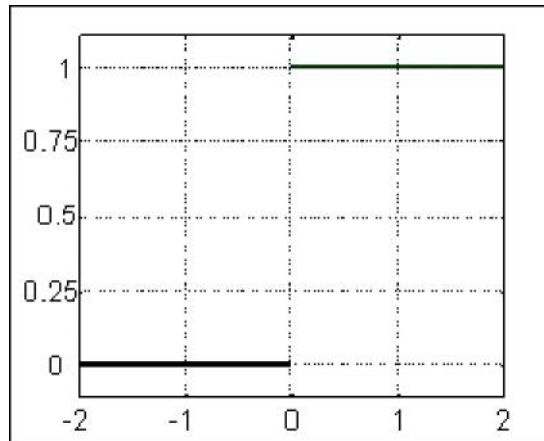
**Figura 4.** Función de Activación Sigmoidal con su expresión matemática.  
(López, 2000)

$$f(\text{neta}) = \frac{2}{1 + e^{-\text{neta}}} - 1$$



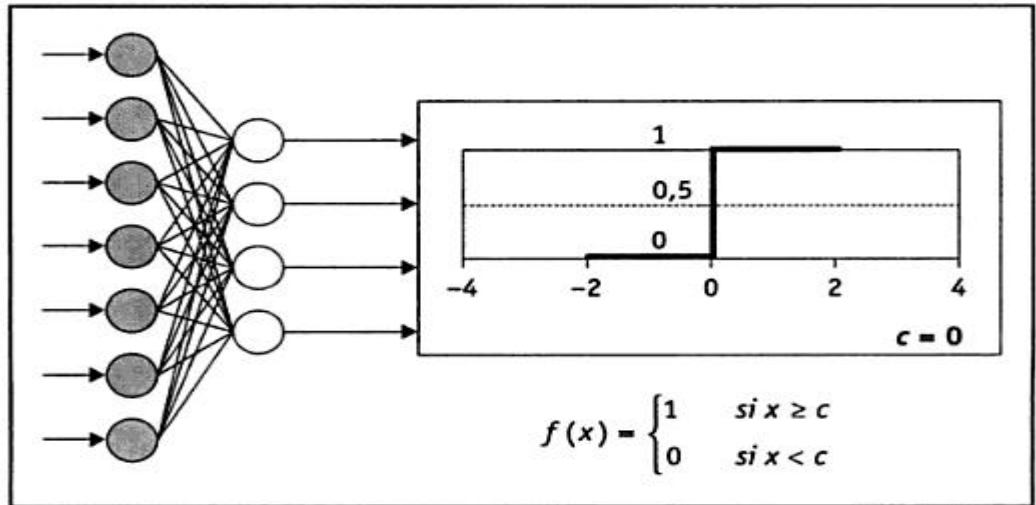
**Figura 5.** Función de Activación Sigmoide Bipolar con su expresión matemática.  
(López, 2000)

$$f(\text{neta}) = \begin{cases} 1 & \text{neta} \geq 0 \\ 0 & \text{neta} < 0 \end{cases}$$

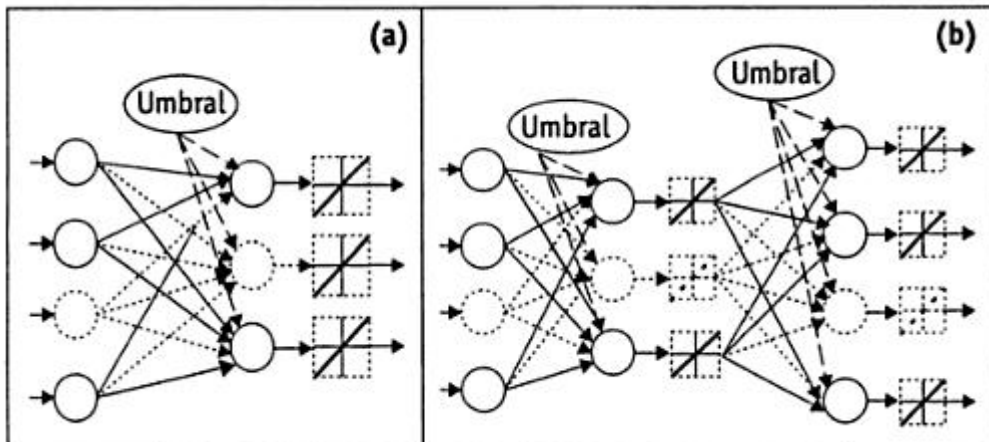


**Figura 6.** Función de Activación Escalón con su expresión matemática.  
(López, 2000)

#### 2.4.1.4 Estructura Básica de una Red Neuronal Artificial



**Figura 7.** Redes neurales de una sola capa  
(Flores, 2008)



**Figura 8.** Redes neurales multicapa  
(Flores, 2008)

### 2.4.1.5 Capacidades de las Redes Neuronales Artificiales

La neurona artificial fue creada como ya se dijo antes para simular o emular las características de funcionamiento básico de una neurona biológica. Básicamente, se aplica un conjunto de entradas a la neurona, cada una de estas entradas representa una salida de una neurona, cada entrada es multiplicada por su peso o ponderación correspondiente análoga al grado de conexión de la sinapsis. Se suman todas las entradas y así determinamos el nivel de excitación o activación de la neurona. Una representación vectorial se expresa en la siguiente ecuación.

$$RED (NET) = X * W$$

Siendo  $RED$  la salida,  $X$  el vector de entrada y  $W$  el vector de pesos.

La señal de salida  $RED$  es procesada por una función de activación  $F$  para producir la señal de salida de la neurona  $OUT$  (salida). La Función  $F$  puedes ser una función umbral, lineal o no lineal que emula con mayor exactitud las características de transferencia no lineales de las neuronas biológicas.

Normalmente la señal de salida  $NET$  suele ser procesada por una función de activación

$F$  para producir la señal de salida de la neurona  $OUT$ . La función  $F$  puede ser una función lineal, o una función umbral o a su vez una función no lineal que simula con mayor exactitud las características de transferencia no lineales de las neuronas biológicas.

Las funciones  $F$  más utilizadas son la función Sigmoid y Tangente hiperbólica.

## **2.4.2 REDES NEURONALES DE TIPO BIOLÓGICO.**

### **2.4.2.1 Redes Neuronales Dirigidas a una Aplicación.**

Las Redes Neuronales Artificiales tienen poca conexión con las Redes Neuronales Biológicas ya que no se dispone de mucho conocimiento del sistema nervioso en su totalidad, por lo cual se debe definir otra estructura y otras funcionalidades de conexión distintas a las Redes Neuronales Biológicas. Como características generales de las Redes Neuronales Artificiales se tiene:

**Auto Organización y Adaptabilidad:** Como su nombre lo indica utilizan algoritmos de aprendizaje adaptativo y auto organización por lo que se puede adaptar de una manera más eficiente.

**Procesado No Lineal:** Incrementa la capacidad de red de aproximar y también tiende a ser más robusta contra el ruido.

**Procesado paralelo:** se utiliza para el procesamiento de alto nivel de interconectividad.

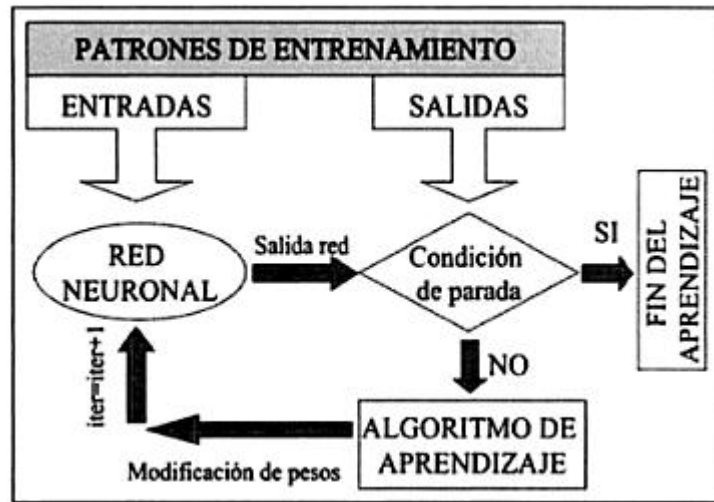
### **2.4.2.2 Redes Neuronales Artificiales Supervisadas y No Supervisadas.**

Las Redes Neuronales Artificiales se clasifican en 3 tipos: redes de pesos fijos, redes no supervisadas, y redes de entrenamiento supervisado. Para las redes de pesos fijos no existe ningún tipo de entrenamiento.

### **2.4.2.3 Reglas de entrenamiento Supervisado**

Para el entrenamiento supervisado constituye de varios pares de patrones de entrada y salida, al conocer la salida implica que este entrenamiento se beneficia la supervisión de un maestro.





**Figura 9.** Diagrama del entrenamiento supervisado  
(Hernandez,2006)

#### 2.4.2.4 Reglas de entrenamiento No Supervisado

Para el entrenamiento no supervisado los datos de entrenamiento solo consisten en los patrones de entrada, la red queda sin beneficio de un maestro y es por lo cual la red empieza a adaptarse basado en las experiencias de los patrones de entrenamiento anteriores



**Figura 10.** Diagrama del entrenamiento no supervisado  
(Hernández, 2006)

## **2.4.3 MODELOS NO SUPERVISADOS**

### **2.4.3.1 Redes de Memoria Asociativa de Pesos Fijos.**

Una de las clases de este entrenamiento es la red de pesos fijos. La principal característica de redes asociativas de pesos fijos son pesos pre establecidos o pre calculados.

### **2.4.3.2 Modelos Supervisados**

#### **2.4.3.2.1 Redes Neuronales Artificiales basadas en Decisión.**

El Perceptrón lineal aplica cuando se sabe que las clases de patrones son separables por límites de decisión lineales. (Basogain, 2004)

## **2.4.4 APRENDIZAJE DE LA RED NEURONAL**

### **2.4.4.1 Aprendizaje Hebbiano**

Tal vez el más influyente trabajo en la historia de conexionismo es la contribución de Hebb (1949), en la que presentó una teoría del comportamiento base, tanto como sea posible, sobre la fisiología del sistema nervioso.

El concepto más importante que sale del trabajo de Hebb fue su declaración formal (conocido como el postulado de Hebb) de cómo puede ocurrir el aprendizaje. El aprendizaje se basa en la modificación de las conexiones sinápticas entre neuronas.

En concreto, cuando un axón de la célula A está lo suficientemente cerca para excitar una célula B y repetida o persistentemente toma parte en el disparo de ella, algún proceso de crecimiento o cambio metabólico se lleva a cabo en una o en ambas células de tal manera que la eficiencia de A, como una de los células de excitación de B, se incrementa. Los principios que subyacen a esta declaración se han conocido como Hebbian Learning o aprendizaje de Hebbian.

Prácticamente, la mayoría de las técnicas de aprendizaje de redes neuronales puede ser considerada como una variante de la regla de aprendizaje de Hebb. La idea básica es que si dos neuronas se activan simultáneamente, su interconexión debe ser fortalecida. (Ajith Abraham, 2004)

#### **2.4.4.2 Regla de Entrenamiento del Perceptrón**

El perceptrón es una sola capa de red neuronal cuyos pesos podrían ser entrenados para producir un correcto vector objetivo cuando se presenta con la entrada correspondiente vectorial.

La técnica de entrenamiento utilizado se denomina regla “Perceptron learning” o aprendizaje perceptrón. Perceptrones son especialmente adecuados para problemas sencillos en clasificación de patrones.

Se tiene un conjunto de muestras de aprendizaje consistentes de un vector  $x$  de entrada y una de salida deseada  $d^{(k)}$ .

Para una tarea de clasificación, el  $d^{(k)}$  es por lo general +1 o -1.

“La regla perceptrón-aprendizaje es muy simple y se puede afirmar a continuación: (Hastiey & Simardz, 1997)

1. Se empieza con pesos aleatorios para las conexiones.
2. Se elige un vector de entrada  $x$  del conjunto muestras de entrenamiento.
3. Si la salida  $y_k$  diferente  $D^{(k)}$  (el perceptrón da una respuesta incorrecta), modificar las conexiones  $w_i$  según:

$$w_i = (dk - yk) xi ; ( = tasa de aprendizaje).$$

4. Vuelva al paso 2.”

Hay que tomar en cuenta que el procedimiento es muy similar a la regla de Hebb, la única diferencia es que cuando la red responde correctamente, los pesos sin conexión son modificados. (Hastiey & Simardz, 1997)

## **2.4.5 ENTRENAMIENTO DE LAS REDES NEURONALES ARTIFICIALES**

La característica principal de las redes neuronales artificiales es su capacidad de aprendizaje. El entrenamiento de las redes neuronales artificiales tiene similitudes con el desarrollo intelectual de las personas. Sin embargo hay que tomar a consideración que el aprendizaje de las redes neuronales artificiales es limitado.

La parte principal del entrenamiento de una red neuronal artificial es que la aplicación, para un conjunto de entradas genere el conjunto de salidas deseadas. El proceso de entrenamiento consiste en la aplicación secuencial de diferentes procesos o vectores de entrada para que se ajusten a los pesos de las interconexiones todo esto mediante un proceso predeterminado.

Durante el entrenamiento los pesos convergen gradualmente hacia los valores que produce que las entradas generen vectores de salida deseados.

Estos algoritmos de entrenamiento se dividen en 2 grupos: Supervisado y No Supervisado

### **2.4.5.1 Entrenamiento Supervisado**

Estos algoritmos necesitan del emparejamiento de cada vector de entrada con su correspondiente vector de salida. "El entrenamiento consiste en presentar un vector de entrada a la red, calcular la salida de la red, compararla con la salida deseada, y el error o diferencia resultante se utiliza para realimentar la red y cambiar los pesos de acuerdo con un algoritmo que tiende a minimizar el error."

### **2.4.5.2 Entrenamiento No Supervisado**

El sistema neuronal con entrenamiento asistido no es muy lógico desde el punto de vista biológico ya que se hace difícil que el cerebro compare las salidas

deseadas con las salidas reales ya que no se sabría diferenciar cual es la salida deseada en caso de que exista.

El sistema con entrenamiento no supervisado son más lógicos desde el punto de vista biológico. Este modelo fue desarrollado por Kohonen en 1984 y otros investigadores, en este modelo no se requieren de un vector de salidas deseadas por lo cual no se realiza la comparación entre salidas reales y salidas esperadas. El conjunto de vectores de entrenamiento se forma únicamente de los vectores de entrada. EL algoritmo de entrenamiento de este modelo modifica los pesos de la red de tal forma que produzca vectores de salida consistentes.

“El proceso de entrenamiento extrae las propiedades estadísticas del conjunto de vectores de entrenamiento y agrupa en clases los vectores similares.”  
(Basogain, 2004)

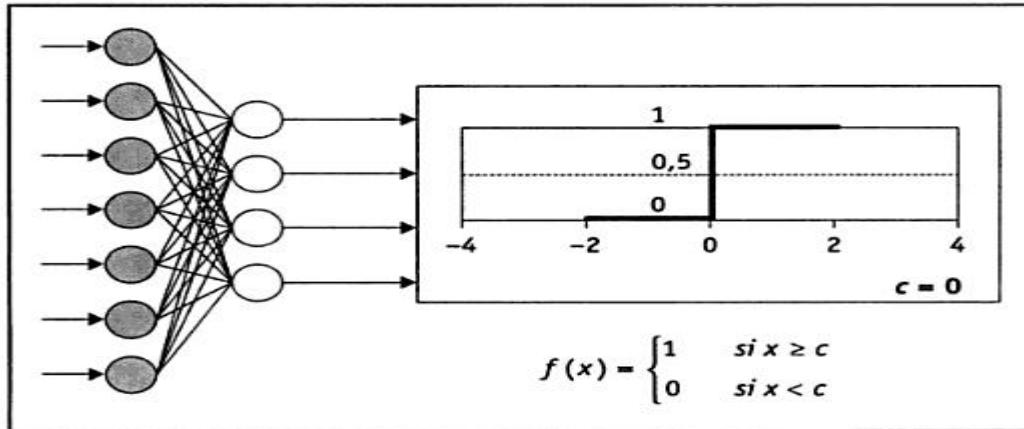
## **2.4.6 PERCEPTRÓN**

### **2.4.6.1 Arquitectura**

La arquitectura del Perceptrón también llamada mapeo de patrones (pattern-mapping), aprende a clasificar los modelos a través del aprendizaje supervisado, generalmente este modelo suele contener vectores con valores binarios (valores entre 0,1).

El Perceptrón está representado por dos capas de unidades procesadores (PE) y en tan solo una de las capas presenta la capacidad de modificar o adaptar los pesos de las conexiones. Esta arquitectura permite capas adicionales que no disponen de la capacidad de modificar o alterar sus propias conexiones.

“En el gráfico se muestra la unidad procesadora básica del Perceptrón. Las entradas  $a_i$  llegan por la parte izquierda, y cada conexión con la neurona  $j$  tiene asignada un peso de valor  $w_{ji}$ .”

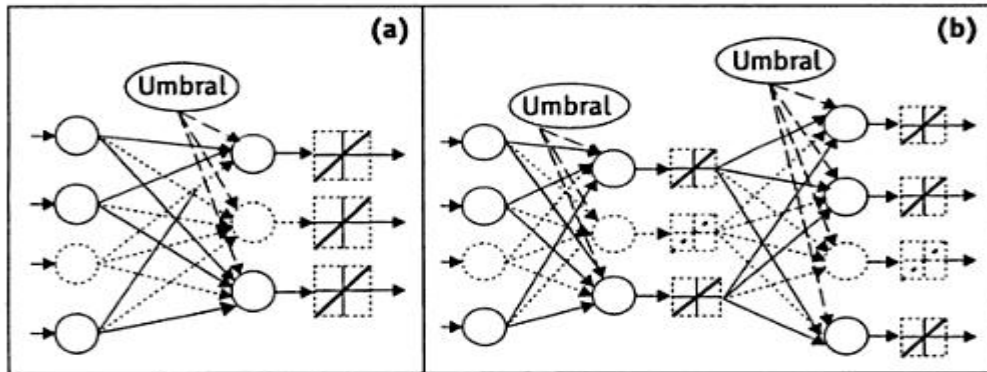


**Figura 11.** Unidad procesadora básica del Perceptrón  
(Basogain, 2004)

“La unidad procesadora del Perceptrón realiza la suma ponderada de las entradas según la ecuación

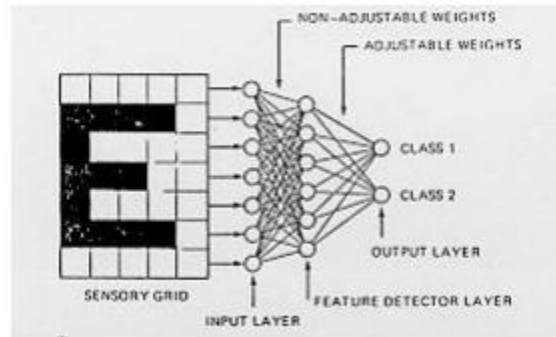
$$S_j = \sum a_i w_{ji}$$

En el siguiente grafico se muestra la red Perceptrón de dos capas con capas de entrada y de salida.



**Figura 12.** Red Perceptrón de dos capas  
(Basogain, 2004)

En la figura se muestra al Perceptrón donde la entrada es la imagen de la letra E y la salida es la categorización de la entrada en dos clases.



**Figura 13.** Red Perceptron entradas y salidas  
(Basogain, 2004)

#### 2.4.6.2 Entrenamiento del Perceptrón

El entrenamiento del Perceptrón consiste en presentar a la red todos los elementos del conjunto de entrenamiento constituido por parejas de vectores (entrada y salida deseada) de forma secuencial.

El objetivo del entrenamiento es llegar a un conjunto de valores de los pesos de la red de forma que responda correctamente a todo el conjunto de entrenamiento. Después del entrenamiento los pesos no son ya modificados y

la red está ya en disposición de responder adecuadamente a las entradas que se le presenten.

La adaptación de los pesos se puede realizar mediante diferentes reglas.

En todo proceso de entrenamiento el comportamiento de la red inicialmente va mejorando hasta que llega a un punto en el que se estabiliza y se dice que la red ha Convergió. Esta convergencia tiene dos posibilidades, la primera consiste en que la red haya aprendido correctamente el conjunto de entrenamiento o la segunda se trata de que la red no ha aprendido todas las respuestas correctas.

## **2.4.7 ALGORITMO RETRO PROPAGACIÓN**

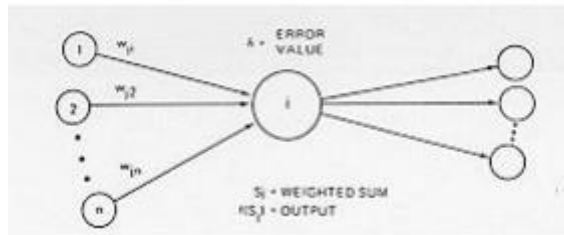
Red retro propagación es un método de entrenamiento multicapa. Su punto fuerte se concentra en la capacidad de entrenar capas ocultas y así superar posibilidades limitadas de las redes de una capa.

“El concepto básico de retro propagación fue presentado en 1974 por Paul Werbos e independientemente reinventado por David Parker en 1982, y también presentado en 1986 por Rumelhart, Hinton y Willians.”

### **2.4.7.1 Arquitectura de la Red Retro Propagación**

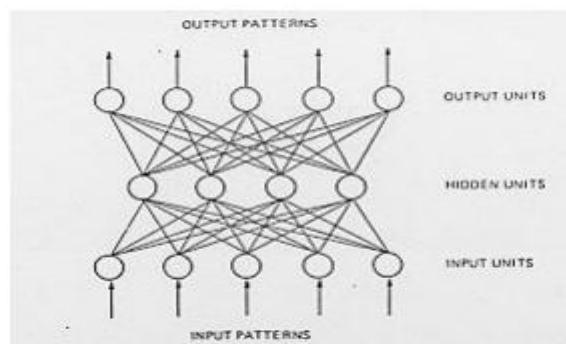
“La unidad procesadora se caracteriza por realizar una suma ponderada de las entradas llamada  $S_j$ , presentar una salida  $a_j$  y tener un valor  $\theta_j$  asociado que se utilizará en el proceso de ajuste de los pesos. El peso asociado a la conexión desde la unidad  $i$  a la unidad  $j$  se representa por  $w_{ji}$ , y es modificado durante el proceso de aprendizaje.”





**Figura 14.** Retro Propagación  
(Basogain, 2004)

Retro propagación utiliza tres o más capas de unidades procesadoras. La capa inferior es capa de entrada se caracteriza por ser la única capa cuyas unidades de proceso reciben entradas desde el exterior. La siguiente capa es la capa oculta y todas las unidades procesadoras están interconectadas con la capa inferior y con la capa superior. La capa superior es la capa de salida que presenta la respuesta.



**Figura 15.** Retro propagación conectada completamente  
(Basogain, 2004)

#### 2.4.7.2 Algoritmo de Entrenamiento

Las redes retro propagación son un método no supervisado de entrenamiento. A la red se le presenta parejas de patrones, el patrón de entrada con su pareja un patrón de salida deseada. Por cada presentación de los pesos son ajustados para disminuir el error entre la salida deseada y la respuesta de la red.

El algoritmo de retro propagación tiene fase de propagación hacia adelante y hacia atrás, las 2 fases se realizan por cada patrón presentado en la sesión de entrenamiento.

### **2. 4.7.3 Propagación hacia Adelante**

En esta clase de propagación hacia adelante inicia cuando se presenta un patrón en la capa de entrada. Cada unidad de entrada se corresponde con un elemento del vector patrón de entrada. Estas unidades de entrada toman el valor de su correspondiente elemento del patrón de entrada y se calcula el valor de activación o nivel de salida de la primera capa. Luego las demás capas realizan la fase de propagación hacia adelante que determinará el nivel de activación de las otras capas.

“La unidad procesadora  $j$  obtiene la cantidad  $s_j$  según la siguiente ecuación.

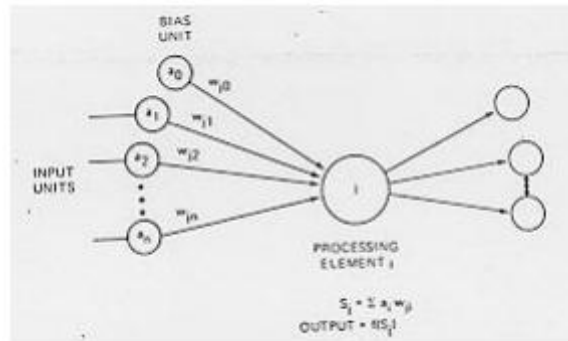
$$S_j = \sum_i a_i w_{ji} \text{ ec.}$$

y genera la salida o nivel de activación según la ecuación que se tiene a continuación.

$$\text{Salida} = f(S_j)$$

La función  $f$  es una función umbral genérica, entre las que cabe destacar la función Sigmoid y la función Hiperbólica.”

El valor de la salida de la unidad  $j$  es enviado a lo largo de todas las conexiones de salida.



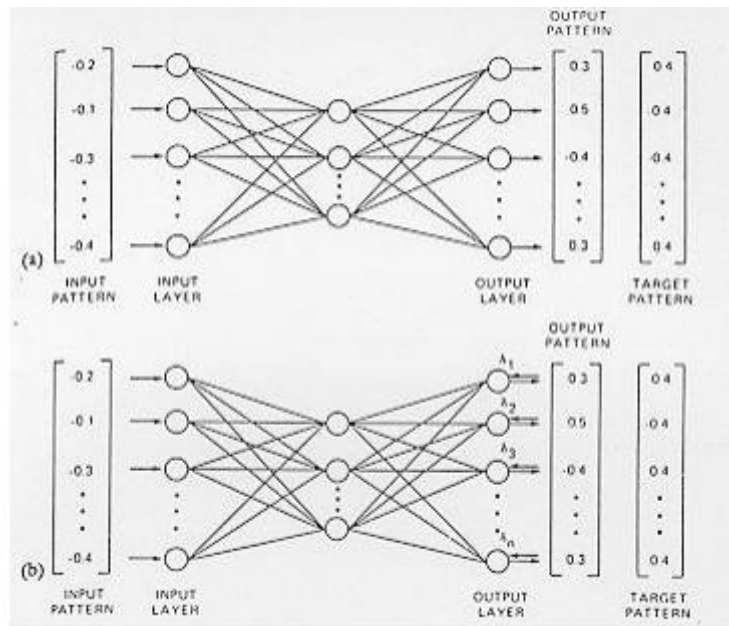
**Figura 16.** Fase de propagación hacia adelante  
(Basogain, 2004)

#### 2.4.7.4 Propagación hacia Atrás

Al completar la fase de propagación hacia adelante se inicia la fase de corrección o fase de propagación hacia atrás. Los cálculos de las modificaciones de los pesos de las conexiones empiezan por la capa de salida y continua hacia atrás a través de todas las capas hasta llegar a la capa de entrada.

#### 2.4.7.5 Ajuste de Pesos de la Capa de Salida

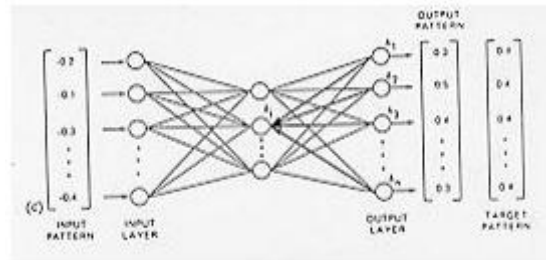
El ajuste de los pesos de la capa de salida es relativamente fácil ya que existe y se conoce el valor deseado para cada una de las unidades de la capa de salida. “Cada unidad de la capa de salida produce un número real como salida y se compara con en valor deseado especificado en el patrón del conjunto de entrenamiento.”



**Figura 17.** Ajuste de Pesos de Salida  
(Basogain, 2004)

#### 2.4.7.6 Ajuste de Pesos de las Capas Ocultas

En estas capas no se tiene vectores de salidas deseadas por lo que no se puede seguir el método de propagación.



**Figura 18.** Calculo de los valores de las capas ocultas  
(Basogain, 2004)

## 2.4.8 PERCEPTRÓN MULTICAPA

Este modelo tiene las siguientes características:

- Capa de entrada: Esta capa se encarga de recibir las señales de entrada que serán transmitidas a la siguiente capa.
- Capas ocultas: Esta capa se encarga de recibir las señales de la capa de entrada y procesarlas de una manera no lineal este set de datos y pasarla a la siguiente capa.
- Capa de salida: Esta capa se encarga de recibir los datos proporcionados por las capas ocultas y generar la respuesta de la red para cada patrón de entrada hacia el exterior.

“La propagación de los patrones de entrada en el perceptrón multicapa define una relación entre las variables de entrada y variables de salida de la red. Esta relación se obtiene propagando hacia delante los valores de entrada. Cada neurona de la red procesa la información recibida por sus entradas y produce una respuesta o activación que se propaga, a través de las conexiones correspondientes, a las neuronas de la siguiente capa. (Universidad Carlos II, Madrid, 2012)

### **Aprendizaje en el Perceptrón multicapa**

1. Se inicializan los pesos y umbrales (valores aleatorios próximos a 0).
2. Se presenta un patrón "n" de entrenamiento y se propaga hacia la salida, obteniendo la salida de la red "y(n)"
3. Se evalúa el error cuadrático, "e(n)", cometido por la red para cada patrón.
4. Se aplica la Regla Delta Generalizada para modificar pesos y umbrales:  
Se calculan los valores " " para todas las neuronas de la capa de salida.

Se calcula " " para el resto de neuronas, empezando por la última capa oculta y terminando en la capa de entrada.

Se modifican los pesos y umbrales.

5. Se repiten los pasos 2, 3 y 4 para todo el patrón de entrenamiento.

6. Se evalúa el error total de la red.

7. Se repite hasta alcanzar el error mínimo de entrenamiento, realizando "m" ciclos. Se pueden establecer otros criterios de parada:

El error del entrenamiento se estabilice.

El error de validación se estabilice.

El error de validación aumente”.

(Universidad Carlos II, Madrid,2012)

## **2.5 APLICACIÓN WEB**

Se denomina aplicación web a un sitio web que tiene páginas con contenidos específicos o a su vez sin determinar su contenido, cuando el usuario solicita una página al servidor se determina el resultado final de la página.

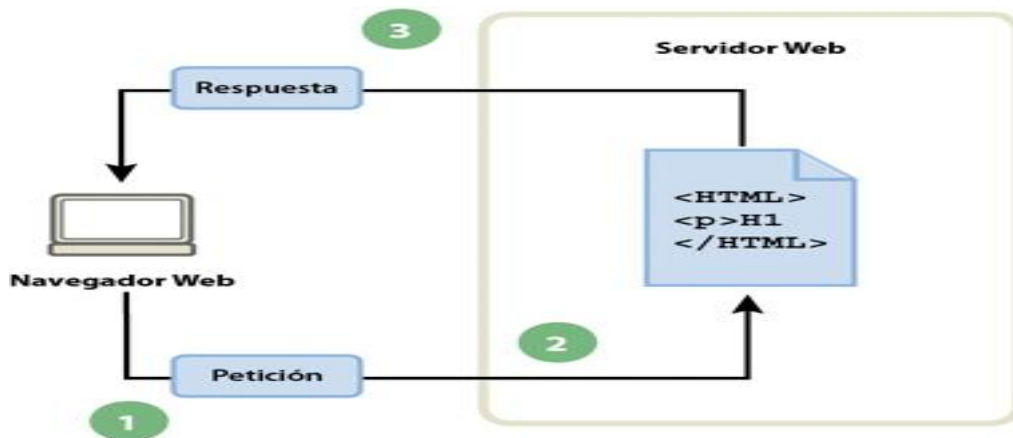
Las aplicaciones web se crean en respuesta a las necesidades que presente el usuario dependiendo de estas y si su contenido es específico o no se denomina estática o dinámica.

### **2.5.1 PROCESAMIENTO PAGINA WEB ESTATICA**

El sitio web estático consta de un conjunto de páginas y de hojas html, xhtml dependiendo de la IDE con la que se trabaja, relacionados alojados en un servidor web.

Un servidor web es un software que provee dichas páginas web en respuesta a los navegadores web que estén a la espera de la petición, la petición por una respuesta a la página web depende de lo que contenga la página web puede ser botones, links, etc. (Adobe Dreamweaver, 2009)

1. El navegador Web solicita la página estática.
2. El servidor localiza la página.
3. El servidor Web envía la página al navegador solicitante.



**Figura 19.** Figura de aplicación web

## **METODOLOGÍA**



## **3. METODOLOGÍA**

### **3.1. ALCANCE**

Realizar una aplicación orientada a usuarios que estén aprendiendo las 4 operaciones matemáticas básicas, es decir para niños de escuela, que requieran de un método de aprendizaje gráfico e interactivo con el usuario tanto en la escritura como en el análisis de la operación.

### **3.2. HERRAMIENTAS / TECNICAS**

Las herramientas técnicas que servirán de apoyo en el desarrollo del software son:

El sistema será desarrollado en lenguaje de programación Java el cual es código abierto y no requiere de licencia para su uso con IDE NetBeans, el modelo de red neuronal también se lo realizará en Java con el uso de la librería Neuroph, el cual contiene los métodos inteligentes que permitirá el ensamblaje de la parte web con la parte inteligente.

La máquina donde se realizará el sistema web funciona correctamente con las especificaciones básicas de NetBeans que es una maquina con procesador core I3 con 4GB de memoria RAM.

### **3.3. MÉTODOS**

“Método y metodología son dos conceptos diferentes. El método es el procedimiento para lograr los objetivos. Metodología es el estudio del método”.

El método que servirá de guía en este trabajo es el método hipotético-deductivo el cual consiste en que un investigador propone una hipótesis de los datos y

leyes más generales. En este método se proponen hipótesis mediante procesos inductivos y procedimientos deductivos para llegar a conclusiones particulares a partir de dichas hipótesis que se deben comprobar experimentalmente que para el desarrollo de este sistema web la experimentación es un proceso fundamental.

(Behar,2008)

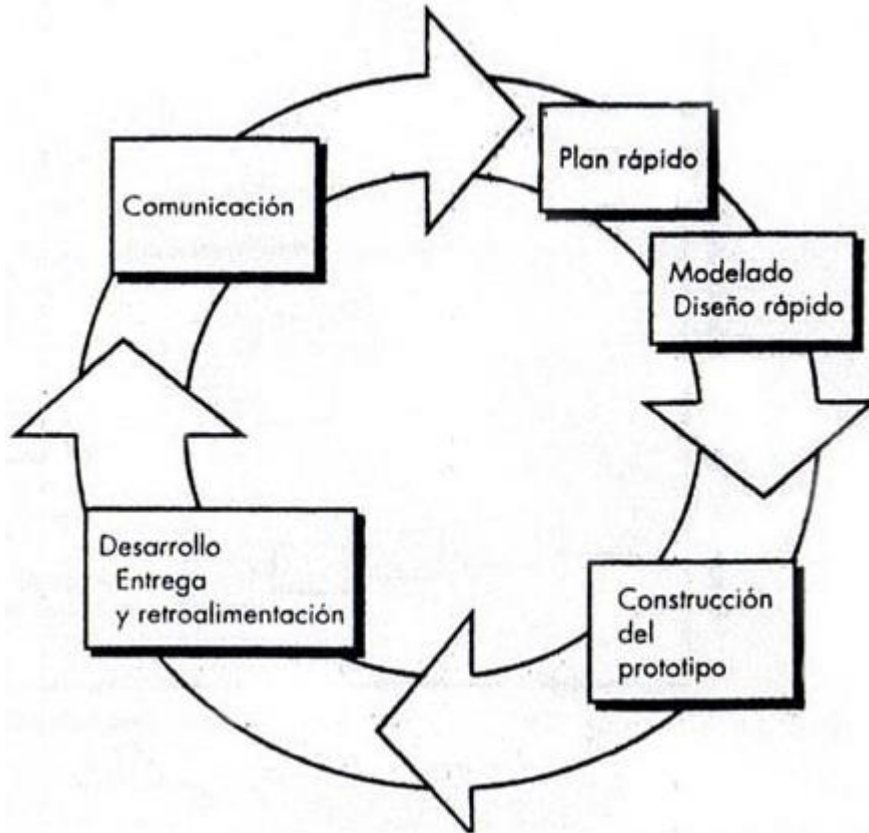
### **3.3.1 UNIDAD DE ANÁLISIS**

El proyecto está enfocado al reconocimiento de caracteres manuscritos utilizando técnicas y algoritmos de Inteligencia Artificial como redes de Kohonen, Retro Propagación, Perceptron Multicapa.

### **3.3.2 MODELO DE PROTOTIPO**

Este modelo permite que partes del sistema o el sistema en su totalidad se construya rápidamente para aclarar ciertos aspectos en los que se aseguren que todas las partes involucradas en el desarrollo de la aplicación estén de acuerdo en lo que se desea implementar así como también la solución que se propone para esta necesidad y así poder minimizar el riesgo y la incertidumbre en el desarrollo, el modelo se encarga de desarrollar diseños para que sean analizados y prescindir de ellos a medida se agreguen nuevas especificaciones. El modelo de prototipo se aplica cuando se definen un conjunto de objetivos generales para el software sin delimitar los requisitos de entrada, procesamiento y salida, es decir cuando no se está seguro de la eficacia de un algoritmo, así como también la adaptabilidad del sistema. El modelo principalmente se encarga de ayudar a entender cuál será el resultado de la construcción del sistema cuando los requisitos no estén satisfechos.

A continuación se muestran las etapas para la elaboración del Modelo de Prototipo.



**Figura 20.** Diagrama del Modelo de Prototipos  
(Pressman, 2005)

### 3.3.2.1 Comunicación

Es la comunicación inicial entre el ingeniero de software y el cliente, es aquí donde se definen los objetivos del proyecto, se definen los requisitos globales con los que contará la aplicación.

En la comunicación se deberá entender a rasgos generales lo que el usuario plantea para el diseño de la aplicación, aquí definiremos la factibilidad del proyecto, así como también los requerimientos funcionales y no funcionales.

### **3.3.2.2 Plan rápido**

El plan rápido se presentará como serán implementadas las interfaces de usuario y que contendrá cada ventana en el aspecto visual, no contendrá códigos ni algoritmos de programación ya que solo es un diseño de interfaces de como quedara la aplicación.

### **3.3.2.3 Modelado diseño rápido**

El modelado diseño rápido plantea un modelado rápido de la aplicación, así como las interfaces que tendrá el prototipo inicial.

### **3.3.2.4 Construcción del prototipo.**

La construcción del prototipo se da a partir del diseño rápido y en este se implementará las interfaces definitivas del sistema, el primer prototipo completamente funcional con su código fuente y su integración con la herramienta inteligente del proyecto.

Para el proyecto del reconocimiento de caracteres tendremos las primeras ventanas del sistema aun sin la funcionalidad de la red neuronal.

### **3.3.2.5 Desarrollo, entrega y retroalimentación**

En esta etapa se realizará toda la aplicación partiendo del primer prototipo e incorporando los algoritmos, librerías y las interfaces de usuario, para proceder a hacer la entrega del prototipo final y pasar a la retroalimentación en caso de que sea necesario. (Pressman, 2005)

## **ANÁLISIS DE RESULTADOS**

## 4 ANÁLISIS DE RESULTADOS

### 4.1 COMUNICACIÓN

La comunicación del ingeniero de software con el cliente en esta fase se verá diseñada por la factibilidad que tiene el sistema tanto técnica como de la infraestructura con la que se cuenta para la realización de este proyecto.

El reconocimiento de caracteres manuscritos en la actualidad es una herramienta de las más utilizadas no solo para dispositivos de escritorio sino también para dispositivos móviles, esta herramienta permite ayudar en el aprendizaje de la operación matemática tanto como en el del diseño del número que se va a escribir.

El objetivo del proyecto es desarrollar una aplicación web con reconocimiento de la respuesta brindada por el usuario de las 4 operaciones matemáticas básicas.

#### 4.1.1 FACTIBILIDAD TÉCNICA

Para el desarrollo de la aplicación se cuenta con los recursos tanto en software como en hardware.

Descripción	Si	No
Requisitos software Windows 7, superior	x	
Requisitos Hardware Core i3, superior	x	
Manejo java IDE NetBeans	x	
Servidor Glassfish	x	
Librería Prime Faces	x	

Librerías RNA Neuroph	x	
-----------------------	---	--

**Tabla 1.** Factibilidad técnica.

#### 4.1.2 FACTIBILIDAD ECONÓMICA

En la factibilidad económica del sistema la inversión de recursos en hardware y software no es mayor, ya que los requisitos mínimos para el desarrollo y el funcionamiento de la aplicación o son mayores y se tomara en cuenta algoritmos de programación para mejorar el rendimiento del sistema.

Rubro	US\$	Fuente financiamiento
Equipo core i3	500	Donación
Licencia Windows 8	160	Donación
Total:	660	

**Tabla 2.** Factibilidad Económica

#### 4.1.3 FACTIBILIDAD OPERATIVA

El sistema al ser diseñado para niños es fácil de manejar ya que tiene la interfaz muy sobria, sin sobrecarga de imágenes lo que hace que no sea difícil su manejo.

#### 4.1.4 REQUERIMIENTOS NO FUNCIONALES

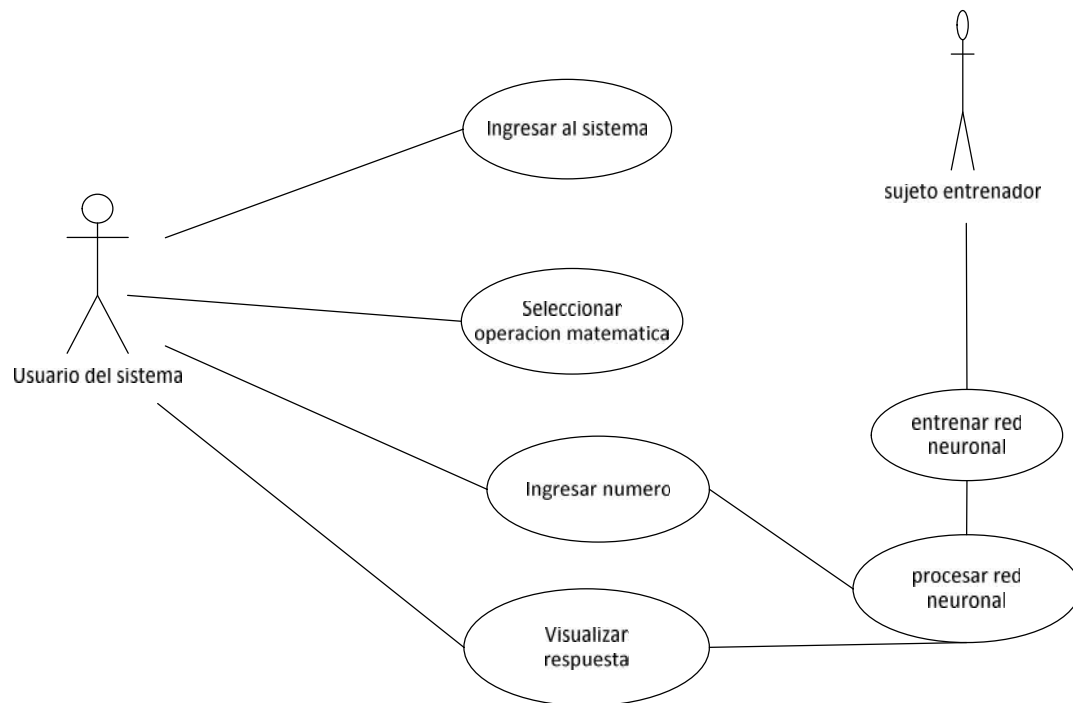
Los requerimientos fundamentales de la aplicación son:

- Ser una herramienta web.
- Estar alojada en cualquier equipo que pueda funcionar como servidor con los requisitos de hardware mencionados anteriormente.
- Interfaces de usuario simples, de fácil manipulación.

#### 4.1.5 REQUERIMIENTOS FUNCIONALES

##### CASOS DE USO DEL SISTEMA

En la figura se muestra el caso de uso del proceso general del sistema.



**Figura 21.** Caso de uso general del sistema

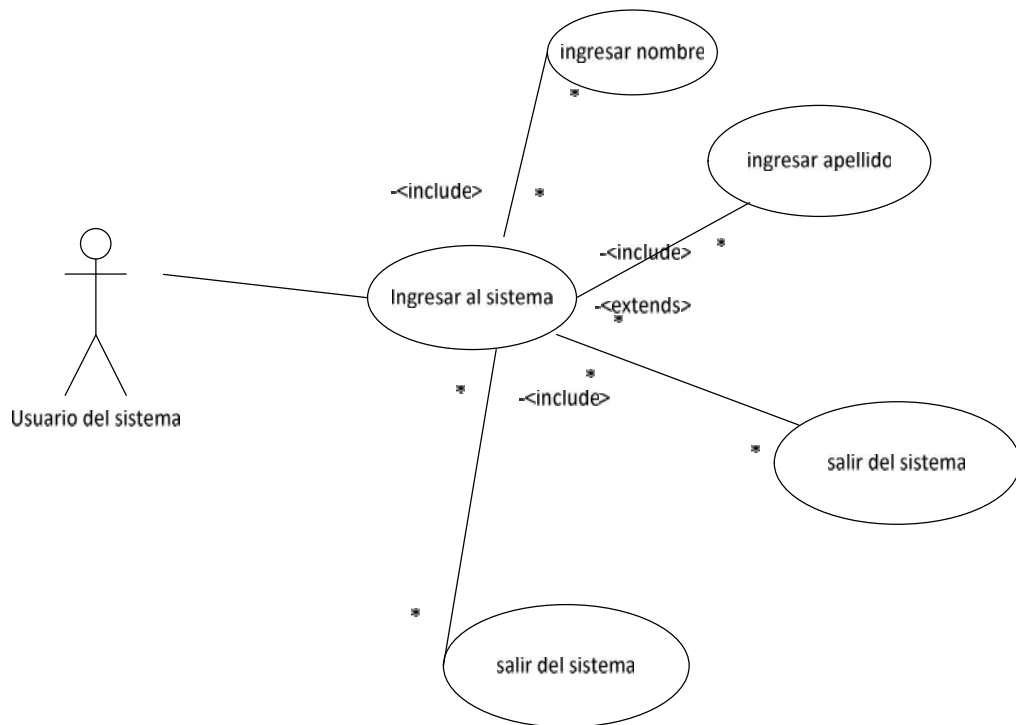


El sistema usa librería de redes neuronales artificiales Neuroph que contiene los métodos de Perceptrón Multicapa y Retro Propagación, dentro del método del Perceptrón Multicapa se encuentra la función de la base del conocimiento, esto trata de un archivo donde se tiene los números ya entrenados, el usuario del sistema no debe entrenar esta base de conocimiento, el sistema procesará el dato de entrada ingresado por el usuario del sistema en este caso el número y proveerá una respuesta.

La base del conocimiento de la red neuronal artificial es previamente entrenada por sujetos experimentales.

#### **4.1.5.1 Caso de uso ingresar al sistema**

En esta figura se muestra el proceso de ingreso al sistema, una vez que se cumpla con este proceso se puede ingresar al menú principal donde se encuentran las operaciones matemáticas básicas.



**Figura 22. Ingresar al sistema**

**Descripción Caso de Uso Ingresar al Sistema**

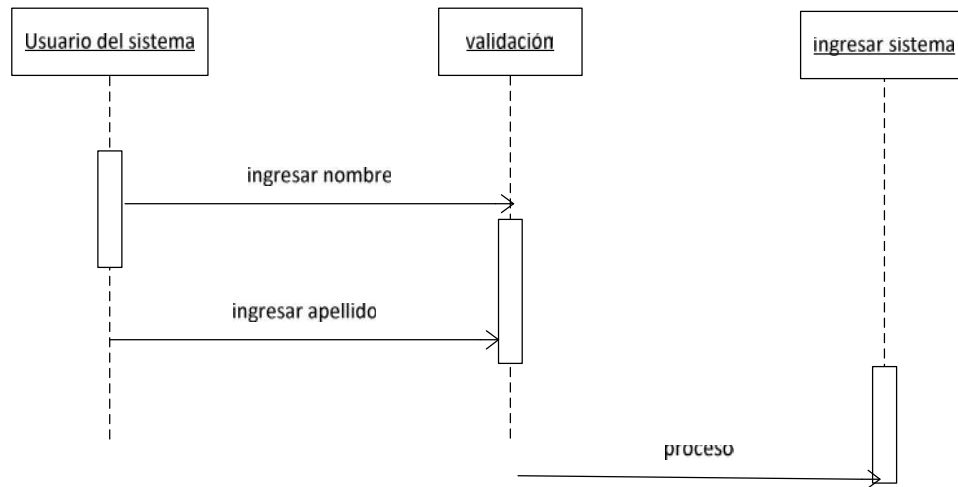
A continuación se detalla el caso de uso ingresar al sistema

<p><b>Nombre del caso de uso</b></p>	<p>Ingresar al sistema</p>
<p><b>Descripción</b></p>	<p>Para ingresar al sistema es requerido el ingreso de nombre, apellido y pulsar el botón ingresar</p>

<b>Actor</b>	Usuario del sistema	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor</b>	<b>Sistema</b>
<b>1</b>	Ingresar nombre, el usuario debe ingresar nombre	
<b>2</b>	Ingresar apellido, el usuario debe ingresar su apellido	
<b>3</b>	Pulsar botón ingresar	Ingresar al menú principal
<b>Flujo alterno</b>		
<b>1</b>	Salir	Salir del sistema
<b>Pre Condición</b>		
<b>Post Condición</b>		

**Tabla 3.** Ingresar al sistema

#### 4.1.5.2 Diagrama secuencial Ingresar al Sistema

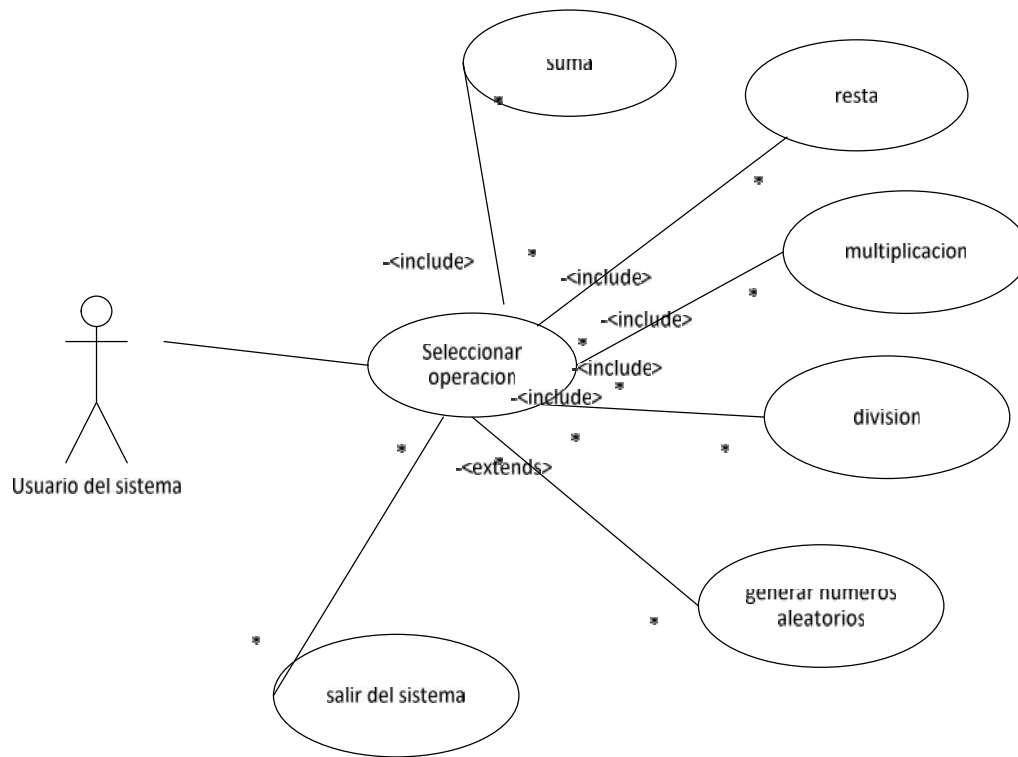


**Figura 23.** Diagrama secuencial Ingresar al sistema

En la figura se muestra el diagrama secuencial de ingresar al sistema, donde se muestra la forma en la que el sistema permite el ingreso al menú principal donde constan las 4 operaciones fundamentales.

#### 4.3.5.3 Caso de uso Seleccionar Operación Matemática.

En esta figura se muestra el proceso de selección de la operación matemática.



**Figura 24.** Caso de uso selección operación matemática

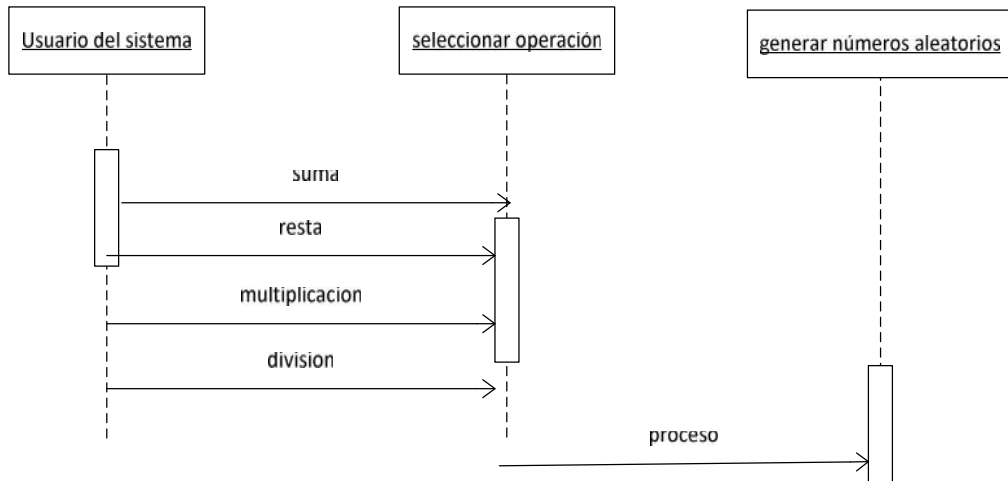
**Descripción caso de uso seleccionar operación matemática.**

<p><b>Nombre del caso de uso</b></p>	<p>Selección operación matemática</p>
<p><b>Descripción</b></p>	<p>Selección el operador matemático, desplegar en el sistema el nivel de la operación, si es simple 1 cuadro de texto si es doble 2.</p>

<b>Actor</b>	Usuario del sistema	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor</b>	<b>Sistema</b>
1	suma	Ingresa a la interfaz suma
2	resta	Ingresa a la interfaz resta
3	multiplicación	Ingresa a la interfaz multiplicación
4	división	Ingresa a la interfaz división
5	Generar números aleatorios	Genera números aleatorios de acuerdo al nivel.
<b>Flujo Alternativo</b>		
1	Salir	Salir del sistema
<b>Pre Condición</b>		
<b>Post Condición</b>		

**Tabla 4.** Selección operación matemática

#### 4.1.5.4 Diagrama secuencial Selección operación matemática

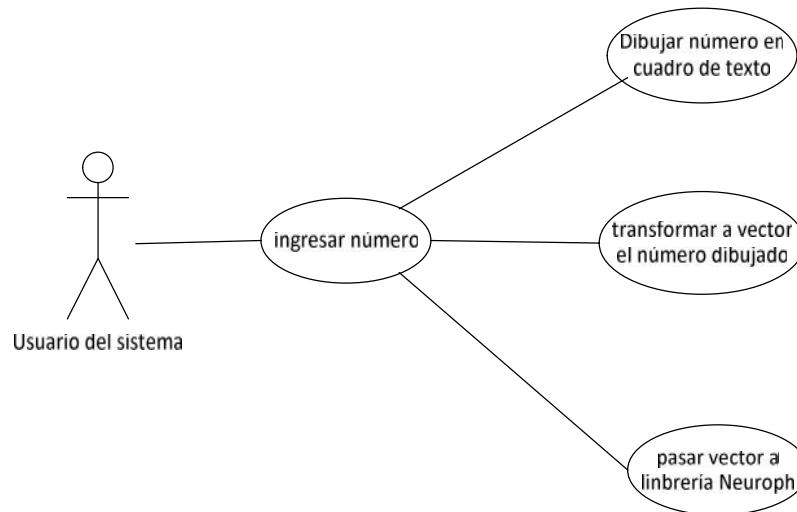


**Figura 25.** Diagrama secuencial selección operación matemática.

En la figura se muestra el diagrama secuencial de la selección de la operación matemática, en este paso se selecciona la operación matemática básica y se define también el nivel de la operación si va a ser básica activa solo un cuadro de texto para ser reconocido por la RNA y si es de nivel medio activa 2 cuadros de texto para ser reconocidos por proceso RNA.

#### 4.1.5.5 Caso de uso Ingresar número.

En esta figura se muestra el dibujo del número el cual se va a reconocer por la RNA.



**Figura 26.** Ingresar número

#### Descripción caso de uso dibujar número.

<b>Nombre del caso de uso</b>	Ingresar número
<b>Descripción</b>	Dibujar número que será reconocido por la RNA



<b>Actor</b>	Usuario del sistema	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor</b>	<b>Sistema</b>
<b>1</b>	Dibujar número en cuadro de texto	Recuperar los pixeles por donde se dibujó
<b>2</b>	Transformar a vector el número	Procesar los pixeles y transformarlos a vector
<b>3</b>	Pasar vector a librería	Alimentar la librería (vector)
<b>Pre Condición</b>	Entrenar red neuronal	
<b>Post Condición</b>		

**Tabla 5.**Ingresar número

#### 4.1.5.6 Diagrama secuencial ingresar número

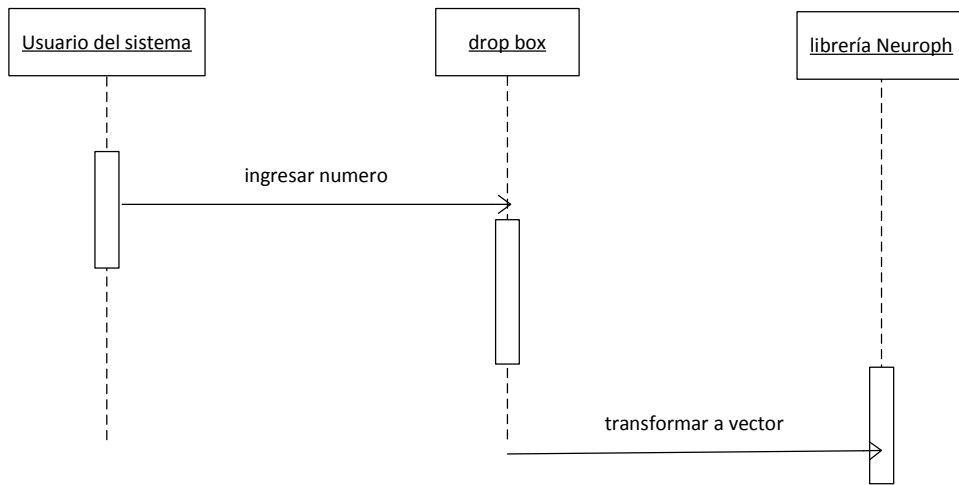


Figura 27. Diagrama ingresar número

En la figura se muestra el proceso en el cual se dibuja el número se lo transforma a vector y se pasa a la red neuronal para que sea procesado.

#### 4.1.5.7 Caso de uso procesar red neuronal

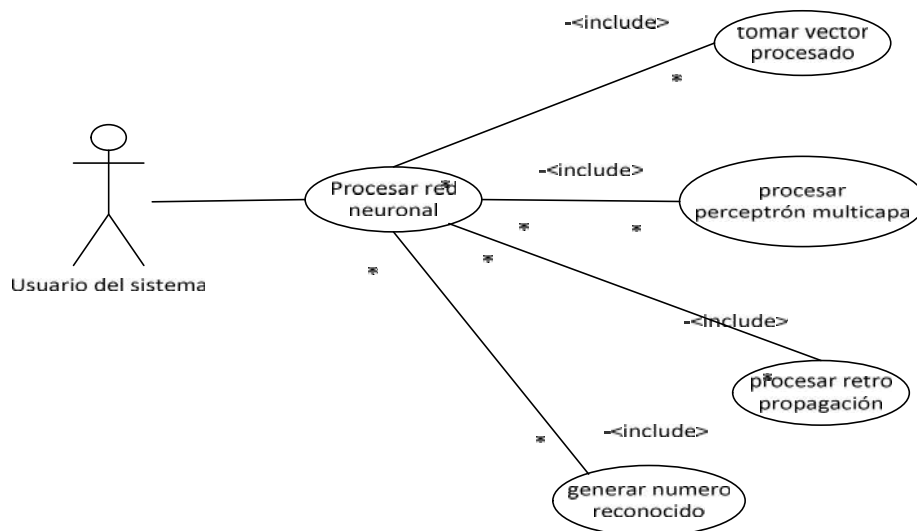


Figura 28. Procesar red neuronal

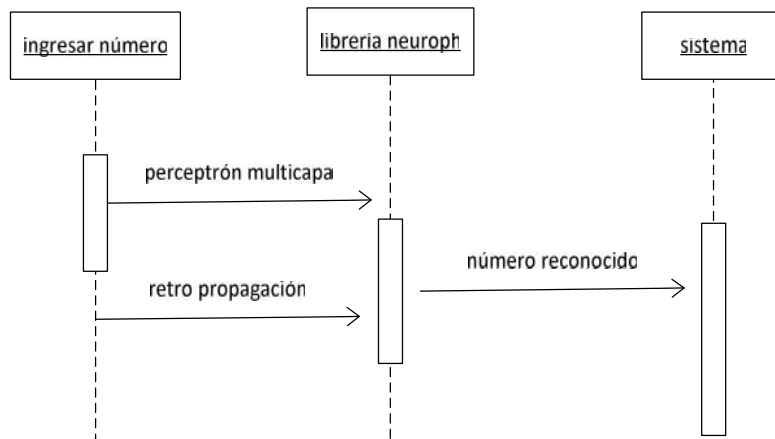
## Descripción caso uso Procesar Red Neuronal

<b>Nombre del caso de uso</b>	Procesar Red Neuronal	
<b>Descripción</b>	Reconocer el número utilizando librería Neuroph	
<b>Actor</b>	Usuario del sistema	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor</b>	<b>Sistema</b>
<b>1</b>	Tomar vector procesado	Selecciona el vector ya procesado
<b>2</b>	Procesar perceptrón multicapa	Proceso de librería
<b>3</b>	Procesar retro propagación	Proceso de librería

4	Generar número reconocido	Termino de proceso de librería y genera número
<b>Pre Condición</b>		
<b>Post Condición</b>		

**Tabla 6.** Procesar red neuronal.

#### 4.1.5.8 Diagrama secuencial procesar red neuronal.

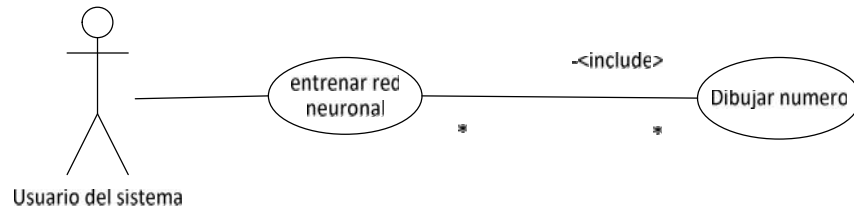


**Figura 29.** Diagrama de secuencia del procesar red neuronal.

En la figura se muestra el proceso en el cual se procesa el número, en la fase anterior se desplegó un vector que contenía ceros y unos, este vector alimenta a la red neuronal que y esta con procesos de librería y utilizando técnicas como

retro propagación y Perceptron multicapa reconoce el parámetro ingresado por el usuario y lo reconoce de acuerdo a lo que tenga en la base del conocimiento.

#### 4.1.5.9 Caso de uso respuesta



**Figura 30.** Reconocimiento de número

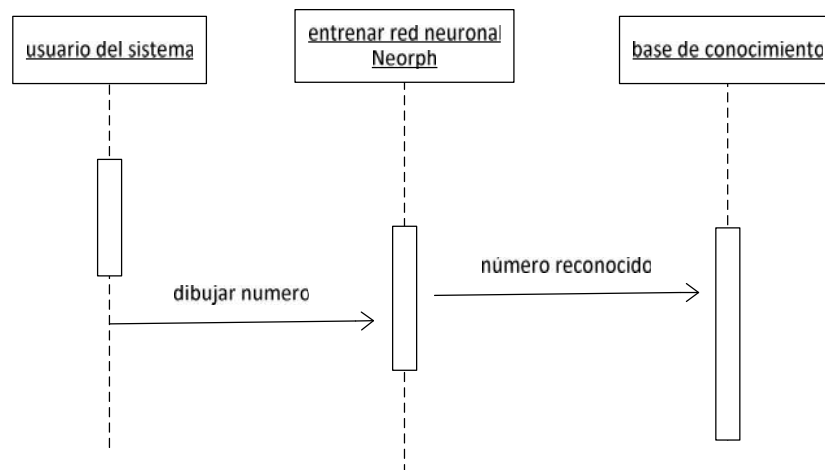
#### Descripción caso uso respuesta.

<b>Nombre del caso de uso</b>	Entrenar red neuronal
<b>Descripción</b>	Entrenar la red neuronal dibujando números que crean la base del conocimiento
<b>Actor</b>	Usuario del sistema
<b>Flujo Básico</b>	

Paso	Actor	Sistema
1	Dibujar numero	Almacenar en base de conocimiento
Pre Condición		
Post Condición		

**Tabla 7.** Entrenar red neuronal

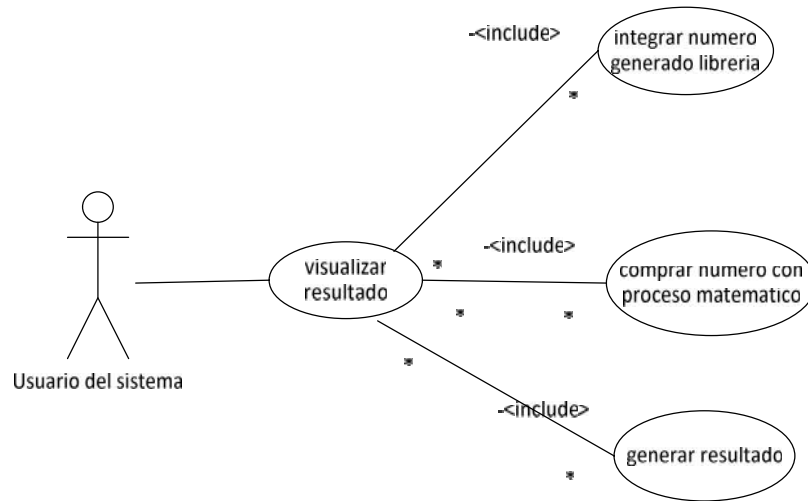
#### 4.1.5.10 Diagrama secuencial entrenar red neuronal



**Figura 31.** Diagrama de secuencia entrenar red neuronal.

En la figura se detalla el proceso de entrenamiento de la red la cual está dada por los sujetos de entrenamiento que son los sujetos que dibujan los números para que la red sea alimentada por estos patrones

#### 4.1.5.11 Caso de uso visualizar resultado



**Figura 32.** Visualizar resultado

#### Descripción caso uso Visualizar Resultado

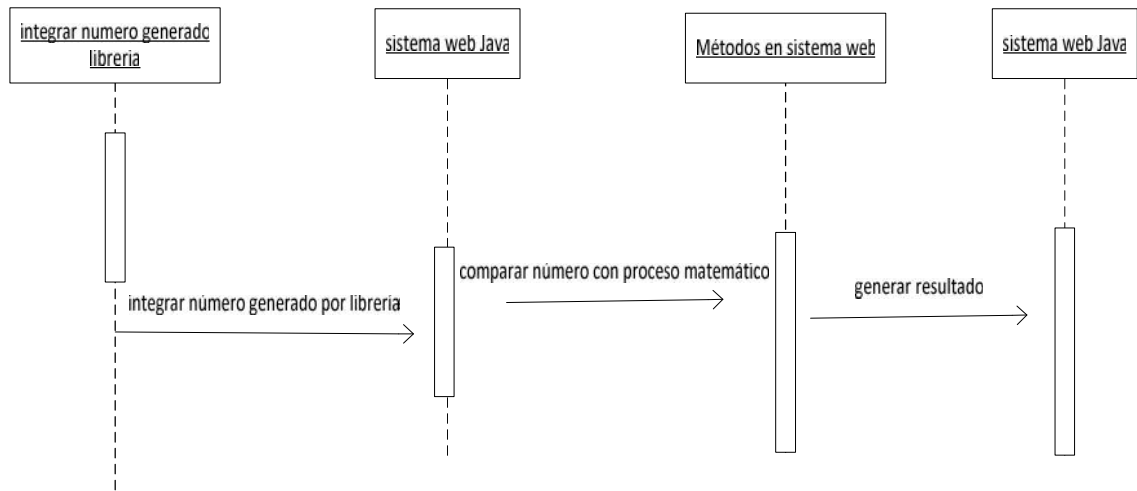
<p><b>Nombre del caso de uso</b></p>	<p>Visualizar resultado</p>
<p><b>Descripción</b></p>	<p>Conocer el resultado dado por los procesos tanto matemáticos como inteligentes</p>

<b>Actor</b>	Usuario del sistema	
<b>Flujo Básico</b>		
<b>Paso</b>	<b>Actor</b>	<b>Sistema</b>
<b>1</b>	Integrar número generado librería	Combinar procesos inteligentes y web
<b>2</b>	Comparar número con proceso matemático	Verificar la respuesta
<b>3</b>	Generar resultado	Mostrar el número
<b>Pre Condición</b>		
<b>Post Condición</b>		

**Tabla 8.** Visualizar resultado



#### 4.1.5.12 Diagrama secuencial visualizar resultado

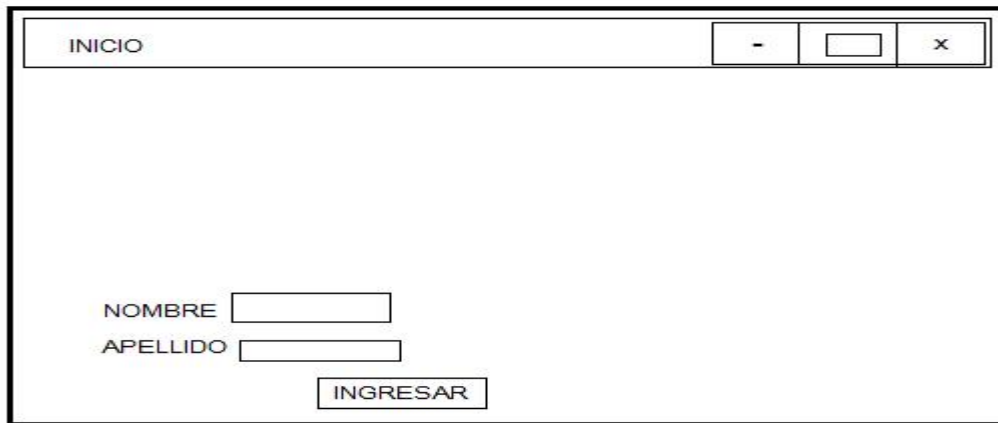


**Figura 33.** Diagrama de secuencia entrenar red neuronal.

## 4.2 PLAN RÁPIDO

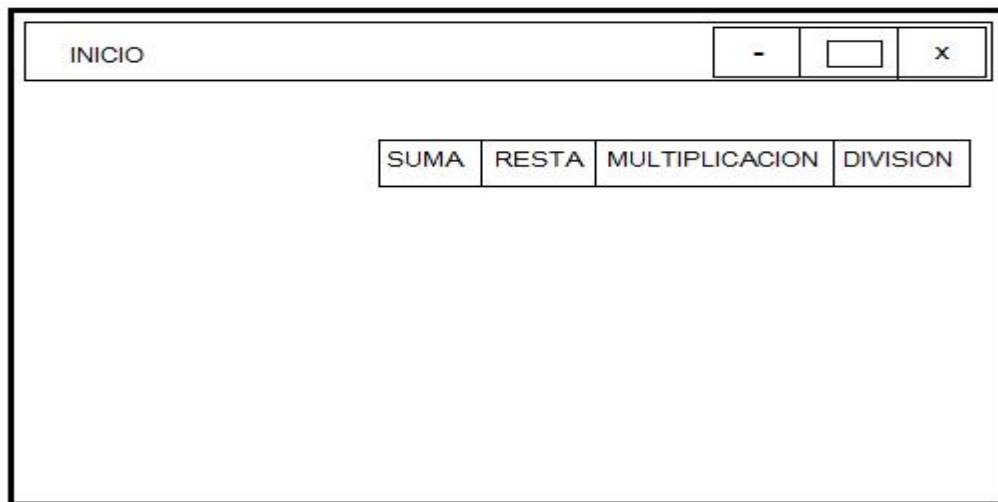
Para el diseño del plan rápido se tendrá los bosquejos de la interfaces que intervendrán en el primer prototipo de la aplicación.

Interfaz de inicio, en esta interfaz se tiene el ingreso al sistema donde se solicita el nombre, apellido y pulsar el botón ingresar para pasar a la siguiente interfaz.



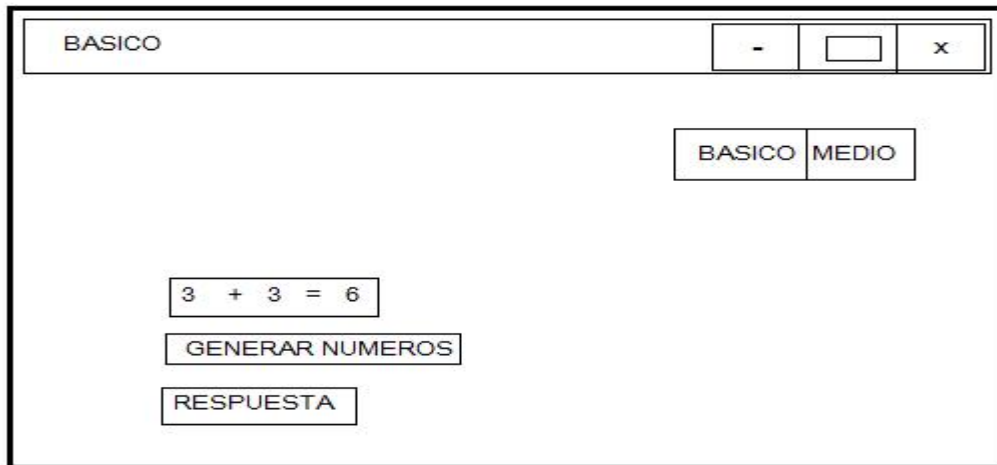
**Figura 34.** Diagrama Interfaz Inicio

La siguiente interfaz contendrá las operaciones matemáticas



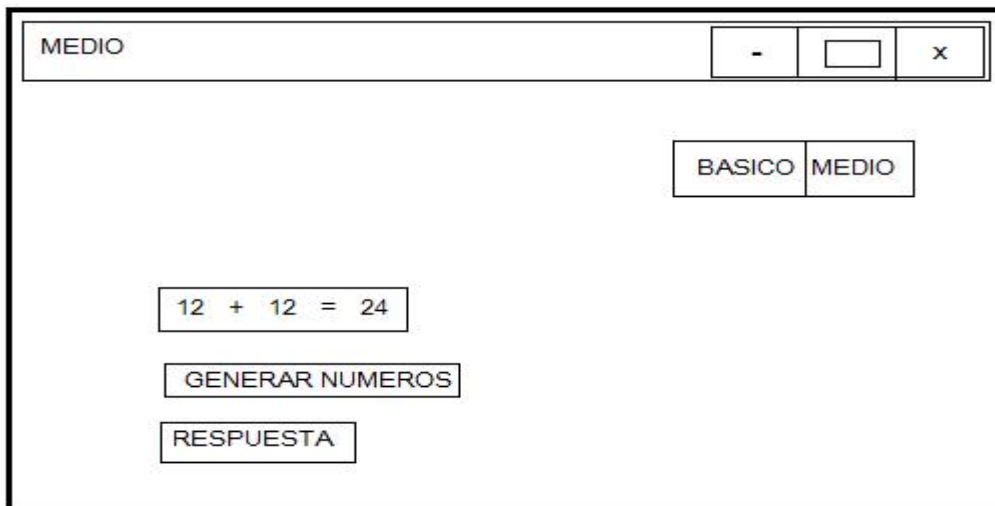
**Figura 35.** Diagrama Interfaz menú principal

La interfaz de suma básica tendrá la generación de números aleatorios, el casillero para ingreso de respuesta y el botón respuesta que es el botón que procesa la información ingresada y genera el resultado en la interfaz



**Figura 36.** Diagrama Interfaz suma básica

La interfaz suma medio contada 2 casilleros para ingreso de respuesta, el boton de generar números aleatorios y el boton respuesta que genera la respuesta de la operación matemática.



**Figura 37.** Diagrama Interfaz suma media

### 4.3 MODELADO DISEÑO RÁPIDO

Para el modelado diseño de la aplicación se presentan las interfaces de la aplicación y el avance en la construcción de la interface.



En la figura se muestra el modelado de la interfaz suma básica sin botones de ejecución.



En la figura se muestra el modelado de la interfaz suma medio sin los casilleros donde se ubica la respuesta.

## 4.4 CONSTRUCCIÓN DEL PROTOTIPO

Para la construcción general del prototipo se utilizará casos de uso, que definirán como trabajará el sistema con su interfaz.

Para el diseño rápido tenemos las interfaces que contendrá el sistema las cuales se detallan a continuación.

Interfaz de ingreso donde el usuario va a ingresar al sistema.



**Figura 38.** Ingreso al menú principal

2.- La siguiente Interfaz se tiene el menú principal donde podemos seleccionar las operaciones matemáticas básicas.



**Figura 39.** Interfaz del Menú Principal

3.- La última Interfaz del prototipo inicial es la de la operación suma que contiene los números aleatorios y la casilla de respuesta.

Este primer prototipo es el más extenso en cuanto a la cantidad de interfaces ya que los demás prototipos solo se refieren a las operaciones matemáticas en sus 2 niveles.

El nivel básico en el cual solo se tiene un cuadro de texto donde se ingresará un solo número y el nivel medio en el cual se tiene 2 cuadros de texto para tener 2 números en su respuesta.



**Figura 40.** Nivel suma básico



**Figura 41.** Nivel suma medio

## **4.5 DESARROLLO, ENTREGA Y RETROALIMENTACIÓN**

Luego de ya tener el diseño rápido y los diagramas sobre cómo va a funcionar la aplicación se procede al desarrollo de la misma.

### **4.5.2 DESARROLLO DE LAS INTERFACES DE USUARIO Y CÓDIGO FUENTE DE LA APLICACIÓN**

#### **4.5.2 .1 Entrenamiento**

El set de entrenamiento se compone por las entradas que se le da al sistema, es decir los números los cuales servirán de patrón.

#### 4.5.2 .2 Set de Entrenamiento

El set de entrenamiento es un archivo de texto, donde contiene los patrones dibujados de los números y la cantidad de veces que este fue entrenado, este set contiene la figura de números almacenados en vectores de ceros y unos.

Los ceros y unos representan en la matriz donde se escribe el número si es un pixel por donde se dibujó en la matriz recibe el número 1 si no está pintado recibe el número 0.

La activación de las neuronas de salida viene dado por el siguiente patrón:

0 = 1,0,0,0,0,0,0,0,0,0

1 = 0,1,0,0,0,0,0,0,0,0

2 = 0,0,1,0,0,0,0,0,0,0

3 = 0,0,0,1,0,0,0,0,0,0

4 = 0,0,0,0,1,0,0,0,0,0

5 = 0,0,0,0,0,1,0,0,0,0

6 = 0,0,0,0,0,0,1,0,0,0

7 = 0,0,0,0,0,0,0,1,0,0

8 = 0,0,0,0,0,0,0,0,1,0

9 = 0,0,0,0,0,0,0,0,0,1



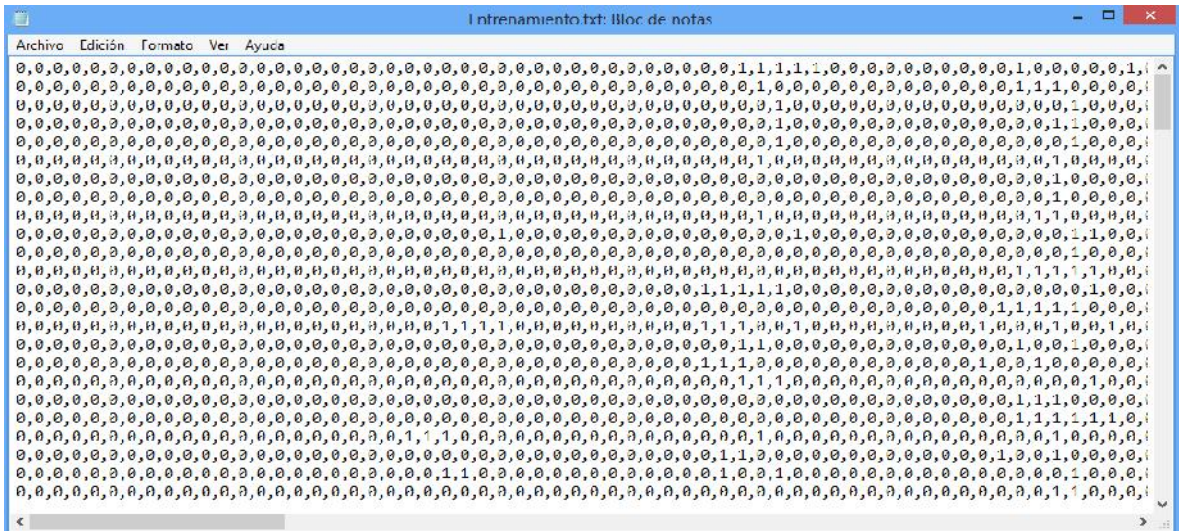


Figura 42. Hoja de Texto del Reconocimiento

#### 4.5.2.3 Números en el set de entrenamiento

Los números que se dibujan en el cuadro de texto son representados en vector de la siguiente forma:

0000000000000000	0000000000000000	0000000000000000	0000000000000000
0000000000000000	0000000000000000	0000000000000000	0000000000000000
0000000111110000	0000000010000000	0000000000000000	0001111110000000
0000001000001000	0000001110000000	0000001111100000	0000000001100000
0000010000000100	0000001010000000	0000010000100000	0000000000100000
0000110000000010	0000000010000000	0000000000100000	0000000000100000
0000110000000010	0000000010000000	0000000000100000	0000011011000000
0000100000000010	0000000010000000	0000000000100000	0000011111000000
0000100000000010	0000000010000000	0000000000100000	0000000000100000
0000100000000010	0000000010000000	0000000000100000	0000000000100000
0000100000000010	0000000010000000	0000000000100000	0000000000100000
0000100000000010	0000000010000000	0000000000100000	0000000000100000
0000100000000010	0000000010000000	0000000000100000	0000000000100000
0000100000000010	0000000010000000	0000000000100000	0000000000100000
0000110000001000	0000000010000000	0000000011000000	0000000000010000
0000000111110000	0000000010000000	0000011000000000	0000000000010000
0000000000000000	0000000000000000	0000100000000100	0011111111110000
0000000000000000	0000000000000000	0000011111110000	0000000000000000

0000000000000000	0000000000000000	0000000000000000	0000000000000000
0000000000000000	0000000000000000	0000000000000000	0000000000000000
0000000000000000	0000000000000000	0000000000000000	0000000000000000
00000000000001000	0000111111110000	0000000100000000	0000000000000000
000000000000011000	0000100000000000	0000000100000000	0000011111110000
0000000000000101000	0000100000000000	0000000100000000	00000000000010000
00000000000001001000	0000100000000000	0000000100000000	00000000000010000
0000000000000110001000	0000100000000000	0000000100000000	00000000000010000
0000000000000100001000	0000011111110000	000000010011110000	00000000000010000
0000011111111100	00000000000010000	000000010100001000	0000011111111111
000000000000001000	00000000000010000	000000010100001000	00000000000010000
0000000000000001000	00000000000010000	000000010000001000	00000000000010000
00000000000000001000	00000000000010000	000000010000001000	00000000000010000
000000000000000001000	00000000000010000	000000010000001000	00000000000010000
0000000000000000001000	00010000000010000	000000011000001000	00000000000010000
00000000000000000001000	00000110000110000	0000000111110000	00000000000010000
0000000000000000000000	00000001111000000	00000000000000000	00000000000010000

0000000000000000	0000000000000000
0000011100000000	0000000000000000
0000010011100000	0000000110000000
0000110000010000	0000000111110000
0000100000010000	00000001000001000
0000100000010000	000000010000001000
0000010000010000	00000001000001000
0000001110010000	0000000110011000
0000011101110000	0000000001101000
0000010000001000	0000000000001000
0000100000001000	0000000000001000
0000100000001000	00000000000010000
0000100000001000	00000000000010000
0000010000011000	0000000000100000
000000111100000	0000000011000000
0000000000000000	0000000100000000

**Figura 43.** Números en Fase de Entrenamiento

En la figura se muestra la página de inicio donde se accede a la aplicación.

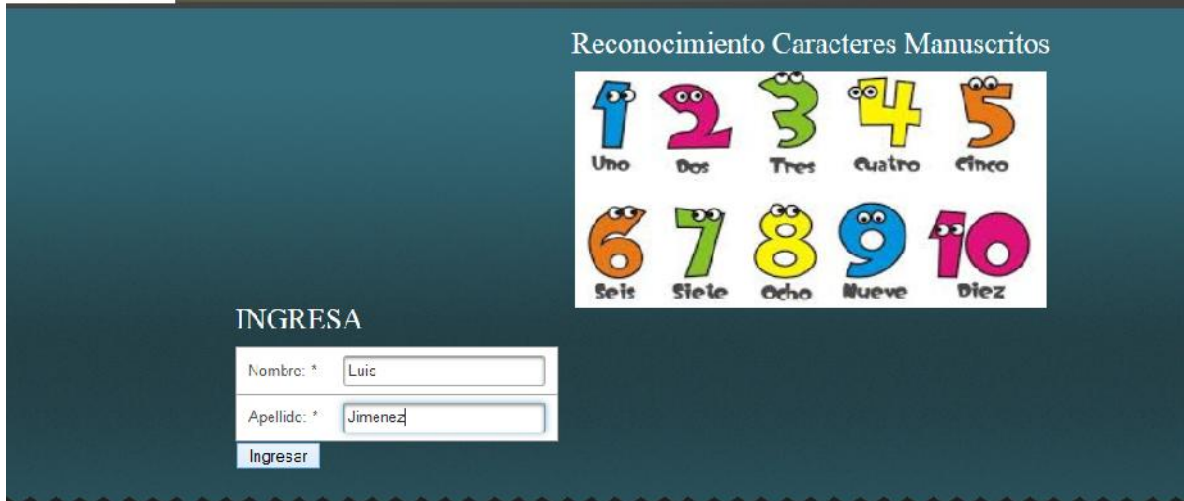


Figura 44. Página de inicio

### Botón Ingresar código fuente

El botón ingresar al ser presionado re direcciona al menú principal tomando en cuenta las validaciones en el ingreso de nombre y apellido

```
public String ingresar() {  
    return "menu.xhtml?faces-redirect=true";  
}
```

## Menú de la aplicación

El menú de la aplicación contiene las operaciones fundamentales.



**Figura 45.** Menú principal de la aplicación

## Interfaz

Las cuatro operaciones fundamentales se encuentran en las pestañas del menú principal, estas abren un xhtml que contiene la página donde se encuentran los números y el cuadro de texto donde se ingresa el número

### Interfaz Suma Básico

En la pestaña de suma básica se tiene los números aleatorios y su botón generar números, el cuadro de texto donde se va a ingresar la respuesta y el botón respuesta.

UNIVERSIDAD  
TECNOLOGICA EQUINOCCIAL

Basico Medio

SUMA NIVEL BASICO

• REGRESAR

$0 + 7 = 7$

generar numeros

respuesta

Respuesta Correcta 7

1 2 3  
4 5 6  
7 8 9  
0

Figura 46. Suma básica con la respuesta correcta



**Figura 47.** Suma básica con la respuesta incorrecta

### Botón generar números código fuente

Código fuente en xhtml

```
<p:commandButton value="generar numeros" action="#{control.aleSumBasi()}" ajax="false" />
```

Código fuente en la clase control

```
public void aleSumBasi() {
    double c, d;

    do {
        c = Math.random() * 10;
        d = Math.random() * 10;
        num1 = (int) c;
        num2 = (int) d;

    }while ( num1+num2 >9 );
}
```

## Botón respuesta código fuente

Código fuente en xhtml

```
<p:commandButton value="respuesta" action="#{control.suma()}" ajax="false" />
```

Código fuente clase Control

```
public void suma() throws FileNotFoundException, IOException {  
  
    String trainingSetFileName = "C:\\Test\\Entrenamiento.txt";  
  
    rna.MPerceptron1 uno = new MPerceptron1();  
  
    int inputsCount = 256;  
    int outputsCount = 10;  
  
    // crear data set  
    DataSet trainingSet = null;  
  
    try {  
        trainingSet = TrainingSetImport.importFromFile(trainingSetFileName, inputsCount, outputsCount, ",");  
    } catch (IOException ex) {  
        Logger.getLogger(MPerceptron1.class.getName()).log(Level.SEVERE, null, ex);  
    } catch (NumberFormatException ex) {
```

```

// crear multi layer perceptron
MultiLayerPerceptron neuralNet = new MultiLayerPerceptron(TransferFunctionType.SIGMOID, 256, 20, 10);

// set parametros
MomentumBackpropagation learningRule = (MomentumBackpropagation) neuralNet.getLearningRule();
learningRule.setLearningRate(0.2);
learningRule.setMomentum(0.7);

// aprender el set de entrenamiento
neuralNet.learn(trainingSet);
System.out.println("Listo!");

// test perceptron
System.out.println("Probando la RNA");
int a=uno.testFaceRecognition(neuralNet, trainingSet);
FileWriter fw = new FileWriter("C:\\Test\\Vector.txt");
fw.write("" + a );
fw.flush();
fw.close();
//uno.testFaceRecognition(neuralNet, trainingSet);

```

```

    }
} catch (IOException ex) {
    Logger.getLogger(Control.class.getName()).log(Level.SEVERE, null, ex);
}
respuesta = Integer.parseInt(rp);
if (respuesta == res) {

    FacesMessage mensaje = new FacesMessage(FacesMessage.SEVERITY_INFO,
        "Respuesta Correcta " + res, null);
    FacesContext.getCurrentInstance().addMessage("frmPrincipa", mensaje);

} else {

    FacesMessage mensaje = new FacesMessage(FacesMessage.SEVERITY_INFO,
        "Respuesta Incorrecta "+ respuesta +" ,Respuesta Correcta es " + res, null);
    FacesContext.getCurrentInstance().addMessage("frmPrincipa", mensaje);

}

```



## Interfaz Suma Medio

En esta pestaña se activa 2 cuadros de texto para ingresar la respuesta.



Figura 48. Suma Medio

### Botón generar números código fuente

Código fuente en html

```
<p:commandButton value="generar numeros" action="#{control.aleSumMed()}" ajax="false" />
```

Código Fuente en la clase control

```

////// NIVEL MEDIO SUMA
public void aleSumMed() {
    double c, d;

    do {
        c = Math.random() * 100;
        d = Math.random() * 100;

        num1 = (int) c;
        num2 = (int) d;

    }while ( (num1+num2>99) || (num1+num2<10));
}

```

## Botón respuesta código fuente

Código fuente en xhtml

```

<p:commandButton value="respuesta" action="#{control.sumaM()}" ajax="false" />

```

Código fuente clase control

```

int a=uno.testFaceRecognition(neuralNet, trainingSet);
FileWriter fw = new FileWriter("C:\\\\Test\\\\num1.txt");
fw.write("" + a );
fw.flush();
fw.close();

int b=dos.testFaceRecognition(neuralNet, trainingSet);
FileWriter fw1 = new FileWriter("C:\\\\Test\\\\num.txt");
fw1.write("" + b );
fw1.flush();
fw1.close();

String rp = "", rp1="";

```

El primer prototipo del sistema llega hasta el módulo de suma básica y suma medio, los otros prototipos llevarán los algoritmos ya programados en este prototipo para optimizar el tiempo de desarrollo y los resultados.

### Interfaz resta básica



Figura 49. Resta Básica

## Botón respuesta código fuente

```
// test del perceptron

int a=testFaceRecognition(neuralNet, trainingSet);
FileWriter fw = new FileWriter("C:\\\\Test\\\\num1.txt");
fw.write("" + a );
fw.flush();
fw.close();

int res;
String rp = "";
res = num1 - num2;

String sCurrentLine;
BufferedReader br = null;
br = new BufferedReader(new FileReader("C:\\\\Test\\\\num1.txt"));

try {
    while ((sCurrentLine = br.readLine()) != null) {
        rp = sCurrentLine;
    }
}
```

## Interfaz resta medio

The screenshot displays the user interface for a subtraction exercise. At the top left is the logo of the Universidad Tecnológica Equinoccial (UTE). To its right, the text 'UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL' is visible. On the far right of the header are two buttons: 'Basico' and 'Medio'. Below the header, the text 'RESTA NIVEL INTERMEDIO' is centered, followed by a link '• REGRESAR'. The main area features a subtraction problem: '0 - 0 = ' with two empty rectangular boxes for the answer. Below this problem are two buttons: 'generar numeros' and 'respuesta'.

Figura 50. Resta Medio

## Botón respuesta código fuente

```
respuesta = Integer.parseInt(rp);
respuesta2 = Integer.parseInt(rp1);

numero1 = String.valueOf(respuesta);
numero2 = String.valueOf(respuesta2);

concatenado = numero1 + numero2;
res1 = Integer.parseInt(concatenado);

if (res1 == res) {

    FacesMessage mensaje = new FacesMessage(FacesMessage.SEVERITY_INFO,
        "Respuesta Correcta " + res1, null);
    FacesContext.getCurrentInstance().addMessage("frmPrincipa", mensaje);

} else {

    FacesMessage mensaje = new FacesMessage(FacesMessage.SEVERITY_INFO,
        "Respuesta Incorrecta "+ res1 + " ,Respuesta Correcta es " + res, null);
    FacesContext.getCurrentInstance().addMessage("frmPrincipa", mensaje);

}
```

## Pestaña multiplicación nivel básica

UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL  
UTE

UNIVERSIDAD  
TECNOLÓGICA EQUINOCCIAL

Basico Medio

MULTIPLICACION NIVEL BASICO

• REGRESAR

2 x 4 = 8

generar numeros

respuesta

Respuesta Correcta 8

Figura 51. Multiplicación nivel Básica

## Botón respuesta código fuente

```
String rp = "";  
res = num1 * num2;  
  
String sCurrentLine;  
BufferedReader br = null;  
br = new BufferedReader(new FileReader("C:\\Test\\num1.txt"));  
  
try {  
    while ((sCurrentLine = br.readLine()) != null) {  
        rp = sCurrentLine;  
        // System.out.println(rp);  
    }  
} catch (IOException ex) {  
    Logger.getLogger(Control.class.getName()).log(Level.SEVERE, null, ex);  
}  
respuesta = Integer.parseInt(rp);  
if (respuesta == res) {  
  
    FacesMessage mensaje = new FacesMessage(FacesMessage.SEVERITY_INFO,  
        "Respuesta Correcta " + res, null);  
    FacesContext.getCurrentInstance().addMessage("frmPrincipa", mensaje);  
}
```

## Interfaz multiplicación nivel medio

UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL  
UTE

UNIVERSIDAD  
TECNOLÓGICA EQUINOCCIAL

Basico Medio

MULTIPLICACION NIVEL INTERMEDIO

• REGRESAR

6 x 6 = 36

generar numeros

respuesta

Respuesta Correcta 36

Figura 52. Multiplicación Nivel Medio

## Botón respuesta código fuente

```
respuesta = Integer.parseInt(rp);
respuesta2 = Integer.parseInt(rp1);

numero1 = String.valueOf(respuesta);
numero2 = String.valueOf(respuesta2);

concatenado = numero1 + numero2;
res1 = Integer.parseInt(concatenado);

if (res1 == res) {

    FacesMessage mensaje = new FacesMessage(FacesMessage.SEVERITY_INFO,
        "Respuesta Correcta " + res1, null);
    FacesContext.getCurrentInstance().addMessage("frmPrincipa", mensaje);

} else {

    FacesMessage mensaje = new FacesMessage(FacesMessage.SEVERITY_INFO,
        "Respuesta Incorrecta "+ res1 + " ,Respuesta Correcta es " + res, null);
    FacesContext.getCurrentInstance().addMessage("frmPrincipa", mensaje);

}
```

## Interfaz división nivel básico

The screenshot displays the user interface for a basic division exercise. At the top left is the logo of Universidad Tecnológica Equinoccial (UTE). The header includes the university name and two navigation buttons: 'Basico' and 'Medio'. The main content area is titled 'DIVISION NIVEL BASICO' and contains a 'Regresar' link. The central part of the interface shows a math problem:  $2 / 1 =$  followed by a box containing the handwritten answer '2'. Below the problem are two buttons: 'generar numeros' and 'respuesta'. At the bottom, a blue message box with an information icon states 'Respuesta Correcta 2'.

Figura 53. División Nivel Básica

## Botón respuesta código fuente

```
String rp = "";  
res = num1 / num2;  
  
String sCurrentLine;  
BufferedReader br = null;  
br = new BufferedReader(new FileReader("C:\\\\Test\\num1.txt"));  
  
try {  
    while ((sCurrentLine = br.readLine()) != null) {  
        rp = sCurrentLine;  
        // System.out.println(rp);  
    }  
} catch (IOException ex) {  
    Logger.getLogger(Control.class.getName()).log(Level.SEVERE, null, ex);  
}  
respuesta = Integer.parseInt(rp);  
if (respuesta == res) {
```

## Pestaña división nivel medio



The screenshot displays the UTE (Universidad Tecnológica Equinoccial) interface. At the top left is the UTE logo. The main header reads "UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL". On the right, there are two buttons: "Basico" and "Medio". Below the header, the page title is "DIVISION NIVEL INTERMEDIO" with a "Regresar" link. The main content area shows a math problem:  $72 / 6 = 12$ . Below the problem are two buttons: "generar numeros" and "respuesta". At the bottom, a blue box with an information icon contains the text "Respuesta Correcta 12".

Figura 54. División Nivel Medio



## Botón respuesta código fuente

```
rna.MPerceptron1 uno = new MPerceptron1();
rna.MPerceptron dos = new MPerceptron();
int inputsCount = 256;
int outputsCount = 10;

// crear set entrenamiento
DataSet trainingSet = null;

try {
    trainingSet = TrainingSetImport.importFromFile(trainingSetFileName, inputsCount, outputsCount, ",")
} catch (IOException ex) {
    Logger.getLogger(MPerceptron1.class.getName()).log(Level.SEVERE, null, ex);
} catch (NumberFormatException ex) {
    Logger.getLogger(MPerceptron1.class.getName()).log(Level.SEVERE, null, ex);
}

// crear perceptron multicapa
MultiLayerPerceptron neuralNet = new MultiLayerPerceptron(TransferFunctionType.SIGMOID, 256, 5, 10);

// parametros set entrenamiento
MomentumBackpropagation learningRule = (MomentumBackpropagation) neuralNet.getLearningRule();
```

## 4.6 PRUEBAS

Las pruebas del sistema el primer prototipo fueron realizadas en la Prefectura de Imbabura situada en la ciudad de Ibarra. Las pruebas que se realizaron en dicho establecimiento fueron enfocadas en su mayoría al aspecto visual del sistema, es decir al diseño que se le dio al mismo para tener una interfaz amigable con el usuario al cual va a ser dirigido esta aplicación.

Para las pruebas que se realizaron se tuvo diferente tipo de usuarios que aplicaron al sistema y se dieron las siguientes conclusiones:

- El sistema no debe contener mucha carga de botones ni links externos ya que esto crea distracciones en los usuarios
- El sistema debe ser fácil de manejar y todo al alcance visual del usuario
- El sistema debe contener interfaces simples, sin muchos colores que opaquen el verdadero enfoque del programa

- El sistema debe mostrar la respuesta reconocida así como también si la respuesta esta incorrecta
- No debe contener un registro de usuario
- Debe contener una interfaz inicial donde solo conste nombre y apellido pero no genere ningún registro, dicha interfaz es solo informativa de ingreso de datos sin validar ningún registro de usuario.
- El sistema en las operaciones matemáticas debe ser generado lo más fácil posible sin números negativos, sin residuos.

En el centro de atención integral para personas con discapacidad se atienden alrededor de 50 pacientes por mes, las pruebas se realizaron con un promedio de 17 pacientes en distintas fechas y las pruebas no demoraron más de 10 minutos, las pruebas fueron realizadas en una sola máquina ya que el centro no cuenta con infraestructura informática.

El centro se dedica a dar atención en Medicina Física y Rehabilitación a los pacientes y no se instaló el sistema en ninguna de las maquinas administrativas con las que se tiene en el centro de atención integral.

## **CONCLUSIONES Y RECOMENDACIONES**

## **5 CONCLUSIONES Y RECOMENDACIONES**

A continuación se expone las conclusiones y recomendaciones que se obtuvieron en el desarrollo de la tesis sobre reconocimiento de caracteres manuscritos.

### **5.1 CONCLUSIONES**

- Al usar la librería Neuroph que brinda una compatibilidad con varias arquitecturas de redes neuronales, simplifica la labor al momento de manipular la red neuronal y sus respectivas pruebas, para el caso de la realización de esta tesis las redes que se utilizaron de esta librería son perceptrón multicapa y retro propagación.
- Al momento de entrenar a la red neuronal es importante que el patrón de entrenamiento en su mayoría sea distinto para tener mayor cantidad de posibilidades del reconocimiento.
- Al realizar las pruebas en la red neuronal se puede concluir que en la capa de salida en el set de entrenamiento debe contener valores únicos.
- El desempeño del procesamiento de la respuesta mejora sustancialmente en la medida en que se reduce la cantidad de datos en la capa de entrada.
- En la red neuronal para su correcto desempeño es necesario no perder datos significativos al momento de recuperar los valores del número dibujado, es decir el vector que alimenta la entrada para su comparación con la base de conocimiento.
- El modelo de prototipos ayuda mucho en este tipo de aplicaciones ya que se va desarrollando desde el prototipo inicial y una vez que este funcione correctamente se acopla los demás prototipos que contienen la estructura de los algoritmos similares y benefician al tiempo estimado de desarrollo de la aplicación.

- El uso de la herramienta java al ser código abierto, beneficia la interacción entre la librería Neuroph con el sistema web, sin necesitar de otras herramientas que pueden generar costos importantes en licencias.
- Se debe evitar perder valores significativos, es decir la recuperación del vector el cual se genera al dibujar en la matriz, debe contener todos los datos por donde este paso dentro del cuadro de dibujo.

## **5.2 RECOMENDACIONES**

- Para mejorar el prototipo inicial se puede probar con otros métodos inteligentes como WEKA o una fusión entre la librería Neuroph y WEKA para optimizar los resultados.
- Si se va a utilizar la librería Neuroph es necesario realizar una investigación previa sobre qué modelo que presenta la librería puede servir para optimizar el sistema ya que no todos contribuyen con los objetivos planteados.
- Al momento de dibujar el número en el sistema, se debe tratar de dibujarlo de manera correcta para que el sistema no confunda con otro número y genere una respuesta incorrecta.
- Es necesario para alimentar la capa de entrada de red neuronal reducir la matriz donde se dibuja el número, para que el procesamiento tenga mayor eficiencia y no demore en procesar en equipos que tengan pocos recursos de hardware.
- Se recomienda tomar en cuenta los valores de la capa oculta que provee la librería Neuroph para mejorar el desempeño de las capas de salida esto genera la mejoría en el reconocimiento de la respuesta.
- Para mejorar el dibujo del número, se recomienda utilizar dispositivos de entrada que sean táctiles, tanto para el entrenamiento como para el uso del sistema.

- Se recomienda tener un set de entrenamiento que contenga patrones dibujados de diferentes personas para ampliar la gama del diseño del número así el reconocimiento tendrá un mayor grado de acierto.
- Es recomendable entrenar la red con una cantidad de datos aceptable para que tenga un set de datos fiable con lo cual el reconocimiento sea más preciso.

## **BIBLIOGRAFÍA**

## **BIBLIOGRAFÍA**

Educarm. Curso temático Acceso a la Lecto – Escritura. Educarme PDF

Metodología Kumon. [www.kumon.com](http://www.kumon.com)

William Confer, Richard Chapman (2000). Handwritten Character Recognition for Cheap Devices. Auburn University, 106 Dunstan Hall, Auburn AL 36849, USA.

Marc Parizeau, Alexandre Lemieux, and Christian Gagné (2001). Character Recognition Experiments using Unipen Data. Laboratoire de vision et systèmes numériques(LVSN), Département de génie électrique et de génie informatique, Université Laval, Ste-Foy (Qc), Canada, G1K 7P4.

M. D. Garris, R. A. Wilkinson, and C. L. Wilson (1991). Methods for Enhancing Neural Network Handwritten Character Recognition. Advanced Systems Division National Institute of Standards and Technology.

Ajith Abraham (2004). Artificial Neural Network. Oklahoma State University, Stillwater, OK, USA

Trevor Hastiey Patrice Y. Simardz (1997). Metrics and Models for Handwritten Character Recognition. Statistics Department, Stanford University, CA94305.

Mundo inteligencia. [http://mundointeligencia.comuv.com/index\\_cono.html](http://mundointeligencia.comuv.com/index_cono.html)



Rumbaugh J, Jacobson I & Booch G,(2007). *El lenguaje unificado de modelado. Manual de referencia* –segunda edición, España, Editorial Pearson Education.

Xabier Basogain Olabe (2004), REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES, Escuela Superior de Ingeniería de Bilbao, EHU.

Jesús Alfonso López(2000). Redes Neuronales introducción y conceptos básicos. Inteligencia Computacional Tripod Colombia.

Mario Gómez Martínez, Tutorial sobre Redes Neuronales artificiales, Universidad Politécnica de Valencia 46022 Valencia, España

<http://www.lab.inf.uc3m.es/~a0080630/redes-de-neuronas/perceptron-multicapa.html>

Leonor Hernández López, Predicción y optimización de emisores y consumo mediante redes neuronales en motores diésel, Editorial Reverte, Barcelona 2006

Cuevas V.(2000).Procedimientos de Estrategias, Técnicas y métodos activos para activar los procesos del aprendizaje.

Raquel Flores López, Las redes neuronales artificiales fundamentos teóricos, aplicaciones prácticas, Editorial Netbiblo, La Coruña España 2008.

Daniel Behar, Metodología de la investigación, Editorial Shalom, Madrid España  
2008

<http://www.familias.apoclam.org/tecnicas-de-estudio-en-primaria.html>

<https://helpx.adobe.com/es/dreamweaver/using/web-applications.html>

Juan Jesús Romero, Inteligencia Artificial y Computación Avanzada, Editorial  
Fundación Alfredo Brañas, Santiago de Compostela España 2007

Antonio Goncalves, Beginning Java EE6 Platform with Glassfish 3 from Novice  
to Professional, Editorial Welsh, USA 2009.

Sevarac Z, Goloskokovic I, Tait J, Carter-Greaves L, Morgan A, Steinhauer

V, Koprivica M, Jovanovic I, Joksovic N,(2014). Java Neural Network  
framework, <http://neuroph.sourceforge.net/documentation.html>

Stuart Russell, Inteligencia Artificial un Enfoque Moderno, Editorial Pearson,  
España 2004

**ANEXO 1**  
**Manual de Usuario**

## Manual de Usuario

A continuación se presenta un breve manual de usuario de la aplicación.

Al ser una aplicación realizada en ambiente web, se puede acceder a ella mediante cualquier dispositivo que tenga señal de internet, el sistema debe estar alojado en un servidor.

La primera página que se nos presenta es la de inicio, en esta página se tiene el ingreso al menú principal, para el ingreso se debe ingresar nombre y apellido.



**UNIVERSIDAD**  
TECNOLOGICA EQUINOCCIAL

UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL  
**UTE**

Reconocimiento Caracteres Manuscritos

1	2	3	4	5
Uno	Dos	Tres	Cuatro	Cinco
6	7	8	9	10
Seis	Siete	Ocho	Nueve	Diez

**INGRESA**

Nombre: \* Luis

Apellido: \* Jimenez

**Ingresar**

En el momento que se ingresa carga la nueva página que es la que contiene el menú principal.

En esta página se encuentra las 4 operaciones fundamentales básicas, el usuario podrá ingresar a cualquiera de ellas.

La primera pestaña contiene el inicio que es la página donde se encuentra el menú, la segunda pestaña es el modulo suma, la tercera pestaña contiene el modulo resta, la cuarta pestaña contiene el modulo multiplicación para terminar con la quinta pestaña que es el módulo de la división.



## Módulo suma

Al acceder a la pestaña suma se despliega todo el modulo suma con sus 2 niveles que son básico en el cual el nivel de las sumas no sobrepasa un carácter para su respuesta y el nivel medio el cual el nivel de las sumas no sobrepasa los 2 caracteres en su respuesta.



El botón generar números genera los números aleatorios para que se ejecute la suma.



El botón respuesta es el que ejecutara todo el proceso interno de reconocimiento y generara la respuesta.

El sistema reconoce la respuesta que se le escriba, por lo que si no se escribe bien el numero el sistema generara la respuesta con los números que reconoció



Los siguientes módulos cumplen con el mismo proceso.

## Módulo resta



UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL  
UTE

UNIVERSIDAD  
TECNOLOGICA EQUINOCCIAL

Basico Medio

RESTA NIVEL BASICO

- REGRESAR

0 - 0 =

generar numeros

respuesta

En el módulo resta tenemos los botones de generar números que genera números aleatorios tomando en cuenta que no generará números que puedan dar respuestas negativas.



UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL  
UTE

UNIVERSIDAD  
TECNOLÓGICA EQUINOCCIAL

Basico Medio

RESTA NIVEL INTERMEDIO

• REGRESAR

93 - 47 = [ ] [ ]

generar numeros

respuesta

Y el botón respuesta que es el que dará la respuesta reconocida.

### Módulo multiplicación

UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL  
UTE

UNIVERSIDAD  
TECNOLÓGICA EQUINOCCIAL

Basico Medio

MULTIPLICACION NIVEL BASICO

• REGRESAR

1 x 8 = 8

generar numeros

respuesta

Respuesta Correcta 8

En el módulo multiplicación tenemos 2 botones, generar números, que es el botón que nos da la generación de los números que se van a multiplicar.



The screenshot shows the interface for the 'MULTIPLICACION NIVEL INTERMEDIO' module. At the top left is the UTE logo. The header includes 'UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL' and two buttons: 'Basico' and 'Medio'. Below the header, the title 'MULTIPLICACION NIVEL INTERMEDIO' is displayed, followed by a 'REGRESAR' link. The main area shows a multiplication problem:  $1 \times 67 =$  followed by two input boxes containing the handwritten number '22'. Below the problem are two buttons: 'generar numeros' and 'respuesta'. A feedback box at the bottom indicates: 'Respuesta Incorrecta 22, Respuesta Correcta es 67'.

Y el botón respuesta que es el que dará la respuesta reconocida.

### Módulo división



The screenshot shows the interface for the 'DIVISION NIVEL BASICO' module. At the top left is the UTE logo. The header includes 'UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL' and two buttons: 'Basico' and 'Medio'. Below the header, the title 'DIVISION NIVEL BASICO' is displayed, followed by a 'Regresar' link. The main area shows a division problem:  $1 / 1 =$  followed by one input box containing the handwritten number '1'. Below the problem are two buttons: 'generar numeros' and 'respuesta'. A feedback box at the bottom indicates: 'Respuesta Correcta 1'.

El módulo división contiene 2 botones, el botón generar número el cual genera números aleatorios tomando en cuenta que no proveerá residuo en la respuesta ni división para cero.

The screenshot shows the user interface of a division calculator. At the top left is the logo of Universidad Tecnológica Equinoccial (UTE). To the right of the logo, the text 'UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL' is displayed. Further right are two buttons labeled 'Basico' and 'Medio'. Below this header, the text 'DIVISION NIVEL INTERMEDIO' is shown, followed by a link 'Regresar'. The main area displays a division problem:  $48 / 2 = 24$ . Below the problem are two buttons: 'generar numeros' and 'respuesta'. At the bottom, a message box with an information icon states 'Respuesta Correcta 24'.

Y el botón respuesta que es el que dará la respuesta reconocida.

**ANEXO 2**  
**Carta de pruebas iniciales**